MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
TARAS SHEVCHENKO NATIONAL UNIVERSITY OF KYIV

Qualification scientific
work on manuscript rights

OLEKSANDR KUNICHIK

UDC 004.93

PHD THESIS

DEVELOPMENT OF A CROSS-PLATFORM MODEL OF THE UNIFIED ALGORITHMIC ENVIRONMENT FOR THE RECOGNITION OF EXPLOSIVE OBJECTS

121 Software Engineering

12 Information Technology

Applying for the Doctor of Philosophy degree

The PhD Thesis contains the results of own research. The use of ideas, results and texts of other authors are linked to the corresponding source

_____ Oleksandr Kunichik

Supervisor – **Vasyl Tereshchenko,** Doctor of Physical and Mathematical Science, Professor

KYIV-2025

ANNOTATION

Oleksandr Kunichik. Development of a cross-platform Model of the Unified Algorithmic Environment for the recognition of explosive objects. — *Qualifying scientific work as a manuscript.*

Thesis for the Doctor of Philosophy Degree in Specialty 121 "Software Engineering" (12 — Information Technology). — Taras Shevchenko National University of Kyiv, Kyiv, 2025.

Abstract

The persistent threat of landmines, explosive ordnance, and other explosive objects (EO) continues to claim lives, inflict devastating injuries, and impede socio-economic development in conflict-affected regions worldwide. EO comprise a variety of munitions containing explosives, including bombs, warheads, rockets, artillery shells, mines, torpedoes, as well as improvised explosive devices and other dangerous objects that can explode under certain conditions [12]. The Russian aggression against Ukraine has resulted in large-scale contamination of the territory with mines and unexploded ordnance (UXO). The process of demining and overcoming the consequences of the war is complicated by the large number of types of EO, the significant areas of contaminated territories, and the variety of mining methods and tactics used. The widespread presence of minefields makes it impossible to safely access land for agriculture, infrastructure development, and housing reconstruction, which significantly hinders the recovery of de-occupied areas. Mine action in Ukraine began in 2014, when Russia illegally annexed Crimea and launched the war in the Donbas. After the full-scale invasion in February 2022, Ukraine faced unprecedented levels of mine contamination and, according to preliminary estimates, became the most mined country in the world [13].

Traditional, mostly manual, demining methods remain essential, but are associated with high risk, significant time consumption, and low efficiency, especially in large areas. The search for and removal of EO is further complicated

by the wide variety of their types, their unpredictable and often concealed locations, and the diversity of landscapes and weather conditions encountered in contaminated areas. For example, dense vegetation, uneven terrain, and varying weather conditions can significantly hinder the detection process. In this context, innovative technologies, such as deep learning and computer vision methods, are opening up new opportunities to improve the effectiveness and safety of mine action.

To address the urgent global challenge of EO contamination, this thesis develops and evaluates a unified algorithmic environment for real-time EO detection, leveraging 3D printing, advanced data augmentation, deep learning, and a user-friendly cross-platform application. This research encompasses the creation of a comprehensive dataset using 3D-printed replicas of prevalent landmine types, the application of a novel two-stage data augmentation strategy, the training and optimization of a YOLO object detection model (tested were versions 5, 8, and 11), and the development of a cross-platform application, along with a messenger bot interface, for efficient and accessible EO detection.

The aim of this dissertation is to develop and evaluate a unified algorithmic environment for real-time EO detection that utilizes a cross-platform application, a messenger bot interface, and cloud-based processing. This research endeavors to provide an efficient and accessible method for EO detection by leveraging the scalability of cloud computing, the accessibility of mobile devices and messaging platforms, and the power of deep learning.

To accomplish this goal, the following objectives were established:

1. Develop methodologies to address the lack of data by using augmentation methods, particularly a two-stage augmentation approach.
2. Create a comprehensive dataset for training computer vision models that includes images of 3D-printed replicas of the most common anti-personnel landmines in Ukraine, obtained under different weather conditions (clear, cloudy, rain, snow, etc.).

3. Train and optimize YOLOv8 and YOLOv11 computer vision models on the created dataset by leveraging the increased number and diversity of images from the previous steps, applying the developed two-stage augmentation methodology, and fine-tuning model hyperparameters.
4. Evaluate the effectiveness of trained models on real landmine images from a distinct dataset obtained from professional deminers.
5. Design, implement, and evaluate a user-friendly cross-platform application capable of both online (utilizing the Google Cloud Platform API) and offline (using an optimized on-device model) EO detection.
6. Develop and integrate a messenger bot interface that interacts with the same Google Cloud Platform API for EO detection, providing an alternative access point to the system.
7. Implement the ability within the cross-platform application to correct recognition results by marking objects and selecting the correct EO type, contributing to ongoing model refinement.
8. Establish a secure data transmission mechanism within the application for sending operational data, including user feedback and corrections, to a central server to facilitate continuous improvement of the machine learning models.

The **object** of the present study is the Model of the Unified Algorithmic Environment (MUAE) [1] for unifying the process of recognizing explosive objects and its practical implementation in the form of a cross-platform application and messenger bot. The **subject** of research includes deep learning models for EO recognition, data augmentation methods, dataset creation technologies, data collection methods, as well as the architecture of a multi-platform EO detection system focused on online and offline operation.

[1]The Model of a Unified Algorithmic Environment (MUAE) is a conceptual model for a universal system employing a single algorithmic framework, shared data structures, and a common toolkit to solve a complex set of interrelated applied tasks.

Methodology

The research methodology employed in this thesis encompasses a multi-faceted approach, integrating techniques from data collection and curation to model development and system implementation. The key components of the methodology include:

— **Dataset Collection and Preparation:** This involves the acquisition of EO images through various methods, including the creation of a novel dataset based on 3D-printed replicas (Chapter 4), the collection of real landmine images from demining professionals, and the utilization of an expert data contribution platform integrated within the developed application (Chapter 6).

— **Data Augmentation:** To enhance the size and diversity of the training dataset, a two-stage data augmentation strategy is employed, as detailed in Chapter 3. This includes both basic transformations (e.g., rotation, scaling, flipping) and advanced techniques (e.g., MixUp, CutMix, mosaic).

— **Deep Learning Model Development and Adaptation:** The research focuses on the development, optimization, and adaptation of deep learning models for landmine detection. Initially, the YOLOv8 architecture was employed. The process included hyperparameter tuning, model training, and validation using the augmented dataset of 3D-printed replicas. Subsequently, in Chapter 7, the methodology was extended to incorporate a new, previously untrained EO type using the YOLOv11 model, demonstrating the system's adaptability to evolving real-world scenarios.

— **Cross-Platform Application Development:** A cross-platform application is developed using the Qt framework, QML, and C++ (Chapter 6). The application incorporates the trained YOLO models for both online (cloud-based) and offline (on-device) detection, and provides fea-

tures for user interaction, data annotation, and feedback.

— **Cloud Infrastructure and Distributed Systems:** The system leverages the Google Cloud Platform (GCP) for scalable and reliable operation. This includes the use of Google Cloud Functions for serverless computing, Google Cloud Storage for data storage, and Google Firestore for database management.

— **Object-Oriented Programming:** The software components of the system, including the cross-platform application and some parts of the cloud infrastructure, are implemented using object-oriented programming principles in C++ and Python to ensure modularity, maintainability, and code reusability.

— **Evaluation and Validation:** The performance of the developed models and the application is evaluated using standard metrics such as precision, recall, mAP, and others (Section 2.9). Experiments are conducted on both synthetic datasets (3D-printed replicas) and real landmine images to assess the system's effectiveness and generalizability.

At the outset of this research, a comprehensive review of existing studies in the field of EO detection and recognition was conducted. This review identified the primary challenges faced by researchers in developing state-of-the-art EO recognition methods based on computer vision. The analysis revealed that while deep learning models offer the potential for high accuracy and speed in the recognition process, their effective training is critically hindered by the severe scarcity of high-quality training data in this domain. To address the issue of limited data, this study employed 3D printing to create models replicating the visual characteristics of anti-personnel landmines commonly found in Ukraine. Furthermore, the role of data augmentation techniques in enhancing the performance of deep learning models was investigated. A wide range of augmentation methods were considered, from basic spatial transformations and pixel-level adjustments to more advanced techniques such as MixUp and

CutMix.

This research proposes a two-stage augmentation strategy (see Section 3), which, as demonstrated, increased the recall of the YOLOv8 model from 89.2% (without augmentation) to 92.6% (with two-stage augmentation) on the test dataset. The methodology was initially developed for the YOLOv5 model and subsequently successfully adapted to newer versions – YOLOv8, and to YOLOv11. A significant advantage of the two-stage augmentation method was demonstrated when transitioning from the YOLOv8 model to YOLOv11. Specifically, applying the mosaic type augmentation in both the first and second stages improved precision from 95% to 98% and recall from 92% to 97% (see Table 7.1). These results underscore the effectiveness of the YOLO architecture, particularly when combined with the developed two-stage augmentation strategy. However, the study also highlights the challenges and limitations associated with data augmentation in EO detection, such as the risk of creating irrelevant or harmful augmentations, over-reliance on augmentation, and the need to ensure a balanced representation of all classes in the training dataset. The results of this stage of the research will be used in the following sections, which focus on creating a dataset based on 3D-printed synthetic images of landmines and developing a cross-platform system that utilizes a mobile app for both EO detection (recognizing objects in images) and data collection (allowing in-app annotation and upload) to improve the model.

In accordance with the formulation of the strategy for applying augmentation methods (see Section 3), an investigation was conducted into existing methods for creating datasets of EO images. Among the approaches considered for dataset creation, the use of 3D printing to create replicas of common EO types and the photographing of these replicas was selected. The resulting 3D-printed landmine replicas were then employed to create a dataset, with manual annotation of the data including the marking of object boundaries (bounding boxes). Subsequently, experiments were conducted involving the training of deep learn-

ing models, specifically YOLOv8, where the previously developed methodology for applying augmentation methods was also employed. The trained models were then tested on images of real landmines obtained from professional deminers. The evaluation of this dataset revealed that while the models exhibited the capacity to recognize EO, further enhancement was necessary, particularly through the augmentation of the training dataset with additional images obtained from professional deminers. The efficacy of the synthetic dataset, comprising 1,438 photographs of 3D-printed replicas of five prevalent EO types, in training the YOLOv8 model was substantiated, attaining a precision of 98.0% and a recall of 98.2% on the test set.

The research findings underscore the potential of 3D printing in creating diverse and representative training datasets for computer vision models, offering a safe, ethical, and cost-effective approach for both research and practical training. Despite the high performance on synthetic data, a difference in effectiveness was observed when compared to the performance on real landmine images, underscoring the necessity for further advancements in the 3D printing process and data generation. The analysis of individual class performance reveals substantial variations in precision and recall across different types of landmines, underscoring the need for further investigation into the features utilized by the model for classification and methods to enhance the model's capacity to discern between EO and visually similar background objects and debris.

The next stage involved developing a cloud-based service with a messenger bot interface to provide practical access to the trained EO detection models. The service integrates with a popular messenger platform, Telegram, that supports bot creation. This platform was chosen for its large user base, robust bot API, and end-to-end encryption capabilities. The system utilizes the trained YOLOv8 model and integrates it with the GCP for efficient processing and scalability. The bot is designed to identify various EO types with high accu-

racy and provides users with supplementary information through integration with Google Gemini. This feature enhances user understanding of the detected threats and contributes to raising public awareness about the safety measures associated with explosive objects.

To improve the performance metrics of the models and expand the scope of their practical application, a multi-platform application has been developed that is capable of operating on a variety of operating systems. The application utilizes previously developed and trained models. The application has been tested with the help of volunteers, including professional deminers and military engineers. The developed cross-platform application for real-time EO detection is a valuable tool for demining teams, humanitarian organizations, and civilians in explosive-affected areas. The application supports both online and offline modes, using deep learning models. On the test dataset, the application demonstrates a recall of 88% with an average processing time of 2.1 seconds per image. An important feature of the application is the ability for users to directly adjust the recognition results. This allows for the collection of feedback data to correct model errors and augment the training set. The application also provides for the transmission of detection results, including operational logs, local database copies, and images (both original and user-modified) to a remote server. The data obtained in this manner serves as a valuable source of information for the further improvement of the models. Future development initiatives include expanding recognition to new types of EO, improving the existing model based on collected data and user feedback, and the creation of a training module. Continuous development and improvement of this application can make a significant contribution to global demining efforts and enhance the safety of communities.

Building on the previous sections, the developed software components, including the cross-platform application and messenger bot, together form a unified algorithmic environment. This environment supports the centralized train-

ing, deployment, and ongoing refinement of deep learning models for a wide range of EO detection tasks, including user education, real-time detection, and the collection of user-annotated data for continuous model improvement.

The research encompasses several key stages. Firstly, a comprehensive review of existing methods for detecting EO was conducted, along with an in-depth analysis of the challenges associated with data limitations and potential solutions. Secondly, various deep learning model architectures, specifically YOLOv5, YOLOv8, and YOLOv11, were developed, trained, and comparatively analyzed. Thirdly, a software package, including a cross-platform application and a messenger bot, was developed, and its effectiveness was evaluated under conditions closely approximating real-world scenarios. The results of this research are expected to make a significant contribution to solving the problem of mine hazards, thus contributing to restoring the safety and well-being of the population in Ukraine and other affected regions.

While the primary focus of the research is on computer vision-based detection methods using visible light images, the proposed approaches can be adapted for use with other types of sensors, such as infrared cameras, lidars, magnetometers, and ground-penetrating radar (GPR). The effectiveness of the developed methods is evaluated through training and testing of machine learning models on the collected and augmented datasets, using the metrics described in Section 2.9.

The scientific novelty of the study is as follows:

For the first time:

- A new algorithmic platform based on the Model of the Unified Algorithmic Environment (MUAE) for Explosive Objects (EO) detection has been developed, based on common principles of data processing, unified methods of augmentation, dataset creation, annotation and tuning of hyperparameters of deep learning models. The MUAE platform integrates a cross-platform application with offline and semi-automatic

annotation capabilities, a messenger bot, and a cloud API, providing data collection, training, deployment, and iterative model improvement.

- An offline EO detection capability integrated into a cross-platform application has been implemented, enabling the detection of explosives on mobile devices without an internet connection.
- A semi-automatic annotation mechanism integrated into a cross-platform application, allowing users to correct recognition results (add, delete, change labels and boundaries of objects) and send annotated data for further model improvement.
- A cloud API based on GCP has been created for EO recognition tasks, which allows online recognition and is integrated with a cross-platform application and a messenger bot; in addition to recognition, this bot provides additional information about the recognized objects using the large Google Gemini language model.

Improvements were made to:

- The methodology for forming training datasets using 3D-printed copies that reproduce the visual characteristics of EO common in Ukraine was developed. The trained models were then tested on a separate set of real explosive objects provided by demining specialists and volunteers.
- A methodology for applying data augmentation is proposed by developing and implementing a two-stage augmentation optimized for the YOLO family of models (YOLOv5, YOLOv8, YOLOv11).

The theoretical significance of the obtained results is as follows:

- **Extension of the methodology for creating training data:** The efficacy of a methodology for generating training datasets for EO recognition based on 3D-printed replicas has been experimentally validated, paving the way for the generation of controlled and secure data for training deep learning models. This methodology can be adapted to other computer vision tasks where there is a shortage of real-world data.

- **Improvement of application of augmentation methods:** A two-stage data augmentation strategy was developed and validated for the YOLO family of models. This strategy exhibited a substantial enhancement in the recognition completeness rate, a critical factor for EO detection.
- **Development of the concept of a single algorithmic environment:** A Model of Unified Algorithmic Environment (MUAE) was proposed and implemented to integrate different stages of recognition system development (data collection, training, deployment, feedback) into a single, manageable process. This model has the potential to serve as a foundational framework for the development of analogous systems in other application domains.

The Practical Value of the results obtained is the development of a ready-to-use software package for the recognition of EO. This software package has the following advantages:

- **Increased efficiency and safety of humanitarian demining:** A cross-platform application with offline recognition capability allows deminers to quickly identify EO on mobile devices directly in the field, even in the absence of an internet connection. The average image recognition time is 2.1 seconds, which ensures near real-time operation.
- **Simplifying the process of training and informing:** The messenger bot with the integrated Google Gemini language model can be used to train deminers, military personnel and civilians by providing information on the types of EO and rules of safe behavior.
- **Ensures fast data collection and updating:** A semi-automatic annotation mechanism integrated into the application allows for the rapid collection of data on new types of EO, recognition errors, and peculiarities of real-world conditions, which contributes to the continuous improvement of models.

- **Ability to integrate with other systems:** The developed cloud API can be used both independently for online recognition and integrated into other systems, for example, to automate the demining process using unmanned aerial vehicles (UAV) and robots.
- **Potential for scaling and adaptation:** The developed system can be adapted to recognize other types of threats (not only EO) and for use in other regions.

The results of this study have significant potential for practical application in the field of humanitarian demining, especially in Ukraine and other regions affected by armed conflicts. The developed cross-platform software system, which integrates innovative methods of data collection and analysis, contributes to solving the urgent problem of mine risk and will positively impact the reconstruction and development of the affected areas.

The proposed data collection and augmentation approaches, including the innovative use of 3D printing to create realistic EO models, enable the creation of high-quality and representative training sets. This improves the accuracy and reliability of the machine learning models, which is critical for effective and accurate EO detection and localization.

The developed cross-platform application facilitates the involvement of a wide range of experts and volunteers in the data collection process, ensuring the data is up-to-date and relevant to real-world conditions. This enables a rapid response to changes in the mine situation and increases the efficiency of mine action.

The integration of the trained machine learning models into a multi-platform software system creates a powerful tool for automating the process of detecting and locating EO. This will significantly reduce risks to personnel involved in demining operations and accelerate the clearance of areas from mines and other unexploded objects. Furthermore, the software package can be used to monitor and control the effectiveness of demining operations and to plan further

activities.

The use of the developed system will accelerate the recovery of the affected regions by enabling access to agricultural land, restoring infrastructure, and creating conditions for sustainable economic development. Reducing the number of casualties among civilians and soldiers by improving the accuracy and safety of EO detection is a crucial humanitarian outcome of this research.

The results and developed technologies have the potential for wide application not only in Ukraine but also in other countries facing the problem of mine hazards. The proposed approaches can be adapted to different types of terrain and conditions, making them a versatile tool for humanitarian demining. Thus, this thesis makes a significant contribution to the global problem of mine hazards and helps to restore the safety and well-being of the population in conflict-affected regions.

Personal contribution of the applicant: The dissertation is an independent scientific work that highlights the author's ideas and developments that allowed him to solve the tasks. The work contains theoretical and methodological provisions and conclusions formulated by the author personally. The ideas, provisions, or hypotheses of other authors used in the dissertation have appropriate references and are used only to support the applicant's ideas. The author conducted the research and experiments independently, the created software product is entirely the result of the applicant's work.

Keywords: Machine Learning, Deep Learning, Artificial Intelligence, Computer Vision, Humanitarian Demining, Explosive Objects (EO) Recognition, Data Augmentation, Two-step Augmentation, YOLO, Landmine Recognition, Cross-platform application, Cloud Computing, Model of Unified Algorithmic Environment (MUAE), Messenger Bot, Mobile application.

АНОТАЦІЯ

Кунічік О.В. Розробка кросплатформної Моделі Єдиного Алгоритмічного Середовища для розпізнавання вибухонебезпечних предметів. Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 121 «Інженерія програмного забезпечення» (12 — Інформаційні технології). — Київський національний університет імені Тараса Шевченка, Київ, 2025.

Забруднення територій наземними мінами та вибухонебезпечними предметами (ВНП) є однією з найсерйозніших проблем, що постають перед світом сьогодні, несучи загрозу життю людей та перешкоджаючи соціально-економічному розвитку постраждалих від конфліктів регіонів. До ВНП належать різноманітні боєприпаси, що містять вибухові речовини, зокрема: бомби, боєголовки, ракети, артилерійські снаряди, міни, торпеди, глибинні бомби, а також саморобні вибухові пристрої та інші небезпечні предмети, здатні вибухати за певних умов [12]. російська агресія проти України призвела до масштабного забруднення території мінами та боєприпасами, що не розірвалися. Процес розмінування та подолання наслідків війни ускладнює велика кількість різновидів ВНП, значні площі забруднених територій, а також різноманіття застосованих методів і тактик мінування. Широке застосування мінних загороджень унеможливлює безпечний доступ до земель для ведення сільського господарства, розвитку інфраструктури та відбудови житла, що суттєво стримує відновлення деокупованих територій. Протимінна діяльність в Україні розпочалася у 2014 році, коли росія незаконно анексувала Крим та розв'язала війну на Донбасі. Після повномасштабного вторгнення в лютому 2022 року Україна зіткнулася з безпрецедентним рівнем мінного забруднення, ставши, за попередніми оцінками, найбільш замінованою країною у світі [13].

Традиційні, переважно ручні, методи розмінування залишаються не-

замінними, проте вони пов'язані з високим ризиком, значними часовими витратами та низькою ефективністю, особливо на великих площах. Пошук та знешкодження ВНП ускладнюється широким спектром їх типів, непередбачуваністю розташування, а також різноманіттям ландшафтів та погодних умов. У цьому контексті інноваційні технології, зокрема методи глибокого навчання та комп'ютерного зору, відкривають нові можливості для підвищення ефективності та безпеки протимінної діяльності.

Ця дисертація спрямована на вирішення нагальної глобальної проблеми забруднення місцевості вибухонебезпечними предметами шляхом розробки нового уніфікованого алгоритмічного середовища для виявлення ВНП у реальному часі. Дослідження зосереджене на створенні комплексного набору даних з використанням 3D друкованих копій поширених типів ВНП, застосуванні передових методів доповнення даних, навчанні та оптимізації моделі виявлення об'єктів YOLO (було протестовано версії 5, 8 та 11), розробці крос-платформного додатка, а також месенджер-бота для доступного та ефективного виявлення ВНП.

Метою цієї дисертаційної роботи є розробка та оцінка уніфікованого алгоритмічного середовища для виявлення ВНП у режимі реального часу, що використовує крос-платформний додаток, інтерфейс месенджер-бота та хмарну обробку даних. Це дослідження має на меті запропонувати ефективний та доступний метод виявлення ВНП, використовуючи масштабованість хмарних обчислень, доступність мобільних пристроїв та платформ обміну повідомленнями, а також потужність глибокого навчання.

Для досягнення цієї мети були поставлені наступні завдання:

1. Розробити методології вирішення нестачі даних шляхом використання методів аугментації, зокрема, двоетапного підходу до аугментації.
2. Створити вичерпний набір даних для навчання моделей комп'ютерного зору, що включає зображення 3D друкованих копій найпоши-

реніших в Україні протипіхотних мін, отриманих за різних погодних умов (яскраве сонце, хмарність, дощ, сніг тощо).

3. Навчити та оптимізувати моделі комп'ютерного зору YOLOv8 та YOLOv11 на створеному наборі даних, використовуючи збільшену кількість та різноманітність зображень, отриманих на попередніх етапах, застосовуючи розроблену двоетапну методологію аугментації та точно налаштовуючи гіперпараметри моделі.
4. Оцінити ефективність навчених моделей на реальних зображеннях ВНП з окремого набору даних, отриманого від професійних саперів.
5. Спроектувати, реалізувати та оцінити зручний крос-платформний додаток, здатний як до онлайн (з використанням API Google Cloud Platform), так і до офлайн (з використанням оптимізованої моделі на пристрої) виявлення ВНП.
6. Розробити та інтегрувати інтерфейс месенджер-бота, який взаємодіє з тим самим API Google Cloud Platform для виявлення ВНП, забезпечуючи альтернативну точку доступу до системи.
7. Реалізувати в рамках крос-платформного застосунку можливість виправлення результатів розпізнавання шляхом маркування об'єктів та вибору правильного типу ВНП, що сприятиме постійному вдосконаленню моделі.
8. Створити безпечний механізм передачі даних у додатку для надсилання операційних даних, включаючи відгуки користувачів та виправлення, на центральний сервер для сприяння безперервному вдосконаленню моделей машинного навчання.

Об'єктом дослідження є процес розпізнавання вибухонебезпечних предметів (ВНП), що включає побудову, навчання, оптимізацію та розгортання відповідних моделей глибокого навчання, а також створення Моделі Єдиного Алгоритмічного Середовища (МЄАС) для інтеграції цих моделей у крос-платформенний застосунок та месенджер-бот.

Модель Єдиного Алгоритмічного Середовища (МЄАС) – це алгоритмічна архітектура, що дозволяє створювати універсальні системи для розв'язання комплексу взаємопов'язаних задач на базі єдиної алгоритмічної платформи або каркасу (framework). В основі МЄАС лежить використання спільних структур даних та уніфікованого набору алгоритмічних інструментів і процедур, що забезпечує єдиний підхід до вирішення різноманітних проблем у межах однієї системи. Така архітектура може бути реалізована різними способами, наприклад, через рекурсивно-паралельний алгоритм або як мульти-алгоритмічна платформа, але ключовим залишається спільне середовище, яке уможливлює ефективне розв'язання широкого класу прикладних задач шляхом уніфікації та перевикористання обчислювальних компонентів.

Предмет дослідження є моделі глибокого навчання для розпізнавання ВНП, методи аугментації даних, технології створення наборів даних, методи збору даних, а також архітектура мультиплатформної системи виявлення ВНП, орієнтованої на роботу в режимах онлайн та офлайн.

Методологія

Застосована в цій дисертації методологія дослідження охоплює багатогранний підхід, що поєднує методи збору й обробки даних, розробки моделей та впровадження систем. Основні складові методології включають:

— **Збір та підготовка даних:** Це включає отримання зображень ВНП різними методами, зокрема створення оригінального набору даних на основі 3D друкованих копій (Розділ 4), збір зображень реальних мін від професійних саперів, а також, потенційно, використання краудсорсингової платформи, інтегрованої в розроблений додаток (Розділ 6).

— **Аугментація даних:** Для збільшення розміру та різноманітності навчального набору даних застосовується двоетапна стратегія аугментації, детально описана в Розділі 3. Вона включає як базові

трансформації (наприклад, обертання, масштабування, перевертання), так і просунуті методи (наприклад, MixUp, CutMix, мозаїка).

— **Розробка та адаптація моделей глибокого навчання:** Дослідження зосереджено на розробці, оптимізації та адаптації моделей глибокого навчання для виявлення ВНП. Спочатку була використана архітектура YOLOv8. Процес включав налаштування гіперпараметрів, навчання моделі та перевірку з використанням доповненого набору даних 3D друкованих копій. Згодом, у Розділі 7, методологію було розширено для включення нового, раніше не тренованого типу ВНП з використанням моделі YOLOv11, демонструючи адаптивність системи до мінливих реальних сценаріїв.

— **Розробка крос-платформного застосунку:** Крос-платформний застосунок розроблено з використанням фреймворку Qt, QML та C++ (Розділ 6). Застосунок включає навчені моделі YOLO як для онлайн (на основі хмарних сервісів), так і для офлайн (на пристрої) виявлення ВНП, а також надає можливості для взаємодії з користувачем, анотації даних та зворотного зв'язку.

— **Хмарна інфраструктура та розподілені системи:** Система використовує Google Cloud Platform (GCP) для забезпечення масштабованості та надійності роботи. Це включає використання Google Cloud Functions для хмарних обчислень, Google Cloud Storage для зберігання даних та Google Firestore для управління базами даних.

— **Об'єктно-орієнтоване програмування:** Програмні компоненти системи, включаючи крос-платформний застосунок та деякі частини хмарної інфраструктури, реалізовані з використанням принципів об'єктно-орієнтованого програмування на C++ та Python для забезпечення модульності, зручності обслуговування та можливості повторного використання коду.

— **Оцінка та валідація:** Ефективність розроблених моделей і засто-

сунку ретельно оцінюється з використанням стандартних метрик, таких як точність, повнота, mAP та інших (Розділ 2.9). Експерименти проводяться як на синтетичних наборах даних (3D друковані копії), так і на реальних зображеннях ВНП для оцінки ефективності та узагальнювальної здатності системи.

На початку дослідження було проведено огляд існуючих робіт у галузі виявлення та розпізнавання вибухонебезпечних предметів (ВНП), що дозволило виокремити основні проблеми та виклики, з якими стикаються дослідники при розробці сучасних методів розпізнавання ВНП на основі комп'ютерного зору. Аналіз виявив, що моделі глибокого навчання здатні забезпечити високу точність та швидкість розпізнавання, проте критичною перешкодою для їх ефективного навчання є гостра нестача якісних навчальних даних у цій предметній області. Для подолання проблеми обмежених даних було використано 3D друк моделей, що відтворюють візуальні характеристики протипіхотних мін, поширених в Україні. Також було досліджено роль методів аугментації (штучного розширення даних) у покращенні результатів роботи моделей глибокого навчання. Розглянуто широкий спектр методів аугментації, від базових просторових перетворень і перетворень на рівні пікселів до просунутих методів, таких як MixUp, CutMix.

Було запропоновано двоетапну стратегію аугментації (див. Розділ 3), яка, як було продемонстровано, підвищила повноту (recall) моделі YOLOv8 з 89,2% (без аугментації) до 92,6% (з двоетапною аугментацією) на тестовому наборі даних. Методологія була спочатку розроблена для моделі YOLOv5, а згодом успішно адаптована до новіших версій — YOLOv8, а також до YOLOv11. При переході від моделі YOLOv8 до YOLOv11 було продемонстровано значну перевагу двоетапного методу аугментації. Зокрема, застосування аугментації типу «мозаїка» як на першому, так і на другому етапі, покращило показники точності (precision) з 95% до 98%, а повноту з

92% до 97% (див. Таблицю 7.1). Отримані результати підкреслюють ефективність архітектури YOLO, особливо у поєднанні з розробленою двоетапною стратегією аугментації. Однак дослідження також висвітлює виклики і обмеження, пов'язані з аугментацією даних при виявленні ВНП, такі як ризик створення нерелевантних або шкідливих аугментацій, надмірна залежність від аугментації та необхідність забезпечення збалансованого представлення всіх класів у навчальному наборі даних. Результати цього етапу дослідження будуть використані у наступних розділах, присвячених створенню датасету на основі 3D друку синтетичних зображень ВНП і розробці крос-платформної системи, яка використовує мобільний додаток як для виявлення ВНП (розпізнавання об'єктів на зображеннях), так і для збору даних (з можливістю редагування та вивантаження на сервер в додатку) з метою вдосконалення моделі.

Після формулювання стратегії застосування методів аугментації (Розділ 3) досліджено існуючі методи створення наборів даних із зображень ВНП. Серед розглянутих методів створення датасетів обрано використання тривимірного друку для створення копій поширених типів ВНП та фотографування цих копій. Отримані 3D друковані копії використовуються для створення датасету, проводиться ручна анотація даних з розміткою меж об'єктів (bounding boxes). Потім проводяться експерименти з навчання моделей глибокого навчання, зокрема YOLOv8, де також використовується розроблена раніше методика застосування методів аугментації. Отримані моделі тестуються на зображеннях справжніх ВНП, отриманих від професійних саперів. На основі результатів тестування робиться висновок, що, хоча отримані моделі і демонструють можливість розпізнавання ВНП, їх необхідно продовжувати покращувати, зокрема шляхом розширення навчального набору даних за допомогою додаткових зображень, отриманих від професійних саперів. Створений синтетичний набір даних, що складається з 1438 фотографій 3D друкованих копій п'яти поширених типів ВНП,

продемонстрував свою ефективність у навчанні моделі YOLOv8, досягнувши точності 98,0% і повноти 98,2% на тестовому наборі даних.

Результати дослідження також підкреслюють потенціал 3D друку у створенні різноманітних і репрезентативних навчальних наборів даних для навчання моделей комп'ютерного зору, пропонуючи безпечний, етичний та економічно ефективний підхід як для досліджень, так і для практичної підготовки. Незважаючи на високі показники на синтетичних даних, виявлено різницю в ефективності при порівнянні з показниками на реальних зображеннях ВНП, що підкреслює необхідність подальшого вдосконалення процесу 3D друку та генерації даних. Аналіз продуктивності окремих класів також показує значні відмінності в точності та повноті розпізнавання між різними типами ВНП, що наголошує на необхідності подальшого дослідження ознак, які використовуються моделлю для класифікації, і методів покращення здатності моделі розрізняти наземні ВНП і візуально схожі фонові об'єкти та сміття.

Наступним етапом стала розробка хмарного сервісу з інтегрованим інтерфейсом месенджер-бота, що забезпечує практичний доступ до навчених моделей виявлення ВНП. Цей сервіс інтегровано з популярною платформою обміну повідомленнями Telegram, яка підтримує створення ботів. Такий вибір платформи зумовлений її широкою розповсюдженістю, надійним API для створення ботів та можливостями наскрізного шифрування. Система використовує навчену модель YOLOv8 та інтегрована з GCP для ефективної обробки даних і масштабованості. Бот розроблений для ідентифікації різних типів ВНП з високою точністю та надає користувачам додаткову інформацію завдяки інтеграції з Google Gemini. Ця функція покращує розуміння користувачем виявлених загроз і сприяє підвищенню обізнаності про мінну безпеку.

З метою покращення показників роботи моделей та розширення можливостей їх практичного застосування, було розроблено мультиплатфор-

мений додаток, здатний функціонувати в різних операційних системах. У додатку використовуються раніше розроблені та навчені моделі. Тестування додатку проводилося за допомогою волонтерів, серед яких були професійні сапери та військові інженери. Створений крос-платформний додаток для виявлення ВНП у режимі реального часу є цінним інструментом для груп розмінування, гуманітарних організацій та цивільного населення на територіях, що постраждали від ВНП. Додаток підтримує роботу як в онлайн, так і в офлайн-режимах, використовуючи навчені на 3D друкованих копіях ВНП моделі глибокого навчання. На тестовому наборі даних додаток демонструє повноту розпізнавання 88% при середньому часі обробки одного зображення 2,1 секунди. Важливою особливістю додатку є можливість коригування результатів розпізнавання безпосередньо користувачем. Це дозволяє збирати дані зворотного зв'язку для виправлення помилок моделі та поповнення навчальної вибірки. Додаток також забезпечує передачу результатів виявлення, включаючи журнали роботи, локальні копії баз даних, а також зображення (як оригінальні, так і модифіковані користувачем) до віддаленого серверу. Отримані дані є цінним джерелом інформації для подальшого вдосконалення моделей. Майбутні ініціативи з розвитку включатимуть розширення кількості типів ВНП, що розпізнаються, вдосконалення існуючої моделі з урахуванням зібраних даних та відгуків користувачів. Постійна розробка і вдосконалення цієї програми може зробити значний внесок у глобальні зусилля з розмінування і підвищити безпеку громад.

Спираючись на попередні розділи, розроблені програмні компоненти, включаючи крос-платформний додаток та месенджер-бот, разом утворюють єдине алгоритмічне середовище. Це середовище підтримує централізоване навчання, розгортання та постійне вдосконалення моделей глибокого навчання для широкого спектру завдань виявлення ВНП, включаючи навчання користувачів, розпізнавання в режимі реального часу та збір ано-

тованих користувачами даних для безперервного покращення моделі.

Дослідження включає декілька ключових етапів. По-перше, проведено комплексний огляд існуючих методів виявлення ВНП, а також поглиблений аналіз проблем, пов'язаних з обмеженістю даних, та потенційних шляхів їх вирішення. По-друге, розроблено та проведено порівняльний аналіз різних архітектур моделей глибокого навчання, зокрема YOLOv5, YOLOv8 та YOLOv11. По-третє, розроблено програмний комплекс, що включає крос-платформний додаток та месенджер-бот, та проведено оцінку його ефективності в умовах, наближених до реальних. Очікується, що результати цього дослідження зроблять вагомий внесок у вирішення проблеми мінної небезпеки, сприяючи відновленню безпеки і добробуту населення в Україні та інших постраждалих регіонах.

Основна увага в дослідженні приділяється методам виявлення на основі комп'ютерного зору, що використовують зображення у видимому світлі. Проте, запропоновані підходи можуть бути адаптовані для застосування з іншими типами сенсорів, такими як інфрачервоні камери, лідари, магнітометри та георадари. Ефективність розроблених методів оцінюється шляхом навчання та тестування моделей машинного навчання на зібраних наборах даних, які було розширено із застосуванням описаних у Розділі 3 методів аугментації, з використанням метрик, описаних у Розділі 2.9.

Наукова новизна одержаних результатів полягає у наступному:

1. Вперше розроблено нову алгоритмічну платформу на основі Моделі Єдиного Алгоритмічного Середовища (МЄАС) для розпізнавання вибухонебезпечних предметів (ВНП), що базується на загальних принципах обробки даних, єдиних методиках аугментації, створення наборів даних, анотування та налаштування гіперпараметрів моделей глибокого навчання. МЄАС інтегрує крос-платформний застосунок (з можливостями офлайн-роботи та напівавтоматичного анотування), месенджер-бот та хмарний API, забезпечуючи збір даних,

навчання, розгортання та ітеративне вдосконалення моделей.

2. Вперше реалізовано функцію офлайн розпізнавання ВНП, інтегровану в крос-платформенний застосунок, що дозволяє проводити виявлення вибухонебезпечних предметів на мобільних пристроях без підключення до мережі Інтернет.
3. Вперше реалізовано інтегрований у крос-платформний застосунок механізм напівавтоматичного анотування, що дозволяє користувачам коригувати результати розпізнавання (додавати, видаляти, змінювати мітки та межі об'єктів) і відправляти анотовані дані для подальшого вдосконалення моделі.
4. Вперше для задач розпізнавання ВНП створено хмарний API на базі GCP, що забезпечує онлайн розпізнавання та інтегрований з крос-платформним застосунком і месенджер-ботом; цей бот, окрім розпізнавання, надає додаткову інформацію про виявлені об'єкти за допомогою великої мовної моделі Google Gemini.
5. Запропоновано методику формування навчальних наборів даних шляхом використання 3D-друкованих копій, що відтворюють візуальні характеристики ВНП, поширених в Україні, з подальшим тестуванням навчених моделей на окремому наборі реальних ВНП, наданих фахівцями з розмінування та волонтерами.
6. Запропоновано методику застосування аугментації даних шляхом розробки та впровадження двоетапної аугментації, оптимізованої для моделей сімейства YOLO (YOLOv5, YOLOv8, YOLOv11).

Теоретичне значення одержаних результатів полягає у наступному:

- **Розширення методології створення навчальних даних:** Запропоновано та експериментально підтверджено ефективність методології створення навчальних наборів даних для розпізнавання ВНП на основі 3D-друкованих реплік, що дозволяє генерувати контрольовані та безпечні дані для навчання моделей глибокого на-

вчання. Ця методологія може бути адаптована для інших задач комп'ютерного зору, де існує дефіцит реальних даних.

- **Вдосконалення методів аугментації:** Розроблено та валідовано двоетапну стратегію аугментації даних, оптимізовану для моделей сімейства YOLO. Ця стратегія продемонструвала значне покращення показників повноти розпізнавання, що є критично важливим для виявлення ВНП.
- **Розвиток концепції єдиного алгоритмічного середовища:** Запропоновано та реалізовано Модель Єдиного Алгоритмічного Середовища (МЄАС), яка забезпечує інтеграцію різних етапів розробки систем розпізнавання (збір даних, навчання, розгортання, зворотний зв'язок) в єдиний, керований процес. Ця модель може слугувати основою для створення подібних систем в інших прикладних областях.

Практичне значення одержаних результатів полягає у розробці готового до впровадження програмного комплексу для розпізнавання вибухонебезпечних предметів (ВНП), що має такі переваги:

- **Підвищення ефективності та безпеки гуманітарного розмінування:** Крос-платформний застосунок з можливістю офлайн-розпізнавання дозволяє саперам оперативно ідентифікувати ВНП на мобільних пристроях безпосередньо в польових умовах, навіть за відсутності інтернет-з'єднання. Середній час розпізнавання зображення становить 2.1 секунди, що забезпечує роботу в режимі, близькому до реального часу.
- **Спрощення процесу навчання та інформування:** Месенджер-бот з інтегрованою мовною моделлю Google Gemini може бути використаний для навчання саперів, військовослужбовців та цивільного населення, надаючи інформацію про типи ВНП та правила безпечної поведінки.

- **Забезпечення швидкого збору та оновлення даних:** Інтегрований у застосунок механізм напівавтоматичного анотування дозволяє оперативно збирати дані про нові типи ВНП, помилки розпізнавання та особливості реальних умов, що сприяє постійному вдосконаленню моделей.
- **Можливість інтеграції з іншими системами:** Розроблений хмарний API може використовуватись як самостійно для онлайн розпізнавання, так і бути інтегрованим у інші системи, наприклад, для автоматизації процесу розмінування з використанням БПЛА та роботів.
- **Потенціал для масштабування та адаптації:** Розроблена система може бути адаптована для розпізнавання інших типів загроз (не лише ВНП) та для використання в інших регіонах.

Результати цього дослідження мають значний потенціал для практичного застосування у сфері гуманітарного розмінування, особливо в Україні та інших регіонах, що постраждали від збройних конфліктів. Розроблений крос-платформний програмний комплекс, який інтегрує інноваційні методи збору та аналізу даних, сприяє вирішенню нагальної проблеми мінної небезпеки та матиме позитивний вплив на відбудову та розвиток постраждалих територій.

Запропоновані в роботі підходи до збору та аугментації даних, включаючи інноваційне використання 3D друку для створення реалістичних моделей ВНП, дозволяють формувати високоякісні та репрезентативні навчальні набори. Це покращує точність та надійність моделей машинного навчання, що є критично важливим для ефективного та точного виявлення і локалізації ВНП.

Розроблений крос-платформний застосунок сприяє залученню широкого кола фахівців та волонтерів до процесу збору інформації про ВНП, забезпечуючи її актуальність та відповідність реальним умовам. Це дозволяє

оперативно реагувати на зміни в мінній обстановці та підвищити ефективність протимінної діяльності.

Інтеграція навчених моделей машинного навчання у багатоплатформний програмний комплекс створює потужний інструмент для автоматизації процесу виявлення та локалізації ВНП. Це дозволить суттєво знизити ризики для особового складу, залученого до операцій з розмінування, та значно прискорити процес очищення територій від мін та інших вибухонебезпечних предметів. Крім того, програмний комплекс може бути використаний для моніторингу та контролю ефективності операцій з розмінування, а також для планування подальших заходів.

Використання розробленої системи сприятиме пришвидшенню відбудови постраждалих регіонів, відкриваючи доступ до сільськогосподарських угідь, відновлюючи інфраструктуру та створюючи умови для сталого економічного розвитку. Зменшення кількості жертв серед цивільного населення та військовослужбовців завдяки підвищенню точності та безпеки виявлення ВНП є найважливішим гуманітарним аспектом дослідження.

Отримані результати та розроблені технології мають потенціал для широкого застосування не лише в Україні, але й в інших країнах, які стикаються з проблемою мінної небезпеки. Запропоновані підходи можуть бути адаптовані до різних типів місцевості та умов, що робить їх універсальним інструментом для гуманітарного розмінування. Таким чином, ця дисертаційна робота робить вагомий внесок у вирішення глобальної проблеми мінної небезпеки та сприяє відновленню безпеки і добробуту населення у постраждалих від війни регіонах.

Особистий висновок здобувача: Дисертація є самостійною науковою працею, в якій висвітлені власні ідеї і розробки автора, що дозволили вирішити поставлені завдання. Робота містить теоретичні та методичні положення і висновки, сформульовані дисертантом особисто. Використані в дисертації ідеї, положення чи гіпотези інших авторів мають відповідні по-

силання і використані лише для підкріплення ідей здобувача. Автор провів дослідження і експерименти самостійно, створений програмний продукт є повністю результатом роботи дисертанта.

Ключові слова: Машинне навчання, Глибоке навчання, Штучний інтелект, Комп'ютерний зір, Гуманітарне розмінування, Розпізнавання вибухонебезпечних предметів (ВНП), Доповнення даних, Двоетапна аугментація, YOLO, Розпізнавання мін, Крос-платформний застосунок, Хмарні Обчислення, Модель Єдиного Алгоритмічного Середовища (МЄАС), Месенджер-Бот, Мобільний додаток.

LIST OF PUBLICATIONS OF THE APPLICANT ON THE TOPIC OF THE DISSERTATION

In accordance with the findings of the thesis, a total of 11 scientific papers were published, including two in a Ukrainian periodical scientific and professional publication that is indexed by the Scopus database and included in category "A" of the List of Scientific and Professional Publications of Ukraine; three in a Ukrainian professional publication of category "B"; and six in the materials of reports of conferences and abstracts.

Scientific papers confirming the approbation of the dissertation materials (included in the list of Ministry of Education and Science of Ukraine):

1. O. Kunichik and V. Tereshchenko, "Analysis of modern methods of search and classification of explosive objects," *Artif. Intell. Intell. Syst.*, vol. 27, no. 2, pp. 52–59, 2022.
2. O. Kunichik and V. Tereshchenko, "Improving the accuracy of landmine detection using data augmentation: a comprehensive study," *Artif. Intell. Intell. Syst.*, vol. 28, no. 2, pp. 42–54, 2023.
3. O. Kunichik, "Review of strategies to overcome the lack of data in landmine detection," *Artif. Intell. Intell. Syst.*, vol. 29, no. 3, pp. 99–103, 2024.
4. O. Kunichik and V. Tereshchenko, "Determining the effectiveness of using three-dimensional printing to train computer vision systems for landmine detection," *East.-Eur. J. Enterp. Technol.*, vol. 5, no. 1(131), pp. 17–29, 2024.
5. O. Kunichik, "Landmine detection with a mobile application," *East.-Eur. J. Enterp. Technol.*, vol. 6, no. 2(132), pp. 6–13, 2024.

Scientific papers confirming the testing of the thesis materials:

1. O. Kunichik and V. Tereshchenko, "Analysis of modern methods of search and classification of explosive objects," in *Proc. 22nd Int. Sci. Tech. Conf. "Artif. Intell. Intell. Syst. (AIIS'2022)"*, Kyiv, Ukraine, Dec. 2022.
2. O. Kunichik, "Detection of explosive objects," in *Proc. 21st Int. Sci. Pract. Conf. "Shevchenko Spring–2023"*, Kyiv, Ukraine, Apr. 2023, pp. 92–93.
3. O. Kunichik and V. Tereshchenko, "Enhancing landmine detection through the integration of 3D printing and deep learning techniques," in *Proc. 10th Int. Conf. "Inf. Technol. Implementation" (Satellite)*, Kyiv, Ukraine, Nov. 2023, pp. 28–29.
4. O. Kunichik and V. Tereshchenko, "Advancing landmine detection: a methodological breakthrough using 3D printing and computer vision," in *Proc. Int. Sci. Conf. "Artif. Intell.: Achievements, Challenges, Risks"*, Kyiv, Ukraine, Mar. 2024, pp. 447–450.
5. O. Kunichik, "Using a Telegram bot to detect landmines with artificial intelligence,"in *Proc. 24th Int. Sci. Tech. Conf. "Artif. Intell. Intell. Syst." (AIIS'2024)*, Kyiv, Ukraine, Oct. 2024, pp. 216–220.
6. O. Kunichik, "Mobile application for landmine detection," in *Proc. 11th All-Ukr. Sci. Pract. Conf. Students, Postgraduates Young Sci. "United Sci.: Prospects Interdisciplinary Res."*, Kyiv, Ukraine, Nov. 2024, pp. 311–313.
7. O. Kunichik, "Cloud-based landmine detection service with messenger bot integration," in *Proc. 23rd Int. Sci. Conf. "Neural Netw. Technol. Appl. NNTA-2024"*, Kramatorsk, Vinnytsia, Ternopil, Ukraine, Dec. 2024, pp. 216–220.

CONTENTS

List of Abbreviations **38**

INTRODUCTION **40**

Chapter 1. **The Present State of Research** **53**

1.1. Overview of Existing Approaches to Explosive Objects Detection 53

1.2. Applying Augmentation Methods 59

1.3. The Issue of Inadequate Data in Contemporary Studies 60

1.4. Review of Existing Software and Technologies for Explosive Objects Detection 64

1.5. Conclusions to Chapter 1 67

Chapter 2. **Setting the main objectives** **68**

2.1. Definitions of Explosive Objects 68

2.2. Relevance of the Problem 69

2.3. Estimation of the Number of Explosive Objects on the Territory of Ukraine 72

2.4. Obstacles in the Detection of Explosive Objects 75

2.5. A Methodology for Using Augmentation to Enrich Datasets . . 76

2.6. Methodology for Using 3D Printing to Create Datasets 77

2.7. Methodology for Using Expert Data Contribution to Improve Models 78

2.8. Selection of a Machine Learning Method 80

2.9. Formulas and Definitions 84

2.9.1. The Loss Functions 84

2.10. Conclusions to Chapter 2 92

Chapter 3. **The Application of Augmentation to Enhance the Quality of Datasets** **94**

3.1. The Necessity to Enhance Explosive Objects Detection 94

3.2. Common Augmentation Techniques 95

3.2.1. The Basic Augmentation Methodologies 95

3.2.2. Spatial Transformations 96

3.2.3. Pixel-Level Variations 97

3.2.4. Geometric and Morphological Transformations 100

3.3. Advanced Augmentation Techniques 101

3.3.1. Mosaic, MixUp, and CutMix 101

3.3.2. Augmentation Utilizing GANs 101

3.3.3. Sim2Real Augmentation 103

3.3.4. AutoAugment and Learned Augmentation Policies . . . 104

3.3.5. Adversarial Examples 104

3.3.6. Neural Style Transfer 105

3.4. Preprocessing Techniques 107

3.5. Dataset Description and Analysis 107

3.6. The Two-stage Augmentation Method 112

3.7. An Additional Set of Metrics to Assess the Performance of the Dataset. 116

3.8. Results of Applying the Two-Stage Augmentation Method . . . 121

3.9. Conclusions to Chapter 3 124

Chapter 4. **Addressing Data Scarcity in Explosive Objects Detection with 3D Printing** **127**

4.1. Addressing Data Scarcity with 3D Printing 127

4.2. Methodology: Using 3D Printing for Dataset Generation 128

4.3. Objective of the 3D Printing Technique 130

4.4. Primary Hypotheses of the Methodology 131

4.5. Rationale for Model Selection in Explosive Objects Detection . 133
4.6. The Process of 3D Printing and the Tools Used in the Study . . 135
4.7. Rationale for Selecting the YOLO Model 136
4.8. Dataset of 3D-Printed Explosive Objects 138
4.9. Preparing the Dataset: Annotation, Processing, and Augmentation 141
4.10. Training the YOLOv8 Model on the 3D-Printed Landmine Dataset 144
4.11. Results and Discussion 148
4.12. Evaluation on Real Landmine Data 150
4.13. Discussion 156
4.13.1. Summary of Findings 156
4.13.2. Comparison with Existing Research 157
4.13.3. Limitations 158
4.13.4. Future Research Directions 158
4.14. Conclusions to Chapter 4 160

Chapter 5. **Development and Deployment of a Cloud-Based Explosive Objects Detection Service with Messenger Bot Integration** **162**

5.1. Introduction: A Cloud-Based Explosive Objects Detection Service for Messenger Platforms 162
5.2. Related Work 163
5.3. Goals and Objectives 164
5.4. Architecture of the Cloud-Based Explosive Objects Detection System 165
5.5. Implementation Details 167
5.5.1. Messenger Bot Implementation 168
5.5.2. Cloud Function Implementation 169
5.5.3. The Explosive Objects Recognition Module Integration . 171
5.6. Integration with Google Gemini and Additional Bot Features . 173

5.6.1. Collaboration with Google Gemini 173
5.6.2. Target Audience and Deployment Context 174
5.6.3. Language Support and Additional Features 175
5.7. Conclusions to Chapter 5 175

Chapter 6. **Design, Implementation, and Evaluation of a Cross-Platform Application for Explosive Objects Detection** **177**
6.1. Introduction 177
6.2. Objectives of the Cross-Platform Application 178
6.3. The Subject and Primary Hypotheses During the Application Development 180
6.4. Technical Specifications: Hardware and Software Environment . 182
6.4.1. Hardware 182
6.4.2. Software Architecture and Technologies 183
6.5. System Design and Architecture 188
6.6. Explosive Objects Detection Model: YOLOv8 Implementation and Training 192
6.7. Operational Algorithm of the Cross-Platform Application . . . 193
6.7.1. Semi-Automatic Annotation Functionality 195
6.8. The Cross-Platform Application: Design, Functionalities, and Security 197
6.8.1. User Interface and Interaction 197
6.8.2. Data Security and Privacy 199
6.8.3. Technical Specifications 199
6.9. Performance Evaluation of the Mobile Application 200
6.9.1. Evaluation Methodology 201
6.9.2. Results of Recognition 201
6.9.3. User Refinement of Detection Results: The Edit Mode . 203
6.9.4. Data Transmission for Model Refinement 204

6.10. Discussion 206
6.10.1. Summary of Findings 206
6.10.2. Application's Main Features 207
6.10.3. Comparison with Existing Solutions 207
6.10.4. Limitations of the Current System 208
6.11. Conclusions to Chapter 6 209

Chapter 7. **Adding New Types of Explosive Objects** **211**
7.1. User Request and Requirements Gathering 211
7.2. Data Acquisition and Preparation 212
7.2.1. Video Acquisition 212
7.2.2. Data Annotation with Roboflow 213
7.3. Dataset Development and Augmentation 215
7.4. Model Training with YOLOv11 216
7.5. Model Deployment 219
7.5.1. Deployment to Google Cloud Platform (Messenger Bot) 219
7.5.2. Deployment to the Cross-Platform Application 219
7.6. Assessment and Verification 219
7.7. Conclusions to Chapter 7 220

Chapter 8. **Conclusions and Future Work** **223**
8.1. Future Research Directions 225

Bibliography **228**

List of Figures **244**

List of Tables **248**

ACKNOWLEDGEMENTS **249**

APPENDIX A. EXTENDED MODEL INFORMATION AND LANDMINE DATASET **250**

APPENDIX B. LIST OF THE APPLICANT'S PUBLICATIONS ON THE APPROVAL OF THE RESULTS OF THE DISSERTATION Articles in professional publications of Ukraine (included in the list of the Ministry of Education and Science of Ukraine) 254

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AP	Average Precision
API	Application Programming Interface
APL	Anti-Personnel Landmine
ATL	Anti-Tank Landmine
AXO	Abandoned Explosive Ordnance
CNN	Convolutional Neural Network
CW	Clockwise
CCW	Counter-Clockwise
DETR	DEtection TRansformer
EO	Explosive Objects
ERW	Explosive Remnants of War
FN	False Negative
FP	False Positive
GCP	Google Cloud Platform
GPR	Ground-Penetrating Radar
GPU	Graphics Processing Unit
GUI	Graphical User Interface
ID	Identification
IDE	Integrated Development Environment
IoU	Intersection over Union
JNI	Java Native Interface
MAE	Mean Absolute Error
mAP	Mean Average Precision
mAP50	Mean Average Precision at 50% IoU threshold

mAP50-95	Mean Average Precision averaged over IoU thresholds from 50% to 95% in steps of 5%
ML	Machine Learning
MSE	Mean Squared Error
ONNX	Open Neural Network Exchange
ORB	Oriented FAST and Rotated BRIEF
Qt	Cross-platform application development framework
QML	Qt Modeling Language
RAM	Random Access Memory
R-CNN	Regions with CNN features
SSE	Sum of Squared Errors
SSD	Single Shot MultiBox Detector
TP	True Positive
TN	True Negative
UAV	Unmanned Aerial Vehicle
UN	United Nations
UNMAS	United Nations Mine Action Service
UXO	Unexploded Ordnance
YOLO	You Only Look Once

INTRODUCTION

Background and Motivation

The persistent threat of Explosive Objects (EO) continues to inflict devastating human casualties and hinder socio-economic development in conflict-affected regions across the globe. EO pose a persistent and insidious threat to civilians, not only during periods of armed conflict but also for decades afterward. According to Landmine Monitor reports, 608 casualties were recorded due to landmines and explosive remnants of war (ERW) in Ukraine in 2022. Following the full-scale invasion by Russian forces in February 2022, the number of civilian casualties from landmines increased tenfold compared to the previous year, when there were at least 58 victims [14]. Of particular concern is the risk to children, who may be attracted to mines due to their innocuous or even toy-like appearance. The effective identification and clearance of EO are essential for the revitalization of war-torn regions. The impact of explosive objects on agriculture and infrastructure is also a major concern, as it renders vast swathes of land unusable.

