

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК 004.932.72

«До захисту допущено»

Завідувач кафедри

О.В. Коваль
(підпис) (ініціали, прізвище)

“ ” _____ 2018р.

Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення
за спеціалізацією Програмне забезпечення розподілених систем
на тему: Виявлення об'єкту на зображенні з відеокамер

Виконав: студент 6 курсу, групи ТВ-71мп

Колот Сергій Сергійович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник доц., доц., к.т.н. Шаповалова С. І.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

**Національний технічний університет України
“ Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією - Програмне забезпечення розподілених систем

ЗАТВЕРДЖУЮ
Завідувач кафедри

Коваль О.В.

(прізвище, ініціали)

_____ (підпис)

« ____ » _____ 2018р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Колот Сергій Сергійович

(прізвище, ім'я, по батькові)

1. Тема дисертації Виявлення об'єкту на зображенні з відеокамер

Науковий керівник Шаповалова Світлана Ігорівна доц., доц., к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “05” листопада 2018 року № 4072-с

2. Строк подання студентом дисертації 11.12.2018

3. Об'єкт дослідження є алгоритмічна база та програмне забезпечення аналізу зображень

4. Предмет дослідження є методи машинного навчання для пошуку та виокремлення об'єктів на зображенні

5. Перелік питань, які потрібно розробити _____

1. дослідити існуючі підходи та методи виявлення об'єктів на зображеннях

2. провести порівняння існуючих методів пошуку об'єктів на зображеннях з використанням каскадних класифікаторів

3. створити програмний прототип для пошуку об'єктів на зображеннях

4. провести обчислювальні експерименти для визначення більш ефективних наборів параметрів роботи класифікаторів

6. Орієнтований перелік ілюстративного матеріалу підходи до виокремлення

заданого об'єкту на зображенні, програмні засоби реалізації каскадних

класифікаторів, класифікатори OpenCV, архітектура програмної системи,

графічний інтерфейс користувача, порівняльний аналіз результатів

досліджень

7. Орієнтований перелік публікацій доповідь «Розпізнавання об'єктів з навмисним маскуванням»; стаття «Системи моделювання Convolutional Neural Networks»

8. Дата видачі завдання « 11 » вересня 2017 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз проблеми виявлення об'єктів на зображеннях	11.09.2017-12.11.2017	
2	Аналіз існуючих методів пошуку об'єктів на зображеннях з використанням каскадних класифікаторів	13.11.2017-15.01.2018	
3	Аналіз існуючих методів пошуку антропометричних точок обличчя	16.01.2018-05.03.2018	
4	Вибір програмних методів та засобів реалізації	06.03.2018-27.06.2018	
5	Опанування обраних засобів реалізації	28.06.2018-18.08.2018	
6	Розробка схеми роботи програмного застосунку пошуку об'єктів на зображеннях	19.08.2018-30.08.2018	
7	Розробка архітектури програмного забезпечення	01.09.2018-20.09.2018	
8	Розробка дизайну програмного прототипу для пошуку об'єктів на зображеннях	21.09.2018-10.10.2018	
9	Розробка програмного застосунку та проведення обчислювальних експериментів для визначення більш ефективних наборів параметрів роботи класифікаторів	11.10.2018-30.11.2018	
10	Оформлення документації	01.12.2018-07.12.2018	

Студент

_____ (підпис)

Колот С.С.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Шаповалова С.І.

_____ (прізвище та ініціали)

РЕФЕРАТ

У зв'язку з стійкою тенденцією до інтелектуалізації програмного забезпечення виникає потреба у розв'язанні задач прийняття рішень на основі зображень. Однією з таких задач проблеми обробки та розпізнавання зображень є виокремлення об'єктів на кадрі фото та відеозйомки. Розв'язання цієї задачі потребується в таких галузях:

- системи ідентифікації осіб (обличчя, роговиця ока);
- системи контролю руху (виокремлення автомобільних знаків);
- інтелектуальне налаштування фотозйомки.

Саме тому дослідження та реалізація методів визначення заданого об'єкту є **актуальною проблемою** і має практичне застосування.

Зв'язок роботи з науковими програмами, планами, темами. Основні дослідження за темою магістерської роботи здійснено згідно з планами наукових досліджень кафедри Автоматизації проектування енергетичних процесів та систем національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського», в тому числі в рамках науково-дослідницької роботи «Інтелектуальна обробка графічної інформації» (реєстраційний номер 0117U006081).

Метою дослідження є програмна реалізація методів пошуку об'єктів на зображенні з відеокамери із застосуванням машинного навчання.

Реалізація цієї мети передбачає виконання таких **завдань**:

- дослідити існуючі підходи та методи виявлення об'єктів на зображеннях;
- провести порівняння існуючих методів пошуку об'єктів на зображеннях з використанням каскадних класифікаторів;
- створити програмний прототип для пошуку об'єктів на зображеннях;

- провести обчислювальні експерименти для визначення більш ефективних наборів параметрів роботи класифікаторів.

Об'єктом дослідження є алгоритмічна база та програмне забезпечення аналізу зображень.

Предметом дослідження є методи машинного навчання для пошуку та виокремлення об'єктів на зображенні.

Методи дослідження. Розв'язання поставлених задач виконувались засобами машинного навчання, зокрема з використанням: каскадного класифікатору на основі ознак Haar (Haar features); каскадного класифікатору на основі ознак LPF(LBP-Features); методу пошуку антропометричних точок обличчя на основі локальних бінарних ознак (Local Binary Features)

Практичне значення одержаних результатів. Теоретичні дослідження стали основою побудови прототипу системи пошуку об'єкту на зображеннях з відеокамери на основі каскадних класифікаторів.

Апробація результатів дисертації

Положення дисертації оприлюднено в доповідях на 2 міжнародних та 1 всеукраїнській наукових конференціях:

1. XV-XVI Міжнародних науково-практичних конференції аспірантів, магістрантів і студентів «Сучасні проблеми наукового забезпечення енергетики», Київ, 2017-2018 р.

2. IV-V науково-практичних дистанційних конференціях молодих вчених і фахівців з розробки програмного забезпечення, 2017-2018 р.

Публікації. Наукові положення дипломної роботи опубліковані у 4 роботах:

1. Колот С.С. Автономна синхронна генерація ключів шифрування на основі нейронних мереж/ С.С. Колот, С.І. Шаповалова // Сучасні проблеми наукового забезпечення енергетики: матеріали XV Міжнародної науково-

практичної конференції аспірантів, магістрантів і студентів, Київ, 25-28 квітня 2017 р. у 2 т. - К: КПІ ім. Ігоря Сікорського, 2017.- Т.2.- С.204.

2. Колот С.С. Автономна синхронна генерація ключів шифрування на основі нейронних мереж/ С.С. Колот, С.І. Шаповалова // Сучасні аспекти розробки програмного забезпечення: Збірник наукових праць IV науково-практичної дистанційної конференції молодих вчених і фахівців з розробки програмного забезпечення, 30 квітня 2017. – Черкаси: видавець Чабаненко Ю.А., 2017.-С. 61.

3. Колот С.С. Розпізнавання об'єктів з навмисним маскуванням / С.С. Колот, С.І. Шаповалова // Сучасні проблеми наукового забезпечення енергетики: матеріали XVI Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів, Київ, 24-27 квітня 2018 р. у 2 т. - К: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2018.- Т.2.- С.191.

4. Шаповалова С.І. Системи моделювання Convolutional Neural Networks / С.І. Шаповалова, С.С. Колот // Сучасні аспекти розробки програмного забезпечення: Збірник наукових праць V науково-практичної дистанційної конференції молодих вчених і фахівців з розробки програмного забезпечення, 15 травня 2018. – Черкаси: видавець Чабаненко Ю.А., 2018. – С. 209-213.

Ключові слова: пошук об'єктів на зображенні, машинне навчання, класифікатори, комп'ютерне бачення, антропометричні точки обличчя.

ABSTRACT

In connection with the steady tendency to intellectualize the software there is a need for solving decision-making problems based on images. One of the challenges in image processing and recognition is the selection of objects in the photo and video frame. The solution of this task is required in the following areas:

- systems of identification of persons (face, cornea of the eye);
- traffic control systems (marking off road signs);
- Intelligent photo setup.

That is why research and implementation of methods for determining a given object is an **actual problem** and has practical application.

Relationship of work with scientific programs, plans, themes. The main research on the theme of the master's work was carried out in accordance with the plans of scientific research of the Department of Automation of the design of energy processes and systems of the National Technical University of Ukraine "Kyiv Polytechnic Institute. Igor Sikorsky ", including in the framework of the research work "Intelligent processing of graphic information "(registration number 0117U006081).

The purpose of the study is the software implementation of methods for finding objects on the image from a video camera with the use of machine learning.

The realization of this goal involves the following **tasks**:

- explore existing approaches and methods for detecting objects in images;
- to compare existing methods of finding objects on images using cascade classifiers;
- create a software prototype for finding objects in images;
- conduct computational experiments to determine more efficient sets of job parameters of the classifiers.

The object of the study is the algorithmic database and image analysis software.

The subject of the research is the methods of machine learning for the search and selection of objects in the image.

Research methods. The solution of the set tasks was carried out by means of machine learning, in particular using a cascade classifier based on Haar features (Haar features); cascaded classifier based on LPF (LBP-Features); Method of searching anthropometric points of a face based on local binary features (Local Binary Features)

The practical value of the results. Theoretical studies have become the basis for constructing a prototype of the object search system on images from a video camera based on cascade classifiers.

Approbation of the results of the dissertation

The thesis has been published in the reports of 2 international and 1 all-Ukrainian scientific conferences:

1. XV-XVI International Scientific and Practical Conference of Postgraduates, Masters and Students "Modern Problems of Scientific Supply of Power Engineering", Kyiv, 2017-2018.

2. IV-V Scientific and Practical Distance Conferences of Young Scientists and Software Development Specialists, 2017-2018.

Publications. Scientific terms of thesis are published in 4 works:

1. Kolot SS Autonomous synchronous generation of encryption keys based on neuronal networks / SS Kolot, SI Shapovalova // Modern problems of scientific support of power engineering: materials of the XV International scientific and practical conference of postgraduates, masters and students, Kyiv, April 25-28, 2017 in 2 t. - K: KPI them. Igor Sikorsky, 2017.- Vol.2.- P. 204.

2. Kolot SS Autonomous synchronous generation of encryption keys based on neural networks / SS Kolot, SI Shapovalova // Modern aspects of software development: Collection of scientific works of the IV scientific and practical distance conference of young scientists and software development specialists, April 30, 2017. - Cherkasy: publisher Chabanenko Yu.A., 2017.- C. 61

3. Kolot SS Recognition of objects with intentional masking / SS Kolot, SI Shapovalova // Modern problems of scientific support of power engineering: materials of the XVIII International Scientific and Practical Conference of Postgraduate Students, Graduates and Students, Kyiv, April 24-27, 2 t. - K: KPI them. Igor Sikorsky, View of "Polytechnic", 2018.- Vol.2.- P.191.

4. Shapovalova SI Convolutional Neural Networks Simulation Systems / SI Shapovalova, SS Kolot // Modern aspects of software development: Collection of scientific works of the V scientific and practical distance conference of young scientists and software development specialists, May 15, 2018. - Cherkasy: publisher Chabanenko Yu.A., 2018. - P. 209-213.

Keywords: object detection, machine learning, classifiers, computer vision, anthropometric points of the person.

ЗМІСТ

Реферат	4
Abstract.....	7
Перелік умовних позначень, символів, скорочень і термінів	12
Вступ.....	13
1. Задача пошуку об'єктів на зображеннях	15
1.1. Мета та призначення програмного забезпечення.	15
1.2. Задачі, які розв'язуються.....	15
1.3. Вимоги до програмного забезпечення.....	16
1.4. Потенційні користувачі	16
2. Методи виділення об'єктів на зображеннях.....	17
2.1. Підходи до пошуку об'єкту на зображенні.	17
2.1.1. Машинне навчання	18
2.1.2. Глибинне навчання.....	19
2.2. Каскадні класифікатори.....	22
2.2.1. Haar Cascade Classifier	23
2.2.2. LBP Cascade Classifier.....	25
2.3. Знаходження антропометричних точок обличчя	27
2.4. OpenCV	29
3. Опис програмної реалізації	31
3.1. Засоби розробки	31
3.1.1. Платформи та мови програмування.....	31
3.1.2. Засоби розробки системи пошуку об'єктів на зображеннях	34
3.2. Реалізація класів системи пошуку об'єкта	38
3.3. Методика роботи користувача з програмною системою	41
3.3.1. Інсталяція та системні вимоги	41
3.3.2. Сценарії роботи користувача з системою	44

4.	Обчислювальні експерименти.....	47
4.1.	Виявлення обличчя	47
4.2.	Виявлення номерних знаків	49
4.3.	Виокремлення очей людини	50
4.4.	Характеристики роботи класифікаторів	51
4.5.	Порівняння ефективності локалізації антропометричних точок	52
5	Розробка стартап проекту «Object-detection system»	54
5.1	Опис ідеї проекту	54
5.2	Технологічний аудит ідеї проекту.....	56
5.3	Аналіз ринкових можливостей запуску стартап проекту	56
5.4	Розроблення ринкової стратегії проекту.....	64
5.5	Розроблення маркетингової програми стартап-проекту	67
	Висновки по роботі.....	71
	Використані джерела	72
	Додаток А	75
	Додаток Б.....	82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – Convolutional neural network сторткова нейронна мережа

LBP – local binary patterns локальні бінарні шаблони

LBF – local binary features локальні бінарні характеристики

OpenCV – Open Source Computer Vision Library бібліотека комп'ютерного зору з відкритим вихідний кодом

R-CNN, Fast R-CNN, Faster R-CNN – Region proposal convolutional neural network

RPN – Region proposal network сторткова нейронна мережа з регіонами

SSD – Single Shot Multi Box Detector

YOLO – You Only Look Once назва алгоритму пошуку та класифікації об'єктів

VGG16 – Visual Geometry Group архітектура нейронної мережі

SVM – support vector machine метод опорних векторів

ВСТУП

У зв'язку з стійкою тенденцією до інтелектуалізації програмного забезпечення виникає потреба у розв'язанні задач прийняття рішень на основі зображень. Одною з таких задач проблеми обробки та розпізнавання зображень є виокремлення об'єктів на кадрі фото та відеозйомки. Розв'язання цієї задачі потребується в таких галузях:

- системи ідентифікації осіб (обличчя, роговиця ока);
- системи контролю руху (виокремлення автомобільних знаків);
- інтелектуальне налаштування фотозйомки.

