

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

*На правах*

*рукопису*

УДК \_\_\_\_\_

**ВИПУСКНА КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

Тема: «Розпізнавання емоцій обличчя»

Спеціальність 121 «Інженерія програмного забезпечення»

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

Студент: Товстиженко Юрій Олександрович

групи ІПЗм-21

\_\_\_\_\_

(підпис, дата)

Науковий керівник:

Ковалюк Тетяна Володимирівна к.т.н,

доцент

— \_\_\_\_\_

(підпис, дата)

Допускається до захисту з питань  
нормоконтролю

Завідувач кафедри Бичков О. С.,

доктор технічних наук, професор

\_\_\_\_\_

(підпис, дата)

Київ – 2022

Рішенням Екзаменаційної комісії  
випускна кваліфікаційна робота студента

Товстиженко Юрія Олекса́ндровича

захищена з оцінкою

---

Голова Екзаменаційної комісії

д.т.н. проф. Бондарчук Андрій

---

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ:

Завідувач кафедри програмних систем і технологій

\_\_\_\_\_ (О.С.Бичков )

«\_\_\_» \_\_\_\_\_ 2022 р.

## ЗАВДАННЯ

### НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ

### СТУДЕНТУ

Товстиженко Юрія Олександровича

1. Тема випускної кваліфікаційної магістерської роботи: Розпізнавання емоцій обличчя затверджені наказом вищого навчального закладу від «\_\_\_» \_\_\_\_\_ 2022 р. № \_\_\_\_\_
2. Строк здачі студентом закінченої роботи «\_\_\_» \_\_\_\_\_ 2022 р.
3. Вихідні дані до роботи:
  - Вхідний набір даних з більше, ніж 35000 зображень
  - обличчів сімома емоціями; модель нейромережі Tensorflow;
  - метод оптимізації Адама
  - мова програмування Python та відповідні бібліотеки

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити): дослідження предметної області; огляд наукової літератури; визначення вимог та завдань; опис архітектури побудови нейромережі, датасетів, технології розробки та їх обґрунтування, інструментів
5. Перелік графічного матеріалу (з точним забезпеченням обов'язкових креслень): блок-схеми, UML-діаграми (Use case та Sequence)
6. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
I.	Ковалюк Т.В.		Товстиженко Ю.О.
II.	Ковалюк Т.В.		Товстиженко Ю.О.
III.	Ковалюк Т.В.		Товстиженко Ю.О.

7. Дата видачі завдання «\_\_» \_\_\_\_\_ 2022 р.

Керівник \_\_\_\_\_ Ковалюк Т.В.  
(підпис, дата)

Завдання прийняв до виконання \_\_\_\_\_ Товстиженко Ю.О.  
(підпис, дата)

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Затвердження теми роботи	15.12.2021- 25.12.2021	<b>Виконано</b>
2.	Розгляд та вивчення теоретичних відомостей стосовно обраної тематики	26.12.2021- 14.01.2022	<b>Виконано</b>
3.	Огляд методів та класифікацій нейронних мереж	15.01.2022- 30.01.2022	<b>Виконано</b>
4.	Огляд технологій та інструментів розробки ПЗ	31.01.2022- 12.02.2022	<b>Виконано</b>
5.	Програмна реалізація нейронних мереж	13.02.2022- 30.03.2022	<b>Виконано</b>
6.	Аналітична робота, висновки	31.03.2022- 20.04.2022	<b>Виконано</b>
7.	Оформлення пояснювальної записки, створення SAD	21.04.2022- 07.05.2022	<b>Виконано</b>
8.	Передзахист	26.05.2022	<b>Виконано</b>
9.	Захист	26.05.2022	<b>Виконано</b>

Студент-магістр

\_\_\_\_\_ Товстиженко Ю.О.  
(підпис, дата)

Керівник роботи

\_\_\_\_\_ Ковалюк Т.В.  
(підпис, дата)

## ЗМІСТ

<b>ВСТУП</b> .....	10
<b>РОЗДІЛ I</b> .....	13
<b>ТЕОРЕТИЧНІ ПРИЧИНИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ДЛЯ РОЗПІЗНАВАННЯ ЕМОЦІЙ ОБЛИЧЧЯ</b> .....	13
<b>1.1. Розпізнавання емоцій обличчя: основна характеристика та методи</b> .....	13
<b>1.2. Огляд ринку конкурентів. Програмні забезпечення для розпізнавання емоцій обличчя FaceReader, eMotion Software, MMER_FEASy</b> .....	19
Висновки до першого розділу .....	26
<b>РОЗДІЛ II</b> .....	27
<b>МЕТОДИ ПОБУДОВИ АРХІТЕКТУРИ НЕЙРОННИХ МЕРЕЖ</b> .....	27
<b>2.1. Нейрона мережа: функції, переваги та алгоритм роботи</b> .....	27
<b>2.2. Нейронна мережа CNN та її архітектура</b> .....	36
2.2.1. Використана методологія.....	42
Висновки до другого розділу .....	44
<b>РОЗДІЛ III</b> .....	45
<b>РЕАЛІЗАЦІЯ ПЗ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ</b> .....	45
<b>3.1. Вибір технологій розробки та її обґрунтування</b> .....	45
3.1.1. Вибір платформи для розробки .....	45
3.1.2. Вибір мови програмування. ....	45
3.1.3. Вибір бібліотек Python .....	46
<b>3.2. Моделювання системи використовуючи UML діаграми</b> .....	46
<b>3.3. Модель та реалізація нейронної мережі</b> .....	52
3.3.1. Розпізнавання емоцій обличчя.....	52
Висновки до третього розділу .....	59
<b>ВИСНОВКИ</b> .....	60
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	61
<b>ДОДАТКИ</b> .....	63
Додаток А .....	63

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ - програмне забезпечення

ОС - операційна система

БД - база даних

CNN (Convolutional Neural Network) - згорткова нейронна мережа

ML (Machine Learning) – машинне навчання

AI (Artificial Intelligence) - штучний інтелект

NN (Neural Network) - нейронна мережа

CV(Computer Vision) - комп'ютерний зір

RNN (Recurrent Neural Network) - рекурентна нейронна мережа

## АНОТАЦІЯ

**Випускна кваліфікаційна магістерська робота** містить 72 сторінки, 30 рисунки та 12 джерел використаних в роботі.

**Тема** магістерської роботи: Програмне забезпечення розпізнавання емоцій обличчя.

**Об'єкт дослідження:** емоції обличчя, які розпізнаються за допомогою згорткових нейронних мереж

**Предмет дослідження:** методи та алгоритми згорткових нейронних мереж для розпізнавання емоцій обличчя

**Мета дослідження:** покращення результатів та підвищення достовірності розпізнавання емоцій обличчя з використанням згорткових нейронних мереж та спростити розробку CNN, використати методи оптимізації моделей, що допомогло б отримати найбільшу точність та якість тренування.

**Результатом дослідження:** є програмне забезпечення для розпізнавання емоцій обличчя. Реалізація програмного забезпечення була впроваджена за допомогою згорткової нейронної мережі. Проектування взаємодії користувача та системи було проілюстровано за допомогою UML-діаграм.

**Ключові слова:** згорткова нейронна мережа, емоції обличчя, датасет

## ANNOTATION

**The final qualifying master's thesis contains** 72 pages, 30 drawings and 12 sources used in the work.

**Thesis topic** of master's work: Software for recognition of facial emotions.

**The object** of the thesis is the recognition of facial emotions with the help of information technologies.

**The subject** of the thesis is the features of recognition of facial emotions with the use of convolutional neural networks

**The aim** of the work is to recognition of facial emotions using convolutional neural networks

**The result** is a software for recognizing facial emotions. The implementation of the software was implemented with the help of a convolutional neural network. The design of the interaction between the user and the system was illustrated with the help of UML-diagrams.

**Keywords:** convolutional neural network, facial emotions, dataset

## ВСТУП

За останні десятиліття використання машин у суспільстві значно зросло. У наш час машини використовуються в багатьох галузях промисловості. Зі збільшенням контакту з людьми взаємодія також має стати більш гладкою та природною. Щоб досягти цього, машини мають бути забезпечені можливостями, які дозволяють їм розуміти навколишнє середовище. Зокрема, наміри людини. Коли йдеться про машини, цей термін включає комп'ютери та роботів. Різниця між ними полягає в тому, що роботи включають можливості взаємодії в більш просунутому обсязі, оскільки їхня конструкція передбачає певний ступінь автономності.

Коли машини здатні цінувати своє оточення, розвивається якесь машинне сприйняття. Люди використовують свої органи чуття, щоб отримати уявлення про навколишнє середовище. Тому машинне сприйняття має на меті імітувати органи чуття людини, щоб взаємодіяти з навколишнім середовищем. Нині машини мають кілька способів фіксувати стан навколишнього середовища за допомогою камер і датчиків. Отже, використання цієї інформації з відповідними алгоритмами дозволяє сформувати машинне сприйняття. В останні роки використання алгоритмів глибокого навчання виявилось дуже успішним у цьому відношенні. Комп'ютери, навчені за допомогою методів глибокого навчання, змогли досягти деяких дивовижних завдань. Наприклад, ці завдання включають вміння вивчати китайську мову, розпізнавати об'єкти на зображеннях і допомагати в медичній діагностиці.

Афективні обчислення стверджують, що виявлення емоцій необхідно для того, щоб машини краще виконували свою мету. Наприклад, використання роботів у таких сферах, як догляд за літніми або в якості носіїв у лікарнях, вимагає глибокого розуміння навколишнього середовища. Емоції на обличчі дають інформацію про внутрішній стан суб'єкта. Якщо машина здатна отримати послідовність зображень обличчя, то використання методів глибокого

навчання допоможе машинам усвідомлювати настрій свого співрозмовника. У цьому контексті глибоке навчання може стати ключовим фактором для побудови кращої взаємодії між людьми і машинами, забезпечуючи машинам певну самосвідомість, що покращить її зв'язок з природним інтелектом.

Підсумовуючи все вищезазначене, в магістерській роботі буде зроблена розпізнавання емоцій обличчя через веб-камеру з використанням згорткових нейронних мереж.

**Об'єкт:** емоцій обличчя, які розпізнаються за допомогою ЗНН

**Предмет:** методи та алгоритми (інструменти) для розпізнавання емоцій обличчя з використанням згорткових нейронних мереж

**Мета:** покращити якість та збільшити достовірність розпізнавання емоцій обличчя з використанням згорткових нейронних мереж та спростити розробку CNN, використати методи оптимізації моделей, що допомогло б отримати найбільшу точність та якість тренування

**Завдання:**

1. Здійснити аналіз наукової літератури застосування сучасних технологій для розпізнавання емоцій обличчя
2. Розглянути методи побудови та принципи нейронних мереж
3. Визначити технології та інструменти та розробити програмне забезпечення для розпізнавання емоцій обличчя через веб-камеру.

Дослідити отримані результати розпізнавання емоцій обличчя за

4. допомогою аналітичних інструментів та визначити переваги запропонованих підходів

Для того, аби досягти поставлених завдань, були використані наступні методи, а саме:

- теоретичні (аналіз досліджуваного явища) надали можливість зробити узагальнення та науково-теоретичне обґрунтування;
- штучного інтелекту (машинне навчання, оптимізація архітектури штучних нейронних мереж)

## РОЗДІЛ І

### ТЕОРЕТИЧНІ ПРИЧИНИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ДЛЯ РОЗПІЗНАВАННЯ ЕМОЦІЙ ОБЛИЧЧЯ

#### 1.1. Розпізнавання емоцій обличчя: основна характеристика та методи

Системи машинного зору та розпізнавання зображень міцно закріпилися в сучасному світі. Основним завданням систем машинного зору є розпізнавання об'єктів, а також їх зовнішнього вигляду та розташування в просторі. Повноцінне спілкування між людьми неможливе без прояву та аналізу емоцій. Тому в сучасних людино-машинних системах часто доводиться використовувати методи розпізнавання емоцій. Одним з основних способів розпізнати емоції іншої людини є аналіз візуальної інформації. Тому автоматизація цього процесу, очевидно, повинна базуватися на використанні методів і засобів комп'ютерного зору. Завдання на розпізнавання емоцій можна використовувати в різних сферах людської діяльності. В даний час робототехніка активно розвивається. Основний напрям наукової роботи в робототехніці пов'язаний зі створенням інтелектуальних роботів. Розпізнавання емоцій людини роботами дозволяє підвищити ступінь інтелектуалізації їх взаємодії, наприклад, забезпечити правильну реакцію на стан людини. Особливо важливо правильно визначити стан людини при загрозових для життя випадках. Наприклад, в системі розпізнавання втомленості людини в автомобілі на основі аналізу зображень обличчя, отриманих з відеокамери. Іншою сферою застосування методів автоматичного розпізнавання емоцій є забезпечення безпеки людей за допомогою систем безпеки. Наприклад, забезпечення систем безпеки засобами виявлення та супроводу людей, які проявляють емоції агресії, страху. У сфері маркетингу можна використовувати відповідні системи для швидкого відстеження та реагування на різні проблеми в торгових центрах, супермаркетах та інших точках продажу товарів і послуг. Ефективність ряду цих маркетингових завдань можна значно підвищити за

рахунок автоматичного розпізнавання емоцій клієнтів. Розпізнавання емоцій також використовується в ряді інших областей, таких як телекомунікації, відеоігри, анімація, психіатрія, комп'ютерне навчання тощо. Хоча сфера розпізнавання емоцій є досить перспективною, на даний момент вона не дуже розвинена. Основною причиною цього є відсутність єдиних стандартів розробки алгоритмів, а також відсутність єдиних баз даних, створених для навчання алгоритмам розпізнавання емоцій. Робота алгоритмів розпізнавання емоцій полягає у складанні набору ознак, за якими вхідне зображення буде дорівнювати стандартному, і визначення його належності до певного класу, в даному випадку певної емоції. Емоції відображають суб'єктивне ставлення людини до різних явищ. Емоційні реакції — це специфічна реакція людини на ситуації, що зачіпають її інтереси. Є емоційні прояви, які неконтрольовані і впізнавані. При однаковій дії одних і тих же зовнішніх факторів різні люди можуть мати різні емоції. Емоції проявляються по-різному і в поєднанні один з одним. Розпізнавання емоцій за виразом обличчя - складний розумовий процес. Людина відносно легко розпізнає емоції обличчя, ходи, жести іншої людини. Однак це складне завдання для комп'ютерних систем. Нечітка логіка (FL) і системи встановлення прецедентів (CBR) є двома добре відомими методами для реалізації інтелектуальних систем класифікації. Кожен метод має свої переваги і недоліки. FL, наприклад, забезпечує інтуїтивно зрозумілий інтерфейс, спрощує процес отримання знань і мінімізує обчислювальну складність системи з точки зору використання часу та пам'яті. З іншого боку, у ПЛ виникають проблеми в процесі впровадження знань, що ускладнює прийняття розумної реалізації системи. CBR уникає цих проблем, використовуючи останні дані, які використовуються для пошуку результату системи для вхідних даних. Точність системи CBR зростає із збільшенням кількості прецедентів. Однак збільшення кількості випадків може означати додаткову складність обчислень. Впровадження гібридних систем, що містять FL і CBR для розпізнавання виразів обличчя, є ефективнішим і більш керованим, ніж окремі методи. Для реалізації механізму – емоцій на етапі

класифікації пропонується використовувати систему нечітких висновків, яка працює з вісьмома рисами обличчя. Кожен знак описується нечіткою множиною, яка характеризує зміну ознаки в емоціях.

Результуюча емоція семи основних (злість, огида, смуток, нейтральна, радість, здивування, страх) виходить на основі ознаки в результаті функціонування системи міркувань за прецедентом. Ми використовуємо шість основних емоцій як вираз обличчя. Щоб зменшити складність модуля вибору, ми використали спрощену схему зчитування ознак, використовуючи лише вісім основних елементів обличчя. Схема окремих зон регіонів усіх елементів обличчя. Вибрані області дають оцінки емоційного стану і мають значення на екрані (0 - 1). Назвіть це стан елемента обличчя. По-перше, формується автономна система СВР, щоб створити прецеденти там, де є елементи обличчя. Пізніше, використовуючи невизначену природу станів, ми використовували нечіткі множини для визначення обличчя. На третьому етапі, щоб забезпечити гнучкість, нечіткий набір правил розміщується в системі відкликання для абітурієнтів, яка може сама коригуватися шляхом навчання на основі досвіду.

Основний алгоритм автоматичного розпізнавання емоцій складається з наступних етапів:

1. Отримання зображення та первинна обробка зображення;
2. Виділення обличчя на зображенні;
3. Підбір елементів обличчя;
4. Вибір ключових моментів на елементах;
5. Класифікація емоцій.

Модуль введення в розділ – кольорові зображення, які необхідно вибрати з обличчя та його елементів. Перший крок - це розташування обличчя, у вхідному зображенні. Щоб виявити вилучення обличчя, вимийте пікселі кольору шкіри на зображеннях. Після виявлення обличчя в області обличчя

з'являються елементи. Усі основні елементи підібрані для визначення емоційного виразу: брови, щоки, губи та підборіддя, лоб, ніс, зуби, очі. Модуль попередньої обробки зображень і вибору функцій призначає значення кожному елементу з попередніми прецедентами в базі даних.

