Міністерство освіти і науки України
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА
Кафедра прикладних інформаційних систем

122 Комп'ютерні науки
Освітня програма «Прикладне програмування»

Кваліфікаційна робота бакалавра

на тему:

ВЕБ-ЗАСТОСУНОК РОЗПІЗНАВАННЯ ОБЛИЧЧЯ

Виконав студент 4 курсу групи ПП−43

_____
(Підпис)
Богдановський Д. В.
(прізвище, ім'я, по батькові)

_____

Керівник проф., д.т.н. Домрачев В. М.
(прізвище, ім'я, по батькові)

_____

_____
(Резолюція «До захисту»)

Попередній захист:

_____
(Висновок: "До захисту в екзаменаційній комісії ")

Завідувач кафедри _____ Плескач В.Л. _____
(Підпис) (Прізвище, ініціали) (Дата)

Засвідчую, що у цьому курсовому проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2021 року

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Номер	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	26.10.2020	
2.	Видача завдання кваліфікаційної роботи бакалавра	23.11.2020	заява
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	01.12.2020	
4.	Затвердження плану кваліфікаційної роботи бакалавра	18.02.2021	
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	25.02.2021	
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	05.03.2021	
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	09.04.2021	
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	07.05.2021	
9.	Подання роботи у першому варіанті	11.05.2021	
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	12.05.2021	
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	**24.05.2021**	
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедру	28.05.2021	
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботу)	11.06.2021	
14.	Захист кваліфікаційної роботи бакалавра	25.06.2021	

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЕКТУ

Складові частини дипломного проекту	Обсяг,арк.
Титульний аркуш	1
Завдання до дипломного проекту(календарний план проекту)	3
Відомість дипломного проекту	1
Пояснювальна записка до дипломного проекту	1
Анотація	1
Анотація(іноземною мовою-англійською)	1
Зміст	2
Перелік умовних позначень,символів,одиниць,скорочень і термінів	1
Вступ	2
1 Загальнотеоретичні питання	17
2 Програмне та апаратне забезпечення. Архітектура системи	11
3 Програмна та апаратна реалізація системи «Розпізнавання облич»	4
Висновки	1
Перелік посилань	1
Додатки	10

				ДП ХХХХ 00.000.00		
	ПІБ	Підп.	Дата			
Розробн.	Богдановський			Відомість дипломного проекту	Лист	Листів
Керівн.	Домрачев				1	48
Н/контр.						
Зав.каф.	Плескач					

Анотація (реферат)

Дипломна робота: 80 с., 20 рис., 16 джерел, 2 дод.

Метою дипломної роботи є забезпечення ефективного пошуку обличь на зображеннях та відео на основі створення веб-сервісу, реалізованого засобами PHP, HTML, JS, CSS.

Для досягнення мети вирішено такі завдання:

- дослідити процес розробки веб-сервісів у цілому;
- проаналізувати особливості створення веб-сервісів за допомогою PHP;
- проаналізувати існуючі методи створення веб-сервісів;
- розробити веб-сервіс пошуку культурних заходів міста.

Об’єкт дослідження.

Об’єктом дослідження є веб-сервіс культурних заходів міста.

Предмет дослідження.

Предметом дослідження є засоби створення веб-сервісу культурних заходів на мові PHP.

Методи дослідження.

Методом дослідження є системний аналіз і синтез щодо особливостей розробки веб-сервісів.

Наукова новизна результатів дослідження.

Наукова новизна результатів дослідження полягає в комплексному проведенні аналізу процесу розробки веб-сервісів та можливості надання користувачам зручного веб-сервісу для розпізнавання обличь.

Практичне значення та значущість роботи.

Розроблений веб-сервіс дозволяє використовувати сервіс для пошуку обличь на відео та фото.

Ключові слова: веб-сервіс, PHP, MVC, HTML, CSS, SQL, JS, Bootstrap**.**

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

БД(англ Database) – база даних.

ТЗ – технічне завдання.

ПЗ – програмне забезпечення.

SDLC – життєвий цикл розробки системи.

MVC – патерн розробки ПЗ.

PC – персональний комп’ютер.

Компіляція – це процес переводу коду у зрозумілий для комп’ютера, після чого код зможе бути виконаний.

Адаптивний дизайн – дизайн сайту, який коректно відображається на будь-якому розмірі екрану. Сайт має певні точки(break points), при яких сайт змінює свій вид згідно файлу стилей.

Чутливий дизайн – дизайн сайту, схожий на адаптивний дизайн, проте має одну відмінність. Тоді як адаптивний дизайн міняє форму тільки на певних точках, чутливий дизайн постійно міняє свої розміри, немовби резиновий.

SQL запит – звертання до бд(бази данних) для отримання, зміни або повного видалення інформації з неї.

SQL (*Structured Query Language*) – спеціальна мова запитів до бд(бази данних). Мета цієї мови є управління базами даних Фреймворк – вже створений програмний код, що надає можливість використовувати його функції для користі розробника.

Бібліотека – готові методи та функції, що можуть використовувати розробники для полегшення та пришвидшення своєї роботи. Відрізняється від фреймворку тим, що бібліотека підключається до проекту, а фреймворк є основою для програмування.

PHP (HyperText Preprocessor) – це особлива скриптова мова яка працює на стороні сервера, яка використовується для розробки статичних веб-сайтів або динамічних веб-сайтів або веб-програм.

HTML (HyperText Markup Language) – спеціальна мова для розмітки. Що використовується для створення сторінок.

JS – набір готових класів, процедур, функцій, структур і констант, що надаються застосунком (бібліотекою, сервісом) чи операційною системою для використання у зовнішніх програмних продуктах.

Open Server – локальний сервер.

NetBeans – середовище розробки, що даэ змогу створювати проекти на мові програмування PHP.

Хеадер – верхня частина сторінки сайту, де знаходиться меню.

Футер – нижня частина сторінки сайту.

Сайдбар – частина сайту, що знаходиться збоку від головної інформації

ВСТУП

Сучасні комп'ютерні технології розвиваються в швидких темпах. Та зараз неможливо представити наше світ без комп'ютерів і мережі Інтернет. Дисципліна комп’ютерний зір розвивається неймовірно швидко, в неймовірних темпах.

Комп'ютерний зір (computer vision) – це програмні та технічні засоби, які дозволяють зчитувати інформацію в цифровій формі відеозображень, обробляти її та повертати результат в формі, для використання в реальному часі спеціальними засобами.

Зараз не існує загальноприйнятого формулювання проблеми комп'ютерного зору. А що навіть ще важливіше, не існує якоїсь типового визначення того, як має вирішуватися питання комп'ютерного зору. На томість, є доволі багато різних методів, що дають можливість вирішувати різні, певні завдання комп'ютерного зору. Ці методи доволі часто залежать від того, які завдання необхідно виконати і дуже рідко вони можуть бути загальними для багатьох застосувань. Велика кількість методів та програм, які все ще перебувають в положенні фундаментальних досліджень, та з часом все більша кількість методів, які починають використовувати в комерційних продуктах, де вони дуже часто складають велику частину системи, яка може вирішувати велику кількість завдань.

У переважної більшості практичних завдань комп'ютерного зору комп'ютери попередньо запрограмовані для вирішення окремих задач, однак, методи, засновані на знаннях, стають все більш загальними. Розпізнавання образів вважається сферою, що використовує різноманітні методи отримання інформації з відеопотоку, і в основному, базуються на статистичному підході. Істотна частина цієї області присвячена фактичним використанням цих методів.

Складність завдання розпізнавання осіб можна обґрунтувати наступними причинами: обличчя людини - це динамічний об'єкт, який має високу ступінь змінності в зовнішньому вигляді (за формою і кольором шкіри); різні параметри освітлення, певні типом і напрямком джерела світла, фрагментарне перекриття персон іншими предметами сцени; необхідність локалізації і визначення осіб, що знаходяться в довільному положенні в просторі. Однак існуючі системи локалізації і розпізнавання осіб не завжди враховують дані особливості, що не дозволяє досягти прийнятного рівня розпізнавання на зображеннях і відеопослідовність

1.1 Огляд сучасних методів розпізнавання осіб

По суті, роботу будь-якого алгоритму розпізнавання можна описати діаграмою на малюнку 1.1:

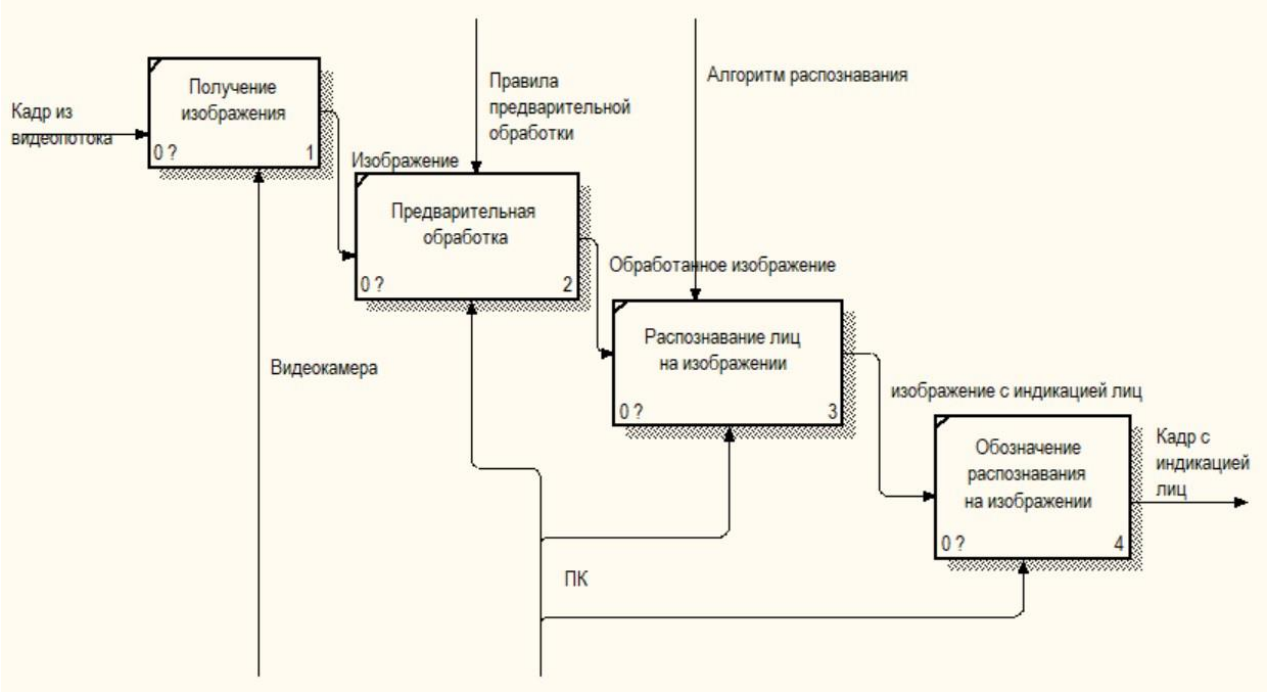

Рисунок 1.1 - Декомпозиція загальної діаграми розпізнавання осіб з відеопотоку

Контекстна діаграма представлений в додатку А.