Саме тому дослідження та реалізація методів визначення заданого об'єкту є **актуальною проблемою** і має практичне застосування.

Зв'язок роботи з науковими програмами, планами, темами. Основні дослідження за темою магістерської роботи здійснено згідно з планами наукових досліджень кафедри Автоматизації проектування енергетичних процесів та систем національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського», в тому числі в рамках науково-дослідницької роботи «Інтелектуальна обробка графічної інформації» (реєстраційний номер 0117U006081).

Метою дослідження є програмна реалізація методів пошуку об'єктів на зображенні з відеокамери із застосуванням машинного навчання.

Реалізація цієї мети передбачає виконання таких **завдань**:

- дослідити існуючі підходи та методи виявлення об'єктів на зображеннях;
- провести порівняння існуючих методів пошуку об'єктів на зображеннях з використанням каскадних класифікаторів;
- створити програмний прототип для пошуку об'єктів на зображеннях;
- провести обчислювальні експерименти для визначення більш ефективних наборів параметрів роботи класифікаторів.

Об'єктом дослідження є алгоритмічна база та програмне забезпечення аналізу зображень.

Предметом дослідження є методи машинного навчання для пошуку та виокремлення об'єктів на зображенні.

Методи дослідження. Розв'язання поставлених задач виконувались засобами машинного навчання, зокрема з використанням: каскадного класифікатору на основі ознак Haar (Haar features); каскадного класифікатору на основі ознак LBP (LBP-Features); методу пошуку антропометричних точок обличчя на основі локальних бінарних ознак LBF (Local Binary Features)

Практичне значення одержаних результатів. Теоретичні дослідження стали основою побудови прототипу системи пошуку об'єкту на зображеннях з відеокамери на основі каскадних класифікаторів.

Структура та обсяг дипломної роботи. Магістерська дисертація складається зі вступу, п'яти розділів, висновку, переліку посилань з 33 найменувань, 2 додатки, і містить 25 рисунків, 28 таблиць. Повний обсяг магістерської дисертації складає 83 сторінок, з яких перелік посилань займає 3 сторінки, додатки – 9 сторінки.

1. ЗАДАЧА ПОШУКУ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

У розділі наведено мету та призначення програмного забезпечення яке розроблено на основі досліджень магістерської дисертації. Наведено задачі які розв'язуються в рамках роботи та описані вимоги до програмного забезпечення.

1.1. Мета та призначення програмного забезпечення.

Метою розробки є створення програмного продукту, призначеного для пошуку об'єкту на зображеннях та з відео камери.

Призначення програмного забезпечення полягає в програмній реалізації методів пошуку об'єкту на зображеннях для подальшого їх використання.

Розпізнавання зображень знаходить широке застосування в різних додатках - це може бути контроль топології друкованих плат, текстури тканини, контроль доступу до інформації щодо ідентифікації особи (біометрична ідентифікація), доступ до об'єктів обмеженого доступу, оперативний пошук в картотеці зображень, дактилоскопія і ін. Широке поширення отримують біометричні системи ідентифікації людини. Біометричні системи ґрунтуються на унікальних біологічних характеристиках людини, які важко підробити і які однозначно визначають конкретну людину. До таких характеристик відносяться відбитки пальців, форма долоні, візерунок райдужної оболонки, зображення сітківки ока, індивідуальні характеристики особи.

1.2. Задачі, які розв'язуються

У рамках роботи програмне забезпечення що розробляється розв'язує такі задачі:

- пошук об'єкту – пошук заданого об'єкту на зображенні із застосуванням машинного навчання, а саме із застосування каскадних класифікаторів.

- пошук точок обличчя – пошук та відображення антропометричних точок обличчя.

1.3. Вимоги до програмного забезпечення

За завданням для виокремлення об'єкту має використовуватись каскадні класифікатори їх програмну реалізацію представлено в бібліотеці з відкритим вихідним кодом OpenCV.

Система пошуку об'єкту повинна надавати такі можливості:

- завдання об'єкту пошуку (наприклад: обличчя, номерний знак);
- задання параметрів алгоритмам пошуку об'єкту на зображенні;
- для класифікаторів застосування заздалегідь навчених каскадних класифікаторів;
- підключення зовнішнього джерела зображень (наприклад вибір папки з зображеннями);
- пошук та відображення антропометричних точок обличчя людини;

1.4. Потенційні користувачі

Потенційними користувачами та галузями застосування можуть бути:

- промислові підприємства;
- охоронні підприємства;
- державні органи контролю дорожнього руху;
- системи розпізнавання особи.

Також застосування даного програмного забезпечення можливе як складова частина для інтеграції в інше програмне забезпечення якому необхідно знаходити об'єкт чи антропометричні точки обличчя для подальшої обробки цих даних.

2. МЕТОДИ ВИДІЛЕННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

В другому розділі визначені базові підходи до вирішення задачі пошуку об'єкту на зображенні. Та надано більш детальну інформацію стосовно каскадних класифікаторів для виокремлення заданих об'єктів з використанням ознак Хаар. Наведено загальну інформацію про пошук антропометричних точок обличчя та їх застосуванні. Наприкінці розділу надано опис OpenCV як програмного засобу реалізації каскадних класифікаторів та детектору антропометричних точок.

2.1. Підходи до пошуку об'єкту на зображенні.

Комп'ютерне бачення – це галузь інформатики, яка вивчає сприйняття комп'ютерами графічної інформації, а саме зображення. Це спосіб, за допомогою якого комп'ютери збирають та інтерпретують візуальну інформацію з навколишнього середовища.

Зазвичай зображення спочатку обробляється на нижчому рівні, щоб покращити якість зображення, наприклад, видалити шум. Тоді картина обробляється на більш високому рівні, наприклад, виявлення шаблонів і форм, і тим самим намагаючись визначити, що на малюнку [1].

На рисунку 2.1 можна бачити існуючі підходи до реалізації задачі пошуку об'єкту.

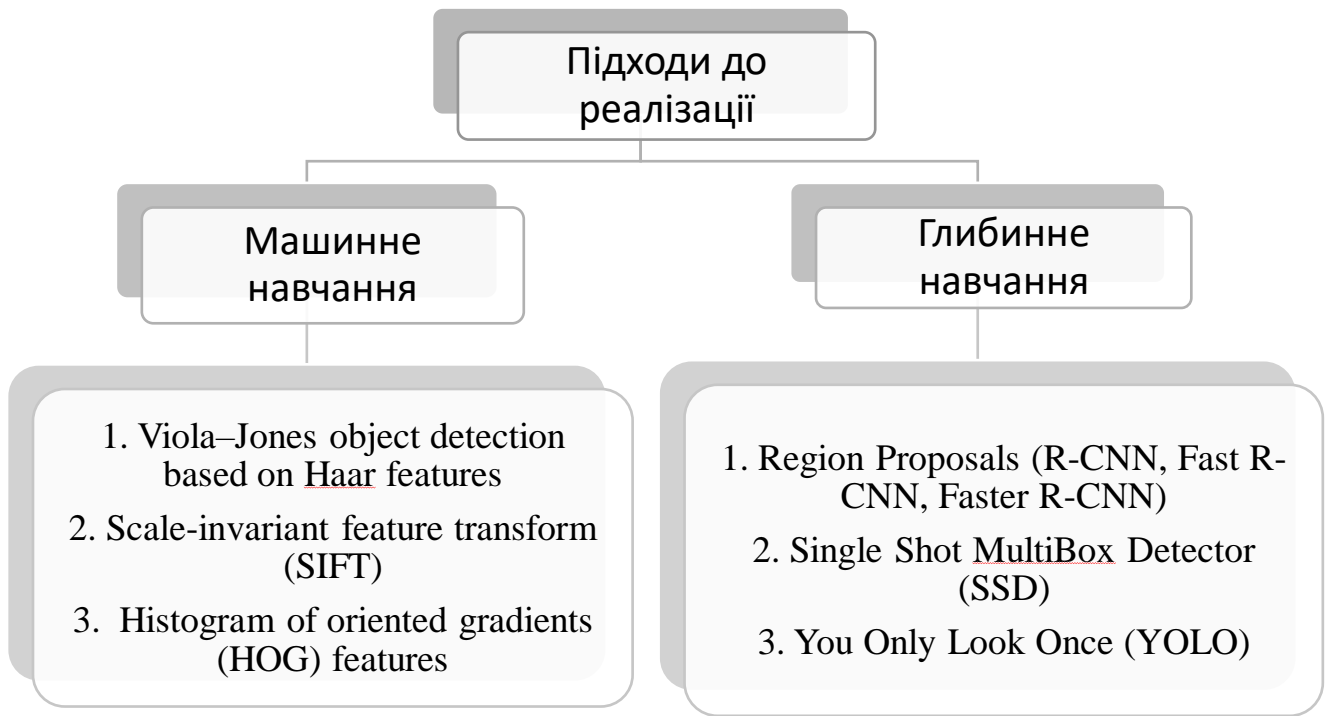


Рис. 2.1 – Підходи реалізації задачі пошуку об'єкту на зображенні

2.1.1. Машинне навчання

Машинне навчання - це практика використання алгоритмів для аналізу даних, вивчення їх, а потім визначення або передбачення чогось в світі. Як підхід воно базується на «навчанні» з використанням великої кількості даних і алгоритмів, які дають можливість дізнатися, як виконати поставлене завдання.

До цього підходу відносяться наступні методи:

- Виділення Haar-features це модифікація ідеї використання вейвлетів Haar [2] яку запропонували Viola та Jones в роботі «Rapid object detection using a boosted cascade of simple features» [3]. Вона, розглядає сусідні прямокутні регіони в певному місці в вікні виявлення, підсумовує інтенсивність пікселів у кожному регіоні та обчислює різницю між цими сумами. Ця різниця потім використовується для розділення підрозділів зображення.
- SIFT – базується на тому що для будь-якого об'єкта на зображенні можна отримати цікаві точки на об'єкті, щоб забезпечити "опис об'єкта". Цей опис, витягнутий з зображення, може потім бути використаний для ідентифікації

об'єкта при спробі знайти об'єкт на зображенні, що містить багато інших об'єктів. Для забезпечення надійного розпізнавання важливо, щоб функції, витягнуті з тренувального зображення, були виявлені навіть під час зміни масштабу зображення, шуму та підсвічування. Такі точки, як правило, лежать на висококонтрастних ділянках зображення, наприклад, у краях об'єкта. Цей метод запропонував David Lowe's в своєму патенті "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image"[4].

- Виділення характеристик HOG – Суттєвою думкою, що стоїть за гістограмою дескриптора орієнтованих градієнтів, є те, що зовнішній вигляд і форма локального об'єкта в межах зображення можуть бути описані розподілом градієнтів інтенсивності або напрямків краю. Зображення поділяється на невеликі об'єднані області, які називаються клітинами, а для пікселів у кожній комірці складається гістограма градієнтних напрямків. Дескриптор - об'єднання цих гістограм. Для покращення точності місцеві гістограми можуть бути контрастно-нормалізовані, обчисливши міру інтенсивності в більшій області зображення, називається блоком, а потім використовуючи це значення для нормалізації всіх комірок у блоці. Ця нормалізація призводить до кращої інваріантності до змін у висвітленні та затемненні. Цей метод був запропонований Navneet Dalal та Bill Triggs в їх роботі «Histograms of Oriented Gradients for Human Detection» [5]

2.1.2. Глибинне навчання.