As a result of the ongoing war, Ukraine has experienced significant contamination of its territories and is considered one of the most heavily mined countries in the world [13]. Landmines and ERW not only cause numerous civilian casualties but also limit access to agricultural land, impede infrastructure reconstruction, and hinder the country's economic development.

The evolution of methodologies employed in addressing the challenge of EO detection is chronologically delineated as such:

Early methods:

- **Traditional methods:** Initially, manual methods such as metal detectors and probes were used to detect EO [15]. These methods were

characterized by their inefficiency, danger, and slowness.

- **The advent of automation:** The evolution of technology subsequently gave rise to pioneering efforts to automate the process of EO detection [16,17]. These efforts employed an array of sensors and signal processing techniques, including acoustic, radar, and infrared methods, among others.

Modern Approaches:

- **The integration of artificial intelligence and machine learning:** In recent decades, machine learning (ML) and artificial intelligence (AI) have become pivotal instruments in EO detection. Many algorithms and models have emerged, capable of analyzing sensor data [29, 31, 34].
- **The employment of unmanned aerial vehicles (UAV)** has emerged as a significant tool for rapidly and safely surveying vast areas for EO [33, 35]. These vehicles can be equipped with a range of sensors and recognition systems, enabling effective detection of explosive objects.

Conventional demining techniques, while essential, are perilous, time-consuming, and ineffective on a large scale. The diversity of EO, their locations, and environmental factors further complicate the process of detection and clearance. The utilization of modern technologies, such as ML and AI, holds considerable promise in enhancing the efficiency and safety of demining operations. These technologies can significantly accelerate the demining process and mitigate the risk to operators during traditional demining.

The development of an algorithmic environment for recognizing explosive objects is a significant task, as it facilitates the integration of models for EO recognition, thereby significantly accelerating and diversifying the application of these models. The utilization of these models in software, mobile phones, and unmanned aerial vehicles (UAV) holds considerable promise. The developed software system facilitates EO detection and data collection for model training,

with numerous potential applications, including a training program to enhance EO detection skills and the transmission of detected and recognized explosive object coordinates to relevant authorities.

The deep learning models obtained using the techniques explored in the construction of the algorithmic environment can be used for humanitarian demining, for recognizing unknown explosive objects, for EO safety training, for professional training, for recognition from mobile devices, UAV, robots, and for online education.

The object of the present study is the Model of the Unified Algorithmic Environment (MUAE) for unifying the process of recognizing explosive objects. This includes the practical implementation in the form of a cross-platform application and messenger bot.

The Model of the Unified Algorithmic Environment is an algorithmic architecture designed to enable the creation of universal systems for addressing interconnected tasks, based on a singular algorithmic platform or framework. It utilizes shared data structures and a unified set of algorithmic tools, ensuring a consistent approach to diverse problems within a system. The model's implementation can be achieved through diverse methodologies, including recursive-parallel algorithms or multi-algorithmic platforms. The model's foundational principle is the establishment of a common environment, which is achieved through the unification and reuse of components, thereby facilitating the efficient resolution of a wide range of applications.

The subject of research includes deep learning models for EO recognition, particularly the YOLO architecture (versions 5, 8, and 11), data augmentation methods with an emphasis on the developed two-stage augmentation strategy, dataset creation technologies using 3D printing of EO replicas, data collection methods including an expert data contribution platform via the application, as well as the architecture of a multi-platform EO detection system focused on online and offline operation.

Research Methods

This thesis employs a multi-faceted methodology encompassing the acquisition and curation of visual data, the development and training of deep learning models, and the implementation of a cross-platform application and messenger bot interface. To address the scarcity of real-world EO data, a two-stage data augmentation strategy, as detailed in Chapter 3, is employed. This includes both basic transformations (e.g., rotation, scaling, flipping) and advanced techniques (e.g., MixUp, CutMix, mosaic). Subsequently, a novel dataset was created based on 3D-printed replicas of prevalent landmine types, as described in Chapter 4. The research focuses on the development, optimization, and adaptation of deep learning models for EO detection, specifically utilizing the YOLO architecture (versions 5, 8, and 11). This includes hyperparameter tuning, model training, and validation using the augmented dataset of 3D-printed replicas. Chapter 7 further details the pipeline for incorporating new, previously untrained EO types, demonstrating the system's adaptability to evolving real-world scenarios by training and deploying a YOLOv11 model. The development of a scalable distributed system using the Google Cloud Platform (GCP), including Google Cloud Functions, Google Cloud Storage, and Google Firestore, supports the application's online functionality and data management needs. A messenger bot, detailed in Chapter 5, provides an alternative user interface and integrates with Google Gemini for enhanced information retrieval. The cross-platform application, developed using the Qt framework, QML, and C++, incorporates the trained models for both online and offline EO detection and provides features for user interaction, data annotation, and feedback. The software components are implemented using object-oriented programming principles in C++ and Python to ensure modularity, maintainability, and code reusability. A platform for expert data contribution, integrated into the application, facilitates data collection from a wider user base of professionals. The performance of the developed models and the application is evaluated us-

ing standard metrics such as precision, recall, mAP, and others, as defined in Section 2.9.

Furthermore, the developed software package includes functionalities that enable continuous model improvement and dataset expansion. Specifically, the application allows users to correct recognition results, providing a mechanism for refining model predictions and generating additional annotated data. This user feedback, along with other operational data, is securely transmitted to a central server, contributing to the accumulation of a substantial and varied dataset for subsequent use in a range of EO detection applications.

The study aims to develop a program that utilizes computer vision with deep learning models for the recognition of explosive objects. The objective is to explore methods for generating datasets with limited data, leveraging augmentation and three-dimensional printing to enhance the dataset, and employing an expert data contribution platform to collect and evaluate models. The software package is being developed to analyze the outcomes of recognizing real mines, as collected by professional deminers, with subsequent refinement of the models through feedback.

The overarching objective of the study delineated the necessity to address the research tasks enumerated below:

- Conduct and analyze existing practical and theoretical studies in the field of explosive object recognition.
- Formulate the main directions of building datasets in the conditions of insufficient data.
- Analyze augmentation methods, analyze tools for creating datasets and building models.
- Develop a dataset for training computer vision models that will include images of 3D-printed copies of the most common anti-personnel mines in Ukraine, obtained in different weather conditions (clear, cloudy, rain, snow).

- Employ YOLO computer vision models, optimize their functionality, and augment the dataset, as described in the previous step. This process involves increasing the quantity of images, applying augmentation methods, and adjusting hyperparameters.
- Evaluate the effectiveness of the trained models on real EO images from the user's own dataset, obtained from professionals.
- The development and evaluation of a Telegram bot for EO detection, unrestrained by the choice of platform, is the eighth objective.
- The development and evaluation of a mobile application that utilizes deep learning models for EO detection is imperative. This application should facilitate EO recognition in both online and offline modes.
- Implement the ability to correct the recognition results by marking objects and selecting the type of EO.
- Provide the ability to transmit data on the program's operation for the purpose of enhancing ML models.

The aim of this thesis is to formulate and assess a unified algorithmic environment for real-time EO detection that incorporates a cross-platform application, a messenger bot interface, and cloud-based processing. This research aspires to offer an efficient and accessible method for EO detection by harnessing the scalability of cloud computing, the accessibility of mobile devices and messaging platforms, and the capabilities of deep learning.

To accomplish this objective, the following specific objectives have been established:

1. Develop and validate methodologies to address data scarcity using data augmentation techniques, with a focus on a novel two-stage augmentation approach.
2. Construct a comprehensive dataset for training computer vision models, comprising images of 3D-printed replicas of prevalent anti-personnel landmines found in Ukraine, captured under diverse environmental con-

ditions (e.g., clear, cloudy, rain, snow).

3. Train and optimize YOLOv8 and YOLOv11 computer vision models on the created dataset, leveraging its increased size and diversity, applying the developed two-stage augmentation methodology, and fine-tuning model hyperparameters for optimal performance.
4. Evaluate the generalization performance of the trained models on a distinct dataset of authentic landmine images obtained from professional demining teams.
5. Design, implement, and evaluate a user-friendly, cross-platform application capable of both online detection, utilizing the GCP API, and offline detection, employing an optimized on-device model.
6. Develop and integrate a messenger bot interface that interacts with the same GCP API to provide an alternative access point for EO detection.
7. Implement, within the cross-platform application, a mechanism for users to correct recognition results by annotating objects and selecting the correct EO type, thereby contributing to ongoing model refinement.
8. Establish a secure and privacy-preserving data transmission mechanism within the application for sending operational data, including user feedback and corrections, to a central server to facilitate continuous improvement of the ML models.

Scientific Novelty

This thesis introduces several novel contributions to the field of EO detection, integrating 3D printing technology, deep learning models, and an expert data contribution platform to enhance the safety and efficiency of demining operations.

The scientific novelty of the study is as follows:

For the first time:

- An expert data contribution platform was integrated into a cross-platform application for EO detection, enabling the collection of annotated data

for training and refining deep learning models. This platform empowers individuals (who are professionals in this field) to contribute to demining efforts through a user-friendly interface, significantly expanding the volume and diversity of the training data, and ultimately leading to more robust and reliable detection models.

- A cross-platform EO detection application with both online and offline capabilities was developed and evaluated. This application provides accessibility across various devices (e.g., smartphones, tablets, computers) and operational modes (online, offline), enabling deminers to utilize the technology in diverse field conditions, regardless of internet connectivity.
- A messenger bot interface was developed and integrated, which utilizes the cloud-based API for EO recognition and provides users with additional information about detected objects, obtained using Google Gemini's large language model.

Further advancements were made in:

- Augmentation methodologies through the development and implementation of a novel two-stage augmentation technique, specifically tailored for training YOLO models. This technique refines the YOLO object detection algorithm by incorporating a pre-training augmentation phase with targeted transformations (e.g., geometric distortions, noise injection, and mosaic augmentation) and an online augmentation phase during model training, resulting in improved model generalization and more accurate EO detection in varied and challenging environments.
- Methodologies for creating training datasets based on 3D-printed replicas that reproduce the visual characteristics of anti-personnel landmines prevalent in Ukraine. The effectiveness of this methodology was experimentally validated through subsequent testing of trained models on real EO images.

The practical value of the study lies in the development of a unified algorithmic environment for EO detection. This environment integrates a number of key components: a methodology for creating training data based on 3D-printed replicas, a two-stage data augmentation strategy, optimized deep learning models (YOLOv5, YOLOv8, and YOLOv11), a cross-platform application with offline and online detection capabilities, and a messenger bot interface with access to a cloud-based API and integration with a large language model for providing background information.

The proposed two-stage augmentation method provides high recognition accuracy, which is experimentally confirmed: accuracy increased from 89.2% to 92.6% on the test dataset consisting of images from the internet. The use of 3D-printed landmine replicas to create a training dataset has demonstrated high efficiency, achieving a precision of 98.0% and a recall of 98.2% on the test set of 3D-printed replicas and, importantly, ensures high accuracy in recognizing real EO (average precision of 91.0% and recall of 79.1% on an independent dataset of real EO). The developed cross-platform software package allows EO to be detected on many devices, including mobile phones, making it particularly useful in the field. Moreover, the developed software product is capable of detecting EO offline, which is extremely important, especially given that in areas contaminated by EO, there is usually poor or no internet coverage.

The application also supports the annotation of detected EO, allowing users to adjust the recognition results and send this data for further model improvement. The developed API, which is used by the application for online recognition, is also integrated into a messenger bot, which provides a user-friendly interface for interaction without the need to install additional software. The bot can recognize sent or forwarded photos and also provide background information about the detected EO using a large language model. The algorithmic environment also supports the ability to add new types of EO at the request of users. An example of adding new types of EO, training the model, and

deploying it is described in Section 7. The two-stage augmentation methods can be applied to different algorithms, as demonstrated by the transition from YOLOv8 to YOLOv11.

Possible Future Research Directions

Based on the results obtained in this study, several promising directions for future research are identified. These directions offer opportunities for researchers and developers to further advance and refine the technologies for detecting EO, including:

- **Expanding the Functionality and Application Domains of the Mobile Application.** The mobile application can be adapted for use by professional deminers and other specialists, providing the capability to recognize unknown EO in the field and ensuring rapid analysis and identification of potential threats. Furthermore, integrating the application into training programs for deminers and military personnel will allow for effective training on recognizing EO using both 3D-printed replicas and real specimens. The development of a specialized training mode within the application to simulate real-world EO detection scenarios is also being considered. In addition, the application can be adapted for use by the civilian population in areas with potential EO contamination, allowing them to identify suspicious objects and promptly inform the appropriate authorities by sending data directly through the application.
- **Further Development of the Unified Algorithmic Environment.** Future work involves further expanding and supplementing the dataset, including data obtained from users of the mobile application and messenger bot. This will enable the creation of more diverse and representative training sets to improve ML models. Also, it is a priority to integrate the developed models into robotic platforms and unmanned aerial vehicles (UAV) to automate the process of humanitarian demi-

ning, ensuring safe and efficient area surveying. In addition, optimized models for rapid online recognition integrated into various devices such as robots, UAV, and body cameras of military personnel are to be developed for real-time EO detection.

- **Expanding the Functionality of the Messenger Bot.** It is advisable to expand the bot's functionality for educational purposes, allowing a wide audience to obtain information about EO types, safety rules, and the identification of unknown objects. It is also promising to extend the messenger bot's support for platforms other than Telegram to reach a larger number of users.
- **Generalizing the Methodology.** A promising direction is to adapt the developed methodology, including 3D printing and two-stage augmentation, to create datasets and train models for recognizing other types of EO besides anti-personnel landmines. It is also worth exploring the possibilities of applying the developed approaches in related fields where there is a problem of insufficient training data, for example, to recognize rare objects or anomalies in images.
- **Technological Improvements.** Future work may be directed towards integrating data from different sensors (infrared cameras, lidars, magnetometers, ground-penetrating radars) to improve the accuracy and reliability of EO detection, especially under challenging conditions (e.g., hidden or camouflaged objects). Furthermore, it is advisable to carry out further optimization of deep learning models to increase processing speed and reduce hardware requirements, particularly for efficient operation in offline mode on mobile devices. A separate direction is the improvement of 3D models and their printing to enhance realism.

The implementation of these outlined future tasks will significantly improve the developed unified algorithmic environment and expand its scope of application, contributing to solving the problem of EO danger in Ukraine and the

world.

Personal contribution of the applicant

The dissertation is the product of the author's independent research endeavors. The scientific supervisor is to formulate the problem and make recommendations on the choice of methods for solving intermediate tasks.

The applicant has published 11 scientific papers [1–11]: 6 abstracts of conferences and 5 scientific articles. Creative contribution to published works is as follows:

- Articles and abstracts [2,3,7,8,10] are written in co-authorship with the supervisor. The work [8] is devoted to the refinement of augmentation methods and the creation of a methodology for increasing the amount of data in EO image sets. The works [2, 3, 10] are devoted to the use of three-dimensional printing to create deep learning models. The paper [7] is devoted to an overview of current research areas in the field of EO detection. The supervisor contributed to the reviewing of the text of the articles to bring them to the form of a scientific work. O. Kunichik conducted the main research after discussing possible areas of work.
- Abstracts [1] are devoted to the review and analysis of existing methods of searching for explosive objects.
- Abstracts [4] are devoted to the use of Telegram bot for EO detection.
- Abstracts [5] are devoted to the use of a mobile application for EO detection.
- Abstracts [6] are devoted to the use of cloud technologies and messengers for EO detection.
- Article [9] is devoted to an overview of strategies for overcoming data deficiencies in the creation of datasets for training deep learning models for EO detection.
- Article [11] is devoted to the use of a mobile application for EO detection.

- The problem statement and the development of conference abstracts were made with the support of the supervisor, the author made presentations at all conferences independently.

The research results were approbated at seven conferences:

1. The 22nd International Scientific and Technical Conference "Artificial Intelligence and Intellectual Systems" (AIIS'2022) — December 8–9, 2022 — Kyiv, Ukraine.
2. The 21st International Scientific and Practical Conference "Shevchenko Spring–2023" — April 15, 2023 — Kyiv, Ukraine.
3. The 10th International Conference "Information Technologies and Implementation" (Satellite) — November 21, 2023 — Kyiv, Ukraine.
4. The International Scientific Conference, "Artificial Intelligence: Achievements, Challenges, and Risks" — March 15–16, 2024 — Kyiv, Ukraine.
5. The 24th International Scientific and Technical Conference "Artificial Intelligence and Intellectual Systems" (AIIS'2024) — October 18–19, 2024 — Kyiv, Ukraine.
6. The 11th All-Ukrainian Scientific and Practical Conference of Students, Postgraduates and Young Scientists "United by Science: Prospects of Interdisciplinary Research" — November 21–22 — Kyiv, Ukraine (a publication only).
7. The 23rd International Scientific Conference "Neural Network Technologies and Applications NNTA-2024" — December 11–12, 2024 — Kramatorsk – Vinnytsia – Ternopil, Ukraine.

CHAPTER 1
THE PRESENT STATE OF RESEARCH

This section provides an overview of extant research on explosive objects detection using the latest technologies, the creation of ML models for detection, the use of mobile technologies, unmanned aerial vehicles (UAV), and other technologies that can speed up the demining process.

1.1. Overview of Existing Approaches to Explosive Objects Detection

Building upon the need for innovative EO detection solutions outlined in the Introduction, this chapter reviews existing technologies for explosive objects (EO) detection and clearance. Despite the need for innovation, conventional techniques, such as the use of metal detectors and manual probing with a sapper's probe, remain essential in many demining operations. However, these methods are often characterized by considerable time investment, requiring meticulous manual scanning of large areas, and present a significant risk to the safety of demining personnel, creating a need for faster, safer, and more efficient solutions. Consequently, there is a growing interest in exploring alternative approaches to EO detection and clearance. Current research in EO detection focuses primarily on the following technologies:

- **Acoustic-Seismic Methods:** Employ sound and seismic waves to detect mine-induced anomalies in the ground [18]. These methods analyze the changes in wave propagation caused by the presence of buried objects. While effective for shallowly buried EO, they are relatively slow and susceptible to noise interference.

- **Ground Penetrating Radar (GPR):** Uses radio waves to detect objects underground based on differences in electromagnetic properties [19, 20]. GPR transmits electromagnetic pulses into the ground and analyzes the reflected signals to create an image of subsurface objects. It offers a considerable penetration depth, yet it is susceptible to interference from non-mine objects with similar dielectric properties.
- **Electromagnetic Induction (EMI):** Detects the metal components in EO by measuring changes in an induced electromagnetic field [21]. EMI is a relatively cost-effective technique but is less reliable for EO with minimal metal content.
- **Infrared (IR):** Detects thermal contrasts between buried objects and their surroundings [22]. IR imaging relies on the difference in thermal conductivity between the object and the surrounding soil. However, its effectiveness is constrained by factors such as the depth of detection, the presence of vegetation, and diurnal temperature variations.
- **Nuclear Quadrupole Resonance (NQR):** Detects specific materials in explosives by analyzing their response to radio frequency pulses [23]. While highly accurate in identifying specific explosive compounds, NQR requires expensive and sophisticated equipment.
- **Thermal Neutron Activation (TNA):** Detects specific materials in explosives by analyzing gamma radiation emitted after neutron irradiation [24]. This method is also quite accurate but requires expensive and sophisticated equipment, including a neutron source.
- **Backscattered Neutron Method:** Involves analyzing scattered neutrons to detect explosives. This method is based on the principle that different materials scatter neutrons differently. However, it is sensitive to soil moisture variations [25].
- **X-ray Backscatter Method:** Utilizes X-rays to create an image of buried objects based on the backscattered radiation [26]. The intensity

of backscattered X-rays varies depending on the density and atomic number of the object. However, it should be noted that this method has a limited depth of penetration.

- **Magnetic Anomaly Detection (MAD):** Employs sensors to detect changes in magnetic fields caused by the presence of metallic objects [27]. This method is cost-effective and portable; however, its effectiveness is limited to metal mines and it is less effective in noisy environments.
- **Computer Vision-Based Methods:** Utilize cameras and image processing algorithms to detect visual features of EO [36]. These methods can be deployed on various platforms, including handheld devices, robots, and UAV. While offering advantages in terms of cost, accessibility, and potential for real-time detection, their effectiveness can be limited by factors such as lighting conditions, vegetation cover, and the visual similarity between EO and background objects.

Among these diverse approaches, computer vision-based methods, particularly those employing deep learning techniques, have emerged as a promising avenue for research. Their ability to learn complex patterns from visual data, coupled with advancements in processing power and algorithm design, offers the potential to overcome some of the limitations of traditional methods. The following section will delve into the rationale for selecting the YOLO architecture as the foundation for the EO detection system developed in this research.

In the article [28], the authors conducted a comprehensive review of various EO detection methods, meticulously analyzing their operational principles, strengths, and weaknesses. The paper provides a detailed exposition of the predominant methods and offers recommendations on the integration of approaches such as MAD and GPR to enhance efficiency and portability. The authors conducted a thorough analysis of the extant approaches to the search for explosive objects, encompassing a wide range of methods for detecting ob-

jects underground and diverse algorithms. The authors underscore the intricacies inherent in the processing of sensor data, attributable to the diversifying conditions prevalent during experimental iterations, and the necessity of calibrating equipment to specific parameters, including soil type, EO placement depth, prevailing weather conditions, and antenna frequency. The profusion of variables necessitated for EO detection engenders a requirement for substantial computing capacity. The article further provides insights into the financial implications of equipment, albeit with the caveat that the cited prices may be outdated. Nonetheless, this information facilitates cost estimation. It is important to note that the article does not contain original experimental data to support the conclusions, nor does it take into account the diversity of real-world environmental conditions, thus limiting the generalizability of the analysis. Additionally, the article focuses on traditional methods, which limits the overview of modern innovations.

One of the most prevalent methods for detecting explosive objects is GPR systems. The work [29] uses three neural networks to analyze radar images to determine the shape of an object underground, material, and additional features such as depth, size, etc. The authors observe that the prevailing focus in numerous contemporary studies is on the recognition of cylindrical objects, while the full classification of objects remains under-explored. The gprMax [30] program, a de facto standard in the field, was utilized to generate the dataset. The study's findings indicate that the shape of objects can be determined with an average accuracy of 90%, while material identification exhibits an average accuracy of 99% to 100%. Additionally, the depth estimation error ranged from 4% to 16%. While this study has yielded favorable outcomes, the approach employing multiple neural networks exhibits a lack of flexibility, necessitating the continual addition or modification of existing networks. The utilization of generated data is a common practice in this field; however, the employment of real data is crucial to optimize the approximation to reality. A prevalent

challenge in this subject area is the scarcity of sufficient samples in datasets.

The article [31] proposes a convolutional neural network (CNN)-based pipeline for detecting buried landmines using ground penetrating radar (GPR) B-scan data, achieving up to 95% accuracy with minimal preprocessing. A key strength of this approach is its ability to generalize well to real-world GPR data despite being trained primarily on synthetic datasets, addressing the challenge of limited labeled data. The system's reliance on characteristic hyperbolic signatures in B-scans simplifies the detection process, making it highly efficient and practical for deployment. The verification process utilizes 9 objects to acquire GPR images; however, the testing and setup process is cumbersome, hindering practical applications. The study utilizes gpr-max software [30] to generate data, enabling the creation of GPR images. The authors intend to conduct further experiments with antennas; however, it remains uncertain whether the images generated by gpr-max can be effectively utilized in real-world scenarios.

Nevertheless, the study's evaluation is constrained to a particular dataset, giving rise to concerns regarding its generalizability to varied soil and environmental conditions. Moreover, while the approach proves effective in detecting buried objects, it does not extend to full object classification or leverage multidimensional data.

The work [32] explores the search for explosive objects from varied angles, encompassing diverse types of pipes and a jug. However, the analysis is conducted manually within the Rad Explorer software package. A synthesis of the extant works employing GPR reveals the following conclusions:

- Insufficiently sized datasets.
- The predominant role of human involvement in training and object identification.
- The findings are substantiated through experimental means, employing datasets that are often limited, artificial, and fitted, thereby raising concerns about the reliability of the obtained accuracy.

- A paucity of classes is evident in object classification, necessitating either the redesign of algorithms or the addition of layers of neural networks for new classes.
- Deep learning approaches, such as CNN, are trained on synthetic datasets, which are generally ideal and may produce larger errors in real-world use.

Visual methods of detecting explosive objects with a camera are of the utmost importance in the context of the large number of landmines in the occupied and de-occupied territories of Ukraine, including in buildings, on the outside of structures, and in trees. The article [33] provides a model for constructing a cost-effective system used to search for explosive objects that are not visible on the surface. The algorithm utilizes two morphological image processing techniques, erosion and dilation, to refine input still images by removing irrelevant areas and enhancing EO detection accuracy through an area opening process. However, the study's experimental nature is constrained by two factors. Firstly, the dataset contains a limited number of objects, including the objects used (tuna cans), which limits the generalizability of the findings. Secondly, the lack of testing on real EO hinders the study's external validity.

In the article [34], the authors investigated various sensing techniques, image processing techniques, and recognition and classification techniques. The article serves as a foundational framework for future research, wherein the authors describe the algorithmic actions to achieve the objective of EO recognition using a neural network. The system demonstrates an accuracy of up to 90% in classifying EO under various conditions, such as rotation or partial closure. However, the absence of real-world testing and the reliance on successful segmentation hinder the system's practical application. The study encompasses a range of image sources, including ultrasound, conventional cameras, GPR, infrared cameras, and two types of mines utilized in a laboratory setting.

The work of Binghamton University [35, 37], which focuses on the recogni-

tion of a single mine (PFM-1) using a drone and a camera (visible light and infrared), is noteworthy. However, the utilization of a single EO for testing currently restricts the system's versatility. Nevertheless, this limitation does not preclude the possibility of conducting analogous experiments with other EO types.

Consequently, the problem of locating explosive objects using alternative classical approaches remains salient. This subject is being investigated from both theoretical and practical standpoints. However, most works consider some aspects of this problem. These works typically employ simplified tools for problem resolution, such as bulky equipment, laboratory experiments, limited datasets, and generated data. These tools do not fully consider the diversity of EO types, methods of placement, and weather conditions, which limits their effectiveness. This section will provide an overview of strategies to address the problem of data deficiency.

1.2. Applying Augmentation Methods

Given the heterogeneity of images and tasks, there is a necessity for specialized augmentation methods. To this end, numerous studies have developed frameworks and libraries, providing a comprehensive range of image augmentation methodologies. A notable contribution to the comprehensive survey of image augmentation techniques has been made by the study [38], which undertakes a thorough evaluation of the impact of these methods on fundamental computer vision tasks, including semantic segmentation, image classification, and object detection. The article presents a comprehensive review of image augmentation methods, classifying them into basic methods (e.g., flipping, rotation), advanced approaches (e.g., auto-enhancement, deep generative models), and their application in various computer vision tasks, such as image classification, object detection, and semantic segmentation. The study highlights

the strengths and weaknesses of these methods, demonstrates their positive impact on model performance, and addresses issues such as a lack of theoretical research, evaluation metrics, and scalability. The study concludes with conclusions on future directions, emphasizing the need for a better theoretical framework and more effective augmentation strategies.

The imgaug [39] library contains numerous techniques, including flipping, rotation, noise addition, contrast variation, and others that are employed in the study. Additionally, TensorFlow [40] introduced "Keras preprocessing layers," a module integrated into TensorFlow that facilitates resizing, scaling, rotation, flipping, and other image augmentation processes. This article also includes a practical guide that explains how to use these layers to process datasets and train models.

While these advances demonstrate the effectiveness of augmentation, issues such as scalability, evaluation metrics, and theoretical frameworks highlight areas for future research and development. The employment of augmentation to increase the quantity of data in a EO detection dataset constitutes a substantial instrument that facilitates the enhancement of dataset quality. This augmentation process entails the implementation of diverse transformations on images that are challenging to obtain systematically in real-world settings.

1.3. The Issue of Inadequate Data in Contemporary Studies

Convolutional neural networks (CNN) are a particular kind of artificial neural network that has been the subject of several studies due to its ability to process images and other data with spatial structure. CNN achieve this through the use of convolutional layers, which are specialized circuits that can detect local features in the input data. In paper [41], CNN are used to recognize landmines from magnetometer images, demonstrating their efficacy in this application. The research results presented herein illustrate the high accuracy

of EO detection from unmanned aerial vehicles (UAV), and the system was trained on an artificial dataset. However, the absence of a rigorous testing phase on real data calls into question its practical applicability. The reliance on generated images in the study underscores significant concerns regarding data security and the dearth of publicly accessible data. A notable drawback is the system's inability to recognize plastic landmines, highlighting the need for further refinement. In essence, the study fails to meet the fundamental criterion of providing reliable data for analysis, relying instead on the empirical observation that the generated data exhibits a high degree of similarity to real data. Furthermore, the system has not been subjected to a real-world testing environment.

The employment of machine learning algorithms in the domain of GPR-based EO detection represents a prevalent methodological approach. GPR functions by employing a ground-based antenna to transmit electromagnetic waves of limited duration into the soil, with the waves reflecting back when they encounter an underground object. The return signals are then examined to obtain an image or profile of subterranean structures. This methodological approach has been employed in numerous recent studies to achieve more precise and effective detection. In [42], GPR data underwent analysis using CNN, yielding a precision rate of over 93% in the detection of buried objects, including landmines. Nevertheless, concerns persist regarding the practical applicability of this system in EO detection contexts. A potential rationale for these concerns may be attributed to the experimental design of the study, which was conducted within a laboratory environment. The extrapolation of the proposed approaches' effectiveness to real-world scenarios remains uncertain. It is imperative to test and refine the methodology in real-world settings, considering the variable characteristics of soil types to ensure its effectiveness in diverse environments.

The study [43] conducted in Ukraine utilized CNN for the classification

of explosive objects, achieving a precision rate of 97.8%. Nevertheless, the study was predominantly theoretical in nature and relied on data previously collected [42], incorporating a dataset. In the article [44], a comprehensive survey of machine and deep learning algorithms utilized in GPR data processing is provided, with the authors emphasizing the challenge of limited dataset for training. The study's challenge domain is its failure to propose viable solutions to the limited data issue, though the authors hypothesize that synthesizing imagery with augmentation methods may offer a solution.

As demonstrated in study [45], the combination of GPR and CNN holds promise for detecting subterranean objects. The study's findings suggest the feasibility of identifying improvised explosive devices buried beneath roadways. However, the study's primary limitation is the paucity of test data. The development of a more extensive dataset could enhance the research, thereby aligning it more closely with practical real-world applications.

In [46], a robotic platform was utilized to detect landmines, and the findings suggested the feasibility of using GPR for landmine detection, though the experiment's complexity raised questions about its practicality. The study's primary limitations were the restricted dataset (comprising only two types of landmines) and the absence of independent dataset validation, which led to concerns regarding its applicability in real-world settings.

The augmentation of limited datasets has emerged as a promising solution to address this limitation. A study [8] demonstrated that data augmentation methods enhance the efficiency of deep learning models for EO detection, achieving 97.4% precision and 92.6% recall. However, the study's reliance on internet data, characterized by its limited variety, necessitates the incorporation of additional images to enhance the dataset's diversity.

Other recent studies have examined various methods of EO detection, including the use of UAV equipped with multispectral and thermal sensors [36]. However, this study's primary limitation is its focus on a single type of EO,

the dataset's modest size (165 images derived from six orthophotos), and the restriction to a limited range of environmental settings. One potential solution to these challenges lies in augmenting the dataset with additional images and in establishing more varied testing conditions.

In [47], an analysis of landmine installation patterns is conducted, exploring the potential for detecting landmines based on a map with other detected ones in proximity. An artificial dataset with patterns is generated to facilitate this study. However, the practical challenges of implementing this method persist, as the placement of landmines is influenced by various terrain, situational, and problem-specific factors, rendering a universal guideline for their deployment ineffective. This observation aligns with the conclusions drawn from previous studies, which indicate that artificially generated data does not adequately reflect the practical implementation of the outcomes.

Study [7] analyzes modern demining methods, demonstrating a broad array of detection techniques. However, the study does not address the issue of insufficient data for experiments, and the challenges of implementing the studied methods persist. This may be attributed to the time-consuming and intricate nature of establishing EO detection systems, as they were all conducted in laboratory settings with limited experimental data.

Consequently, a significant proportion of EO detection studies have been hindered by a paucity of data. To address this limitation, many surveys have resorted to generating synthetic imagery [31, 41, 47], utilizing limited datasets [36,44,46], or even reusing datasets from other studies [43]. Alternatively, three-dimensional computer modeling can be employed to address the challenge of inadequate data, as evidenced by the study on car traffic [48]. However, such data is characterized by a deficiency in realism. Augmentation techniques have also been explored [8], yet the generation of images or the augmentation of dataset size does not rectify the issue of data heterogeneity.

These considerations underscore the necessity for a comprehensive study

aimed at identifying methodologies to overcome the challenge of limited data during the training process of machine learning systems. A review of extant literature reveals that this deficit represents a significant impediment to the development of effective detection systems. Consequently, research endeavors directed towards addressing this concern through the utilization of 3D-printed landmine replicas for the generation of data for model training are highly pertinent. Furthermore, this approach would ensure the complete safety of the testing process, thereby eliminating the possibility of an accidental detonation of the experimental material.

1.4. Review of Existing Software and Technologies for Explosive Objects Detection

A comprehensive overview of contemporary demining technologies is presented in [49], including metal detectors, ground-penetrating radar, infrared sensors, magnetometers, and other advanced devices. This comprehensive study furnishes a meticulous exposition on the recent advancements in demining technologies. Notably, the study does not put forth any pragmatic solutions that could be implemented to facilitate the process of demining. As the work indicates, a limited number of projects have secured funding and advanced to the implementation stage. However, the study does not provide specific examples to support this assertion. A 2013 United Nations (UN) report [50] indicates numerous projects with potential were not realized due to a lack of financial resources or a method of aligning donors with technological capabilities. The aforementioned report, however, makes mention of the use of mobile applications in the context of EO safety training. Furthermore, it details plans for the development of an application designed for the training of individuals in the detection, handling, and disposal of improvised explosive devices. Consequently, initiatives have emerged to leverage mobile technologies in mine action;

however, these do not possess automatic recognition functionality.

In another study, potential applications of smartphones, in conjunction with artificial intelligence (AI), for uses in the medical industry were thoroughly investigated [51]. It is noteworthy that this domain bears a notable relation to that of EO detection, given the potential repercussions of erroneous judgments. The study reviews several mobile applications and discusses the potential of smartphone cameras. In this article, the authors illustrate the potential for utilizing AI to identify objects in the medical domain. However, the question of enhancing models by incorporating feedback from the application remains unresolved. The authors further underscore the challenges associated with on-device recognition, attributable to the intricate nature of the models. Addressing these complexities may be achieved by implementing server-based recognition as a solution. This approach bears resemblance to that delineated in reference [52]. However, it is hindered by its reliance on a desktop computer as a server, which restricts the system's mobility and its deployment in field settings where access to a desktop computer may be limited or nonexistent.

The authors of the study [51] observe that the advancement of object recognition software is hindered by the need for more data and the absence of feedback. The implementation of online recognition has been posited as a potential solution. This approach was explored in [53] in the context of a study examining the feasibility of employing a mobile application to identify explosive objects using virtual reality. However, it is important to note that the study does not employ its own dataset and model, a decision that limits its generalisability to a substantial number of EO. Instead of utilizing its own dataset, the application employs a third-party developed augmented reality software. The evaluation of performance is undertaken by the application's authors through the utilization of the accuracy metric. However, this metric is inadequate in providing a comprehensive assessment of the efficacy of EO detection models. In the context of EO detection, alternative metrics such as recall and F1-measure hold greater

significance, as these consider the integral role of false positives and false negatives in evaluating performance. The implementation of the application is achieved through three scripts that execute the primary functions. Nevertheless, challenges persist regarding feedback processes and the annotation of false positive results.

In terms of reporting instances of EO detection, the State Emergency Service of Ukraine's (SES) website offers a dedicated reporting option [54]. The SES website offers insights into the present state of demining operations, the extent of contaminated and occupied territories in Ukraine, and related matters. To report a EO, one is instructed to visit the service's website and to register using BankID, an electronic identification system. The BezpekaInfo website [55] contains a variety of resources focused on landmine safety. The overarching objective of the project is the dissemination of information in an accessible manner to a target audience of children. Additionally, an online course is available. Nonetheless, it is imperative to acknowledge that the efficacy of these resources hinges on enhancing their accessibility through mobile applications and offline functionality, thereby addressing the challenges faced by users in rural and disconnected regions [54, 55]. The process necessitates internet connectivity to access the intended information, which is currently only available through these websites.

In contrast, the paper [56] puts forth an alternative proposal of utilizing mobile devices as a substitute for metal detectors for EO detection. This innovative approach utilizes the smartphone's built-in magnetic sensor to detect EO. It should also be noted, however, as indicated by the author in the same publication, that the system remains in the experimental phase and has yet to be put into practical use. The economic viability of employing smartphones for EO detection is not yet established, as the use of a conventional sensor suffices.

Consequently, at the present moment in time, there is an utter dearth of software capable of detecting EO in images. Furthermore, existing online re-

sources are inadequate in terms of recognizing EO, highlighting the need for further research and development in this area to enhance the efficacy of EO detection methodologies. The integration of smartphones into the reconnaissance process holds significant potential for enhancing its efficiency and speed.

1.5. Conclusions to Chapter 1

The problem of finding explosive objects remains a significant concern, as evidenced by the substantial number of studies aimed at developing effective methods of detection and disposal.

The employment of data augmentation techniques has been shown to enhance the quality and diversity of datasets utilized for training machine learning models, thereby augmenting their efficacy.

Addressing the paucity of data in this domain is promising, and the use of three-dimensional printing methods to create realistic models of explosive objects for training and testing recognition systems is a particularly salient approach.

Presently, there is an absence of a universally applicable software solution capable of effectively recognizing diverse EO types and integrating data to enhance model performance.

Consequently, there is a pressing need for the creation of a centralized and algorithmic environment that would facilitate collaboration between data, datasets, algorithms and ML models. Such a framework would enable the integration of diverse approaches and software implementations, thereby ensuring the efficacy of ML models and the safety of individuals operating within complex, high-risk environments.

CHAPTER 2
SETTING THE MAIN OBJECTIVES

2.1. Definitions of Explosive Objects

The term "Explosive Objects" is generally understood to denote a broad spectrum of categories that encompass or potentially encompass explosive material, thereby constituting a potential hazard to human life and well-being. An examination of the diverse categories of EO reveals a multifaceted nature, and it is imperative to explore them in greater depth.

- Anti-vehicle EO, which are designed to destruction tanks, armored personnel carriers, and other heavy military vehicles. They are usually larger than anti-personnel landmines and contain a greater proportion of explosives.
- Anti-personnel EO, on the other hand, are defined as mines designed to cause death to humans. These EO can be categorized based on various factors, including high-explosive, fragmentation, jumping, and others.
- Direct-action anti-personnel EO are a category of mines that, upon detonation, propel a projectile of fragments in a predetermined direction, thereby increasing the radius of impact.
- Improvised explosive devices (IED) are a category of munitions that diverge from conventional military models. They can be fabricated from a wide range of materials and possess varying detonation mechanisms.
- Shells refer to munitions utilized for the purpose of artillery, such as those discharged from guns, howitzers, mortars, and similar weaponry. Notably, unexploded shells pose a persistent threat, as they can remain hazardous for extended durations due to the inherent delays in the

detonation process.

- Rockets are defined as self-propelled munitions that deliver explosives to a target via a rocket motor; unexploded rockets may still contain explosive components, while remnants from expended rockets may also pose a hazard.
- Cluster munitions refer to bombs or shells that contain a high number of submunitions, i.e., smaller bombs. These munitions are dispersed over a large area and often do not detonate immediately, thereby constituting a persistent threat.
- Explosive remnants of war (ERW) are a general designation for all types of explosive remnants of war. Such remnants may include unexploded bombs, shells, mines, grenades, or ammunition, amongst other types of explosives.

2.2. Relevance of the Problem

EO continue to pose a substantial and ongoing threat to global security, even in the aftermath of armed conflicts. A minimum of 10,254 individuals sustained injuries due to landmines and ERW worldwide during the period of 2021 – 2022, with 3,843 fatalities, 6,370 injuries, and 41 cases with unknown outcomes [14, 57]. In 85% of the documented cases, the victims were identified as civilians, with approximately 50% of these civilians being minors. The magnitude of the problem in mined regions is considerable, with Ukraine being a prime example. For instance, the State Emergency Service (SES) of Ukraine has identified approximately 128,000 square kilometers of mined land and 14,000 square kilometers of mined water, which have had a substantial and adverse impact on nearly six million inhabitants of Ukraine [58]. The evidence underscores the necessity for the use of modern demining techniques.

According to the Landmine Monitor [57], 608 individuals sustained injuries

from explosives in Ukraine in 2022. Following Russia's full-scale invasion, civilian casualties from landmines surged tenfold relative to 2021 (58 victims). A significant concern is the danger posed to children, who may be attracted to these hazardous things due to their frequently misleadingly innocuous appearance. The identification and subsequent removal of EO is imperative for the rehabilitation of war-affected areas, as these objects can severely compromise agricultural productivity and infrastructure, rendering large regions hazardous and inaccessible.

The restoration of normality in contaminated areas necessitates the capacity to identify explosive objects, for which a variety of approaches are employed, including manual demining and the deployment of canines for bomb detection [59]. The existing methodologies for EO detection are diverse, but they frequently encounter limitations. Conventional methods such as metal detectors and manual probing are both labor-intensive and hazardous. While advanced technology, including infrared imaging and ground-penetrating radar (GPR), holds promise, it can be costly and generate false positives. Canine units are constrained by factors such as fatigue and environmental conditions and require extensive training. A mobile EO detecting application can facilitate the identification of these objects by enabling the capture of photographs of potentially hazardous objects and the subsequent notification of the relevant authorities.

The utilization of AI, specifically ML algorithms, possesses the capability to efficiently identify EO; however, the employment of such models is presently restricted to a select group of demining experts. The integration of state-of-the-art technologies has the potential to markedly accelerate the demining process. However, the demining community persists in employing methods that are regarded as outdated and conventional, such as the use of metal detectors and bayonet-like tools (probes).

Computer vision offers significant potential to enhance the efficiency and safety of EO detection by leveraging advanced algorithms for analysis of both

conventional images and GPR and infrared images. It is noteworthy that certain landmines are composed of plastic, a material that conventional demining techniques, such as metal detection, are ineffective against. However, computer vision algorithms can effectively identify such EO. A significant challenge in developing ML models is the scarcity of data for algorithmic training and evaluation. The acquisition of authentic EO data for model training is fraught with ethical and security concerns, as even deactivated landmines can pose significant risks, making the procurement of authentic samples for research a perilous undertaking [60] .

Furthermore, it is imperative to enhance the capacity to identify unidentified explosive devices that sappers may encounter in the course of their duties. In such scenarios, mobile applications can serve as invaluable resources, as their feedback mechanism enables them to function as a data source when the model fails to accurately detect explosive objects. The development of mobile applications for EO detection using computer vision models represents a promising area of research. This is due to the widespread usage of mobile devices and the development of sophisticated ML techniques. These sophisticated machine learning techniques, in turn, facilitate the secure detection of EO, guide the training of demining professionals, improve model efficacy, and notify pertinent demining services.

Despite the pressing demand for dependable EO detection systems, numerous challenges impede advancement. Manual EO detection by deminers remains the primary method of demining, but the pace of this demining is inadequate. According to the most recent estimates from early 2024, it will require almost 700 years to clear the Ukrainian territories potentially polluted with explosive devices [61]. This projection is based on the current estimate of 500 demining groups, comprising approximately 5,000 specialists, and is expected to result in the clearance of 4,700 square kilometers over the span of 20 years. Consequently, it is projected that 174,000 square kilometers potentially contaminated

with EO can be cleared within a period of 740 years (Figure 2.1).

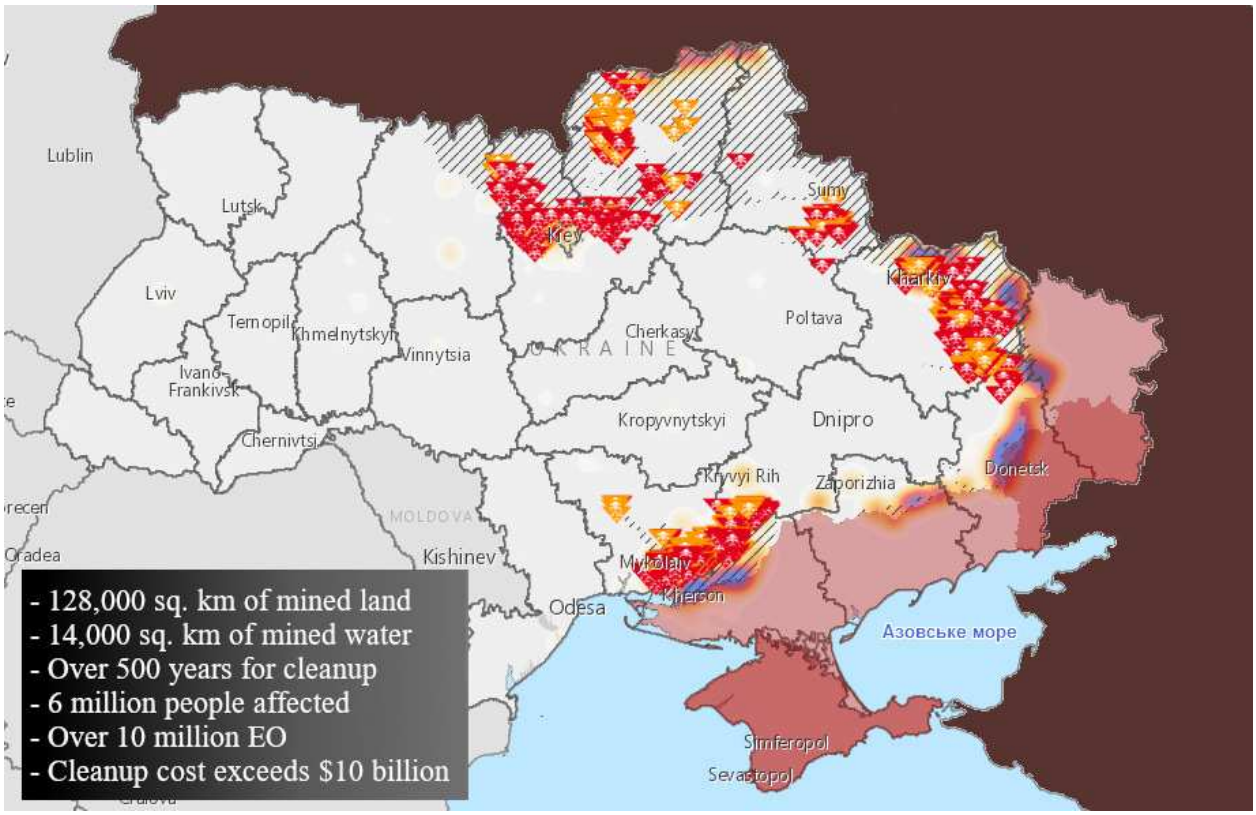


Figure 2.1: The territory of Ukraine contaminated by EO. Source https://ua.imsma.org [62]

Taking into account [58], it can be concluded that 128,000 square kilometers potentially contaminated by EO can be cleared within 545 years, which is also too long a period. The water contamination is not considered as it is difficult to predict how long it can take to demine 14,000 square kilometers of water. The next section contains an estimate of the number of explosive objects on the territory of Ukraine.