Існуючі алгоритми виявлення осіб можна розбити на чотири категорії:

- емпіричний метод;

- метод характерних симетричних ознак;

- розпізнавання за допомогою шаблонів, заданих розробником;

- метод знаходження за певними зовнішніми ознаками, які навчають системи. Емпіричний підхід основується на «знаннях зверху-вниз»

(Knowledge based top-down methods) передбачають реалізацію алгоритму, що реалізує певний набір правил, яким повинен відповідати ділянку зображення, щоб можна було визнати його обличчям людини. Створення таких правил є спосіб формалізувати емпіричні знання про те, як саме має виглядати обличчя людини на зображеннях. У підсумку по ним визначається: є особа на ділянці зображення чи ні.

Найпростіші правила:

- присутня значна різниця в яскравості між центральною частиною і верхньою частиною особи;

- яскравість і колір центральній частині особи є однорідними;

- кардинально відрізняються за яскравістю щодо певної частини персони, два симетрично розташованих очі, ніс і рот

Для згладжування перешкод застосовується метод сильного зменшення зображення піддає зображення зменшенню в розмірах. Це також зменшує обчислювальні операції (рисунок 1.2). Метод сприяє більш простому виявлення зони рівномірного розподілу яскравості (зона передбачуваного знаходження особи), щоб в подальшому виконати перевірку на наявність сильно відрізняються по яскравості областей всередині: саме такі області можна з різною часткою ймовірності віднести до «особі».

Рисунок 1.2 - Метод Yang & Huang

Метод побудови гістограм користується вертикальної і горизонтальної гістограмами (рисунок 1.3). В областях-кандидатах відбувається пошук рис

обличчя. При отриманні гістограми певної форми можна визначити ймовірність наявності осіб.

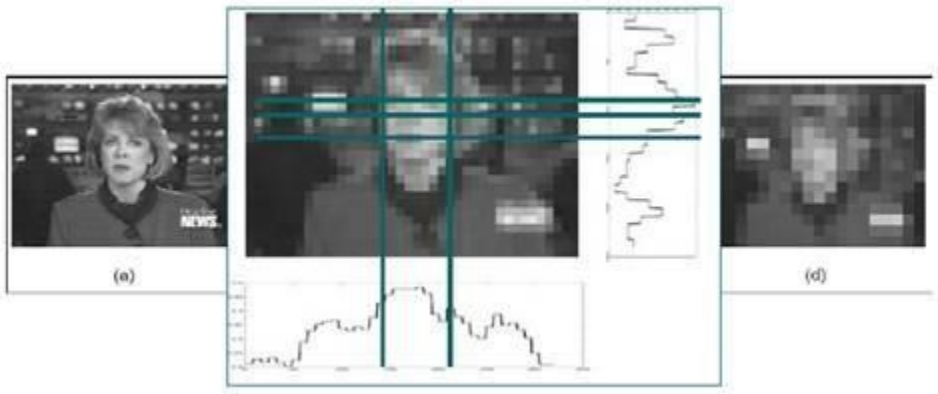

Рисунок 1.3 - Метод Kotropoulos & Pita

Даний підхід використовувався на зорі розвитку комп'ютерного зору через малі вимог до обчислювальної потужності процесора для обробки зображення.

Описані методи мають непогані показники по визначенню осіб на зображеннях на однорідному фоні. Їх легко реалізувати за допомогою машинного коду, що дозволило розробити безліч подібних алгоритмів. Недоліком є їх абсолютна непридатність при наявності складного заднього фону і чутливість до нахилу та певного повороту голови.

Методи деяких певних симетричних признаків, що основуються на знаннях знизу-вгору (Feature invariant approaches) утворюють другу групу методів визначення об'єктів. Тут видно інший підхід до проблеми: не відбувається формалізації протікають в людському мозку процесів в явному вигляді. Прихильники цього підходу намагаються знайти інваріантні особливості, виявити неявні закономірності та властивості об'єктів, без огляду на кут нахилу і положення.

Головні етапи алгоритмів з певної групи методів:

- виявлення: кордону особи, форма, яскравість, текстура, колір;
- знаходження на малюнку певних ознак персони: очі, ніс, ріт;
- об'єднання всіх знайдених інваріантних ознак і їх верифікація.

У складних сценах передбачається пошук правильних геометричний розташування форм особи. Для цього застосовується гауссовский похідний

фільтр з безліччю різних масштабів і орієнтацій. Слідом випадковим перебором виконується пошук відповідності виявлених рис обличчя, їх взаємне розташування.

Суть методу угруповання ознак с застосуванням другої похідної гауссовского фільтра для пошуку цікавлять областей зображення (Рисунок 1.4). Після цього групуються краю навколо кожної такої області за допомогою порогового фільтра.

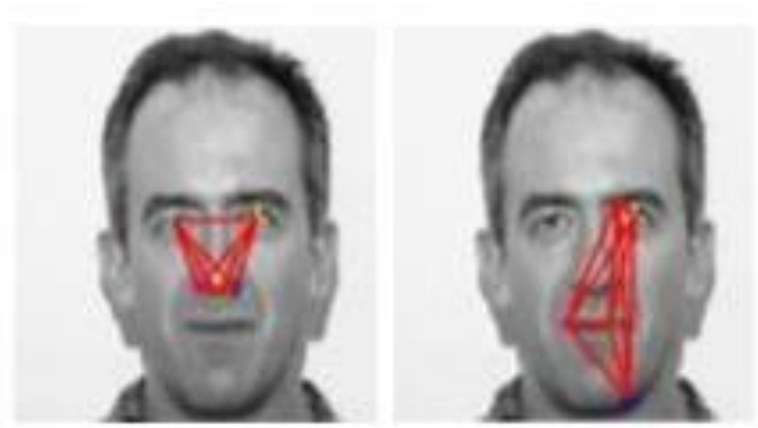

Рисунок 1.4 - Вірні і випадкові спрацьовування

Для комбінування знайдених ознак використовується оцінка за допомогою байєсівської мережі. Відбувається вибірка рис обличчя. У цій групі методів є ряд недоліків. Великий вплив має складний задній фон зображення, при якому можуть виникати проблеми з виявленням. При невеликому захаращених особи іншими об'єктами, засветке або виникненні шумів відсоток достовірного розпізнавання також сильно падає. Основа цих підходів - Емпірика, це одночасно є їх сильною та слабкою частиною. В даному випадку виявлення об'єктів на зображенні відноситься до завдань високої складності через наступних факторів: велика мінливість об'єкта розпізнавання, залежність від освітлення, умов зйомки.

Застосування емпіричних правил дозволяє звести задачу розпізнавання об'єктів на зображенні до певної кількості відносно простих перевірок. Однак ці методи головної категорії до тих пір поки дуже далекі по ефективності від вже давно успішно функціонуючого інструменту - людського зору, оскільки дослідники, які вирішили стати на цей, стикаються з низкою серйозних труднощів. По-перше, процеси, що протікають в людському мозку далеко не

повністю вивчені, і набір емпіричних знань, який на даний момент доступні на свідомому рівні, далеко не вичерпує весь спектр підсвідомих інструментів. По-друге, неможливо перенести неформальний людський досвід в набір певних правил, тому що в ряді випадків це може привести до великої кількості помилкових спрацьовувань, або навпаки - виявлення взагалі не відбудеться.

Розпізнавання з використанням шаблонів, визначених розробником (Template Matching Methods). Шаблони вирішують який дефолтний образ особи, наприклад, шляхом опису певних основних властивостей певних частин персони і їх певного взаємного розташування. Виявлення персони з використанням шаблону основується на перевірці кожної з частин зображення на схожість певним шаблонам. Особливості підходу:

- два види шаблонів:

а) не деформуються; б) деформуються;

- шаблони заздалегідь запрограмовані, необучаеми;

- використовується кореляція для визначення особи на малюнку чи відео.

Метод визначення особи з використанням тривимірних форм передбачає використання шаблонів у вигляді пар відносин яркостей в двох областях. Для детекції обличчя потрібно просканувати всі картинки на схожість із визначеним шаблоном. Це необхідно робити з різним масштабом (рисунок 1.5).

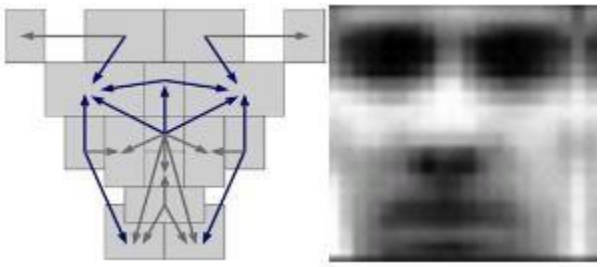

Рисунок 1.5 - Метод детектування особи

Їх корисна особливість для методу - здатність виділити формузмінних об'єктів в певних межах навчального сету з певною кількістю визначених параметрів. Ця параметризація являється компактною та точною,також вона може використовуватися для створення ефективно працюючих систем класифікації.

Серед достоїнств розпізнавання за допомогою шаблонів можна виділити відносну простоту реалізації і хороші результати спрацьовування для зображень з нескладним заднім фоном. Основний недолік цього методу - необхідність калібрування шаблону поблизу із зображенням обличчя. Доцільність методу не вважається високою через складність обчислення шаблонів для різних поворотів особи і ракурсів.

Як показує практика пошук персон на картинках за допомогою певних методів, які створенні на побудові певної математичної моделі зображення персони, основується на повному переборі всіх фрагментів зображення різних розмірів та створення перевірки повністю кожного з шматків на знаходження персони. Через те, що повний перебір має дуже великі недоліки, та має надзвичайно велику обчислювальну складність. Автори застосовують різні методи зменшення фрагментів, які необхідно перебрати.

Перелік більшості принципи методів:

- Схоластика: всі фрагменти скануються вікном і показуються векторами цінності.

- Блокова структура: зображення розділяється на багато пересічних та непересічних ділянок (Рисунок 1.6) будь яких масштабів і робиться оцінка використовуючи алгоритм оцінки ваг векторів.