Для обчислення подібності між двома прецедентами визначається міра відстані  $D[1]$ , яка є функцією наступних трьох факторів: індекс ефективності  $e_i$ ; частота  $f_i$ , відмінності у значенні елемента у вхідному зображенні та в базі прецедентів

де  $\alpha$  і  $\beta$  — нормовані константи,  $K_1, \dots, K_8$  — контрольні вагові коефіцієнти, які визначають відносну важливість восьми елементів. Вони специфічні для профілю кожної людини. Кожна людина має якісь особливості, які є особливими, тобто змінює свій внесок у кінцевий висновок.

$$D(U_i, U^*) = \frac{\alpha \sum_{j=1}^8 \left| \beta K_j \left[ \frac{\Delta V_{ij}}{V_{ij}} \right] \right|}{e_i f_i \sum_{j=1}^8 K_j}$$

Система введення складається із значень елементів, показаних на малюнку 1.1, значення елемента об'єднується, щоб належати вхідним функціям:  $\mu_{X_i}(V): V_i \rightarrow [0, 1]$ , де  $x_i \in T_i$  з  $T_i$  - задане умови для функції, що відповідає  $V_i$ . Терміни  $T_i$  для різних значень і наведені нижче:

Лоб - T3 = {Вниз і маленький, нормальний, розтягнутий і більший}.

Щоки - T4 = {Плоскі та витягнуті, нормальні, повні та підняті}.

Брови - T2 = {центр, нормальний, витягнутий назовні}.

Губи - T6 = {Притиснуті\_закриті, нормальні, відкриті}.

Зуби - T7 = {невидимі, трохи назовні, екстра розкриті}.

Підборіддя - T8 = {нормальний, радикальний}.

Ніс - T5 = {нормальний, радикальний}.

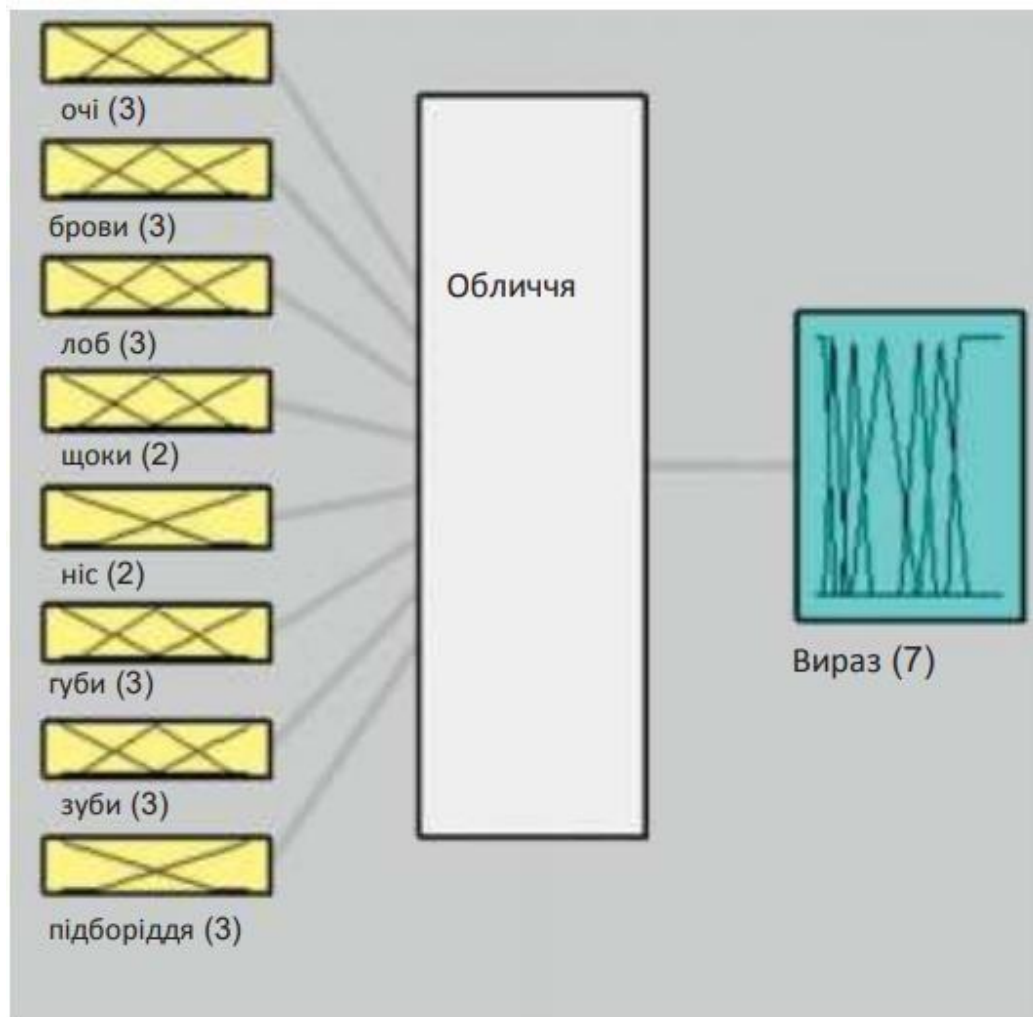


Рисунок 1.1 - Нечіткий висновок для визначення виразу.

Функції належності вихідного виразу обличчя представлені як  $\mu_o$  (вираз), де  $O = \{\text{Смуток, Нормальний, Гнів, Відраза, Радість, Здивування, Страх}\}$ . На основі представленого була розроблена структура системи визначення виразу обличчя, яка зображена на рисунку 1.2.

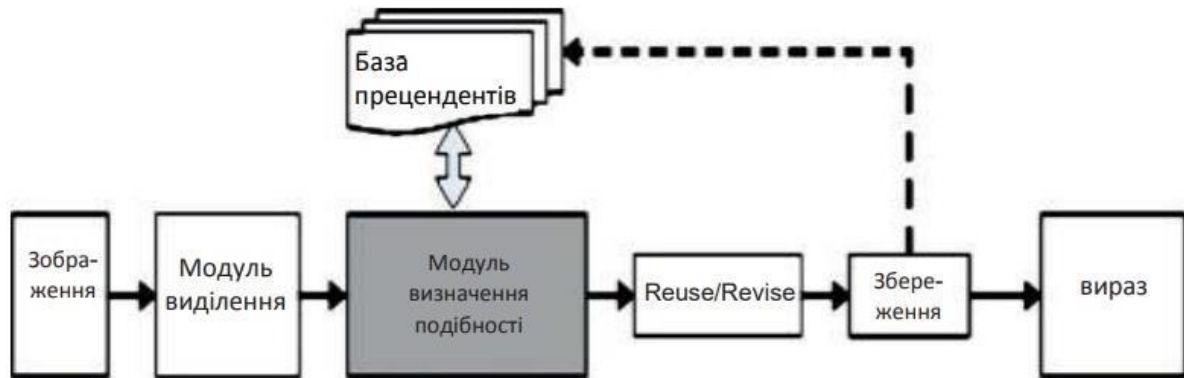


Рисунок 1.2 - Структура системи визначення міміки на основі міркувань

## **1.2. Огляд ринку конкурентів. Програмні забезпечення для розпізнавання емоцій обличчя FaceReader, eMotion Software, MMER\_FEASy**

Розпізнавання емоцій є частиною великого пласта науки, об'єднаного в назви "Розпізнавання образів" та "Обробка візуальної інформації". У наші дні технології розпізнавання перестають бути недосяжними і формується нове віяння, що тягне за собою всіх зацікавлених у методах ідентифікації та обробки знайдених об'єктів та їх ознак. Вони плавно перетікають із області фантастики у реальне життя.

### **FaceReader**

З комерційних рішень на ринку систем розпізнавання емоцій (emotion-recognition systems) найбільш досконалим та цікавішим для розгляду в контексті завдання розпізнавання емоцій на сьогоднішній день є продукт FaceReader голландської компанії Noldus Information Technology.

Програма може чітко інтерпретувати такі висловлювання особи, як «щасливе», «сумне», «сердитий», «здивований», «зляканий», «невдоволений» і «нейтральний», як видно на Рисунку 1.3. Крім того, FaceReader здатний по обличчях людей визначати їх вік, стать та етнічну приналежність. FaceReader не потребує навчання та додаткового настроювання.



Рисунок 1.3 – Програма FaceReader

У програмі реалізовано технології комп'ютерного зору. Зокрема, це метод Active Template, що полягає у накладенні на зображення обличчя шаблону, що деформується: (Рисунок 1.4)

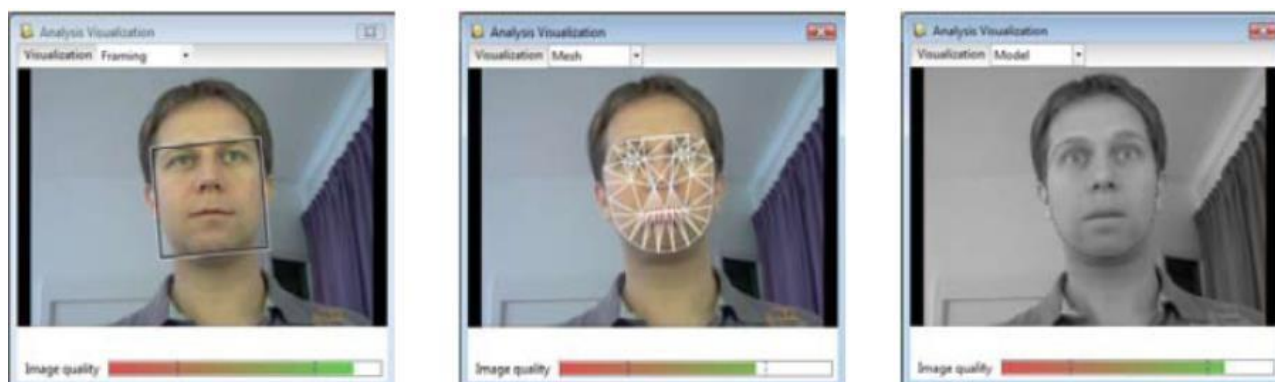


Рисунок 1.4 – Режими програми FaceReader

Реалізовано метод Active Appearance Model, за допомогою якого можна створювати штучну модель особи з урахуванням контрольних точок та деталей поверхні та порівнювати її із зразками, закладеними на згадку.

Класифікація відбувається методами нейронних мереж з тренувальним набором у 2000 фотографій.

### **Можливості програми:**

- середній відсоток розпізнавання емоцій становить 89%. Для деяких емоцій він вищий, для деякого нижче;
- нахил особи може бути будь-яким у площині, його система виявить;
- програма працює із завантажуванним відео у форматах з кодеками MPEG1, MPEG2, XviD, DivX4, DivX5, DivX6, DV-AVI та uncompressed AVI, причому визначати емоції можна пофреймно або повністю при перегляді всього відео. Також FaceReader може працювати зі статичними зображеннями, а також у реальному часі, якщо у користувача підключено веб-камера;
- програма чудово візуалізована: завжди можна подивитися гістограми, діаграми, процентаж емоцій, що виражаються. А на таймлайні видно прояви мікроекспресій у певний проміжок часу;
- FaceReader генерує два текстові файли, один – це лог прояву емоцій, а інший – статичний для пов'язаної з програмою унікальної системи управління візуальними даними The Observer XT, розробленої цією ж компанією.

### **Недоліки програми:**

- FaceReader не натренований для розпізнавання дітей віком до 5 років;
- Якщо людина в окулярах, то розпізнавання емоцій не є точною або класифікація не ведеться;
- Люди з різним кольором шкіри по-різному сприймаються системою, програму повністю адаптовано;
- Навернена особа не детектується.

### **eMotion Software**

Система eMotion Software [2] відома тим, що її фундатори розпізнали емоції на картині "Мона Ліза". Результат показав, що вона була на 83% щаслива, 9% відбивали огиду, 6% страх і лише на 2% Мона Ліза сердилась.

А ще система відома тим, що це по суті перше комерційне платне «коробкове» рішення. Поряд із цим рішенням група розробників запустила сайт GladOrSad.com – відповідно Visual Recognition взяла першість і у відкритті веб-ресурсу, присвяченого онлайн-розпізнаванню емоцій. Рисунок 1.5



Рисунок 1.5 – Програма eMotion Software

Першим відомим користувачем eMotion Software стала компанія Unilever, яка впровадила систему розпізнавання в апарат продажу морозива – Unilever Share Happy. Люди посміхаються автомату, автомат дає за посмішки безкоштовне морозиво!

Якщо людина виявляє емоції, посміхається, хмуриться чи корчить гримасу, то тисячі дрібних м'язів обличчя перебувають у роботі. Система розпізнавання емоцій, або ERS (Emotion-recognition system), створює 3D-модель обличчя з виявленням 12 ключових областей, таких як куточки ока та куточки рота.

У цих програмах алгоритм, що відслідковує, ідентифікує ті ж емоції, їх тут шість: гнів, сум, страх, здивування, огида і щастя, а також сьоме – це їхнє змішання.

Програмне забезпечення не є особливо вимогливим до обчислювальної машини за технічними характеристиками. Про деталі реалізації алгоритму невідомо, т.я. технологія тримається в секреті, брошур із поясненнями я теж не знайшов, на жаль.

### **MMER\_FEASy**

Знову ж таки у розробці використано методологію накладання на особу певної деформованої маски Active Appearance Model methodology, яка дозволяє обчислювати потрібні параметри в реальному часі. Робота з маскою показана на малюнку нижче: (Рисунок 1.6)

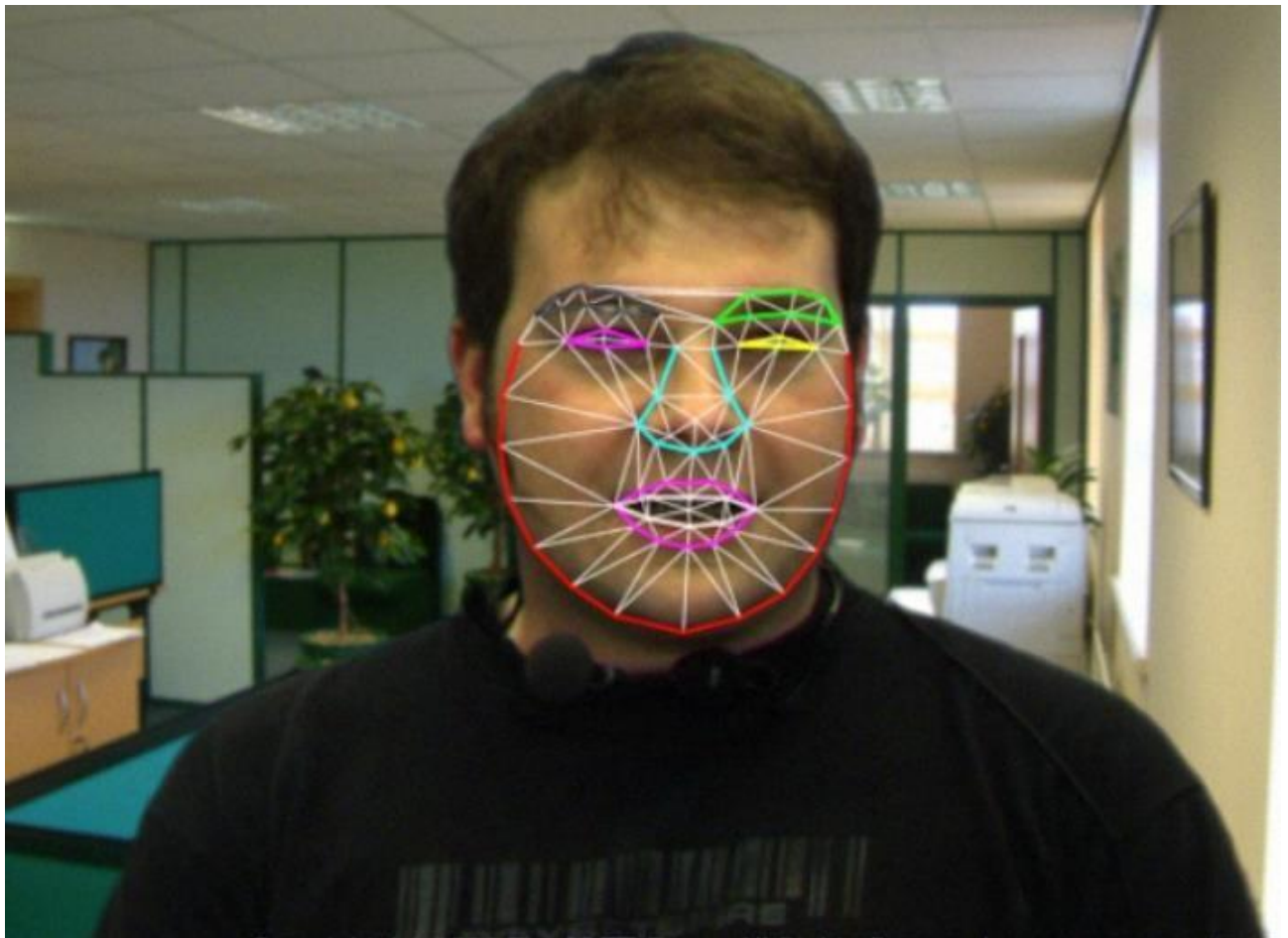


Рисунок 1.6 – Програма MMER\_FEASy

Система використовує три модулі, що підключаються - MMER\_Lab, MMER\_GPU і MMER\_Locate [3].