2.3. Estimation of the Number of Explosive Objects on the Territory of Ukraine

Russia has discharged approximately 10,000 shells per day, and when combined with the number discharged by Ukraine, the total daily tally reaches

12,000 [63]. Over the course of the 1,000[1] days of warfare that have elapsed thus far, the aggregate daily expenditure of munitions by both parties amounts to 12 million. According to experts [64], approximately 20% of these shells do not detonate, resulting in a substantial quantity of unexploded ordnance, estimated at 2.4 million. Furthermore, estimations indicate that the total number of shells discharged by both sides during the initial year of the war (2022) amounted to approximately 20 million [64], which suggests that the number of unexploded shells could potentially exceed 5.6 million.

The number of EO in the minefields that extend for hundreds of kilometers along the contact line is challenging to estimate. For instance, in the southern part of Ukraine alone (Kherson, Zaporizhzhia, and Donetsk regions), the length of mined areas is 420 km, and the density of EO in these areas reaches up to 5 mines per square meter on some sections, according to the Minister of Defense of Ukraine [65]. While the peak density is reported as 5 EO per square meter, a more reasonable average density across the entire mined area might be around 1 mine per square meter. Assuming a typical width of a minefield to be at least 20 meters, which is considered a minimum and can be significantly greater, for example, 100 or even 200 meters in some areas along the front line, the total area of mined land in these regions can be estimated. This area is 420,000 meters (length) multiplied by 20 meters (width), resulting in 8,400,000 square meters. Applying an average density of 1 EO per square meter to this area gives an estimated total of approximately 8.4 million EO. However, it is also important to consider that a minefield width of 20 meters might be a significant underestimation, especially given the presence of multiple Russian defense lines (at least three) in this area. A wider mined area would naturally lead to a significantly higher estimate of the total number of explosive ordnance.

[1]Note: This figure represents more than three years since the full-scale invasion began (as of the dissertation's writing). It's important to note that substantial explosive ordnance contamination has been present since Russia's occupation of Crimea and the invasion of Donbas started in 2014.

According to the Minister of the Armed Forces of Ukraine, as articulated during his address at the forum "Ukraine. The Year 2024," the active frontline extends approximately 1,200 kilometers, while the total defense line spans 3,200 kilometers. While the density of mines varies significantly along the frontline, reaching up to 5 per square meter in some areas of southern Ukraine, even a conservative lower bound estimate of 1,000 EO per kilometer of the frontline suggests at least 3.2 million EO along the total defense line. Estimations for the southern regions, heavily contaminated with minefields at densities reaching up to 5 mines per square meter, indicate a significant number, potentially around 8.4 million EO. When also considering the potential for over 5.6 million unexploded shells from artillery fire, the scale of explosive ordnance contamination in Ukraine is immense, suggesting a total of at least 17 million EO. It is important to note that the 420 km of mined areas in the south are likely part of the larger 3,200 km total defense line, meaning these southern areas might have been included in both estimations. However, even if the southern minefields were counted twice, this overlap does not significantly reduce the overall order of magnitude of our estimate due to the aforementioned reasons, such as the varying intensity of the conflict and the corresponding density of mines, which in many areas is clearly much higher than our conservative lower bound of 1 per meter of front. Consequently, considering the extensive minefields on both sides, including Ukrainian defensive positions, the total number of EO and unexploded ordnance could realistically reach at least 17 million, which is likely a lower bound estimation. It is important to note that in border areas and locations of high intensity conflict, the density of EO is likely to be considerably higher. This figure does not include mines dispersed by remote mining systems, which would further compound the challenges and costs associated with demining operations. It was estimated [66] that the cost to clear a single mine was approximately $1,000, suggesting that the financial burden of removing such a substantial quantity of explosive devices would be substantial.

2.4. Obstacles in the Detection of Explosive Objects

In light of the aforementioned challenges, the development of innovative strategies is imperative to enhance EO detection capabilities and address the limitations of existing systems. By addressing these challenges, it is possible to facilitate the development of more effective and efficient EO detection systems, with the ultimate goal of preserving lives and aiding in the reconstruction of war-impacted communities. A promising demining technique involves the integration of ML algorithms with UAV [36, 67, 68]. However, the complexity and variety of landmines and shells necessitates substantial datasets to develop effective machine learning algorithms. A significant challenge in the aforementioned surveys is the limited availability of comprehensive datasets suitable for training and evaluating detection algorithms. Additionally, the management of actual EO poses a risk to data acquisition and testing. Augmentation techniques [69–71] can significantly augment the volume of data available for training. Paper [8] examines the application of various augmentation methods for improving EO detection and intoduces two-stage augmentation for YOLO algorythm. This work will be discussed in greater detail in the next chapter.

The following challenges are identified when creating ML models for explosive objects recognition:

- **Insufficient data:** The development of accurate and dependable EO detection algorithms is impeded by a substantial paucity of diverse and comprehensive datasets. These datasets are imperative for the training of ML algorithms to accurately identify and categorize EO. However, acquiring such data presents a significant challenge due to various variables, including the scarcity of accessible imagery and the logistical obstacles associated with data collection in conflict zones. The restricted availability of data significantly hinders researchers' capacity to create algorithms that can generalize across various terrains and EO varieties.

The absence of diverse data can impede the efficacy of algorithms, potentially leading to heightened hazards to human safety.

- **Varied Explosive Hazards:** The topography affected by explosive objects exhibits remarkable diversity, as does the variety of EO types, including anti-personnel, anti-tank, and improvised explosive devices (IED). Each type of EO possesses distinct features, including differences in shape, size, and composition, which can complicate detection. A significant proportion of EO are composed of plastic materials, which often renders them undetectable by metal detectors. The variety of EO poses significant challenges for detection algorithms, which must reliably identify and classify a diverse array of explosive devices. Conventional detection techniques, such as metal detectors, frequently prove inadequate for identifying non-metallic EO, thereby exacerbating the detection challenge.
- **Safety Issues During Experiments:** Ethical and safety concerns pose significant challenges to conducting studies with actual EO for data collection. The management of actual EO poses a substantial threat to human life and can lead to catastrophic consequences if not handled properly. These safety concerns underscore the imperative for identifying alternative methods for data collection that prioritize human safety and environmental integrity.

2.5. A Methodology for Using Augmentation to Enrich Datasets

The efficacy of ML models in EO detection is contingent on the diversity, quantity, and reliability of the training data. The endeavor to gather a diverse and representative dataset is encumbered by hurdles, including constraints pertaining to accessibility, ethical considerations, and security concerns. The absence of comprehensive data presents substantial challenges to the advancement

and enhancement of machine learning algorithms, potentially constraining their efficacy in varied and unpredictable domains.

In addressing these challenges, data augmentation has emerged as a prominent technique. This method circumvents data restrictions by augmenting existing datasets with generated variations. Augmentation tactics encompass spatial alignment, pixel intensity alteration, geometric changes, and compositing, all aimed at imparting a sense of real-world variability to the information. This study explores the diverse applications of data augmentation in EO detection, emphasizing its role in overcoming data limitations.

The augmentation of artificial data enhances the magnitude and variety of existing datasets, thereby addressing issues of data scarcity. Employing various transformations, including spatial, pixel-based, and temporal alterations (covering day-night transitions), augments the quality and breadth of training data, mitigates the risk of model overfitting, and enables models to adjust to real-world variability, thus enhancing the accuracy of EO detection.

In the context of EO detection, augmentation is imperative for enhancing the efficacy of machine learning models, particularly deep learning methodologies. The utilization of diverse transformations, including rotation, scaling, cropping, flipping, and noise augmentation, can improve the diversity and quality of the training data. These modifications are crucial for enhancing the model's capacity to generalize and effectively identify EO across diverse contexts.

2.6. Methodology for Using 3D Printing to Create Datasets

Given the inherent risks associated with the management of actual EO and the paucity of data available for training machine learning models, it is imperative to explore alternative research methodologies. The utilization of 3D-printed EO models offers a potential solution to this challenge by facilitating

secure testing and comprehensive data acquisition. Furthermore, the use of 3D printing technology allows for the scaling or alteration of EO, thereby enabling the production of more resilient and versatile testing models. The integration of 3D printing technology in the fabrication of EO for UAV deployment holds significant potential in facilitating the training of models on printed replicas. This approach is particularly advantageous in addressing the challenges posed by detection and improvised explosive devices (Figure 2.2).

Research projects focused on the use of 3D-printed replicas for developing EO detection methods are highly relevant. The results of these research have the potential to significantly improve safety measures and speed up the demining process in Ukraine.

2.7. Methodology for Using Expert Data Contribution to Improve Models

Data scarcity is mitigated through augmentation techniques and the use of 3D printing. To create a multi-platform aggregation environment capable of identifying numerous EO, it is essential to continually augment datasets. The characteristics of numerous explosive devices are challenging to replicate via 3D printing or enhance through augmentation, rendering images of actual EO the sole dependable source of data for training deep learning models. This task can be executed by employing a platform for expert data contribution, disseminating the program as a mobile application or on a personal computer, and developing a bot through integration into messaging platforms. This approach optimizes engagement with skilled sappers, enabling continuous data collection for model training via a feedback mechanism that transmits data to the server. This methodology is elaborated upon in subsequent sections of the study.

Figure 2.2: Printed battle EO: The inner portion contains explosive material, while the interlayer between the outer shell is filled with bolts, serving as shrapnel

2.8. Selection of a Machine Learning Method

The selection of an appropriate machine learning method is crucial for attaining high accuracy in the identification of EO. This study employs the YOLO (You Only Look Once) family of algorithms [72], renowned for their real-time object detection capabilities. Specifically, the initial development focused on YOLOv8 [73] due to the following factors:

- **Exceptional Balance of Precision and Speed:** YOLOv8 exhibits a remarkable balance between accuracy and processing speed in object detection tasks, surpassing numerous other leading algorithms while maintaining the speed necessary for practical, near real-time applications.
- **Architectural Efficiency:** YOLOv8 features an enhanced architecture that facilitates the effective detection of objects of varying sizes and shapes, which is crucial for identifying diverse explosive devices under various conditions. Figure 2.3 illustrates the architecture of the YOLOv8 model.
- **Ease of Implementation and Customization:** YOLO is comparatively user-friendly and configurable, thus streamlining the development, training, and deployment of a tailored EO recognition system.

Furthermore, the research findings were validated by implementing YOLOv11 [75], the latest iteration of the algorithm at the time of this writing. This substantiates the autonomy of the proposed methodologies with respect to the specific configuration of the ML algorithm and demonstrates their adaptability to advancements within the YOLO family. The choice of a specific YOLO version can be tailored based on the desired level of accuracy and available computational resources. It is imperative to acknowledge that in the context of EO detection, the highest possible accuracy is paramount, given the potential life-threatening consequences of misclassification.

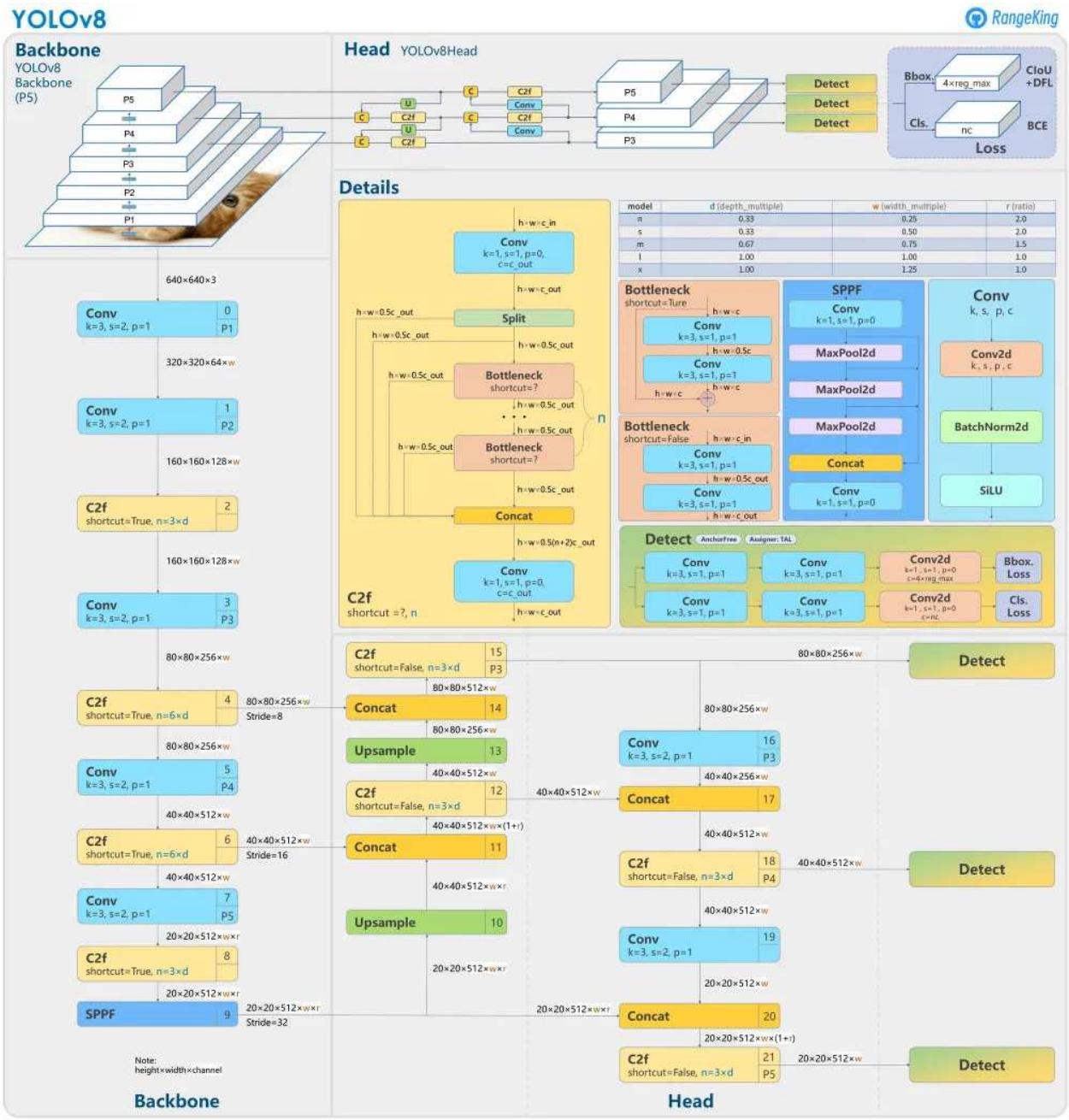


Figure 2.3: Architecture of the YOLOv8 model. Further insights into the specifics of this architecture can be found in the detailed guide on the Roboflow Blog [74]. Data sourced from [73], visualization by RangeKing on GitHub

YOLOv11 uses components from previous versions, such as SPPF (Spatial Pyramid Pooling – Fast), and introduces several novel components that enhance its performance: C2PSA (Cross Stage Partial with Spatial Attention) and C3K2 (Cross-Stage Partial with kernel size 2) [76–78]. These components are briefly

described below:

— **SPPF:** A module designed to efficiently extract multi-scale features using multiple pooling operations with different effective receptive fields. By reusing pooled results, SPPF reduces computational redundancy while preserving the ability to detect objects of varying sizes. It is an optimized version of the Spatial Pyramid Pooling (SPP) module [79], retained in recent YOLO versions for its speed and efficiency.

— **C2PSA:** Is a novel component in YOLOv11 that enhances the model's ability to focus on salient regions within an image. The article describes C2PSA as a notable addition, placed after the Spatial Pyramid Pooling - Fast (SPPF) block, which enhances spatial attention in the feature maps. This spatial attention mechanism allows the model to more effectively concentrate on key regions of the image, potentially increasing the accuracy of detecting objects of various sizes and positions. As described in [78], this attention mechanism contributes to the dynamic adjustment of the model's receptive field and the prioritization of features from areas most likely to contain objects of interest.

— **C3K2:** YOLOv11 replaces the C2f block found in earlier YOLO versions with the C3K2 block. According to [78], C3K2 is a more computationally efficient implementation of the Cross-Stage Partial bottleneck.

Figure 2.4 provides a simplified visual representation of the YOLOv11 model's architecture, highlighting the integration of the aforementioned components. The architecture of YOLO can be applied to a variety of tasks, including object detection, image classification, oriented object detection, object tracking, pose estimation, and instance segmentation.

Figure 2.5 illustrates the performance of various YOLO versions, including YOLOv5, YOLOv8, and YOLOv11, in comparison to EfficientDet and RTDETRv2, two other state-of-the-art object detection models.

The graph demonstrates the evolution of the YOLO family and its com-

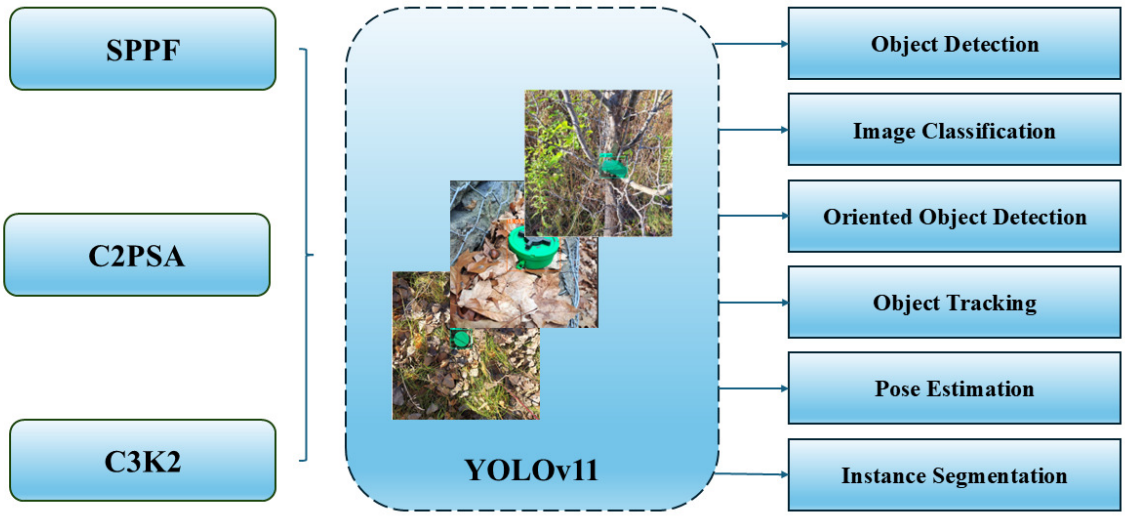


Figure 2.4: Simplified architecture of the YOLOv11 model, highlighting key components: SPPF, C2PSA, and C3K2. Data sourced from [78]

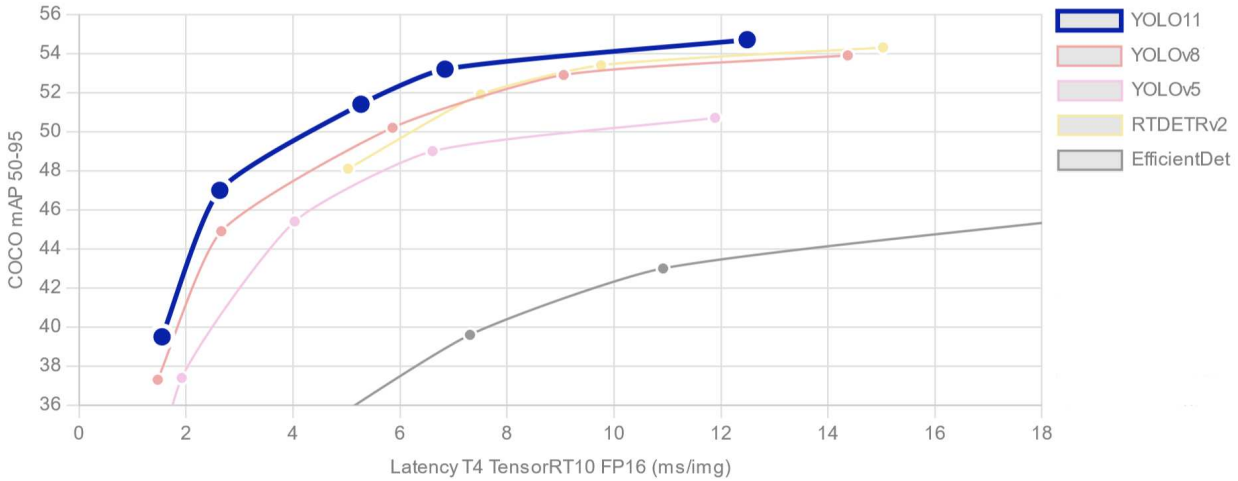


Figure 2.5: Performance comparison of different YOLO versions (v5, v8, v11), RTDETRv2, and EfficientDet on the COCO object detection benchmark dataset, measured using the mAP50-95 metric. Data sourced from [85]

petitive performance on the widely used COCO dataset [80], measured using the mAP50-95 metric. As depicted in Figure 2.5, the performance landscape of object detection models reveals a trade-off between speed and accuracy. While

an algorithm RTDETRv2-x [81] presents a comparable mAP50-95 of 54.3% to YOLOv8x's 53.9%, and a slightly faster inference speed (15.03 ms/img vs. 14.37 ms/img), YOLOv11x demonstrates better performance, achieving the highest mAP50-95 of 54.7% at a speed of 12.49 ms/img. EfficientDet [82] exhibits significantly lower computational efficiency in this comparison. This comparison, along with in-depth analyses in [83, 84], further justifies the selection of the YOLO architecture for this research.

While the initial development and training utilized YOLOv8, the adaptability of the proposed methodologies to newer YOLO versions, specifically YOLOv11, demonstrates the forward-looking nature of this research.

2.9. Formulas and Definitions

To accurately assess the model's performance during training and evaluate the impact of data augmentation, it is essential to define the key definitions used in ML. A crucial element in model training is the **loss function**, which quantifies the difference between the model's predictions and the ground truth values. While metrics like precision, recall, F1-score, mAP, and IoU (defined in Section 2.9.1.1) are used for evaluating overall model performance on a held-out dataset, loss functions guide the optimization process during training.

2.9.1. The Loss Functions. Several loss functions are commonly employed, depending on the task. For instance, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Sum of Squared Errors (SSE) are frequently used in regression tasks. These are formally defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{2.1}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.2}$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{2.3}$$

where:

- n is the number of data points.
- y_i is the actual value for the i-th data point.
- $\hat{y}_i$ is the predicted value for the i-th data point.

In a nutshell:

— **The Mean Absolute Error (MAE)** (2.1) is a measure of the average magnitude of the errors in a set of predictions, without considering their direction (i.e., whether the prediction is above or below the true value). It is calculated as the average of the absolute differences between the predicted values and the actual values. A lower MAE indicates better model accuracy.

— **The Mean Squared Error (MSE)** (2.2) is another crucial metric in ML. The MSE quantifies the average of the squared differences between the predicted values and the actual values. Squaring the errors gives more weight to larger errors. MSE is a commonly used loss function, and its value is always non-negative. A lower MSE indicates that the model's predictions are closer to the actual values, suggesting better model performance.

— **The Sum of Squared Errors (SSE)** (2.3) is defined as the total sum of the squared differences between the predicted values and the actual values, thereby serving as a measure of the overall discrepancy between the data and the model's predictions. A smaller SSE indicates a tighter fit of the model to the data, suggesting that the model has effectively captured the underlying patterns in the data. SSE is frequently employed as an optimization criterion during model training, where the objective is to minimize SSE by adjusting the model's parameters.

For object detection tasks, as in this research, models like YOLO employ specialized loss functions. Specifically, YOLOv8 utilizes a combination of losses to address different aspects of the detection task: bounding box regression and classification. These losses have demonstrated improved performance in object detection, particularly for smaller objects, which is highly relevant in the context of EO detection.

Bounding Box Regression Loss quantifies the discrepancy between the predicted bounding box coordinates and the ground truth bounding box coordinates. Accurate localization of EO is crucial for their safe removal, making this loss component a critical aspect of the training process. YOLOv8 uses a combination of Complete Intersection over Union (CIoU) [86] and Distribution Focal Loss (DFL) [87] for bounding box regression.

CIoU extends the traditional Intersection over Union (IoU) metric 2.7 by not only considering the overlap area but also incorporating the central point distance and aspect ratio difference between the predicted and ground truth boxes.

DFL addresses the issue of continuous label representation in bounding box regression. Instead of treating the target location as a single value, DFL models it as a distribution over possible locations. This allows the model to learn a more nuanced and accurate representation of the bounding box coordinates.

Classification Loss quantifies the error between the predicted class probabilities and the true class labels for each bounding box. In the context of EO detection, this loss ensures that the model accurately distinguishes between EO and other objects or background.

YOLOv8 employs **Binary Cross-Entropy (BCE) Loss** for classification. BCE Loss, also known as Log Loss, is suitable for binary classification problems, where the outcome is either 0 or 1 (in this case, EO or not EO). The BCE loss function is defined as follows:

$$BCE(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{2.4}$$

where:

- n is the number of data points (in this case, bounding boxes).
- y_i is the true class label for the i-th bounding box (1 if it contains an EO, 0 otherwise).
- $\hat{y}_i$ is the predicted probability for the i-th bounding box, indicating the model's confidence the box contains an EO (a value between 0 and 1).

BCE Loss penalizes the model more heavily when it confidently predicts the wrong class. Let's break it down:

— **When the true label (y_i) is 1 (EO present):** The first term inside the summation, $y_i \log(\hat{y}_i)$, becomes active. If the predicted probability ($\hat{y}_i$) is close to 1, the $\log(\hat{y}_i)$ is close to 0, resulting in a small loss. However, if the model incorrectly predicts a low probability (close to 0), $\log(\hat{y}_i)$ becomes a large negative number, leading to a large loss.

— **When the true label (y_i) is 0 (no EO):** The second term inside the summation, (1 - y_i) log(1 - $\hat{y}_i$), becomes active. If the predicted probability ($\hat{y}_i$) is close to 0, log(1 - $\hat{y}_i$) is close to 0, resulting in a small loss. But if the model incorrectly predicts a high probability (close to 1), log(1 - $\hat{y}_i$) becomes a large negative number, resulting in a large loss.

In essence, BCE Loss encourages the model to output high probabilities for bounding boxes that contain EO and low probabilities for those that do not. A lower Loss signifies that the model's predicted probabilities are well-aligned with the true class labels, thereby indicating enhanced classification performance.

2.9.1.1. Evaluation Metrics. However, while loss functions provide valuable insights into the model's performance during training, they do not always provide a comprehensive evaluation of the model's performance. It is essential

to evaluate the model's performance on a held-out test set using metrics that reflect the real-world impact of correct and incorrect predictions. A fundamental tool for understanding these metrics is the **Confusion Matrix**.

A confusion matrix provides a visual representation of a classification model's performance by displaying the counts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). For binary classification problems, such as EO detection, the confusion matrix takes the form of a 2x2 table, as illustrated in Figure 2.6:

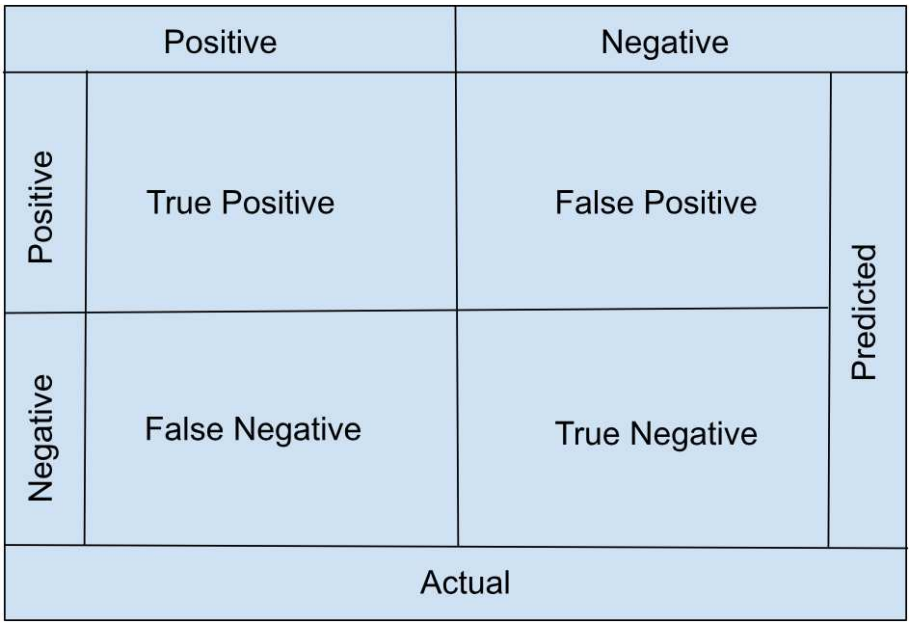


Figure 2.6: Confusion matrix

where:

- **True Positives (TP)** are the cases where the model correctly predicted the presence of an EO.
- **False Positives (FP)** are the cases where the model incorrectly predicted the presence of an EO when there was none (a "false alarm").
- **False Negatives (FN)** are the cases where the model failed to predict the presence of an EO when one was actually present (a "miss").
- **True Negatives (TN)** are the cases where the model correctly predicted the absence of an EO.

The confusion matrix is a fundamental data structure that facilitates the calculation of other critical evaluation metrics, such as precision and recall. These metrics provide a more nuanced understanding of the model's performance in the context of EO detection.

Precision measures the proportion of correctly identified positive cases (True Positives) out of all the cases that the model predicted as positive (True Positives + False Positives). It is calculated by formula 2.5. In the context of EO detection, it answers the question: Out of all the bounding boxes that the model predicted to contain EO, how many actually contained EO?

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{2.5}$$

Recall (calculated by formula 2.6) measures the proportion of correctly identified positive cases (True Positives) out of all actual positive cases (True Positives + False Negatives). In the context of EO detection, it answers the question: Out of all the actual EO present in the images, how many did the model correctly identify?

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{2.6}$$

Importance in EO Detection:

- **High Precision** is crucial in EO detection because false alarms (predicting an EO where there is none) can lead to wasted resources and potentially dangerous situations for demining teams.
- **High Recall** is equally important because failing to detect actual EO (false negatives) can have severe consequences, leaving these dangerous objects in the ground.

Therefore, achieving a balance between high precision and high recall is

paramount in EO detection.

Figure 2.7: Ground truth (blue) and predicted (green) bounding boxes. IoU is calculated from their intersection and union areas

Precision and recall are primarily concerned with the classification aspect of EO detection (i.e., determining whether a bounding box contains an EO or not). However, another crucial metric is the **intersection over union (IoU)** metric, which is defined as follows:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}. \tag{2.7}$$

Also known as the Jaccard Index, this metric assesses the accuracy of the predicted bounding box's location and size. The IoU is calculated as the area of intersection between the predicted and ground truth bounding boxes divided by the area of their union.

Formula using Confusing matrix:

Importantly, IoU can also be expressed in terms of True Positives (TP), False Positives (FP), and False Negatives (FN):

$$IoU = \frac{True\ Positives}{True\ Positives + False\ Positives + False\ Negatives}. \quad (2.8)$$

In simpler terms:

It is imperative to conceptualize two overlapping rectangles: one delineating the predicted bounding box and the other delineating the ground truth. The IoU is the ratio of the area where these rectangles overlap to the total area covered by both rectangles.

Note: The formula above, expressed in terms of TP, FP, and FN, represents an overall metric of how well the model's predictions align with the ground truth objects at the instance level (based on a chosen IoU threshold). It is not the calculation of IoU for a single bounding box overlap (see formula 2.7).

The following list provides a detailed explanation of the terms and conditions of the aforementioned formula:

— An IoU of 1 indicates a perfect overlap, i.e., the predicted box exactly matches the ground truth box.

— An IoU of 0 signifies a complete absence of overlap.

— In practice, an IoU above 0.5 is often considered a good prediction.

Relevance to EO Detection:

In the domain of EO detection, the accuracy of localization is paramount. A high IoU value indicates that the predicted bounding box closely corresponds to the real-world location and dimensions of the EO, providing crucial insights for demining operations.

Relationship to Box Loss:

The term "IoU" is directly related to the concept of "Box Loss," which was previously mentioned. While "Box Loss," calculated using the mean squared error (MSE) metric, quantifies the difference in coordinates, "IoU" provides a more intuitive measure of overlap. In many object detection models, including

YOLO, the "Box Loss" is often derived from or incorporates "IoU" calculations to optimize the bounding box predictions during the training process. Some common variants include:

— Generalized IoU (GIoU)
— Distance-IoU (DIoU)
— Complete-IoU (CIoU)
— Efficient-IoU (EIoU)
— Focal-EIoU (FEIoU)
— Spatial-Exclusion (SE)-IoU

These variants have been proposed to overcome the limitations of traditional IoU and enhance bounding box regression accuracy.

Box loss, therefore, measures the accuracy of the predicted bounding box location using MSE, while class loss (BCE Loss) measures the accuracy of the predicted class label. In addition to these loss functions, metrics like precision, recall, and IoU provide a comprehensive evaluation of the model's performance in EO detection.

2.10. Conclusions to Chapter 2

This chapter has established the critical need for advanced EO detection and management, particularly in conflict-affected regions like Ukraine. The chapter has defined various categories of explosive threats, including anti-personnel and anti-vehicle landmines, improvised explosive devices (IED), and unexploded ordnance (UXO), emphasizing their devastating impact on civilian populations, especially children. The scale of the problem in Ukraine has been quantified, highlighting the millions of explosive remnants that contaminate the land, hindering agriculture, infrastructure development, and overall safety.

To address the challenges of EO detection, this chapter has explored several key methodologies and technologies. Data augmentation techniques, detailed

in Section 2.5, are presented as a crucial tool for overcoming data scarcity and improving the robustness of deep learning models. The use of 3D printing for creating safe and diverse training datasets, as outlined in Section 2.6, has been proposed as an innovative solution for model development. Furthermore, the potential of expert data contribution via the application for data collection, discussed in Section 2.7, has been highlighted.

The chapter has also justified the selection of the YOLO algorithm, described in Section 2.8, for EO detection, emphasizing its balance of precision, speed, and architectural efficiency. The definition of key performance metrics, including loss functions and evaluation metrics (Section 2.9), has provided a framework for assessing the effectiveness of the proposed methods.

In summary, this chapter has laid the groundwork for developing a comprehensive EO detection system by establishing the relevance of the problem, outlining key objectives, and introducing the core methodologies that will be employed in the subsequent chapters. The integration of data augmentation, 3D printing, an expert data contribution platform, and the YOLO algorithm represents a promising approach to enhancing the safety and efficiency of EO detection and contributing to the reconstruction and rehabilitation of conflict-affected areas.

CHAPTER 3
THE APPLICATION OF AUGMENTATION TO ENHANCE THE QUALITY OF DATASETS

3.1. The Necessity to Enhance Explosive Objects Detection

Data augmentation is a technique used to deliberately augment the quantity and diversity of a training dataset for ML models. This process involves the production of new, slightly modified versions of existing data points or the generation of synthetic data points derived from the existing ones. This augmentation enhances the model's ability to generalize and perform well with unfamiliar input, particularly when the initial dataset is limited.

In the intricate domain of EO detection, the acquisition of extensive datasets is a significant difficulty, underscoring the essential function of data augmentation. The cornerstone of proficient EO detection models is a dataset that accurately represents the various types of EO distributed across different terrains, atmospheric conditions, and emplacement types. Nonetheless, the endeavor to compile such an extensive collection has practical challenges, including logistical, ethical, and security obstacles. The scarcity of varied photographs of EO presents a significant impediment, hindering the creation of models that are universally applicable. In this context, augmentation is a viable solution. Augmentation artificially enhances dataset diversity by adeptly applying various alterations to existing images. This procedure generates a dataset that, although derived from a limited collection of real samples, reflects the unpredictability and complexity of real-world situations.

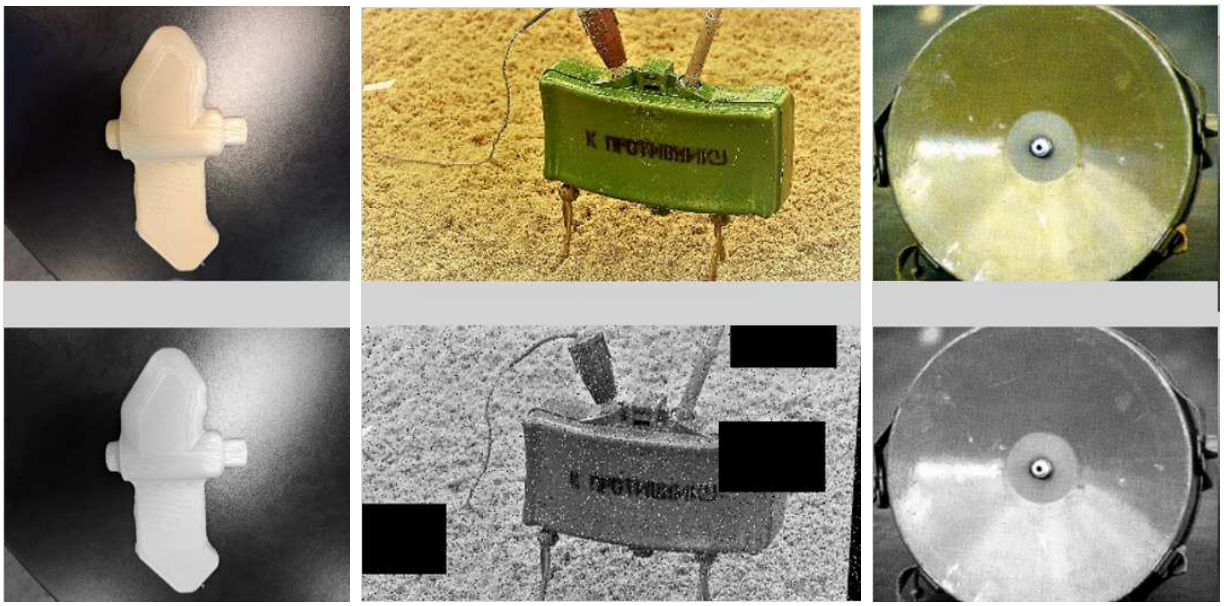
The utilization of restricted datasets inevitably give rise to the issue of overfitting, a phenomenon in which models, in their pursuit of accuracy, become

constrained by the particulars of the training data, thereby reducing their effectiveness in novel scenarios. The dearth of authentic EO imagery further exacerbates this issue, as models, devoid of adequate diversity, tend to memorize dataset features, culminating in subpar performance in real-world settings.

Augmentation, therefore, emerges as a crucial technique, producing numerous synthetic variations from a foundational set, thereby enhancing the model's variability. This enhancement mitigates the hazards of model overfitting, facilitating the development of models that, despite being grounded on insufficient empirical data, can identify EO across the various environmental conditions.

3.2. Common Augmentation Techniques

3.2.1. The Basic Augmentation Methodologies. EO detection is significantly enhanced by data augmentation, which employs various methods to enrich and diversify the dataset. This section focuses on the primary categories of augmentation techniques pertinent to this domain, including spatial transformations, pixel-level variations, and geometric alterations. The impact of different approaches on various objects can vary significantly. The selection of an appropriate algorithm for a particular object is typically achieved through experience and experimentation. For certain EO types, such as the round MON-100 and MON-200 (Figure 3.1c), grayscale considerably diminishes model accuracy, yet for others, like the PFM-1 (petal) (Figure 3.1a), it enhances accuracy. The PFM-1 often exhibits a distinctive shape that is more readily identified in grayscale, whereas the MON-100 and MON-200 may rely more on color variations for accurate detection. The MON-100/200, when converted to grayscale, transforms into basic round shapes, whereas the petal, exhibiting a broad spectrum of hues, enhances accuracy instead. In the case of MON-50, grayscale is a viable option due to its capacity to exhibit variations in color, as illustrated in Figure 3.1b.



(a) PFM-1 (b) MON-50 (c) MON-100

Figure 3.1: Grayscale Augmentation

3.2.2. Spatial Transformations. Spatial transformations are of critical importance in the augmentation of data by virtue of their capacity to manipulate the spatial configuration of pixels within a given image (Figure 3.2). These transformations effect alterations in the geometric structure of the image while preserving the essential meaning of the image, such as the object's identity and its general context, thereby giving rise to a multitude of variations of the original image. This diversity is indispensable for the training of robust ML models that are capable of effectively generalizing to unseen data and accommodating variations in object orientation, size, and position.

These techniques include:

- **Rotation:** This involves rotating the image by a certain angle, providing different perspectives and orientations of the object. Rotation can help the model learn to recognize the object regardless of its orientation.
- **Scaling:** This involves resizing the image, either by zooming in to magnify the object or zooming out to show a wider context. Scaling can help the model recognize the object at different sizes and scales.

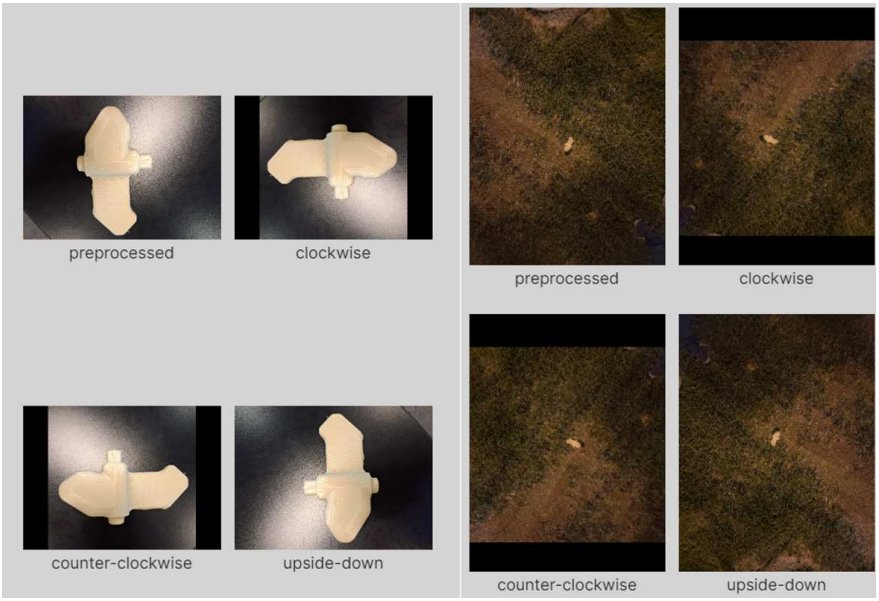


Figure 3.2: Example of Rotation

- **Cropping**: This involves selecting a specific region of the image and discarding the rest. Cropping can help the model focus on the important features of the object and reduce the influence of background noise.
- **Flipping:** This involves creating a mirror image of the original image. Flipping can help the model learn to recognize the object regardless of its horizontal orientation.

By strategically utilizing these spatial adjustments, either separately or collectively, the size and variety of the dataset can be substantially enhanced. This consequently results in improved model performance, especially regarding accuracy and robustness when faced with novel, unexplored data. The utilization of these techniques is particularly significant in areas like EO detection, where the capacity to generalize across differences in object appearance and ambient conditions is crucial.

3.2.3. Pixel-Level Variations. Pixel-level modifications are a powerful set of techniques that alter the visual characteristics of images at the fundamental level of individual pixels (Figure 3.3). By manipulating these low-level attributes, a wide range of real-world conditions and variations can be sim-

ulated, significantly enhancing the robustness and generalizability of machine learning models. These techniques include:

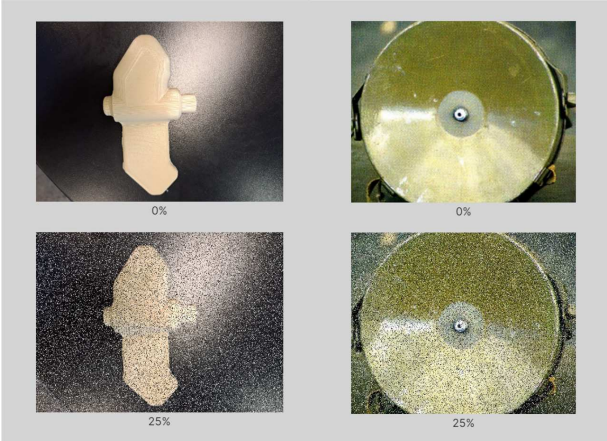


Figure 3.3: Example of Noise

- **Brightness Adjustments:** Modifying the overall brightness of an image can simulate different lighting conditions. For instance, increasing brightness can mimic well-lit environments, while decreasing it can simulate low-light, nighttime, or overcast scenarios. This is particularly relevant in EO detection, where visibility can vary drastically depending on the time of day, weather, and surrounding environment.
- **Contrast Adjustments:** Altering the contrast changes the difference between the darkest and lightest areas of an image. Increasing contrast can make features more pronounced, while decreasing it can create a more muted or washed-out appearance. This can help the model learn to identify EO under varying levels of visual clarity, such as when they are partially obscured by dust, shadows, or vegetation.
- **Saturation Adjustments:** Saturation refers to the intensity of colors in an image. Increasing saturation makes colors more vibrant, while decreasing it moves the image closer to grayscale. Adjusting saturation can help the model become invariant to color variations caused by different lighting, weather, or camera sensors.

- **Hue Shifts:** Modifying the hue rotates the colors in the image along the color wheel. This can simulate variations in the color of the environment or the EO itself, which can occur due to factors like rust, vegetation growth, or different manufacturing processes.
- **Color Temperature Adjustments:** Changing the color temperature alters the overall color balance of the image, making it appear warmer (more yellow/red) or cooler (more blue). This simulates the effect of different light sources, such as sunlight at different times of day or artificial light.
- **Noise Incorporation:** Adding noise to an image introduces random variations in pixel values, mimicking imperfections found in real-world data. Different types of noise can be used:
  - **Gaussian Noise:** Adds random values from a Gaussian distribution to each pixel, simulating sensor noise or general image degradation.
  - **Salt-and-Pepper Noise:** Randomly replaces some pixels with very bright (salt) or very dark (pepper) values, simulating faulty pixels or transmission errors.
  - **Speckle Noise:** Multiplicative noise that can be used to simulate noise in images acquired using technologies such as SAR (Synthetic Aperture Radar) or ultrasound, which can be relevant if such sensors are used for EO detection.
- **Channel Shuffling/Dropping:** In color images, the order of the color channels can be shuffled (e.g., RGB to BGR) or one or two channels can be randomly dropped, forcing the model to rely on less information and potentially learn more robust features.

By modifying these pixel-level attributes, data augmentation can create a diverse set of image variations that are more representative of the complexities of real-world scenarios. These modifications help machine learning models generalize better to unseen data, improving their ability to perform accurately and

reliably in various situations. In the context of EO detection, this translates to models that are more robust to changes in lighting, weather, image quality, and sensor variations, ultimately leading to more effective and reliable detection systems.

3.2.4. Geometric and Morphological Transformations. Geometric modifications are alterations to an image that involve manipulation of its structure, such as stretching or curving. These modifications can facilitate models' learning to identify objects that are misaligned or deformed.

Morphological methods, including dilation, erosion, and shear (Figure 3.4), alter the shape and characteristics of an image. Morphological operations typ-

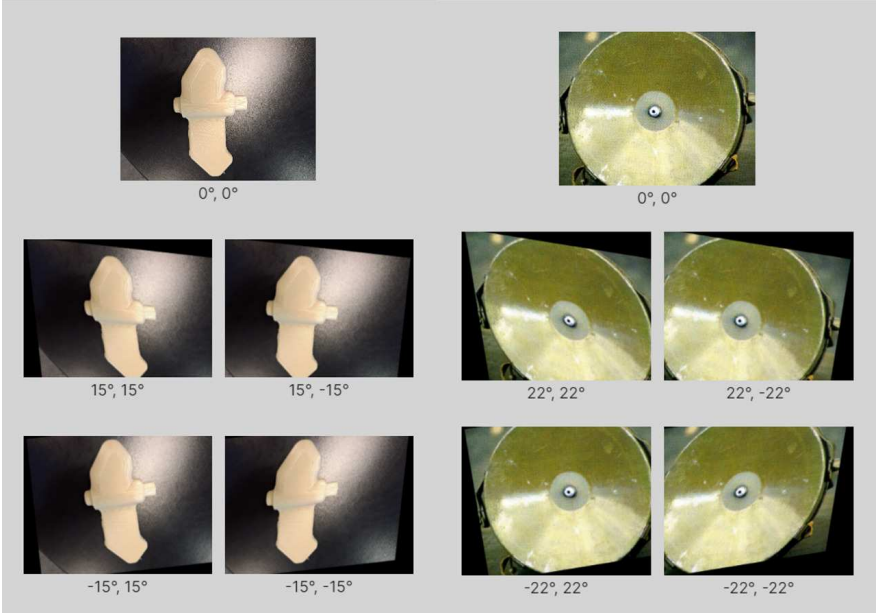


Figure 3.4: Shear augmentation

ically work on the shape and structure of objects in binary (black and white) or grayscale images. Dilation enlarges the dimensions of objects, whereas erosion diminishes them. Shear alters the image by displacing pixels in a specific direction, resulting in a slanted or skewed effect. These strategies can enhance models' resilience to subtle alterations in shape or texture, thus enhancing their ability to detect EO under diverse settings. The augmentation of data in EO

detection is significantly enhanced by geometric and morphological modifications, which are essential for this purpose. Through the simulation of various settings and scenarios, models are prepared for real-world challenges, thereby enhancing accuracy and reliability.

3.3. Advanced Augmentation Techniques

While basic augmentation methods, such as spatial transformations and pixel-level variations, provide a valuable foundation for improving model robustness, advanced augmentation techniques offer the potential to further refine the model's performance and address specific challenges inherent in EO detection. In this research, enhanced data augmentation occupies a crucial role within the YOLO training process, augmenting the diversity of the data and, consequently, the model's performance.

3.3.1. Mosaic, MixUp, and CutMix. Mosaic, MixUp and CutMix [88, 89] are advanced augmentation techniques that go beyond simple image manipulation. CutMix pastes a patch from one image onto another, while MixUp blends the images. Mosaic combines several images into one. They involve blending elements from multiple images along with their corresponding labels. This not only increases label diversity but also provides the model with a broader spectrum of visual features to learn from. As a result, the model's understanding of various EO types is enhanced, reducing the likelihood of false positives. In this study, MixUp is used in conjunction with mosaic augmentation (Figure 3.5)).

3.3.2. Augmentation Utilizing GANs. Generative Adversarial Networks (GANs) have emerged as a powerful tool in data augmentation [90], particularly for generating synthetic images that closely resemble real-world examples. In the context of EO detection, GANs offer the potential to create

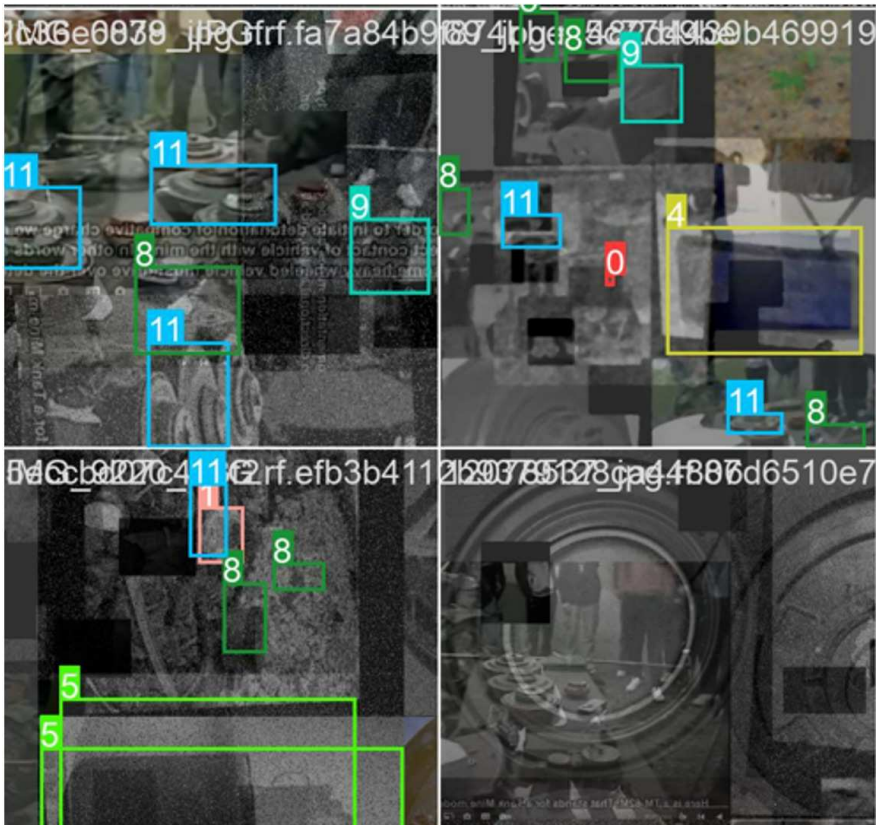


Figure 3.5: Mosaic of mix-ups

realistic images of EO in diverse environments and varying conditions, effectively expanding the training dataset. A GAN comprises two neural networks: a generator, tasked with creating synthetic images, and a discriminator, which evaluates the authenticity of these images, distinguishing them from real examples. This adversarial process, where the generator strives to fool the discriminator, refines the generator's ability to produce increasingly realistic images.