Глибинне навчання це метод впровадження машинного навчання. У процесі глибокого навчання кожен рівень навчається перетворювати вхідні дані в дещо більш абстрактне та складне уявлення. У прикладі розпізнавання зображення вихідний ввід може бути матрицею пікселів; перший репрезентативний шар може абстрагувати пікселі та кодувати краї; другий шар може складати і кодувати розташування країв; третій шар може кодувати ніс і

очі; і четвертий шар може визнати, що зображення містить обличчя. Важливо відзначити, що глибокий процес навчання дозволяє дізнатися, які функції оптимально розміщати на якому рівні самостійно. (Звичайно, це не повністю виключає необхідність ручної настройки: наприклад, різне число шарів і розмірів шару можуть забезпечити різні ступені абстракції.) [6][7]

До цього підходу відносяться наступні методи:

- R-CNN – Архітектура мережі була розроблена командою з UC Berkley для застосування Convolution Neural Networks до задачі пошуку об'єкту. Інститути, які на той момент підходи до вирішення таких завдань наблизилися до максимуму своїх можливостей і значимо поліпшити їх показники не виходило. CNN добре показували себе в класифікації зображень. Для цього на вхід CNN подавалося не все зображення цілком, а попередньо виділені іншим способом регіони, на яких імовірно є якісь об'єкти. Цю архітектуру запропонували Ross Girshick, Jeff Donahue, Trevor Darrell та Jitendra Malik у своїй роботі «Rich feature hierarchies for accurate object detection and semantic segmentation» На той момент підходів до виділення регіонів було кілька, автори обрали Selective Search [8], хоча вони вказують, що особливих причин для переваги саме його немає.[9]

- Fast R-CNN – Незважаючи на високі результати, продуктивність R-CNN була все ж невисока, особливо для більш глибоких мережах (таких як VGG16). Крім того, навчання bounding box regressor і SVM вимагало збереження на диск великої кількості ознак, тому воно було дорогим з точки зору розміру сховища. Автори Fast R-CNN запропонували прискорити процес за рахунок пари модифікацій: пропускати через CNN не кожен з 2000 року регіонів-кандидатів окремо, а все зображення цілком. Запропоновані регіони потім накладаються на отриману загальну карту ознак; замість незалежного навчання трьох моделей (CNN, SVM, bbox regressor) поєднати всі процедури тренування в одну. Цю архітектуру запропонував Ross Girshick у своїй роботі «Fast R-CNN» [10]

- Faster R-CNN – Після поліпшень, зроблених в Fast R-CNN, найвужчим місцем нейронні мережі виявився механізм генерації регіонів-кандидатів. У 2015 команда з Microsoft Research змогла зробити цей етап значно швидшим. Вони запропонували обчислювати регіони не по споконвічного зображенню, а знову ж таки по карті ознак, отриманих з CNN. Для цього був доданий модуль під назвою Region Proposal Network (RPN). В рамках RPN по витягнутих CNN ознаками ковзають «міні-нейромережею» з невеликим (3x3) вікном. [11]

- SSD – метод виявлення об'єктів у зображеннях за допомогою однієї глибини нейронної мережі. Під час прогнозування мережа генерує бали для наявності кожної категорії об'єктів у кожному вікні за замовчуванням та створює корективи для зон, щоб краще відповідати формі об'єкта. Крім того, мережа поєднує в собі прогнози з карт різних об'єктів з різною роздільною здатністю для обробки об'єктів різного розміру. SSD є більш простим по відношенню до методів, що потребують області на зображеннях, оскільки це повністю виключає генерацію областей та інкапсулює всі обчислення в єдину мережу. Цей метод запропонував Wei Liu та інші у роботі «SSD: Single Shot MultiBox Detector». [12]

- YOLO – підхід до виявлення об'єктів вирішує проблему виявлення об'єктів як проблему регресії для просторово відокремлених областей та пов'язаних з ним імовірностей класу. Це одиночна нейронна мережа яка передбачає обмежувальні області та класові ймовірності які отримуються безпосередньо з повних зображень за один цикл виконання. Оскільки весь процес виявлення виконується єдиною мережею то швидкість роботи цього методу порівняно велика. Цей підхід запропонували Joseph Redmon , Santosh Divvala, Ross Girshick та Ali Farhadi у їх роботі «You Only Look Once: Unified, Real-Time Object Detection» [13]

2.2. Каскадні класифікатори

Каскадний класифікатор складається з переліку етапів, де кожен етап складається зі списку характеристик. Система виявляє ці об'єкти, переміщаючи вікно над зображенням. Кожна стадія класифікатора називає конкретний регіон, визначений поточним розташуванням вікна, як позитивний, так і негативний - позитивний результат значить що об'єкт був знайдений, або негативний означає, що вказаний об'єкт не був знайдений на зображенні. Якщо пошук дає негативний результат, то класифікація цього конкретного регіону завершується і розташування вікна переноситься в наступне місце. Якщо маркування дає позитивний результат, то область переходить на наступний етап класифікації. Класифікатор дає остаточний вердикт позитивного, коли всі етапи, включаючи останній, дають результат, говорячи про те, що об'єкт знайдено на зображенні.

Істинне позитивне значення означає, що об'єкт, про який йде мова, дійсно знаходиться на зображенні, а класифікатор відмітив об'єкт на зображенні як позитивний результат. Неправильне позитивне означає, що процес маркування помилково визначає, що об'єкт розташований на зображенні, хоча це не так. Помилкове негативне наслідування відбувається тоді, коли класифікатор не може виявити фактичний об'єкт на зображенні, а справжнє негативне значення означає, що об'єкт якого нема на зображенні не був знайдений класифікатором, оскільки він не є об'єктом, про який йде мова. Для того, щоб добре працювати, кожен етап каскаду повинен мати низький рівень помилково-негативного результату пошуку, коли фактичний об'єкт класифікується як не об'єкт, то класифікація цієї гілки зупиняється, і ніяким чином не можна виправити помилку. Проте кожен етап може мати відносно високу помилково-негативний результат, тому що навіть якщо n -й етап класифікує не об'єкт як фактично об'єкт, то ця помилка може бути виправлена в $n+1$ -му та наступних стадіях класифікатора.

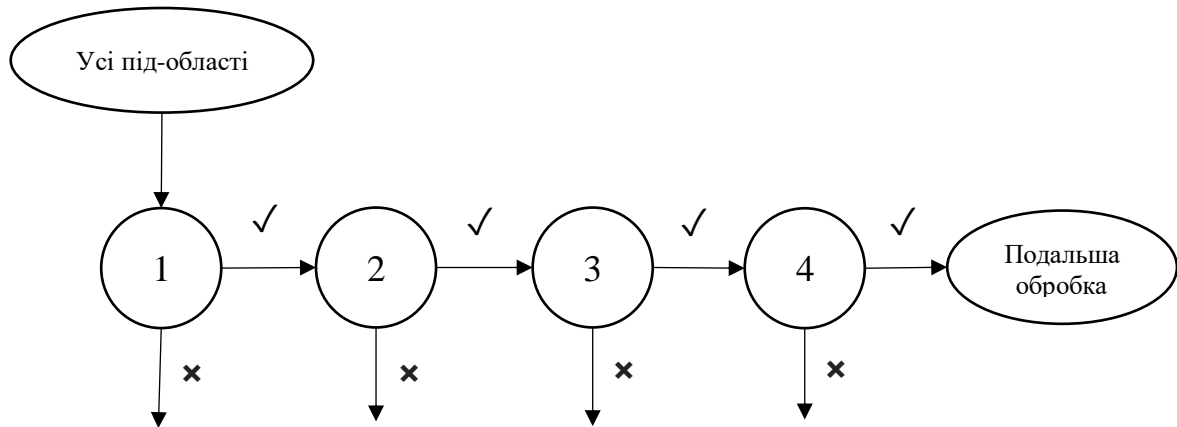


Рис. 2.1 – Поетапна робота класифікатору

На рисунку 2.1 можна бачити загальний принцип роботи каскадного класифікатору.

2.2.1. Haar Cascade Classifier

Haar Cascade Classifier [14] - це підхід до машинного навчання, алгоритм, створений Паулом Віолою та Майклом Джонсом; який навчається з безлічі позитивних зображень (з обличчями) та негативними зображеннями (без обличчя).

Haar Cascade - це класифікатор, який використовується для виявлення об'єкта, для якого він був підготовлений. Кращі результати отримуються за допомогою високоякісних зображень та збільшення кількості етапів, для яких класифікатор навчається. Можна також використовувати попередньо визначені Каскади Haar, які доступні у відкритому доступі [15].

Haar-Feature - це майже так само, як ядро в CNN, за винятком того, що в CNN значення ядра визначаються тренуванням, тоді як Haar-Feature визначається вручну.

Ось деякі Haar-features. Перші два - це "крайові функції", які використовуються для виявлення країв. Третій - це "лінійна функція", а четверта - "чотири прямокутника", найчастіше використовується для виявлення похилої лінії.

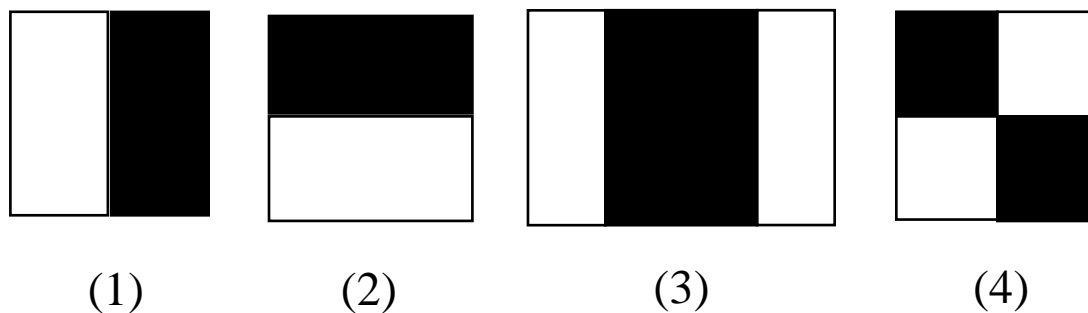


Рисунок 2.2 – Звичайні Haar-features

Вони можуть виглядати приблизно так:

-1	-1	5
-1	-1	5
-1	-1	5

(1)

5	5	5
-1	-1	-1
-1	-1	-1

(2)

-1	5	-1
-1	5	-1
-1	5	-1

(3)

5	-1	-1
-1	5	-1
-1	-1	5

(4)

Рисунок 2.3 – Представлені числами

Кожне ядро 3x3 рухається по всьому зображенню і робить множення матриці кожною частиною 3x3 зображення, підкреслюючи деякі функції та згладжуючи інші.

Haar-features хороші при виявленні країв та ліній. Це робить його особливо ефективним у виявленні обличчя. Наприклад, на зображенні людини Haar-features зможе виявити її око (область, яка є темною на вершині та яскравіше внизу).

Приклад роботи можна описати наступним чином. Скажімо, ми маємо базу даних зображень з людськими обличчями. Зазвичай, область очей темніше, ніж область щоки. Тому Haar-features для виявлення обличчя - це сукупність двох сусідніх прямокутників, які лежать над оком та областю щоки. Положення цих прямокутників визначається відносно вікна виявлення, яке діє як обмежувальне поле для цільового об'єкта (обличчя в даному випадку).



Рисунок 2.4 – може використовуватися для виявлення антропометричних точок.

Однак, оскільки Haar-features повинні бути визначені вручну, існує певна межа типів речей, які вона може виявити. Якщо ви дасте класифікатору (нейронну мережу або будь-який алгоритм, що виявляє обличчя), то він зможе виявляти об'єкти лише з чіткими краями та лініями. Навіть як детектор обличчя, якщо ми трохи маніпулюємо обличчям (скажімо, прикриваємо очі сонячними окулярами або нахилиємо голову в бік), класифікатор на основі Haar, можливо, не зможе розпізнати обличчя. З іншого боку, згорткове ядро має більш високий ступінь свободи (оскільки це визначається тренуванням), і він може розпізнавати частково захищені обличчя (залежно від якості навчальних даних).

З переваг є те що Haar-Features не потребує навчання, ми можемо створити класифікатор з порівняно невеликим набором даних. Все що потрібно робити, - це тренувати вагові коефіцієнти для кожної особливості, що дозволяє добре тренувати класифікатор, не маючи багато навчальних зображень. Крім того, класифікатор також має більш високу швидкість виконання так як необхідно порівняно небагато обчислень.

2.2.2. LBP Cascade Classifier

Як і будь-який інший класифікатор, Local Binary Patterns(LBP)[16], також потрібно навчатись сотням зображень. LBP - це візуально-текстурний дескриптор, а обличчя також складаються з мікро-візуальних моделей.

Таким чином, LBP-Features використовуються для створення вектору характеристик, який класифікує обличчя з не обличчя.

Кожне навчальне зображення поділяється на деякі блоки, як показано на малюнку нижче.



Рисунок 2.5 – Матриця LBP

Для кожного блоку LBP переглядає 9 пікселів (вікно 3×3) одночасно, і з особливим інтересом до пікселя, розташованого в центрі вікна.

Порівнюються центральне значення пікселя зі значенням пікселя кожного сусіда у вікні 3×3 . Для кожного сусіднього пікселя, більшого або дорівнює пікселю центру, він встановлює його значення до 1, а для інших він встановлює їх на 0.

Після цього оновлені значення пікселів (які можуть бути або 0 або 1) за годинниковою стрілкою і утворюють двійкове число. Далі, він перетворює двійковий номер у десяткове число, а десяткове число - це нове значення центрального пікселя. Ми робимо це для кожного пікселя в блоці.