MMER\_Locate забезпечує знаходження особи на зображенні, MMER\_Lab класифікує деякі ознаки даного зображення, а MMER\_GPU забезпечує ефективну роботу всієї системи: (Рисунок 1.7)

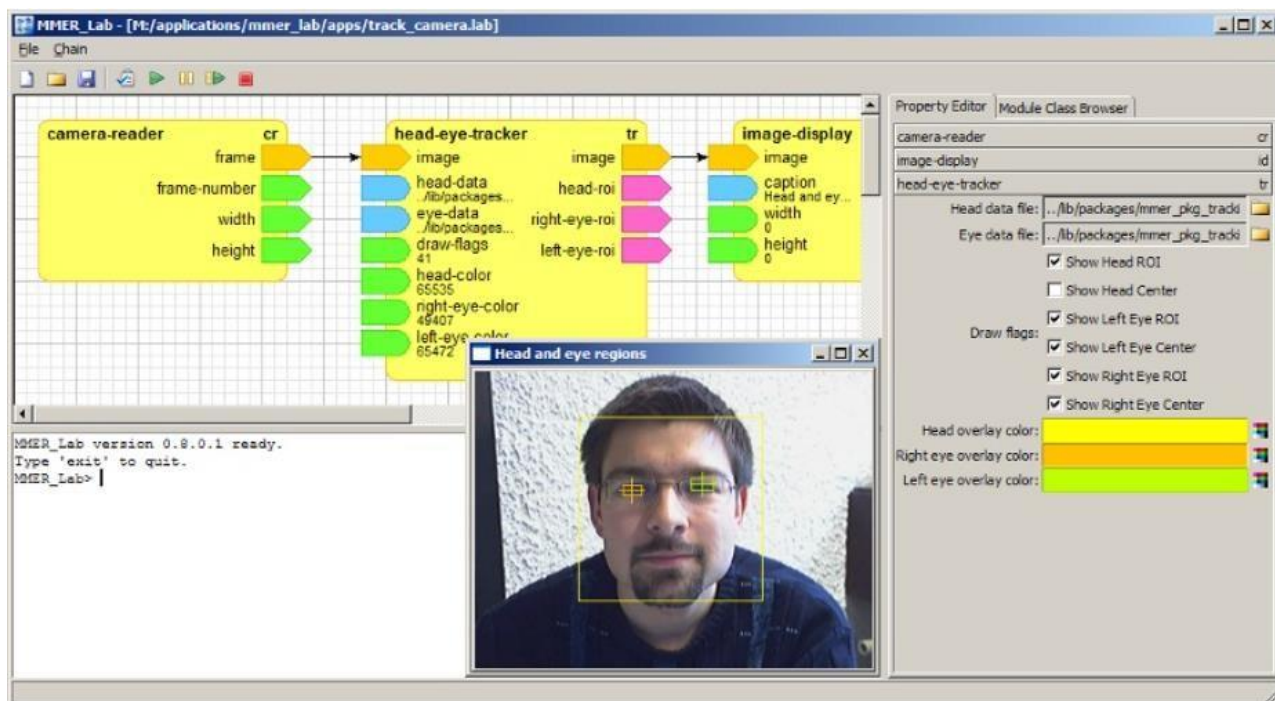


Рисунок 1.7 – Інтерфейс системи MMER\_FEASy

## Висновки до першого розділу

Таким чином, був проведений теоретичний аналіз джерел, який дозволяє зробити наступні висновки:

- розпізнавання емоцій обличчя – спосіб, що дозволяє виявляти, обробляти, аналізувати та ідентифікації емоції відповідно до вхідного зображення. Метою даного методу є розпізнавання та опрацювання комп'ютером емоцій людини так, як це робить звичайна людина;
- розглянуто основні методи розпізнавання емоцій та методи із застосуванням штучних нейронних мереж;
- вивчено та розглянуто розпізнавання емоцій у технології CBR, а саме: зроблено огляд поняття технології CBR, розглянуто особливості застосування, основні завдання, переваги та недоліки CBR, виділено характерні ознаки та визначено основні етапи алгоритму розпізнавання емоцій;
- описані та розглянуті програмні продукти у сфері розпізнавання емоцій обличчя, де кожен має свої переваги та недоліки. Тож, такий аналіз допоміг проаналізувати та зорієнтуватись у функціоналі до кожного продукту, їх можливостях, доступності та особливостях використання.

Таким чином, все вищеописане стає підґрунтям для подальшої роботи з об'єктом, що досліджується. Тож, проаналізовано та вивчено наявні алгоритми для розпізнавання емоцій обличчя. Також було визначено необхідні засоби та методи для реалізації програмного забезпечення для розпізнавання емоцій.

## РОЗДІЛ II

### МЕТОДИ ПОБУДОВИ АРХІТЕКТУРИ НЕЙРОННИХ МЕРЕЖ

#### 2.1. Нейрона мережа: функції, переваги та алгоритм роботи

Нейронні мережі – це набір алгоритмів, створених за моделлю людського мозку, які призначені для розпізнавання закономірностей. Вони інтерпретують сенсорні дані за допомогою свого роду машинного сприйняття, маркування або групування вихідних даних. Шаблони, які вони розпізнають, є числовими і містяться у векторах, у які мають бути переведені всі дані реального світу, будь то зображення, звук, текст чи часові ряди.

Нейронні мережі допомагають групувати та класифікувати. Ви можете розглядати їх як рівень кластеризації та класифікації поверх даних, які ви зберігаєте та керуєте. Вони допомагають групувати немарковані дані відповідно до подібності між прикладами вхідних даних і класифікують дані, якщо у них є призначений навчальний набір. (Нейронні мережі також можуть витягувати функції, які подаються в інші алгоритми для кластеризації та класифікації; тому ви можете розглядати глибокі нейронні мережі як компоненти більших програм машинного навчання, які включають алгоритми підкріплення, класифікації та регресійного навчання.)

Які проблеми вирішує глибоке навчання, і, що ще важливіше, чи може воно вирішити ваші? Щоб знайти відповідь, потрібно задати кілька запитань:

Які результати мене турбують? У задачі класифікації ці результати є мітками, які можна застосувати до даних: наприклад, `spam or not_spam` у фільтрі електронної пошти, `good_guy` або `bad_guy` у виявленні шахрайства,

angry\_customer або happy\_customer в управлінні відносинами з клієнтами. Інші типи проблем включають виявлення аномалій і кластеризацію, що корисно в рекомендаційних системах, які виявляють схожість.

Чи є у мене правильні дані? Наприклад, якщо у вас виникли проблеми з класифікацією, вам знадобляться позначені дані. Чи є необхідний набір даних загальнодоступним, чи його можна створити (за допомогою служби анотації даних, наприклад Scale або AWS Mechanical Turk)? У цьому прикладі електронні листи зі спамом будуть позначені як спам, а мітки дозволять алгоритму відображати вхідні дані до класифікацій, які вас цікавлять. Ви не можете знати, що маєте правильні дані, доки не отримаєте їх. Якщо ви науковець з даних, який працює над проблемою, ви не можете довіряти комусь, хто скаже вам, чи достатньо хороші дані. Тільки безпосереднє вивчення даних дасть відповідь на це питання.

Глибоке навчання порівнює вхідні дані з результатами. Він знаходить кореляції. Він відомий як «універсальний апроксиматор», оскільки він може навчитися апроксимувати невідому функцію  $f(x) = y$  між будь-яким входом  $x$  і будь-яким виходом  $y$ , припускаючи, що вони загально пов'язані (наприклад, кореляція або причинно-наслідковий зв'язок). спілкування). підключення). У процесі навчання нейронна мережа знаходить правильний  $f$  або правильний спосіб перетворення  $x$  в  $y$ , чи є  $f(x) = 3x + 12$  або  $f(x) = 9x - 0,1$ . Ось кілька прикладів того, що може зробити глибоке навчання.

Класифікація

Усі завдання класифікації залежать від заданих наборів даних; тобто люди повинні перенести свої знання в набір даних, щоб нейронна мережа дізналася кореляцію між мітками та даними. Це відомо як навчання з наглядом.

Ідентифікувати обличчя, ідентифікувати людей на зображеннях, розпізнавати вираз обличчя (сердитий, щасливий)

Ідентифікуйте об'єкти на зображеннях (знаки зупинки, пішоходів, розмітка смуги...)

Розпізнавати жести на відео

Визначати голоси, визначати мовців, транскрибувати мовлення в текст, розпізнавати почуття в голосах

Класифікувати текст як спам (в електронних листах) або шахрайський (у страхових платежах); розпізнавати настрої в тексті (відгуки клієнтів)

Будь-які теги, які можуть створити люди, будь-які результати, які вас турбують і корелюють з даними, можна використовувати для навчання нейронної мережі.

## Кластеризація

Кластеризація або групування – це виявлення подібності. Глибоке навчання не вимагає міток подібності. Швидке навчання називається навчанням без нагляду. Дані без міток є найпоширенішими даними у світі. Один із законів машинного навчання: чим більше даних може натренувати алгоритм, тим точнішим він буде. Тому навчання без нагляду має потенціал для створення високоточних моделей.

Пошук: порівняйте документи, зображення чи звуки, щоб знайти схожі елементи.

Виявлення аномалій. Зворотною стороною виявлення подібності є виявлення аномалій або незвичайної поведінки. У багатьох випадках незвичайна поведінка дуже пов'язана з речами, які ви хочете виявити та запобігти.

Прогнозна аналітика: регресії

Класифікація дозволяє глибокому навчанню співвідносити, скажімо, пікселі на зображенні та ім'я людини. Це можна назвати статичним прогнозом. Так само, отримавши достатньо точні дані, поглиблене навчання здатне встановити кореляції між поточними та майбутніми подіями. Він може повертатися між минулим і майбутнім. Майбутня подія в певному сенсі схожа на етикетку. Глибоке навчання не обов'язково дбає про час або той факт, що щось ще не сталося. Враховуючи часові ряди, глибоке навчання може прочитати рядок чисел і передбачити, яке число, швидше за все, буде наступним.

Поломки обладнання (центри обробки даних, виробництво, транспорт)

Розлади здоров'я (інсульт, інфаркти на основі статистики життя та даних ЗМІ)

Відтік клієнтів (прогнозування ймовірності того, що клієнт піде на основі веб-активності та метаданих)

Текучість кадрів (те саме, але для працівників)

Чим краще ми можемо передбачити, тим краще ми зможемо запобігти та запобігти. Як бачите, з нейронними мережами ми рухаємося у світ, де буде менше сюрпризів. Не нуль сюрпризів, лише трохи менше. Ми також переходимо у світ розумніших агентів, які поєднують нейронні мережі з іншими алгоритмами, такими як покращене навчання для досягнення цілей.

З цим коротким оглядом глибокого навчання давайте подивимося, з чого складаються нейронні мережі.

## Елементи нейронної мережі

Глибоке навчання — це назва, яку ми використовуємо для «нейронних мереж із стек»; тобто мережі, що складаються з кількох шарів.

Шари складаються з вузлів. Вузол — це просто місце, де відбуваються обчислення, нещільно нанесені на нейрон в мозку людини, який спрацьовує, коли зустрічає достатню кількість подразників. Вузол поєднує вхідні дані з набором коефіцієнтів або ваг, які або посилюють, або гасять цей вхід, тим самим призначаючи значення вхідних даних щодо завдання, яке намагається вивчити алгоритм; напр. який вхід є найбільш корисним - це класифікація даних без помилок? Ці вхідні вагові добутки підсумовуються, а потім сума передається через так звану функцію активації вузла, щоб визначити, чи повинен і в якій мірі цей сигнал просуватися далі по мережі, щоб вплинути на кінцевий результат, скажімо, акт класифікації. Якщо сигнали проходять, нейрон був «активованим».

Ось діаграма того, як може виглядати один вузол. (Рисунок 2.1)

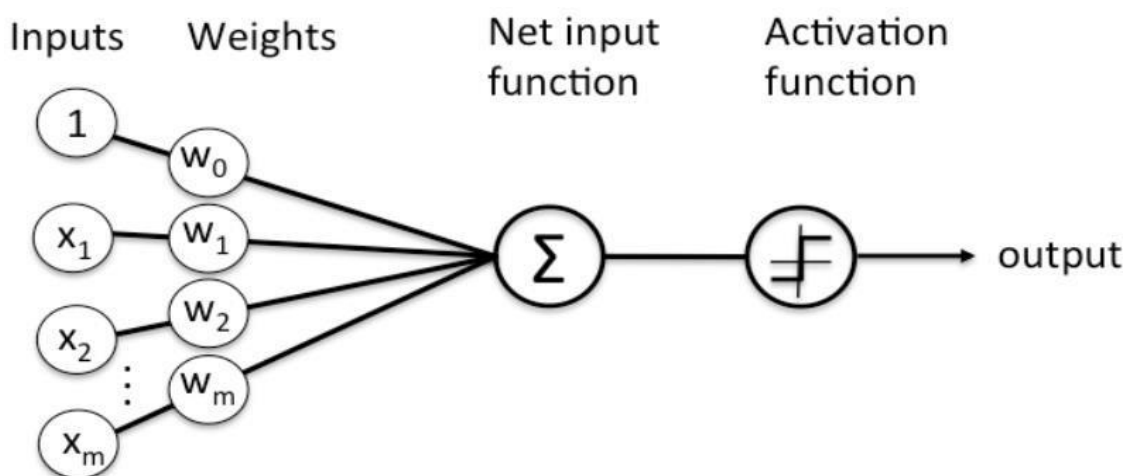


Рисунок 2.1 Вигляд одного вузла

Вузловий шар — це ряд нейроноподібних перемикачів, які вмикаються або вимикаються, коли вхід подається через мережу. Вихід кожного шару одночасно є входом наступного шару, починаючи з початкового рівня введення, який отримує ваші дані. (Рис 2.2)

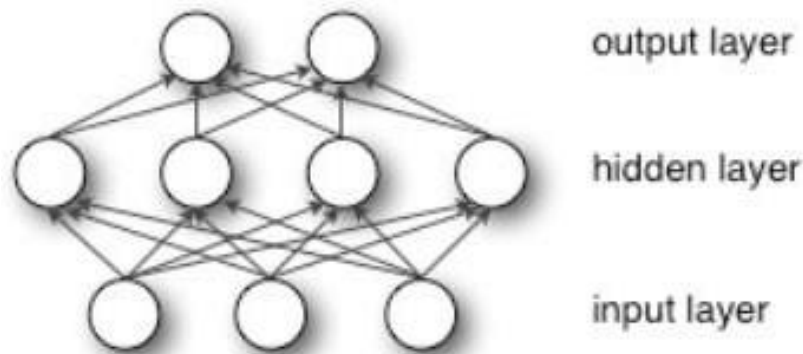


Рисунок 2.2 Слої вузла

Комбінація регульованих ваг моделі з вхідними функціями – це те, як ми оцінюємо ці функції з точки зору того, як нейронна мережа класифікує та групує вхідні дані.

Ключові поняття глибоких нейронних мереж

Мережі глибокого навчання відрізняються від більш звичайних одношарових прихованих нейронних мереж своєю глибиною; тобто кількістю шарів вузлів, через які повинні пройти дані в багатоетапному процесі розпізнавання образів.

Попередні версії нейронних мереж, такі як перші перцептрони, були неглибокими і склалися з одного вхідного і вихідного шарів і щонайбільше одного прихованого шару між ними. Більше трьох рівнів (включаючи вхідний і вихідний) кваліфікуються як «глибоке» навчання. Настільки глибокий — це не просто модне слово, що алгоритми, здається, читають Сартра і слухають групи,

про які ви ніколи не чули. Це строго визначений термін, який означає більше ніж один прихований шар.

У мережах глибокого навчання кожен рівень вузлів навчається на певному наборі функцій на основі результатів попереднього рівня. Чим далі ви рухаєтеся в нейронну мережу, тим складніші функції можуть розпізнавати ваші вузли, оскільки вони об'єднують і рекомбінують функції з попереднього шару.