Potential Benefits:

— **Generating Diverse Scenarios:** GANs can generate images of EO under various conditions (e.g., different soil types, vegetation, lighting, and levels of obscuration) that might be underrepresented in the original dataset.

— **Expanding Limited Datasets:** They can significantly expand limited datasets, which is a common challenge in EO detection.

Challenges:

— **Training Instability:** GANs are notoriously difficult to train, often suffering from instability issues like mode collapse (where the generator produces limited varieties of samples).
— **Computational Cost:** Training GANs can be computationally expensive, requiring significant resources.

Despite these challenges, the potential of GANs to enhance EO detection datasets is significant. The integration of synthetic images generated by GANs into the training set can improve the model's ability to generalize and accurately identify EO in diverse real-world scenarios. This method is planned for implementation and evaluation in subsequent phases of this research.

3.3.3. Sim2Real Augmentation. Sim2Real [91] is a powerful augmentation technique that aims to bridge the gap between virtual simulations and real-world data. This approach leverages the ability to create highly realistic simulations of environments and objects, in this case EO, and use these simulations to generate synthetic training data. These simulations can encompass a wide range of scenarios and challenges that might be difficult or impossible to capture with real-world data collection. For instance, one can simulate various soil types, vegetation densities, weather conditions, times of day, and even different types of EO burial or obscuration.

By training models on a combination of simulated and real-world data, Sim2Real enables them to learn robust features that are invariant to the specific characteristics of each domain. The primary advantage of incorporating Sim2Real into EO detection is the potential to significantly enhance the model's ability to generalize across diverse and unseen conditions. This surpasses the limitations of relying solely on basic camera images, which may not capture the full spectrum of real-world variability. The model becomes more adept at detecting EO, even when faced with variations in terrain, lighting, or occlusion that were not present in the real-world training data.

3.3.4. AutoAugment and Learned Augmentation Policies. AutoAugment represents a paradigm shift from manually designed augmentation strategies to automatically learned policies. This technique, along with other learned augmentation methods, employs search algorithms, often based on reinforcement learning, to discover the optimal augmentation policies for a given dataset and model. Instead of relying on fixed transformations, AutoAugment explores a vast search space of augmentation operations (e.g., rotation, shear, color adjustments) and their associated probabilities and magnitudes.

Potential Benefits:

— **Data-Specific Optimization:** AutoAugment can identify augmentation strategies that are specifically tailored to the characteristics of the EO dataset, potentially leading to greater performance improvements than generic augmentations.

— **Automated Process:** It automates the process of finding effective augmentations, saving time and effort compared to manual design.

Challenges:

— **Computational Cost:** The search process can be computationally expensive, requiring significant resources and time.

— **Search Space Design:** The effectiveness of AutoAugment depends on the design of the search space, which can be complex and require domain expertise.

While AutoAugment and other learned augmentation policies hold great promise, their computational demands and complexity have limited their application in the current stage of this research. However, future work will investigate the potential of these techniques to further optimize the augmentation pipeline for EO detection.

3.3.5. Adversarial Examples. Adversarial examples are specifically crafted inputs designed to intentionally mislead ML models. In the context of image

classification, these are images that have been subtly perturbed, often in a way that is imperceptible to the human eye, but that cause the model to make an incorrect prediction with high confidence. While seemingly counterintuitive, incorporating adversarial examples during training can significantly enhance model robustness.

Potential Benefits:

— **Improved Robustness:** By exposing the model to adversarial examples during training, it learns to be less sensitive to small perturbations and more robust to potential attacks or noisy inputs.

— **Identification of Weaknesses:** Adversarial examples can help identify vulnerabilities and blind spots in the model, providing insights into areas where the model's decision-making process can be improved.

Application to EO Detection:

In EO detection, adversarial examples could simulate scenarios where EO are intentionally camouflaged or where sensor data is corrupted. Training on such examples can enhance the model's ability to correctly identify EO even under challenging conditions.

Challenges:

— **Computational Cost:** Generating adversarial examples can be computationally expensive.

— **Overfitting to Adversarial Examples:** There's a risk that the model might overfit to specific types of adversarial examples, reducing its performance on regular data.

Despite these challenges, exploring the use of adversarial examples in the context of EO detection is a promising avenue for future research, particularly for improving the security and reliability of detection systems.

3.3.6. Neural Style Transfer. Neural Style Transfer is a technique that blends the content of one image with the artistic style of another. This is

achieved by leveraging the feature representations learned by CNN. While not a traditional augmentation technique, Neural Style Transfer offers a unique way to generate variations in the training data, potentially improving the model's ability to generalize across different visual styles.

Potential Application to EO Detection:

In the context of EO detection, Neural Style Transfer could be used to:

— **Vary Environmental Conditions:** Apply the style of images taken under different lighting conditions, weather, or times of day to existing EO images, creating synthetic data that reflects a wider range of environments.

— **Simulate Different Sensor Types:** Transfer the style of images captured by different sensor modalities (e.g., thermal, infrared) to optical images, potentially enhancing the model's ability to generalize across sensor types.

Challenges:

— **Relevance of Style:** The choice of style images needs to be carefully considered to ensure that the generated images are relevant to the task of EO detection.

— **Computational Cost:** Style transfer can be computationally demanding, especially for high-resolution images.

While Neural Style Transfer hasn't been incorporated into the current stage of this research, it presents an interesting avenue for future exploration, particularly for generating more diverse and representative training data for EO detection models.

Employing these sophisticated augmentation methodologies enables the model effectively handle diverse and complex datasets. This enhanced data improves the accuracy and adaptability of the models. These techniques are reshaping the capabilities of EO detection, leading to more effective and safer solutions.

3.4. Preprocessing Techniques

The present study's primary focus is on data augmentation, although it is essential to address the preliminary processes of preprocessing. While pretreatment does not expand the dataset size like augmentation does, it remains a fundamental element in most ML workflows. A crucial procedure involves resizing all images to ensure uniformity throughout the dataset. The study identified and employed various preprocessing technologies to enhance data quality. Specifically:

— Auto-Orient was employed to standardize image orientation, guaranteeing homogeneity in model input.

— The resizing of all images ensured uniform dimensions, thereby maintaining consistency within the dataset.

— The employment of auto-adjust contrast enhancement led to enhanced image clarity and discernibility, thereby facilitating superior pattern detection by the models.

— Initially, the conversion of all photographs to grayscale was contemplated; however, it was ultimately decided to supplement only 30% of the dataset in this fashion, as it produced more favorable results.

3.5. Dataset Description and Analysis

The initial phase of this research involved a comprehensive search for publicly available images of landmines. This process yielded 2,128 images covering 23 distinct landmine types, as summarized in Table 3.1 and visualized in Figure 7. These images were sourced from various online repositories and databases. In addition to landmine images, the dataset was supplemented with images of visually similar objects, categorized as "not an EO," to enhance the model's ability to discriminate between landmines and benign objects. The dataset

was annotated using the Roboflow platform [93], and a summary of the initial dataset is provided below:

Dataset Statistics:

— **Total Images**: 2,128

— **Total Annotations**: 3,334

— **Average Annotations per Image**: 1.6

— **Average Image Resolution**: 0.60 MP (Median: 833x671)

— **Image Resolution Range**: 0.01 MP to 16.04 MP

— **Image Size Distribution**:

- Tiny (under 32x32): 4 images (0.2%)
- Small (32x32 – 320x320): 60 images (2.9%)
- Medium (320x320 – 640x640): 491 images (23.1%)
- Large (640x640 – 1280x1280): 689 images (32.4%)
- Jumbo (over 1280x1280): 884 images (41.5%)

Annotation Distribution:

— **Images with Single Annotation:** 1,683 images (see figure 3.6a)

— **Images with Multiple Annotations**: 445 images (see figure 3.6b)

Annotation Heat Map:

Figure 3.7 presents an annotation heat map, visualizing the spatial distribution of annotations within the dataset. The color gradient indicates the density of annotations per grid cell, with warmer colors representing higher concentrations.

The annotation heat map (Figure 3.7) reveals a non-uniform distribution of bounding box placements within the images. Notably, the lower-right quadrant exhibits the highest concentration of annotations, indicated by the red hue, while the density progressively decreases towards the upper-left quadrant. Furthermore, the region of highest annotation density occupies approximately one-quarter of the total heat map area. This concentration suggests that the annotated bounding boxes, on average, cover a significant portion of the image

Table 3.1

Initial Dataset Composition: Landmine Types and Image Counts

ID	Class Name	Total	Train	Valid	Test
0	PFM-1	708	518	106	84
1	VS-50	178	106	40	32
2	PMN	171	112	39	20
3	not an EO	76	58	12	6
4	PMN-4	231	182	36	13
5	TM-62-M	219	157	37	25
7	TM-62-P	84	61	12	11
8	TM-57	74	53	13	8
9	TM-72	79	56	13	10
10	TM-62-Д	34	22	5	7
11	TM-89	23	16	7	0
12	TM-46	144	110	22	12
13	M14	174	118	46	10
14	MON-50	200	141	36	23
15	M-21	74	50	14	10
16	PMN2	168	120	19	29
17	POMZ-2	95	66	19	10
18	M2A1	74	51	14	9
19	OZM-72	170	115	32	23
20	POM-2	168	116	33	19
21	MON-90	30	17	8	5
22	MON-100	83	55	17	11
23	TM-83	34	21	8	5
24	PMN-4	43	26	13	4

(a) Single annotation. (b) Multiple annotations.

Figure 3.6: Examples of images with different numbers of annotations

area, indicating that the EO in the dataset are typically captured at close range or with a relatively large zoom factor. This observation may be attributed to the tendency for online images to depict EO at a larger scale to enhance visibility and clarity. However, it also implies that the dataset may under-represent EO as they would appear when viewed from further distances or with a wider field of view. Furthermore, the concentration of annotations in the lower-right quadrant raises concerns about potential positional bias in the trained model. The model might inadvertently learn to associate that specific region of the image with the presence of EO, potentially reducing its ability to generalize to different object placements. Consequently, a model trained primarily on this dataset might exhibit reduced sensitivity to smaller, more distant, or differently positioned EO in real-world scenarios. Therefore, the model's ability to detect such objects could be negatively impacted. This limitation highlights the importance of the data augmentation techniques discussed in section 3.2.2,

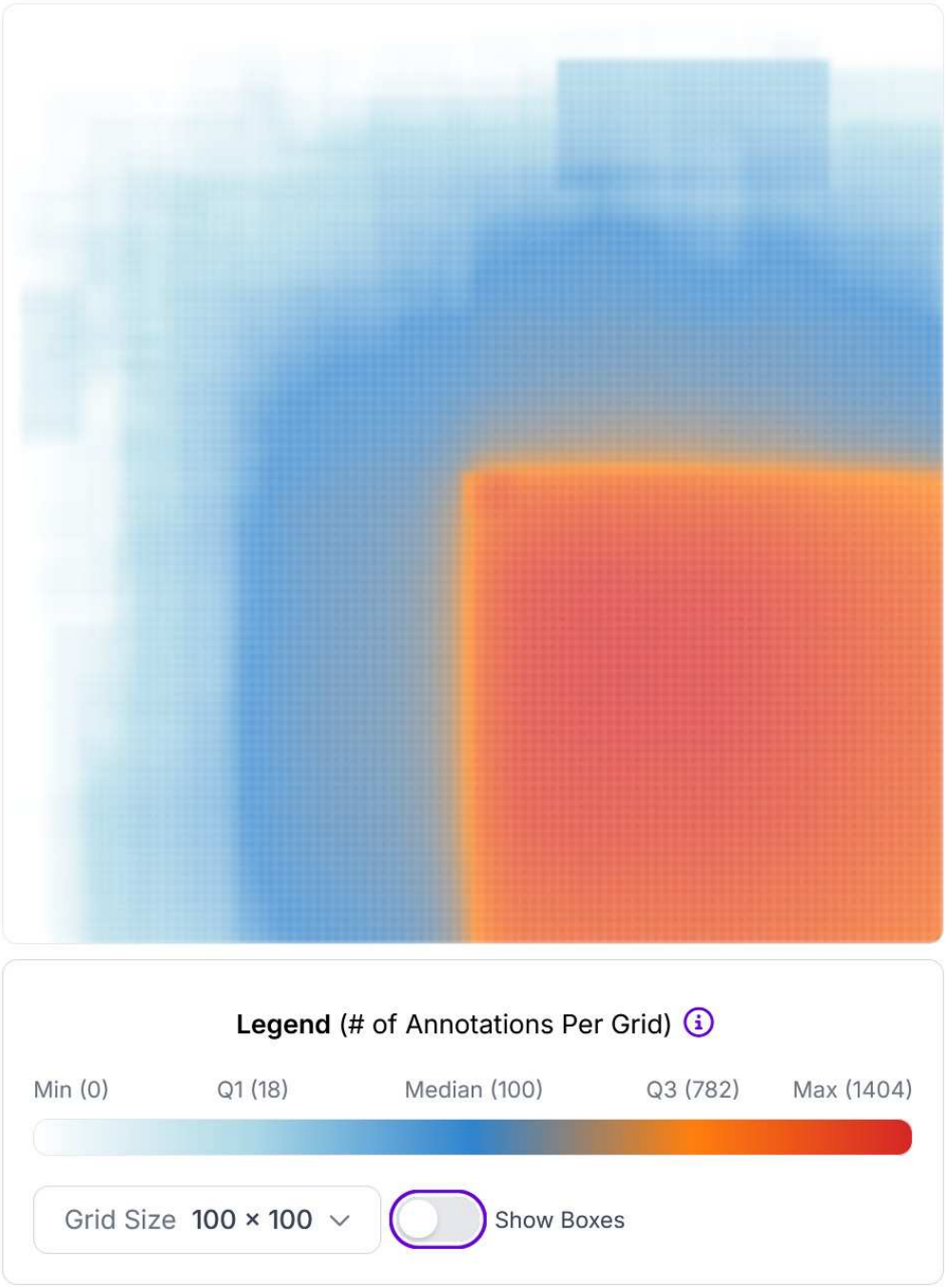


Figure 3.7: Annotation Heat Map

particularly geometric transformations like rotation and flipping, which can help to mitigate positional biases by presenting the model with objects in a wider variety of orientations and locations within the image. Additionally, the controlled data generation afforded by 3D printing, as described in Chapter 4, allows for the systematic capture of images with greater variability in object

size, distance, and position. Nevertheless, future data collection efforts should focus on acquiring images with greater variability in object scale, distance, and location within the image frame to improve the model's generalizability.

3.6. The Two-stage Augmentation Method

The initial dataset comprised a diverse collection of landmine images obtained from the internet under varied conditions, forming the foundation of this research. Preliminary alterations to the data were executed on the Roboflow platform, where the model was subsequently released [93]. A series of experiments were conducted on a dataset comprising 2,128 images that were retrieved from the Internet [92]. Multiple augmentations were implemented, including grayscale, cutoff, rotation, flip, shift, blur, and noise, tailored specifically for the YOLOv5 model [94]. At this stage, the more modern YOLOv8 model was employed [73], and different augmentation techniques were tested again. Results of experiments are available on Table 3.2.

Table 3.2: Results of experiments with preprocessing

ID	Pr.[1]	Rec.[2]	mAP[3]	Preprocessing	Augmentation
6	93.6	82.4	94.4	Auto-Orient, Stretch to 640x640	No
7	91.3	83.0	89.1	Auto-Orient, Stretch to 640x640, Adaptive Equalization	Grayscale: 30%
19	93.4	84.2	91.0	Auto-Orient, Stretch to 640x640, Adaptive Equalization	Grayscale: 30%, Cutout: 3 boxes, 21%

Continued on the next page

ID	Pr.[1]	Rec.[2]	mAP[3]	Preprocessing	Augmentation
20	91.2	81.4	88.8	Auto-Orient, Stretch to 640x640, Adaptive Equalization	Grayscale: 40%
21	92.1	81.1	89.6	Auto-Orient, Stretch to 640x640, Adaptive Equalization	Grayscale: 40%, Cutout: 3 boxes, 21%
26	89.0	84.6	88.7	Auto-Orient, Fit (white edges) in 640x640	Grayscale: 30%, Cutout: 3 boxes, 21%
27	91.7	82.1	89.5	Auto-Orient, Fit (black edges) in 640x640	Grayscale: 30%, Cutout: 3 boxes, 21 %
29	90.3	85.9	91.3	Auto-Orient, Fit 640x640	Grayscale: 30%, Cutout: 3 boxes, 21%
39	91.7	85.2	91.2	Auto-Orient, Fit 640x640, Contrast Stretching	Grayscale: 30%, Cutout: 3 boxes, 21%
41	96.1	90.2	94.3	Auto-Orient, Fit 640x640, Contrast Stretching	Grayscale: 30%, Cutout: 3 boxes, 21%
42	95.2	89.4	93.7	Auto-Orient, Fit 640x640, Contrast Stretching, Flip: Horizontal, Vertical	Grayscale: 30%, Cutout: 3 boxes, 21%

Continued on the next page

ID	Pr.[1]	Rec.[2]	mAP[3]	Preprocessing	Augmentation
47	93.7	93.0	94.8	Auto-Orient, Fit 640x640, Contrast Stretching	Grayscale: 30%, Cutout: 3 boxes 21%
48	91.6	90.6	93.9	Auto-Orient, Fit 640x640, Contrast Stretching	Grayscale: 30%, Cutout: 3 boxes 21%, Noise 25%, Sheer 15%, Crop, Rotate

[1] Precision (formula 2.5).

[2] Recall (formula 2.6).

[3] mAP: Mean Average Precision (formula 3.3).

It should be noted that not all experiments are represented in the table; the experiment ID serves as the sequence number for the series of experiments. In total, approximately fifty experiments were conducted during this stage of research. The table illustrates that, with successive iterations, the outcomes of the experiments exhibited enhancement following the implementation of diverse augmentation techniques.

It is imperative to acknowledge that no universally applicable "recipe" for methods exists, as each dataset necessitates a bespoke approach. In the context of EO detection, the augmentation methods for different types of EO may vary, necessitating the development of multiple models. To ensure expeditious recognition, it is essential to consolidate all data into a unified dataset and identify augmentation methods that are effective across the entire dataset. In scenarios where recognition speed is not a critical concern and the utilization of multiple models is feasible, it is advantageous to divide the dataset into smaller units and select parameters for each subset.

The two-stage augmentation method involves applying augmentation meth-

ods in two stages, with additional preprocessing of the data. In addition to the methods selected above, the first stage may include methods such as mosaic, noise, and others (Table 3.3).

Table 3.3

YOLO Augmentation Methods

Augmentation	**Param.**	**Description**
Mosaic	mosaic	Create a mosaic of four images
HSV Hue Shift	hsv_h	Shift hue in HSV color space
HSV Saturation Shift	hsv_s	Shift saturation in HSV color space
HSV Value Shift	hsv_v	Shift value in HSV color space
Degrees Rotation	degrees	Rotate images by specified degrees
Translate	translate	Translate images by specified values
Scale	scale	Scale images by specified factors
Shear	shear	Apply shear transformations
Flip Vertical	flipud	Flip images vertically
Flip Horizontal	fliplr	Flip images horizontally
Mixup	mixup	Apply mixup to combine images

The second stage uses augmentation methods of the YOLO algorithm during the process of training. By default, the algorithm adds mosaic augmentation for the last 10 epochs of training. The number of epochs can be increased by using the 'mosaic' parameter. All available augmentation methods for the second stage (YOLO augmentation) are available in the 3.3 table. It is important to note that all these parameters of the YOLO algorithm can be adjusted or disabled. The default values of the parameters and hyperparameters of the YOLO algorithm are enumerated in Table 4.4.

3.7. An Additional Set of Metrics to Assess the Performance of the Dataset.

This section introduces additional metrics for evaluating the performance of object detection models, particularly in the context of EO detection. To quantify the model's performance, several key metrics are used, including average precision (AP), mean average precision (mAP), and intersection over union (IoU).

Average Precision (AP) summarizes the precision-recall curve into a single value by calculating the average precision at different recall levels. It is defined as the area under the precision-recall curve, computed by integrating precision with respect to recall over the interval [0, 1]:

$$AP = \int_0^1 p(r)dr, \tag{3.1}$$

where $p(r)$ is the precision at recall level r, calculated using formula (2.5), and r is the recall, calculated using formula (2.6). In practice, this involves ranking the model's predictions by confidence and calculating the precision and recall at each position in the ranked list. The formula then becomes:

$$AP = \sum_{k=1}^{n} P(k)\Delta R(k), \tag{3.2}$$

where:

- n is the total number of relevant items (e.g., total number of ground truth bounding boxes for a specific EO type).
- $P(k)$ is the precision at cutoff k in the list of ranked retrieval results.
- $\Delta R(k)$ is the change in recall from the previous cutoff to cutoff k, which can be written as $R(k) - R(k-1)$.
- $R(k)$ is recall at cutoff k

A common variation is the interpolated AP, which uses the maximum precision for any recall value greater than or equal to the current recall level. This helps to reduce the effects of "wiggles" in the precision-recall curve.

Mean Average Precision (mAP) is calculated by averaging the AP scores across all classes:

$$mAP = \frac{1}{C} \sum_{k=1}^{C} AP_k, \tag{3.3}$$

where AP_k denotes the average precision of class k, and C is the total number of classes.

It is important to note that the reliability of mAP as a performance metric is influenced by the balance of the dataset. In a balanced dataset, where each class has a roughly equal number of samples, mAP provides an unbiased estimate of the model's performance across all classes. However, in an imbalanced dataset, where some classes have significantly more samples than others, mAP can be skewed towards the performance of the majority classes. Therefore, it is crucial to consider the class distribution when interpreting mAP, especially in applications like EO detection where certain object types may be more prevalent than others.

Intersection over union (IoU), also known as the Jaccard index, quantifies the overlap between the predicted bounding box and the ground truth bounding box. It is calculated as the area of overlap divided by the area of union, as shown in formula (2.7), or equivalently, in terms of TP, FP, and FN, as shown in formula (2.8). The closer the IoU value is to 1, the more accurately the predicted bounding box aligns with the ground truth.

mAP50 is the mean average precision calculated at an IoU threshold of 0.5 (50%). This means a prediction is considered a True Positive only if its IoU with the ground truth is 0.5 or greater.

mAP50-95 is a more stringent metric. It is calculated by averaging the

mAP values obtained at multiple IoU thresholds ranging from 0.5 to 0.95 in steps of 0.05 (i.e., at IoU thresholds of 0.5, 0.55, 0.6, ..., 0.9, 0.95). This provides a better evaluation of the model's ability to accurately localize objects across a range of overlap thresholds.

While mAP50 provides a general measure of detection accuracy, mAP50-95 offers a stricter evaluation of the model's ability to accurately localize objects. Using both metrics provides a more comprehensive understanding of the model's performance. This helps identify areas for model improvement. For example, a high mAP50 coupled with a low mAP50-95 suggests that the model is generally good at identifying the presence of EO but needs improvement in accurately predicting the precise bounding box coordinates, indicating a potential area for improvement in the bounding box regression component of the model.

F1-score

While AP and mAP provide valuable insights into the model's performance across various recall levels, it is often useful to have a single metric that balances precision and recall. The F1-score is the harmonic mean of precision and recall, providing such a balanced measure:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{3.4}$$

The F1-score ranges from 0 to 1, with 1 representing perfect precision and recall. It is particularly useful when there is an uneven class distribution or when both false positives and false negatives have significant consequences, as is the case in EO detection. A higher F1-score indicates a better balance between precision and recall.

Fitness

In the context of ML, and specifically within the YOLO framework, fitness is a metric used to evaluate the overall performance of a model during training. It is calculated as a weighted combination of key performance indicators,

designed to guide the optimization process towards models that perform well on the specific task. For this research, the fitness function used in the YOLO framework is adopted, defined as:

$$Fitness = 0.0 \times Precision + 0.0 \times Recall + 0.1 \times mAP50 + 0.9 \times mAP50\_95. \tag{3.5}$$

This formula assigns the highest weight to mAP50-95 (90%), reflecting the importance of precise localization across a range of overlap thresholds. A smaller weight is given to mAP50 (10%), while precision and recall are not directly included in this specific fitness calculation (weights of 0.0). This weighting scheme emphasizes the model's ability to accurately localize objects, which is crucial in the context of EO detection.

These aforementioned measures enable the assessment of various models' performance on the same, standardized dataset. It is important to note that model performance might vary considerably across diverse datasets. Therefore, evaluation should occur on data that accurately represents the actual application, highlighting the importance of having a representative and balanced dataset for calculating mAP and other relevant metrics. While the loss functions discussed in Section 2.9 guide the model's training process, these metrics (AP, mAP, IoU, F1-score) provide a direct and interpretable evaluation of the model's performance on the task of EO detection. The concept of fitness, although more application-specific, provides a way to combine multiple performance indicators into a single objective function for optimization and model selection.

The following table evaluates different augmentation techniques using precision, recall, mAP50, mAP50-95, and fitness metrics (Table 3.4).

Table 3.4 presents the results of applying different augmentation techniques to the dataset discussed in [92]. Subsequent to evaluating several configurations,

Table 3.4

Comparison of Augmentation Techniques

Augmentation Technique(s)	Precision	Recall	mAP50	mAP50-95	Fitness
No 1st stage, All 2nd stage (YOLO)	99	89.2	95.3	76.7	78.6
Grayscale, Rotate, Noise, Flip, All 2nd stage	97.4	92.6	95.7	77.3	79.2
Grayscale, Rotate, Noise, Flip (only 1st stage)	96.6	79	89.1	68.1	70.2
Grayscale	96.2	68.9	84.6	66.1	68
Noise	93.3	74.9	86.4	64.7	66.9
Rotate	90.8	73.3	86.6	65.7	67.8
Flip	90.2	76.1	86.9	68.4	70.2
Bounding Box Rotate	90.1	70.5	83.6	63.3	65.4
Mosaic	91.6	71.8	82	61.3	63.4
No augmentation	91.6	71.8	82	61.3	63.4
Blur	87.6	75.1	83.8	61.6	63.8
Cutout	86	63.2	80.2	61.5	63.3

the augmentations from Table 3.5 were chosen as a starting point for further testing.

Table 3.5

Augmentation Types and Variations

Augmentation	Method
Flip	Horizontal
	Vertical
90° Rotate	CW
	CCW
	Upside Down
Grayscale	Apply to 30% of images
Noise	Up to 15% of pixels

These parameters will likely be adjusted in the future, but among all the available augmentations, these, along with a few others, have demonstrated the most promising results in initial experiments. The relatively high precision and recall scores across multiple techniques suggest that the model, in general,

performs well. However, the variations in mAP50 and mAP50-95 indicate that there is room for improvement in terms of accurate bounding box localization. The Fitness metric further condenses these results into a single value, facilitating a straightforward comparison between different augmentation strategies.

3.8. Results of Applying the Two-Stage Augmentation Method

This section presents the results of training and evaluating the YOLOv5 and YOLOv8 models on the EO detection dataset [92], augmented using the two-stage augmentation strategy outlined in Section 3.3. The primary objective of this augmentation approach is to enhance the models' ability to generalize to unseen data and improve their robustness to variations in EO appearance, environmental conditions, and image quality.

The YOLOv5 and subsequently the YOLOv8 models were employed for EO recognition. Both models were trained on augmented data from each stage, as detailed in Table 3.4. The amalgamation of diverse augmentations, including spatial transformations (rotation, flipping), pixel-level variations (grayscale, noise), and geometric transformations, guaranteed that the models encountered a broad spectrum of variances during training. This comprehensive augmentation strategy aims to mitigate the risk of overfitting to the specific characteristics of the training data and improve the models' ability to generalize to unseen data.

The effectiveness of this two-stage augmentation approach is evident in the significant improvement in model performance, particularly in the crucial metric of recall. As shown in Table 3.4, the application of single augmentations alone is not sufficient to achieve optimal results. The implementation of only first-stage augmentations yields a precision of 96.6% and a recall of 79%. Conversely, the application of only second-stage augmentations attains a precision of 99% but a recall of 89.2%. Notably, the integration of both first- and second-stage

augmentations results in a precision of 97.4% and a recall of 92.6%, underscoring the necessity of considering both stages in the augmentation process.

As indicated in Section 2.9, in the field of EO detection, recall is of paramount importance. This metric directly reflects the model's ability to identify all actual EO present in an image, thus minimizing the number of false negatives (missed EO). Given the potentially life-threatening consequences of failing to detect EO, maximizing recall is a critical objective.

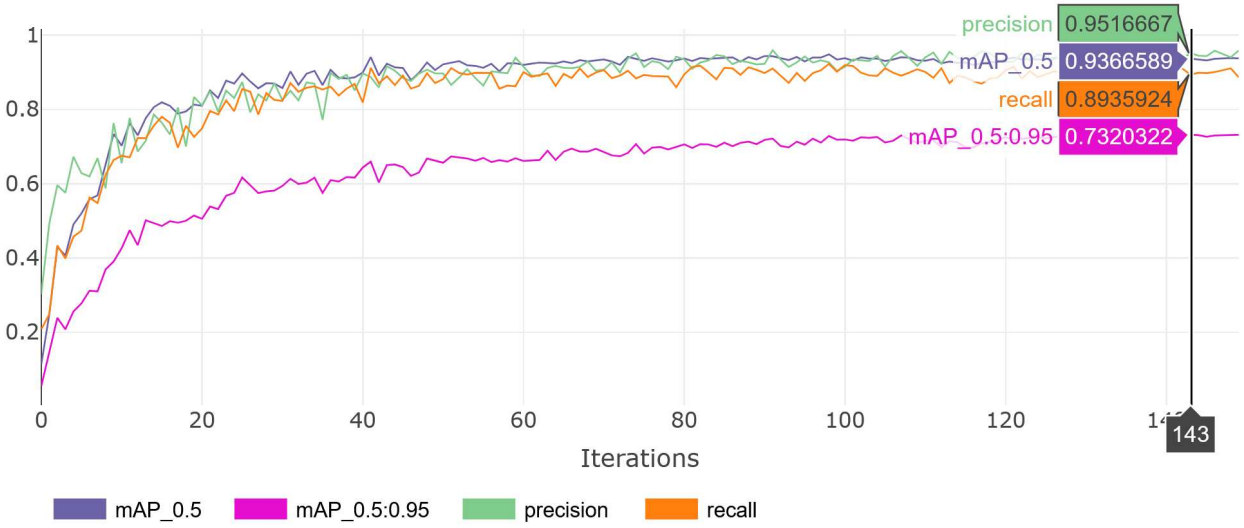


Figure 3.8: Performance metrics of the YOLOv5 model trained with two-stage augmentation for 150 epochs

Figure 3.8 illustrates the performance metrics of the YOLOv5 model trained with the augmentation techniques described in Section 3.3. Notably, the model achieved a recall of 89.3%.

Employing the more recent YOLOv8 model (Figure 3.9), further improvements were observed. With the same two-stage augmentation strategy, YOLOv8 achieved a recall of 92.9%, representing a significant improvement over YOLOv5 and demonstrating the effectiveness of the refined architecture in combination with comprehensive data augmentation.

Furthermore, it is worth noting that other key performance indicators, including Fitness (defined in formula 3.5), Box Loss, and Class Loss (calculated

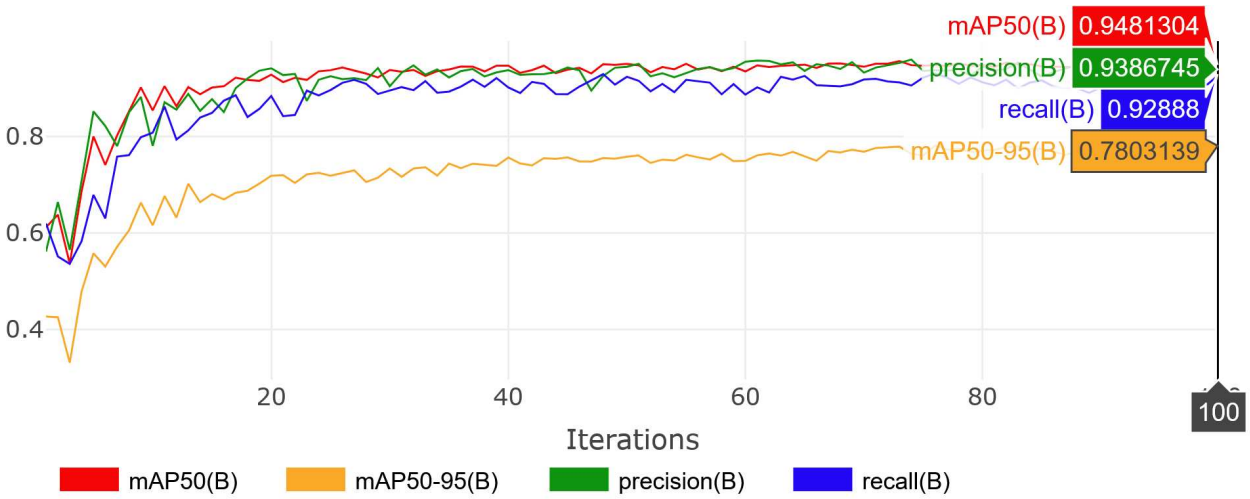


Figure 3.9: Performance metrics of the YOLOv8 model trained with two-stage augmentation

using BCE Loss, as defined in formula 2.4), exhibited desirable trends. Fitness, which reflects the overall performance considering precision, recall, mAP50, and mAP50-95, showed a general improvement with more comprehensive augmentation. Similarly, both Box Loss and Class Loss, which quantify the accuracy of bounding box predictions and class label predictions, respectively, demonstrated a general decrease across experiments, indicating improved model accuracy during training.

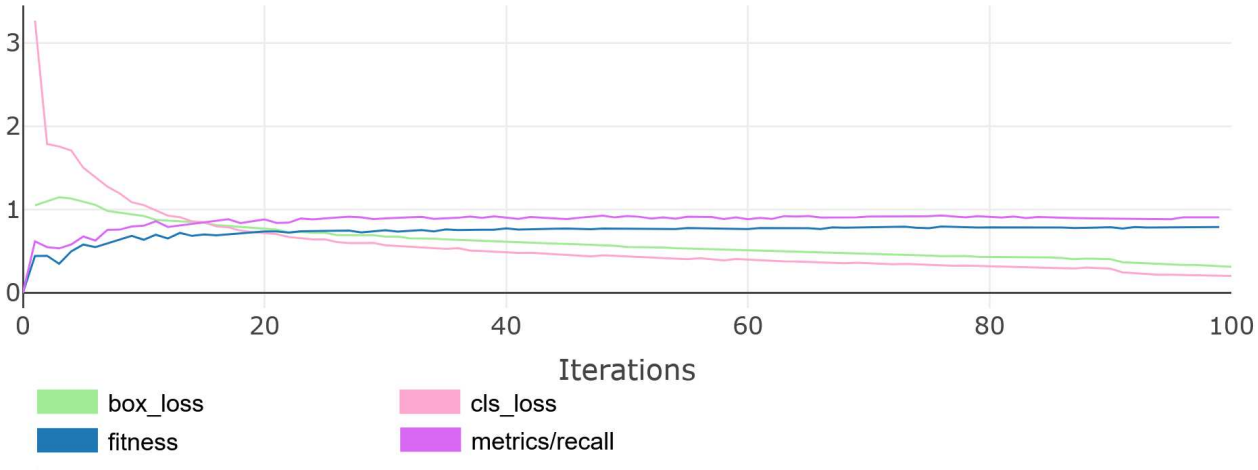


Figure 3.10: Evolution of Box Loss, and Class Loss, Fitness, and Recall during YOLOv8 training

As illustrated in Figure 3.10, the training of the YOLOv8 model demonstrates a concurrent increase in Fitness, accompanied by a synchronous decrease in both Box Loss and Class Loss. This observation indicates the efficacy of the model's training process. A salient observation is the pronounced decline in both Class and Box losses after the 90th epoch. This enhancement aligns with the default configuration of the YOLOv8 training pipeline, which disables the mosaic augmentation during the final 10 epochs. Mosaic augmentation introduces substantial variations and complexities in the initial training stages, potentially impeding the fine-tuning of bounding box and classification parameters. However, its removal in the final epochs appears to enable the model to concentrate on refining its predictions based on individual, non-augmented images. This targeted training, devoid of the augmented complexity, is hypothesized to enhance the model's ability to refine bounding box regression and classification accuracy, thereby resulting in the observed decline in losses. Additionally, the graph demonstrates a more significant increase in the recall rate during these final epochs, suggesting that the model is becoming more adept at accurately identifying EO. This finding lends further support to the hypothesis that fine-tuning the model on non-augmented data, following the initial training with mosaic augmentation, is advantageous for enhancing its performance.

3.9. Conclusions to Chapter 3

This chapter has explored the critical role of data augmentation in enhancing the performance of object detection models for the challenging task of EO detection. Through a detailed examination of various augmentation techniques, ranging from basic spatial and pixel-level transformations to advanced methods like MixUp, CutMix, and GANs, it has been demonstrated that augmentation has the potential to address the limitations of limited real-world datasets and

improve model robustness and generalization.

The experimental results, summarized in Tables 3.2 and 3.4, underscore the effectiveness of the proposed two-stage augmentation strategy. The training process began with the YOLOv5 model and progressed to the YOLOv8 model, leveraging augmented data from both the initial and advanced phases. This comprehensive approach ensured that the models were exposed to a wide variety of data transformations. It is noteworthy that the integration of diverse augmentation techniques resulted in a substantial enhancement in recall, from 89.2% without augmentation to 92.6% with the two-stage approach and employing the YOLOv8 model. This enhancement is particularly significant in the context of EO detection, where minimizing false negatives (missed EO) is of the utmost importance. The findings underscore the efficacy of the YOLOv8 architecture, particularly when augmented with effective data augmentation techniques. This augmentation led to a substantial enhancement in accuracy when compared to the earlier YOLOv5 model, as evident in Figures 3.8 and 3.9. The experiments with different types of augmentation highlighted the significance of data diversity in training robust models.

Moreover, an analysis of additional key metrics, including precision, mAP50, mAP50-95, and fitness, yielded a comprehensive understanding of the model's performance. The favorable trends observed in these metrics, along with the decrease in Box Loss and Class Loss during training (Figure 3.10), further validate the effectiveness of the augmentation strategy. The employment of a fitness function, as delineated in formula 3.5, facilitated a comprehensive evaluation of diverse augmentation techniques, thereby guiding the selection of the most promising approaches.

The comparative analysis presented in Table 3.4 revealed that while individual augmentation techniques, such as Grayscale, Noise, Rotate, and Flip, can improve model performance to some extent, their combination within a well-structured, two-stage augmentation pipeline yields the most substantial gains.

This highlights the importance of a comprehensive and diverse augmentation approach in achieving optimal results.

However, it is imperative to acknowledge the challenges and limitations associated with data augmentation in the context of EO detection. Unwarranted augmentation can inadvertently result in the creation of images that do not accurately reflect real-world mine risk scenarios. For instance, converting images to grayscale may enhance certain characteristics but may also impede the model's capacity to discern different types of EO that rely on color features for differentiation. While augmentation techniques can enrich a dataset and improve model performance, over-reliance on them can intuitively harm model performance. Excessive or inappropriate augmentation can cause the model to prioritize irrelevant features. Regular performance evaluation is crucial for monitoring and adjusting augmentation strategies. Moreover, some augmentation methods may disproportionately affect different classes in the dataset, leading to an unbalanced dataset where some classes are overrepresented. Therefore, it is very important to use augmentation methods that maintain a consistent representation of all classes.

These insights gained from this study on data augmentation will serve as a crucial foundation for the subsequent chapters. Chapter 4 will explore the use of 3D printing to generate synthetic EO images, and the augmentation techniques developed in this chapter will be instrumental in diversifying and enriching the datasets created through this novel approach. The knowledge of effective augmentation strategies will inform the creation of realistic and varied training data, ensuring that models trained on 3D-printed EO images can generalize effectively to real-world scenarios. Furthermore, Chapter 6 will detail the development of a mobile application for professional data contribution for EO data. The augmentation methods from this chapter will be applied to the data collected through the mobile application, enabling us to train robust models even with potentially limited or inconsistent data from diverse sources.

CHAPTER 4
ADDRESSING DATA SCARCITY IN EXPLOSIVE OBJECTS DETECTION WITH 3D PRINTING

4.1. Addressing Data Scarcity with 3D Printing

The creation of resilient and dependable ML models for EO detection significantly depends on the accessibility of extensive, varied, and high-quality datasets. As stated in Chapter 3, data augmentation is essential for improving model performance; yet, it cannot entirely mitigate the inherent constraints caused by the limited availability of authentic EO images. This shortage arises from the logistical, ethical, and safety issues associated with gathering data on active EO in the field.

This chapter examines the novel application of 3D printing technology to produce synthetic datasets for training EO detection models, thereby addressing this significant bottleneck.

The fundamental concept is to utilize the capabilities of 3D printing to produce precise physical copies of several EO varieties. These replicas, derived from accessible 3D models, can thereafter be photographed under regulated conditions, emulating diverse surroundings, backdrops, and image characteristics. This method presents numerous significant benefits. Firstly, it facilitates the generation of an ostensibly infinite quantity of images, so successfully mitigating the issue of data scarcity. Furthermore, it affords comprehensive control over the data production process, facilitating the systematic alteration of variables including EO kind, orientation, burial depth, surrounding vegetation, and lighting conditions. Achieving this degree of control is challenging, if not unattainable, when gathering data in the field. Thirdly, it mitigates the risks

and ethical dilemmas connected with managing live explosives.

This chapter delineates the process and findings associated with employing 3D printing to generate a synthetic dataset for EO identification. The procedure for selecting and modifying 3D models of anti-personnel landmines is outlined, specifically the PFM-1, PMN, PMN-2, OZM-72, and MON-50, selected for their commonality and varied attributes. The models are subsequently 3D-printed, followed by a series of experiments to capture images of the replicas in diverse conditions. These circumstances are designed to replicate real-world scenarios. The chapter elaborates on the methods for developing such situations.

The resultant synthetic dataset is utilized alongside the augmentation strategies outlined in Chapter 8 to train and assess the efficacy of YOLO object detection model. Through the regular alteration of 3D printing parameters and image acquisition conditions, this research intends to generate a dataset that is both extensive and diverse, while accurately reflecting the obstacles faced in actual EO detection situations.

This chapter's findings illustrate the considerable potential of 3D printing to mitigate data shortages and facilitate the development of more efficient and dependable EO detection systems. The capacity to produce synthetic data in regulated environments creates new opportunities for developing resilient models that can effectively generalize to real-world difficulties. This research ultimately aids in the overarching objective of creating automated solutions to expedite humanitarian demining initiatives and improve the safety of impacted communities.

4.2. Methodology: Using 3D Printing for Dataset Generation

This section details the methodology employed for creating a dataset for EO detection using 3D-printed replicas. The method's foundation is the generation of realistic 3D-printable models of EO and the systematic capture of images of

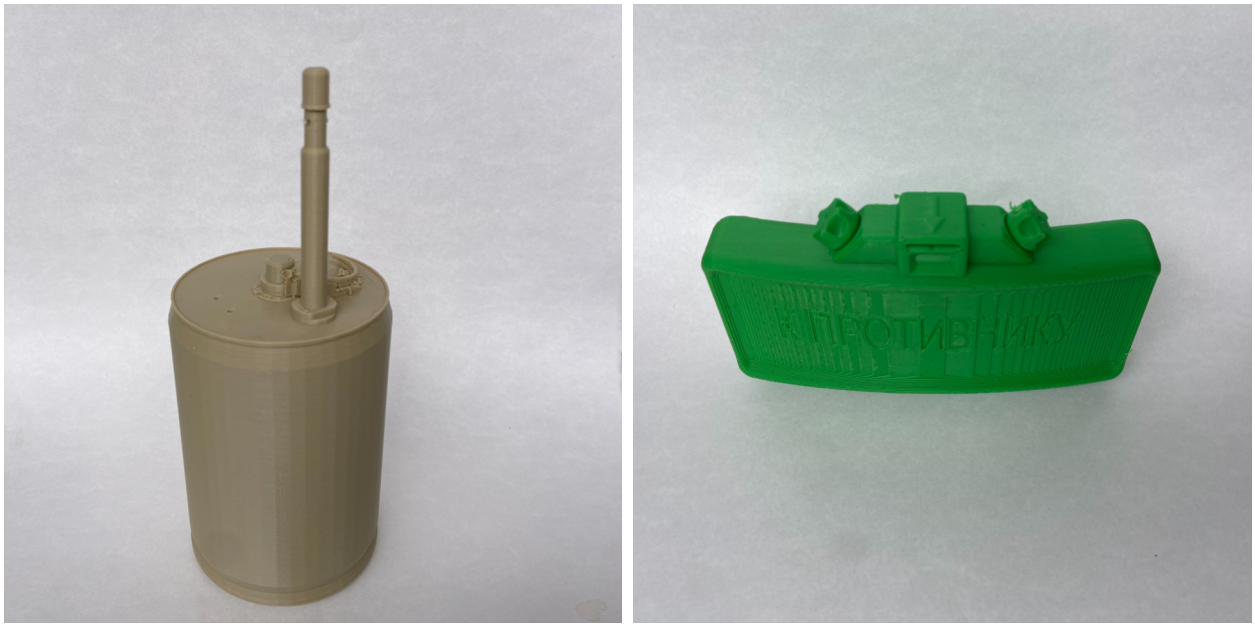
these replicas under various conditions to compile a diverse and representative dataset.

The process commences with the acquisition or creation of accurate 3D models of EO. This may involve using existing online repositories of 3D models or developing custom models based on technical specifications and images of real EO. Subsequently, these models are used to fabricate physical replicas of the EO using a suitable 3D printing process and material. The choice of printing parameters (e.g., resolution, material) can be adjusted to control the fidelity and characteristics of the replicas. Optionally, the physical characteristics of the 3D-printed replicas can be modified, such as size, shape, and color. This step introduces further variations into the dataset and can enhance the model's robustness to anomalies and variations in real-world EO. For instance, Figure 4.1 illustrates examples of landmine replicas that have been scaled down in size to test the hypothesis that modified objects can still be effectively used for model training.

Following replica preparation, controlled photographic experiments (surveys) are conducted to capture images of the 3D-printed replicas under a variety of conditions, including varying environmental factors (lighting, weather, background), object positioning (orientation, burial depth), and camera parameters (angle, distance). Finally, the collected images are organized and labeled to create a structured dataset suitable for training computer vision models. This process involves assigning appropriate class labels and potentially bounding box annotations to each landmine replica in the images.

Furthermore, the ability to modify physical characteristics of the replicas, such as size, shape, and color, provides an additional dimension for data augmentation. By systematically altering these properties, variations in EO appearance due to factors like manufacturing differences, weathering, or partial obscuration can be simulated. This capability enhances the dataset's diversity and improves the robustness of trained models to anomalies that might be

encountered in real-world scenarios.



(a) OZM-72 (b) MON-50

Figure 4.1: Examples of 3D-printed landmine replicas scaled down in size to evaluate the feasibility of using modified objects for model training

This 3D printing-based methodology offers a practical and ethical solution to the data scarcity challenge in EO detection. By generating synthetic data under controlled conditions, this research can create datasets that are both extensive and representative of the complexities encountered in real-world scenarios. The subsequent sections provide a detailed account of the implementation of this method, encompassing the selection of EO types, 3D printing procedures, image acquisition protocols, and the outcomes of model training using the generated dataset.

4.3. Objective of the 3D Printing Technique

The primary objective of the research presented in this chapter is to assess the efficacy of employing 3D-printed landmine replicas for training computer vision models to address the issue of data scarcity. This will enable the de-

velopment of a framework for acquiring ML models derived from 3D-printed replicas. The resultant models can be utilized to detect actual EO, with potential applications in UAV-based or robotic detection systems.

To attain this objective, the following specific aims are established:

— Develop a dataset for training computer vision models, consisting of photographs of 3D-printed replicas of prevalent anti-personnel landmines in Ukraine. The dataset will encompass diverse weather conditions, including clear, foggy, rainy, and snowy environments.

— Train and optimize models on the generated dataset. This involves expanding the dataset through augmentation techniques (as detailed in Chapter 3) and fine-tuning model hyperparameters, such as learning rate and batch size, for optimal performance.

— Evaluate the trained models' performance on a separate dataset of authentic landmine photographs provided by domain experts, using the metrics defined in Section 2.9 to assess their real-world applicability.

4.4. Primary Hypotheses of the Methodology

This study examines the efficacy of employing 3D-printed EO replicas, as described in the methodology outlined in Section 4.2, to train computer vision models for EO identification. The primary hypotheses guiding this research are:

Hypothesis 4.1. 3D-printed replicas, as detailed in Section 4.2, can effectively train computer vision models for successful EO identification in real-world scenarios.

Hypothesis 4.2. Models initially trained on 3D-printed replicas can be further enhanced with real-world data by including additional data into the dataset, augmenting the number of training epochs, and fine-tuning model hyperparameters.

The following assumptions underpin this research:

— The 3D-printed replicas, created using the methodology outlined in Section 4.2, accurately replicate the essential shape, texture, and visual attributes of actual landmines relevant for visual detection.

— The diverse survey settings (variations in illumination, backdrop, and angle) employed during the image acquisition of the 3D-printed replicas, as described in Section 4.2, provide adequate data variability for effective model training.

For clarity, the following simplifications were adopted in this study:

— Certain landmine models were reduced in scale during the 3D printing process (see Section 4.2), potentially impacting detection accuracy. This potential impact will be analyzed in the results.

— The models were trained exclusively on visual data (photographs), disregarding other potential detection modalities (e.g., magnetic properties).

— The detection performance for replicas of metal landmines might differ from those fabricated from metal due to variations in material properties, such as metallic luster and texture. This will be taken into account during the evaluation.

— The study employs photographs captured under various weather conditions and from diverse viewpoints. However, this does not encompass all potential real-world permutations. The effectiveness of EO recognition may improve with a larger and more varied dataset, which will be explored in future work.

In this research, replicas of landmines frequently used in Ukraine were fabricated, and a dataset comprising photographs of these replicas was compiled. These replicas were utilized to train a computer vision model, employing their varied appearances and simulated settings to emulate real-world scenarios.

4.5. Rationale for Model Selection in Explosive Objects Detection

To ensure the relevance and applicability of this research to the war in Ukraine, the selection of EO models was based on a careful assessment of the types being utilized in the region. The ongoing war in Ukraine has seen the widespread use of anti-personnel landmines by Russian forces. Reports from humanitarian organizations and military sources indicate that a minimum of 13 varieties of anti-personnel mines are being employed [98]. These EO are positioned on the ground, occasionally concealed, but in numerous instances, they are at least partially visible. This visibility, even when partial, makes them potentially detectable using computer vision techniques, which is the focus of this research.

For this study, the following landmine types were selected due to their extensive prevalence in the conflict zone, their surface-laid deployment (making them visually detectable), and the significant threat they pose to civilians [96]:

— **PMN: (Figure 4.2a)** This landmine is colloquially referred to as a "Widow" owing to its substantial trinitrotoluene concentration (200 grams), which typically leads to fatalities upon detonation.

— **PMN-2: (Figure 4.2b)** Despite containing just half the quantity of trinitrotoluene compared to the PMN, this landmine is extensively utilized and manufactured in many countries.

— **OZM-72: (Figure 4.1a)** This landmine commonly known as the "Jumping Witch" or "Frog." Upon activation, the OZM-72 ascends to a height of 60 to 80 cm, detonating at torso level, so amplifying the potential harm to individuals in proximity.

— **MON-50 (Figure 4.1b):** Characterized by its distinctive design this landmine is frequently concealed beneath grass or foliage. The landmine possesses an extensive damage radius of 50 meters.