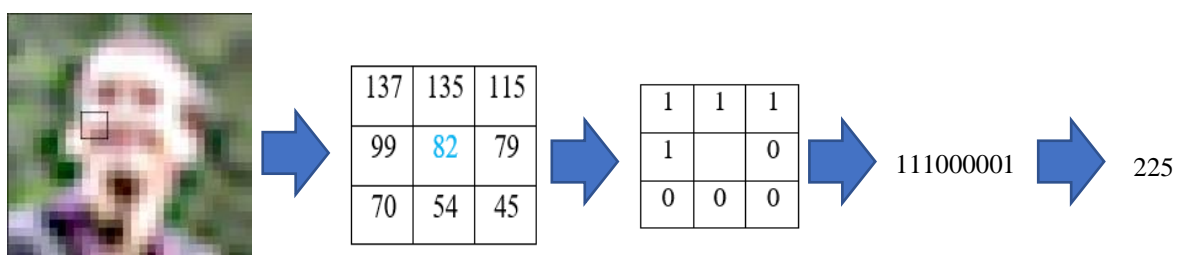


Рисунок 2.6 – Конвертація LBP у бінарну матрицю.

Потім всі значення блоків перетворюються в гістограму, тому тепер ми отримали одну гістограму для кожного блоку на зображенні, як це:

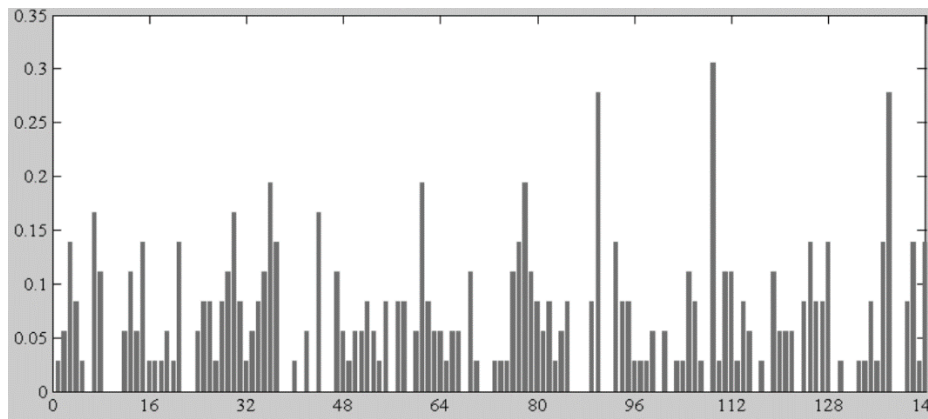


Рисунок 2.6 – Гістограма LBP.

Після цього відбуваються об'єднання гістограми блоків, щоб створити вектор одного об'єкта для одного зображення, який містить всі характеристики, в яких ми зацікавлені. Отже, це те, як ми отримуємо характеристики LBP з зображення.

2.3. Знаходження антропометричних точок обличчя

Інженери та дослідники комп'ютерного зору намагалися зрозуміти людське обличчя вже дуже давно. Найбільш очевидним застосуванням особистого аналізу є розпізнавання обличчя. Але для того, щоб бути в змозі ідентифікувати людину на зображенні, нам спочатку потрібно знайти, де на зображенні знаходиться обличчя. Тому виявлення обличчя - розміщення обличчя на зображенні та повернення області зображення, що містить обличчя, - це необхідна область дослідження. У 2001 році Пол Віола та Майкл Джонс значною мірою вирішили проблему своєю статтею під назвою "Швидке визначення об'єктів за допомогою розширеного каскаду простих функцій"[17].

Виявлення антропометричних точок обличчя є підмножиною проблеми прогнозування форми. Отримавши на вхід зображення, предиктор форми намагається локалізувати ключові точки інтересу вздовж форми.

У контексті антропометричних точок обличчя метою є виявлення важливих форм обличчя, використовуючи методи прогнозування форми.

Таким чином, визначення орієнтирів обличчя має два етапи:

1. локалізація обличчя на зображенні;
2. визначення основних точок обличчя.

Виявлення обличчя можна досягти різними шляхами. У будь-якому випадку фактичний алгоритм, який використовується для виявлення обличчя на зображенні, не має значення. Натомість важливо те, що за допомогою певного методу ми отримаємо рамку для обличчя (тобто, координати обличчя на зображенні).

Існує цілий ряд детекторів точок обличчя, але всі методи, намагаються локалізувати наступні області обличчя: рот, ніс, права та ліва брова, очі, контур обличчя.

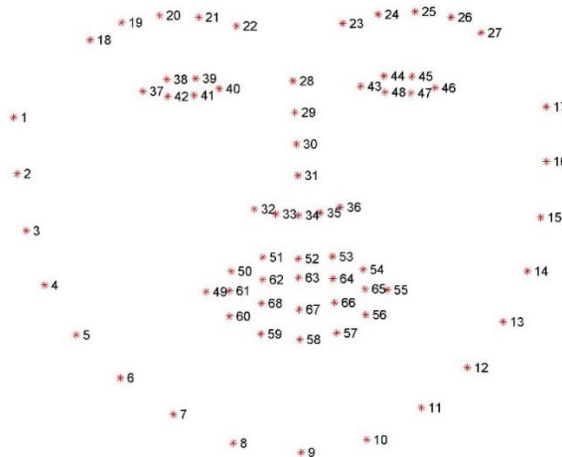


Рис. 2.7 – Розташування 68 антропометричних точок обличчя

Антропометричні точки обличчя можуть бути використанні для багатьох цілей, наприклад таких:

- Виокремлення очей на обличчі для того щоб визначити що водій не заснув за кермом, для застосування в відео реєстраторах, які знімають водія та салон машини. Шляхом визначення відстані між точками що лежать на верхній та нижній частині ока.
- Виявлення антропометричних точок обличчя покращує розпізнавання обличчя. Точки обличчя можуть бути використані для вирівнювання

зображень обличчя так що після вирівнювання розташування обличчя на всіх зображеннях приблизно однакове. Алгоритми розпізнавання обличчя, навчені з вирівняними зображеннями, будуть виконувати набагато краще, і це було підтверджено багатьма науковими дослідженнями[18].

- Визначення пози голови. Після того як ви знаєте кілька точок обличчя, ви також можете визначити позу голови. Іншими словами, ви можете з'ясувати, як голова орієнтована в просторі або куди людина дивиться.
- Морфінг обличчя. Лицьові точки можуть бути використані для вирівнювання обличчя, які потім можуть бути змінені, щоб створювати міжобразні зображення.
- Віртуальний макіяж. Виявлені точки можуть використовуватись для розрахунку контурів рота, очей тощо, для накладання віртуальної косметики.
- Заміна обличчя. Якщо у вас є визначенні точки на двох різних зображеннях обличчя то таким чином можна легко замінити одне обличчя на інше.

2.4. OpenCV

OpenCV (Open Source Computer Vision Library) [15] – це бібліотека програмного забезпечення з відкритим кодом для комп'ютера та машинного навчання. OpenCV був побудований, щоб забезпечити загальну інфраструктуру для комп'ютерних програм зору та прискорити використання сприйняття машин у комерційних продуктах. OpenCV, що є продуктом ліцензії BSD, дозволяє підприємствам легко використовувати та змінювати код.

Бібліотека має понад 2500 оптимізованих алгоритмів, що включає в себе повний комплект як класичного, так і сучасного комп'ютерного бачення та алгоритмів машинного навчання. Ці алгоритми можуть бути використані для виявлення та розпізнавання облич, визначення об'єктів, класифікації людських дій у відео, рухання руху камери, відстеження рухомих об'єктів, витягування 3D-

моделей об'єктів, створення 3D-точкових хмар із стереокамер, зшивання зображень разом для отримання високої роздільної здатності зображення цілісної сцени, знайти схожі зображення з бази даних зображень, видалити червоні очі з зображень, створених за допомогою спалаху, стежити за рухами очей, визнати декорації та встановити маркери для накладання його на додану реальність тощо. Бібліотека широко використовується в компаніях, дослідницьких групах та урядових структурах.

Програми з використанням OpenCV застосовуються для: виявлення вторгнень в відео спостереження, контроль за допомогою шахтного обладнання, допомагають роботам в навігації та виявленні об'єктів, виявлення аварій, інтерактивне мистецтво, перевірки злітно-посадкові смуги, перевірки етикетки на продуктах на заводах у всьому світі, для швидкого виявлення обличчя.

Він має інтерфейси C ++, Python, Java та MATLAB і підтримує Windows, Linux, Android та Mac OS. OpenCV спрямовується в основному на додатки в режимі реального часу та використовує інструкції MMX та SSE, коли це можливо. В даний час активно розвиваються повнофункціональні інтерфейси CUDA та OpenCL. OpenCV написано спочатку на C++ і має шаблонний інтерфейс, який працює без проблем з контейнерами STL.

Основними характеристиками OpenCV є:

- відкритий вихідний код;
- поширюється за ліцензією BSD;
- наявність реалізації каскадних класифікаторів;
- наявність реалізації детекторів антропометричних точок;
- наявність зовнішніх інтерфейсів для різних мов програмування.

Однак OpenCV є тільки інструментом для розробки програмного забезпечення із застосуванням машинного навчання та для його використання потрібно розробити програмну оболонку для використання в розв'язанні поставлених задач та використання для роботи із відео джерелом інформації.

3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

В даному розділі описані обрані засоби розробки та реалізації системи пошуку об'єктів на зображеннях та проведені обчислювальні експерименти з тестовими завданнями та приведені їх результати в вигляді таблиць та рисунків. Наведено найбільш ефективні набори параметрів роботи класифікаторів для вирішення певних задач пошуку.

3.1. Засоби розробки

В даній роботі було застосовано бібліотеку EmguCV [19].

EmguCV – це кроссплатформена огортка бібліотеки обробки зображень OpenCV для застосуванні в .NET. Вона дозволяє викликати функції OpenCV з мов сумісних з .NET, таких як C#, VB, VC ++, IronPython тощо. Обгортку можна скомпілювати в Visual Studio, Xamarin Studio та Unity, вона може працювати на Windows, Linux, Mac OS, iOS, Android і Windows Phone.

3.1.1. Платформи та мови програмування

Для вирішення поставленого завдання необхідно використовувати функціональну, ефективну і зручну платформу для розробки, що дозволяє застосовувати принципи об'єктно-орієнтованого програмування. В якості такої платформи була обрано середовище .NET [29, 30, 31].

Перевагами цієї технології є:

- Об'єктно-орієнтований підхід - платформа .NET спочатку будується на принципах об'єктно-орієнтованого програмування.
- Потужний інструментарій - поставляється разом із середовищем бібліотека базових класів володіє достатнім функціоналом для вирішення завдань практично будь-якої складності.

- Поділ коду - в .NET спосіб поділу коду між додатками значно відрізняється від попередніх реалізацій за рахунок використання збірок. Складання володіють формальними засобами для управління версіями і допускають одночасне існування поруч кількох різних версій збірок.

- Підтримка мов високого рівня - це властивість сприятливо позначається на зручності використання, швидкості написання і читабельності коду, що вкрай важливо для подальшої підтримки програми.

Visual Studio - Середовище розробки Visual Studio, що поставляється разом з .NET, надає необхідний інструментарій для ефективного і швидкого створення додатків з графічним інтерфейсом.

Поява технології .NET спричинило масову реконструкцію деяких мов програмування, які прагнуть використовувати ті чи інші можливості платформи, такі як C ++ і Visual Basic. Microsoft вирішили запропонувати розробникам альтернативу - мова, орієнтований спеціально .NET і створили C #. Самі розробники мови описують його, як простий, сучасний, об'єктно-орієнтована і безпечний мову програмування. Синтаксично C # нагадує C ++ і Java, що дозволило програмістам за досить короткий час вивчити тонкощі нової мови.

Незважаючи на те, що C # і .NET призначені в першу чергу для веб-розробки, їх також активно застосовують для створення додатків, які повинні встановлюватися на машині кінцевого користувача, де і буде виконуватися вся обробка даних. Розробку таких додатків забезпечує бібліотека Windows Forms, що дозволяє проектувати графічний інтерфейс. Система, описана в даній роботі, розроблена саме за допомогою бібліотеки Windows Forms.

Загальномовне середовище виконання (CLR) грає ключову роль в організації всієї технології. Вона підтримує сувору систему правил, якою має слідувати проміжний мову. Ця мова є код, який називається керованим. Слід зазначити, що вхідний потік даних може являти собою і некерований код, а також керований і некерований одночасно. Некеровані фрагменти вхідних даних є звичайними машинними інструкціями, які без змін надходять в процесор.

Показником на те, що вхідний потік містить керований код, є спеціальний біт в заголовку файлу. Керований код породжує об'єкти з керованим часом життя. Такі об'єкти автоматично знищуються, коли потреба в них відпадає. Це властивість наділяє середу CLR здатністю боротьби з витокami пам'яті, що сприятливо позначається на продуктивності програм.

Для створення об'єктів з керованим часом життя в керований код поміщаються також метадані, які містять деякі інструкції про породжуваних об'єктах, їх властивості, методи і події. Керований код перед запуском повинен пройти процес верифікації, який визначає, чи буде код намагатися отримати доступ до неправильних адресами пам'яті або виконувати інші невірні дії. Код, що задовольняє верифікації, називається надійним. Можливість верифікації дозволяє CLR забезпечувати високий рівень ізоляваності об'єктів при мінімальному зниженні продуктивності.

CLR полегшує розробку компонентів і програм, об'єкти яких взаємодіють за допомогою мови СІЛ. Об'єкти, написані на різних мовах, можуть взаємодіяти один з одним, і їх поведінка може бути тісно пов'язаним. Міжмовна взаємодія можлива тому, що мовні компілятори і інструменти цільової машини використовують загальну систему типів, певну в СТС, і слідує загальним правилам при створенні, використанні та зв'язуванні типів.