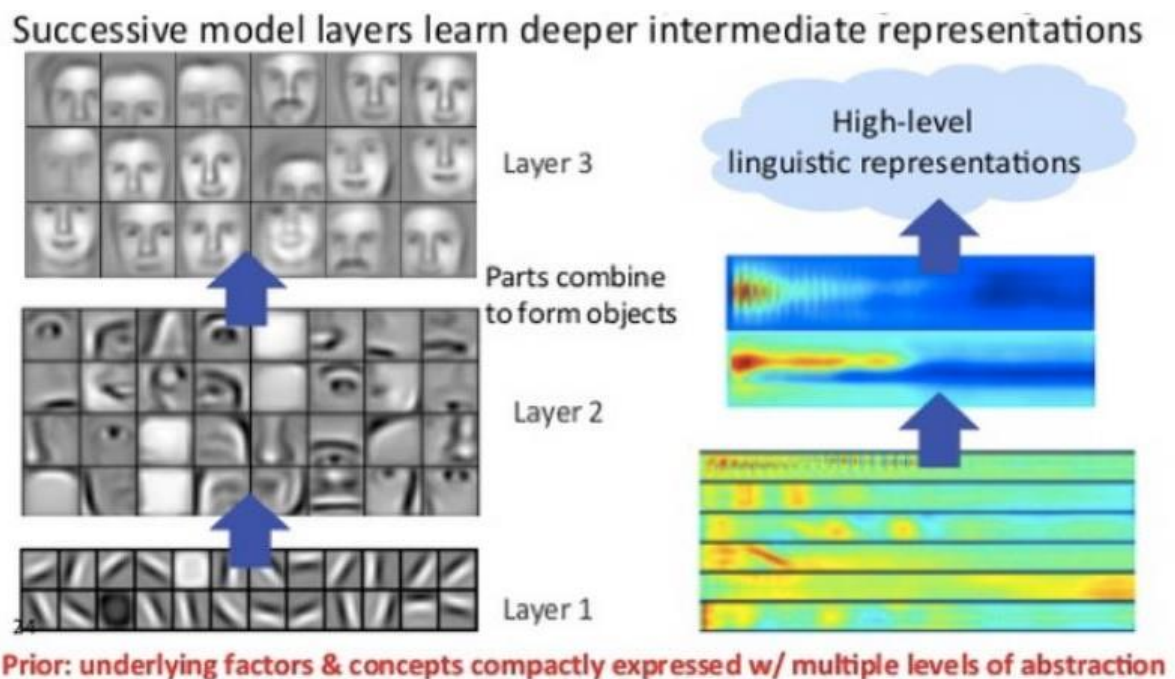


Рисунок 2.3 Успішне навчання слоїв моделі

Це відомо як ієрархія функцій, і це ієрархія зростаючої складності та абстракції. Це робить мережі глибокого навчання здатними обробляти дуже великі набори даних з мільярдами параметрів, що проходять через нелінійні функції.

Перш за все, ці нейронні мережі здатні виявляти приховані структури в немаркованих, неструктурованих даних, які становлять переважну більшість даних у світі. Іншими словами, неструктуровані дані є необробленими медіа;

тобто зображення, тексти, відео та аудіозаписи. Таким чином, однією з проблем, які найкраще вирішує поглиблене навчання, є обробка та групування немаркованих немаркованих носіїв у світі, виявлення подібності та аномалій у даних, які ніхто не організував у реляційну базу даних і ніколи не називав.

Наприклад, глибоке навчання може отримати мільйон зображень і згрупувати їх за схожістю: кішки в одному кутку, криголами в іншому, а в третьому всі фотографії вашої бабусі. Це основа так званих розумних фотоальбомів.

Тепер застосуйте ту ж ідею до інших типів даних: розширене навчання може поєднувати необроблений текст, наприклад електронні листи чи статті новин. В одному кутку векторного простору можуть з'являтися листи, наповнені гнівними скаргами, а в інших – задоволені клієнти чи спам-боти.

За допомогою часових рядів дані можна згрупувати навколо нормальної/здорової поведінки та аномальної/небезпечної поведінки. Якщо дані часового ряду генеруються смартфоном, це дасть уявлення про здоров'я та звички користувачів; якщо його генерує автозапчастина, його можна використовувати для запобігання катастрофічних поломок.

На відміну від більшості традиційних алгоритмів машинного навчання, мережі глибокого навчання виконують автоматичне видалення функцій без втручання людини. Враховуючи, що видалення функцій — це завдання, на виконання якого командам науковців з даних можуть знадобитися роки, поглиблене навчання — це спосіб обійти обмеження експертів. Це збільшує спроможність невеликих команд науки про дані, які не є масштабованими за своєю природою.

Під час навчання на немаркованих даних кожен рівень вузла в глибокій мережі автоматично вивчає функції, багаторазово намагаючись відновити вхідні дані, з яких він бере свої вибірки, намагаючись мінімізувати різницю між мережевими припущеннями та розподілом ймовірностей вхідних даних. . Обмежені машини Больцмана, наприклад, створюють так звані реконструкції.

У процесі ці нейронні мережі вчаться розпізнавати кореляції між певними відповідними характеристиками та оптимальними результатами – вони встановлюють зв'язки між сигналами сигналів і тим, що представляють ці функції, будь то повна реконструкція чи мічені дані.

Мережу поглибленого навчання, яку навчають мічені дані, можна потім застосувати до неструктурованих даних, надаючи їй доступ до набагато більшого входу, ніж мережі машинного навчання. Це рецепт вищої продуктивності: чим більше даних може навчити мережа, тим точніше вона буде. (Погані алгоритми, засвоєні з великої кількості даних, можуть перевершити хороші алгоритми, засвоєні на дуже низьких рівнях.) Здатність глибокого навчання обробляти та вчитися на величезних обсягах немаркованих даних дає йому явну перевагу перед попередніми алгоритмами.

Мережі глибокого навчання закінчуються на початковому рівні: логістиці, або softmax, класифікаторі, який призначає ймовірність певного результату або мітки. Ми називаємо це прогнозним, але він є прогнозним у широкому сенсі.

## 2.2. Нейронна мережа CNN та її архітектура

Людський мозок обробляє величезну кількість інформації, як тільки ми бачимо зображення. Кожен нейрон працює у своєму рецептивному полі і пов'язаний з іншими нейронами таким чином, що вони охоплюють все поле зору. Подібно до того, як кожен нейрон реагує на стимули лише в обмеженій області поля зору, яка називається рецептивним полем у системі біологічного зору, кожен нейрон в CNN також обробляє дані лише в своєму рецептивному полі. Шари розташовуються таким чином, що спочатку відкриваються простіші візерунки (лінії, криві тощо), а потім — більш складні (грані, предмети тощо). За допомогою CNN ви можете перетворити бачення на комп'ютері.

Архітектура згорткової нейронної мережі (Рисунок 2.4)

CNN зазвичай має три рівні: згортковий, агрегаційний і повністю зв'язаний.

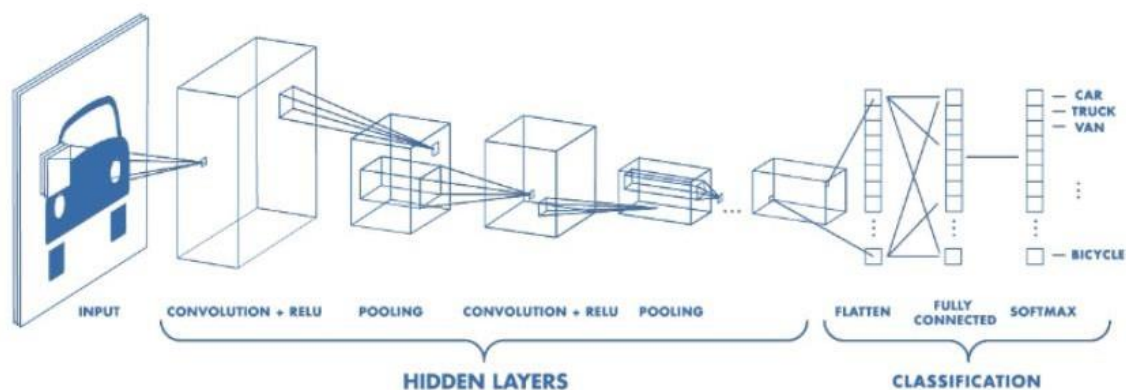


Рисунок 2.4 Архітектура CNN

Шар згортки є основним будівельним блоком CNN. Він несе основну частину обчислювального навантаження мережі.

Цей рівень виконує точковий твір між двома матрицями, де одна матриця являє собою набір параметрів, які можна вивчати, інакше відомих як ядро, а інша матриця є обмеженою частиною рецептивного поля. Ядро просторово менше за зображення, але глибше. Це означає, що якщо зображення складається з трьох каналів (RGB), висота і ширина ядра будуть просторово малими, але глибина поширюється на всі три канали.

Під час прямого проходження ядро ковзає по висоті та ширині зображення, створюючи зображення цієї сприйнятливої області. Це створює двовимірне представлення зображення, відоме як карта активації, яка забезпечує основну відповідь у кожному просторовому положенні зображення. Ковзний розмір ядра називається кроком.

Якщо ми маємо вхідні дані розміром  $W \times W \times D$  і кількість ядер  $D_{out}$  з просторовим розміром  $F$  з кроком  $S$  і значенням заповнення  $P$ , то розмір вихідного обсягу можна визначити за такою формулою:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

Це дасть вихідний обсяг  $W_{out} \times W_{out} \times D_{out}$ . (Рисунок 2.5)

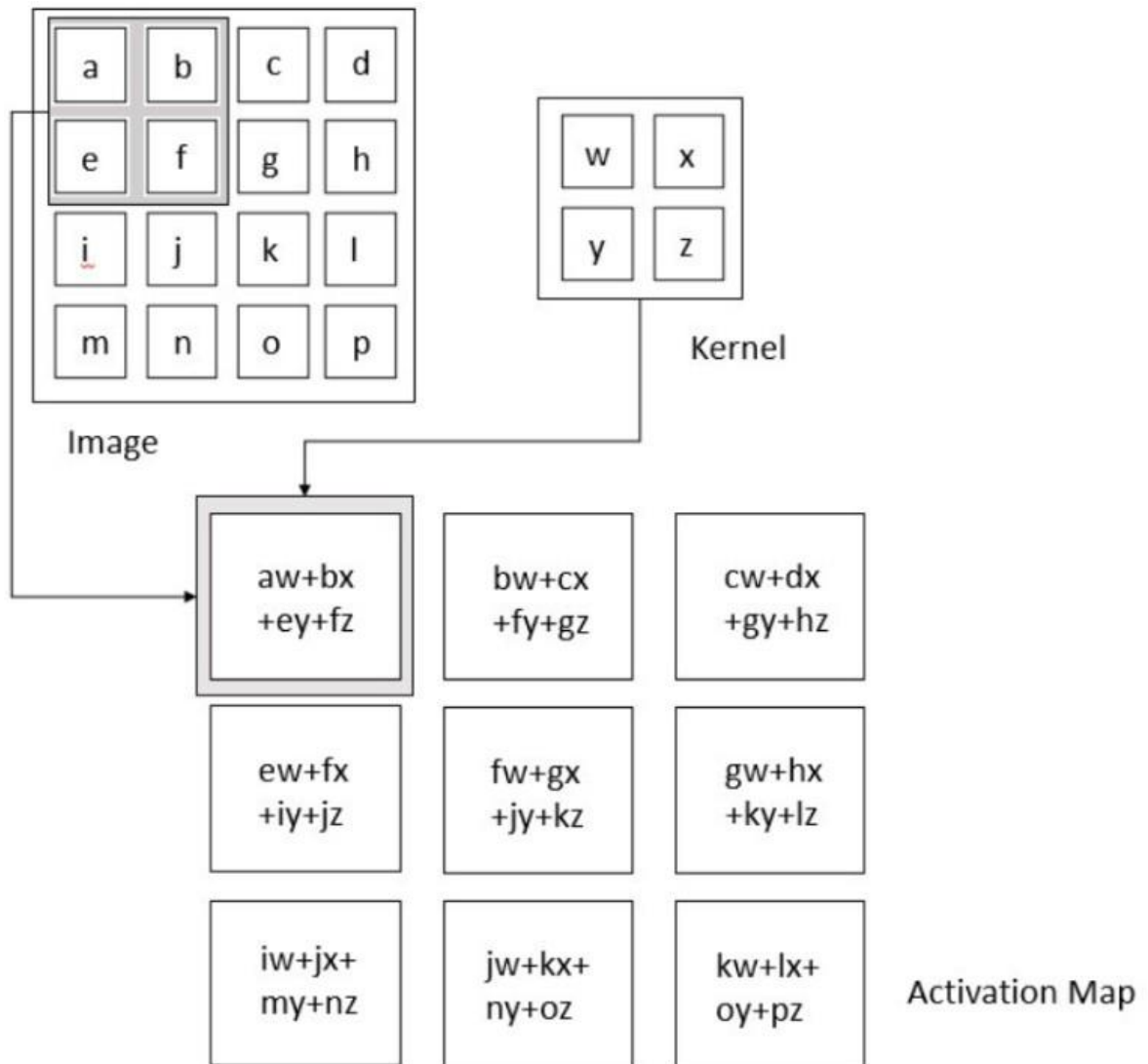


Рисунок 2.5 - Операція згортки

## Convolution

Convolution (згортка) використовує три важливі ідеї, які спонукали дослідників комп'ютерного зору: розріджена взаємодія, спільне використання параметрів і еквівалентне представлення. Ми детально опишемо кожен з них.

Тривіальні шари нейронної мережі використовують множення матриці на матрицю параметрів, які описують взаємодію між вхідним і вихідним блоком. Це означає, що кожен вихідний блок взаємодіє з кожним вхідним блоком.

Однак згорткові нейронні мережі мають розріджені взаємодії. Це досягається за рахунок зменшення розміру ядра, ніж вхідного, наприклад, зображення може мати мільйони або тисячі пікселів, але при обробці ядром ми можемо знайти значну інформацію, що складається з десятків або сотень пікселів. Це означає, що нам потрібно зберігати менше параметрів, що не тільки зменшує потребу моделі в пам'яті, але й покращує статистичну ефективність моделі.

Якщо обчислення одного об'єкта в просторовій точці  $(x_1, y_1)$  є корисним, він також має бути корисним в іншій просторовій точці, скажімо  $(x_2, y_2)$ . Це означає, що для одного двовимірного зрізу, тобто для створення єдиної карти активації, нейрони змушені використовувати один і той же набір ваг. У традиційній нейронній мережі кожен елемент вагової матриці використовується один раз, а потім ніколи не переглядається, в той час як мережа згортки має загальні параметри, тобто для отримання ваги, застосованої до одного входу, такої ж, як вага, що використовується в інших місцях. .

Завдяки спільному використанню параметрів шари згорткової нейронної мережі матимуть властивість еквівалентності трансляції. Він каже, що якщо ми змінимо вхід, то вихід зміниться таким же чином.

### **Pooling Layer**

Рівень агрегації замінює вихід мережі в певних місцях шляхом отримання остаточної статистики найближчого джерела. Це допомагає зменшити просторовий розмір перегляду, що зменшує кількість необхідних обчислень і ваг. Операція злиття обробляється для кожної частини подання окремо.

(Рисунок 2.6)

Існує кілька функцій агрегації, таких як середнє значення прямокутного околиці, швидкість L2 для прямокутного околиці та середньозважене значення на основі відстані від центрального пікселя. Однак найпопулярнішим процесом є максимальний пул, який сповіщає про максимальний вихід з околиці.

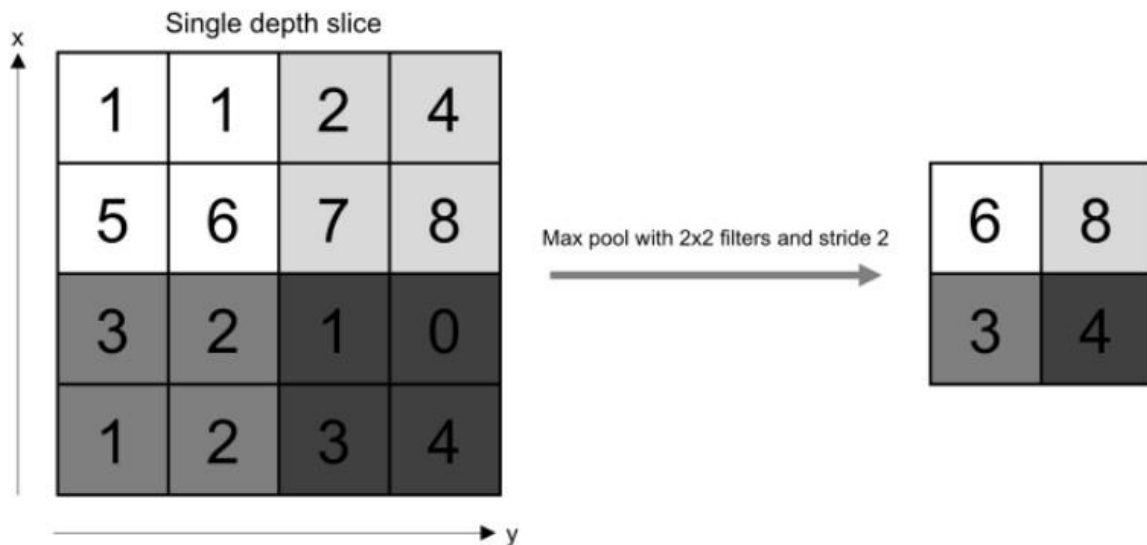


Рисунок 2.6- Операція pooling

Якщо ми маємо карту активації розміром  $W \times W \times D$ , ядро об'єднання просторового розміру  $F$  і кроку  $S$ , то розмір вихідного об'єму можна визначити за такою формулою [3]:

$$W_{out} = \frac{W - F}{S} + 1$$

Це дасть вихідний обсяг розміром  $W_{out} \times W_{out} \times D$ .

У всіх випадках об'єднання в пул забезпечує деяку інваріантність перекладу, що означає, що об'єкт буде впізнаваним незалежно від того, де він з'являється у кадрі.