— **PFM-1 (Figure 4.2c):** Prevalent, typically impacting the feet or hands. This landmine, dispersed from helicopters or rockets, can include extensive regions, including tree canopies, vegetation, roofs, or building facades. The various colors of the landmine can captivate children's attention.

Table 4.1 presents the physical characteristics and triggering mechanisms of the selected landmine models.

Table 4.1

Characteristics of Selected Landmine Models

Landmine	Type	Triggering Mechanism	Visual Features
PMN (4.2a)	Blast	Pressure	Round, often brown or green
PMN-2 (4.2b)	Blast	Pressure	Round, often brown or green, with a distinctive pressure plate
OZM-72 (4.1a)	Fragmentation	Tripwire or command	Cylindrical, often with visible fragmentation elements
MON-50 (4.1b)	Fragmentation	Command or tripwire	Rectangular, directional fragmentation mine
PFM-1 (4.2c)	Blast	Pressure	Small, "butterfly" shape, often brightly colored

(a) PMN (b) PMN-2 (c) PFM-1

Figure 4.2: Examples of 3D-printed landmine replicas originally insize

To provide a visual representation of these threats, Figure 4.1 displays the scaled-down replicas of the MON-50 and OZM-72 landmines, while Figure 4.2 showcases the in-size replicas of the PMN, PMN-2, and PFM-1 landmines.

It is important to note that all anti-personnel landmines are prohibited under international law, specifically the Convention on the Prohibition of the

Use, Stockpiling, Production and Transfer of Anti-Personnel Mines and on their Destruction (the Anti-Personnel Mine Ban Convention), adopted in 1997 [97].

The majority of antipersonnel landmines, including those examined in this study, lack self-deactivation capabilities. This indicates they can remain operational for numerous years post-installation. Therefore, the selection of these specific EO models ensures that this research addresses some of the most pressing and impactful aspects of the EO threat in Ukraine, providing a foundation for developing effective detection models.

4.6. The Process of 3D Printing and the Tools Used in the Study

3D printing technology facilitates the production of detailed replicas of specific landmines, thereby enhancing research initiatives. This section details the methodologies and instruments employed in the study. Software:

— **Blender** (The Netherlands) [99]: An open-source software utilized for the creation and modification of 3D models, Blender was employed to construct and alter 3D representations of the chosen EO.

— **GIMP** (USA) [100]: This complimentary and open-source graphic editor was utilized for post-processing and refinement of the landmine model photographs.

Device: the Prusa i3 MK3S+ (Czech Republic) [101] The 3D printer was selected for its precision, dependability, and capacity to replicate intricate components. This printer comes bundled with slicing software and other necessary utilities for 3D printing.

Specifications for printing:

— **Infill:** the models were printed with a minimum infill of 5 – 15%, which did not compromise their strength.

— **Size modification:** certain models, such the OZM-72 and MON-50, were diminished to enhance compatibility with printing capabilities.

The simplification from section 4.4 that recognition quality would remain unaffected by size reduction was evaluated on these landmines.

The publicly accessible models utilized in this study are enumerated in Table 4.2.

Table 4.2

Examples of Printed Models

ID	Qty	Color	Infill	License	Author	Source
PMN	1	Green	5%	CC BY-NC-ND	mussy	[102]
PMN-2	2	Green	5%	CC BY, CC BY-NC-SA	Håkon Benjaminsen, Jonathan Lavoie	[103], [104]
OZM-72	1	Gray	5%	CC BY	valterjherson1	[105]
MON-50	1	Green	5%	CC BY-NC-ND	mussy	[106]
PFM-1	3	Gray	15%	CC BY-NC-ND	mussy	[107]

Table 4.2 lists the authors, the license, and the infill % utilized during the model's printing process.

4.7. Rationale for Selecting the YOLO Model

Having established the methodology for creating a dataset of 3D-printed landmine replicas (Sec. 4.8), the next critical step is selecting an appropriate computer vision model for training and evaluation. Detecting EO with computer vision is a sophisticated endeavor that necessitates high precision, dependability, and real-time or near-real-time processing capabilities. A review was done to assess the suitability of current computer vision models for object detection in EO detection.

Summary of the models:

— **R-CNN** ([108]) and its derivatives (Fast R-CNN [109], Faster R-CNN [110], Mask R-CNN [111]) exhibit lower inference speeds due to their multi-stage process of region proposal and feature extraction, yet achieve high accuracy.

— **SSD** (Single Shot MultiBox Detector [112]): This model outperforms the original R-CNN in speed and simultaneously predicts object locations and classifications. However, its precision can be lower, particularly for small objects.

— **YOLO** (You Only Look Once [72]): A rapid and precise object detecting algorithm. It processes images in a single pass (hence the name 'You Only Look Once'), rendering it suitable for real-time applications.

Selection of models: This stage of the study used the YOLOv8 model [73] due to its several characteristics that render it especially appropriate for EO identification.

— **Strong Balance of Precision and Speed:** YOLOv8 offers a strong balance between precision and speed, making it well-suited for real-time object detection tasks like those required for UAV-based EO surveys.

— **Real-time Processing:** YOLOv8's ability to process images in a single pass enables real-time performance, which is crucial for rapid EO detection.

— **Adaptability:** YOLOv8's architecture can be readily adapted to detect the specific visual features and varying sizes of different EO types.

— **Reduced False Positives:** YOLO models, in general, demonstrate a reduced tendency to generate false positives in the absence of objects, a critical factor in minimizing unnecessary and potentially dangerous interventions in safe areas.

It should be noted that at the time of this research, YOLOv11 was available. However, the methodology and findings presented here are expected to be applicable to newer YOLO versions as well, demonstrating the flexibility of the approach and its adaptability to advancements in the underlying detection algorithm.

The YOLO algorithm was introduced in 2015 by Redmon et al. [72]. The approach computes the bounding boxes and corresponding class probabilities

directly from the complete images in a single pass. The algorithm analyzes images in real time at a rate of 45 frames per second. Fast YOLO, a smaller and faster variant of the original model, can process up to 155 frames per second while achieving double the mean average precision of other real-time detectors [72]. While YOLO may exhibit slightly higher localization errors compared to some two-stage detectors, it demonstrates a significantly reduced propensity to generate false positives in the absence of objects. This renders it a dependable option for applications like EO detection, where minimizing false alarms is crucial. One of the most popular iterations, YOLOv8 [73], has been utilized across multiple domains. These encompass real-time aerial object identification for surveillance [113], tracking individuals who shoot [114], and high-resolution aerial imaging [115]. The technique has demonstrated potential in medical applications, including brain tumor identification [116] and real-time arrhythmia monitoring [117]. Furthermore, it has been employed in vehicular security via license plate and vehicle model recognition [118].

To evaluate the effectiveness of the trained model in detecting EO, a two-pronged testing approach was employed. First, the model was tested on additional images of the 3D-printed replicas, captured in various locations and under differing conditions. This allowed us to determine whether additional training data or further model refinement was necessary. Second, the model underwent evaluation using a distinct dataset comprised of photographs of actual landmines sourced from professional deminers. This evaluation on real-world data assessed the model's ability to learn from replica images and generalize that learning to real images.

4.8. Dataset of 3D-Printed Explosive Objects

Building upon the methodology outlined in Section 4.2, this section describes the dataset of 3D-printed landmine replicas created for training and eval-

uating the object detection models. The dataset contains 1,438 photographs, encompassing a range of conditions, including sunny days, cloudy days, mornings, evenings, rain, fog, and snow, with an approximately equal distribution across five landmine types: PMN, PMN-2, OZM-72, MON-50, and PFM-1. This study involved the compilation of a comprehensive dataset captured under various conditions to simulate potential real-world scenarios. The purpose of this dataset is to train and evaluate computer vision models for EO detection, addressing the data scarcity issue discussed in Section 4.1.

The photographs were captured using a 64-megapixel camera, with standard settings of ISO 50, 24 mm focal length, F1.8 aperture, and a shutter speed of 1/60. These settings were chosen to maintain a consistent depth of field and ensure sufficient image sharpness under various lighting conditions. Multiple resolutions were employed during image acquisition (3468×4624, 3000×4000, 3456×3456), with a common square resolution of 3456x3456. For initial model training, the original images were standardized to a resolution of 640x640 pixels, which is a common input size for YOLO models and facilitates the use of augmentation techniques discussed in Chapter 3. Subsequently, the trained models were also evaluated on higher resolution images (1280x1280 pixels) to assess their performance on actual landmine photographs.

The photographs were captured in natural surroundings, including forest floors, gravel paths, sandy terrain, and areas adjacent to water bodies. Images were taken under varying lighting situations (e.g., direct sunlight, shade, overcast) and included distracting elements (e.g., apples, garbage, stones) to evaluate the model's ability to distinguish EO from common objects. Different types of vegetation, such as tall grass, leaf litter, and bushes, were also included to assess the model's performance in varied environments. Monochromatic photographs were incorporated to mitigate the effects of color discrepancies between the 3D-printed replicas and real landmines, and to evaluate the model's ability to identify EO based on shape and texture alone, as color may not always be a

Figure 4.3: Images in the dataset: *a* – OZM-72 on a snowy backdrop; *b* – PMN-2 on a snowy surface; *c* – PMN-2 on a layer of leaves above an asphalt surface; *d* – PFM-1, camouflaged to blend with the surrounding stones; *e* – MON-50 positioned on a tree in a monochrome depiction; *f* – a monochrome representation of a PMN landmine in a foliage adjacent to a fallen tree trunk; *g* – PMN in a creek; *h* – PMN-2 surrounded by apples; *i* – PFM-1 in a trashed environment

reliable feature in real-world scenarios.

This diverse dataset, encompassing various EO types, environmental conditions, and imaging parameters, provides a robust foundation for training and evaluating the performance of object detection models for EO detection. The dataset will be used to train models, as described in the following section.

4.9. Preparing the Dataset: Annotation, Processing, and Augmentation

Having established the methodology for creating the 3D-printed landmine replica dataset (Sec. 4.8), the next crucial step involved annotating the collected images and preprocessing the data for model training. The gathered images were subjected to annotation, which involved drawing bounding boxes around each landmine replica and assigning the appropriate class label (e.g., PMN, PMN-2, OZM-72, etc.). Several tools and platforms were evaluated for the purpose of generating annotations.

The evaluation of platforms' functionalities for producing EO detection datasets is as follows:

— **LabelImg** ([119]) is a straightforward and complementary tool suitable for minor tasks or the preliminary phase of annotation (e.g., pilot annotation or initial data exploration). However, it possesses restricted functionalities and lacks automation support.

— **makesense.ai** ([120]) is a web-based application for annotating datasets. It provides an intuitive interface and accommodates various forms of annotations, rendering it suitable for medium-sized projects. Nevertheless, it did not support persistent storage of annotation results at the time of the study.

— **TORAS (Toronto Annotation Suite)** ([121]) is a web-based platform for picture annotation that incorporates technologies for interac-

tive segmentation and contemporary AI models, such as SAM (Segment Anything Model), to expedite the annotation process. While TORAS possesses user-friendly annotation capabilities, at the study's initiation, it imposed a 1GB data size restriction, whereas the data amount surpassed 3GB.

— **Roboflow** ([93]) is a comprehensive solution that encompasses the entire process from image collecting and annotation to model training and deployment on devices. It provides automatic annotation. However, the web interface for annotation exhibited slower performance during annotation tasks.

— **Google Cloud Platform** ([122]) facilitates model creation and provides annotation tools, but it offers little control over the training process, such as limitations on model selection, training methodology, and hyperparameter configuration.

— **Microsoft Azure** ([123]) allows for the generation of models using pre-trained models, which accelerates the development process; however, it offers fewer customization possibilities relative to other platforms, including a streamlined training procedure and hyperparameter configurations.

— **Amazon SageMaker** ([124]) offers workflow management and automated annotation capabilities; however, it necessitates the utilization of Amazon Web Services, which might be challenging to navigate.

None of the platforms fully met the criteria of this study. Therefore, a custom annotation application was developed to meet the specific requirements of this research, offering greater flexibility and control over the annotation process. The application generated annotations in the YOLO format, consisting of text files with bounding box coordinates and class labels for each image. Data was initially stored using a self-developed application on Google Cloud Storage ([122]), where photographs were stored in both 640x640 and 1280x1280

resolutions to facilitate both initial model training and subsequent evaluation on higher-resolution images. The annotated data was then uploaded to the Roboflow platform for data augmentation. The augmentations specified in Table 4.3 were implemented using Roboflow's built-in tools.

Table 4.3

Roboflow Augmentations and Their Purpose

Augmentation Name	**Purpose of Use**
Flip (Horizontal, Vertical)	To prevent memorizing the positions of objects and improve model robustness to object orientation.
90° Rotate (CW, CCW, Upside Down)	To prevent memorizing the positions of objects and improve model robustness to object orientation.
Grayscale (Apply to 45% of images)	To address varying object scales, enhance robustness to color variations, and improve performance under challenging recognition conditions.
Noise (Up to 5% of pixels)	To address occlusion of small objects, simulate sensor noise, and improve model robustness to image imperfections.
Mosaic	To enhance data variety by combining multiple images into a single training sample, simulating cluttered scenes and improving object detection in complex backgrounds.

Table 4.3 delineates the objective of each augmentation approach employed in this study. Compared to the initial augmentation strategy outlined in Table 3.5, Mosaic augmentation was added to further enhance dataset diversity. Ad-

ditionally, the proportion of images subjected to Grayscale augmentation was increased from 30% to 45% to improve robustness to color variations, and the intensity of Noise augmentation was reduced from 15% to 5% of pixels to better reflect realistic noise levels encountered in the target application.

4.10. Training the YOLOv8 Model on the 3D-Printed Landmine Dataset

This section details the process of training the YOLOv8 object detection model on the dataset described in Section 4.8, utilizing the augmentation strategies outlined in Chapter 3. The primary goal of this training process is to develop a robust model capable of accurately detecting EO in images.

The training process was conducted using a Tesla T4 GPU, provided through the Google Colab cloud computing platform, to leverage its computational power and accessibility for accelerated model training. At this juncture, the YOLOv8 algorithm is employed, using augmentations from Table 3.5. Table 4.4 presents the default augmentation parameters available in YOLOv8, along with their descriptions. The "Value" column indicates the default parameter settings in the YOLOv8 framework. For the experiments, the augmentations specified in Table 3.5 were primarily utilized, with modifications as described in Section 4.9 and Table 4.3.

The model's performance was evaluated using the metrics defined in Section 2.9, including precision, recall, and mAP50 (see Table 4.5).

Upon selecting the types of landmines and identifying their models in the public domain, 3D replicas were produced, with several copies created for certain models. In the trials, the landmines were coated in various colors, positioned at diverse angles, partially overlapped, and photographed with other items. The dataset creation procedure was executed iteratively through a series of surveys (100 – 200 new images each), experiments, and model training, using

Table 4.4

Default YOLOv8 Augmentation Parameters and Their Descriptions

Key	Value	Description
hsv_h	0.015	Image HSV-Hue augmentation (fraction)
hsv_s	0.7	Image HSV-Saturation augmentation (fraction)
hsv_v	0.4	Image HSV-Value augmentation (fraction)
degrees	0.0	Image rotation (+/- degree)
translate	0.1	Image translation (+/- fraction)
scale	0.5	Image scale (+/- gain)
shear	0.0	Image shear (+/- degree)
perspective	0.0	Image perspective (+/- fraction), range 0-0.001
flipud	0.0	Flip image up-down (probability)
fliplr	0.5	Flip image left-right (probability)
mosaic	1.0	Image mosaic (probability)
mixup	0.0	Image mixup (probability)
copy_paste	0.0	Copying part of images one to one (probability)

augmentation techniques, outcome evaluation, and target setting for further iterations. Table 4.5 encapsulates the findings of multiple trials demonstrating the impact of picture quantity, diverse augmentations, and configurations on the model's performance.

Table 4.5: Comparison of the Results of Different Experiments

ID	Qty	Sz.[1]	Pr.[2]	Rec.[3]	mAP[4]	Sp[5]	Note
1	195	640	74.9	71.2	80.8	2	Prepr. 3.6 and Aug. 3.5 (1st stage), Aug. 4.4 (2st stage)

Continued on next page

Table 4.5 – continued from previous page

ID	Qty	Sz.[1]	Pr.[2]	Rec.[3]	mAP[4]	Sp[5]	Note
2	130	640	72.5	64.9	66.7	4	Same as 1 Prepr. 3.6 and Aug. 3.5 (1st stage), Aug. 4.4 (2st stage)
3	250	640	89.4	84.9	94.5	14	Same as 1
3.1	250	640	90.8	84.3	91.5	14	Same as 1, but Grayscale 100%
4	400	640	74.9	71.2	80.5	20	Same as 1
4.1	400	640	95.0	91.9	93.4	25	As 1, but Grayscale incr. to 60% and Noise decr. to 2%
5	500	640	93.8	89.8	94.1	28	Same as 1
6	500	640	92.9	86.6	92.8	48	Same as 1, but Noise decr. to 5%
6.1	500	640	95.2	94.6	97.6	48	As in 6, but Grays.: 45%, Mosaic
7	800	640	91.9	88.6	94.1	45	Same as 1
7.1	800	1280	96.5	91.6	96.4	170	As 1, but Resize 1280x1280
8	1000	640	96.8	96.4	97.8	59	Same as 6.1
8.1	1000	640	97.3	94.8	98.4	65	As 6.1, but YOLOv8 Aug., Copy-Paste (0.5)
9	1000	1280	98.5	97.7	98.9	240	As 6.1, but Resize to 1280x1280

Continued on next page

Table 4.5 – continued from previous page

ID	Qty	Sz.[1]	Pr.[2]	Rec.[3]	mAP[4]	Sp[5]	Note
10	1438	640	97.9	96.7	98.7	120	Same as 6.1
11	1438	1280	98.0	98.2	99.3	240	As 6.1, but Resize to 1280x1280

[1] Resolution of the side of the square image in pixels.

[2] Precision (formula 2.5).

[3] Recall (formula 2.6).

[4] mAP: Mean Average Precision (formula 3.3).

[5] Average image processing speed per second.

Table 4.5 illustrates that during the studies, the model's performance progressively improved; nevertheless, the picture processing time per second, and consequently the total time, also escalated.

The YOLOv8 algorithm was trained on a dataset comprising 1,438 images, and the implementation of enhancement techniques resulted in an augmented image count of 3,452. The photographs were allocated into training (3,021, or 87.5%), validation (287, or 8.3%), and test (144, or 4.2%) groups. During the testing phase, the augmentation methods provided in 4.4 were adjusted to enhance model performance. The model demonstrated 98.0% precision and 98.2% recall, indicating a favorable outcome on the application of 3D-printed models for landmine detection. The model's training process was conducted iteratively, with adjustments made to the batch size of images processed per epoch (a complete run through the training dataset) and exploration of various data augmentation techniques. A batch size of 32 was employed for images with a resolution of 640×640 pixels. However, for images with higher resolution (1280×1280), the maximum batch size was constrained to 8 due to limitations in computational resources, leading to a decline in learning performance by approximately fourfold (refer to Table 4.5 for trials 7 and 7.1, 8 and 9). The

incorporation of supplementary augmentations, including Mixup and Copy-Paste, led to an enhancement in precision but a reduction in model recall (see Table 4.5: Experiments 8 and 8.1). However, given the heightened significance of recall in the EO detection task, these augmentations were excluded from the final model. The incorporation of mosaic augmentation, which amalgamates several images into a single entity, markedly enhanced the model's performance, attaining a precision of 98.0% and a recall of 98.2%.

4.11. Results and Discussion

Following the training methodology described in Section 4.10, the model was evaluated on the 3D-printed EO dataset. This section presents the results of this evaluation, focusing on the key metrics defined in Section 2.9. Specifically, the results from Experiment 8.1 are analyzed (Table 4.5), which employed the two-stage augmentation strategy with Copy-Paste augmentation.

Figure 4.4 illustrates the evolution of precision, recall, mAP50, and mAP50-95 during the training of the model over 100 epochs. As shown in the figure, precision, recall, and mAP values generally increased over the course of training, stabilizing around epoch 80. This indicates that the model progressively learned to better detect EO with increasing exposure to the training data. However, it should be noted that the number of training epochs used in this study (100 – 150) was chosen as sufficient for the experiments. In reality, the number of epochs should be 1,000 to 3,000, depending on the learning speed.

Figure 4.5 presents the normalized confusion matrix for the validation dataset, obtained from the YOLOv8 model trained on the 3D-printed landmine dataset (Experiment 8.1). The matrix provides a detailed breakdown of the model's performance across different EO classes.

The confusion matrix reveals that the model achieved high precision for most landmine classes, particularly for MON-50 (100%, class 14) and PMN-2

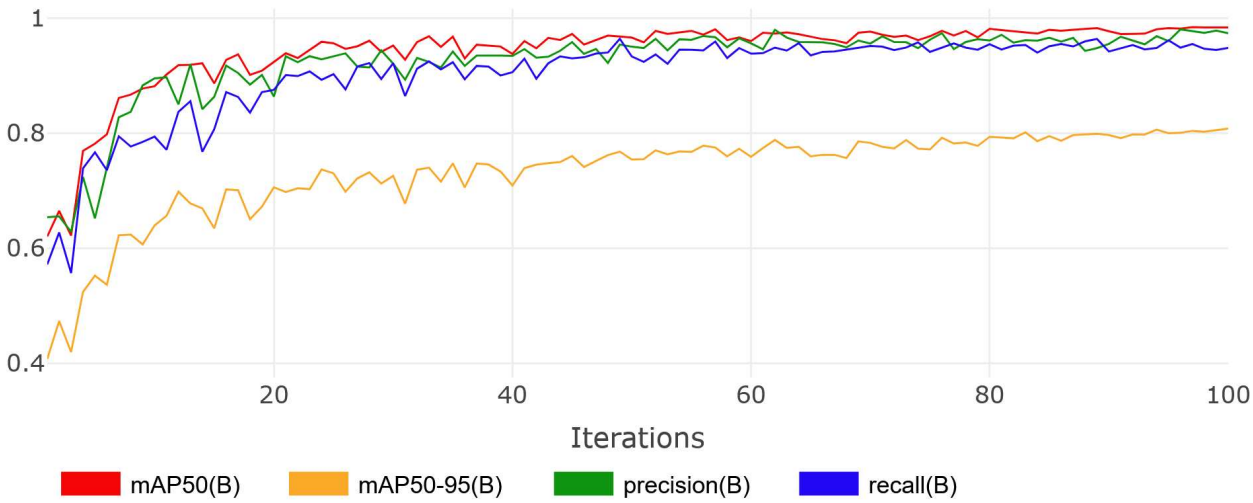


Figure 4.4: Evolution of precision, recall, mAP50, and mAP50-95 during the training of the model over 100 epochs (Experiment 8.1, Table 4.5)

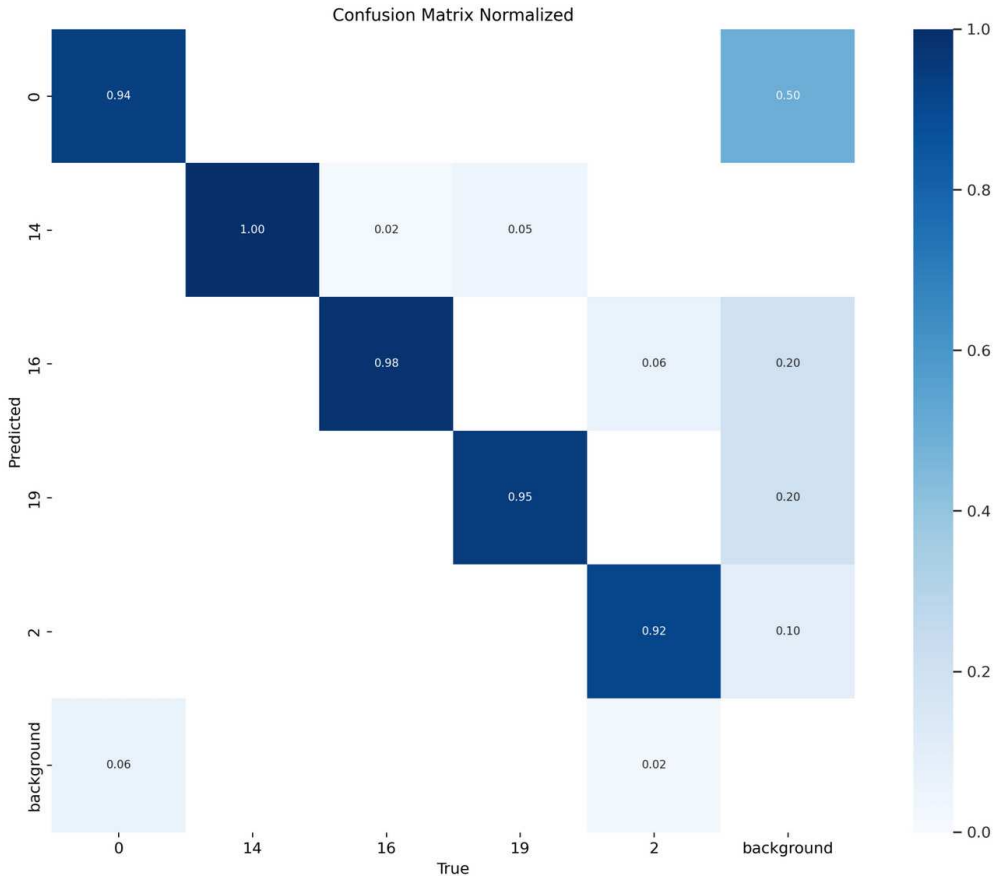


Figure 4.5: Normalized confusion matrix for the model on the validation dataset of 3D-printed landmines (Experiment 8.1). Class labels: 0 – PFM-1; 2 – PMN; 14 – MON-50; 16 – PMN-2; 19 – OZM-72

(98%, class 16). These results suggest that the model is able to reliably distinguish these EO types from the background and other objects. However, the model exhibited a higher false positive rate for the PFM-1 class (6%, row 0), indicating that some background elements were misclassified as PFM-1 landmines. The challenge in detecting PFM-1 in its natural environment can be attributed to its small size and varied coloration, as well as the fact that it is designed to blend in with its surroundings. On the other hand, the model infrequently overlooks EO, as evidenced by the elevated precision across all categories. In instances where a misidentification occurs, the model assigns the EO to a category that calls for scrutiny (e.g., the OZM-72 is occasionally categorized as the MON-50). This suggests that the model is generally able to identify the presence of an EO, even if it sometimes misclassifies the specific type.

Figure 4.6 illustrates examples of successful landmine replica detections by the model on images that were not included in the training dataset.

These results demonstrate the model's ability to generalize to new images and variations in EO appearance and background. The model successfully detected EO across different types, including the challenging PFM-1, with relatively high confidence scores (see 4.12).

Further analysis and discussion of these results, including an evaluation on real EO images and an assessment of the limitations of the current approach, are detailed in the following two sections.

4.12. Evaluation on Real Landmine Data

Having trained the YOLOv8 model on the dataset of 3D-printed landmine replicas (Section 4.10), this section presents the crucial evaluation of the model's performance on a dataset of real EO images. This assessment aims to determine the model's ability to generalize from the synthetic training data to real-world

(a) PFM-1 (gray) on foliage (b) PMN-2 partially obscured (c) MON-50 on tree

(d) OZM-72 on foliage (e) PMN on grass (f) PFM-1 on leaves

(g) MON-50 on snow (h) PMN-2 on snow (i) PFM-1 on snow and leaves

Figure 4.6: Examples of successful landmine replica detections by the model on images not included in the training dataset

scenarios and identify areas for further improvement. Specifically, this research analyzes the results from Experiment 11 (Table 4.5), which utilized the full two-stage augmentation strategy and the largest training dataset.

Figure 4.7 illustrates instances of the model identifying actual landmines, along with some cases of misclassification. The images were obtained from a dataset provided by professional deminers.

Subfigures *(a)*, *(b)*, *(c)*, *(d)*, *(e)*, *(f)*, and *(g)* present instances of correct identification, while subfigures *(h)* and *(i)* depict occurrences where EO were identified but misclassified. The model misclassified several landmines that were not represented in the training dataset, such as the POM-3 and POM-3 parachute cap. These misclassifications likely occurred because these landmines had visual similarities to those on which the model was trained (e.g., the POM-3 parachute cap being similar in shape to the PMN).

Table 4.6 presents the results of evaluating the model from Experiment 11 (Table 4.5) using data that is independent of the training set – namely, images of 3D-printed replicas and real EO.

Table 4.6

Results of Model Testing with New Data Types

Data Type	**Qty**	**Precision (%)**	**Recall (%)**	**mAP50 (%)**	**mAP50-95 (%)**
3D-Printed EO	402	98.4	98.6	99.1	85.1
Real EO	254	91.0	79.1	87.5	65.5

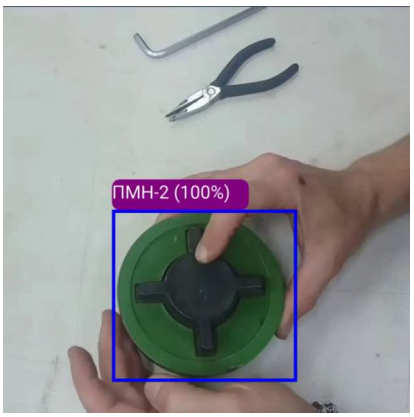
As shown in Table 4.6, the model achieved significantly higher performance on the dataset of 3D-printed landmines compared to the dataset of real ones. This difference is particularly noticeable in the recall and mAP50-95 metrics. The lower performance on real landmines suggests that there are still challenges in generalizing from synthetic data to real-world scenarios. This could be due to factors such as the increased variability and complexity of real-world images, the presence of unexpected objects or environmental conditions not captured

(a) Correctly identified PMN

(b) Correctly identified PMN-2

(c) Correctly identified MON-50

(d) Correctly identified PMN-4



(e) Correctly identified PMN-2

(f) Correctly identified MON-50

(g) Correctly identified OZM-72

(h) POM-3 incorrectly classified as OZM-72

(i) POM-3 parachute cap incorrectly classified as PMN

Figure 4.7: Illustrations of EO identification on photographs of actual landmines

in the synthetic dataset, and subtle differences in visual features between the 3D-printed replicas and actual landmines. The results indicate that the model developed utilizing 3D-printed replicas of landmines demonstrates promising outcomes, serving as a foundation for subsequent enhancements.

In order to implement this model in a practical application, it is necessary to utilize a larger dataset. In light of the observed discrepancy in performance between synthetic and real-world data, and in accordance with the recommendations of platforms such as Google Cloud, which recommend a minimum of 1000 images per class, it can be assumed that a dataset three times larger than the one used in this phase of the work may be required.

Table 4.7 provides a per-class breakdown of precision, recall, mAP50, and mAP50-95 for the real landmine dataset.

Table 4.7

Results of Testing the Model on Real Landmines

Landmine Type	**Precision (%)**	**Recall (%)**	**mAP50 (%)**	**mAP50-95 (%)**
PFM-1	78.0	84.2	89.1	61.2
MON-50	94.3	60.0	77.1	57.9
PMN-2	93.2	90.5	94.1	73.9
OZM-72	92.4	81.0	87.8	63.4
PMN	97.2	79.5	89.4	71.2
Mean	91.0	79.1	87.5	65.5

Figure 4.8 illustrates the normalized confusion matrix, associated with Table 4.7.

The confusion matrix in Figure 4.8 allows for the analysis of the detection outcomes for each type of EO. Class 0 represents PFM-1, class 2 represents PMN, class 14 represents MON-50, class 16 represents PMN-2, and class 19 represents OZM-72. The matrix highlights a significant number of misclassifications for the MON-50 (class 14), mistaking it for the background. This may be attributed to the visual dissimilarity of the 3D-printed MON-50 replica to

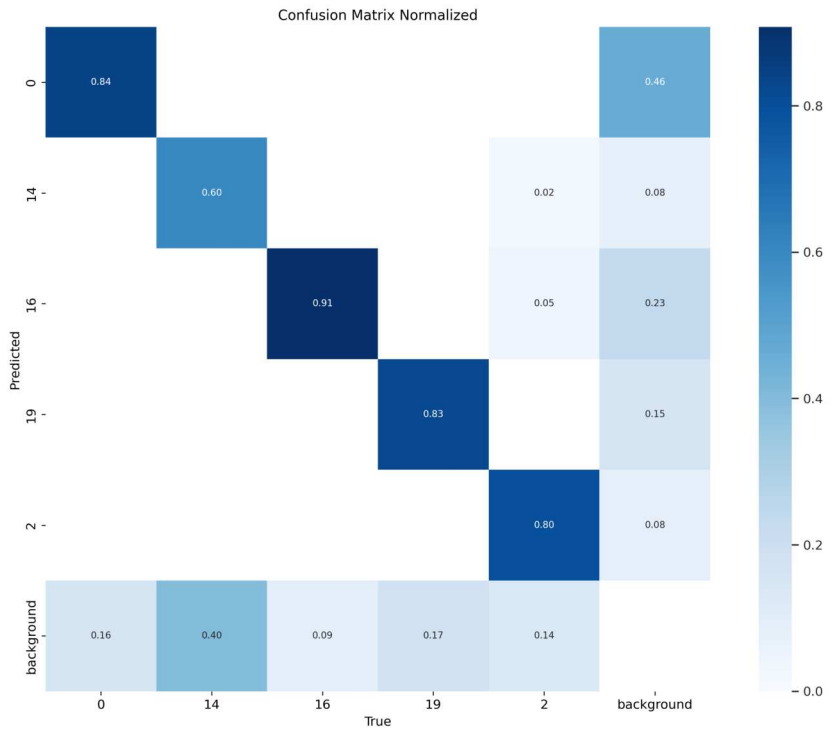


Figure 4.8: Normalized confusion matrix for the dataset from actual EO

an actual MON-50 landmine, potentially due to limitations in the 3D printing process or the materials used. To enhance the outcomes, one may incorporate additional photographs of the MON-50, create a more lifelike model of the landmine, produce this model at full scale, or experiment with various colors during the printing process.

The model generally demonstrates good precision across most EO types, particularly for PMN-2 (93.2%) and PMN (97.2%). However, the recall for MON-50 is notably lower (60.0%), indicating that the model frequently failed to detect this particular EO type in the real-world images. This lower recall for the MON-50 may be due to differences between the scaled-down 3D-printed replica (see Section 4.2) and real EO, including material properties, surface details, and the presence of occlusions and complex backgrounds in real-world images. It also underscores the need for a more comprehensive training dataset for this specific EO type.

4.13. Discussion

The use of 3D-printed EO replicas for creating synthetic training datasets has shown significant potential for EO detection models. The following subsections analyze the key findings, compare the results with existing research, discuss limitations, and outline directions for future work.

4.13.1. Summary of Findings. A dataset was created using 3D models of five prevalent types of anti-personnel landmines identified in Ukraine (Table 4.1). EO replicas were fabricated using a Prusa i3 MK3S+ 3D printer utilizing PLA plastic with a minimum infill of 5 – 10% (Figures 4.1 and 4.2). To guarantee data diversity, 1438 images were captured under various environmental conditions (clear, cloudy, rain, snow, etc.) and from multiple viewpoints (Figure 4.3). The acquired dataset comprises images of EO across various terrains, in diverse lighting conditions and visibility levels, facilitating the training of computer vision models on more realistic data.

The YOLOv8 model [73] was selected for EO recognition because of its superior accuracy and rapidity. The dataset of 3D-printed replicas of landmines was utilized to train the model. The learning process was conducted iteratively, involving incremental adjustments of the model's hyperparameters and the application of diverse data augmentation techniques, specifically those outlined in Tables 3.5 and 4.3. Consequently, a high precision of 98.0% and recall of 98.2% were attained on the test dataset (Table 4.5), validating the efficacy of employing 3D-printed models.

To further evaluate the model's performance, testing was performed on an independent validation dataset comprising 402 images of 3D manufactured landmines, where the model exhibited great performance (Figure 4.6, Table 4.6). This indicates that the model can generalize and effectively identify EO in novel images.

In the concluding phase of the research, the developed model was evaluated using an independent dataset comprising 254 images of actual EO (Figure 4.7). Table 4.6 indicates that the model attained an average precision of 91.0% and a recall of 79.1%. The results are due to the significant similarity between printed landmines and actual ones, together with the superior learning and generalization capabilities of the YOLO algorithm.

4.13.2. Comparison with Existing Research. A primary advantage of this research is the capacity to do studies in authentic conditions. In contrast to [41, 42], which performed studies at a sandy site containing buried EO, this study executed experiments across multiple locations (parks). Moreover, in contrast to [43], which employed the dataset from [42], this study utilized its own dataset. An further advantage is the diversity of landmines employed in this research, selected for their prevalence in the war in Ukraine, in contrast to [36], which utilized solely the PFM-1 type. Unlike [48], where data was transmitted to a computer for further processing to detect EO from a UAV, this study employs the YOLO framework for real-time object recognition. In contrast to research [41–43], which utilized limited or synthetic data, this study evaluated the model using actual EO, hence enabling an assessment of its practical applicability. The insufficiency of extensive data for analysis, along with the absence of comparative results on actual landmines, may result in the methodology being ineffective in practical applications.

This study illustrates that in situations of limited data, 3D printing can effectively augment the training and testing datasets for models intended to identify actual EO. Models developed on printed landmines can serve as a foundation for testing in actual conditions. In this scenario, it is essential to identify instances where the model misidentifies the object and include more data into the training process. Research such as [41–43] may attain superior outcomes by integrating 3D printing for dataset generation. The superior performance of

the models developed in this study is ascribed to the methodical construction of the dataset, the visual resemblance of the models to actual landmines, and the dependability of contemporary computer vision methods.

4.13.3. Limitations. Nonetheless, for certain EO types, such as the MON-50 and PMN, both precision and recall were inferior compared to others (Table 4.7). These outcomes may stem from several factors. The training dataset images were captured from a greater distance (1 – 1.6 meters) compared to the close-range images in the test dataset. Secondly, the 3D-printed replica of the MON-50 landmine failed to accurately replicate the shape and texture of the authentic EO, potentially impacting the precision of its identification.

The matrix in Figure 4.8 indicates that the low recall for MON-50 in Table 4.7 results from the failure to distinguish all landmines. Furthermore, for OZM-72 (class 19), the proportion of unidentified EO is marginally elevated at 17%. Nonetheless, the model exhibited a high degree of accuracy in detecting EO.

The current analysis highlights the necessity of minimizing the object's dimensions. The diminished dimensions of the MON-50 model may have led to its decreased recognition accuracy. A further potential limitation is the camera distance of 1 – 2 meters employed during data collecting. Modifying the model's design or investigating other computer vision techniques may be essential for proficient UAV picture processing, as evidenced in [125] for aircraft recognition. The analysis is constrained by the limited variety of EO kinds included in the dataset. While the predominant types of landmines employed in the Ukraine war were selected, the incorporation of a broader array of EO would have facilitated a more adaptable model. A further limitation is the lack of realism of many 3D representations of landmines, such as the OZM-72 and MON-50.

4.13.4. Future Research Directions. To enhance the model, one may explore contemporary feature extraction techniques, such as ORB [126] or

AKAZE [127], which exhibit more resilience to variations in illumination and perspective compared to traditional SIFT and SURF methods. Furthermore, the implementation of alternative contemporary neural network topologies may be contemplated to enhance the outcomes. EfficientDet [82] is recognized for its superior accuracy and computational efficiency, which is particularly crucial when implementing the model on mobile devices or resource-limited UAV. Transformer-based models (DETR [81], Deformable DETR [128]) exhibit exceptional accuracy on intricate datasets and proficiently handle high-resolution images. This may be advantageous for identifying EO in aerial or UAV imagery. These methodologies may enhance the model's capacity to differentiate between EO and background, particularly for categories with significant discrepancies, hence yielding increased detection accuracy and efficiency. Nonetheless, it is important to acknowledge that certain approaches may necessitate additional data for training or exhibit increased computing complexity. To enhance the precision of real EO detection, it is essential to augment the dataset and refine the realism of 3D models or incorporate additional variations. Exploring supplementary techniques, such data augmentation and hyperparameter optimization, is advisable to enhance the model's generalizability.

Future research will focus on integrating these findings into a unified, multi-platform algorithmic environment, as detailed in Chapter 6. In the course of this research, a mobile application for Android devices was evaluated. This application was designed to facilitate data collection and real-time EO detection. Additionally, a messenger bot interface was developed for user interaction and model deployment. The development of this multi-platform environment represents the next crucial step in translating this research into a practical, field-deployable solution for humanitarian demining.

4.14. Conclusions to Chapter 4

This chapter explored the potential of using 3D-printed landmine replicas to create synthetic datasets for training computer vision models in EO detection. The generated dataset, which consists of 1,438 photographs of 3D-printed replicas of five prevalent landmine types captured under diverse environmental conditions, has proven effective in training the YOLOv8 model. The model achieved a precision of 98.0% and a recall of 98.2% on the test portion of this synthetic dataset. These results validate the core premise of this study: that 3D-printed replicas can be utilized to generate varied and representative datasets for training computer vision algorithms, offering a safe and cost-effective alternative for both research and practical applications. This approach is particularly valuable in the initial training of models and for augmenting datasets that include real landmine images.

The YOLOv8 model, trained on this dataset and employing the two-stage augmentation strategy detailed in Chapter 3, demonstrated the significant impact of specific augmentation techniques. Notably, the combination of augmentations listed in Table 4.5 (Experiment 11) yielded the best results. The analysis of the confusion matrix (Figure 4.5) highlighted areas for potential model enhancement, particularly in differentiating EO from background elements.

Further evaluation was conducted using an independent dataset of 254 photographs of actual landmines, provided by professional deminers. On this dataset, the model attained an average precision of 91.0% and a recall of 79.1% (Table 4.6). While these results are promising, they also reveal a performance gap compared to the synthetic dataset. This gap underscores the challenges of generalizing from synthetic to real-world data and highlights the need for further refinement of the 3D printing and data generation process.

Moreover, the per-class performance analysis (Table 4.7) showed substantial variation in precision and recall across different EO types. For instance, the

precision for PFM-1 detection was 78.0%, while for PMN it reached 97.2%. Similarly, recall ranged from 60.0% for MON-50 to 90.5% for PMN-2. These discrepancies can be attributed to several factors, including the limitations of the 3D-printed replicas in accurately capturing the intricate details of certain landmines (e.g. MON-50), the reduced scale of some replicas, and the variations in image capture distances between the training and testing datasets.

The limited size of the real landmine image dataset is another constraint that should be addressed in future research. Additionally, the confusion matrix analysis (Figure 4.8) revealed a tendency for the model to misclassify certain background elements as EO, particularly for the PFM-1 class. This indicates a need for further investigation into the features that the model is using for classification and for techniques to improve its ability to distinguish between EO and visually similar background objects.

CHAPTER 5
DEVELOPMENT AND DEPLOYMENT OF A CLOUD-BASED EXPLOSIVE OBJECTS DETECTION SERVICE WITH MESSENGER BOT INTEGRATION

5.1. Introduction: A Cloud-Based Explosive Objects Detection Service for Messenger Platforms

Building upon the methodologies for data augmentation and model training established in Chapters 3 and 4, this chapter details the design and implementation of a cloud-based EO detection service accessible through a user-friendly messenger bot interface. This system leverages the cross-platform availability of messenger applications and the scalability of cloud computing to provide a practical and efficient solution for EO identification.

The proposed approach enables users to upload images of suspicious objects to a messenger bot for evaluation, utilizing advanced ML algorithms to deliver real-time assessments of EO probability. By combining the power of AI, cloud computing, and messenger platforms, this project seeks to empower individuals, particularly those in affected regions, to swiftly and accurately identify potential explosive risks. Ultimately, this work contributes to supporting humanitarian demining initiatives, improving public safety, and aiding in the long-term recovery of post-conflict areas like Ukraine.

The key features of this system include:

— A cloud-based architecture utilizing GCP [122] for scalability and reliability.

— A messenger bot interface for seamless interaction with the service.

— Integration with Google Gemini to provide users with supplementary

information regarding identified EO.

The following sections will elaborate on the system's architecture, implementation details, and evaluation, demonstrating its potential as a valuable tool in the ongoing effort to address the global EO challenge.

5.2. Related Work

Currently, there is a scarcity of readily available software solutions capable of providing real-time EO detection through a simple, user-friendly interface like a messenger bot. Moreover, developing, designing, testing, and managing such a user-focused solution requires significant resources. While the ultimate goal is to integrate advanced detection models directly into mobile devices for near-the-ground operation, leveraging existing, widely-used messenger platforms represents a crucial intermediate step. This approach allows for rapid deployment of the models, minimizing development overhead and maximizing accessibility through familiar interfaces already present on most smartphones and computers.

Utilizing existing messaging platforms with bot functionality conserves development resources that can be focused on enhancing the core recognition algorithms. It also allows the dissemination of the model to a broader audience. This is particularly valuable for rapid identification purposes; when users encounter a suspicious object in real life or online, they can submit an image via a messenger bot to receive a prompt assessment. While dedicated mobile applications offer potential advantages for specialized tasks, messenger bots provide immediate, cross-platform accessibility without requiring users to download and install separate applications.

The service utilizes the models developed in previous chapters, which explored the creation of ML models for EO detection, incorporating data augmentation and 3D printing to address data scarcity (Chapters 3 and 4). Although

these efforts demonstrated the potential of AI for EO detection, a significant barrier persisted: how to effectively deploy these advanced models to those who need them most in the field, such as demining teams, humanitarian organizations, and civilians in affected areas. This necessitates a solution that is readily accessible, intuitive, and adaptable to various locations and user requirements.

Several mobile applications exist for mine action, such as those developed by the UNMAS [50] and the Collective Awareness to Unexploded Ordnance (CAT-UXO) [129] application. However, these applications primarily focus on EO awareness, risk education, and reporting of suspected hazards. They lack the capability for automated, real-time EO recognition directly from images. This gap presents a substantial opportunity to leverage the ubiquity of smartphones and the capabilities of cloud computing to establish a more proactive and effective method for EO identification.

Existing solutions for EO detection often suffer from limitations in accessibility, usability, and real-time performance. In this chapter, these limitations are addressed by proposing a cloud-based service that leverages the widespread use of messenger platforms, providing a user-friendly, efficient, and readily deployable tool for EO detection. While dedicated mobile app development remains a long-term goal, the messenger bot approach offers a practical and readily available solution for bridging the gap between advanced detection models and end-users in the field. The following sections detail the design and implementation of this system.

5.3. Goals and Objectives

The primary goal of this project is to address the pressing humanitarian issue of EO detection through the development and evaluation of a novel cloud-based service that leverages the accessibility of messenger platforms and the power of AI. This research aims to provide a near-real-time, user-friendly, and

readily deployable method for EO detection by utilizing the scalability of cloud computing and the functionalities of commonly available smartphones. The proposed system integrates AI, cloud computing, and a messenger bot interface to offer a more accessible, efficient, and user-friendly solution compared to traditional, resource-intensive demining methods.

To achieve this goal, the following specific objectives are defined:

1. **Cloud-Based System Design and Implementation:** Design and implement a cloud-based architecture for an EO detection service, utilizing GCP services, including Cloud Functions for image processing and a pre-trained model, hosted on Roboflow, for object detection.
2. **Messenger Bot Interface Development:** Develop and implement a user-friendly messenger bot interface that allows users to submit images of suspected EO for analysis and receive detection results along with relevant safety information.
3. **Google Gemini Integration:** Integrate the system with Google Gemini to provide users with contextual information about identified EO, such as their type, origin, characteristics, and safety precautions.

5.4. Architecture of the Cloud-Based Explosive Objects Detection System

This section details the architecture and design principles of the cloud-based EO detection service. The system utilizes Google Cloud Functions to implement ML models and process images efficiently. A messenger bot serves as the user interface, facilitating interaction with the service. Users submit photographs of suspicious objects to the bot, which subsequently transmits them to the cloud for evaluation. The results, including the probability of an EO and its likely type, are transmitted to the user via the bot. This architecture facilitates seamless integration with multiple messaging platforms and ensures

adaptability for future modifications, such as incorporating new EO types or integrating with different sensor modalities.

The system features a modular architecture, consisting of three primary components, as illustrated in Figure 5.1:

— **Messenger Bot:** This element functions as the user interface, enabling users to engage with the service via a messenger application. The messenger bot interface was chosen for its widespread accessibility, ease of use, and familiarity to a large user base. It acquires images from users, transmits them to the cloud for processing, and delivers the analysis findings. While Telegram was used for initial development and testing, the modular design allows for adaptation to other messenger platforms.

— **Cloud Functions:** These functions, deployed on the GCP [122], manage image processing and analysis. The Cloud Functions are triggered upon receiving an image from the messenger bot via a webhook. They obtain images from the messaging bot and activate the EO identification module. These functions handle image format verification, resizing, and any necessary preprocessing steps before invoking the EO recognition module. The utilization of serverless Cloud Functions enables efficient resource allocation, automatically scaling based on demand and minimizing operational costs.

— **EO Recognition Module:** The EO Recognition Module utilizes a pre-trained model [73] to detect and classify EO in the images. The inference process is managed by the Roboflow service [93], which hosts the model and provides an API for efficient model invocation. The model was fine-tuned on a diverse dataset of EO images, including both real-world images and synthetic images generated through data augmentation and 3D-printed replicas, as described in Chapters 3 and 4.

As illustrated in Figure 5.1, the system's modular design allows for inde-

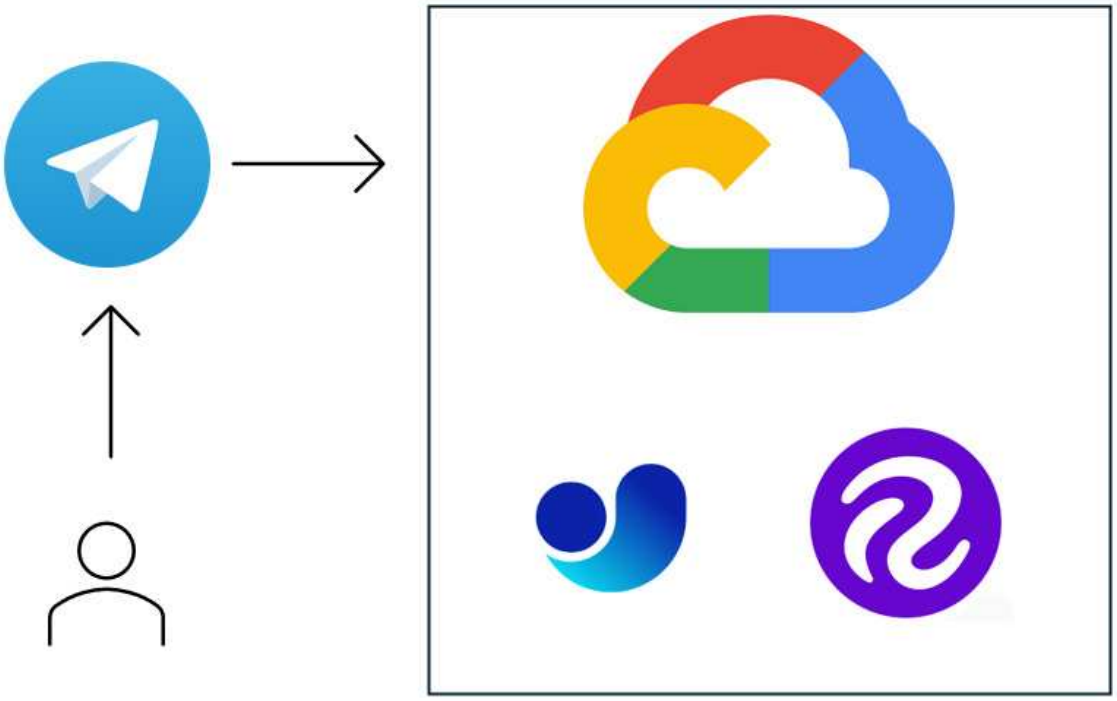


Figure 5.1: System architecture of the cloud-based EO detection service

pendent development and optimization of each component, ensuring flexibility and scalability. The use of cloud functions ensures optimal resource usage and rapid response times, while the messenger bot interface offers a user-friendly and accessible means of engaging with the service. This architecture combines the accessibility of messenger platforms with the power of cloud computing and the accuracy of a state-of-the-art object detection model to create a robust and user-friendly EO detection service. The following section provides further details on the implementation of each component.