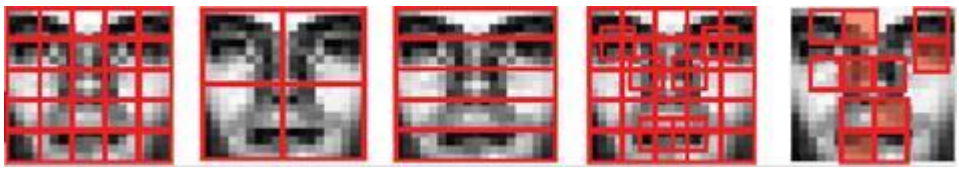

Рисунок 1.6 - Приклади розбивки зображення на ділянки

Для навчання алгоритмів потрібно бібліотека вручну підготовлених зображень облич і «не осіб», будь-яких інших зображень.

Також необхідно зазначити, що найголовнішим завданням являється виділення сильних класифікаторів. Саме ці класифікатори мають найголовніший пріоритет перевірки відшуканих фрагментів в картинці. Кількість більш слабких класифікаторів треба прибирати за рахунок подібності

один на одного, та прибирання класифікаторів, що створили за рахунок шумових викидів.

Перелічимо основні методики виконання цих завдань:

- Штучні нейронні мережі (Neural network: Multilayer Perceptions);
- Метод головних компонент (Principal Component Analysis (PCA));
- Порівняння шаблонів (Template Matching).
- Приховані Марковские моделі (Hidden Markov model);
- Метод гнучкого порівняння на графах (Elastic graph matching)
- Поєднання ФА і методу головних компонент (Mixture of PCA, Mixture of factor analyzers);
- Розріджена мережу вікон (Sparse network of windows (SNoW));
- Активні моделі (Active Appearance Models);
- Адаптоване поліпшення і заснований на ньому Метод Віоли- Джонса та ін.

У методі Віоли-Джонса використовується алгоритм сканування вікна. Виглядає він у такий спосіб: на досліджуваному зображенні встановлюється вікно сканування в початкове положення, обрані використовувані ознаки. Вікно послідовно рухається по зображенню з кроком в 1 клітинку (припустимо, розмір самого вікна є 24 * 24 осередки).

Проводиться послідовне сканування для різних масштабів. Масштабується саме скануючий зображення. Всі відшукані фрагменти відправляються до класифікатору, який «визначає». Обчислювати всі ознаки на малопотужних ПК просто нереально, тому класифікатор повинен реагувати на строго певний набір ознак. Абсолютно логічно, що треба навчити класифікатор з даного набору. Це проводиться автоматично.

1.2 Недоліки існуючих систем

Недоліки методів характерних симетричних ознак: при несуттєве перекритті особи іншими предметами, появу шумів або засветке відсоток коректного розпізнавання суттєво знижується. Також сильно впливає складний

задній фон.

Недоліки методів розпізнавання за допомогою шаблонів: істотним недоліком є те, що потрібно відкалібрувати шаблон поблизу із зображенням обличчя. Доцільність використання також під сумнівом тому необхідно обчислювати шаблони для різних ракурсів і поворотів особи.

Недоліки нейронних мереж: при додаванні нового еталонного особи в БД потрібно повне перенавчання мережі для всього наявного набору (процедура в залежності від розміру вибірки може досягати декількох днів). Проблеми навчання математичного характеру: це виявлення локального оптимума, знаходження кроку, який буде оптимально підходити для моделі, перенавчання і т. д. Э складності при формалізуванні етапів вибору найоптимальнішої архітектурної мережі (треба знайти оптимальну кількість нейронів, шарів, та характер зв'язків). Виходячи з вище викладеного, можна зробити висновок, що нейронна мережа - «чорний ящик» з важко інтерпретуються результатами роботи.

Недоліки методів головних компонентів (Principal Component Analysis, PCA): метод вимагає для свого застосування ідеальних умов таких як: строго одних параметрів точної освітленості, cepelній вираз обличчя та відсутність певних перешкод на обличчі таких як очки та борода.

Недоліки методу порівняння шаблонів (Template Matching): недоліком є те, що потрібно багато ресурсів як для порівняння ділянок зображень, так і для зберігання. Використовується найпростіший алгоритм порівнянь. Це накладає обмеження на вихідні зображення - повинні бути зняті в строго встановлених умовах. Не допустимі помітні зміни ракурсу, освітлення, емоційного вираження.

Недоліки прихованих Марковських моделей (CMM, HMM):

- необхідність підбирати параметри моделі для кожної окремої бази даних;

- у алгоритму не має розрізняючої здатності, тобто алгоритм розвитку тільки робить більше відгук кожного малюнку на власну модель, але не робить менше відгук на всі інші моделі.

Недоліки методу гнучкого порівняння на графах (Elastic graph matching):

- велика складність для обчислення функції розпізнавання;

- не велика здатність до запамятовування нових шаблонів;

- лінійна залежність часу роботи від розміру бази даних осіб.

У підсумку серед основних недоліків існуючих систем можна виділити наступні:

1) вплив навколишнього середовища і світла;

2) споживання великої кількості ресурсів;

3) проблематичне обслуговування;

4) необхідність еталонного вигляду розпізнається об'єкта;

5) повільна швидкість розпізнавання.

1.3 Бібліотека OpenCV

Це спецілаьна бібліотека яка спеціалізується на комп'ютерному зорі та машинному навчанні. Ця бібліотека має відкритий вихідний кід. В цю бібліотеку входять більше за 3000 алгоритмів, в які входять як класичні, так і багато сучасних алгоритмів для комп'ютерного зору та машинного навчання. OpenCV має інтерфейси на багатьох мовах, наприклад: Java, Python, Matlab та C ++.

Крім платформ і підтримки багатьох мов програмування, які дозволяють використовувати додатки на різних системах, бібліотека OpenCV вельми ефективна (в порівнянні з іншими схожими бібліотеками) з точки зору обчислень, так як майже всі функції і оператори в ній векторізованні.

Головні модулі OpenCV:

Спеціальне cxcore – ядро. Воно має основні структури алгоритмів та даних:

- основні методи, які використовують для роботи над багатовимірними числовими масивами

- матрична алгебра, генератор випадкових чисел, та математичні функції

- Видалення, обновлення та запис структур даних в XML та з XML

- основні функції для графіки 2D

CV – це спеціальний модуль для обробки малюнків та комп'ютерного зору

- головні операції над малюнками (геометричні перетворення,

перетворення колірних просторів, фільтрація і т. Д.)

- аналіз зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми)

- аналізація руху, спостереження за об'єктами

- модуль для виявлення об'єктів на зображенні та осіб

- встановлення калібрації камер та елементів встановлення просторової структури

Highgui – це спеціальний модуль який використовується для внесення/ відображення малюнків та відео, та відображення інтерфейсу користувача.

- внесення відео з відеокамер та з спеціальних відео файлів, зчитування / запис зображень.

- спеціальні методи призначених для організацій спеціального інтерфейсу (усі додатки використовують HighGUI)

Cvaux – функції яким ще необхідне доопрацювання та старі функції

- просторовий зір: відповідає за калібрацію

- знаходження відповідностей та кліків в графах

- пошук на зображеннях рис обличчя

CvCam - захоплення відео

- відповідає за здійснення захоплення відео з камер (цей модуль більше не підтримуються, в більш пізніх версіях цього модуля вже не має)У версії 2.2 бібліотека була реорганізована. Замість універсальних модулів cxcore, cvaux, highGUI і інших було створено кілька компактних модулів з більш вузькою спеціалізацією.

Одна з найпопулярніших бібліотек, на данний час. Вона має аж 5 мільйонів скачувань (це без завантажень з репозиторію), та нещодавно ця бібліотека запропонована для основи нового стандарту Khronos.

РОЗДІЛ 2 РОЗРОБКА АЛГОРИТМУ РОЗПІЗНАННЯ ОБЛИЧ

2.1 Використання OpenCV

Метод був розроблений в 2001 році Полом Віолою і Майклом Джонсом і до сих пір – це основа для знаходження об'єктів на малюнкі в реальному часі.

Цей метод базується на таких принципах:

- для швидкого обчислення об’єктів використовується зображення в спеціальному інтегральному уявленні;

- використання признаків Хаара, за якими відбувається пошук потрібного об'єкта (в даному випадку, персони та рис обличчя);

- використання бустінгу (покращення та підсилення) для знаходження усіх ознак об'єкта на певній частині малюнку чи відео;

- усі ознаки, що надходять на вхід класифікатора, дають висновок «Вірно» чи «хибно»;

- для швидокого відкриття віконе використовуються особливі каскади ознак, де особа не знайдена.

Результати пошуку дуже швидкі, хоча самонавчання класифікаторів відбувається вкрай повільно. Тому цей метод був обраний для розпізнавання об'єктів на зображенні. Функція Віоли-Джонса на данний час - це один з найкращих методів по ефективності розпізнавання та швидкість роботи. Також необхідно додати, що цей детектор має не велику можливість спрацювати випадково. Цей метод добре працює та знаходить тригерні риси, навіть коли має невеликий кут. Кут може бути навіть до 30 градусів. Але коли кут нахилу є вище значення, виявлення стає гіршим. При звичайній реалізації алгоритму відсутність необхідного класифікатора не дозволяє знаходити повернені риси людини під будь яким кутом, що сильно ускладнює використання цього алгоритму в виробничих системах з урахуванням зростаючих потреб. Потрібен докладний розбір принципів, на яких базується алгоритм Віоли-Джонса. В загалі цей метод знаходить людей та ознаки обличчя за деякими принципами скануючого вікна.

2.2 Принцип скануючого вікна

Принцип скануючого вікна в загальному вигляді виробляє виявлення особи на зображенні таким чином:

1) Це певне зображення з об’єктами які треба знайти. Це зображення представляється у вигляді матриці з пікселів w*h, де кожен має значення:

- ці значення можуть бути від 0 до 255, для чорно-білого зображення;

- та 0 - 2553, для кольорового зображення (палітра R, G, B).

2) Як результат своєї роботи, алгоритм повинен визначити риси осіб та відобразити їх. Знаходження проводять в певній активній області малюнку чи відео використовуючи прямокутні ознаки Хаара, які мають описувати знайдену персону та риси обличчя:

rectanglei = {x, y, w, h, a}, (2.1)

x та y – це координати прямокутника де w - ширина;

h - висота;

a - кут нахилу прямокутника до вертикальної осі зображення.