CLR забезпечує інфраструктуру, яка дозволяє управляти процесом виконання машинного коду, а також надає різні служби, які можуть бути використані під час виконання. Перед викликом методу він повинен бути скомпільовано в машинні інструкції. Кожен метод, для якого є СІЛ-код, повинен спочатку генеруватися в машинний і потім виконуватися. Кожен наступний раз компілятор не викликається, але використовується створений ним код. Цей процес повторюється до кінця прогону.

Під час виконання керований код отримує додаткове обслуговування, до якого відносяться збір сміття, підвищений захист, взаємодія з некерованим

кодом, підтримка міжмовної налагодження, поліпшення розповсюдження програм і контролю версій.

Переваги мови програмування C#:

— мова програмування C# претендує на справжню об'єктну орієнтованість.

— мова програмування C# покликаний реалізувати компонентно-орієнтований підхід до програмування, який сприяє меншій машинно-архітектурній залежності результуючого програмного коду, більшій гнучкості, переносимості і легкості повторного використання програм.

— принципово важливою відмінністю від попередників є початкова орієнтація на безпеку коду.

— розширена підтримка подієво-орієнтованого програмування.

— мова програмування C# є «рідним» для створення додатків в середовищі Microsoft .NET, оскільки найтісніше і ефективно інтегрований з нею.

3.1.2. Засоби розробки системи пошуку об'єктів на зображеннях

Microsoft Visual Studio – лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати і як відладчик рівня вихідного коду, і як відладчик машинного рівня. Решта вбудованих інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-

редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

Технологія WPF (Windows Presentation Foundation) є частиною платформи .NET і являє собою підсистему для побудови графічних інтерфейсів.

Якщо при створенні традиційних додатків на основі WinForms за відображення елементів управління і графіки відповідали такі частини ОС Windows, як User32 і GDI +, то додатки WPF засновані на DirectX. У цьому полягає ключова особливість рендеринга графіки в WPF: використовуючи WPF, значна частина роботи по відображенню графіки, як найпростіших кнопочок, так і складних 3D-моделей, лягає на графічний процесор на відеокарті, що також дозволяє скористатися апаратним прискоренням графіки.

В основі WPF лежить потужна інфраструктура, заснована на API-інтерфейсі графіки з апаратним прискоренням, який зазвичай використовується в сучасних комп'ютерних іграх. Це означає можливість застосування розвинених графічних ефектів, не сплачуючи за це продуктивністю, як це було в Windows Forms.

Фактично навіть стають доступними такі розширені засоби, як підтримка відеофайлів і тривимірного вмісту. Використовуючи ці засоби (при наявності хорошого інструменту графічного дизайну), можна створювати для користувача інтерфейси і візуальні ефекти, які були просто неможливі в Windows Forms.

Хоча новітні засоби відео, анімації та 3-D часто залучають максимум уваги в WPF, важливо відзначити, що WPF можна застосовувати для побудови

звичайних Windows-додатків зі стандартними елементами управління і звичним зовнішнім виглядом. Фактично використовувати стандартні елементи управління в WPF так само легко, як і в Windows Forms. Більш того, WPF розширює засоби, адресовані саме бізнес-розробникам, включаючи значно вдосконалену модель прив'язки даних, набір класів для друку вмісту і управління чергами друку, а також засоби роботи з документами для відображення величезних обсягів тексту. Доступна навіть модель для побудови додатків на основі сторінок, які працюють в Internet Explorer і можуть запускатися з веб-сайту — і все це без звичних попереджень про безпеку або докучливих запрошень до установки. Взагалі кажучи, WPF комбінує краще зі світу Windows-розробки з новітніми технологіями для побудови сучасних, графічно розвинених, призначених для користувача інтерфейсів.

Навіть якби єдиною перевагою WPF було апаратне прискорення через DirectX, це вже стало б значним удосконаленням, хоч і не революційним. Однак WPF насправді включає цілий набір високорівневих служб, орієнтованих на прикладних програмістів.

Переваги WPF:

— використання традиційних мов .NET-платформи – C# і VB.NET для створення логіки додатка;

— можливість декларативного визначення графічного інтерфейсу за допомогою спеціальної мови розмітки XAML, заснованої на xml, і яка представляє альтернативу програмному створенню графіки та елементів управління, а також можливість комбінувати XAML і C# / VB.NET;

— незалежність від дозволу екрану: оскільки в WPF всі елементи вимірюються в незалежних від пристрою одиницях, додатки на WPF легко масштабуються під різні екрани з різним дозволом;

— нові можливості, яких складно було досягти в WinForms, наприклад, створення тривимірних моделей, прив'язка даних, використання таких елементів, як стилі, шаблони, теми і ін.;

— хороша взаємодія з WinForms, завдяки чому, наприклад, в додатках WPF можна використовувати традиційні елементи управління з WinForms;

— багаті можливості по створенню різних додатків: це і мультимедіа, і двомірна та тривимірна графіка, і багатий набір вбудованих елементів управління, а також можливість самим створювати нові елементи, створення анімацій, прив'язка даних, стилі, шаблони, теми і багато іншого;

— апаратне прискорення графіки – незалежно від того, чи працюєте ви з 2D або 3D, графікою або текстом, всі компоненти програми транслюються в об'єкти, зрозумілі Direct3D, і потім візуалізуються за допомогою процесора на відеокарті, що підвищує продуктивність, робить графіку більш плавною;

— створення додатків під безліч ОС сімейства Windows — від Windows XP до Windows 10.

Якщо вам коли-небудь доводилося проходити через процес проектування додатків Windows Forms з використанням спеціальної графіки в командному середовищі, ви, безсумнівно, стикалися з масою розчарувань. Навіть якщо інтерфейс спроектований з нуля дизайнером графіки, він повинен бути відтворений в коді C#. Зазвичай дизайнеру графіки просто доводиться готувати макет, який потім потрібно болісно транслювати в працюючий додаток.

У WPF ця проблема вирішується за допомогою XAML. При проектуванні WPF-додатки в Visual Studio створене вікно не транслюється в код. Замість цього воно серіалізується в набір дескрипторів XAML. Після запуску програми ці дескриптори використовуються для генерації об'єктів, що становлять призначений для користувача інтерфейс.

Однією з важливих особливостей є використання мови декларативної розмітки інтерфейсу XAML, заснованої на XML.

Мова розмітки XAML (Extensible Application Markup Language) — яка використовується для ініціалізації об'єктів в технологіях на платформі .NET. Стосовно до WPF (а також до Silverlight) дана мова використовується перш за все для створення призначеного для користувача інтерфейсу декларативного

шляху. Хоча функціональність XAML тільки графічними інтерфейсами не обмежується: дана мова також використовується в технологіях WCF і WF, де вона ніяк не пов'язана з графічним інтерфейсом. Тобто її область ширша. Стосовно WPF ми будемо говорити про неї найчастіше саме як про мову розмітки, яка дозволяє створювати декларативним шляхом інтерфейс. Однак знову ж таки повторюся, зводити XAML до одного інтерфейсу було б неправильно, і далі на прикладах ми це побачимо.

Але XAML не є обов'язковою частиною програми, ми взагалі можемо обходитися без неї, створюючи всі елементи в файлі пов'язаного з ним коду на мові C#. Однак використання XAML все-таки дає деякі переваги:

— можливість відокремити графічний інтерфейс від логіки додатку, завдяки чому над різними частинами програми можуть відносно автономно працювати різні фахівці: над інтерфейсом — дизайнери, над кодом — програмісти;

— компактність, зрозумілість;

— код на XAML відносно легко підтримувати;

— при компіляції програми в Visual Studio код в XAML-файлах також компілюється в бінарне представлення коду XAML, яке називається BAML (Binary Application Markup Language). І потім код BAML вбудовується в фінальній збірці як додаток exe або dll-файл.

3.2. Реалізація класів системи пошуку об'єкта

Розроблене програмне забезпечення написано на мові програмування C# з використанням технології WPF. Система пошуку об'єктів реалізована за допомогою класу `Detector`.

Клас `Detector` контролює процес ініціалізації та роботи каскадних класифікаторів. При його ініціалізації використовується вказаний каскадний класифікатор. Окрім того цей клас відповідає за відображення обмежувальних прямокутників навколо знайдених об'єктів.

Клас FacialLandmarksDetector процес ініціалізації та роботи детектору антропометричних точок обличчя, для його роботи зображення попередньо повинне буди оброблене каскадним класифікатором пошуку обличчя. Окрім того цей клас відповідає за нанесення на зображення обличчя антропометричних точок які були знайденні детектором.



Рис. 3.1 – Схема реалізації програмного забезпечення

На рисунку 3.1 можна бачити загальну схему реалізації програмного забезпечення.

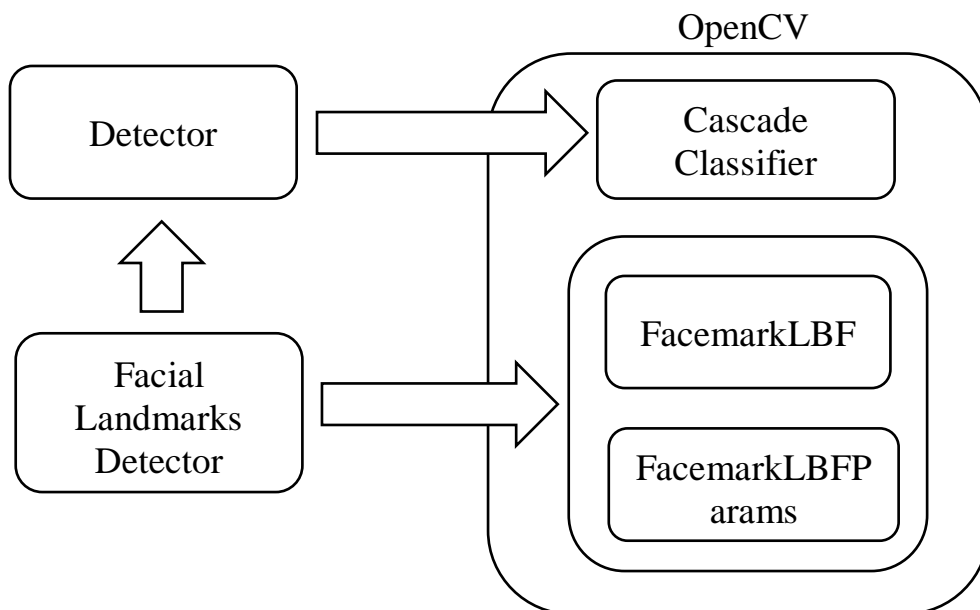


Рис. 3.2 – Схема залежності класів програмної реалізації

В програмне забезпеченні також присутні каскадні класифікатори які були отриманні у вільному доступі разом з бібліотекою OpenCV.

Таблиця 3.1 – Список класифікаторів які присутні у програмному забезпеченні.

№	Призначення	Назва класифікатору	Тип класифікатору
1	Пошук очей	haarcascade_eye	Haar
2	Пошук очей в окулярах	haarcascade_eye_tree_eyeglasses	Haar
3	Пошук передньої частини обличчя	haarcascade_frontalface_default	Haar
		haarcascade_frontalface_alt	
		haarcascade_frontalface_alt2	
		haarcascade_frontalface_alt_tree	
4	Пошук людини	haarcascade_fullbody	Haar
5	Нижня частина тіла	haarcascade_lowerbody	Haar
6	Номерні знаки	haarcascade_russian_plate_number	Haar
		haarcascade_licence_plate_rus_16stages	
7	Пошук обличчя котів	haarcascade_frontalcatface	Haar
		haarcascade_frontalcatface_extended	Haar
		lbpcascade_frontalcatface	LPB
8	Пошук передньої частини обличчя	lbpcascade_frontalface	LPB
		lbpcascade_frontalface_improved	

В зв'язку з наявністю декількох класифікаторів які вирішують одну задачу то існує необхідність у визначенні найбільш ефективного класифікатора в певній категорії задач.

3.3. Методика роботи користувача з програмною системою

3.3.1. Інсталяція та системні вимоги

Для роботи створеного програмного забезпечення необхідна присутність .NET Framework на персональному комп'ютері користувача

На малюнках (4.1-4.5) можна бачити всі етапи настальювання програмного продукту.