5.5. Implementation Details

This section provides a detailed description of the technical implementation of the cloud-based EO detection service.

5.5.1. Messenger Bot Implementation. The messenger bot serves as the primary interface for user interaction, developed using Python and the 'python-telegram-bot' framework [95]. This framework was selected for its ease of use, comprehensive documentation, active community support, and asynchronous architecture, which enables the bot to handle multiple user requests concurrently. The bot is designed to handle image uploads from users, forward these images to the Google Cloud Functions via a webhook, and deliver the detection results back to the user in a clear and informative manner.

The bot utilizes a webhook to establish a communication channel with the Google Cloud Functions. When a user uploads an image, the Telegram API sends a notification to the designated webhook URL, triggering the corresponding Cloud Function for image processing. The webhook URL is secured to prevent unauthorized access.

Upon receiving an image message, the bot extracts the file ID from the Telegram API's response. The following Python code snippet illustrates how the bot downloads the image and uploads it to Google Cloud Storage:

Listing 5.1: Downloading and uploading an image via the Telegram bot

```
    # Create a blob in a GCS bucket
    blob = bucket.blob(filename)

    # Download the file content into a BytesIO object
    file_stream = BytesIO()
    try:
        await new_file.download_to_memory(out=file_stream)
    except TelegramError as e:
        # Handle Telegram API errors (e.g., file not found, download
            failed)
        print(f"Error downloading file: {e}")
        # Send an error message to the user
        return
    file_stream.seek(0) # Reset the stream position to the beginning

    # Upload the file content from the BytesIO object
```

```
    try:
        blob.upload_from_file(file_stream)
    except GoogleCloudError as e:
        # Handle Google Cloud Storage errors (e.g., bucket not found,
            upload failed)
        print(f"Error uploading file to GCS: {e}")
        # Send an error message to the user
        return
```

This code demonstrates the use of the 'BytesIO' object for in-memory file handling, allowing the bot to efficiently download the image from Telegram's servers and then upload it to the designated Google Cloud Storage bucket. The 'try-except' blocks ensure that potential errors during the download or upload process are handled gracefully, preventing crashes and providing informative error messages to the developer and potentially the user. The bot then proceeds to initiate the EO detection process on the uploaded image.

5.5.2. Cloud Function Implementation. The backend processing and analysis are handled by Google Cloud Functions, written in Python and deployed using the Google Cloud Functions framework [122]. Upon receiving an image message from the user, the Telegram bot sends a request to a designated Cloud Function via a webhook. This triggers the execution of the Cloud Function, which performs the following steps:

1. **Image Validation and Preprocessing**: The function first validates the image format and performs necessary preprocessing. This includes resizing the image to the required input dimensions of the model (e.g., 640x640 pixels) and normalizing pixel values.
2. **EO Detection Invocation**: The preprocessed image is then passed as input to the EO Recognition Module. This module utilizes a two-stage object detection approach, potentially employing different models or configurations for increased accuracy and robustness.

The core of the EO detection process involves invoking the Roboflow API

to perform inference using pre-trained models. As illustrated in Listing 5.3, the Cloud Function sends a request to the Roboflow inference endpoint, specifying the filename and the ID of the model to be used. The code first attempts inference with ‘model_x‘. If no objects are detected, it proceeds to use ‘model_y‘, providing a fallback mechanism. Error handling is incorporated to manage potential issues during the inference process. The results are then processed to extract information about the detected objects.

Listing 5.2: Running inference calling Roboflow API using 2 models

```
# Call first runRoboflowInference Cloud Function
inference_url = INFERENCE_URL
params = {'filename': f'{new_file.file_id}.jpg', 'model_id' : '
    model_x'}
response = requests.get(inference_url, params=params)

objects_found = False

if response.status_code != 200:
 await bot.sendMessage(chat_id=chat_id,
      text=f"Error running inference: {response.text}")
 return

print(f"Photo processing model ...")
objects_found = await process_inference_responce(objects_list,
    response, bot, filename, chat_id, user_language)

if not objects_found:
 params = {'filename': f'{new_file.file_id}.jpg', 'model_id' : '
     model_y'}
 response = requests.get(inference_url, params=params)

 if response.status_code != 200:
  await bot.sendMessage(chat_id=chat_id,
       text=f"Error running inference: {response.text}")
  return

 print(f"Photo processing model...")
 objects_found = await process_inference_responce(objects_list,
```

```
        response, bot, filename, chat_id, user_language)
```

The ‘await process_inference_responce‘ function, called after each inference attempt, handles parsing the JSON response from Roboflow, extracting bounding box coordinates, confidence scores, and class labels of detected objects. Subsequently, this information is used to generate alerts or visualizations for the user, indicating the presence and location of potential EO. This demonstrates a robust approach to EO detection, leveraging cloud infrastructure and external API for efficient and scalable processing.

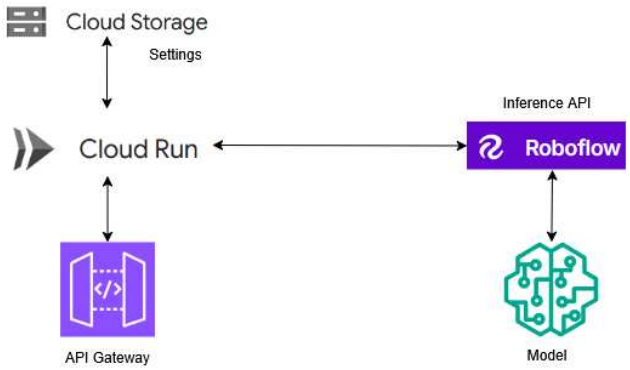


Figure 5.2: Google Cloud architecture with a gateway to Google Cloud Functions (Cloud Run) as an initial entry point, Google Storage for settings, and Roboflow API which hosts models

Picture 5.2 depicts an architecture of GCP. Google Cloud Run takes the main part of orchestration, but the model which is hosted on Roboflow processes images and identifies EO on them.

5.5.3. The Explosive Objects Recognition Module Integration. The EO Recognition Module utilizes a pre-trained YOLOv8 model [73] to detect and classify EO in the submitted images. This model was trained on the diverse dataset described in chapters 3 and 4, which includes real-world EO images, images of 3D-printed landmine replicas, and synthetic images generated through data augmentation techniques.

The trained YOLOv8 model is hosted on the Roboflow platform [93], leveraging its efficient inference API and model versioning capabilities. The Cloud Functions communicate with the Roboflow API to send the preprocessed images for inference and receive the detection results. These results include bounding box coordinates, class labels, and confidence scores for each detected EO, where the confidence scores might determine if an alert is generated or not.

The EO Recognition Module, hosted on Roboflow, returns a JSON response containing the detection results. The Cloud Function then processes this response to extract the bounding box coordinates, class labels, and confidence scores for each detected object. To visually represent these detections, the Cloud Function draws bounding boxes around the detected EO on a copy of the original, preprocessed image. An example of this process is shown in the following Python code snippet:

Listing 5.3: Drawing bounding boxes on the detected objects

```
    for prediction in predictions['predictions']:
        isPredicted = True
        classname = prediction['class']
        confidence = prediction['confidence']
        # Find the object name based on class_id
        print(f"object_name: {classname}")

        x, y, w, h = prediction['x'], prediction['y'], prediction['width
            '], prediction['height']

        # Calculate corner points based on Roboflow's format
        x1 = x - (w / 2)
        y1 = y - (h / 2)
        x2 = x + (w / 2)
        y2 = y + (h / 2)

        draw.rectangle([(x1, y1), (x2, y2)], outline="red", width=11)
```

This code snippet iterates through the predictions, extracts the bounding box coordinates (x, y, width, height), calculates the corner points, and draws a

red rectangle on the image. The resulting image, with bounding boxes overlaid, is then sent back to the user via the Telegram bot.

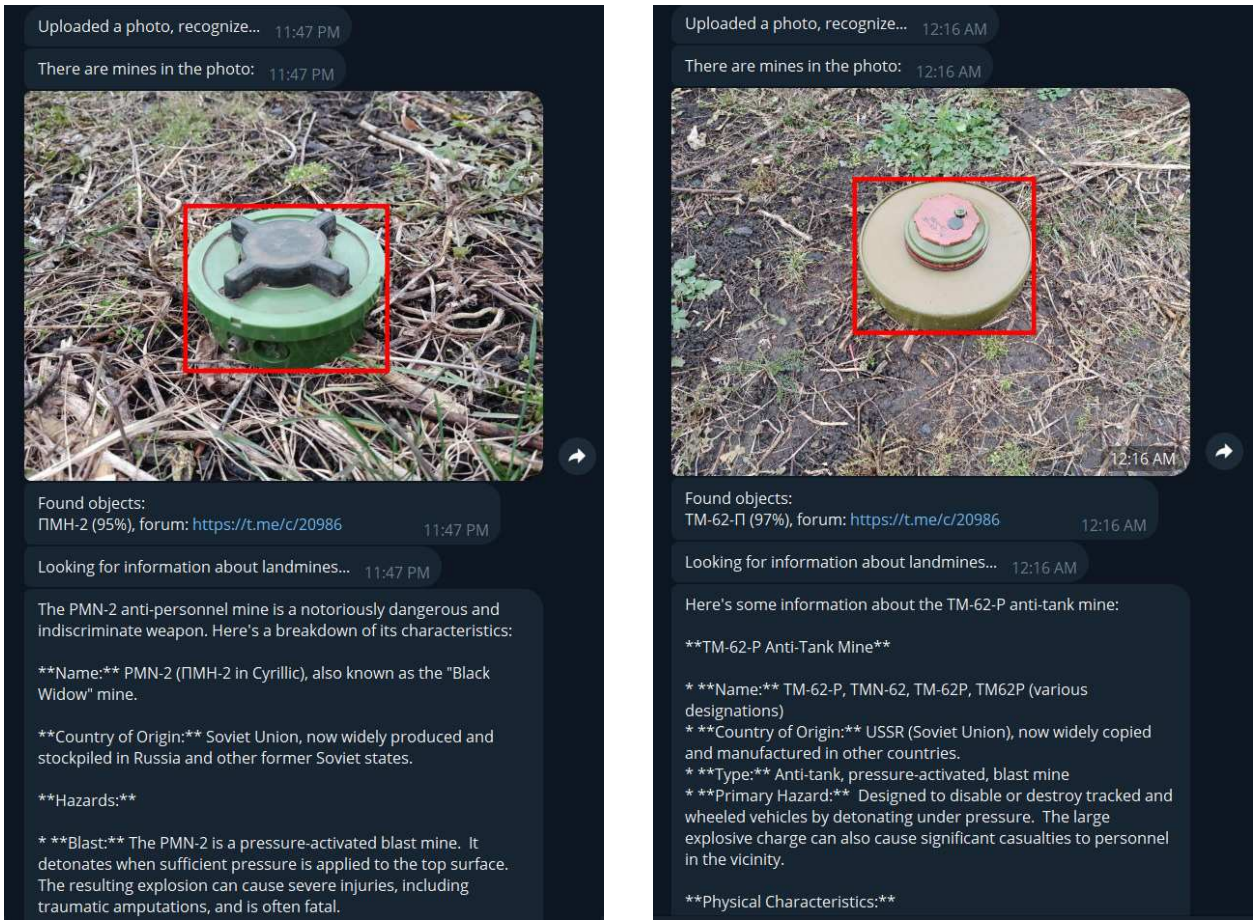
5.6. Integration with Google Gemini and Additional Bot Features

To enhance the informational value and practical utility of the EO detection service, the system incorporates an integration with Google Gemini. This integration aims to provide users with supplementary information about identified EO, thereby increasing situational awareness and promoting safer decision-making.

5.6.1. Collaboration with Google Gemini. Upon the identification of a potential EO by the recognition module (Section 6.5), the bot automatically generates a query to Google Gemini, specifying the detected EO type. Google Gemini then provides a concise summary of relevant facts about the EO. This summary may include:

- **Country of Origin:** The country where the EO was manufactured.
- **Primary Attributes:** Key characteristics such as the EO's type, weight, and dimensions.
- **Usage Methodology:** Information on how the EO is typically deployed and its activation mechanism.
- **Additional Information:** Relevant details such as common variations, safety precautions, and links to external resources for further reading.

Figure 5.3 illustrates an example of a Google Gemini response for a detected EO. This integration significantly enhances the user's understanding of the potential threat, facilitating more informed decision-making in the field.



(a) (b)

Figure 5.3: Screenshots of the messenger bot interface. The source of the images: Forester

5.6.2. Target Audience and Deployment Context. Presently, the messenger bot is tailored for interaction with demining professionals. To facilitate this specialized use case, the bot is integrated with a dedicated forum for deminers. Upon detecting an EO, the bot disseminates information about it and provides a link to the relevant discussion thread on the forum. This enables deminers to swiftly share information, synchronize their efforts, and collaboratively formulate plans.

However, the bot's functionality could be expanded to serve a broader audience, including civilians living in explosive-affected areas and international organizations involved in humanitarian demining. In such scenarios, the bot could

provide links to relevant articles on EO safety, contact information for emergency services and demining organizations, and guidelines on how to respond to a suspicious object. This approach would enhance public understanding of EO risks and promote safer conduct in the presence of potential hazards.

5.6.3. Language Support and Additional Features. To enhance accessibility, the bot supports two interface languages: Ukrainian and English. The interface language is automatically determined based on the user's language settings in their messenger application.

Beyond its primary function of EO detection, the bot offers a range of ancillary features to further improve its utility:

— **EO Addition Request:** It is possible for users to submit requests for the incorporation of new classes into the model recognition capabilities. This subject will be discussed in greater detail in Chapter 7.

— **Recognized EO List:** Users can access a list of EO types currently recognized by the model.

— **Feedback Submission:** Users can provide feedback on the bot's performance, helping to identify areas for improvement and refinement.

These additional features contribute to the bot's flexibility and usefulness as a tool for both professional deminers and potentially the wider public. They also provide valuable mechanisms for continuous improvement and adaptation of the system based on user input and evolving needs.

5.7. Conclusions to Chapter 5

This chapter has presented the design, implementation, and evaluation of a novel cloud-based EO detection service accessible through a user-friendly messenger bot. The developed system demonstrates the significant potential of leveraging readily available technologies, such as advanced machine learning models, cloud computing platforms, and familiar messenger interfaces, to

address the urgent global challenge of EO detection.

The integration of the YOLOv8 model, hosted on Roboflow, with the scalable infrastructure of GCP ensures high identification accuracy and rapid data processing. The choice of Telegram as the initial messenger platform, while adaptable to other platforms, provides immediate access to a vast user base. Furthermore, the integration with Google Gemini enriches the user experience by providing crucial contextual information about identified EO, enhancing situational awareness and promoting safer decision-making.

The project is under active development, with ongoing efforts focused on continuously refining the bot and enhancing the accuracy and robustness of the EO recognition models. A key aspect of this iterative process involves analyzing detection errors, identifying challenging scenarios, and incorporating relevant images into the model's training dataset. This iterative approach allows the system to "learn from its mistakes" and progressively improve its detection capabilities over time.

Future development will prioritize expanding the bot's ability to recognize a wider range of EO types and integrating it with other prevalent messenger platforms. Furthermore, the aim is to incorporate the insights and methodologies from previous chapters, particularly the use of advanced data augmentation techniques (Chapter 3), and 3D-printed replicas (Chapter 4), to further enhance the training dataset and improve the model's ability to generalize to diverse real-world scenarios.

CHAPTER 6
DESIGN, IMPLEMENTATION, AND EVALUATION OF A CROSS-PLATFORM APPLICATION FOR EXPLOSIVE OBJECTS DETECTION

6.1. Introduction

The widespread use of anti-personnel landmines in conflict zones like Ukraine, where an estimated 128,000 square kilometers of land and 13,000 square kilometers of water are contaminated [58], poses a significant threat to human life and hinders post-conflict recovery. Building upon the foundation laid in previous chapters, which focused on generating a robust training dataset using 3D-printed landmine replicas and advanced data augmentation techniques (Chapters 3 and 4), and resulted in trained YOLO models with high accuracy on synthetic data, this chapter details the development and evaluation of a cross-platform application for real-time EO detection, a key component of the unified algorithmic environment. This application is designed to bridge the gap between advanced AI-powered detection models and end-users in the field by providing a readily deployable and user-friendly interface.

The cross-platform application, with a mobile build specifically evaluated and tested by professional deminers, supports near-the-ground operation, facilitates user-friendly interaction, and enables the collection of valuable data for ongoing model refinement. The application is designed for both online and offline operation. When online, it integrates with the GCP API using HTTP calls for EO detection, leveraging the same backend infrastructure as the messenger bot (Chapter 5). In offline mode, it utilizes an on-device version of the YOLO model for local processing.

This chapter focuses on the development of the standalone, cross-platform application, written in C++ with the Qt framework. Subsequent sections detail the application's architecture (Section 6.5), user interface design (Section 6.8), and core functionalities (including image acquisition, online and offline object detection, and data transmission) (Section 6.9). Results and discussion are provided in Section 6.10.

6.2. Objectives of the Cross-Platform Application

Within the scope of the unified algorithmic environment for EO detection, this chapter focuses on the development and evaluation of a cross-platform application designed for deployment on user devices, particularly mobile platforms. The application is designed to operate in both online and offline modes, providing flexibility depending on network availability. The online mode leverages real-time processing via the GCP API detailed in Chapter 5, while the offline mode utilizes an optimized, on-device version of the YOLO model for local processing.

The primary objectives of this cross-platform application are:

1. **Accurate and Efficient EO Detection:** To provide users with a reliable tool for identifying EO in both online and offline modes, achieving high precision and recall while minimizing processing time.
2. **On-Device (Offline) Processing:** To enable EO detection in areas with limited or no internet connectivity, utilizing a streamlined, on-device YOLO model.
3. **Cloud-Based (Online) Processing:** To leverage the scalability and computational power of the GCP for enhanced detection accuracy and access to the latest model updates when an internet connection is available.
4. **User-Friendly Interface:** To provide an intuitive and easy-to-use in-

terface for image acquisition, object detection, and result visualization.

5. **Semi-Automatic Data Annotation:** To facilitate the continuous improvement of the detection model by enabling users to contribute to the dataset through a semi-automatic annotation process.
6. **Data Transmission and Model Refinement:** To establish a secure mechanism for transmitting user-annotated data to a central server for model retraining and refinement.
7. **Adaptability and Extensibility:** To design the application with a modular architecture that allows for future expansion, including the addition of new EO types and integration with other sensors.

The widespread adoption of smartphones and other mobile devices makes them an ideal platform for deploying this application, ensuring broad accessibility. The development of a user-friendly, cross-platform application for mobile EO recognition will streamline and expedite the identification process. Furthermore, there are plans to expand the functionality of the app to include additional features, such as the notification of users regarding the presence of explosive devices, the transmission of their coordinates for the purpose of reporting and mapping, and other related operations.

To achieve these objectives, the following tasks were identified as essential:

1. **Implement Online and Offline Detection Capabilities:** Develop and integrate both online and offline EO detection functionalities within the cross-platform application. This includes:
   — **Online Mode:** Integrating the application with the GCP API for real-time processing using the cloud-hosted YOLO model.
   — **Offline Mode:** Deploying a compressed and optimized version of the YOLO model directly onto the target device (e.g., Android smartphone) for on-device detection without requiring internet connectivity. The efficacy of this offline mode will be assessed through a performance test that evaluates both recall and detection speed.

2. **Enable User Correction and Annotation:** Implement a user-friendly interface within the application to allow users to:
   — Manually annotate objects within images (add bounding boxes).
   — Correct model predictions (adjust bounding box size/position, change object class labels).
   — Select the correct EO type from a predefined list.
   — Delete incorrect detections (false positives).
3. **Facilitate Secure Data Transmission and Model Refinement:** Implement a secure and efficient mechanism for transmitting user-corrected data and operational feedback (including application logs and potentially user-annotated images) from the application to a central server. This data will be used for ongoing analysis, model retraining, and system improvement.

6.3. The Subject and Primary Hypotheses During the Application Development

The subject of this part of research is an information system designed to identify explosive objects in images, specifically focusing on anti-personnel landmines, anti-tank landmines, and shells. This system is manifested as a cross-platform application, with a particular emphasis on its mobile deployment, to facilitate near-the-ground, real-time detection.

The primary hypotheses guiding developement of mobile application are:

Hypothesis 6.1. A cross-platform application, utilizing a deep learning model, can accurately identify the presence of explosive devices in images captured by user devices.

Hypothesis 6.2. Errors committed by the model, such as misidentifications or missed detections, can be utilized for the continuous refinement of the model. These errors are then transmitted from the application to a server for further

training.

This research is predicated on the following assumptions:

— Images captured by users for application verification will generally be of sufficient quality to facilitate accurate recognition by the model.

— Users will be motivated to obtain accurate recognition results and will therefore strive to capture photographs of suitable quality and from appropriate perspectives.

— During the testing phase, the model will be evaluated using images specifically chosen to assess its robustness against a range of environmental conditions, object orientations, and potential false alarm triggers.

For clarity, the following simplifications have been made at this stage of the study:

— The model may not be capable of recognizing explosive objects that were not included in the training dataset. In such cases, the application provides a mechanism for transmitting these unidentified object images and user feedback to the server for future model updates and expansion of the recognized object types.

— Recognition accuracy may be reduced for images of low clarity (e.g., blurry, poorly lit, or obstructed images) or those captured from unconventional angles not well-represented in the training dataset.

— It is hypothesized that a relatively small number of images (approximately 50 – 100) of a new object type, when properly annotated, is sufficient for incorporating that object category into the model's recognition capabilities. After that, the semi-automatic annotation process (described in Section 6.7.1) will be applied iteratively to further improve the model's ability to detect the object.

6.4. Technical Specifications: Hardware and Software Environment

This section details the hardware and software specifications used in the development and testing of the cross-platform EO detection application.

6.4.1. Hardware.

6.4.1.1. Mobile Devices. To ensure the application's broad compatibility and assess its performance across a range of hardware configurations, the study employed a variety of Android devices, spanning Android versions 6 to 14. The primary test models were the Samsung Galaxy A53, powered by an octa-core processor (two Cortex-A78 cores at 2.4 GHz and six Cortex-A55 cores at 2.0 GHz), with 128 GB of storage, 6 GB of RAM, and a 64-megapixel camera; and the Realme 6, featuring an octa-core processor (2x2.05 GHz Cortex-A76 & 6x2.0 GHz Cortex-A55), 128 GB of storage, 8 GB of RAM, and a 64-megapixel camera. These models were chosen as representative examples of mid-range Android devices.

Furthermore, supplementary testing was performed on a diverse set of devices, including:

— Redmi Note 9 Pro
— Redmi Note 11 Pro
— Redmi Note 13 Pro
— Google Pixel 4a 5G
— Samsung Galaxy A23
— Samsung Galaxy Tab S7 FE
— Samsung Galaxy M11
— Xiaomi Mi Play
— Xiaomi 11 Pro
— Xiaomi 12 Pro

— Motorola One Zoom

— IIIF150 B2

6.4.1.2. Cloud Platform. Google Cloud Functions were utilized to handle requests to GCP. The Cloud Functions were configured with a range of RAM capacities, from 128 megabytes (MB) to 256 MB, and the CPU allocation ranged from 1/6 of a virtual processor to 4 virtual processors. The execution timeout was set to 60 seconds. GCP offers horizontal scalability, enabling the automatic allocation of supplementary resources to address surges in request volumes, thereby ensuring responsiveness even under substantial loads. The instance types used were of the 1st generation.

6.4.2. Software Architecture and Technologies.

6.4.2.1. Development Environment. The cross-platform application was developed using the Qt framework (version 6.7.2) [130] within the Qt Creator 13.0.2 IDE (Figure 6.1). The Qt framework was selected for its robust cross-platform capabilities, enabling deployment on various operating systems (including Android, iOS, Windows, macOS, and Linux) from a single codebase. Its comprehensive set of libraries and modules, including Qt Core, Qt Quick, QML, Qt Network, and Qt SQL, along with its efficient performance and support for C++, made it well-suited for this project. Qt Creator 13.0.2 provided an integrated and user-friendly environment for developing, debugging, and deploying the application. Its built-in tools for GUI design (using QML), code editing (Qt Creator), version control (Git), and project management streamlined the development workflow. The application was compiled for Android using the Qt 6.7.2 Clang compilers for arm64-v8a and armeabi-v7a architectures, ensuring compatibility with a wide range of Android devices (with Android versions 6 through 14).

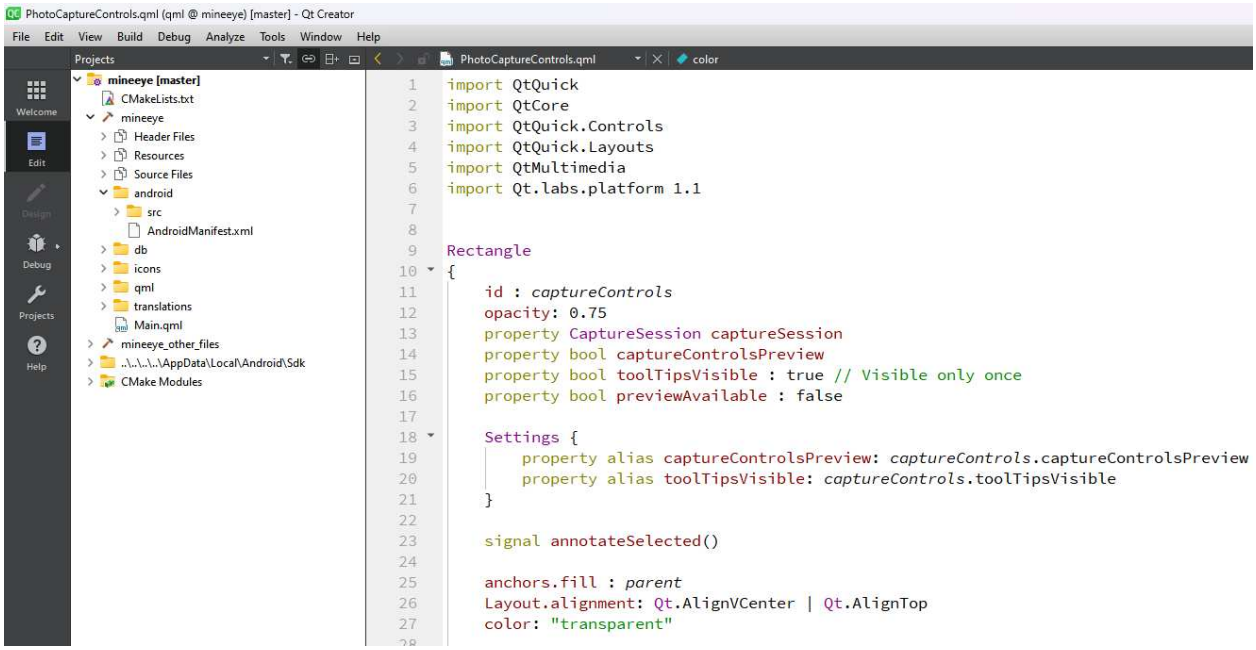


Figure 6.1: Screenshot of the Qt Creator 13.0.2 IDE, showing the *mineeye* project open, with the editor displaying the code for QML and the left panel showing the project structure

6.4.2.2. Cross-Platform Framework and Implementation Details. The application leverages the Qt framework's support for cross-platform development. The user interface was implemented using QML, a declarative language that allows for the creation of dynamic, visually rich, and fluid user interfaces. QML's JavaScript integration facilitated the implementation of UI logic and event handling. Its ability to create resolution-independent UIs was particularly beneficial for ensuring a consistent user experience across different screen sizes and device types.

The application's core logic, including image processing, communication with the cloud backend, and on-device model invocation, is implemented in C++, leveraging its performance and object-oriented features. This C++ backend seamlessly integrates with the QML-based user interface through the Qt framework's signal and slot mechanism, enabling efficient communication and data exchange between the UI and the underlying logic.

The following code snippets illustrate how signals and slots are used to connect QML elements to C++ functions:

Listing 6.1: Connecting QML signals to C++ slots

```
// Connecting a signal emitted after fetching settings to a slot that sets them
QObject::connect(&dataController, &DataController::settingsFetched,
                 &settingsManager, [&](const QJsonObject& newSettingsJson, bool overrideAll) {
                     settingsManager.setSettingsFromJson(newSettingsJson, overrideAll);
                 });

// Connecting a signal from a QML item to a C++ controller slot
QObject::connect(item, SIGNAL(qmlSendAllImagesMenuItemSignal(bool)),
                 &controller, SLOT(sendUpdatedFullImagesDeferWrapper(bool)));
```

In the first example, the ‘settingsFetched‘ signal from the ‘DataController‘ object is connected to a lambda function that calls the ‘setSettingsFromJson‘ method of the ‘SettingsManager‘ object. This allows the application to update its settings when new data is fetched from the server. The second example demonstrates connecting a signal (‘qmlSendAllImagesMenuItemSignal‘) emitted by a QML item (e.g., a button) to a slot (‘sendUpdatedFullImagesDeferWrapper‘) in a C++ controller object. This enables user interactions in the QML UI to trigger actions in the C++ backend.

The Qt Concurrent module was employed to manage multithreading, enabling computationally intensive tasks, such as image processing and model inference, to be performed in the background without blocking the user interface, ensuring a smooth and responsive user experience.

6.4.2.3. Cloud Functions. The backend processing, which encompasses user registration, referral link generation, and online EO detection, is managed by GCP, which are written in Python and deployed using the Google Cloud

Functions framework [122]. These functions are triggered via HTTP requests from the application, and they perform tasks such as image validation, preprocessing, and communication with the GCP API for object detection. The results, which include bounding boxes and class labels, are subsequently formatted and returned to the user. For a more detailed description of the Cloud Functions' implementation, refer to Section 5.5.2.

One of the primary Cloud Functions handles user registration and referral link generation. This function, illustrated in Listing 6.2, is responsible for creating new user accounts and generating unique referral links for each user.

Listing 6.2: GCP cloud function code snippet for user handling

```
def check_gcs_file_exists(bucket_name, file_path):
    """Check if a file exists in Google Cloud Storage."""
    storage_client = storage.Client()
    bucket = storage_client.get_bucket(bucket_name)
    blob = bucket.blob(file_path)
    return blob.exists()

def generate_referral_link():
    """Generate a unique referral link hash."""
    try:
        # Generate a unique identifier
        unique_id = str(uuid.uuid4())

        # Hash the unique identifier
        referral_hash = hashlib.sha256(unique_id.encode()).hexdigest()

        return referral_hash
    except Exception as e:
        logging.error(f"Error generating referral link: {e}")
        return None
```

Listing 6.2 provides a simplified example of a Cloud Function that handles user registration and generates a unique referral link. It also includes a utility function "check_gcs_file_exists" to verify the existence of a file in Google Cloud Storage.

GCP provides a user-friendly interface for deploying, monitoring, and scaling Cloud Functions, also known as Cloud Run functions (Figure 6.2), simplifying the management of the backend infrastructure. The use of serverless functions allows the system to automatically scale based on demand, ensuring efficient resource utilization.

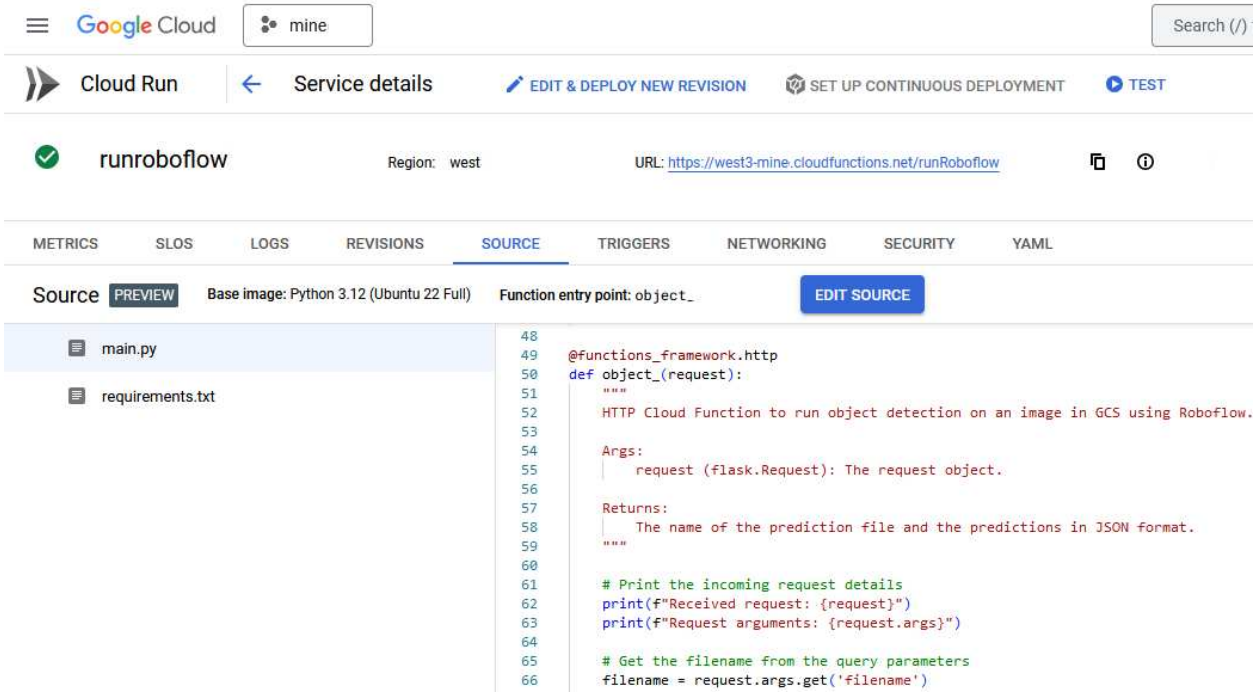


Figure 6.2: Interface of the Google Cloud Run function

6.4.2.4. Object Detection Model. The application utilizes the YOLO object detection model [73], trained on a dataset of 3D-printed EO replicas (Chapter 4) augmented with the two-stage strategy (Chapter 3). For online detection, the application communicates with the GCP API, via the Cloud Functions described in Sections 5.5.2 and 6.4.2.3.

For offline detection, the trained YOLOv8 model was converted to the ONNX format [131] and is deployed directly on the mobile device. The ONNX format was chosen for its efficient inference and compatibility with mobile hardware. The conversion process is illustrated in Listing 6.3.

Listing 6.3: Converting the trained YOLOv8 model to ONNX format for on-device deployment

```
from ultralytics import YOLO

# Load the trained YOLOv8 model
model = YOLO('/path/to/trained_model/best.pt')

# Export the model to ONNX format
# 'dynamic=True' allows for dynamic input shapes during inference
model.export(format='onnx', dynamic=True)
```

The ‘dynamic=True‘ argument during export enables dynamic input shapes, allowing the model to handle images of varying resolutions without retraining. The resulting ONNX model is bundled with the application during the build process. The ONNX Runtime library is used to load and execute the model for on-device inference.

6.5. System Design and Architecture

This section details the design and architecture of the cross-platform application, a core component of the unified algorithmic environment for EO detection. The application is designed to function independently, interfacing directly with the GCP for online processing and employing an on-device model for offline operation.

The system uses a client-server architecture and is developed to be compatible with various operating systems, including Windows, Linux, macOS, iOS, and Android (Figure 6.3). While designed for cross-platform deployment, this study focuses on the Android implementation, which is estimated to be used by 72% of the world’s smartphone users by March 2025 [132].

The primary framework used for development is Qt, version 6.7.2 [130]. This framework facilitates the development of software for multiple operating systems within a single project. While Qt simplifies cross-platform develop-

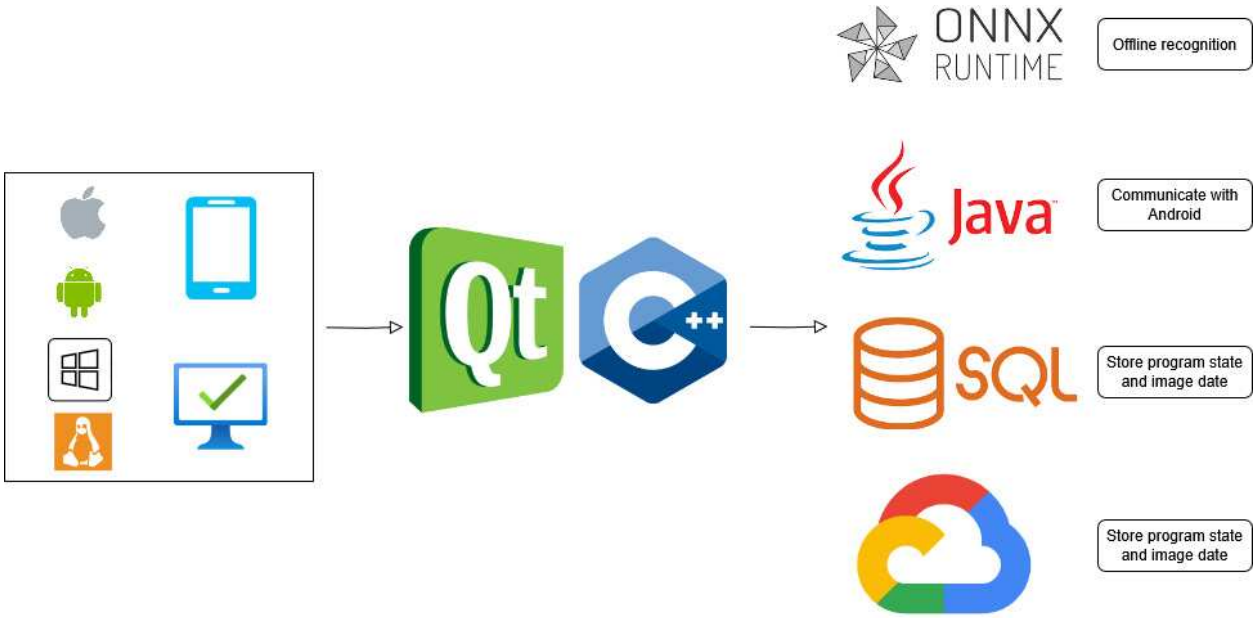


Figure 6.3: Upper-level architecture of the app

ment, the inherent differences between operating systems necessitate thorough testing and platform-specific code adjustments to ensure seamless functionality across all target platforms. Nevertheless, supporting a single codebase with Qt requires significantly fewer resources than creating and maintaining separate codebases for each operating system.

The user interface is developed using QML, a declarative language designed for creating dynamic and visually appealing user interfaces. QML's design, influenced by JavaScript, allows for the seamless integration of JavaScript code to handle scripting and application logic within the Qt framework. QML simplifies the development of OS-native user interfaces, enhancing user engagement and accelerating the development process. This is achieved because QML allows developers to focus on the aesthetics and layout of UI components rather than building them from the ground up. Additionally, the QML component of the system facilitates the selection of objects within an image by enabling users to draw a bounding rectangle or trace an outline. The resulting coordinates are then passed to the C++ backend of the application, where the core business logic resides. This logic includes image preprocessing, user input validation,

communication with server-side modules, and management of user data.

The user interface interacts with the C++ component through clearly defined interfaces. The application leverages C++ features such as object-oriented programming, inheritance, encapsulation, and support for multithreading. Long-running processes are executed in separate threads to prevent interface freezes and maintain responsiveness. The Qt Concurrent module facilitates high-level multithreading, automatically determining the optimal number of threads based on the number of processor cores and eliminating the need for manual management of low-level synchronization primitives.

An SQLite database is used to maintain the application's state, storing information about open files, modification history, user annotations, and other relevant data. SQLite's lightweight nature and support for SQL queries enable efficient data management with minimal resource consumption.

For online operation, the application is designed to transmit images of suspected EO to a server for further study and model training. This functionality is implemented via a module hosted on the GCP [122] (Figure 6.4). Images are

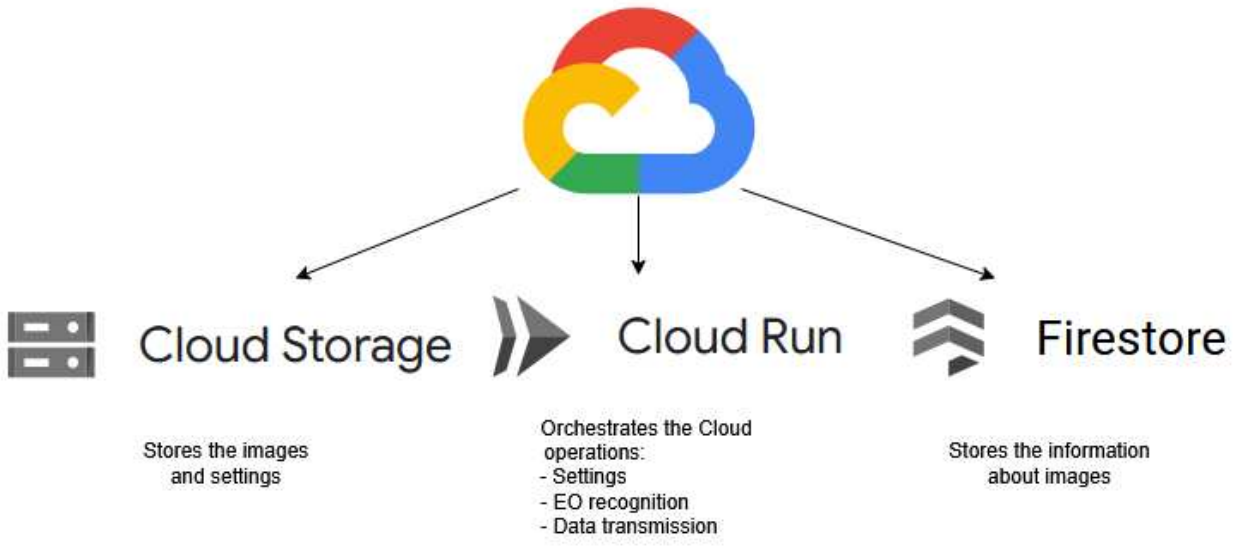


Figure 6.4: Main components of the GCP architecture in the system

stored in Google Cloud Storage, and requests are processed through Google

Cloud Functions, as detailed in Section 5.5.2. A Google Firestore NoSQL database manages system access, storing information about users, files, annotations, and settings. General application settings, EO lists, and other non-sensitive data are stored in Google Cloud Storage. Cloud Functions are also used to download updated EO lists and general settings, as well as to transmit logs and database data for analysis. These functions are written in Python.

For offline object detection, the application incorporates an on-device version of the YOLO model. The model is deployed using the ONNX Runtime module [131], a high-performance inference engine for ONNX models. ONNX (Open Neural Network Exchange) is an open standard for representing ML models, supported by various companies, including Microsoft, IBM, Intel, AMD, and Facebook. The Android operating system, through its Java components, interacts with the ONNX Runtime to perform object recognition directly on the device. The Qt JNI (Java Native Interface) module facilitates communication between the C++ application logic and the Java-based ONNX Runtime. While online mode, using the Roboflow API described in the previous chapter offers advantages in terms of access to the latest model updates and potentially higher processing power, the offline mode ensures functionality in areas with limited or no internet connectivity.

The application supports multiple languages, with English and Ukrainian localizations currently available. Adding new languages is streamlined through the use of the Qt Linguist module, which simplifies the translation of Qt application interfaces.

In summary, the system architecture comprises different modules and layers, promoting modularity and adaptability. This allows for modifications to individual components without affecting the rest of the system. The C++ backend, encompassing the core application logic, could be separated into a distinct service/application. Similarly, the functions executed on GCP could be migrated to other cloud platforms, such as Azure or Amazon Web Services. This modu-

lar design ensures the system's flexibility and alignment with modern software development best practices.

6.6. Explosive Objects Detection Model: YOLOv8 Implementation and Training

The core of the EO detection application lies in the YOLOv8 deep learning model [73], which is responsible for identifying explosive objects within images. This section details the model's training process, the dataset used, and the rationale behind key implementation choices.

The model was trained on a dataset comprising 1,438 photographs of 3D-printed landmine replicas, as detailed in Chapter 4. This dataset was subsequently expanded to 3,452 images through the application of data augmentation techniques, including rotation, scaling, noise addition, and grayscale conversion (see Section 4.10 and Table 4.3). The augmented dataset was divided into three subsets: training (3,021 images, 87.5%), validation (287 images, 8.3%), and testing (144 images, 4.2%). The images were resized to 640x640 pixels to optimize the balance between processing speed and detection accuracy, as this resolution is commonly used in YOLO training.

Table 7.2 presents the key hyperparameters used during the model training process.

Table 6.1

Hyperparameters used for training YOLOv8

Hyperparameter	**Value**
Batch Size	32
Initial Learning Rate	0.01
Number of Epochs	300
Optimizer	Adam

The initial learning rate was set to 0.01, and the algorithm autonomously adjusted it during training within the range of 0.0001 to 0.001. The model was trained for 300 epochs, with early stopping employed if no significant improvement in validation performance was observed.

6.7. Operational Algorithm of the Cross-Platform Application

This section outlines the operational algorithm of the cross-platform EO detection application, detailing the steps involved in processing user input, performing object detection, and handling optional user feedback and data transmission (Figure 6.5).

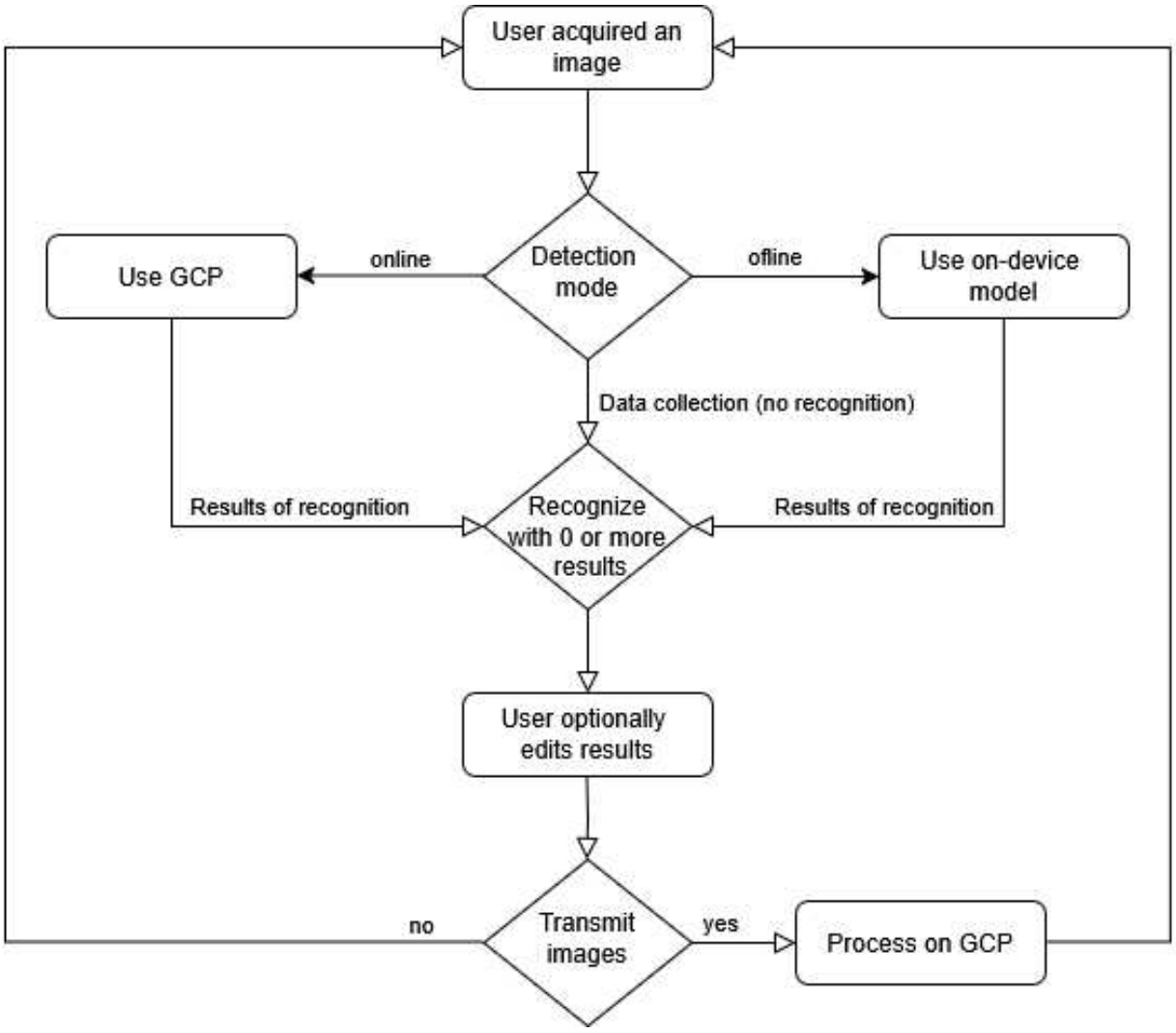


Figure 6.5: Operational algorithm for image processing

The application's workflow can be summarized as follows:

1. **Mode Selection and Image Acquisition:** Upon launching the application, the user is prompted to select an operating mode:
   — **Online Mode:** In this mode, the application leverages real-time processing on a remote server via the GCP API.
   — **Offline Mode:** This mode enables EO detection directly on the user's device without requiring an internet connection. It utilizes a streamlined, on-device version of the YOLO model, optimized for mobile processing.
   — **Data Collection Mode (No Detection):** This mode allows users to capture and store images without performing real-time object detection. It is primarily intended for collecting and annotating data that can be used for future model refinement.

   Subsequently, they can either open an existing image from the device's storage or capture a new image using the device's camera.
2. **EO Detection:**
   — **Online Mode:** The application transmits the image to the GCP via API, where it is analyzed using the cloud-hosted model (as described in 5.5.3 and 6.4.2.3). The analysis results, including object boundaries (bounding boxes) and detected EO types, are then transmitted back to the user's device.
   — **Offline Mode:** The application performs the detection process directly on the device using a localized version of the YOLO model in ONNX format (see 6.4.2.4).
3. **User Refinement of Results (Optional):** The application presents the detection results to the user, highlighting any detected EO with bounding boxes and class labels. Users can then manually adjust the bounding boxes, add or remove detections, or correct the predicted EO type. These refinements are stored locally in the SQLite database.

4. **Data Transmission for Model Improvement (Optional):** If desired, the user can choose to transmit data to the server for analysis and model refinement. This data may include:
   — The application's operational logs.
   — The application's local SQLite database containing user annotations and corrections.
   — Photographs of EO, available in either a reduced (default size 640x640 pixels) or original format.

   User privacy is protected by anonymizing the data before transmission.
5. **Server-Side Data Processing:**
   — Google Cloud Functions handle requests from the application.
   — Images are stored in Google Cloud Storage.
   — The Google Firestore database maintains data about users, files, annotations, and settings.
   — The collected data, including user-corrected annotations, is analyzed to identify patterns in misclassifications, assess the model's performance on different EO types, and evaluate the effectiveness of the user feedback mechanism. This analyzed data is then incorporated into the training dataset to retrain and refine the YOLO model, improving its accuracy and robustness over time.

6.7.1. Semi-Automatic Annotation Functionality. A key feature of the cross-platform application is its integrated semi-automatic annotation capability. This functionality is designed to address the ongoing challenge of data scarcity in the field of EO detection and to leverage the expertise of users in the field. The process works as follows (Figure 6.6):

1. **Initial Detection:** When a user submits an image (either captured directly with the device's camera or loaded from the device's storage), the application performs an initial object detection using either the on-

device model (in offline mode) or the cloud-based API (in online mode).

2. **Result Visualization:** The detection results, including bounding boxes and predicted class labels, are displayed to the user on the image.