## Fully Connected Layer

Нейрони в цьому шарі мають повний зв'язок з усіма нейронами в попередньому та наступному шарах, як це видно у звичайному FCNN. Тому його можна розрахувати, як зазвичай, шляхом множення матриці з подальшим ефектом зміщення.

Рівень FC допомагає відображати перегляд між входом і виходом.

## Non-Linearity Layers

Оскільки згортка є лінійною операцією, а зображення далекі від лінійності, шари нелінійності часто розміщуються відразу після шару згортки, щоб додати нелінійність до карти активації.

Існує кілька типів нелінійних операцій, найпопулярнішими з яких є:

### 1. Sigmoid

Сигмовидна нелінійність має математичний вигляд  $\sigma(x) = 1 / (1 + e^{-x})$ . Він бере число з реальним значенням і «стискає» його в діапазоні від 0 до 1.

Однак дуже небажаною властивістю сигмовидної є те, що коли активація відбувається на будь-якому хвості, градієнт стає майже нульовим. Якщо локальний градієнт стає дуже малим, то при зворотному поширенні він ефективно «вбиває» градієнт. Крім того, якщо дані, що надходять в нейрон, завжди позитивні, вихідні дані сигмовидної кістки будуть або всі позитивними, або всі негативними, що призведе до зигзагоподібної динаміки градієнта градієнта ваги.

## 2. Tanh

Tanh стискає число з реальним значенням до діапазону  $[-1, 1]$ . Як і сигмовидна, активація насичена, але - на відміну від сигмовидних нейронів - її вихід має нульовий центр.

## 3. ReLU

Випрямлений лінійний блок (ReLU) став дуже популярним в останні кілька років. Він обчислює функцію  $f(x) = \max(0, x)$ . Іншими словами, активація просто нульова.

У порівнянні з сигмовидною і танковою, ReLU надійніше і прискорює зближення в шість разів.

На жаль, недоліком є те, що ReLU може бути крихким під час навчання. Великий градієнт, що протікає через нього, може оновити його таким чином, що нейрон більше ніколи не оновлюватиметься. Однак ми можемо працювати з цим, встановивши правильну швидкість навчання.

### 2.2.1. Використана методологія.

Запропонована методологія представлена за допомогою блок-схеми (Рисунок 2.7)

Першим кроком у процесі є навчання набору даних. Набір даних навчається за допомогою CNN шарів.

Навчена модель використовується для розпізнавання емоцій обличчя у введеному зображенні. (Рисунок 2.8)

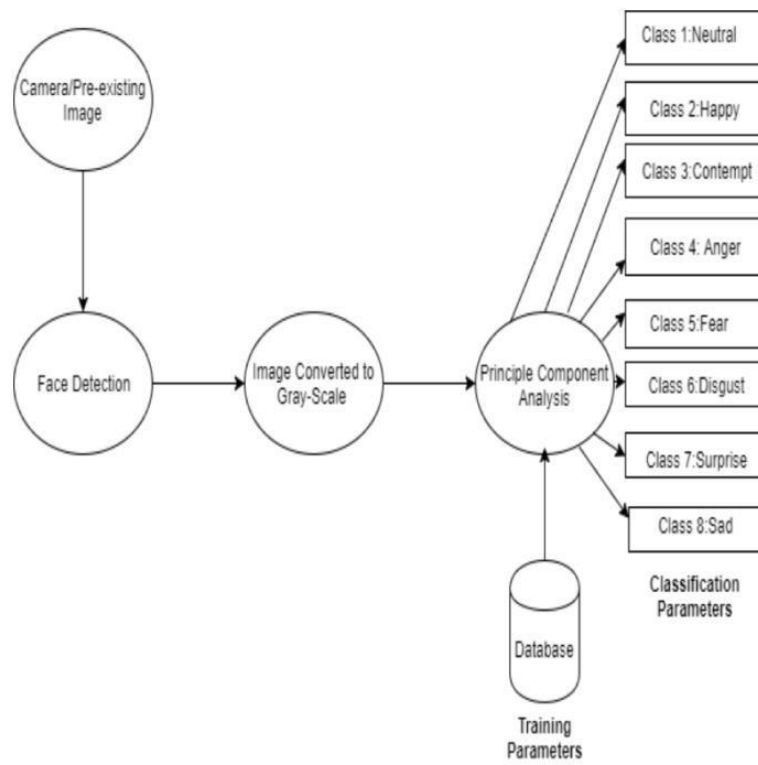


Рис. 2.7 Блок-схема методології розпізнавання емоцій обличчя

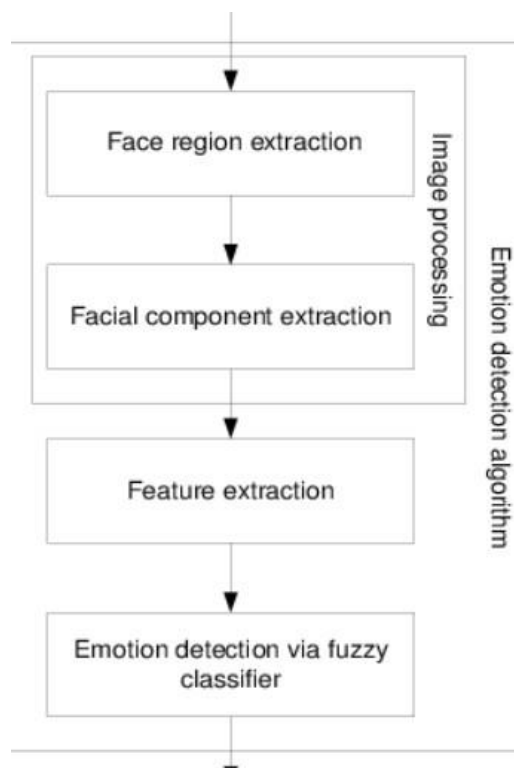


Рисунок 2.8 Алгоритм розпізнавання обличчя

### Висновки до другого розділу

В даному розділі було розглянуто основні методи багат шарових нейронних мереж, а також описана методологія розпізнавання емоцій обличчя шляхом попередньої обробки та навчання набору даних послідовно з CNN. Вхідні зображення спочатку попередньо обробляються за допомогою методів контуру OpenCV. Результати демонструють потенціал використання CNN, яка покращує точність та стабільність. Можливістю для покращення роботи є використання гібридних наборів даних та експерименту з різними функціями активації, а також збільшення кількості шарів нейронної мережі.

## РОЗДІЛ III

### РЕАЛІЗАЦІЯ ПЗ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

#### 3.1. Вибір технологій розробки та її обґрунтування

##### 3.1.1. Вибір платформи для розробки.

Головним завданням було спростити розробку CNN, та використати методи оптимізації моделей, що допомогло б отримати найбільшу точність та якість тренування. Тож, за основу було взято сервіс kaggle, який дозволяє імпортувати вже розміщені датасети в інтернеті з подальшим навчанням CNN.

Також за допомогою аналітичних інструментів Colab Research існує можливість перегляду датасетів, точності навчання, втрат чи перенавчання нейромереж тощо. Також на практиці існує можливість завантажити власні дані з пристроїв. Сервіс має вбудований компілятор, що дозволяє виявляти помилки в розробці нейромереж, швидко їх знаходити та виправляти. Варто зазначити, що всі бібліотеки вже вбудовані в сервіс, що є перевагою над локальною розробкою, адже на встановлення усіх додаткових компонентів потрібен час, який є досить дорогим ресурсом сучасності.

##### 3.1.2. Вибір мови програмування.

Вибір мови програмування був очевидним, бо головною перевагою Python є те, що він є максимально простим для розробки Data Science через свою зручну читабельність та низький поріг входу в розробку CNN.

Вибір мови програмування не обмежується лише пристосуванням до розробки. Такі бібліотеки як Numpy, Matplotlib не просто є частиною Python community в Data Science. Також існують багато розширених фреймворків. Вони призначені для розробки Deep Learning моделей та мають широку підтримку розробників у світі. Серед них найбільш вживаними є Tensorflow, тощо.

### 3.1.3. Вибір бібліотек Python.

Існує безліч таких бібліотек, що допомагають та спрощують роботу з нейронними мережами. Але в магістерській роботі було використано декілька.

Бібліотека Numpy використовується для Python обчислень. Вона має багато складних функцій, алгебраїчних обчислень та рядів, рандомізацію, K-розмірні масиви тощо.

Бібліотека Tensorflow. Вона допомагає взаємодіяти з нейронними мережами та містить велику кількість методів та оптимізаторів для CNN. Зокрема деякі з них були використані в роботі та показали себе на відмінно.

Загальним плюсом усіх використаних бібліотек є велика підтримка розробників, а саме кодової бази, швидкодії роботи функцій, зручність написання тощо.

## 3.2. Моделювання системи використовуючи UML діаграми

UML (Unified Modeling Language) — це мова графічного моделювання загального призначення в галузі програмної інженерії. UML використовується для ідентифікації, візуалізації, проектування та документування артефактів (ключових елементів) програмної системи.

UML розшифровується як Unified modeling language, є стандартизованою мовою візуального моделювання загального призначення в області програмної інженерії. Він використовується для виявлення, візуалізації, проектування та документування артефактів первинної програмної системи. Це допомагає в розробці та характеристиці, особливо програмних систем, які включають

концепцію об'єктної орієнтації. Він описує роботу як програмних, так і апаратних систем.

UML отримала широке поширення в середовищі програмістів, однак в цілому мало використовується в розробці баз даних. Одна з причин цього полягає в тому, що даний напрямок є менш дослідженим. Проте, UML ефективно застосовується в концептуальному моделюванні даних високих рівнів і підходить

У даній роботі було використано такі види діаграм: прецедентів або Use-case і послідовностей або Sequence

Діаграма прецедентів — це динамічна або поведінкова діаграма в UML. Діаграми використання моделюють функціональність системи з акторами та параметрами використання. Параметри використання — це набір дій, послуг і функцій, які повинна виконувати система. У цьому контексті «система» — це те, що розробляється або керується, наприклад веб-сайт. «Актори» — це люди або організації, які виконують певні ролі в системі.

Діаграма прецедентів використовує 2 основних елементи:

1. Use case (прецедент) – опис окремого аспекту поведінки системи з точки зору користувача. Прецедент не показує, «як» досягається певний результат, а тільки «що» саме виконується.
2. Actor (учасник) – безліч логічно пов'язаних ролей, виконуваних при взаємодії з прецедентами (система, підсистема або клас). Учасником може бути людина, роль людини в системі або інша система, підсистема або клас.

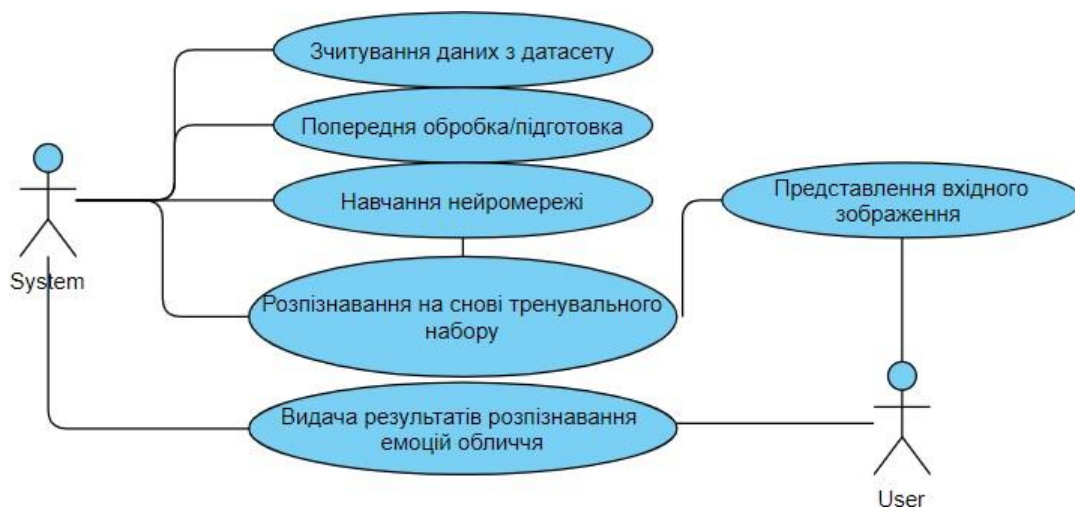


Рис 3.1 Діаграма прецедентів системи розпізнавання текстів

#### 1. Актори:

- користувач;
- система

#### 2. Прецеденти (1-й блок):

- Представлення вхідного зображення

Ці прецеденти об'єднані в одну групу через те, що вони взаємодіють з джерелом датасетів та користувачем в кінці ланцюга прецедентів системи.

#### Прецеденти (2-й блок):

- зчитування даних з датасету;
- попередня обробка та підготовка датасету;
- навчання нейромережі;
- розпізнавання на основі тестового набору даних;
- вивада результатів розпізнавання емоцій обличчя;

Ці прецеденти об'єднані в одну групу через те, що вони всі фактично виконуються системою та представляють результат для користувача.

Діаграма послідовностей – використовується для уточнення діаграм прецедентів – описує поведінкові аспекти системи. Діаграма послідовності показує взаємодію об'єктів в динаміці, в часі. При цьому інформація набуває вигляду повідомлень, а взаємодія об'єктів передбачає обмін цими повідомленнями в рамках сценарію.

На діаграмі послідовностей показано у вигляді вертикальних ліній різні процеси або об'єкти, що існують водночас. Надіслані повідомлення зображуються у вигляді горизонтальних ліній, в порядку відправлення. Основними елементами діаграми послідовності є позначення об'єктів (прямокутники з назвами об'єктів), вертикальні «лінії життя», що зображують плин часу, прямокутники, що зображують діяльність об'єкта або виконання ним певної функції (прямокутники на пунктирній «лінії життя»), і стрілки, що показують обмін сигналами або повідомленнями між об'єктами.

Діаграми послідовностей можуть бути корисними посиланнями для підприємств та інших організацій. Їх перевагами є:

- представляють деталі випадку використання UML;
- моделюють логіку складної процедури, функції або операції;
- показують, як об'єкти та компоненти взаємодіють між собою для завершення процесу;
- планують детальну функціональність наявного або майбутнього сценарію.

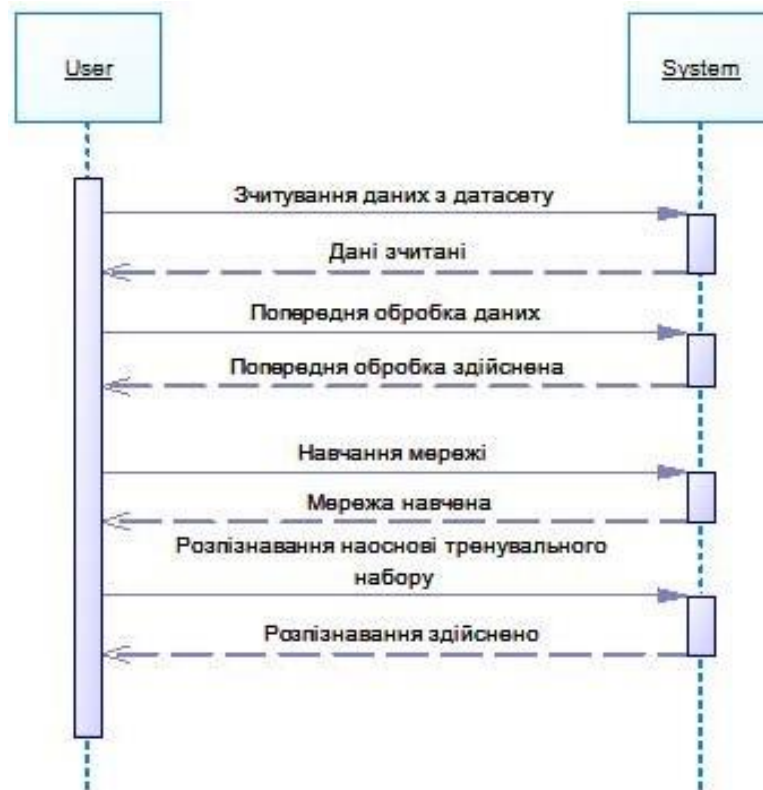


Рис 3.2 Діаграма послідовностей системи розпізнавання текстів

На діаграмі послідовностей видно, що ніякої асинхронності у системі немає, лише синхронні функції. Чітка взаємодія між об'єктами системи.

### 3.3. Модель та реалізація нейронної мережі

#### 3.3.1. Розпізнавання емоцій обличчя.

Для того, щоб почати розпізнавати емоції обличчя, нам потрібно навчити модель це робити. Для цього будемо працювати в kaggle. Тут можна імпортувати дані для тренування та тестування вибірок. Цей проект має на меті класифікувати емоції на обличчі людини в одну із семи категорій, використовуючи глибокі згорткові нейронні мережі. Модель навчається на наборі даних FER-2013, який був опублікований на Міжнародній конференції з машинного навчання (ICML). Цей набір даних складається з 35887 зображень облич у відтінках сірого розміром 48x48 із сімома емоціями: гнів, огида, страх, радість, нейтральний, сумний і здивований.