Інакше кажучи, стосовно до малюнків і фотографій використовується метод скануючого вікна (scanning window): кожен фрагмент зображення сканується вікном. При кожному переміщенні вікна для кожного положення застосовуються класифікатори. Система навчання класифікаторів являється автоматизованою та не потребує участі людини, це одна з причин, чому цей підхід є дуже швидким. Основною метою пошуку та знаходження персон на малюнках та відео зображення використовуючи даний метод, який зазвичай є одним з багатьох кроків на великій дорозі до розпізнавання рис обличчя. Одною з таких проблем є верифікація людей по для розпізнавання персон чи обличчя.

2.3 Інтегральне позначення зображення

Для виконання будь-яких дій з даними використовується інтегральне представлення зображень за методом Віоли-Джонса. Також таке уявлення можна зустріти і в інших методах, таких як SURF, вейвлет-перетворення і багатьох інших. Інтегральне уявлення швидко дозволяє розрахувати загальну яскравість

довільного прямокутника на даному зображенні. Причому час розрахунку завжди константне.

Інтегральне представлення зображення - це матриця, яка збігається за розмірами з вихідним зображенням. Кожен її елемент зберігає суму групування всіх фрагментів, що розполягається зліва чи трохи вище певного елемента. Більшість частин матриці розраховують за формулою:

$L(x,y) = \Sigma i \leq x, j \leq y * I(i, j)$, де $I(i, j)$ - колір пікселя зображень.

Всі елементи матриці – це сумма пікселей в деякому квадраті від (0,0) по (x, y), мається на увазі значення абсолютно всіх пікселів (x, y) є рівнем сумі значень кожного пікселя, що знаходиться зліва та вище цього пікселя (x, y). Вираховування матриці виконується за певний час, що відповідає числу пікселів на малюнку, саме цьому інтегральне відображення розраховується з першого проходу.

Ми розраховуємо матрицю за такою формулою:

$L(x, y) = I(x,y)-L(x-1,y-1)+L(x,y-1)+L(x-1, y)$

Певна інтегральна матриця може дуже швидко обчислювати суму пікселів довільного прямокутника і довільної площі.

Нехай в прямокутнику ABCD (рисунок 2.1) є цікавий для нас об'єкт D:

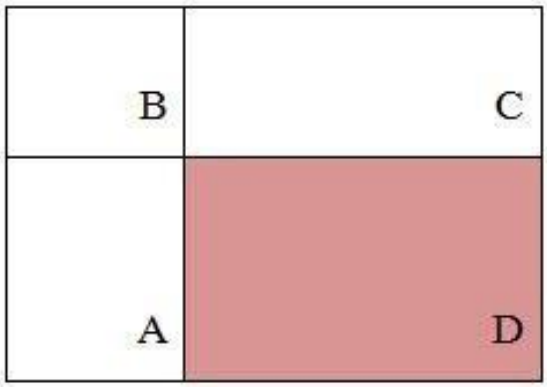

Рисунок 2.1 – Прямоугольник ABCD

З малюнка ми дізнаємося, що суму в середені квадрата ми виражаємо за допомогою суми та різниці суміжних прямокутників за формулою 2.4:

$S (ABCD) = L (A) + L (C) - L (B) - L (D)$. (2.4)

Ця формула використовується для обчислення суми пікселів

прямокутника.

2.4 Признаки Хаара

Ознака - відображення f: X => DF, де DF – велика кількість особливих функцій ознак. Але при умові, що ознаки рівні f1, ..., fn, тоді вектор ознак x = (f1 (x), ..., fn (x)) називається признакова описом об'єкта x∈X. Основний признак опису можна порівнювати з самими об'єктами. Враховуючи це X = Df1 * ... * Dfn називають просторі ознаки.

Ці признаки ми можемо поділити на декілька типів у залежності від кількості Df:

- це певний бінарний признак, DF ={0,1};

- номінальний признак: DF;

- порядковий признак: DF – остання впорядкована множина;

- кількісний признак: DF - безліч дійсних чисел. По дефолту, є прикладні завдання з різними ознаками, для вирішення цих ознак підходять далеко не всі методи. В класичному методі Віоли - Джонса використовуються квадратні ознаки, та вони називаються примітивами Хаара.

Використовуючи розширений метод Віоли - Джонса, який є у бібліотеці OpenCV та зазвичай використовує додаткові ознаки.

Обчислюваним значенням цих ознак - F = X - Y, де X – певна сума значень кольору певних точок перекривається яскравою частиною ознак, а Y - значення кольорові точки закриваються темними ознаками. Для обчислення зазвичай використовується розуміння інтегрального зображення, розглянуте вище. Ознаки Хаара дають точкове значення перепаду яскравості по осі X і Y відповідно.

2.5 Сканування вікна

Алгоритм скануючого вікна з ознаками Хаара виглядає наступним чином:

- є досліджуване зображення, вікно сканування на початковій позиції, які використовуються ознаки встановлені;

- на кожному кроці вікно сканування переміщається по зображенню з кроком в 1 піксель (наприклад, розмір вікна може бути 28 * 28 пікселів);

- при скануванні області у вікні обчислюються близько 200 тис. Різних варіантів розташування ознак за рахунок зміни положення і масштабу у вікні сканування;

- сканування здійснюється послідовно при різних масштабах;

- зображення не масштабується, робить це скануючий вікно (змінюється розмір осередку);

- всі ознаки, які були знайдені на ділянці зображення, передаються класифікатором, який виносить рішення.

Рисунок 2.2 – Візуалізація алгоритму скануючого вікна

Процес пошуку по черзі виробляє обчислення всіх ознак, чого просто неможливо досягти на звичайних домашніх комп'ютерах. Отже, класифікатору потрібно реагувати на строго певну безліч ознак. Виглядає логічним, що для знаходження осіб класифікатор потрібно навчити визначати ознаки, характерні тільки для них. Це досягається автоматичним навчанням.

2.6 Модель машинного навчання

Машинне навчання - процес отримання автоматичним модулем нових знань. Є визнане визначення даного процесу: машинне навчання - це наука, що

вивчає комп'ютерні алгоритми, автоматично улучшающиеся під час роботи. Рисунок 2.3 ілюструє процес навчання машини.

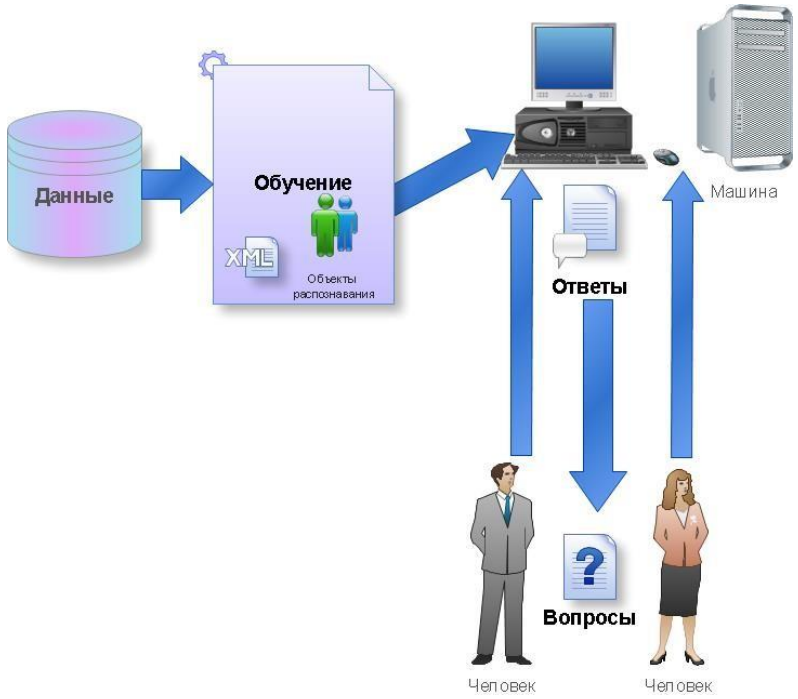

Рисунок 2.3 – Модель машинного навчання

Даний процес входить в концепцію і технологію під назвою Data mining (витяг інформації та інтелектуальний аналіз даних), куди входять крім машинного навчання такі дисципліни, як «Теорія баз даних», «Штучний інтелект», «Алгоритмізація», «Розпізнавання образів та інші» .

Машинне навчання в методі Віоли-Джонса вирішує таку задачу як класифікація шляхом навчання каскадного класифікатора.

2.7 Навчання класифікатора в методі Віоли-Джонсона

В контексті алгоритму, є безліч зображень, які розділені на класи. Вони задаються безліччю зображень для певного класу, до якого вони відносяться (наприклад, це може бути клас «фронтальне положення очей»). Таке безліч називається навчальною вибіркою. Для інших об'єктів не визначена класова приналежність. Потрібно побудувати особливий алгоритм, що може класифікувати певні об'єкти з стартової множини.

Класифікування об'єктів – це значить, що необхідно вказати певний id (або

найменування класу), до якого можна віднести цей об’єкт.

Класифікація об'єкта - id чи назва певного класу, який видається певним алгоритмом класифікації, для конкретного об'єкту.

Класифікатор (classifier) – використовується для задач класифікації - це апроксимуюча функція, яка виносить необхідні рішення, та визначає до якого класу він належить.

Вибірка для навчання – це певні данні для навчання.

Машинне навчання відносить завдання класифікації до спеціального розділу навчання з спеціальним учителем, яке ділить класси. Класифікація зображень та сигналів – це і є розпізнання. При використанні методу Віоли-Джонса для розпізнання особи та ідентифікації иа класифікація є двухклассовий.

Класифікацію можна описати таким чином: є X – певний массив, який зберігає об'єкти, Y – це певна кінцева кількість номерів, які є у класів. Між цими классами має бути залежність - відображення Y *: X => Y. Навчальна вибірка представлена формулою:

Xm = {(x1, y1), ..., (xm, ym)}. (2.5)

Будується спеціальна функція m яка використовує вектор ознак X, який надає відповідь на будь яке можливе спостереження X та може класифікувати об'єкт x∈X. Це стандартне правило має без проблем працювати на будь яких данних.

2.8 Навчання класифікатора в методі Віоли-Джонсона

Для вирішення проблеми складного навчання існує технологія бустінга.

Бустінг (підвищення, підняття) - це клас методів машинного навчання, заснований на ідеї, що комбінація простих класифікаторів (отриманих слабким учнем) може працювати краще, ніж будь-який з простих класифікаторів. Слабкий учень (WL) - це алгоритм навчання, здатний виробляти класифікатори з ймовірністю помилки строго (але незначно) менше випадкового вгадування (0.5, в двійковому випадку). З іншого боку, сильний учень (SL) здатний (враховуючи достатню кількість навчальних даних) давати класифікатори з

довільно малою вірогідністю помилки.