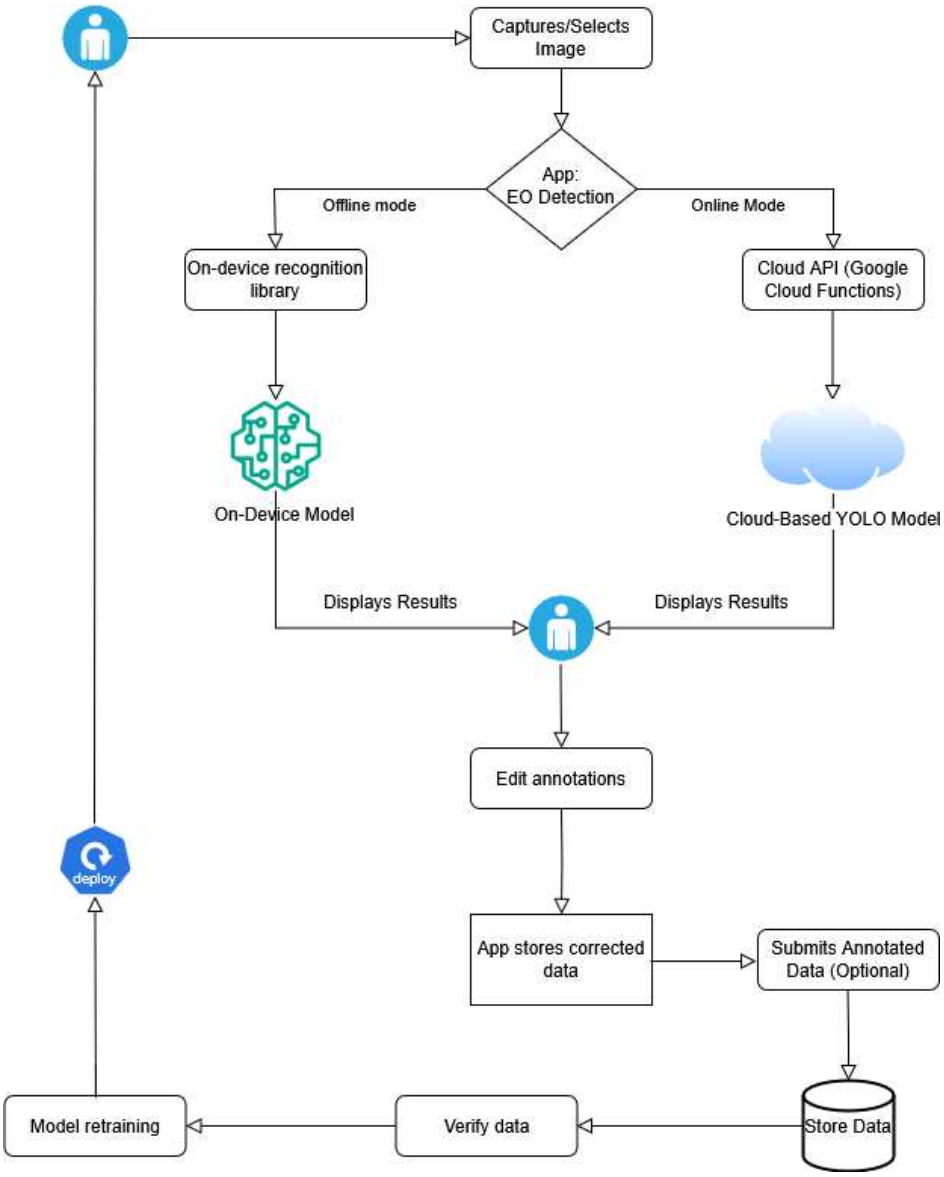


Figure 6.6: Data flow diagram of the semi-automatic annotation process

3. **User Correction and Refinement:** The user is then presented with tools to interact with and modify these initial detections (6.9.3):
   — **Add Bounding Boxes:** Users can manually draw bounding boxes around EO that were missed by the model.
   — **Delete Bounding Boxes:** Users can remove incorrect detections.

— **Modify Bounding Boxes:** Users can adjust the size and position of existing bounding boxes to improve their accuracy.

— **Change Class Labels:** Users can correct the predicted class label if the model misclassified the EO type.

4. **Data Submission:** After making any necessary corrections, the user can choose to submit the annotated image, along with the application's operational logs and (optionally) a copy of the local database, to a central server. This data is then used for further analysis and model retraining.
5. **Anonymization and Security:** All user-submitted data is anonymized to protect user privacy, and secure transmission protocols are employed to ensure data confidentiality.

This semi-automatic annotation process transforms the application from a passive detection tool into an active data collection and model improvement system. It leverages the "human-in-the-loop" approach to continuously enhance the accuracy and robustness of the EO detection models.

6.8. The Cross-Platform Application: Design, Functionalities, and Security

This section details the design and implementation of the cross-platform application, a key component of the unified algorithmic environment for EO detection. The application is designed to function independently, interfacing directly with the GCP for online processing and utilizing an on-device model for offline operation. While designed for cross-platform deployment, initial development and testing focused on the Android operating system.

6.8.1. User Interface and Interaction.

Upon launching the application, new users are required to enter a referral link (Figure 6.7). This link is used for tracking the application's distribution

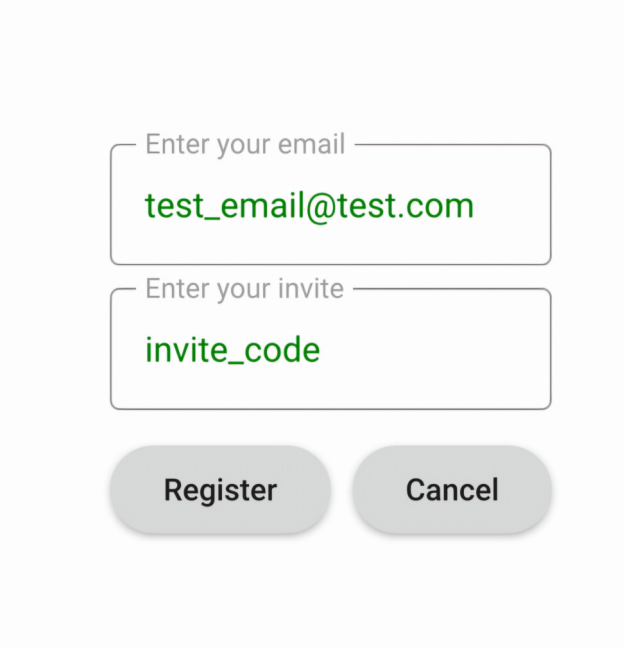


Figure 6.7: New user registration dialog, requiring a referral link

and usage, particularly within specific organizations or projects, and serves as a security measure to prevent unauthorized access. After initial setup, which also involves setting user preferences, the application can operate independently (offline). Online connectivity is only required for synchronizing the list of recognized EO types with the server and for transmitting data for model refinement.

Users can select the application's operating mode through the Settings menu (Figure 6.8):

— **Online Mode:** In this mode, the application leverages real-time processing on GCP, utilizing the same infrastructure as the messenger bot described in Chapter 5.

— **Offline Mode:** This mode enables EO detection directly on the user's device without requiring an internet connection, using a streamlined version of the model optimized for mobile processing.

— **Data Collection Mode (No Detection):** Allows to capture and store images without performing real-time object detection, primarily for collecting and annotating data for future model refinement.

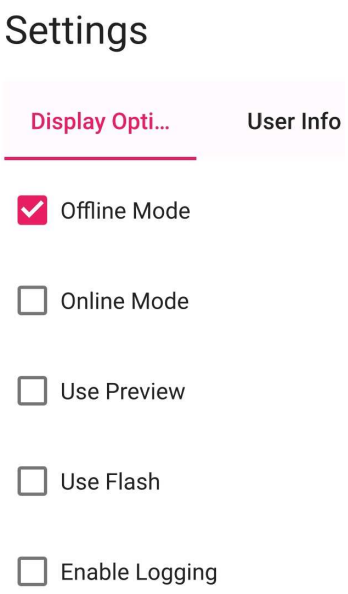
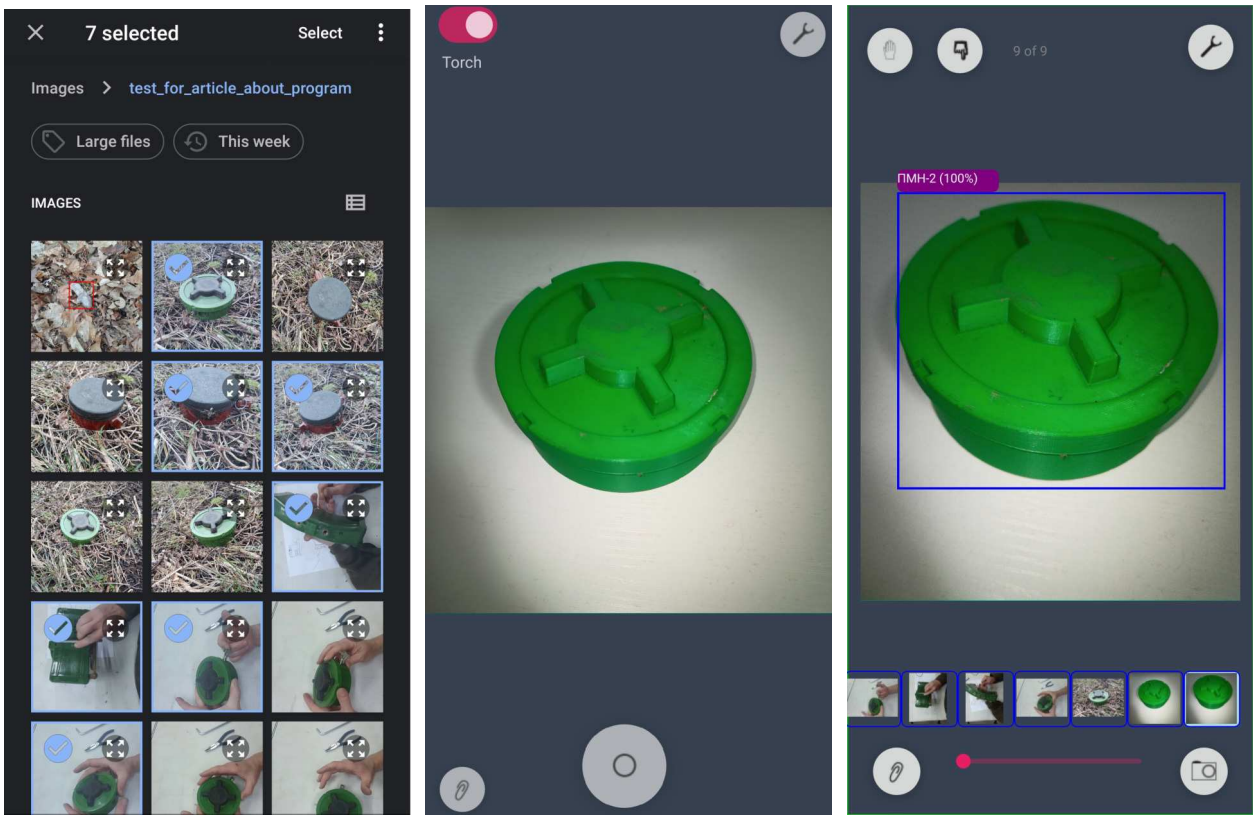


Figure 6.8: Settings menu, allowing users to select the operating mode

After selecting a mode, users can either open an existing image from the device's storage (Figure 6.9a) or capture a new image using the device's camera (Figure 6.9b). Following the image acquiring or selection, the user proceeds to the detection phase. After detection (in online or offline mode) or image selection (in data collection mode), users can annotate the image. The application allows for the simultaneous processing of multiple images (Figure 6.9c). To improve efficiency and reduce processing time, the application stores detection results in a local database. When an image is reopened, the corresponding data is retrieved from the database instead of re-performing the detection process.

6.8.2. Data Security and Privacy. Data transmission between the client application and the server is secured using the OpenSSL library [133]. This ensures the confidentiality and integrity of the data during transit. Furthermore, launching Cloud Functions on the server requires prior authorization, adding another layer of security.

6.8.3. Technical Specifications. The application is compatible with Android devices running Android 6 (Marshmallow) and higher. The applica-



(a) (b) (c)

Figure 6.9: Screenshots of the application's user interface: *a* – file selection dialog; *b* – image capture mode, flash enabled; *c* – editing mode, multiple files open

tion's performance in offline mode may vary depending on the specific device's hardware capabilities. While cloud functions offer horizontal scalability and can handle varying loads, irrespective of the number of clients, it is essential to consider potential bottlenecks in the system.

6.9. Performance Evaluation of the Mobile Application

This section presents the results of evaluating the cross-platform mobile application's ability to detect EO in images, with a focus on the performance of the on-device detection module.

6.9.1. Evaluation Methodology. To assess the effectiveness of the application, it was tested on a dataset of 125 images of five different types of actual landmines: MON-50, PMN, PMN-2, OZM-72, and PFM-1. These images were provided by professional demining experts and used with their permission. The application was evaluated in offline mode, relying on the on-device model for object detection. The primary evaluation metric was recall, as defined in Section 2.9 (Formula 2.6), due to its critical importance in the context of EO detection where failing to identify an EO (a false negative) has far more severe consequences than misclassifying a non-threatening object as an EO (a false positive). The average processing time per image was also recorded.

6.9.2. Results of Recognition. Table 6.2 presents the results of the on-device EO detection for each object type.

Table 6.2

Performance of the On-Device EO Detection Model

Type	**Qty**	**Avg. Time[1] (s)**	**Recognized**			**Recall (%)**
			Correct	**Incorrect[2]**	**Missed[3]**	
MON-50	25	2.2	23	1 (TM-62M)	1	92
PMN	25	2.0	23	–	2	92
PMN-2	25	2.1	23	1 (MON-50)	1	92
OZM-72	25	2.2	20	–	5	80
PFM-1	25	2.0	22	3 (MON-50)	0	88

[1] Average time to process an image in seconds (including image loading and recognition).

[2] Incorrect Class: The EO was detected but misclassified.

[3] False Negatives: EO present in the image but not detected by the model.

The on-device model achieved an overall recall rate of 88%, indicating that the application can serve as an effective tool for EO detection in scenarios where internet connectivity is unavailable. However, the recall rate varied significantly across different landmine types, ranging from 80% for the OZM-72 to 92% for the MON-50, PMN, and PMN-2. This variability suggests that the training

dataset may need to be expanded to include more images of OZM-72, captured under a wider range of conditions. The average processing time for a single image was 2.1 seconds, demonstrating the feasibility of near real-time on-device detection.

Figure 6.10 illustrates a specific case where the model struggled to detect an OZM-72 landmine in two nearly identical frames from a video.

Figure 6.10: Two nearly identical frames from a video depicting the detection of an OZM-72 landmine. The landmine on the left was not detected by the on-device model, highlighting the need for further model refinement. Source: Telegram channel of a military officer with the call sign Forester

This highlights the need for further model refinement and training data diversification. A set of images was intentionally selected to include challenging scenarios with varying image quality, lighting conditions, and object orientations to assess the model's robustness. For the mobile application to work as intended, users must take pictures to reduce the likelihood of recognition errors. Although this assumption is not mandatory, users who wish to accurately identify the type of EO are likely to acquire high quality images. However, this does

not mean that the model is incapable of detecting EO in low resolution images. On the contrary, the use of the application will promote the improvement of the model, mainly by collecting data on its inaccuracies.

6.9.3. User Refinement of Detection Results: The Edit Mode. The application provides an edit mode that allows users to refine the detection results, correcting errors made by the model and contributing to the ongoing improvement of its accuracy. In edit mode, users can modify the detection results if the model fails to detect an EO or misclassifies it as the wrong type.

The following actions are available in edit mode:

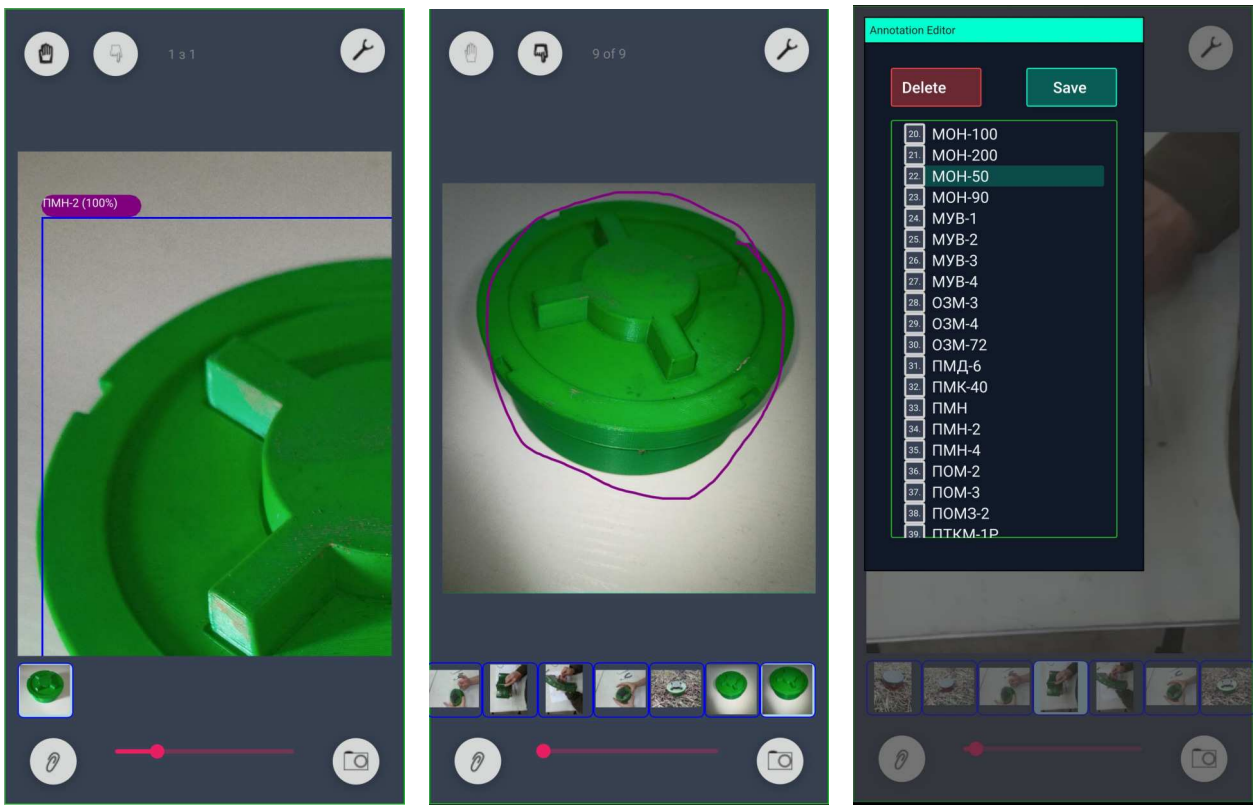
— **Image Navigation:** Users can zoom in and out using pinch gestures and pan across the image by dragging a finger, allowing for close inspection of potential EO (Figure 6.11a).

— **Object Selection:** Users can select an area containing an EO by tracing its outline with a finger gesture (Figure 6.11b). Once the outline is complete, it is automatically converted into a bounding box.

— **Removing selection:** Users can remove selection by tapping outside the bounding box.

— **EO Type Assignment:** Users can assign or modify the EO type associated with a selected area by choosing from a dropdown list of available EO types (Figure 6.11c).

For example, if the model fails to detect an EO in an image, the user can use the edit mode to highlight the object by tracing its outline and then specify its correct EO type from the provided list. All user-provided corrections are stored in the local SQLite database, ensuring that the modified results are retained when the image is reopened or the data is transmitted to the server for model retraining, as described in Section 6.9.4.



(a) Navigating the image using zoom and pan gestures

(b) Selecting an EO by tracing its outline

(c) Choosing the correct EO type

Figure 6.11: Screenshots of the application's edit mode: *a* – navigating the image using zoom and pan gestures; *b* – selecting an EO by tracing its outline; *c* – choosing the correct EO type from a dropdown menu

6.9.4. Data Transmission for Model Refinement. To continuously improve the accuracy and expand the capabilities of the EO detection model, the application enables users to contribute data to a central server. This data includes user-corrected annotations made in the edit mode (Section 6.9.3), providing valuable feedback for model refinement.

The user can choose to transmit data manually or to set up automatic transmission. For manual data transmission, the user must access the settings menu and select either the "Sync with server" option or the "Send images" option (Figure 6.12).

The user can transmit the following data:

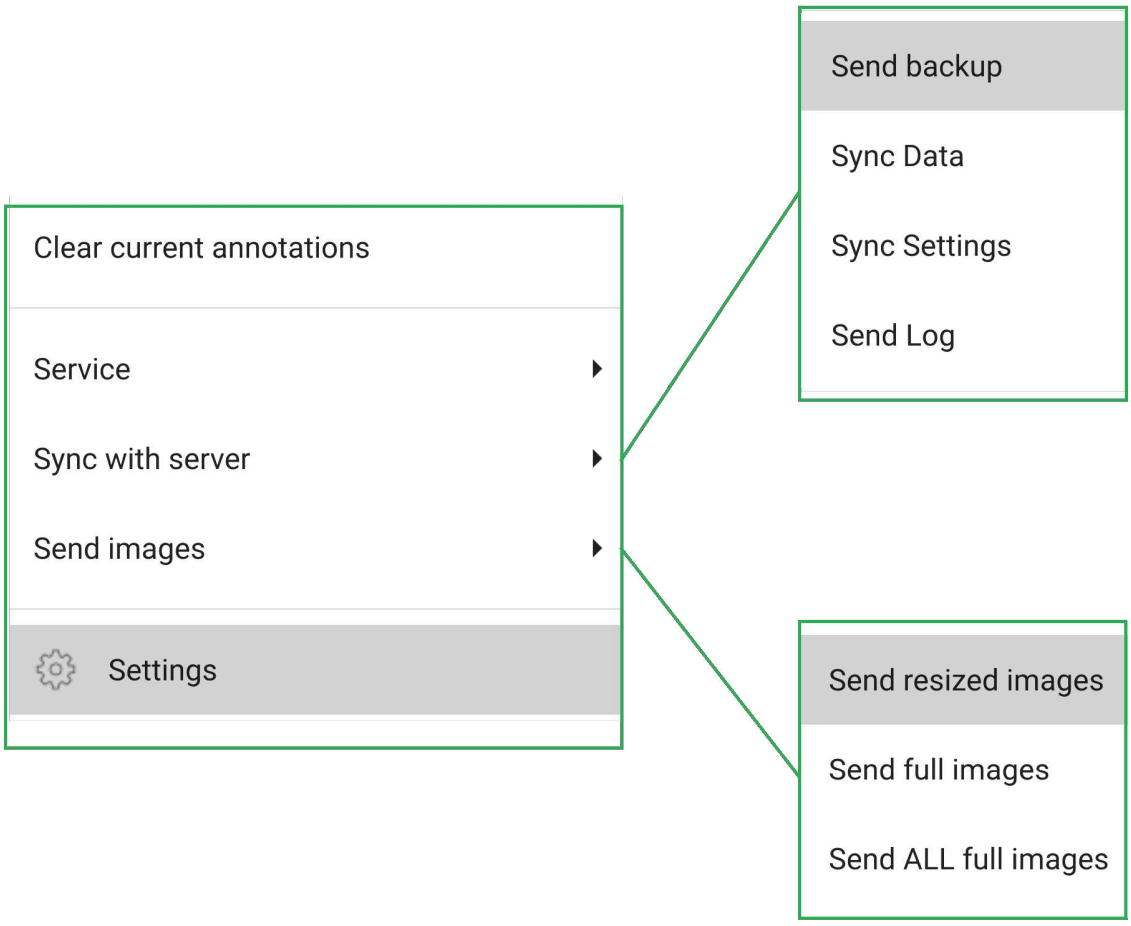


Figure 6.12: Screenshot of the data transfer menu: On the left is the main settings menu; on the right is the submenu for sending and receiving data

— Application operation logs for error analysis and troubleshooting.
— A duplicate of the application database for analysis and troubleshooting.
— Resized images (default size 640x640 pixels) that were active in the application at the time of transfer. Resized images are provided as a bandwidth-saving option for faster uploads.
— Original images that were active in the application at the time of transfer. Original images offer higher fidelity for detailed analysis.
— Any images that have been modified since the last data transfer, including user annotations and corrections. The application uses timestamps and a local database flag to track modified images.

User privacy is protected by anonymizing the data before transmission. All data transfer between the application and the server is encrypted using the OpenSSL library [133], as mentioned in Section 6.8.

Upon submission, the data is securely transmitted to Google Cloud Storage, where it undergoes a validation process to ensure data integrity and consistency. This includes checks for image format, annotation completeness, and data consistency. After validation, the data is incorporated into the training dataset for retraining and refining the model.

The application also includes a synchronization feature that allows the client to synchronize their settings and EO lists with the server using the "Sync with Server" menu option. This unidirectional synchronization allows the client to receive new data and update its settings. For instance, when a new setting or EO type is added on the server, the client can retrieve its value independently of other configurations.

6.10. Discussion

This chapter presented the development and evaluation of a cross-platform mobile application for real-time EO detection, a key component of the unified algorithmic environment introduced in this thesis. The application, leveraging the YOLO model and designed for both online and offline operation, was tested on a dataset of real EO images to assess its performance and generalization capabilities. While the initial development and testing focused on the Android operating system, the integration of the Qt framework with cloud services enables future deployment on alternative operating systems, including iOS, Windows, Linux, and macOS.

6.10.1. Summary of Findings. The application was evaluated on a dataset of real EO images, achieving an overall recall of 88% in offline mode (Table 6.2). This demonstrates the application's potential as an effective tool

for EO detection, particularly in areas with limited internet connectivity. The average processing time of 2.1 seconds per image in offline mode further highlights the feasibility of near real-time on-device detection. However, the analysis also revealed variations in performance across different EO types. Notably, the recall for the OZM-72 landmine was lower (80%) compared to other types (92% for MON-50, PMN, and PMN-2). This discrepancy underscores the need for further model refinement and a more comprehensive training dataset that includes a wider variety of OZM-72 images captured under diverse conditions.

6.10.2. Application's Main Features. The developed cross-platform application offers a user-friendly interface for EO detection. Users can load an image from their device's gallery (Figure 6.9a) or capture a new image using the camera (Figure 6.9b). The user can modify the results if the application fails to identify an EO, its class, or its bounding box. This option is available in non-detection mode (data collection mode), although it is not mandatory. All images accessed from within the application are assigned for transmission to the server for possible incorporation into the model. The application does not collect any personal data or geolocation information and uses modern encryption technologies, specifically the OpenSSL library [133], to protect the data during transmission. The transmission of data is entirely at the discretion of the user.

6.10.3. Comparison with Existing Solutions. The results of this part of the study confirm the effectiveness of using mobile applications for EO detection. This work describes the practical implementation and test results of a mobile application, in contrast to the study by Dorn et al. [49], which focused primarily on theoretical possibilities. This research illustrates the practical applications of a mobile application, although the study of [96] recognizes the usual limitations related to funding for innovative projects. Unlike the mobile

applications mentioned in the UNMAS 2013 report [50], which are seemingly no longer supported, the application created in this study is compatible with most modern Android smartphones. This work has created an application capable of functioning in both online and offline modes, although authors of [51] identified obstacles related to the implementation of the offline mode in mobile applications. Unlike the methodology used in [52], which limits the capabilities of a mobile application to the proximity of a desktop computer, the application created in this study is unencumbered by such limitations. Also, unlike the methodology used in [53], which relied on an external virtual reality engine and lacked its own dataset, this study implements a robust deep learning model that is capable of detecting EO under various conditions. Furthermore, unlike the possibilities of specialized internet resources [54] and [55], the created program has an offline mode and is capable of operating in areas with limited or no internet connectivity.

The developed application effectively performs mobile EO detection. It is suitable for use in practical environments due to its high accuracy, offline functionality, and provision for result verification and feedback. Online testing of the application depends on internet speed, which is often insufficient or unavailable in mined regions.

6.10.4. Limitations of the Current System. Despite the promising results, the current system has several limitations. The online mode is dependent on internet connectivity, which may be unreliable or unavailable in many explosives-affected regions. While the offline mode addresses this issue, the on-device model's performance may be constrained by the processing power and memory limitations of some mobile devices. The current testing has been performed on the Android operating system. Expanding the testing to include iOS devices would enable a comprehensive evaluation of the system's cross-platform compatibility and performance across different mobile ecosys-

tems. Furthermore, it is important to note that recognition accuracy may be affected by factors such as image quality, lighting conditions, and the presence of occlusions. While the application has shown good performance on the test dataset, further evaluation on a more diverse and challenging dataset of real-world images is necessary to fully assess its generalizability.

6.11. Conclusions to Chapter 6

The cross-platform mobile application presented in this chapter offers a promising tool for real-time EO detection. The application's ability to operate in both online and offline modes, coupled with its user-friendly interface and integration with cloud services, makes it a valuable asset for demining teams, humanitarian organizations, and civilians in explosive-affected areas. In both online and offline modes, deep learning models using 3D-printed landmine replicas are used. The program was evaluated offline on the Android operating system, but is deployable on alternative systems. Throughout its development, the application was designed for offline use and optimized for usability. The application of deep learning models allows the system to effectively identify explosive objects with a recall of 89% and an average processing time of 2.1 seconds.

The ability to modify recognition results is also an important function of the app. This is done by selecting an area in the image with a finger or a rectangular selection, and choosing the correct type of the EO afterwards. It enables the integration of user feedback, which can correct model inaccuracies and provide more training data. This feature allows the model to receive additional information about the location and type of EO, allowing it to adapt more effectively to real-world situations and improve recognition accuracy.

The ability to transmit detection results makes the application a significant source of data, especially given the scarcity of information on the subject.

The proposed method allows for a systematic organization of data collection and analysis, thus promoting model improvement. The implementation of this approach represents an important step in the creation of an effective mobile application for EO detection. The ability to transmit various data has been integrated, including application logs, a database copy, low-quality images, original images, and images modified since the last transmission. The lack of a unified data collection system for explosive devices poses a significant challenge to their detection and neutralization. The proposed application can streamline the gathering and examination of data from various contributors within the demining sector. In an area where information access is constrained, this could be invaluable to improving detection models and consequently accelerating demining initiatives in Ukraine.

Future development initiatives include the introduction of new EO variants, improvements to the existing model, and the incorporation of user comments and recommendations. One development focus is the creation of a training module in which users will be asked to identify objects. In addition, functionalities for notification and presentation of further information on identified objects will be incorporated. The ongoing development and refinement of this application, guided by user feedback and advancements in deep learning research, have the potential to significantly contribute to global demining efforts and enhance the safety of communities impacted by the persistent threat of EO.

CHAPTER 7
ADDING NEW TYPES OF EXPLOSIVE OBJECTS

This chapter outlines the procedure for integrating new EO types into the current detecting system. It employs a real case study methodology, illustrating the process from receiving of a user request to the deployment of updated models on both the cloud-based service (accessible through the messaging bot) and the cross-platform application (for offline detection).

7.1. User Request and Requirements Gathering

The procedure begins with a user request to enhance the system's recognition skills for a new type of EO. This case study will focus on the introduction of two new landmine variants: M56 and 9H24. The beginning of this procedure is frequently prompted by a request from the Landmine Bot (Figure 7.1) made by demining experts, humanitarian groups, or other users who have observed EO types not currently accommodated by the system.

Figure 7.1: Request for a new landmine

At this preliminary phase, users generally supply solely the **name** of a new EO type they have encountered. Upon receipt of a request, the subsequent procedures are undertaken to collect the requisite information:

— **Community Inquiries:** Inquiries are conducted via pertinent internet networks, encompassing specialist forums and social media groups

frequented by demining experts and military personnel. These requests seek images, videos, and other pertinent technical information regarding the recently identified EO kind.

— **Online Investigation:** An exhaustive examination of online resources, encompassing technical databases, military manuals, and open-source intelligence (OSINT) sources, is performed to collect information regarding the EO visual attributes, physical specifications, and deployment circumstances.

— **Expert Consultation:** If possible, direct communication is established with demining specialists or organizations that may have first-hand experience of the new EO kind to obtain more information and verify the gathered data.

The data collected through these procedures often encompasses:

— **Visual Attributes:** Images or videos of the EO captured from multiple perspectives and under diverse conditions, if accessible.

— **Physical Description:** Details regarding the object's dimensions, form, hue, inscriptions, and any notable characteristics.

— **Contextual Data:** Details about the customary sites of EO, including terrain classifications and flora, alongside their installation techniques, including surface-laid or subterranean placement.

7.2. Data Acquisition and Preparation

7.2.1. Video Acquisition. This case study primarily utilized video footage from Forester and other military members of the Ukrainian Armed Forces as the main data source about new types of EO.

The command-line utility FFmpeg was utilized to extract individual frames from the videos. The subsequent command was executed:

Listing 7.1: FFmpeg command for frame extraction

```
ffmpeg -i input_video.mp4 -vf fps=1 frame%04d.jpg
```

This command gets frames from the video file at a frequency of one frame per second and stores them as consecutively numbered JPEG images. An example of ffmpeg output is depicted in Fig 7.2.

Figure 7.2: Example of ffmpeg output

Not all frames were included in the dataset; many were eliminated due to excessive repetition or inadequacy for training purposes (insufficient quality, absence of objects, or just partial representation of objects).

7.2.2. Data Annotation with Roboflow. The retrieved frames 7.2 were subsequently uploaded to the Roboflow platform for annotation. Roboflow's annotation capabilities were utilized to delineate bounding boxes around each instance of the new EO kinds in the images, with corresponding class labels (63 and 64) assigned (Figure 7.3).

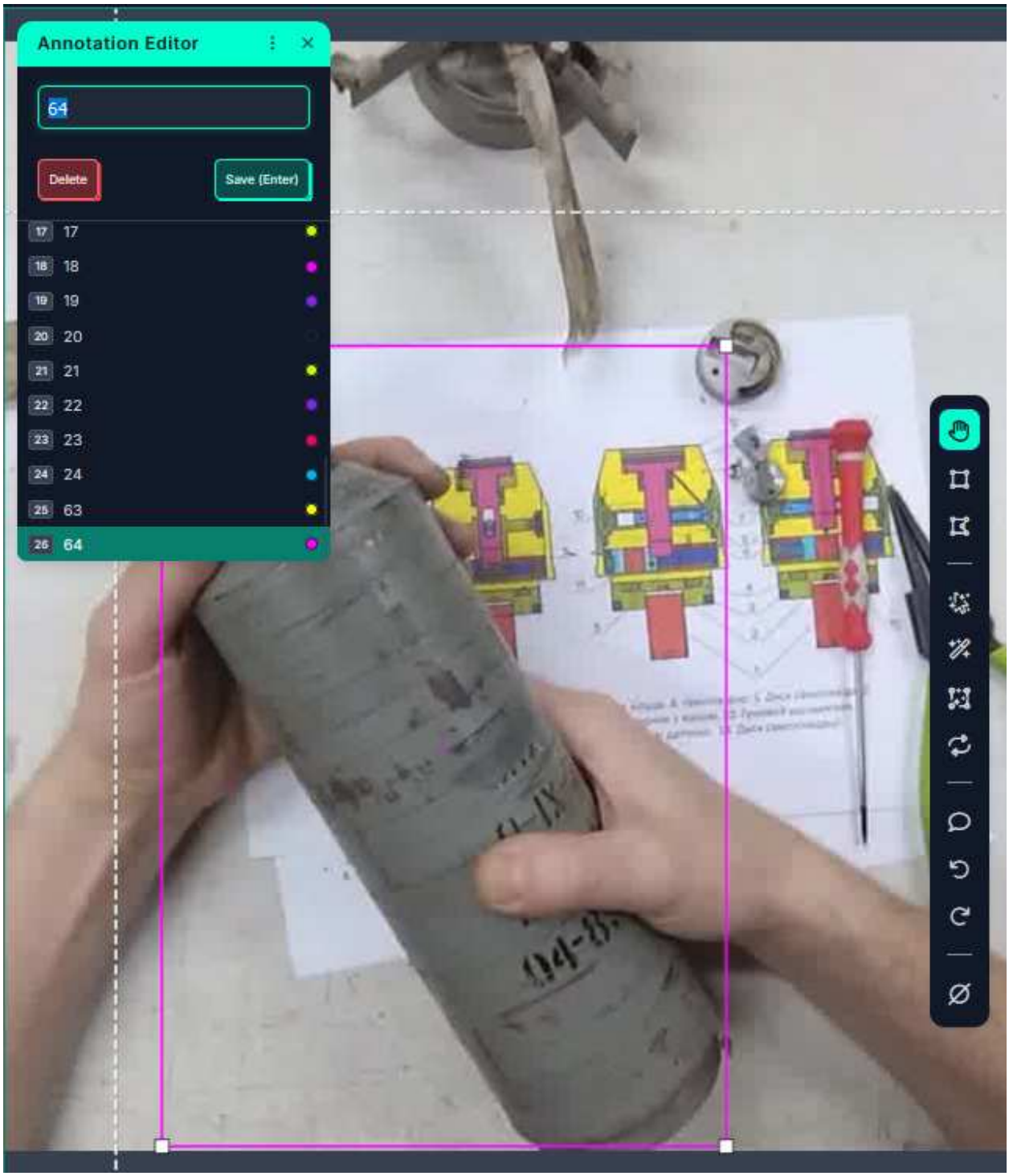


Figure 7.3: Roboflow's annotation interface. The source of the image: Forester

7.3. Dataset Development and Augmentation

Subsequent to the annotation process, the images were integrated into the existing Roboflow dataset. The dataset underwent the two-stage augmentation technique described in Chapter 3. Experiments were conducted with and without mosaic at the first stage of augmentation. In the absence of mosaic augmentation, the precision is 95% and the recall is 92%. Following the incorporation of mosaic, the measures rose to 98% for precision and 97% for recall, respectively. Instances of applied mosaic augmentation are depicted in Figure 7.4, with the corresponding results enumerated in Table 7.1.

(a) Mosaic in color

(b) Mosaic in grayscale

Figure 7.4: Samples from the dataset without and with mosaic applied

Table 7.1

Results of training on 200 epochs without and with mosaic

Type	mAP50	mAP50-95	Precision	Recall
No Mosaic	0.9587	0.8150	0.9501	0.9236
With Mosaic	0.9854	0.8864	0.9786	0.9693

All augmentations from 3.5 implemented in the initial phase, albeit with slightly varied parameters:

— **Flip** (Horizontal, Vertical).

— **90° Rotate** (CW, CCW, Upside down).

— **Grayscale:** Applied to 25% of the images in the dataset.

— **Noise:** Added to a maximum of 1.96% of the pixels (Figure 7.5).

— **Mosaic:** Applied to the whole dataset.

(a) PFM-1 (b) PMN-2

Figure 7.5: Samples from the dataset using 1.96% noise

The second stage entailed the implementation of more sophisticated augmentations, as detailed in Table 4.4. The mosaic was deactivated for the preceding 50 epochs to stabilize the training process.

7.4. Model Training with YOLOv11

The enhanced dataset was subsequently employed to train a novel YOLOv11 model. The training approach used the hyperparameters enumerated in Table 7.2.

Table 7.2

Hyperparameters Used for Training YOLOv11

Hyperparameter	Value
Batch Size	32
Initial Learning Rate	0.01
Number of Epochs	200
Optimizer	SGD
Transfer learning model	yolov11s
Close mosaic	last 50 epochs

The model was trained for 200 epochs, utilizing a batch size of 32. The training procedure was evaluated using the metrics specified in Section 2.9, including precision, recall, mAP50, and mAP50-95 (Figure 7.6). Training was conducted on the NVIDIA GeForce RTX 4070 Laptop GPU.

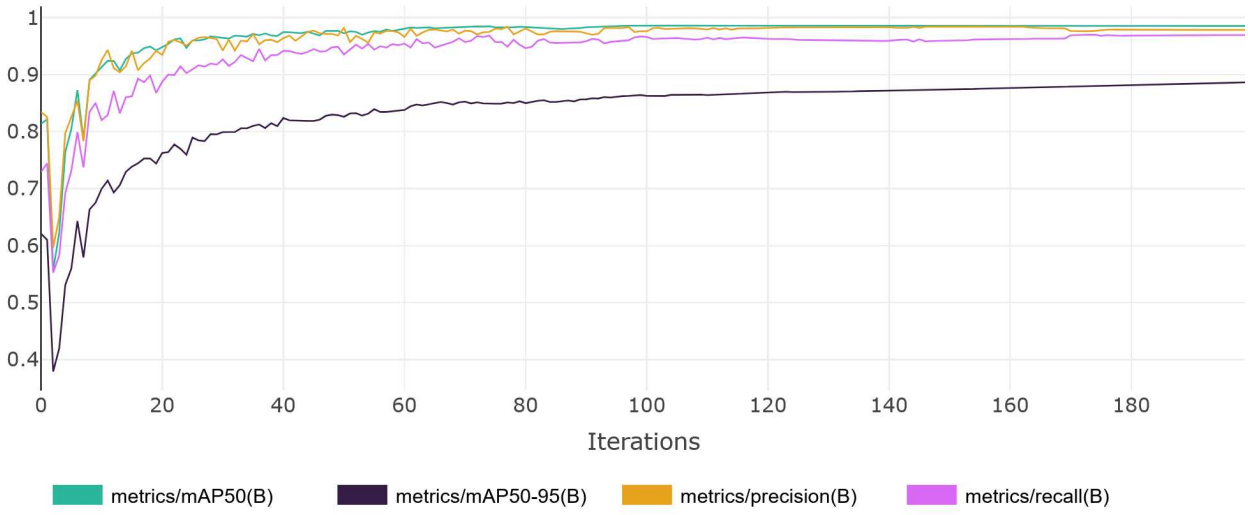


Figure 7.6: Metrics for YOLOv11 training

Figure 7.6 illustrates that around at the 100th epoch, the training process began to settle, accompanied by a consistent increase in mAP50-95. Additionally the Confusion matrix is illustrated on 7.7. From the analysis of Figure 7.7, it is evident that new landmine kinds have a 97% recognition rate.

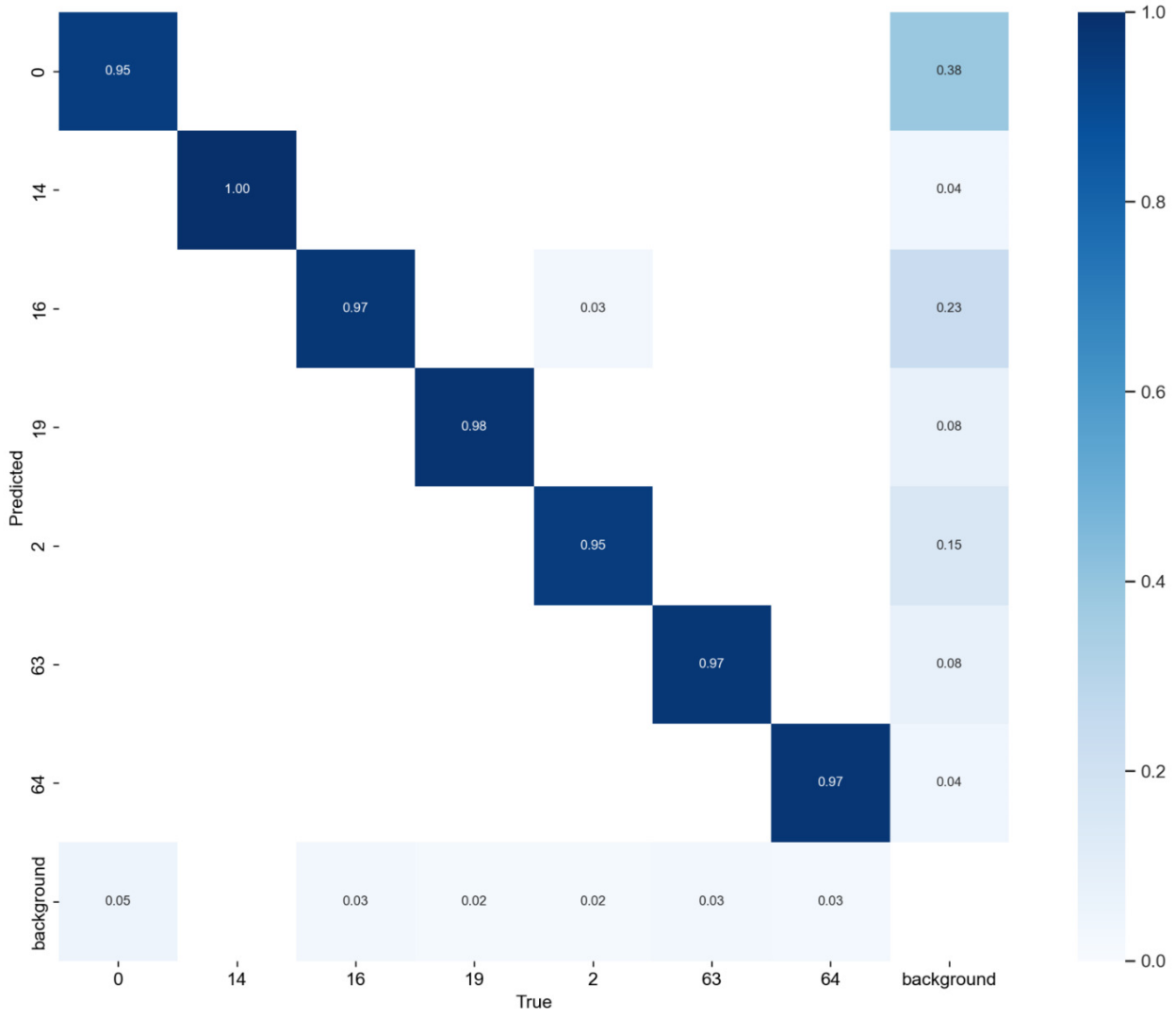


Figure 7.7: Confusion matrix for YOLOv11 algorithm. Class 0 – PFM-1; 14 – MON-50; 16 – PMN-2; 19 – OZM-72; 2 – PMN; 63 – M56; 64 – 9H24

7.5. Model Deployment

7.5.1. Deployment to Google Cloud Platform (Messenger Bot). The YOLOv11 model was deployed to the GCP to facilitate EO detection online through the messaging bot. The deployment method involved uploading the model into the Google Cloud Storage bucket and modifying the configuration files relevant to the model deployment (Figure 7.8).

```
{
    "ROBOFLOW_MODEL_ID": "recognize_mine",
    "ROBOFLOW_VERSION_NUMBER": "7",
    "is_active": true,
    "Name": "v0.17.0",
    "NameUkr": "Усі міни, x-модель",
    "Comment": "All previous types of landmines, including new M56 and 9H24",
    "model_id": "model_7_x"
}
```

Figure 7.8: Edit configuration file for access the new model

7.5.2. Deployment to the Cross-Platform Application. The trained model was translated to ONNX format for offline detection and incorporated into the cross-platform application (see 6.4.2.4). This enables users to do on-device EO detection without the necessity of an internet connection. The integration method entailed incorporating the model into the project and delivering it through the Android Play Market.

7.6. Assessment and Verification

The revised system, incorporating the newly introduced EO categories, was assessed utilizing a reserved test set of images, which included images of M56 and 9H24. The assessment was conducted utilizing both the messenger bot (online mode) and the cross-platform application (offline mode).

Table 7.3 Details the performance of newly incorporated EO and demonstrates that they have been effectively integrated into the model.

Figure 7.9 depicts instances of newly detected EO variants.

Table 7.3

Performance of the EO Detection Model

Type	Qty	Avg. Time[1] (s)	Correct	Incorrect	Recall (%)
M56	66	2.1	65	1	98
9H24	38	2.0	37	1	97

[1] Average time to process an image in seconds (including image loading and recognition).

7.7. Conclusions to Chapter 7

This chapter has illustrated the integration of novel EO kinds into the current detection system. The revised models, implemented on both the GCP and the cross-platform application, now facilitate the identification of M56 and 9H24 among the previously supported kinds. This case study underscores the system's adaptability and extensibility, demonstrating its potential to address evolving real-world requirements.

The use of the state-of-the-art algorithm YOLOv11 for training in this chapter is crucial, confirming that the transfer learning approaches proposed in this research are adaptable to new algorithms. Appendix A has the most recent model featuring over 20 varieties of EO, along with their statistics and samples. The model has been developed by transfer learning from YOLOv11x, the largest model, which offers the highest precision, if not the best speed. The training process for this model lasted approximately 10 days on an NVIDIA GeForce RTX 4070 Laptop GPU.

Please note that the new kinds of EO have been developed utilizing a video that depicts actual landmines, representing a limited range of real-world scenarios. Consequently, although the new classes demonstrated strong performance on the validation dataset, their performance with newly submitted images from varied real-world environments requires further investigation. In this instance, it is advantageous to continue acquiring new images from users to enhance the

(a) M56 (b) M56 (c) M56

(d) 9H24 (e) 9H24

Figure 7.9: Examples of recognized landmines M56 and 9H24. The source of the images: Forester

dataset's diversity. Should the data be inadequate, creating a three-dimensional replica of these landmines and performing a series of trials with the new models may yield valuable training data. This step was omitted as this chapter aims to illustrate the process of incorporating new classes according to user demands.

CHAPTER 8
CONCLUSIONS AND FUTURE WORK

The present thesis investigated the problem of EO detection using computer vision, with a focus on developing a practical and accessible solution utilizing 3D-printed replicas, data augmentation, a cross-platform application, cloud-based API for recognition, and a messenger bot interface. The research addressed the critical need for improved EO detection methods, particularly in regions like Ukraine, which are heavily affected by EO contamination.

The primary contributions of this work are:

1. **Development of a Novel Dataset:** A unique dataset was created using dimensionally accurate 3D-printed replicas of five prevalent anti-personnel landmine types found in Ukraine. The dataset comprises 1,438 images captured under diverse environmental conditions, simulating real-world scenarios (Section 4.8). This dataset served as the foundation for training the YOLO object detection models.
2. **Effective Use of Data Augmentation:** A comprehensive two-stage data augmentation strategy, detailed in Chapter 3, was implemented to enhance the size and diversity of the training dataset. This approach significantly improved the model's robustness and ability to generalize, particularly increasing recall rates. The best-performing augmentation strategy incorporated techniques such as rotation, scaling, noise addition, grayscale conversion, and notably, the mosaic augmentation.
3. **Successful Application of 3D Printing:** This research demonstrated the effectiveness of using 3D-printed landmine replicas for training computer vision models. The YOLOv8 model, trained on the synthetic dataset, achieved a precision of 98.0% and a recall of 98.2% on the

test set of 3D-printed replica images (Section 4.10). This validates the feasibility of using 3D printing to address data scarcity in EO detection.

4. **Development of a Messenger Bot Interface:** A messenger bot interface, detailed in Chapter 5, was developed and integrated with the Google Cloud Platform, providing an alternative and widely accessible means of interacting with the EO detection service.
5. **Design and Implementation of a Cross-Platform Application:** A user-friendly, cross-platform application, detailed in Chapter 6, was developed using the Qt framework, enabling EO detection on various devices, including Android smartphones. The application is equipped with the capacity to function in both online and offline modes, thereby ensuring adaptability to a wide range of operational environments.
6. **Evaluation on Real Landmine Data:** The trained model was evaluated on an independent dataset of 254 real landmine images provided by demining professionals (Section 4.12). While the performance was lower than on the synthetic dataset (achieving an average precision of 91.0% and a recall of 79.1%), the results demonstrate the potential of the system to generalize to real-world scenarios. The model exhibited high precision for PMN (97.2%) and PMN-2 (93.2%), but lower recall for MON-50 (60.0%), indicating areas for further model refinement.
7. **Integration with Google Gemini:** The system was integrated with Google Gemini to provide users with supplementary information about detected EO, enhancing situational awareness and safety (Section 5.6).
8. **User Feedback Mechanism:** The application incorporates a mechanism for users to correct detection errors and submit feedback, creating a valuable feedback loop for continuous model improvement (Section 6.9).

8.1. Future Research Directions

Based on the results and limitations identified in this study, several promising directions for future research are proposed. These directions aim to further develop, improve, and extend the technologies for detecting EO:

1. **Dataset Expansion and Improvement:**
   - Expand the dataset with more real-world landmine images, particularly for underperforming classes like the MON-50, to improve model robustness and generalizability.
   - Create more realistic 3D-printed replicas, focusing on accurate shape, texture, and material properties to better simulate real-world EO appearance.
   - Collect data under a wider range of environmental conditions (e.g., varying lighting, weather, soil types, vegetation) and viewpoints to enhance the dataset's diversity.
2. **Model Refinement:**
   - Investigate more advanced deep learning architectures, such as EfficientDet and Transformer-based models (DETR, Deformable DETR), to potentially improve detection accuracy.
   - Enhance the performance of the on-device model on resource-constrained devices by exploring techniques such as model pruning, quantization, and knowledge distillation.
   - Explore techniques for handling occlusions and challenging backgrounds more effectively, such as investigating attention mechanisms or incorporating contextual information.
   - Incorporate user feedback data, including corrections and annotations, into the model retraining process to address specific weaknesses and improve overall performance.