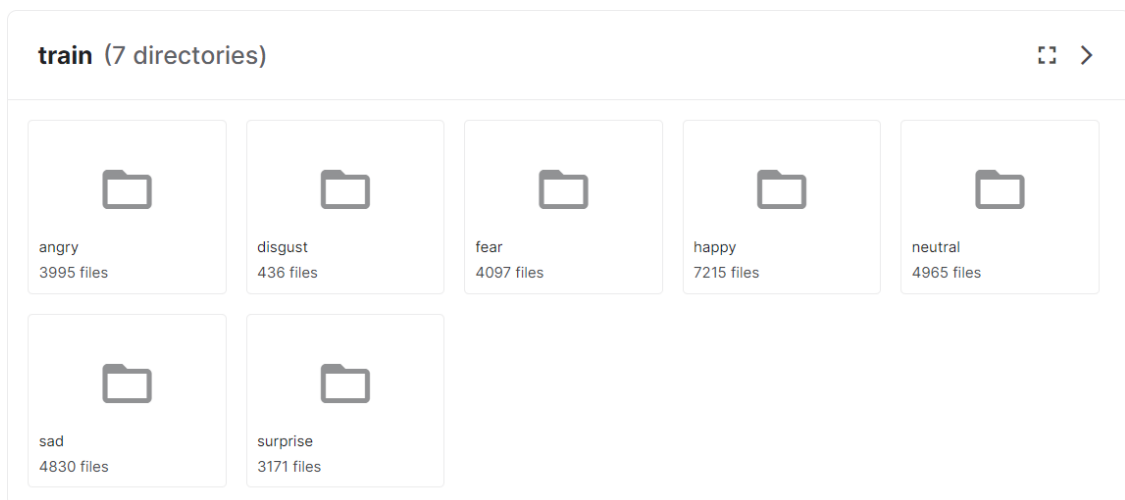


Рис 3.3 Імпорт тренувальних даних навчальної вибірки

Після цього надамо приклад тестової вибірки, яку будемо навчати.

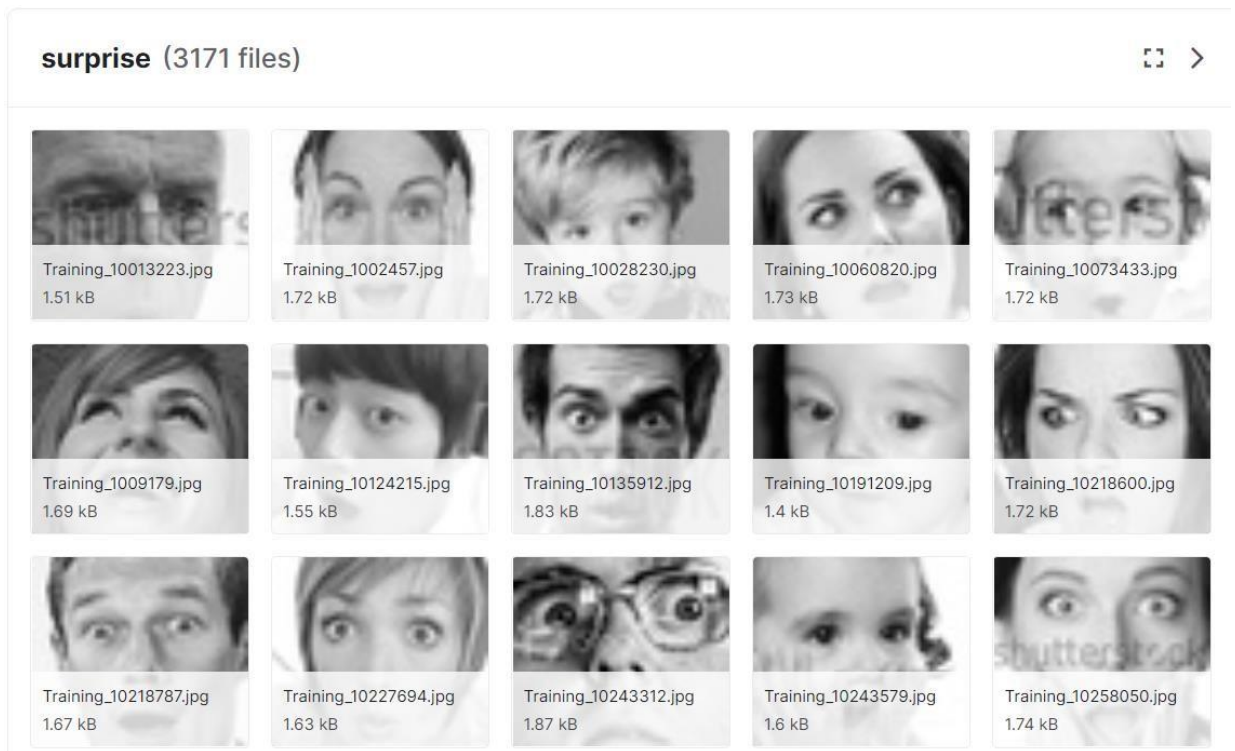


Рис 3.4 Приклад обличч та емоцій, на яких будемо навчати нейромережу

Отже, можна розпочинати тренувати нашу нейромережу.

Для того, щоб почати будувати модель у Tensorflow використовується метод `.Sequential([])`. Цей метод послідовно групує лінійний набір шарів та надає функції навчання та висновків у цій моделі.

Метод `.Flatten()`. Вирівнює вхідні дані. Не впливає на розмір партії. Він приймає параметри вхідного шару на вхід (`input_shape`).

Метод `.Dense()`. Це просто звичайний щільно пов'язаний шар NN. `Dense` реалізує операцію:  $\text{вихід} = \text{активація}(\text{точка}(\text{вхід}, \text{ядро}) + \text{зміщення})$ , де активація – це поелементна функція активації, передана як аргумент активації, ядро – це матриця ваг, створена шаром, а зміщення – це створений вектор зміщення. за рівнем.

Після того, як звичайна архітектура була створена, потрібно розпочати компіляцію моделі із вхідними параметрами. `Model.compile(args)` приймає на вхід такі параметри:

- optimizer= «adam» Метод оптимізації Адама – це метод стохастичного градієнтного спуску, який базується на адаптивній оцінці моментів першого та другого порядку.
- loss= «name» Метою функцій втрат є обчислення величини, яку модель повинна прагнути мінімізувати під час навчання.
- metrics= «value» - Показник – це функція, яка використовується для оцінки ефективності моделі.

Далі можна розпочати тренування нейромережі на тренувальній вибірці. Тут потрібно вказати, що параметр epochs - це кількість епох для навчання моделі. Використаємо метод .fit(args) для тренування базової моделі розпізнавання емоцій обличчя. Побудова моделі була покращена функцією Conv2D() та MaxPool2D() функцією активації(relu) [4][5] (Рис3.5)

```
# Create the model
model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))

# If you want to train the same model or try other models, go for this
if mode == "train":
    model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.0001, decay=1e-6),metrics=['accuracy'])
    model_info = model.fit_generator(
        train_generator,
        steps_per_epoch=num_train // batch_size,
        epochs=num_epoch,
        validation_data=validation_generator,
        validation_steps=num_val // batch_size)
    plot_model_history(model_info)
    model.save_weights('model.h5')
```

Рисунок 3.5 Реалізація побудови та тренування моделі

Зазначимо кількість епох = 50 і отримаємо такі результати навчання на тренувальній вибірці. (Рис.3.6, Рис3.7)

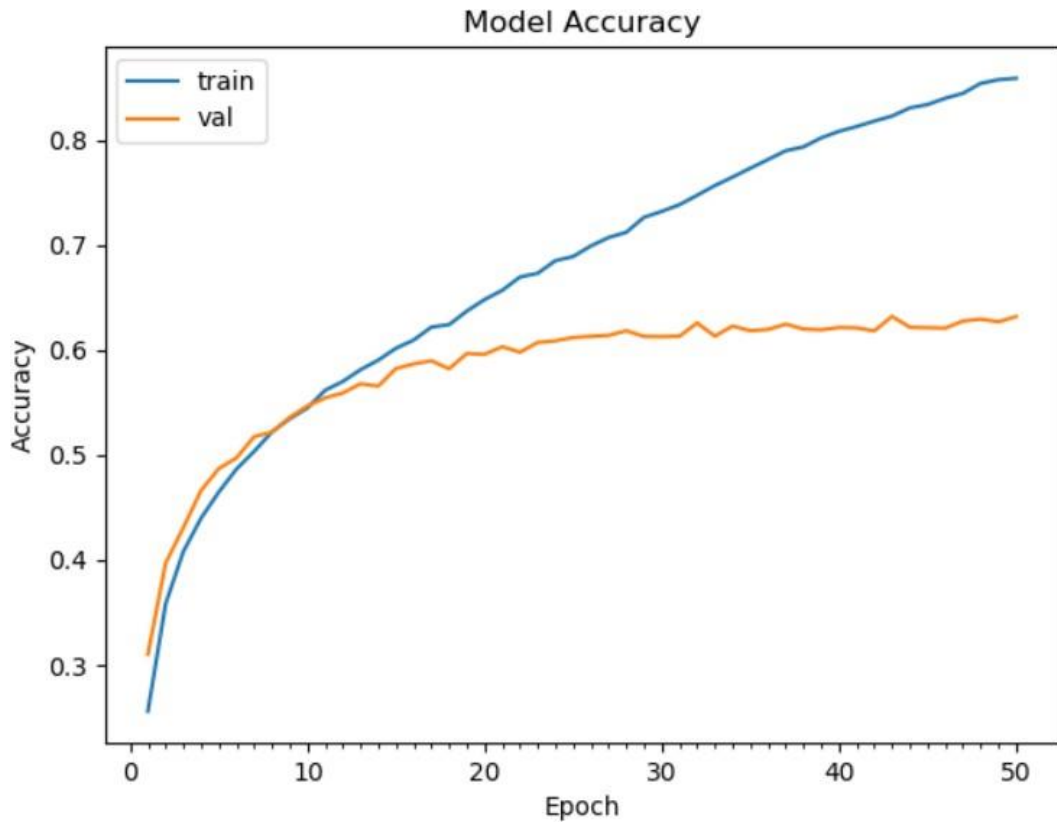


Рисунок 3.6 Тест точності

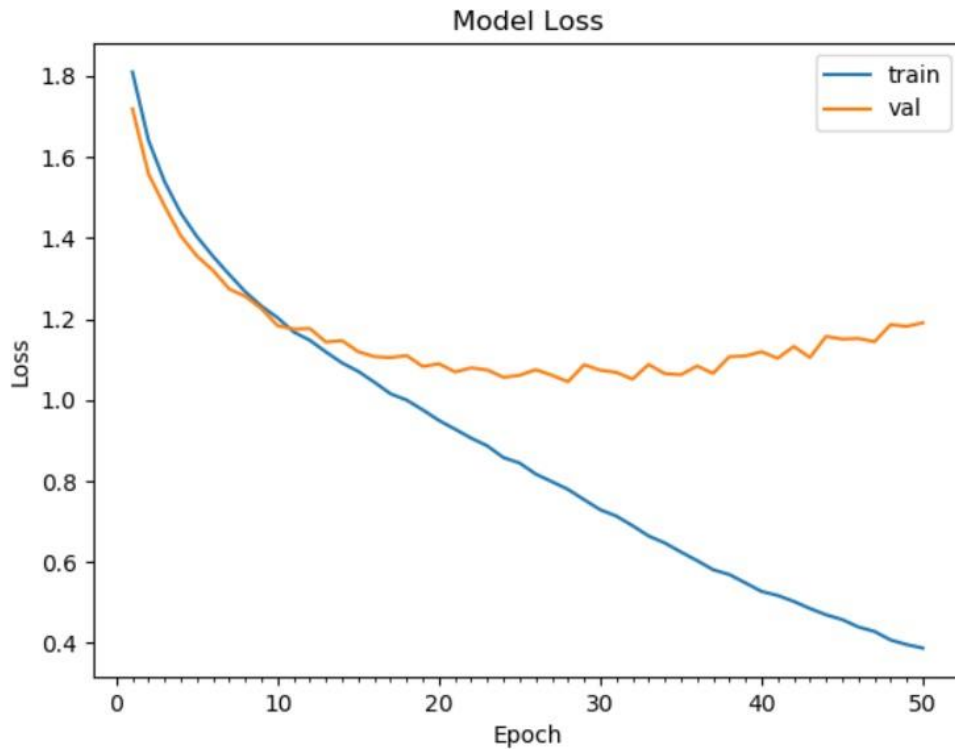


Рисунок 3.7 Тест втрат

І далі застосуємо метод `.evaluate()`, що Повертає значення втрат і значення показників для моделі в тестовому режимі.

Він дорівнює 0.65 і  $\text{loss} = 1.2$

Перед тим як зробити висновок щодо використаних методів навчання згорткової нейронної мережі, наведемо ще графік точності та втрат навчання в залежності від епох. На (Рис. 3.7) Бачимо, що якість навчання постійно зростає. Щоб її покращити – треба збільшити кількість епох. А щодо втрат(Рис. 3.6), то тут майже аналогічно.

Перейдемо до практичного застосування моделі. Нижче буде наведено приклади розпізнавання емоцій обличчя в веб-камери