Ансамбль (або комітет) класифікаторів - це класифікатор, побудований на якійсь комбінації слабких учнів. Стратегія підвищення та ансамблі класифікаторів, полягає в тому, щоб навчити багато слабких класифікаторів і якимось чином об'єднати їх, замість того, щоб намагатися отримати один сильний класифікатор.

Бустінг (англ. Boosting) мета-алгоритм машинного навчання. Основною ідеєю бустінга комбінування слабких функцій, які будуються в ході ітеративного процесу, де на кожному кроці нова модель навчається з використанням даних про помилки попередніх. Сильний навчальний алгоритм класифікатором, добре корелює з вірною класифікацією, на відміну від слабкого. Нарівні з бустінгом в мета-навчанні також розглядають такі поняття, як беггінг (англ. Упаковка) і стекінг(англ. Укладання). Беггінг, на відміну від бустінга, використовує паралельне навчання базових класифікаторів. Стекінг ж комбінує результати різних алгоритмів, отримуючи тим самим більш точну відповідь. Одним з недоліків бустінга те, що він може приводити до побудови громіздких композицій, що складаються з сотень алгоритмів. Такі композиції виключають можливість зберігання оперативної інтерпретації, вимагають обсягів пам'яті для базових алгоритмів і істотних витрат часу на обчислення класифікацій. Багато алгоритми класифікації мають саме таку структуру: спочатку обчислюються оцінки приналежності об'єкта класах, потім вирішальне правило переводить ці оцінки в номер класу. Значення оцінки, як правило, характеризує ступінь впевненості класифікації.

Якщо дані зображення, що містять різні відомі в світі об'єкти, класифікатор може бути навчений на основі них для автоматичної класифікації об'єктів в майбутніх невідомих зображеннях. Прості класифікатори, побудовані на основі деяких ознак зображення об'єкта, зазвичай виявляються малоефективними в класифікації. Використання методів бустінга для класифікації об'єктів - шлях об'єднання слабких класифікаторів спеціальним чином для поліпшення загальної можливості класифікації.

Класифікація ознак є типовою завданням комп'ютерного зору, де визначається, чи містить зображення деяку категорію об'єктів чи ні. Ідея тісно пов'язана з розпізнаванням, ідентифікацією та виявленням. Класифікація по виявленню об'єкта зазвичай містить виділення ознак, навчання класифікатора і застосування класифікатора до нових даних. Є багато способів уявлення категорії об'єктів, наприклад з аналізу форми, за допомогою моделі «мішок слів», за допомогою локальних описателей, таких як SIFT [5], і так далі. Прикладами класифікаторів з учителем служать наївні байєсовські класифікатори [на 28.01.19 не створений], методи опорних векторів [на 28.01.19 не створений], суміш Гауссіан і нейронні мережі. Однак дослідження показали, що категорії об'єктів і їх положення в зображеннях можуть бути виявлені також за допомогою навчання без учителя.

Ідея даного підходу - це розробка більш досконалого алгоритму бустінга AdaBoost (adaptive boosting – еволюційне поліпшення), що в перше використали в 1999 році Йоав Фройнд (Freund) і Робертом Шапіро (Schapire, він має юзати величесзна кількість класифікаторів та навчається на певному спеціальному наборі данних для навчань, чергою застосовуючи ці функції на всіх етапах.

Як приклад подивимось задачу для класифікації на декілька класів, Y = {-1+1}. Для прикладу, базових алгоритмів має повертати всього декілька відповідей -1 і та+1, а також основне правило яке є фіксованим: C (b) = sign (b). Певна композиція має виглядати так:

a (x) = C (F (b1 (x), ..., bT (x)) = sign (ΣT

a1b1 (x)), x ∈ X.(2.7)

Функція якості цієї композиції Qt має вигляд числа помилок, який допускаються нею на навчальній вибірці:

Q (b, W1) = QT = Σ1

[Yi ΣT

atbt (xi)] <0, (2.8)

де Wl = (w1, ..., wl) - вектор ваг об'єктів.

Візуалізація алгоритму представлена в додатку Б. Плюси AdaBoost:

- не погана загальна здатність. В справжніх прикладах у всіх випадках практично завжди будуються моделы, що краще за якістю початкових алгоритмів.Загальна здатність може покращуватися в міру збільшення кількості базових алгоритмів;

- казуальність реалізації;

- власні постійні витрати бустінга малозначні. Час налаштування композиції прямо повністю определяется часом навчання базових алгоритмів;

- можливість виявити об'єкти, які є шумовими викидами. Це найбільш «важкі» об'єкти xi, для яких в процесі нарощування композиції ваги wi приймають максимальні значення.

Проблеми AdaBoost:

«Найбільш значних» об'єктів, на яких не вірно роблять багато базові алгоритми. Але саме ці об'єкти найчастіше виявляються шумовими викидами. В результаті AdaBoost стартує налаштування на шум, що приводить до перенавчання. Проблема вирішується шляхом видалення викидів або застосування менш «агресивних» функцій втрат. Зокрема, застосовується алгоритм GentleBoost;

- AdaBoost вимагає досить довгих навчальних вибірок. Інші методи лінійної корекції, зокрема, беггінг, здатні будувати алгоритми порівнянної якості за меншими вибірках даних;

- буває побудова неоптимального набору базових алгоритмів. Для поліпшення композиції можна періодично повертатися до раніше побудованим

алгоритмам і навчати їх заново;

- бустінг може призводити до побудови громіздких композицій, що складаються з сотень алгоритмів. Такі композиції виключають можливість змістовної інтерпретації, вимагають великих обсягів пам'яті для зберігання базових алгоритмів і істотних витрат часу на обчислення класифікацій.

В наші дні підхід посилення простих класифікаторів є популярним і, ймовірно, найбільш ефективним методом класифікації за рахунок високої швидкості та ефективності роботи і відносну простоту реалізації. Якщо дані зображення, що містять різні відомі в світі об'єкти, класифікатор може бути навчений на основі них для автоматичної класифікації об'єктів в майбутніх невідомих зображеннях. Прості класифікатори, побудовані на основі деяких ознак зображення об'єкта, зазвичай виявляються малоефективними в класифікації. Використання методів бустінга для класифікації об'єктів - шлях об'єднання слабких класифікаторів спеціальним чином для поліпшення загальної можливості класифікації.

Класифікація ознак є типовою завданням комп'ютерного зору, де визначається, чи містить зображення деяку категорію об'єктів чи ні. Ідея тісно пов'язана з розпізнаванням, ідентифікацією та виявленням. Класифікація по виявленню об'єкта зазвичай містить виділення ознак, навчання класифікатора і застосування класифікатора до нових даних. Є багато способів уявлення категорії об'єктів, наприклад з аналізу форми, за допомогою моделі «мішок слів», за допомогою локальних описателей, таких як SIFT, і так далі. Прикладами класифікаторів з учителем служать наївні байєсовські класифікатори [на 28.01.19 не створений], методи опорних векторів [на 28.01.19 не створений], суміш Гауссіан і нейронні мережі. Однак дослідження показали, що категорії об'єктів і їх положення в зображеннях можуть бути виявлені також за допомогою навчання без учителя.

2.9 Каскадна модель розроблюємого алгоритму

Основний алгоритм бустінга, що використовується для пошуку персон

виглядає так:

1) Визначають спочатку най слабкіщі класифікатори.

2) До будь якого переміщення вікна сканування обчислюється прямокутний ознака яка є у будь якому прикладі.

3) Розшукуэться найкраще відповідаючий висмогам поріг для всіх ознаки.

4) Ознаки оцінбється та шукаються кращі ознаки та відповідний поріг.

5) Вибірка оновлюється.

Каскадна модель сильних класифікаторів - це по суті той же дерево вирішення рішень, де кожна гілка дерева побудована таким чином, щоб знаходити майже всі питання, що цікавлять сенсори та відхиляють регіони, які не являються образами. Також, певні вузли дерева знаходяться таким особливим чином, що відповідають ближчим вузлам знаходять до коренів дерева, тим з меншої кількості примітивів він складається і тим самим вимагає меншого часу на прийняттярішення. Даний вид каскадної моделі добре підходить для обробки зображень, на яких загальна кількість детектіруемих образів мало. У цьому випадку метод може швидше прийняти рішення про те, що даний регіон не містить образ, і перейти до наступного. Приклад каскадної моделі сильних класифікаторів представлений на малюнку 2.4:

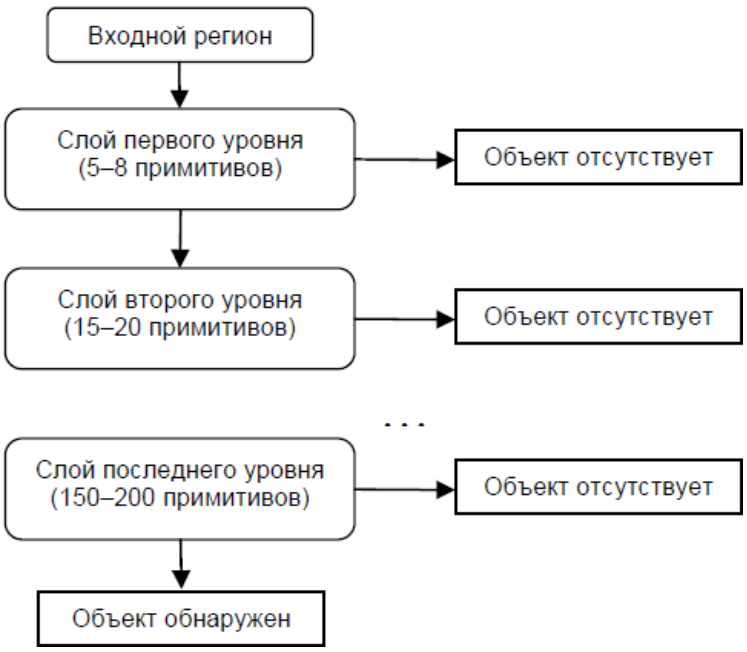

Рисунок 2.4 – Приклад каскадної моделі сильних класифікаторів

Для тренування такого каскаду будуть потрібні наступні дії:

1) задаються значення рівня помилок для кожного етапу (попередньо їх треба кількісно переглянути при застосуванні до зображення з навчального набору) - вони називаються detection і false positive rates - треба щоб рівень detection був високий, а рівень false positive rates низький;

2) додаються ознаки доти, поки параметри обчислюється етапу не досягнуть поставленої рівня, тут можливі такі допоміжні етапи, як:

а. Тестування додаткового маленького тренувального набору.