3. **Cross-Platform Deployment and Optimization:**
   — Deploy the cross-platform application on other operating systems (iOS, Windows, Linux, macOS) and optimize its performance for each platform.
   — Further optimize the on-device model for faster inference and reduced memory footprint, using techniques such as model pruning, quantization, and knowledge distillation.
4. **Integration with Other Sensors:**
   — Explore the integration of data from other sensors, such as magnetometers and ground-penetrating radar (GPR), to enhance detection accuracy, particularly for buried or obscured EO. This could involve developing sensor fusion algorithms to combine visual data with data from other sources.
5. **Real-World Testing and Deployment:**
   — Conduct extensive field testing of the application in collaboration with demining professionals in real-world scenarios to evaluate its performance under realistic conditions and gather user feedback.
   — Develop a deployment strategy for integrating the application into existing demining workflows, considering factors such as user training, data management, and communication protocols.
6. **Unified Algorithmic Environment:**
   — Fully develop and integrate the messenger bot and API components into a unified algorithmic environment, as described in Chapter 6, to provide a seamless user experience across different platforms.
   — Evaluate the performance and usability of the complete system, including the application, messenger bot, and cloud infrastructure, in real-world settings.
7. **Training and Education:**
   — Develop and integrate a training module within the application or

as a separate tool to educate users, including both demining professionals and civilians, on EO identification, safety procedures, and the use of the detection system.

— Incorporate features for providing users with more detailed information about detected EO, potentially drawing upon external resources through the Google Gemini integration.

8. **Personalization:**

— Explore the personalization of the model for specific users, such as deminers, to improve accuracy. This could involve fine-tuning the model based on user-specific data or allowing users to select different model variants tailored to specific environments or EO types.

By pursuing these research directions, the developed system can evolve into a more robust, reliable, and user-friendly tool for EO detection, ultimately contributing to safer and more efficient humanitarian demining operations and reducing the devastating impact of explosive objects on communities worldwide.

BIBLIOGRAPHY

1. Kunichik, O. (2023, April 15). *Detection of explosive objects* [Conference presentation abstract]. The 21st International Scientific and Practical Conference "Shevchenko Spring–2023," Kyiv, Ukraine, 92–93. Available: https://probability.knu.ua/shv2023/ShV_2023.pdf
2. Kunichik, O., & Tereshchenko, V. (2023, November 21). *Enhancing landmine detection through the integration of 3D printing and deep learning techniques* [Conference presentation abstract]. Tenth International Conference Information Technologies and Implementation (Satellite), Kyiv, Ukraine, 28–29. Available: http://iti.fit.univ.kiev.ua/wp-content/uploads/%D0%97%D0%B1%D1%96%D1%80%D0%BA%D0%B0_ITI_2023.pdf
3. Kunichik, O., & Tereshchenko, V. (2024, March 15–16). *Advancing landmine detection: a methodological breakthrough using 3d printing and computer vision* [Conference presentation abstract]. Artificial intelligence: achievements, challenges, risks, Kyiv, Ukraine, 447-450. Available: https://www.ipai.net.ua/docs/ai_conf_15.03.2024.pdf
4. Kunichik, O. (2024, October 18–19). *Using a Telegram bot to detect landmines with artificial intelligence* [Conference presentation abstract]. Twenty-fourth international scientific and technical conference artificial intelligence and intelligent systems AIIS'2024, Kyiv, Ukraine, 216–220. Available: https://www.ipai.net.ua/docs/AIIS-2024.pdf
5. Kunichik, O. (2024, November 21–22). *Mobile application for landmine detection* [Conference presentation abstract]. The 11th All-Ukrainian Scientific and Practical Conference of Students, Postgraduates and Young Scientists "United by Science: Prospects of Interdisciplinary Research," Kyiv,

Ukraine, 311–313.

6. Kunichik, O. (2024, December 11–12). *Cloud-based landmine detection service with messenger bot integration* [Conference presentation abstract]. The 23rd International Scientific Conference "Neural Network Technologies and Applications NNTA-2024," Kramatorsk – Vinnytsia – Ternopil Ukraine, 216–220.
7. Kunichik, O., & Tereshchenko, V. (2022). Analysis of modern methods of search and classification of explosive objects. *Artificial Intelligence and Intelligent Systems,* 27(2), 52–59. Available: https://doi.org/10.15407/jai2022.02.052
8. Kunichik, O., & Tereshchenko, V. (2023). Improving the accuracy of landmine detection using data augmentation: a comprehensive study. *Artificial Intelligence and Intelligent Systems,* 28(2), 42–54. Available: https://doi.org/10.15407/jai2023.02.042
9. Kunichik, O. (2024). Review of strategies to overcome the lack of data in landmine detection. *Artificial Intelligence and Intelligent Systems,* 29 (3), 99–103. Available: https://doi.org/10.15407/jai2024.03.099
10. Kunichik, O., & Tereshchenko, V. (2024). Determining the effectiveness of using three-dimensional printing to train computer vision systems for landmine detection. *Eastern-European Journal of Enterprise Technologies,* 5, 1(131), 17–29. Available: https://doi.org/10.15587/1729-4061.2024.311602
11. Kunichik, O. (2024). Landmine detection with a mobile application. *Eastern-European Journal of Enterprise Technologies,* 6, 2(132), 6–13. Available: https://doi.org/10.15587/1729-4061.2024.317103
12. State Emergency Service of Ukraine. (2023). Educating the population about the risks associated with handling explosive ordnance [PDF file]. Available: https://dsns.gov.ua/upload/2/6/8/9/6/7/1QBewzVACHxt3lQilLbOwC6hsX6WcrOsyFIFiscl.pdf

13. United Nations Development Programme. (2023). In Ukraine, tackling mine action from all sides to make land safe again. Available: https://www.undp.org/european-union/stories/ukraine-tackling-mine-action-all-sides-make-land-safe-again
14. Landmine–Monitor–2022. Available at: https://backend.icblcmc.org/assets/reports/LandmineMonitors/LMM2022/Chapter-Images/Downloads/2022_Landmine_Monitor_web.pdf
15. C. Gooneratne, S. Mukhopadhyay and G. Sen Gupta, "A Review of Sensing Technologies for Landmine Detection: Unmanned Vehicle Based Approach," in *Proc. 2nd Int. Conf. Auton. Robots Agents*, Palmerston North, New Zealand, 2004, pp. 401. Available: https://www.researchgate.net/publication/238109550
16. M. G. Kale, V. R. Ratnaparkhe and A. S. Bhalchandra, "Sensors for Landmine Detection and Techniques: A Review," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 2, no. 1, Jan. 2013. Available: https://www.ijert.org/research/sensors-for-landmine-detection-and-techniques-a-review-IJERTV2IS14pdf
17. T. Hutsul, M. Khobzei, V. Tkach, O. Krulikovskyi, O. Moisiuk, V. Ivashko and A. Samila, "Review of approaches to the use of unmanned aerial vehicles, remote sensing and geographic information systems in humanitarian demining: Ukrainian case," *Heliyon*, vol. 10, no. 7, p. e29142, Jul. 2024, doi: 10.1016/j.heliyon.2024.e29142
18. H. Kasban, O. Zahran, M. El-Kordy, S. M. Elaraby, and F. E. Abd El-Samie, "Automatic object detection from acoustic to seismic landmine images," in *Proc. Int. Conf. Comput. Eng. Syst. (ICCES)*, Cairo, Egypt, 2008, pp. 193–198, doi: 10.1109/ICCES.2008.4772995
19. D. J. Daniels, *Ground Penetrating Radar,* 2nd ed. London, UK: The Institution of Electrical Engineers, 2004, pp. 154–196, doi: 10.1049/PBRA015E

20. F. Lombardi, M. Lualdi, F. Picetti, P. Bestagini, G. Janszen, and L. A. Di Landro, "Ballistic ground penetrating radar equipment for blast-exposed security applications," *Remote Sensing,* vol. 12, no. 4, p. 717, Feb. 2020, doi: 10.3390/rs12040717
21. I. J. Won, D. A. Keiswetter, and T. H. Bell, "Electromagnetic induction spectroscopy for clearing landmines," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 4, pp. 703–709, Apr. 2001, doi: 10.1109/36.917876
22. W. Messelink, K. Schutte, A. Vossepoel, F. Cremer, and J. Schavemaker, "Feature-Based Detection of Landmines in Infrared Images," in *Proc. SPIE– Int. Soc. Opt. Eng.*, vol. 4742, Feb. 2003, doi: 10.1117/12.479081
23. L. Cardona, H. Itozaki, J. Jiménez-Builes, N. Vanegas Molina, and H. Sato-Akaba, "Design of a radio-frequency transceiver coil for landmine detection in Colombia by nuclear quadrupole resonance," *Heliyon*, vol. 6, no. 2, p. e03242, Feb. 2020, doi: 10.1016/j.heliyon.2020.e03242
24. J. Zeng, C. Chu, G. Ding, Q. Xiang, F. Hao, and X. Luo, "Experimental investigation of thermal neutron analysis based landmine detection technology," *Nucl. Techn.*, vol. 36, no. 2, pp. 28–33, Feb. 2013. doi: https://doi.org/10.11889/j.0253-3219.2013.hjs.36.020205
25. H. Khan, Z. Koreshi, and M. Yaqub, "Sensitivity studies of an EO explosive detection system based on neutron backscattering using Monte Carlo simulation," *Nucl. Technol. Radiat. Prot.*, vol. XXXII, pp. 37–43, Mar. 2017, doi: 10.2298/NTRP1701037K
26. A. Faust, R. Rothschild, P. Leblanc, and J. McFee, "Development of a Coded Aperture X-Ray Backscatter Imager for Explosive Device Detection," *IEEE Trans. Nucl. Sci.*, vol. 56, pp. 299–307, Mar. 2009, doi: 10.1109/TNS.2008.2009537
27. J. Lee and H. Lee, "Modeling Residual Magnetic Anomalies of Landmines Using UAV-Borne Vector Magnetometer: Flight Simulations and Experimental Validation," *Remote Sens.*, vol. 16, p. 2916, Aug. 2024, doi:

10.3390/rs16162916

28. Y. Ege, A. Kakilli, O. Kılıç, H. Çalık, H. Çıtak, S. Nazlıbilek and O. Kalender, "Performance Analysis of Techniques Used for Determining Land Mines," *Int. J. Geosci.*, vol. 5, no. 10, pp. 1163–1189, 2014, doi: 10.4236/ijg.2014.510098
29. Y. Zhang, D. Huston and T. Xia, "Underground object characterization based on neural networks for ground penetrating radar data," in *Proc. SPIE 9804, Nondestruct. Charact. Monit. Adv. Mater. Aerosp. Civil Infrastruct.*, 2016, pp. 10–18, doi: 10.1117/12.2219345
30. C. Warren, A. Giannopoulos and I. Giannakis, "gprMax: Open source software to simulate electromagnetic wave propagation for Ground Penetrating Radar," *Comput. Phys. Commun.*, vol. 209, pp. 163–170, 2016, doi: 10.1016/j.cpc.2016.08.020
31. S. Tubaro, P. Bestagini, S. Lameri, F. Lombardi and M. Lualdi, "Landmine Detection From Gpr Data Using Convolutional Neural Networks," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, 2018, doi: 10.5281/zenodo.1159843
32. H. M. Alshamy et al., "Evaluation of GPR Detection for buried objects material with different depths and scanning angles," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1090, no. 1, p. 012042, 2021, doi: 10.1088/1757-899X/1090/1/012042
33. C. Castiblanco, J. Rodriguez, I. Mondragon, C. Parra and J. Colorado, "Air Drones for Explosive Landmines Detection," in *ROBOT2013: First Iberian Robotics Conference. Advances in Intelligent Systems and Computing*, M. Armada, A. Sanfeliu and M. Ferre, Eds. Cham, Switzerland: Springer, 2014, vol. 253, doi: 10.1007/978-3-319-03653-3_9
34. R. Achkar and M. Owayjan, "Implementation of a Vision System for an EO Detecting Robot Using Artificial Neural Network," *Int. J. Artif. Intell. Appl.*, vol. 3, 2012, doi: 10.5121/ijaia.2012.3507

35. A. Nikulin, T. S. De Smet, J. Baur, W. D. Frazer and J. C. Abramowitz, "Detection and Identification of Remnant PFM-1 'Butterfly Mines' with a UAV-Based Thermal-Imaging Protocol," *Remote Sens.*, vol. 10, no. 11, p. 1672, 2018, doi: 10.3390/rs10111672
36. J. Baur, G. Steinberg, A. Nikulin, K. Chiu and T. S. De Smet, "Applying Deep Learning to Automate UAV-Based Detection of Scatterable Landmines," *Remote Sens.*, vol. 12, no. 5, p. 859, 2020, doi: 10.3390/rs12050859
37. J. Baur, K. Dewey, G. Steinberg and F. O. Nitsche, "Modeling the Effect of Vegetation Coverage on Unmanned Aerial Vehicles-Based Object Detection: A Study in the Minefield Environment," *Remote Sens.*, vol. 16, no. 12, p. 2046, 2024, doi: 10.3390/rs16122046
38. S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image Data Augmentation for Deep Learning: A Survey," 2023, arxiv: 2204.08610, doi: 10.48550/arXiv.2204.08610
39. A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, and J. Borovec et al., "Imgaug," 2019. Available: `https://github.com/aleju/imgaug`
40. TensorFlow Developers, "Tensorflow," *Zenodo*, Aug. 17, 2023, doi: 10.5281/zenodo.8256979
41. A. Barnawi, K. Kumar, N. Kumar, B. Alzahrani and A. Almansour, "A Deep Learning Approach for Landmines Detection Based on Airborne Magnetometry Imaging and Edge Computing," *Comput. Model. Eng. Sci.*, vol. 139, no. 2, pp. 2117–2137, 2024, doi: 10.32604/cmes.2023.044184
42. P. Bestagini, F. Lombardi, M. Lualdi, F. Picetti and S. Tubaro, "Landmine Detection Using Autoencoders on Multipolarization GPR Volumetric Data," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 182–195, 2021, doi: 10.1109/tgrs.2020.2984951
43. L. Mochurad, V. Savchyn and O. Kravchenko, "Recognition of Explosive Devices Based on the Detectors Signal Using Machine Learning Methods,"

in *Proc. 4th Int. Workshop Intell. Inf. Technol. Syst. Inf. Secur.*, 2023. Available: https://ceur-ws.org/Vol-3373/paper14.pdf

44. X. Bai, Y. Yang, S. Wei, G. Chen, H. Li and Y. Li et al., "A Comprehensive Review of Conventional and Deep Learning Approaches for Ground-Penetrating Radar Detection of Raw Data," *Appl. Sci.*, vol. 13, no. 13, p. 7992, 2023, doi: 10.3390/app13137992
45. P. Srimuk, A. Boonpoonga, K. Kaemarungsi, K. Athikulwongse and S. Dentri, "Implementation of and Experimentation with Ground-Penetrating Radar for Real-Time Automatic Detection of Buried Improvised Explosive Devices," *Sensors*, vol. 22, no. 22, p. 8710, 2022, doi: 10.3390/s22228710
46. O. A. Pryshchenko, V. Plakhtii, O. M. Dumin, G. P. Pochanin, V. P. Ruban, L. Capineri and F. Crawford, "Implementation of an Artificial Intelligence Approach to GPR Systems for Landmine Detection," *Remote Sens.*, vol. 14, no. 17, p. 4421, 2022, doi: 10.3390/rs14174421
47. A. Barnawi, K. Kumar, N. Kumar, B. Alzahrani and A. Almansour, "A Graph Learning Framework for Prediction of Missing Landmines Using Airborne Magnetometry in IoT Environment," 2023, doi: 10.2139/ssrn.4526746
48. H. Pahadia, D. Lu, B. Chakravarthi and Y. Yang, "SKoPe3D: A Synthetic Dataset for Vehicle Keypoint Perception in 3D from Traffic Monitoring Cameras," in *Proc. 2023 IEEE 26th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2023, vol. 28, pp. 4367–4372, doi: 10.1109/itsc57777.2023.10422667
49. A. W. Dorn, "Eliminating Hidden Killers: How Can Technology Help Humanitarian Demining?," *Stability: Int. J. Secur. Develop.*, vol. 8, no. 1, p. 5, 2019, doi: 10.5334/sta.743
50. Annual Report 2013. United Nations Mine Action Service. Available: https://www.unmas.org/sites/default/files/unmas_2013_annual_report_digital_presentation_0.pdf

51. A. P. Susanto, H. Winarto, A. Fahira, H. Abdurrohman, A. P. Muharram, U. R. Widitha, G. E. Warman Efirianti, Y. A. Eduard George and K. Tjoa, "Building an artificial intelligence-powered medical image recognition smartphone application: what medical practitioners need to know," *Inform. Med. Unlocked*, vol. 32, p. 101017, 2022, doi: 10.1016/j.imu.2022.101017
52. R. Mori, M. Okawa, Y. Tokumaru et al., "Application of an artificial intelligence-based system in the diagnosis of breast ultrasound images obtained using a smartphone," *World J. Surg. Oncol.*, vol. 22, p. 2, 2024, doi: 10.1186/s12957-023-03286-1
53. Q. A. Hameed, H. A. Hussein, M. A. Ahmed, M. M. Salih, R. D. Ismael and M. B. Omar, "UXO-AID: A New UXO Classification Application Based on Augmented Reality to Assist Deminers," *Computers*, vol. 11, p. 124, 2022, doi: 10.3390/computers11080124
54. Interactive map of areas that could potentially be contaminated by explosive ordnance. Official website of The State Emergency Service of Ukraine. Available: `https://mine.dsns.gov.ua/`
55. Bezpeka Info. United Nations Children's Fund (UNICEF). Available: `https://courses.bezpeka.info/home`
56. I. Kalifa, A. Youssif and A. Adel, "The Use of Mobile Technology for Detecting Landmines," *Int. J. Comput. Appl.*, vol. 92, 2014, doi: 10.5120/16008-5034
57. Landmine–Monitor–2023. Available: `https://backend.icblcmc.org/assets/reports/Landmine-Monitors/LMM2023/Downloads/Landmine-Monitor-2023_web.pdf`
58. In Ukraine, 128,000 km2 of land and 14,000 km2 of water area are contaminated with explosives. *Official website of Ministry of Defence of Ukraine.* Available: `https://www.mil.gov.ua/news/2024/10/05/128-000-kv-km-suhodolu-ta-14-000-kv-km-akvatorii-ukraini-zabrudnen`
59. Dog works faster than person with metal detector. Rescue operations by

SES in Mykolaiv. *Hromadske.* Available: https://www.youtube.com/watch?v=HDz17-1yeIk

60. Two SES cadets killed in an explosion in Kharkiv region: what is known. *Suspilne.* Available: https://suspilne.media/kharkiv/493801-dvoe-kursantiv-dsns-zaginuli-pid-cas-vibuhu-na-harkivsini-s
61. I. Osmolovska, "Walking on Fire: Demining in Ukraine," *GLOBSEC*, 2023. Available: https://www.globsec.org/sites/default/files/2023-04/Demining%20in%20Ukraine%20report%20ver5%20web.pdf
62. Interactive map "Informing about the dangers of IEDs and risk prevention education". Available: https://ua.imsma.org/portal/apps/webappviewer/index.html?id=814d2770d197474d8ad7e8014c0a275e&locale=uk
63. Russia producing three times more artillery shells than US and Europe for Ukraine, *CNN.* Available: https://www.cnn.com/2024/03/10/politics/russia-artillery-shell-production-us-europe-ukraine/index.html
64. "Thousands of tons of ammunition in the ground": how long will demining take and how much does it cost? *Radio Free Europe/Radio Liberty (RFE/RL),* Available: https://www.radiosvoboda.org/a/novyny-pryazovya-khersonshchyna-skilky-koshtuye-rozminuvannya/32731054.html
65. Ukrainian Armed Forces dig up five mines per square meter – Reznikov, *Radio Free Europe/Radio Liberty (RFE/RL).* Available: https://www.radiosvoboda.org/a/news-reznikov-minuvannya-rosiya-rozminuvannya-zsu/32546226.html
66. M. Dewing, W. Koerner and Political and Social Affairs Division, "LAND MINES," *Government of Canada Publications*, 1996. Available: https:

//publications.gc.ca/Pilot/LoPBdP/MR/mr141-e.htm

67. Z. Kovács and I. Ember, "Landmine Detection with Drones," *Land Forces Acad. Rev.*, vol. 27, pp. 84–92, 2022, doi: 10.2478/raft-2022-0012
68. O. Cherednychenko, N. Palamarchuk, O. Shemendiuk and V. Martynyuk, "Synthesis of the system for detection of explosive objects on the base of an unmanned aerial vehicle," *Commun., Inf. Cybersecurity Syst. Technol.*, vol. 3, 2023, doi: 10.58254/viti.3.2023.18.163
69. C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, p. 60, 2019, doi: 10.1186/s40537-019-0197-0
70. M. Xu, S. Yoon, A. Fuentes and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognit.*, vol. 137, 2023, doi: 10.1016/j.patcog.2023.109347
71. P. Kaur, B. S. Khehra and E. B. S. Mavi, "Data Augmentation for Object Detection: A Review," in *Proc. 2021 IEEE Int. Midwest Symp. Circuits Syst. (MWSCAS)*, pp. 537–543, 2021, doi: 10.1109/MWSCAS47672.2021.9531849
72. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, doi: 10.1109/cvpr.2016.91
73. G. Jocher, A. Chaurasia and J. Qiu, "Ultralytics YOLOv8," Version 8.0.0, 2023. Available: https://github.com/ultralytics/ultralytics
74. J. Solawetz and Francesco, "What is YOLOv8? A Complete Guide," *Roboflow Blog*, Oct. 23, 2024. Available: https://blog.roboflow.com/what-is-yolov8/
75. G. Jocher and J. Qiu, "Ultralytics YOLO11," Version 11.0.0, 2024. Available: https://github.com/ultralytics/ultralytics
76. J. Terven, D.-M. Córdova-Esparza and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From

YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, pp. 1680–1716, 2023, doi: 10.3390/make5040083

77. J. Gallagher, "What is YOLOv11? An Introduction," *Roboflow Blog*, 2024. Available: `https://blog.roboflow.com/what-is-yolo11/`
78. R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," 2024, arXiv: 2410.17725, doi: 10.48550/arXiv.2410.17725
79. K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *Computer Vision – ECCV 2014*, Springer International Publishing, pp. 346–361, 2014, doi: 10.1007/978-3-319-10578-9_23
80. T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," *arXiv*, 2014, arxiv: 1405.0312, doi: 10.48550/arXiv.1405.0312
81. W. Lv, Y. Zhao, Q. Chang, K. Huang, G. Wang and Y. Liu, "RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer," *arXiv*, 2024, arxiv: 2407.17140, doi: 10.48550/arXiv.2407.17140
82. M. Tan, R. Pang and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *Proc. 2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, doi: 10.1109/cvpr42600.2020.01079
83. J.-a. Kim, J.-Y. Sung and S.-h. Park, "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," *2020 IEEE International Conference on Consumer Electronics – Asia (ICCE-Asia)*, pp. 1-4, 2020, doi: 10.1109/ICCE-Asia49877.2020.9277040.
84. N. Jegham, C. Y. Koh, M. Abdelatti and A. Hendawi. "Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors," *arXiv*, 2024, arxiv: 2411.00201, doi: 10.48550/arXiv.2411.00201

85. Ultralytics YOLO Docs. Available: https://docs.ultralytics.com
86. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," *AAAI,* vol. 34, no. 07, pp. 12993-13000, Apr. 2020, doi: 10.1609/aaai.v34i07.6999
87. X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang and J. Yang, "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," in *Advances in Neural Information Processing Systems,* vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin, Eds. Curran Associates, Inc., pp. 21002–21012, 2020. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/f0bda020d2470f2e74990a07a607ebd9-Paper.pdf
88. S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe and Y. Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features," 2019, arxiv: 1905.04899, doi: 10.48550/arXiv.1905.04899
89. H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," 2017, arxiv: 1710.09412, doi: 10.48550/arXiv.1710.09412
90. C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw and D. Rueckert, "GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks," 2018, arxiv: 1810.10863, doi: 10.48550/arXiv.1810.10863
91. L. Ma, J. Meng, S. Liu, W. Chen, J. Xu and R. Chen, "Sim2Real2: Actively Building Explicit Physics Model for Precise Articulated Object Manipulation," 2023, arxiv: 2302.10693, doi: 10.48550/arXiv.2302.10693
92. O. Kunichik, "Findmine Dataset," *Roboflow Universe*, 2024. Available: https://universe.roboflow.com/oleksandr-kunichik-sugbr/findmine
93. B. Dwyer, J. Nelson, J. Solawetz et al., "Roboflow," Version 1.0, 2022. Available: https://roboflow.com

94. G. Jocher, "Ultralytics YOLOv5," Version 7.0, 2020. Available: https://github.com/ultralytics/yolov5
95. python-telegram-bot. Available: https://python-telegram-bot.org
96. O. L. Didur and M. S. Shevchenko, "LANDMINES: those used or may be used by the troops of the Russian invaders in the land theater of operations," (in Ukrainian), 2023. Available: https://shron1.chtyvo.org.ua/Didur_Oleksandr/Miny_iaki_vykorystovuiutsia_abo_mozhut_vykorystovuvatysia_viiskamy_rosiiskykh_zaharbnykiv_na_sukhopu.pdf?PHPSESSID=7r7ecak3135fa9kap9uoqltok6
97. Anti-Personnel Landmines Convention. Available: https://disarmament.unoda.org/anti-personnel-landmines-convention
98. Landmine Use in Ukraine, Human Rights Watch. Available: https://www.hrw.org/news/2023/06/13/landmine-use-ukraine
99. Blender Online Community. *Blender – a 3D modelling and rendering package*. Stichting Blender Foundation, Amsterdam, 2018. Available: http://www.blender.org
100. GNU Image Manipulation Program. Available: https://www.gimp.org/about
101. Prusa Research. Available: https://www.prusa3d.com/page/about-us_77
102. mussy. *PMN AP Mine (Historical Prop).* (2018). Thingiverse. Accessed: Feb. 6, 2025. [Online]. Available: https://www.thingiverse.com/thing:3157971
103. Håkon Benjaminsen. *PMN-2.* (2023). Thingiverse. Accessed: Feb. 6, 2025. [Online]. Available: https://www.thingiverse.com/thing:5777217
104. Jonathan Lavoie. *PMN-2 Landmine.* (2020). MyMiniFactory. Accessed: Feb. 6, 2025. [Online]. Available: https://mmf.io/o/98395

105. valterjherson1. *OZM-72 Anti-personnel Mine Gameready Lowpoly* (2023). Sketchfab. Accessed: Sep. 15, 2023. [Online]. Available: (Link inactive as of Feb. 6, 2025).

106. mussy. *MON-50 (Historical Prop/ EOD Training).* (2023). Thingiverse. Accessed: Feb. 6, 2025. [Online]. Available: `https://www.thingiverse.com/thing:6139785`

107. mussy. *PFM-1 AP Mine (Historical Prop).* (2019). Thingiverse. Accessed: Feb. 6, 2025. [Online]. Available: `https://www.thingiverse.com/thing:3660586`

108. R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proc. 2014 IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, pp. 580-587, 2014, doi: 10.1109/CVPR.2014.81

109. R. Girshick, "Fast R-CNN," in *Proc. 2015 IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169

110. S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137-1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031

111. K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *Proc. 2017 IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322

112. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Comput. Vis. – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer Int. Publishing, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2

113. D. Reis, J. Kupec, J. Hong and A. Daoudi, "Real-Time Fly-

ing Object Detection with YOLOv8," 2023, arXiv: 2305.09972, doi: 10.48550/arXiv.2305.09972

114. J. R. Waite, J. Feng, R. Tavassoli, L. Harris, S. Y. Tan, S. Chakraborty and S. Sarkar, "Active shooter detection and robust tracking utilizing supplemental synthetic data," 2023, arXiv: 2309.03381, doi: 10.48550/arXiv.2309.03381
115. H. Lou, H. Liu, L. Bi, L. Liu, J. Guo and J. Gu, "Bd-Yolo: Detection Algorithm for High-Resolution Remote Sensing Images," 2023, doi: 10.2139/ssrn.4542996
116. M. Kang, C.-M. Ting, F. F. Ting and R. C.-W. Phan, "RCS-YOLO: A Fast and High-Accuracy Object Detector for Brain Tumor Detection," in *Med. Image Comput. Comput. Assist. Interv. – MICCAI 2023*, 2023, pp. 600–610, doi: 10.1007/978-3-031-43901-8_57
117. G. J. N. Ang, A. K. Goil, H. Chan, J. J. Lew, X. C. Lee, R. B. A. Mustaff et al., "A Novel real-time arrhythmia detection model using YOLOv8," 2023, arXiv: 2305.16727, doi: 10.48550/arXiv.2305.16727
118. V. Agarwal, A. G. Pichappa, M. Ramisetty, B. Murugan and M. K. Rajagopal, "Suspicious Vehicle Detection Using Licence Plate Detection and Facial Feature Recognition," 2023, arXiv: 2304.14507, doi: 10.48550/arXiv.2304.14507
119. LabelImg. Available: `https://github.com/HumanSignal/labelImg`
120. Makesense.ai. Available: `https://www.makesense.ai/`
121. Toras: Toronto Annotation Suite. Available: `https://aidemos.cs.toronto.edu/toras`
122. Google Cloud Platform. Available: `https://cloud.google.com/`
123. Microsoft Azure. Available: `https://azure.microsoft.com/`
124. Amazon SageMaker. Available: `https://aws.amazon.com/sagemaker/`
125. F. Zhou, H. Deng, Q. Xu and X. Lan, "CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote

Sensing Images," *Electronics*, vol. 12, no. 12, p. 2671, 2023, doi: 10.3390/electronics12122671

126. E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. 2011 Int. Conf. Comput. Vis.*, 2011, doi: 10.1109/iccv.2011.6126544
127. P. Alcantarilla, J. Nuevo and A. Bartoli, "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces," in *Proc. British Mach. Vis. Conf. 2013*, 2013, pp. 13.1–13.11, doi: 10.5244/c.27.13
128. X. Zhu, W. Su, L. Lu, B. Li, X. Wang and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," 2020, arxiv: 2010.04159, doi: 10.48550/arXiv.2010.04159
129. CAT-UXO. Available: `https://cat-uxo.com`
130. Qt C++ Framework. Available: `https://www.qt.io`
131. ONNX Runtime. Available: `https://onnxruntime.ai`
132. Mobile Operating System Market Share Worldwide for 2025 year. Statcounter Global Stats. Available: `https://gs.statcounter.com/os-market-share/mobile/worldwide/2025`
133. Secure Sockets Layer (SSL). Available: `https://openssl.org`

LIST OF FIGURES

2.1. The territory of Ukraine contaminated by EO. Source https://ua.imsma.org [62] 72
2.2. Printed battle EO: The inner portion contains explosive material, while the interlayer between the outer shell is filled with bolts, serving as shrapnel 79
2.3. Architecture of the YOLOv8 model. Further insights into the specifics of this architecture can be found in the detailed guide on the Roboflow Blog [74]. Data sourced from [73], visualization by RangeKing on GitHub 81
2.4. Simplified architecture of the YOLOv11 model, highlighting key components: SPPF, C2PSA, and C3K2. Data sourced from [78] 83
2.5. Performance comparison of different YOLO versions (v5, v8, v11), RTDETRv2, and EfficientDet on the COCO object detection benchmark dataset, measured using the mAP50-95 metric. Data sourced from [85] 83
2.6. Confusion matrix 88
2.7. Ground truth (blue) and predicted (green) bounding boxes. IoU is calculated from their intersection and union areas 90

3.1. Grayscale Augmentation 96
3.2. Example of Rotation 97
3.3. Example of Noise 98
3.4. Shear augmentation 100
3.5. Mosaic of mix-ups 102
3.6. Examples of images with different numbers of annotations . . . 110

3.7. Annotation Heat Map 111
3.8. Performance metrics of the YOLOv5 model trained with two-stage augmentation for 150 epochs 122
3.9. Performance metrics of the YOLOv8 model trained with two-stage augmentation 123
3.10. Evolution of Box Loss, and Class Loss, Fitness, and Recall during YOLOv8 training 123

4.1. Examples of 3D-printed landmine replicas scaled down in size to evaluate the feasibility of using modified objects for model training 130
4.2. Examples of 3D-printed landmine replicas originally insize . . . 134
4.3. Images in the dataset: *a* – OZM-72 on a snowy backdrop; *b* – PMN-2 on a snowy surface; *c* – PMN-2 on a layer of leaves above an asphalt surface; *d* – PFM-1, camouflaged to blend with the surrounding stones; *e* – MON-50 positioned on a tree in a monochrome depiction; *f* – a monochrome representation of a PMN landmine in a foliage adjacent to a fallen tree trunk; *g* – PMN in a creek; *h* – PMN-2 surrounded by apples; *i* – PFM-1 in a trashed environment 140
4.4. Evolution of precision, recall, mAP50, and mAP50-95 during the training of the model over 100 epochs (Experiment 8.1, Table 4.5) 149
4.5. Normalized confusion matrix for the model on the validation dataset of 3D-printed landmines (Experiment 8.1). Class labels: 0 – PFM-1; 2 – PMN; 14 – MON-50; 16 – PMN-2; 19 – OZM-72 149
4.6. Examples of successful landmine replica detections by the model on images not included in the training dataset 151
4.7. Illustrations of EO identification on photographs of actual landmines 153
4.8. Normalized confusion matrix for the dataset from actual EO . . 155

5.1. System architecture of the cloud-based EO detection service . . 167
5.2. Google Cloud architecture with a gateway to Google Cloud Functions (Cloud Run) as an initial entry point, Google Storage for settings, and Roboflow API which hosts models 171
5.3. Screenshots of the messenger bot interface. The source of the images: Forester 174

6.1. Screenshot of the Qt Creator 13.0.2 IDE, showing the *mineeye* project open, with the editor displaying the code for QML and the left panel showing the project structure 184
6.2. Interface of the Google Cloud Run function 187
6.3. Upper-level architecture of the app 189
6.4. Main components of the GCP architecture in the system 190
6.5. Operational algorithm for image processing 193
6.6. Data flow diagram of the semi-automatic annotation process . . 196
6.7. New user registration dialog, requiring a referral link 198
6.8. Settings menu, allowing users to select the operating mode . . . 199
6.9. Screenshots of the application's user interface: *a* – file selection dialog; *b* – image capture mode, flash enabled; *c* – editing mode, multiple files open 200
6.10. Two nearly identical frames from a video depicting the detection of an OZM-72 landmine. The landmine on the left was not detected by the on-device model, highlighting the need for further model refinement. Source: Telegram channel of a military officer with the call sign Forester 202
6.11. Screenshots of the application's edit mode: *a* – navigating the image using zoom and pan gestures; *b* – selecting an EO by tracing its outline; *c* – choosing the correct EO type from a dropdown menu 204

6.12. Screenshot of the data transfer menu: On the left is the main settings menu; on the right is the submenu for sending and receiving data 205

7.1. Request for a new landmine 211
7.2. Example of ffmpeg output 213
7.3. Roboflow's annotation interface. The source of the image: Forester 214
7.4. Samples from the dataset without and with mosaic applied . . . 215
7.5. Samples from the dataset using 1.96% noise 216
7.6. Metrics for YOLOv11 training 217
7.7. Confusion matrix for YOLOv11 algorithm. Class 0 – PFM-1; 14 – MON-50; 16 – PMN-2; 19 – OZM-72; 2 – PMN; 63 – M56; 64 – 9H24 218
7.8. Edit configuration file for access the new model 219
7.9. Examples of recognized landmines M56 and 9H24. The source of the images: Forester 221

LIST OF TABLES

3.1. Initial Dataset Composition: Landmine Types and Image Counts 109
3.2. Results of experiments with preprocessing 112
3.3. YOLO Augmentation Methods 115
3.4. Comparison of Augmentation Techniques 120
3.5. Augmentation Types and Variations 120

4.1. Characteristics of Selected Landmine Models 134
4.2. Examples of Printed Models 136
4.3. Roboflow Augmentations and Their Purpose 143
4.4. Default YOLOv8 Augmentation Parameters and Their Descriptions 145
4.5. Comparison of the Results of Different Experiments 145
4.6. Results of Model Testing with New Data Types 152
4.7. Results of Testing the Model on Real Landmines 154

6.1. Hyperparameters used for training YOLOv8 192
6.2. Performance of the On-Device EO Detection Model 201

7.1. Results of training on 200 epochs without and with mosaic . . . 215
7.2. Hyperparameters Used for Training YOLOv11 217
7.3. Performance of the EO Detection Model 220

ACKNOWLEDGEMENTS

I gratefully acknowledge the invaluable assistance of the Ukrainian Armed Forces members, including Lisnyk (Forester), S.T.A.L.K.E.R., Vitaliy, and Artur, for their contributions to this research. Special recognition is extended to Artur, who was the first to assist. I also extend my sincere gratitude to Andriy Mukhin for his valuable support. I would also like to express my sincere gratitude to Nina Liu of the Toronto Public Library for her invaluable assistance with the 3D printing technique, and to Yaroslav Tereshchenko for his insightful consultations at the start of this research. Furthermore, heartfelt thanks are extended to my academic supervisor, Vasyl Tereshchenko, for their unwavering belief in me and steadfast support throughout my studies. Their mentorship and dedication were instrumental in the successful completion of this dissertation. Additionally, I would like to thank the "Perevirena Bavovna" community for their support. Finally, I would like to express my deepest gratitude to my parents, sister, and wife for their boundless love, unwavering support, and constant encouragement throughout my life and academic journey.

APPENDIX A. EXTENDED MODEL INFORMATION AND LANDMINE DATASET

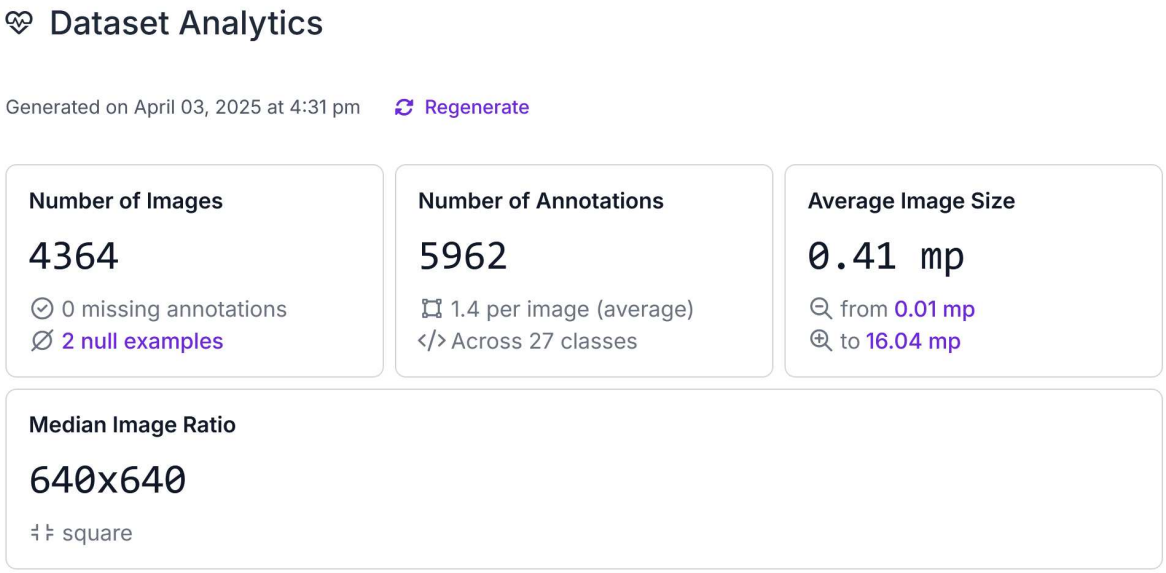


Figure: Sizes of the objects

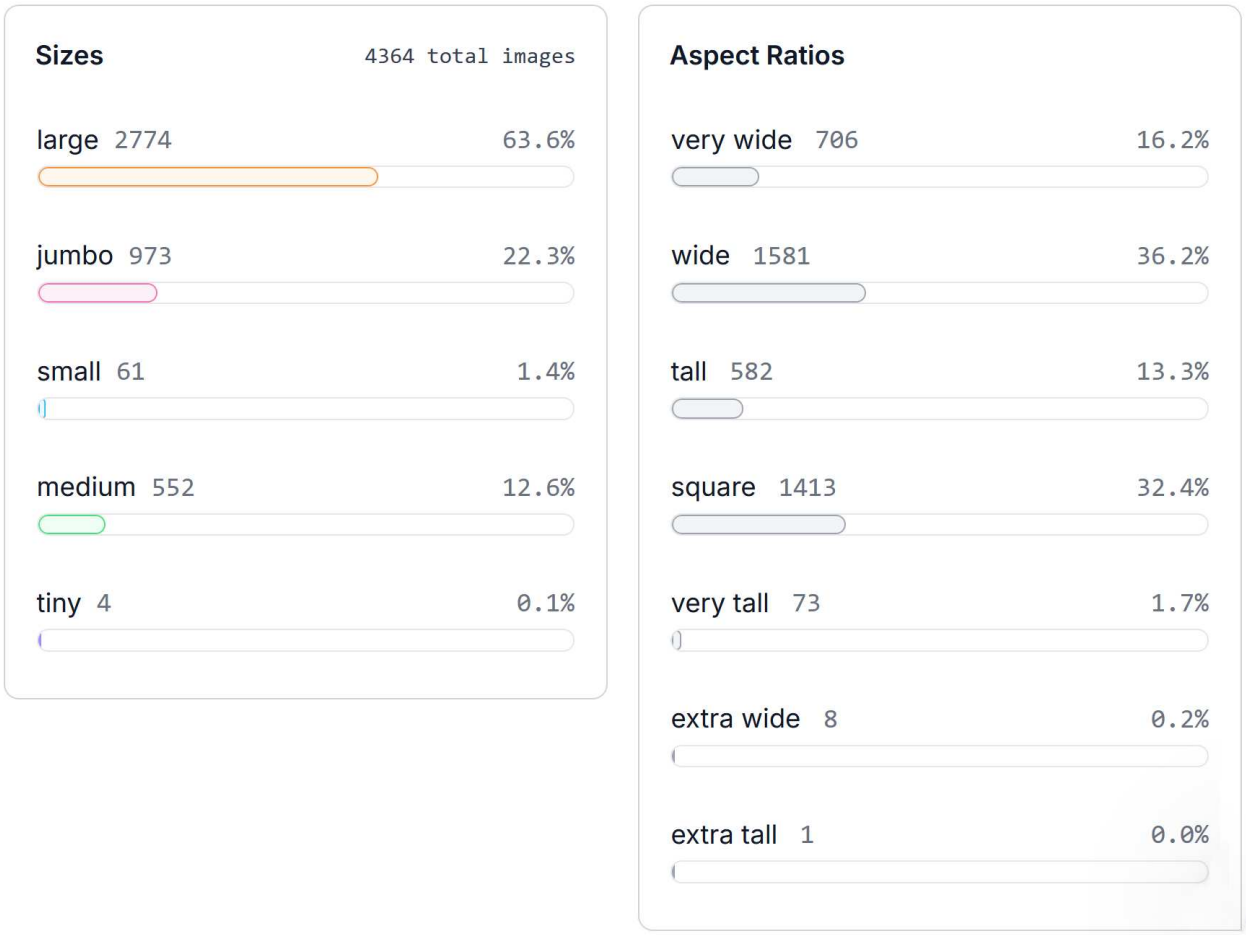


Figure: Extended dataset statistics

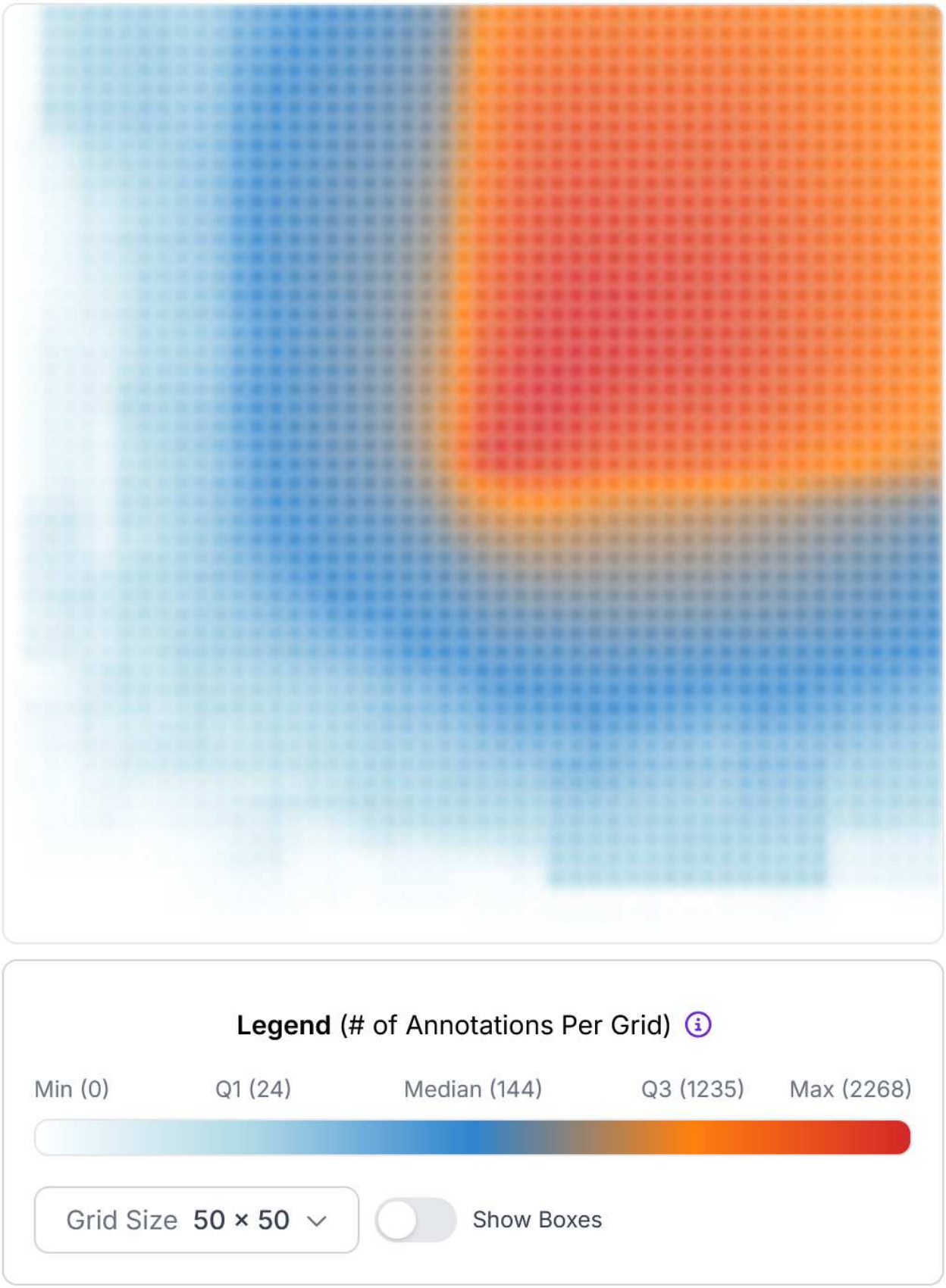


Figure: The heatmap

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure: Images in the dataset: *a* – PFM-1 in a creek; *b* – PMN-2 under the leaves; *c* – PMN-2 on a railway; *d* – OZM-72 on a snowy surface; *e* – PMN-1 near the tree stump; *f* – MON-50 on a fence; *g* – MON-50 on a tree; *h* – OZM-72 in a grass; *i* – PFM-1 in a foliage

APPENDIX B. LIST OF THE APPLICANT'S PUBLICATIONS ON THE APPROVAL OF THE RESULTS OF THE DISSERTATION ARTICLES IN PROFESSIONAL PUBLICATIONS OF UKRAINE (INCLUDED IN THE LIST OF THE MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE)

1. Kunichik, O., & Tereshchenko, V. (2022). Analysis of modern methods of search and classification of explosive objects. *Artificial Intelligence and Intelligent Systems,* 27(2), 52–59. Available: https://doi.org/10.15407/jai2022.02.052
2. Kunichik, O., & Tereshchenko, V. (2023). Improving the accuracy of landmine detection using data augmentation: a comprehensive study. *Artificial Intelligence and Intelligent Systems,* 28(2), 42–54. Available: https://doi.org/10.15407/jai2023.02.042
3. Kunichik, O. (2024). Review of strategies to overcome the lack of data in landmine detection. *Artificial Intelligence and Intelligent Systems,* 29 (3), 99–103. Available: https://doi.org/10.15407/jai2024.03.099
4. Kunichik, O., & Tereshchenko, V. (2024). Determining the effectiveness of using three-dimensional printing to train computer vision systems for landmine detection. *Eastern-European Journal of Enterprise Technologies,* 5, 1(131), 17–29. Available: https://doi.org/10.15587/1729-4061.2024.311602
5. Kunichik, O. (2024). Landmine detection with a mobile application. *Eastern-European Journal of Enterprise Technologies,* 6, 2(132), 6–13. Available: https://doi.org/10.15587/1729-4061.2024.317103

Approbation works

1. Kunichik, O., & Tereshchenko, V. (2022, December 8–9). *Analysis of modern methods of search and classification of explosive objects.* The 22nd International Scientific and Technical Conference "Artificial Intelligence and Intellectual Systems" (AIIS'2022), Kyiv, Ukraine.
2. Kunichik, O. (2023, April 15). *Detection of explosive objects* [Conference presentation abstract]. The 21st International Scientific and Practical Conference "Shevchenko Spring–2023," Kyiv, Ukraine, 92–93. Available: https://probability.knu.ua/shv2023/ShV_2023.pdf
3. Kunichik, O., & Tereshchenko, V. (2023, November 21). *Enhancing landmine detection through the integration of 3D printing and deep learning techniques* [Conference presentation abstract]. Tenth International Conference Information Technologies and Implementation (Satellite), Kyiv, Ukraine, 28–29. Available: http://iti.fit.univ.kiev.ua/wp-content/uploads/%D0%97%D0%B1%D1%96%D1%80%D0%BA%D0%B0_ITI_2023.pdf
4. Kunichik, O., & Tereshchenko, V. (2024, March 15–16). *Advancing landmine detection: a methodological breakthrough using 3d printing and computer vision* [Conference presentation abstract]. Artificial intelligence: achievements, challenges, risks, Kyiv, Ukraine, 447-450. Available: https://www.ipai.net.ua/docs/ai_conf_15.03.2024.pdf
5. Kunichik, O. (2024, October 18–19). *Using a Telegram bot to detect landmines with artificial intelligence* [Conference presentation abstract]. Twenty-fourth international scientific and technical conference artificial intelligence and intelligent systems AIIS'2024, Kyiv, Ukraine, 216–220. Available: https://www.ipai.net.ua/docs/AIIS-2024.pdf
6. Kunichik, O. (2024, November 21–22). *Mobile application for landmine detection* [Conference presentation abstract]. The 11th All-Ukrainian Scientific and Practical Conference of Students, Postgraduates and Young

Scientists "United by Science: Prospects of Interdisciplinary Research," Kyiv, Ukraine, 311–313.

7. Kunichik, O. (2024, December 11–12). *Cloud-based landmine detection service with messenger bot integration* [Conference presentation abstract]. The 23rd International Scientific Conference "Neural Network Technologies and Applications NNTA-2024," Kramatorsk – Vinnytsia – Ternopil Ukraine, 216–220.