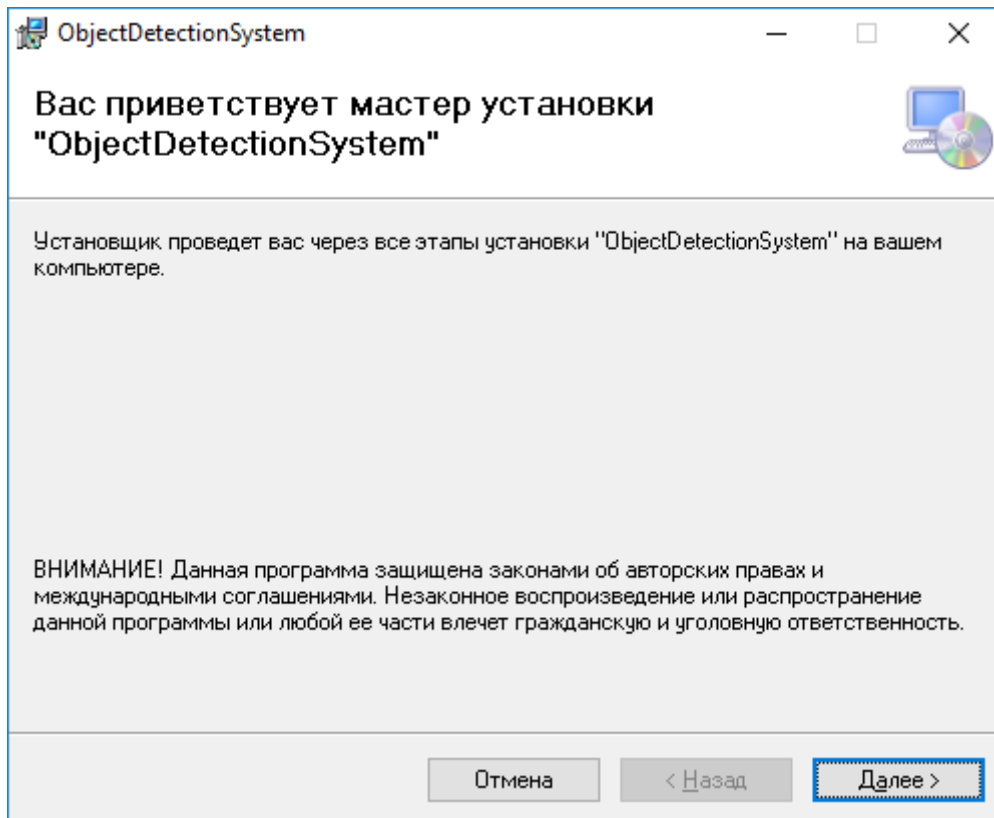


Рисунок 3.3 – Головне вікно істальатора програмного продукту

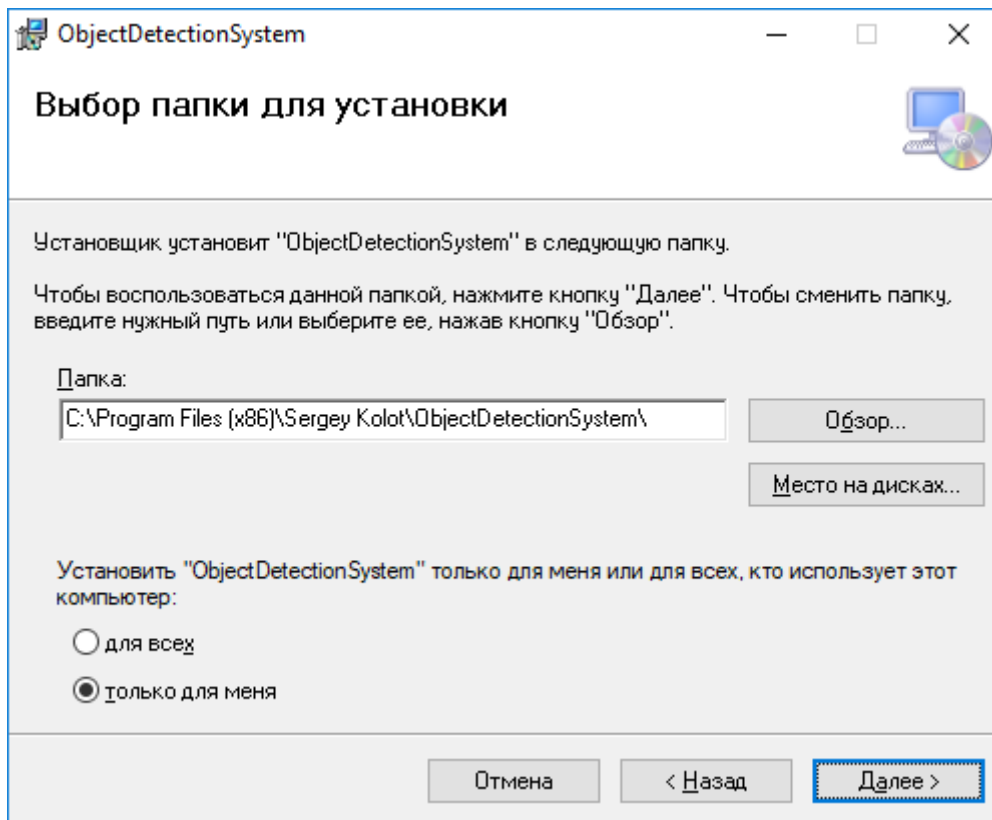


Рисунок 3.4 – Вибір місця інсталяції програмного продукту

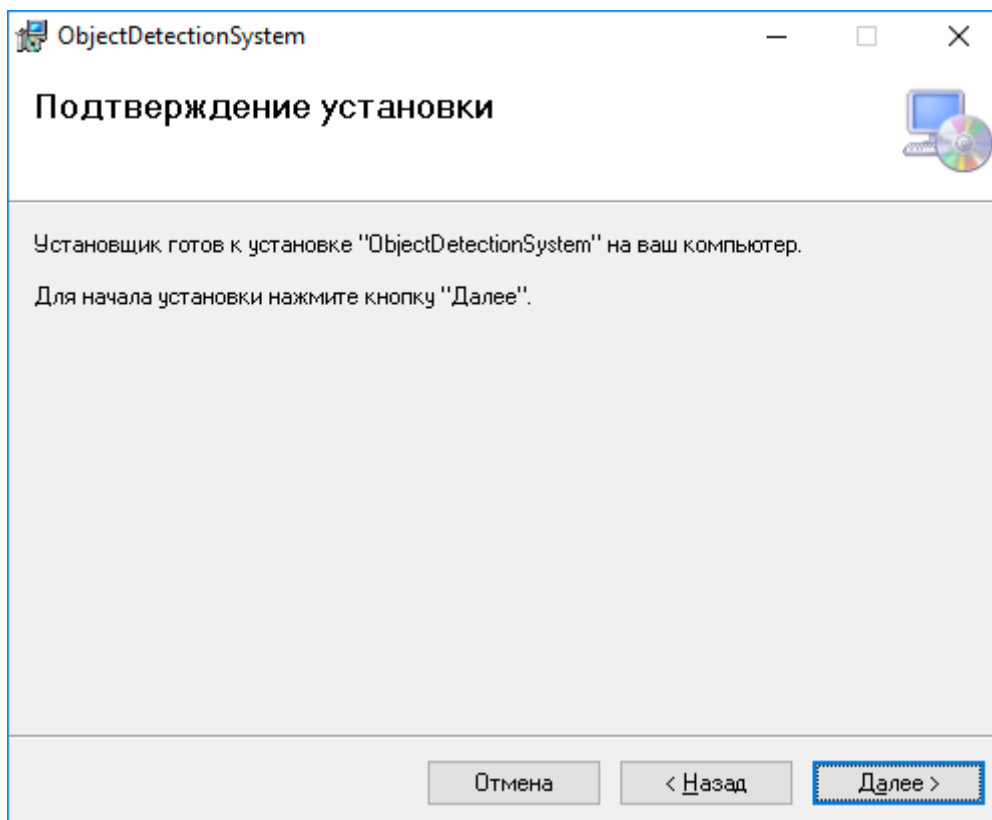


Рисунок 3.5 – Початок процесу встановлення

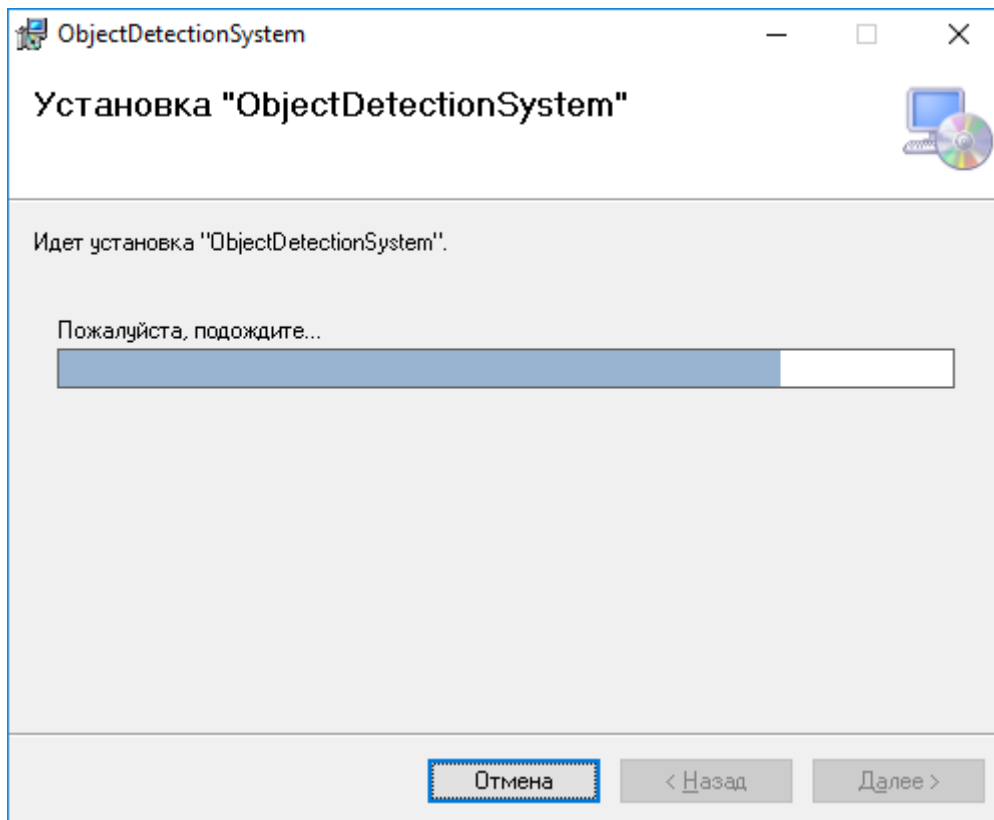


Рисунок 3.6 – Процесс встановлення програмного продукту

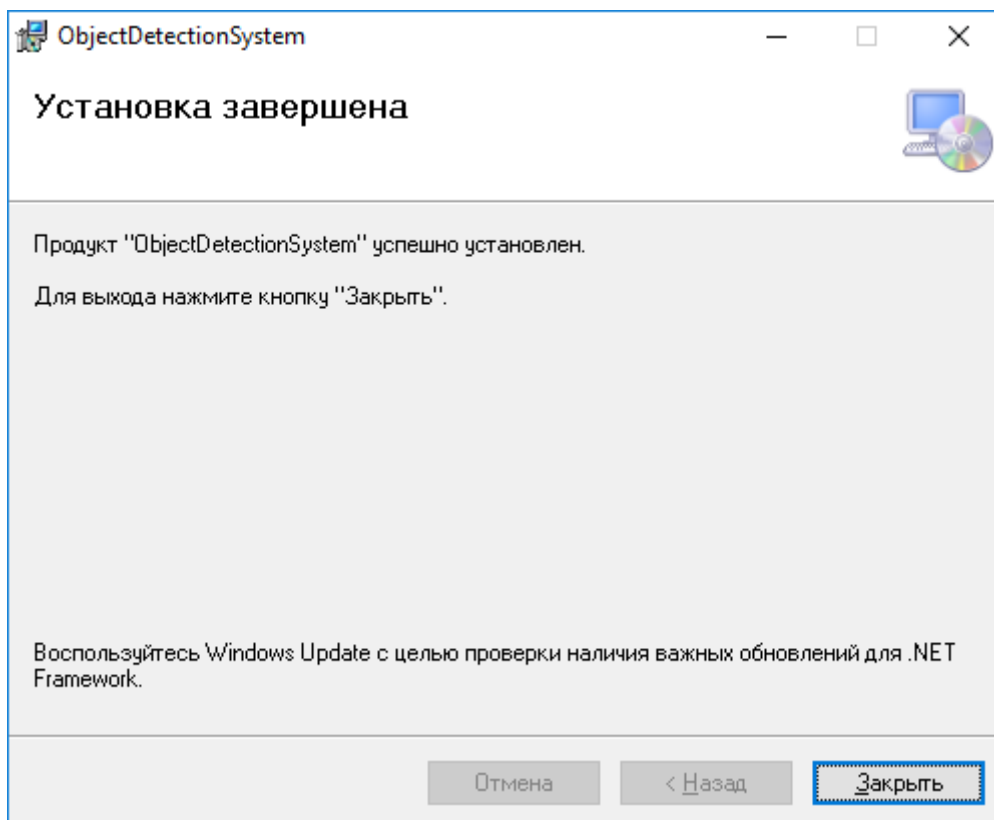


Рисунок 3.7 – Завершения істалювання програмного продукту

3.3.2. Сценарії роботи користувача з системою

На рисунку 3.8 зображене головне вікно програми після її запуску.

У програмі присутні два розділи:

- розділ роботи з відео камерою та одиночними зображеннями;
- розділ перегляду результатів виконаних тестів.

Також на рисунку 3.8 можна побачити область встановлення параметрів роботи каскадних класифікаторів. На рисунку 3.9 можна побачити перелік функцій, які знаходяться в розділі Facial landmarks.