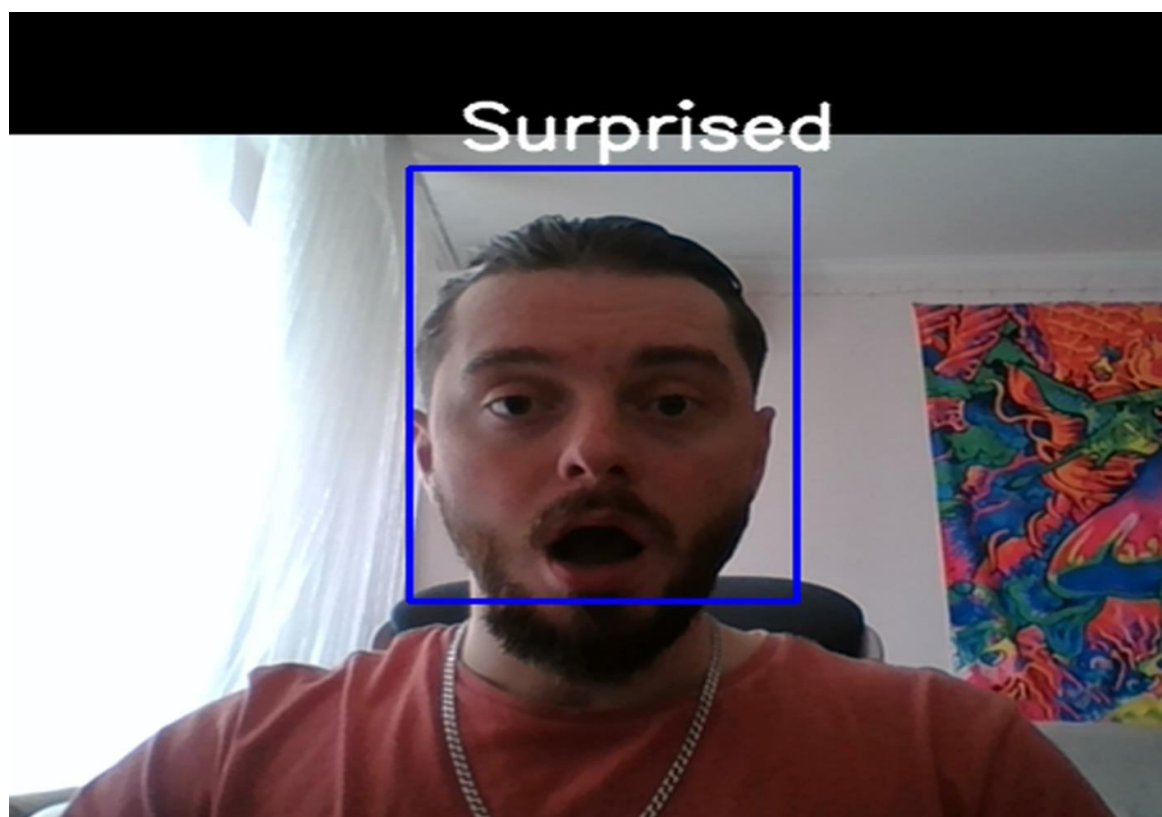


Рисунок 3.8 - Подив

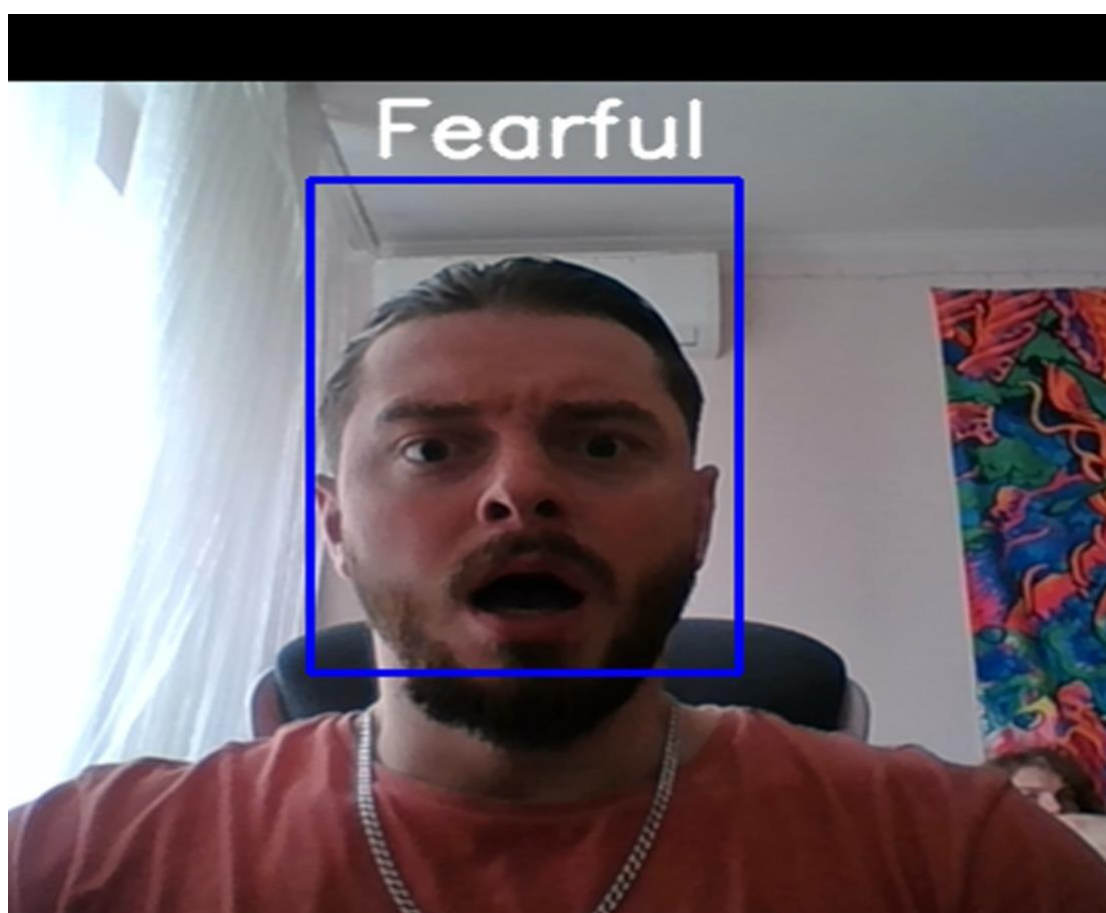


Рисунок 3.9 - Наляканий

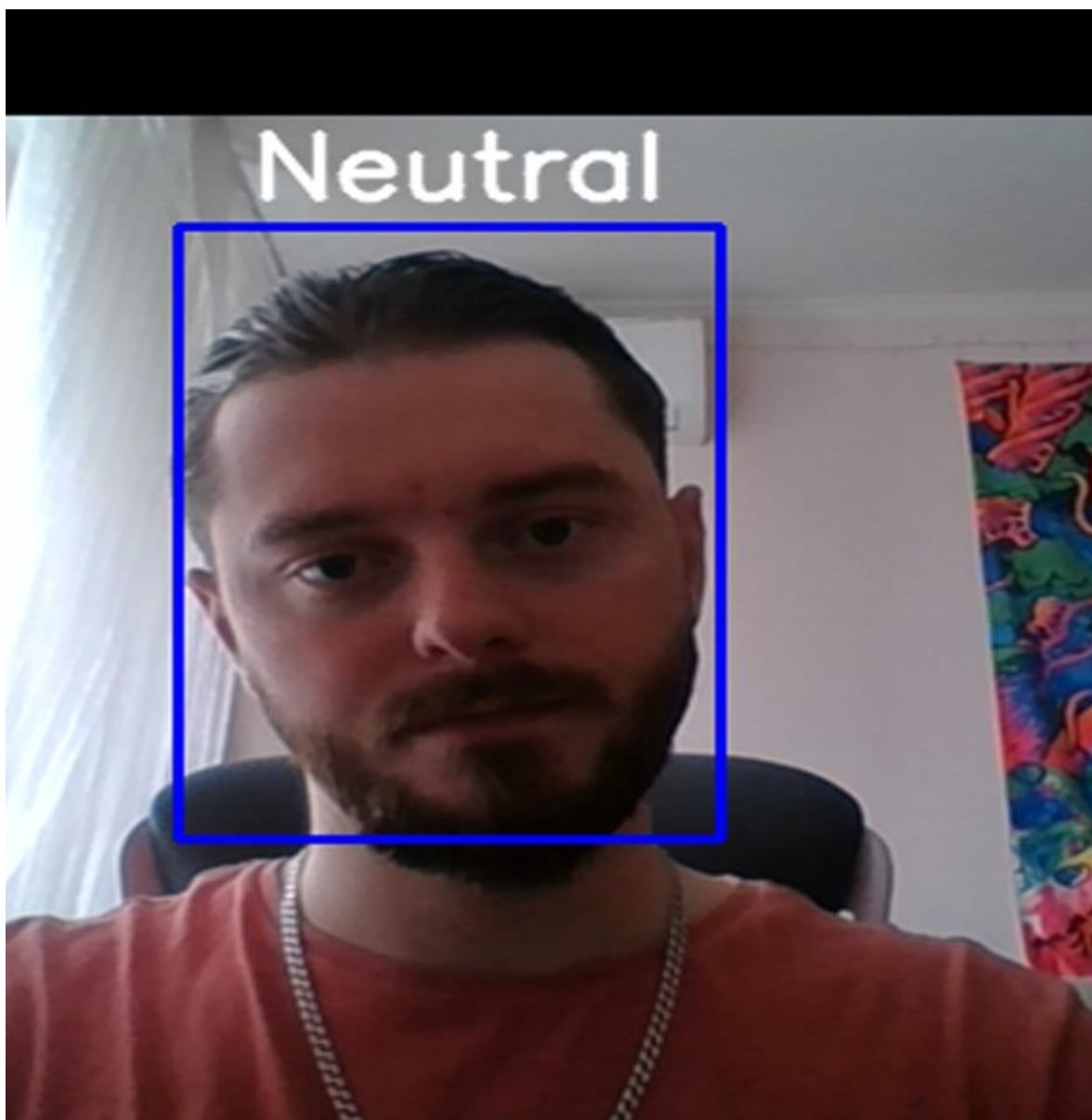


Рисунок 3.8 - Нейтральний

Є можливість розпізнавати декілька об'єктів одночасно (Рис. 3.9)

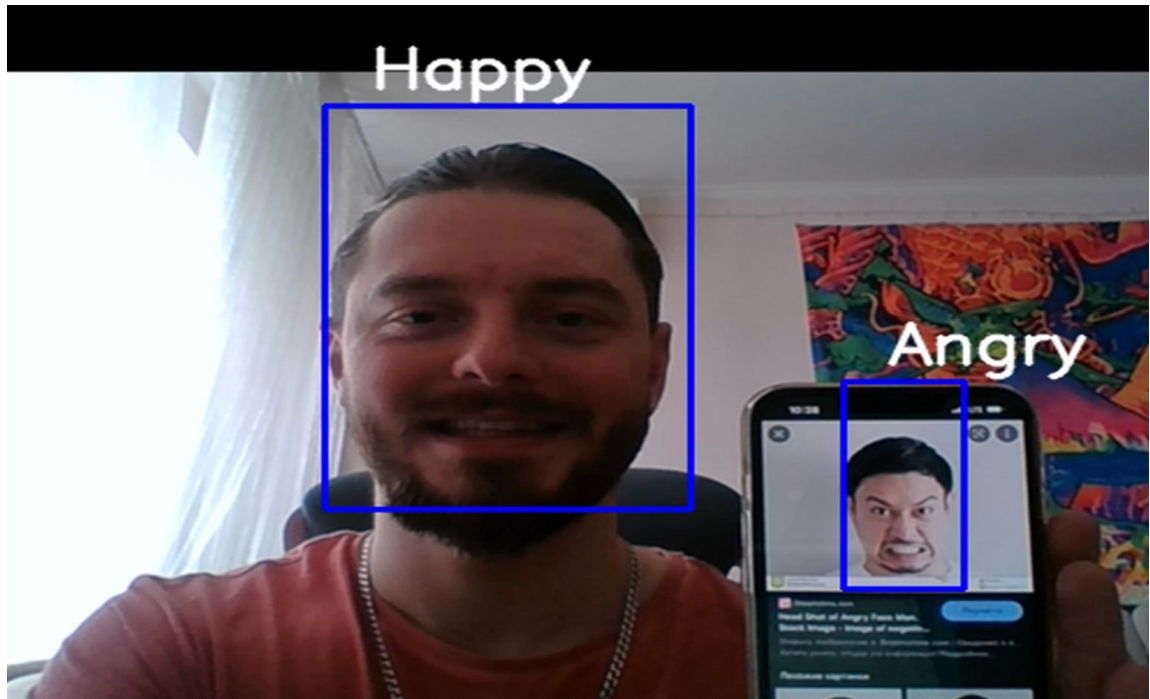


Рисунок 3.9 - Щасливий та злий

### Висновки до третього розділу

В цьому розділі було розглянуто технології розробки та їх обґрунтування для використання. Також було розглянуто переваги та недоліки інструментів для побудови нейронних мереж, препроцесингу і постпроцесингу. Також було розглянуто і побудовано UML-діаграми використання та послідовностей системи, користувача і джерела даних. У цьому розділі також було наведено результати роботи розпізнавання емоцій бличчя. Після проведення аналітичної роботи було визначено, що результат розпізнавання символів є близькими до хорошого, але це також напряму залежить від розміру датасету, шумів зображень тощо.

## ВИСНОВКИ

Під час написання дипломної роботи було розглянуто теоретичні передумови застосування сучасних технологій для розпізнавання емоцій обличчя, зокрема було розглянуто та засвоєно методи розпізнавання емоцій обличчя. Також було проведено огляд конкурентів на ринку, описано їх переваги, недоліки, формати даних з якими вони працюють. Під час проектування системи було розглянуто принципи та методи побудови нейромереж, зроблено огляд найбільш використовуваних шарів штучних нейронних мереж, описано модель CNN, а також процедури препроцесингу і постпроцесингу даних. Варто також зазначити про створення UML-діаграм використання та послідовностей для різних акторів, що допомагає краще оцінити взаємодію системи та користувача. Також було описано технології та інструменти розробки ПЗ, визначено їх переваги та недоліки. У результаті написання практичної частини роботи було розроблено нейромережу та застосована відповідна архітектура з огляду на завдання, визначені оптимальні показники для застосування проміжних шарів нейромереж. Також була проведена порівняльна характеристика результатів навчання та тренування нейромереж. Після результатів роботи було побудовано графіки навчання з подальшим аналізом втрат та виведено показники точності після процесу розпізнавання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Li C., Wang R., Li J., Fei L. Recent trends in intelligent computing, communication and devices. Springer; Singapore: 2020. Face detection based on YOLOv3; pp. 277-284.
2. Ben Krose and Patrick van der Smagt. An introduction to Neural Networks. (1996). – 126.
3. Nguyen H. Fast object detection framework based on mobilenetv2 architecture and enhanced feature pyramid. Journal of Theoretical and Applied Information Technology. 2020; 98 (05).
4. Wang Z., Wang G., Huang B., Xiong Z., Hong Q., Wu H. Chen H. 2020. Face recognition dataset and application. arXiv preprint arXiv: 2003.09093.
5. Інформацією про розпізнавач обличчя FaceReader. Режим доступу: <https://www.abbyy.com/ru-ru/download/facereader/>, вільний.
6. Weng Y., Xia C., A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices. – Mobile Networks and Applications, 2019. –1-22 с.
7. Абраменко, А. Компьютер читает [Электронный ресурс]. — Электрон. текстовые дан. – [Электронный ресурс]. – Режим доступу: <http://www.ocrai.narod.ru/>
8. Белоглазов А.Д. Особенности нейросетевых решений, достоинства и недостатки, перспективы применения. – Т.: Технические науки, 2008. – 105-110 с.
9. Каллан Р. Основные концепции нейронных сетей: Пер. с англ. – М.: издательский дом «Вильямс», 2001. – 291 с.
10. Чередніченко А.О., Шура Н.О, Застосування штучних нейронних мереж як дієвого механізму прийняття ефективних управлінських рішень на підприємстві. – Миколаївський національний університет імені В.О. Сухомлинського, – 2015. – № 4. 628-630 с.
11. Штучні нейронні мережі. – [Електронний ресурс]. – Режим доступу: [http://www.victoria.lviv.ua/html/neural\\_nets/Lecture1.htm](http://www.victoria.lviv.ua/html/neural_nets/Lecture1.htm)

12. Lawrence S., Giles C.L., Tsoi A.C., Back A.D. Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*. 1997;8(1): 98-113.

**ДОДАТКИ****Додаток А****КNU**

Програмне забезпечення розпізнавання текстів у документах  
Software Architecture Document (SAD)

**CONTENT OWNER: Tovstyzhenko Yurii**

**DOCUMENT NUMBER:    RELEASE/REVISION:    RELEASE/REVISION DATE:**

• 1.0

• 1.0

• 14.04.2022

## Table of Contents

<b><u>1.</u></b>	<b><u>Documentation Roadmap</u></b>	<b><u>2</u></b>
	<b><u>1.1.</u></b>	<b><u>Document Management and Configuration Control Information</u></b>
	<b><u>1.2.</u></b>	<b><u>Purpose and Scope of the SAD</u></b>
		<b>3</b>
	<b><u>1.3.</u></b>	<b><u>Viewpoint Definitions</u></b>
		<b>5</b>
	<b><u>1.3.1.</u></b>	<b><u>“Student and engineer” Viewpoint Definition</u></b>
		<b>5</b>
<b><u>2.</u></b>	<b><u>Architecture Background</u></b>	<b><u>5</u></b>
	<b><u>2.1.</u></b>	<b><u>Problem Background</u></b>
		<b>5</b>
	<b><u>2.1.1.</u></b>	<b><u>System Overview</u></b>
		<b>6</b>
	<b><u>2.1.2.</u></b>	<b><u>Goals and Context</u></b>
		<b>6</b>
	<b><u>2.1.3.</u></b>	<b><u>Significant Driving Requirements</u></b>
		<b>6</b>
	<b><u>2.2.</u></b>	<b><u>Solution Background</u></b>
		<b>8</b>
	<b><u>2.2.1.</u></b>	<b><u>Architectural Approaches</u></b>
		<b>8</b>
	<b><u>2.2.2.</u></b>	<b><u>Analysis Results</u></b>
		<b>9</b>
	<b><u>2.2.3.</u></b>	<b><u>Requirements Coverage</u></b>
		<b>9</b>
<b><u>3.</u></b>	<b><u>Referenced Materials</u></b>	<b><u>10</u></b>
<b><u>4.</u></b>	<b><u>Directory</u></b>	<b><u>12</u></b>
	<b><u>4.1.</u></b>	<b><u>Index of Document</u></b>
		<b><u>12</u></b>
	<b><u>4.2.</u></b>	<b><u>Glossary</u></b>
		<b><u>12</u></b>
	<b><u>4.3.</u></b>	<b><u>Acronym List</u></b>
		<b><u>13</u></b>
<b><u>5.</u></b>	<b><u>Sample Figures&amp;Tables</u></b>	<b><u>13</u></b>

## 1. Documentation Roadmap

The Documentation Roadmap should be the first place a new reader of the SAD begins. But for new and returning readers, it is intended to describe how the SAD is organized so that a reader with specific interests who does not wish to read the SAD cover-to-cover can find desired information quickly and directly.

Sub-sections of Section 1 include the following.

Section 1.1 (“Document Management and Configuration Control Information”) explains revision history. This tells you if you’re looking at the correct version of the SAD.

Section 1.2 (“Purpose and Scope of the SAD”) explains the purpose and scope of the SAD, and indicates what information is and is not included. This tells you if the information you’re seeking is likely to be in this document.

Section 1.3 (“Viewpoint Definitions”) explains the viewpoints (as defined by IEEE Standard 1471-2000) used in this SAD. For each viewpoint defined in Section 1.5, there is a corresponding view defined in Section 3 (“Views”). This tells you how the architectural information has been partitioned, and what views are most likely to contain the information you seek.

### 1.1. Document Management and Configuration Control Information

- Revision Number: v1.1
- Revision Release Date: 2022/04/14
- Purpose of Revision: 2022/04/15
- Scope of Revision: basic information about system

### 1.2. Purpose and Scope of the SAD

This SAD specifies the software architecture for SmartSafe system. All information regarding the software architecture may be found in this document, although much information is incorporated by reference to other documents.

The software architecture for a system is the structure or structures of that system, which comprise software elements, the externally-visible properties of those elements, and the relationships among them [Bass 2003]. «Externally visible» properties refers to those assumptions other elements can make of an element, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on. This definition provides the basic litmus test for what information is included in this SAD, and what information is relegated to downstream documentation.

Elements and relationships. The software architecture first and foremost embodies information about how the elements relate to each other. This means that

architecture specifically omits certain information about elements that does not pertain to their interaction. Thus, a software architecture is an abstraction of a system that suppresses details of elements that do not affect how they use, are used by, relate to, or interact with other elements. Elements interact with each other by means of interfaces that partition details about an element into public and private parts. Software architecture is concerned with the public side of this division, and that will be documented in this SAD accordingly. On the other hand, private details of elements—details having to do solely with internal implementation—are not architectural and will not be documented in a SAD.