б. Поріг AdaBoost навмисне знижується з метою знайти більше об'єктів, але в зв'язку з цим якомога більшу кількість неточних визначень об'єктів.

3) якщо false positive rates залишається високим, то додається наступний етап або шар;

4) помилкові виявлення в поточному етапі використовуються як негативні вже на наступному шарі або етапі.

У більш формальному вигляді алгоритм тренування каскаду заданий нижче:

1) Користувач задає значення f (максимально допустимий рівень помилкових спрацьовувань на шар) і d (мінімально допустимий рівень виявлень на шар).

2) Користувач задає цільової загальний рівень помилкових спрацьовувань Ftarget.

3) P - набір позитивних прикладів.

4) N - набір негативних прикладів. 5) F0 = 1,0; D0 = 1,0; i = 0.

6) while (Fi> Ftarget) i = i + 1; ni = 0; Fi = Fi-1 while (Fi = f * Fi-1)

ni = ni + 1 AdaBoost (P, N, ni).Di;

7) Оцінити отриманий каскад на тестовому наборі для визначення Fi і

8) Зменшувати поріг для і-того класифікатора, поки поточний каскад матиме рівень виявлення принаймні d * Di-1 (це ж стосується Fi)

9) N = Ø;

10) Якщо Fi> Ftarget, то оцінити поточний каскад на наборі зображень, які

містять осіб, і додати всі неправильно класифіковані в N.

2.10 Кінцеве представлення алгоритму

Розпізнавання осіб з відеопотоку по методу Віоли-Джонса представлено у вигляді контекстної діаграми (додаток А) і діаграмою декомпозиції (рисунок 2.5)

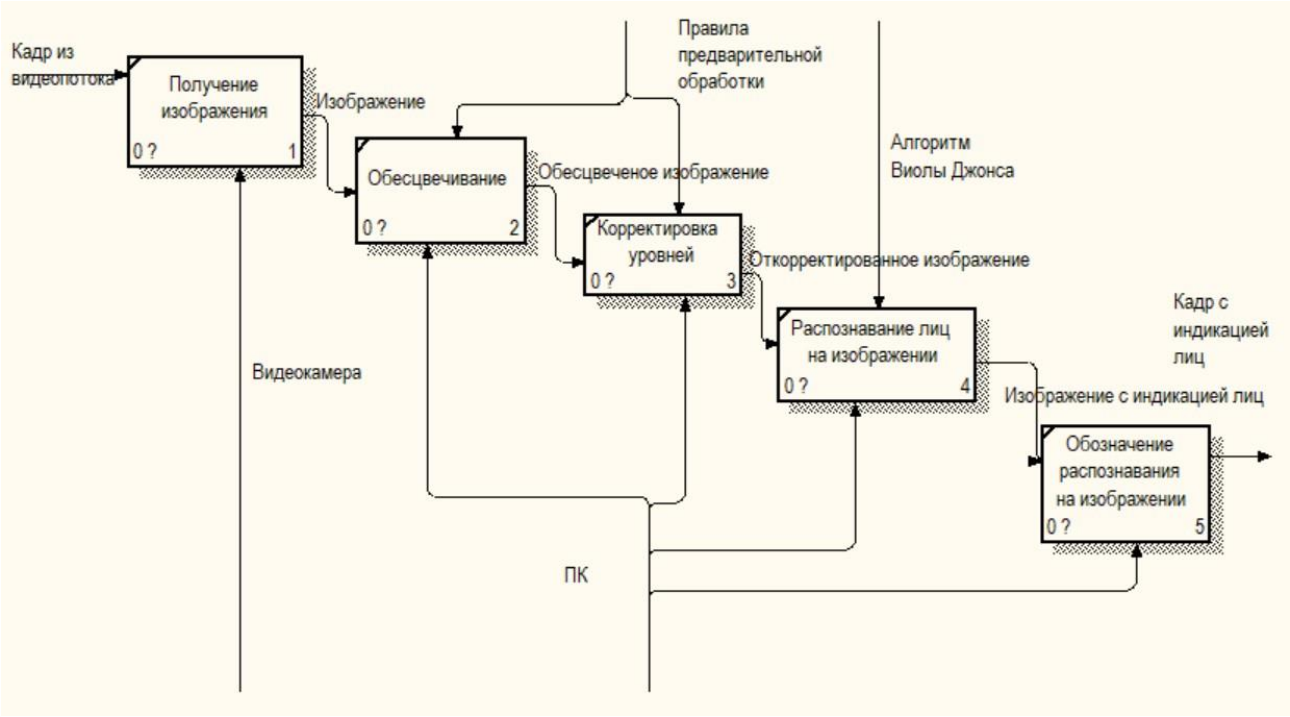

Рисунок 2.5 – Діаграма декомпозиції розпізнавання осіб з відеопотоку по методу Віоли-Джонса

На діаграмі видно, що після отримання зображення воно проходить попередню обробку - «Знебарвлення» і «Коригування рівнів».

Слідом йде безпосереднє розпізнавання методом Віоли-Джонса і виведення зображення з накладеними фігурами індикації осіб.

В кінцевому підсумку процес розпізнавання осіб в відеопотоці буде виглядати наступним чином (рисунок 2.6):

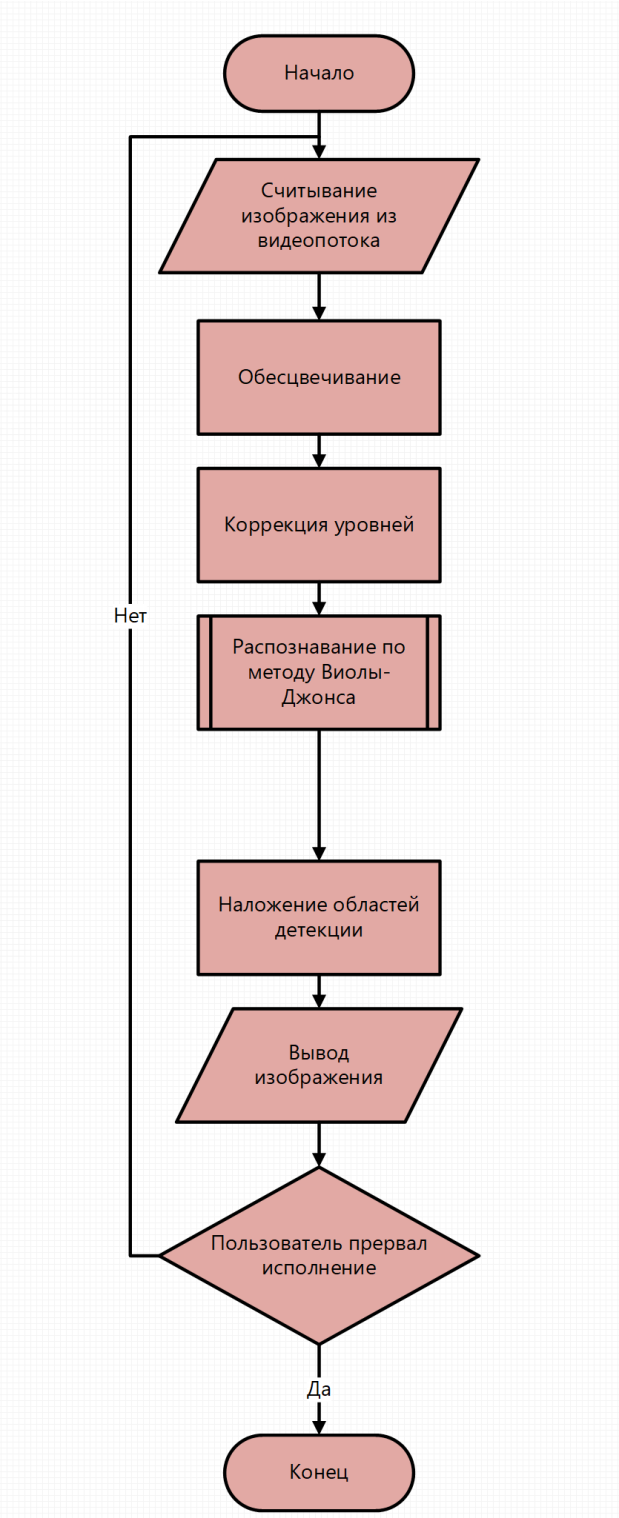

Рисунок 2.6 – Блок-схема алгоритму розпізнавання осіб в відеопотоці

Кожен раз при отриманні зображення з відеопотоку його потрібно привести в належний вигляд для можливості розпізнавання. Відбувається знебарвлення зображення, тому що колір не потрібен при розпізнаванні. Далі коригуються світлові рівні зображення, щоб на розпізнавання в меншій мірі впливала освітленість і знаходження джерела світла. Ключовий блок

розпізнавання за методом Віоли-Джонса виділено на блок- схемою підпрограмою. Якщо алгоритм знайшов особи, то на виході є координати цих осіб на зображенні, і ми можемо відзначити їх. Це зображення показуємо користувачеві і зчитуємо наступний кадр з відеопотоку. На малюнку 2.7 представлено опис алгоритму за методом Віоли-Джонса.

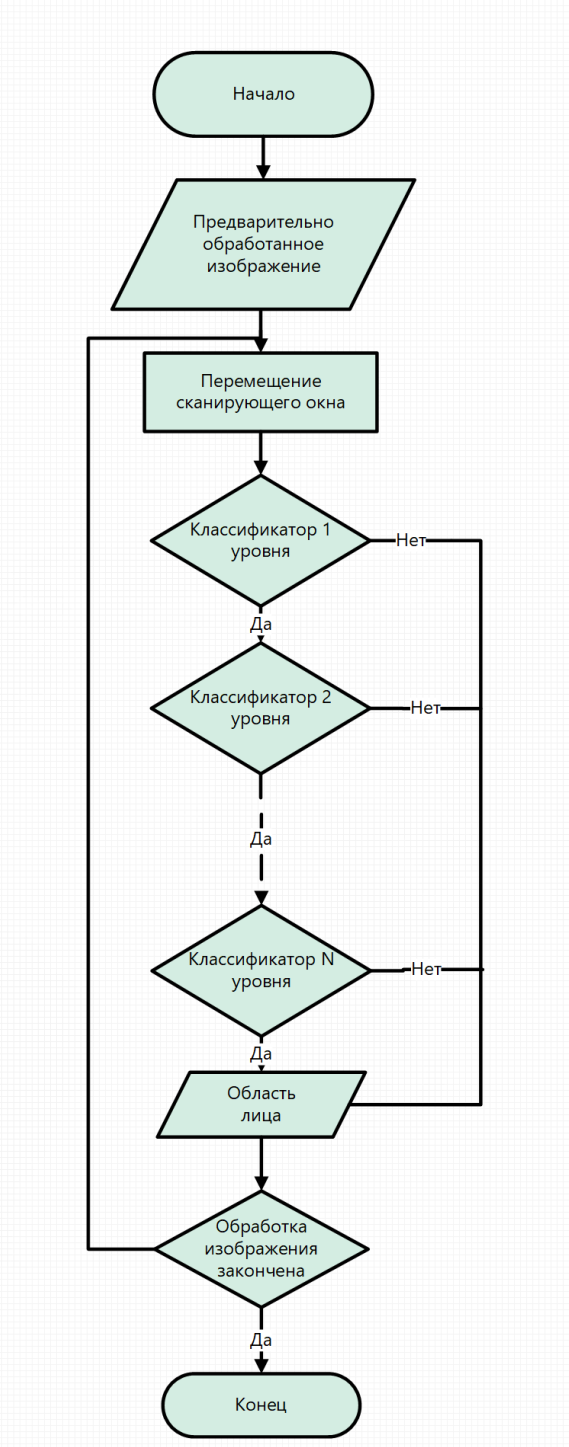

Рисунок 2.7 – Розпізнавання за методом Віоли-Джонса.