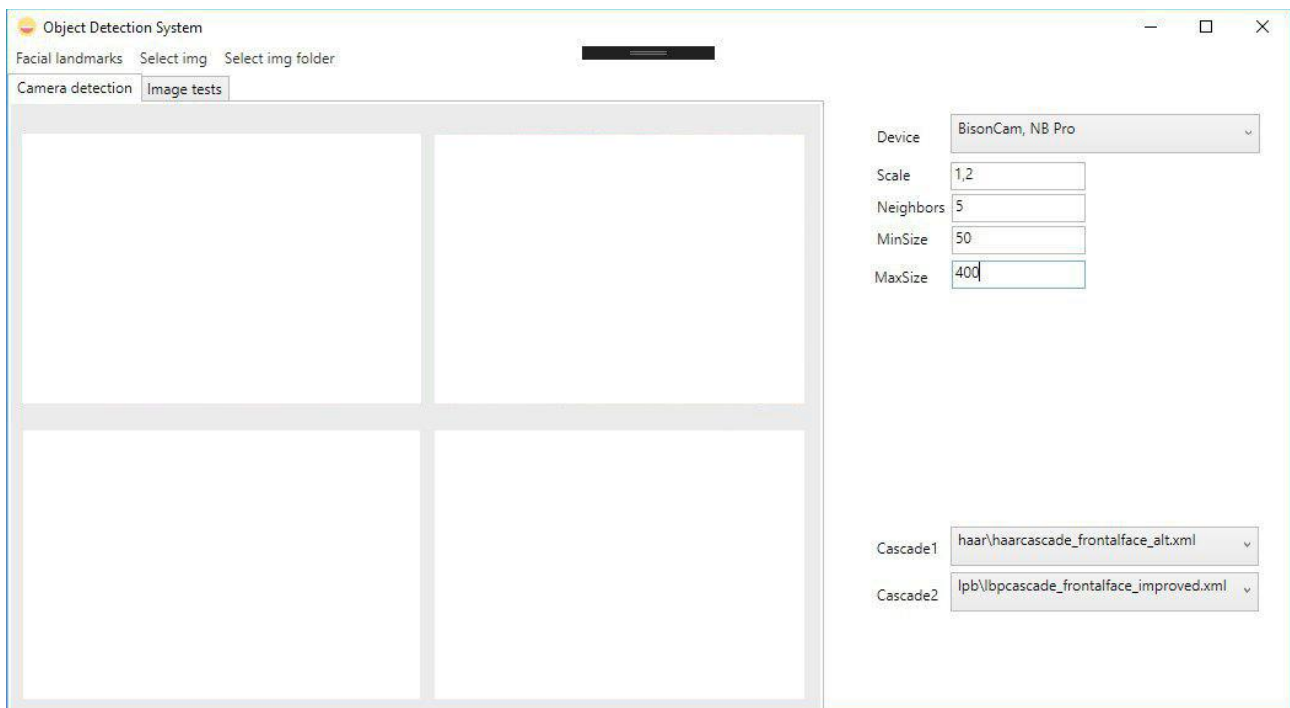


Рисунок 3.8 — Головне вікно програми

Програма працює з усіма знайденими зовнішніми пристроями захоплення відеозображень. В нашому випадку це веб-камера яка вбудована в ноутбучі. Для вибору зовнішнього пристрою є випадючий список з усіма їх назвами.

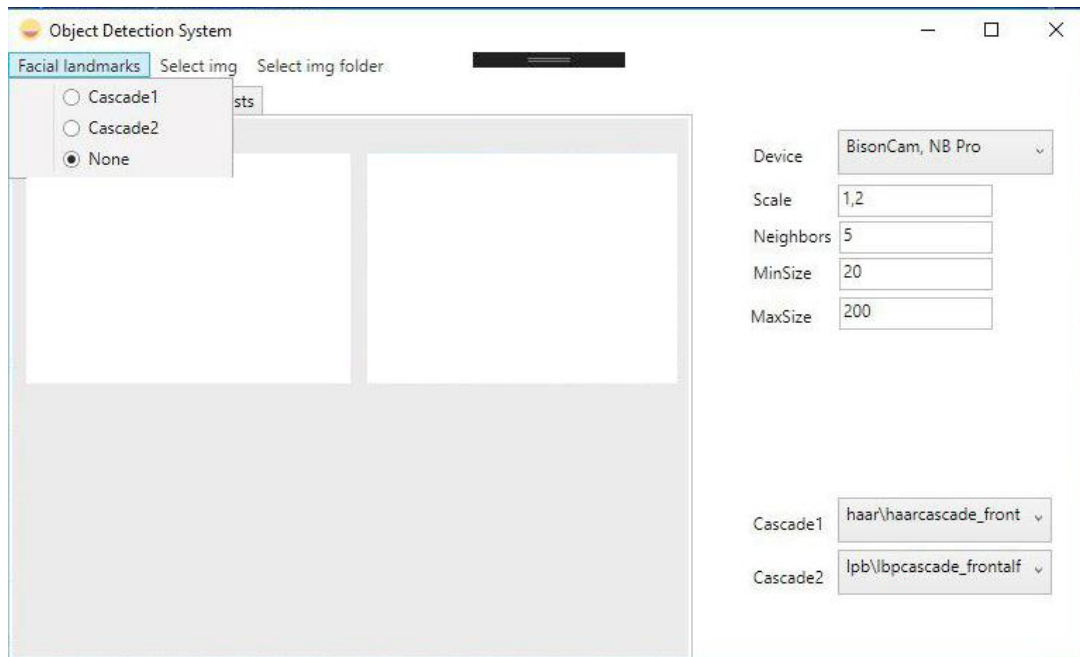


Рисунок 3.9 – Доступні підпункти меню

Під час роботи програми можна змінити вибраний каскадний класифікатор на інший із списку доступних, чи використовувати свій. Його потрібно помістити в папку Resources.

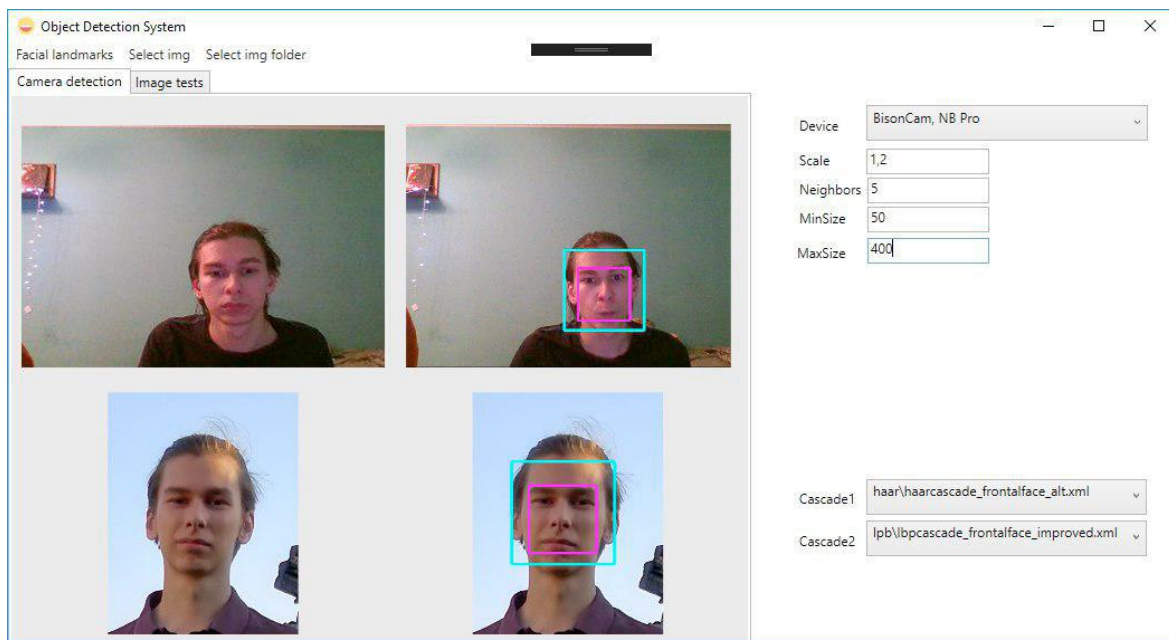


Рисунок 4.8 — Вікно програми під час роботи.

На рисунку 4.9 можна бачити приклад роботи програми під час використання каскадних класифікаторів для виокремлення обличчя та знаходження на основі цих областей антропометричних точок обличчя.

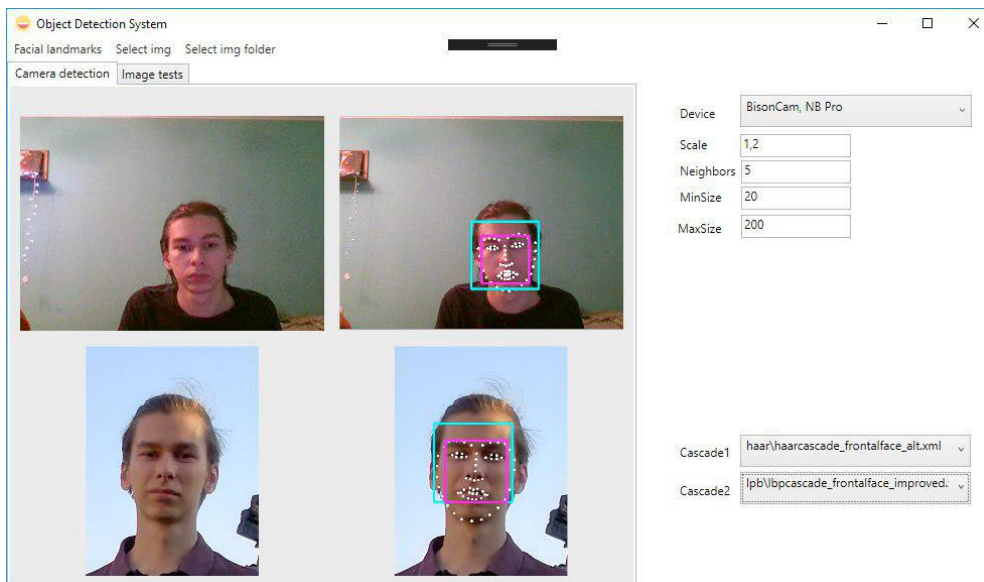


Рис 4.9 – Вікно програми під час роботи

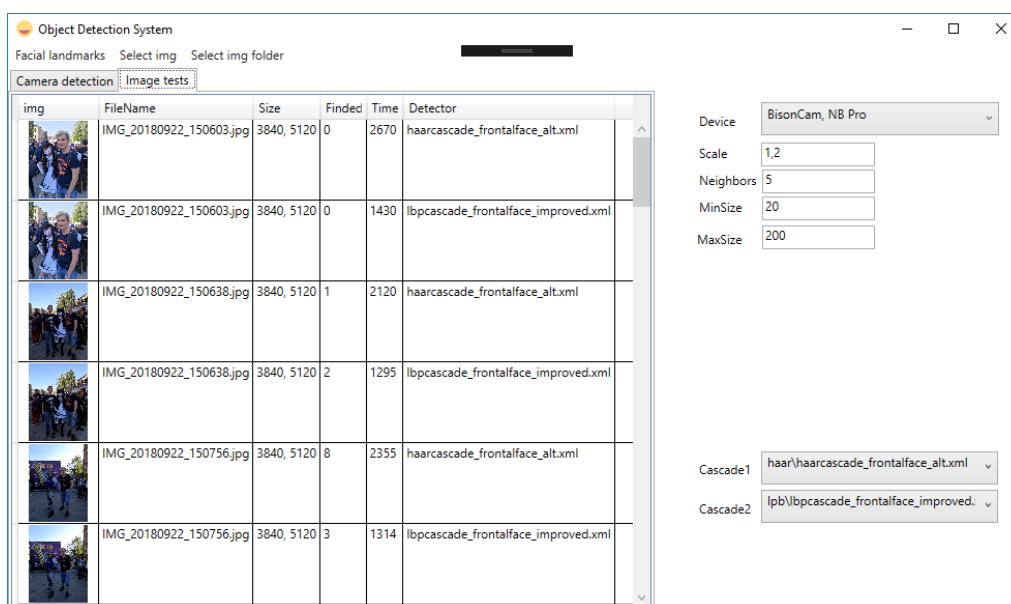


Рисунок 4.10— Вікно програми під час перегляду результатів

На рисунку 4.10 можна побачити таблицю з усіма результатами пошуку заданих об'єктів на зображеннях та опису проведеного пошуку. Данні з цієї таблиці можна легко скопіювати в інше місце.

4. ОБЧИСЛЮВАЛЬНІ ЕКСПЕРИМЕНТИ

У даному розділі наведено порівняння та визначення найбільш ефективного класифікатору та наборів параметрів для вирішення поставлених тестових задач.

4.1. Виявлення обличчя

Вхідною інформацією підсистеми є, зображення з фото чи відео камери. Вихідною інформацією є всі знайдені на зображенні обличчя.

Було проведено 34 експерименти результати зведено до таблиці 4.1.

На рисунку 4.1 наведено результати виокремлення обличчя для тестів №3,4



Рис. 4.1. – Результат виокремлення обличчя

Таблиця 4.1 – Обчислювальні експерименти по виявленню обличчя.