For example, all non-trivial software systems are partitioned into implementation units; these units are given specific responsibilities, and are the basis of work assignments for programming teams. This kind of element will comprise programs and data that software in other implementation units can call or access, and programs and data that are private. In large projects, the elements will almost certainly be subdivided for assignment to sub-teams. This is one kind of structure often used to describe a system. It is a very static structure, in that it focuses on the way the system's functionality is divided up and assigned to implementation teams.

Other structures are much more focused on the way the elements interact with each other at runtime to carry out the system's function. Suppose the system is to be built as a set of parallel processes. The set of processes that will exist at runtime, the programs in the various implementation units described previously that are strung together sequentially to form each process, and the synchronization relations among the processes form another kind of structure often used to describe a system.

None of these structures alone is the architecture, although they all convey architectural information. The architecture consists of these structures as well as many others. This example shows that since architecture can comprise more than one kind of structure, there is more than one kind of element (e.g., implementation unit and processes), more than one kind of interaction among elements (e.g., subdivision and synchronization), and even more than one context (e.g., development time versus runtime). By intention, the definition does not specify what the architectural elements and relationships are. Is a software element an object? A process? A library? A commercial product? It can be any of these things and more. These structures will be represented in the views of the software architecture that are provided in Section 3.

Behavior. Although software architecture tends to focus on structural information, behavior of each element is part of the software architecture insofar as that behavior can be observed or discerned from the point of view of another element. This behavior is what allows elements to interact with each other, which is clearly part of the software architecture and will be documented in the SAD as such. Behavior is documented in the element catalog of each view.

### **1.3. Viewpoint Definitions**

The software is written in Python using open technology solutions and frameworks. The approach proposed in the thesis uses deep learning, TensorFlow Numpy and OpenCV. The developed model can be used to recognize facial emotions. The developed approach involves the selection of individual options from images with faces to express emotions. The developed software does not use data sets with a lot of noise or flaws, but they are available in small quantities for clarity of analytical recognition results.

### **1.3.1 “Student and engineer” Viewpoint Definition**

This viewpoint summarizes the views of a student developing this project and their concerns. The main concern of a student is to get the normal grade. Technologies that were used: Python, Tensorflow, Numpy.

## **2. Architecture Background**

### **2.1. Problem Background**

Today, there is a great demand for recognizing facial emotions through the camera. Therefore, the widespread use of scanning tools has led to the active development of methods for recognizing certain objects in images. This has led to the development of optical facial recognition systems that allow you to automatically analyze and analyze any emotion.

#### **2.1.1. System Overview**

The main essences of the system (Fig. 1) are:

- User – the basic essence of the user;
- Dataset – the basic essence of the dataset of images;
- Training – model training module;
- OpenCV, Tensorflow – external libraries.

#### **2.1.2. Goals and Context**

The object of research is the recognition of facial emotions as a valuable source of information with the help of new information technologies.

The subject of the study is the features of recognizing facial emotions as a valuable source of information through a webcam using a software module.

The aim of the study is to recognize facial emotions as valuable information using a software module through a webcam.

#### **2.1.3. Significant Driving Requirements**

The facial emotion recognition system allows you to digitize a person's emotions with sufficient accuracy.

From this point of view, the system can be represented by the following scheme of precedents (Fig. 2) and sequence diagram (Fig. 3). This diagram shows the relationship between actors and precedents in the system.

#### Description of the precedent chart

##### Actors:

- user;
- system

##### Precedents:

##### 1st block of precedents:

- Input image representation

These precedents are grouped together because they interact with the data source and the user at the beginning and end of the chain of system precedents.

##### 2nd block of precedents:

- reading data from the data set;
- preliminary processing and preparation of the data set;
- neural network training;
- recognition based on test data set;
- issuance of results of facial emotion recognition;

These precedents are grouped together because they are all actually executed by the system and represent a result to the user.

## **2.2. Solution Background**

Pattern recognition is one of the most fundamental problems in the theory of intelligent systems. On the other hand, the task of pattern recognition is of great practical importance.

Facial emotion recognition tasks consist of:

- segmentation;
- recognition;

- further processing;

Offline facial emotion recognition systems are less accurate than online systems, as only spatial information is available for offline systems, and both spatial and temporal information is available for online systems. One promising approach, however, is to evaluate dynamic trajectory information from offline handwriting and then use online recognition algorithms.

### **2.2.1. Architectural Approaches**

The input image is provided to the CNN layers. These levels of CNN learn to extract the necessary functions from the input image. Each layer represents three operations, namely convolution, activation and generation of reduced size.

Initially, the convolution operation uses a primary filter for the first two layers and a primary filter for the last three layers of the input image. The activation operation is then performed using a nonlinear rectified unit (ReLU). Finally, the reduced size of the input image version is generated by an aggregation layer that defines individual areas of the image. In each layer, the height of the image decreases, and channels are added to the display function to create output support. Then the average loss is used to train the neural network. Then the trained model gets used to recognizing the entered image.

Performance maneuvers include the use of hybrid datasets and experimentation with different activation functions, as well as increasing the number of neural network layers.

### **2.2.2. Analysis Results**

The basic data set consists of more than 36,000 images. The images used were obtained from the following open sources: FKI, Kaggle. Achieved accuracy of handwriting recognition separately using CNN is about 99.2%. When it comes to recognizing facial emotions, it all depends on the data set. When using images for training in 30 thousand and testing in 5 thousand. Recognition accuracy decreased to 70.01%.

### **2.2.3. Requirements Coverage**

All experimental tests were performed on a laptop with Intel i3-6100U (2.3 GHz), 8 GB of RAM and 2 GB of video memory. The following components must be installed: TensorFlow, OpenCV, Numpy.

## **3. Referenced Materials**

1. Li C., Wang R., Li J., Fei L. Recent trends in intelligent computing, communication and devices. Springer; Singapore: 2020. Face detection based on YOLOv3; pp. 277-284.
2. Ben Krose and Patrick van der Smagt. An introduction to Neural Networks. (1996). – 126.
3. Nguyen H. Fast object detection framework based on mobilenetv2 architecture and enhanced feature pyramid. Journal of Theoretical and Applied Information Technology. 2020; 98 (05).
4. Wang Z., Wang G., Huang B., Xiong Z., Hong Q., Wu H. Chen H. 2020. Face recognition dataset and application. arXiv preprint arXiv: 2003.09093.
5. Інформацією про розпізнавач обличчя FaceReader. Режим доступу: <https://www.abbyy.com/ru-ru/download/facereader/>, вільний.
6. Weng Y., Xia C., A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices. – Mobile Networks and Applications, 2019. –1-22 с.
7. Абраменко, А. Компьютер читает [Электронный ресурс]. — Электрон. текстовые дан. – [Электронный ресурс]. – Режим доступу: <http://www.ocrai.narod.ru/>
8. Белоглазов А.Д. Особенности нейросетевых решений, достоинства и недостатки, перспективы применения. – Т.: Технические науки, 2008. – 105-110 с.
9. Каллан Р. Основные концепции нейронных сетей: Пер. с англ. – М.: издательский дом «Вильямс», 2001. – 291 с.
10. Чередніченко А.О., Шура Н.О, Застосування штучних нейронних мереж як дієвого механізму прийняття ефективних управлінських рішень на підприємстві. – Миколаївський національний університет імені В.О. Сухомлинського, – 2015. – № 4. 628-630 с.
11. Штучні нейронні мережі. – [Електронний ресурс]. – Режим доступу: [http://www.victoria.lviv.ua/html/neural\\_nets/Lecture1.htm](http://www.victoria.lviv.ua/html/neural_nets/Lecture1.htm)

12. Lawrence S., Giles C.L., Tsoi A.C., Back A.D. Face recognition: A convolutional neural-network approach. IEEE Transactions on Neural Networks. 1997;8(1): 98-113.

## **4. Directory**

### **4.1. Index**

### **4.2. Glossary**

Convolutional neural networks (CNNs) are a subset of machine learning, and they underlie deep learning algorithms. They consist of layers of nodes containing an input layer, one or more hidden layers and an output layer. Each node connects to the other and has a corresponding weight and threshold. If the output of any single node exceeds the specified threshold, this node is activated by sending data to the next level of the network. Otherwise, the data is not transmitted to the next network layer. (Fig. 4)

It is assumed that the alignment of input and target data is "many to one", which limits the length of the target sequence so that it must be the length of the input data.

Recurrent Neural Networks (RNN) is a state-of-the-art algorithm for serial data. This is the first algorithm that remembers its input through internal memory, making it ideal for machine learning tasks that involve sequential data. This is one of the algorithms behind the amazing achievements of deep learning over the past few years.

### **4.3. Acronym List**

- CNN - Convolutional Neural Network
- RNN - Recurrent Neural Network
- ML - Machine Learning
- AI - Artificial Intelligence
- NN - Neural Network
- OpenCV - Open Source Computer Vision Library

## **5. Sample Figures & Tables**

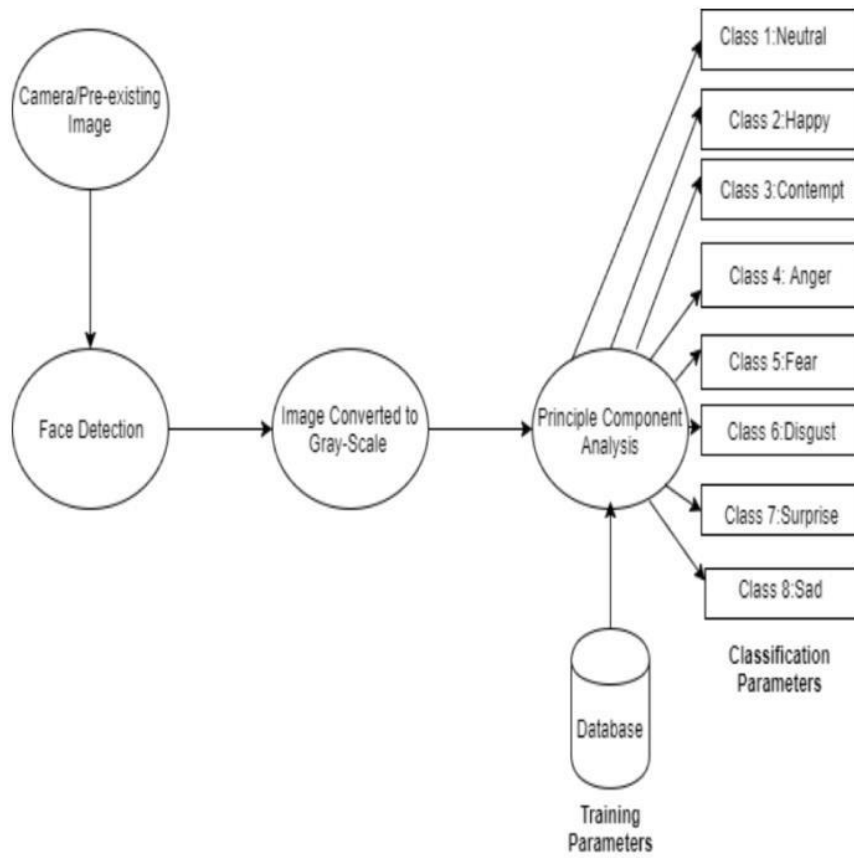


Figure 1 - Block diagram of emotion recognition methodology

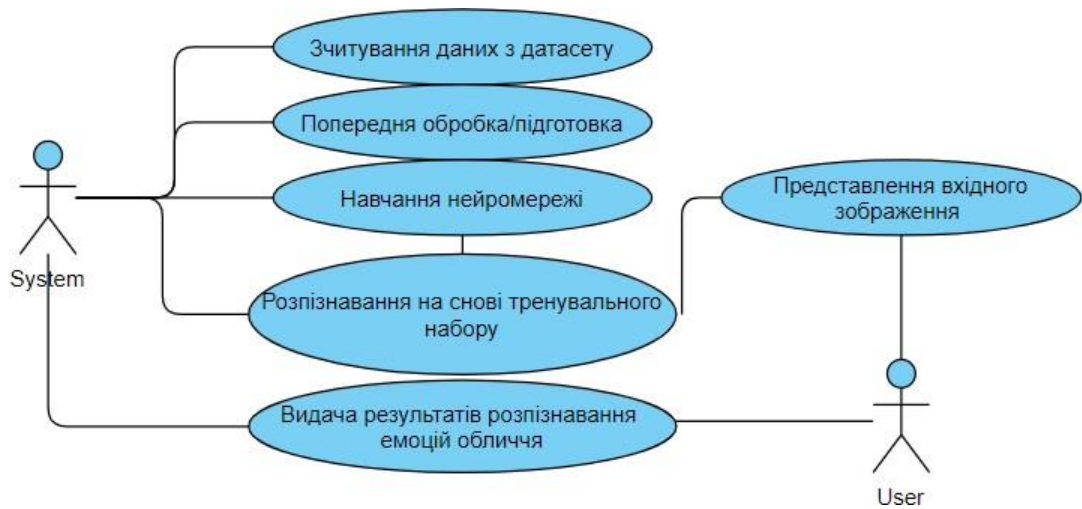


Figure 2 - Use Case diagram

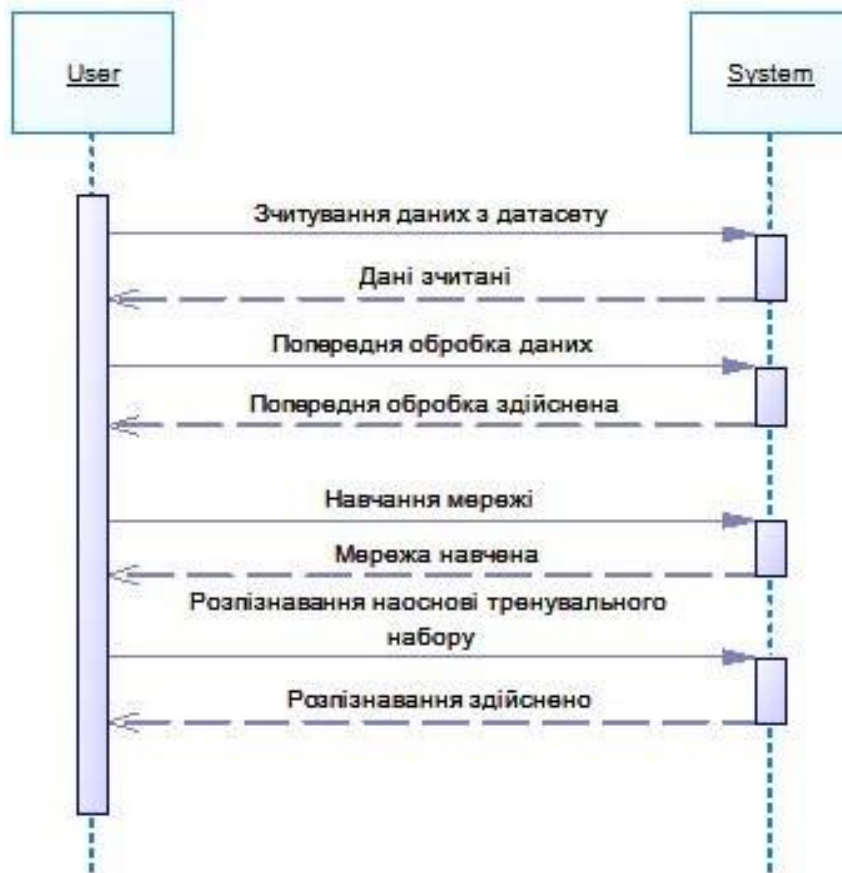


Figure 3 - Sequence diagram

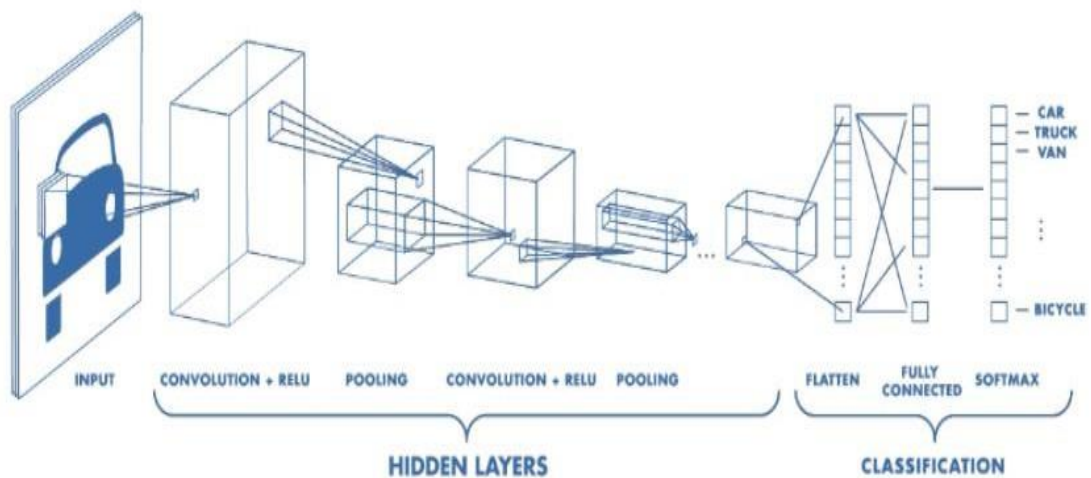


Figure 4 - Convolutional Neural Network principle