Скануючий вікно при кожному своєму проході по зображенню застосовує

каскадний класифікатор. На кожному наступному рівні перевіряються ознаки класифікатора. Якщо ознаки підходять, то перевіряється наступний класифікатор. Після успішного виконання останнього класифікатора програма вважає, що знайшла особа. Якщо будь-який з класифікаторів повертає від'ємне значення виконання, то програма з упевненістю вважає, що потрібний об'єкт не знайдено і скануючий вікно зміщується далі.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОСІБ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ OPENCV

3.1 Основний модуль

Для підготовки роботи розпізнавання потрібно мати натреновані класифікатори.

Для тренування класифікатора потрібні великі набори зображень - позитивних і негативних. Позитивні представляють собою набір шуканих об'єктів. Негативні є фоном. Важливо, щоб зображення були однакового розміру, тому що відбувається склеювання і вказуються ті координати, де знаходиться об'єкт. Простіше кажучи, ті координати, на яких навчається класифікатор. При склеюванні виходить новий набір зображень. Приклади можна взяти як з камери, так і з загальнодоступних баз даних.

Щоб почати навчання, потрібно мати папки «Good» - з позитивними зображеннями і «Bad» з негативними. Файли опису для позитивних і негативних зображень матимуть різну структуру. У негативних він простіше - це просто список відносних шляхів до зображень. Для позитивних запис складніше. Крім шляху також має бути вказано місце розташування об'єкта і розмір. В цілому, кожне позитивне зображення може містити кілька прикладів об'єктів. Найкраще на один кадр мати один об'єкт. "Good \ 17.bmp " – це адреса певного об'єкта щодо деректорії опису. З кількістю підходящих об'єктів на малюнку.

«314257 368» - спеціальні координати квадрата на внесенному малюнку, в якому знаходиться об'єкт. Та коли об'єктів знаходять декілька, то запис набуває вигляду: «Good

\ 17.bmp 8870 240 80 80 600 400 66 65 890 570 54 00 »(Рисунок 3.1).

Рисунок 3.1 – Приклад знімків позитивної вибірки

Тепер створимо підсумковий каскад. Для цих цілей будемо використовувати програми opencv_traincascade.exe і opencv_createsamples.exe. Навчання займає дуже багато часу. Наприклад, для навчання каскаду на 500-1000 зображень процес займе майже цілий день. Створення цих образів здійснюється за допомогою createsamplks.cpp, який в свою чергу завантажує для роботи cvhaartrainig.h. Ці файли є в стандартній бібліотеці OpenCV.

Далі представлені параметри запуску програми. void createTrainingSamples

const char * filename, // назва вихідного файлу, з правильними наборами зображень

const char * imgfilename, // зображення з шуканим об'єктом int bgcolor, // колір зображення

int bgthreshold, // поріг кольоровості, або прозорості 8-бітового зображення const char * bgfilename, // зображення з фоном

int count,

int invert = 10, // поворот зображення з об'єктом в градусах

int maxintensitydev = 30, // максимальне відхилення інтенсивності пікселів

зразків з вихідним елементом

double maxxangle = 1.2,

// максимальний кут повороту в радіанах за X double maxyangle = 1.2,

// максимальний кут повороту в радіанах за Y

double maxzangle = 0.6, // максимальний кут повороту в радіанах за Z int showsamples = 0,1 // якщо не нуль, то кожен зроблений зразок буде

показаний

int winwidth = 32, // така ширина в пікселях буде у кінцевого зразка int winheight = 32 // така висота в пікселях буде у кінцевого зразка

);

Параметри і стадії навчання класифікатора

void createCascadeClassifier (

const char * dirname, // ім'я директорії, в якій створюється і зберігається класифікатор

const char * vecfilename, // ім'я vec-файлу із зображеннями об'єкта якщо такий є

const char * bgfilename, // зображення з файлом, що описує фон int npos, // кількість позитивних зразків

int nneg, // кількість негативних зразків

int nstages, // кількість етапів або стадій навчання

int numprecalculated, // кількість ознак, які повинні бути прегенеріровани

int numsplits, // кількість слабких класифікаторів, що використовуються на кожній стадії навчання: 1 - відросток, 2 і більше - дерева

float minhitrate = 0.996F, // мінімальний бажаний коефіцієнт успіху на кожній стадії навчання класифікатора

float maxfalsealarm = 0.6F, // максимальний бажаний результат помилкових виявлень на кожній стадії

float weightfraction = 0.96F, // вказує параметр використовується ваги, для добре працюючого класифікатора найбільш оптимальним буде параметр 90

int mode = 3, // який використовується алгоритм, встановлює набір типів ознак Хаара, що використовуються під час навчання.

int equalweights = 1,1 // якщо параметр не дорівнює 0, це означає що початкові ваги всіх зразків будуть однаковими

int winwidth = 32, // розміри зразків int winheight = 32, // розміри зразків

intboosttype = 3, // вибір типу алгоритму бустінга, де 0 - DiscreteAdaBoost, 1 - RealAdaBoost, 2 - LogitBoost, 3 - GentleAdaBoost int stumperror = 0); // тип вказується помилки, якщо застосовується алгоритм Discrete AdaBoost.

Результатом даного етапу є готові для використання, натреновані класифікатори. Як тільки класифікатори будуть готові, їх можна буде використовувати в нашому додатку.

Класифікатор - це інструмент, який застосовується в data mining, який використовує класифіковані дані і на їх підставі намагається передбачити, до якого класу варто віднести нові дані.

Розпізнавання хоч і спрацьовує майже миттєво, все ж дає відчутні затримки в плавності якщо послідовно робити обчислення на кожному кадрі зображення, отриманого від камери, а потім відображати результат.

Цей додаток виробляє розпізнавання в паралельному потоці і віддає лише області детекції якщо такі є. Затримок між отриманням кадрів з камери не відбувається.

Паралельні обчислення - спосіб організації комп'ютерних обчислень, при якому програми розробляються як набір взаємодіючих обчислювальних процесів, що працюють паралельно (одночасно). Термін охоплює сукупність питань паралелізму в програмуванні, а також створення ефективно діючих апаратних реалізацій. Теорія паралельних обчислень становить розділ прикладної теорії алгоритмів.

Наочно схема роботи даного механізму показана на малюнку 3.2.

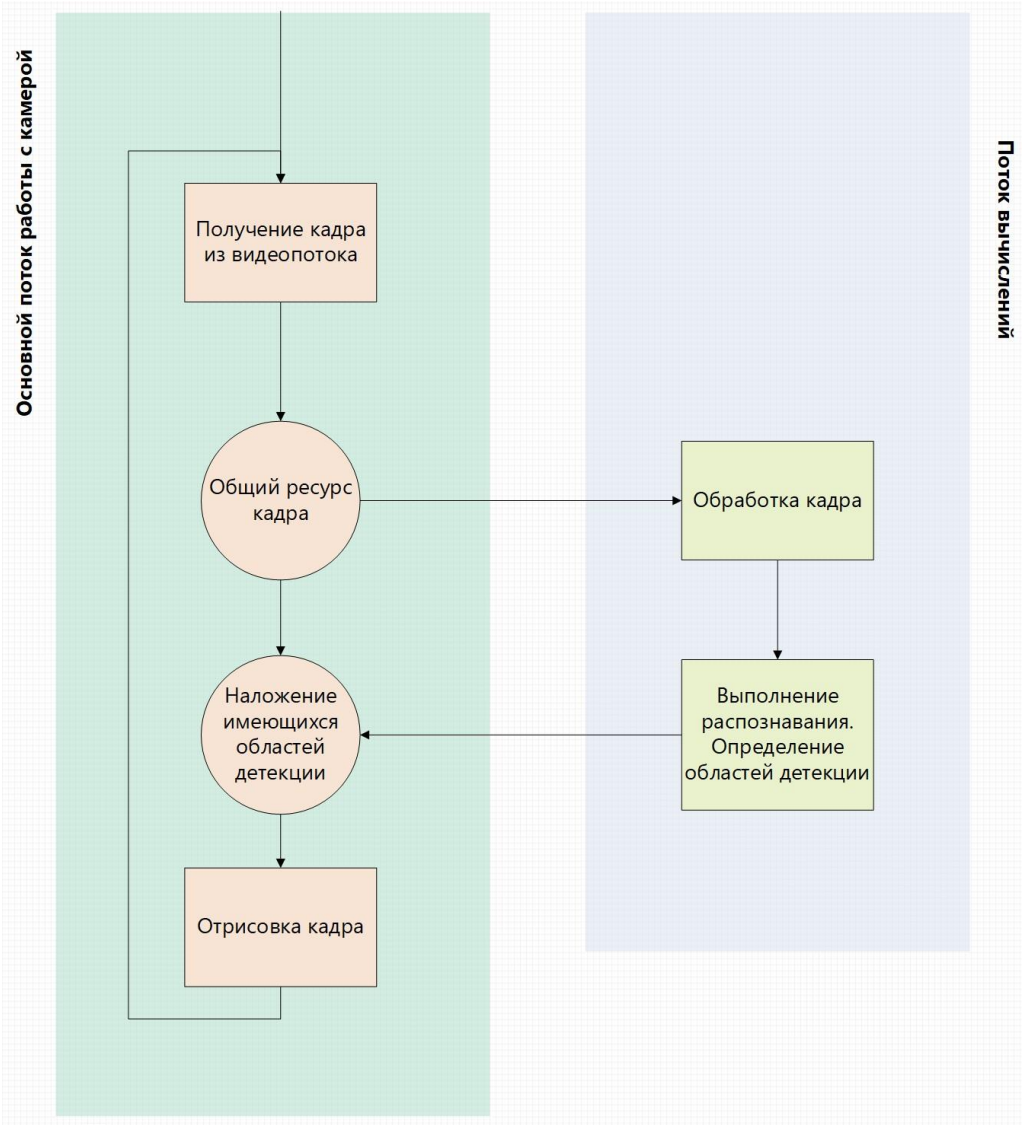

Рисунок 3.2 – Схема розпізнавання обличя з відеопотоку.

Спочатку проводиться ініціалізація распознавателя осіб і отримання ресурсу камери. А також визначення контейнера, куди будуть поміщатися лічені кадри.

recognizer = new Recognizer (canvas, classifier); capture = new opencv_videoio.VideoCapture (0); opencv_core.Mat frameMat = newopencv_core.Mat ();

Далі в циклі зчитуємо кадр, виводимо саме зображення, передає це зображення в распознаватель, отрісовиваємих зони детекції поверх кадру capture.read (frameMat);

canvas.showImage (converter.convert (frameMat));

recognizer.setMat (frameMat); drawDetections (canvas);

Тепер розглянемо роботи розпізнавача, який виконується в окремому потоці.

Визначається контейнер зображення. Для роботи даного методу розпізнавання використовується монохромне зображення, тому перетворимо отримане зображення в монохромне. Далі відбувається вирівнювання гістограми зображення, щоб алгоритм спрацьовував стабільно при різній освітленості.

opencv_core.Mat videoMatGray = new opencv_core.Mat (); if (getMat () == null) {

continue;

}

cvtColor (getMat (), videoMatGray, COLOR_BGRA2GRAY); equalizeHist (videoMatGray, videoMatGray);

Далі йде безпосереднє розпізнавання на основі класифікатора.

Використовується функція detectMultiScale, яка приймає параметри:

- вихідне зображення;

- контейнер, куди будуть записані прямокутники детекції;

- коефіцієнт збільшення масштабу;

- мінімальний сусідній поріг - це те, з якою інтенсивністю знаходиться кожна частина обличчя. Треба налаштовувати вручну, так як без порога детектор генерує багато одних і тих же распознаваний в одному місці. Зазвичай ізольовані виявлення є помилковими, так що має сенс відмовитися від них. Також має сенс об'єднати кілька виявлень для кожної лицьової області в єдину сутність.

Флаговая змінна може приймати кілька значень:

- 0;

- CV_HAAR_DO_CANNY_PRUNING;

- CV_HAAR_FIND_BIGGEST_OBJECT;

- CV_HAAR_DO_ROUGH_SEARCH;

- CV_HAAR_DO_CANNY_PRUNING;

- CV_HAAR_SCALE_IMAGE.

Якщо обраний прапор «практичною обрізки» (0 | CV_HAAR_DO_CANNY_PRUNING), то алгоритм не враховує області зображення, де знаходження особи не очікується. Це знижує час обчислення, і, можливо, усуває деякі хибні виявлення. Пропускалися області ідентифікуються детектором, який виявляє такі «непотрібні» краю по всьому зображенню, перед запуском лицьового детектора. Знову ж, вибір установки даного типу прапора є компромісом вибору між швидкістю і виявленням більшої кількості осіб. Установка прапора призведе до збільшення швидкості обробки, але може привести до пропуску деяких осіб. [4]

opencv_core.RectVector rectVector = new opencv_core.RectVector (); classifier.detectMultiScale (videoMatGray, rectVector, 1.1, 7, CV_HAAR_DO_CANNY_PRUNING | CV_HAAR_FIND_BIGGEST_OBJECT,

new opencv_core.Size (48, 48), new opencv_core.Size ()); setDetections (rectVector);

Методом setDetections зони детекції передаються для відтворення основних потоком роботи камери.

Це основні принципи роботи програми.

3.2 Тестування розробленої програми

Для перевірки працездатності і швидкодії алгоритму були проведені тестові випробування.

Для самого тесту потрібен масив зображень в локальній директорії (10 зображень). Всі зображення мають різний розмір, є кольорові, а є і чорно-білі фотографії, з різною кількістю людей в різних положеннях на них.

Для визначення статистики правильності розпізнавання був проведений підрахунок помилок розпізнавання за формулами:

P1 = Nрасп: Nобщ, (3.1)

P2 = Nнерасп: Nобщ, (3.2)

де P1 - помилки першого роду; P2 - помилки другого роду;

Nрасп - кількість розпізнаних об'єктів; Nнерасп - кількість нерозпізнаних об'єктів;

Nобщ - загальна кількість об'єктів.

Для проведення тестів був використаний вбудований файл OpenCV - performance.cpp, де ведеться підрахунок всіх даних і вибудовування ROC-кривої. Приклад підрахунку етапів або стадій класифікатора:

int * numclassifiers = newint [cascade-> count]; numclassifiers [0] = cascade-> stage_classifier [0] .count; for (i = 1; i <cascade-> count; i ++)

// послідовне застосування класифікаторів

{Numclassifiers [i] = numclassifiers [i-1] + cascade-

> Stage_classifier [i] .count;}

У таблиці 3.1 представлені результати тестування алгоритму із зазначенням точності роботи з натренованим набором для розпізнавання рис обличчя.

Данные\Алгоритм	Виола-Джонс	P_1/P_2
Массив изображений (10 штук)	91%, ~7 сек (7475 ms)	70% / 30%

Таблиця 3.1 - Результати тестування

Дані \ Алгоритм

Віола-Джонс

P1 / P2

Масив зображень (10 штук)

91%, ~ 7 сек (7475 ms)

70% / 30%

Використані для тесту зображення були показані на камеру в процесі

роботи програми. При знаходженні особи програма виділяє ділянку зеленим прямокутником. Результати показані на малюнках в додатку Г.

На предобработку фотографії витрачається 250-300 мс. Саме визначення особи відбувається швидше, так як функція визначає типи спрацювали класифікаторів і блискавично видає результат. Але за рахунок винесення обчислень в окремий потік, цей час не враховується.

За результатами тестування основна частина осіб була розпізнана навіть незважаючи на різницю освітлення, перекриття та положення. Ті особи, що не

були розпізнані, не володіють достатніми ознаками, щоб класифікатор визнав їх придатними. Як правило, це відбувається, коли на відео присутня лише частина особи, зображення розмите, занадто далеко. В інших випадках відсутність розпізнавання є не недоліком алгоритму, а недостатньою натренованістю класифікатора.

ВИСНОВОК

В ході виконання роботи були досягнуті всі поставлені цілі. Відповідно до поставлених цілей були досліджені основні існуючі методи розпізнавання осіб.

На сьогоднішній день розпізнавання облич у відео зображенні стає дуже необхідним. Ведеться дуже багато досліджень в цій області. На одній з конференцій була презентація однієї цікавої системи, розробленої німецькими вченими, в якій програмне забезпечення распознавало фігури людей, і в залежності від того, куди рухався чоловік програма автоматично повертала камеру і стежила за ним. Дану систему можливо використовувати для автоматичного запису лекцій, які читає викладач біля дошки.

Шляхом виявлення достоїнств і недоліків було визначено найкращий метод для розпізнавання в потоці. Метод Віоли-Джонса на даний момент є максимально відповідним за всіма параметрами для задач розпізнавання об'єктів в потоці. Був розроблений і реалізований алгоритм у вигляді програми і використання бібліотеки комп'ютерного зору OpenCV, яка має функції для: оброблення картинки, калібрування зображення за прикладом, зміна візуальних змін, знаходження подібності, аналіз зміни знаходження об'єктів, визначення форми об'єкту та відстежування об'єкта, 3D-реконструкція , поділення об'єкта, визначення жестів і т.д. Ця бібліотека дуже популярна за рахунок своєї відкритості та можливості безкоштовно використовувати як в навчальних, так і комерційних цілях.

Розроблене додаток дозволяє здійснювати відеоспостереження, виділяючи особи людей в відеопотоці, а також здійснювати автоматичне стеження з фотофіксацією осіб людей, які були зафіксовані під час спостереження. Цей механізм дозволяє робити фотозвіт з датою і часом, коли спрацювала детекция. При цьому відпадає необхідність зберігати всю відеозапис спостереження. Потрапили в кадр особи будуть збережені і доступні до перегляду, як в додатку, так і в папці провідника. Крім того, є можливість корекції зображення з метою поліпшення якості картинки в умовах поганої освітленості.

Перелік використаної літератури

1. Миколаїв, М. І., Вимірювальний контроль із застосуванням неіндустріальний камер в виробничих умовах [Текст] / М. І. Миколаїв - Наука, 1989. - 312 с.

2. Горшенин, В. А., Геодезичний інваріант в біометрики особи [Текст] / В. А. Горшенин -. 2014. - 4 с. - (наук. Стаття)

3. Алфімцев, А.Н., Розробка і дослідження методів захоплення, відстеження та розпізнавання динамічних жестів [Текст] / А. Н. Алфімцева МГТУ ім. Н.е. Баумана. - Москва: 2008. - 226 с.

4. Ірматов, А.А., Спосіб і система для розпізнавання особи з урахуванням списку людей, що не підлягають перевірці [Текст] / А. А. Ірматов, Д. Ю. Буряк - Корпорація «Самсунг електронікс Ко., Лтд.». - Москва: 2010. - 22 с.

5. Колесник, А. В., Розподілена програмна система розпізнавання осіб [Текст] / А. В. Колесник, Ю. В. Ладиженський - VI міжнародна науково - технічна конференція студентів, аспірантів та молодих науковців «Інформатика і комп'ютерні технології» - Донецьк : ДонНТУ, 2010. - 251 с.

6. Умяров, Н. Х., Виділення особи на знімку з відеопотоку з метою його розпізнавання [Текст] / Н. Х. Умояров, О. І. Федяєв - VII міжнародна науково - технічна конференція студентів, аспірантів та молодих науковців «Інформатика та комп'ютерні технології »- Донецьк: ДонНТУ, 2011. - 177 с.

7. Костецька, Г. Ю., Кодування зображень людських облич за допомогою самоорганізується карти Кохонена [Текст] / Г. Ю. Костецька, О. І. Федяєв - V міжнародна науково - технічна конференція студентів, аспірантів та молодих науковців «Інформатика та комп'ютерні технології »- Донецьк: ДонНТУ, 2009. - 268 с.