№	Назва файлу	Розмір	Кіль- кість об- личь	Знай- дено	Швид- кість, мс	Класифікатор
1	20180922_150603.jpg	3840x5120	2	0	2575	haarcascade_frontalface_alt
2	20180922_150603.jpg	3840x5120	2	0	1457	lbpcascade_frontalface_improved
3	20180922_150638.jpg	3840x5120	3	2	2160	haarcascade_frontalface_alt
4	20180922_150638.jpg	3840x5120	3	2	1398	lbpcascade_frontalface_improved
5	20180922_150756.jpg	3840x5120	2	8	2469	haarcascade_frontalface_alt
6	20180922_150756.jpg	3840x5120	2	3	1494	lbpcascade_frontalface_improved
7	20180922_151340.jpg	3840x5120	2	1	3152	haarcascade_frontalface_alt
8	20180922_151340.jpg	3840x5120	2	1	1403	lbpcascade_frontalface_improved
9	20180922_151534.jpg	3840x5120	5	5	3247	haarcascade_frontalface_alt
10	20180922_151534.jpg	3840x5120	5	3	1553	lbpcascade_frontalface_improved
11	20180922_152202.jpg	3840x5120	2	2	3312	haarcascade_frontalface_alt
12	20180922_152202.jpg	3840x5120	2	2	1412	lbpcascade_frontalface_improved
13	20180922_152214.jpg	3840x5120	2	3	3183	haarcascade_frontalface_alt
14	20180922_152214.jpg	3840x5120	2	1	1414	lbpcascade_frontalface_improved
15	20180922_152641.jpg	3840x5120	2	0	3419	haarcascade_frontalface_alt
16	20180922_152641.jpg	3840x5120	2	0	1390	lbpcascade_frontalface_improved
17	20180922_153133.jpg	3840x5120	2	1	3072	haarcascade_frontalface_alt
18	20180922_153133.jpg	3840x5120	2	1	1380	lbpcascade_frontalface_improved
19	20180922_154231.jpg	3840x5120	2	0	2259	haarcascade_frontalface_alt
20	20180922_154231.jpg	3840x5120	2	0	1308	lbpcascade_frontalface_improved
21	20180922_154905.jpg	3840x5120	1	1	2005	haarcascade_frontalface_alt
22	20180922_154905.jpg	3840x5120	1	1	1227	lbpcascade_frontalface_improved
23	20180922_161407.jpg	3840x5120	2	2	2318	haarcascade_frontalface_alt
24	20180922_161407.jpg	3840x5120	2	2	1289	lbpcascade_frontalface_improved
25	20180922_161649.jpg	3840x5120	2	0	2232	haarcascade_frontalface_alt
26	20180922_161649.jpg	3840x5120	2	2	1207	lbpcascade_frontalface_improved
27	20180922_161651.jpg	3840x5120	2	0	2307	haarcascade_frontalface_alt
28	20180922_161651.jpg	3840x5120	2	2	1222	lbpcascade_frontalface_improved
29	20180922_174554.jpg	3840x5120	1	0	2144	haarcascade_frontalface_alt
30	20180922_174554.jpg	3840x5120	1	0	1234	lbpcascade_frontalface_improved
31	20180922_174949.jpg	3840x5120	1	2	2635	haarcascade_frontalface_alt
32	20180922_174949.jpg	3840x5120	1	1	1472	lbpcascade_frontalface_improved
33	20180922_200126.jpg	3840x5120	1	0	2122	haarcascade_frontalface_alt
34	20180922_200126.jpg	3840x5120	1	0	1309	lbpcascade_frontalface_improved

Проаналізувавши таблицю можна зробити висновок що в загалом класифікатор haarcascade_frontalface_alt працює довше але не завжди дає біль

правильні результати, з іншого боку lbpcascade_frontalface_improved в усіх випадках виконує пошук набагато швидше та в деяких випадках знаходить обличчя навіть краще.

4.2. Виявлення номерних знаків

Було проведено 20 експериментів результати зведено до таблиці 4.2.

Таблиця 4.2 – Обчислювальні експерименти по номерних знаків

№	Назва файлу	Розмір	Кількість знаків	Знайдено	Швидкість, мс	Класифікатор
1	P9170008.JPG	1024x768	1	0	39	haarcascade_licence_plate_rus_16stages
2	P9170008.JPG	1024x768	1	1	73	haarcascade_russian_plate_number
3	P9170011.JPG	1024x768	1	0	33	haarcascade_licence_plate_rus_16stages
4	P9170011.JPG	1024x768	1	1	62	haarcascade_russian_plate_number
5	P9170013.JPG	1024x768	1	0	37	haarcascade_licence_plate_rus_16stages
6	P9170013.JPG	1024x768	1	1	70	haarcascade_russian_plate_number
7	P9170017.JPG	1024x768	1	0	35	haarcascade_licence_plate_rus_16stages
8	P9170017.JPG	1024x768	1	2	66	haarcascade_russian_plate_number
9	P9170018.JPG	1024x768	2	2	75	haarcascade_licence_plate_rus_16stages
10	P9170018.JPG	1024x768	2	1	61	haarcascade_russian_plate_number
11	P9170024.JPG	1024x768	1	0	35	haarcascade_licence_plate_rus_16stages
12	P9170024.JPG	1024x768	1	0	65	haarcascade_russian_plate_number
13	P9170042.JPG	640x480	1	0	24	haarcascade_licence_plate_rus_16stages
14	P9170042.JPG	640x480	1	4	29	haarcascade_russian_plate_number
15	P9170051.JPG	640x480	1	0	12	haarcascade_licence_plate_rus_16stages
16	P9170051.JPG	640x480	1	1	25	haarcascade_russian_plate_number
17	P9170052.JPG	640x480	1	0	11	haarcascade_licence_plate_rus_16stages
18	P9170052.JPG	640x480	1	1	24	haarcascade_russian_plate_number
19	PB150002.JPG	640x480	1	0	11	haarcascade_licence_plate_rus_16stages
20	PB150002.JPG	640x480	1	2	23	haarcascade_russian_plate_number



Рис. 4.2. – Результат виокремлення номерних знаків на зображенні

На рисунку 4.2 наведено результати виокремлення номерних знаків для тестів №9,10.

4.3. Виокремлення очей людини

Було проведено 20 експериментів результати зведено до таблиці 4.3.

Таблиця 4.3 – Обчислювальні експерименти по виявленню очей на зображенні.

№	Назва файлу	Розмір	Кількість очей	Знайдено	Швидкість, мс	Класифікатор
1	20180922_150603.jpg	5120x3840	4	4	898	haarcascade_eye
2	20180922_150638.jpg	5120x3840	6	1	769	haarcascade_eye
3	20180922_150756.jpg	5120x3840	4	3	808	haarcascade_eye
4	20180922_151340.jpg	5120x3840	4	2	1083	haarcascade_eye
5	20180922_151534.jpg	5120x3840	10	11	1008	haarcascade_eye
6	20180922_152202.jpg	5120x3840	4	7	1131	haarcascade_eye
7	20180922_152214.jpg	5120x3840	4	6	1552	haarcascade_eye
8	20180922_152641.jpg	5120x3840	4	5	1415	haarcascade_eye
9	20180922_153133.jpg	5120x3840	4	4	1332	haarcascade_eye
10	20180922_154231.jpg	5120x3840	4	5	997	haarcascade_eye
11	20180922_154905.jpg	5120x3840	2	3	806	haarcascade_eye
12	20180922_161407.jpg	5120x3840	3	1	1143	haarcascade_eye
13	20180922_161649.jpg	5120x3840	4	5	912	haarcascade_eye
14	20180922_161651.jpg	5120x3840	4	4	1032	haarcascade_eye
15	20180922_174554.jpg	5120x3840	3	7	979	haarcascade_eye
16	20180922_174949.jpg	5120x3840	6	6	1596	haarcascade_eye
17	20180922_200126.jpg	5120x3840	2	0	913	haarcascade_eye
18	20180922_200610.jpg	5120x3840	2	2	866	haarcascade_eye
19	20180922_200722.jpg	5120x3840	2	2	749	haarcascade_eye

На рисунку 4.3 наведено результати виокремлення очей для тесту №16.



Рис. 4.3. – Результат виокремлення очей на зображенні

За результатами виконання тестових задач каскадний класифікатор `haarcascade_eye` показав позитивні результати та високу швидкість роботи.

4.4. Характеристики роботи класифікаторів

В ході проведення обчислювальних експериментів які були описані раніше було отримано данні про найбільш ефективні набори параметрів роботи каскадних класифікаторів для вирішенні певних задач.

Результати зведено до таблиці 4.4. В таблиці використовуються такі позначення: `Scale` – масштаб вікна пошуку об'єкту, `Min neighbors` – кількість областей, які перетинаються при пошуку об'єкту, `Min size` – мінімальний розмір вікна пошуку об'єкту, `Max size` – максимальний розмір вікна пошуку об'єкту.

Таблиця 4.4 – Параметри роботи класифікаторів.

Класифікатор	Scale	Min neighbors	Min size	Max size
haarcascade_frontalface_alt	1.2	5	20	-
lbpcascade_frontalface_improved				
haarcascade_licence_plate_rus_16stages	1.2	5	-	-
haarcascade_russian_plate_number				
haarcascade_eye	1.2	10	40	200

Так як в програмному забезпеченні присутні і інші класифікатори для вирішення таких самих задач то для них теж можна використовувати ці параметри.

4.5. Порівняння ефективності локалізації антропометричних точок

В ході проведенні обчислювальних експериментів для пошуку антропометричних точок використовувалися різні каскадні класифікатори пошуку обличчя на зображенні. А саме haarcascade_frontalface_alt та lbpcascade_frontalface_improved.

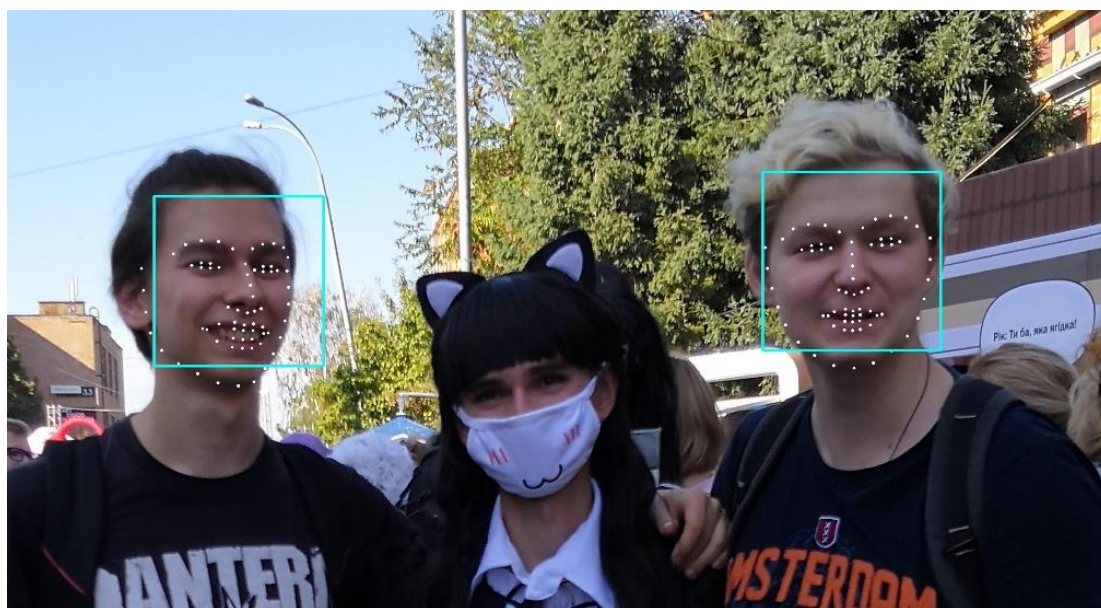


Рис. 4.4 – Приклад роботи детектору антропометричних точок

На рисунку 4.4 можна бачити приклад роботи детектору антропометричних точок обличчя із застосуванням класифікатору `haarcascade_frontalface_alt`.



Рис. 4.5 – Приклад роботи детектору антропометричних точок

На рисунку 4.5 можна бачити приклад роботи детектору антропометричних точок обличчя із застосуванням класифікатору `lbpcascade_frontalface_improved`.

На основі проведених експериментів можна зробити висновок що класифікатор `haarcascade_frontalface_alt` який навчений знаходити більшу частину обличчя та голови набагато ефективніші в роботі з детектором антропометричних точок обличчя так як виявлення точок відбувається з більшою точністю. А класифікатор `lbpcascade_frontalface_improved` хоча знаходить обличчя і швидше проте знайдені обмежувальні координати виділяють лише невелику область обличчя та виявлення точок обличчя виконується не зовсім точно.

5 РОЗРОБКА СТАРТАП ПРОЕКТУ «OBJECT-DETECTION SYSTEM»

5.1 Опис ідеї проекту

В цьому розділі буде проведено аналіз стартап проекту Object-detection system.

Ідея проекту полягає в створенні програмного забезпечення для пошуку заданих об'єктів, що уточнено наведено в таблиці 5.1.

У таблиці 5.1 зображено зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 5.1. Опис ідеї стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система пошуку заданих об'єктів на зображенні з відео камери	Системи ідентифікації осіб	Використання знайденого обличчя для розпізнавання особи
	Біометричні системи ідентифікації	Визначення очей для ідентифікації за роговицею
	Системи контролю руху	Виокремлення автомобільних номерів для подальшого їх розпізнавання
	Системи безпеки керування	Стеження за очима для запобігання засинанню водія

Проведено аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів:

– визначено перелік техніко-економічних властивостей та характеристик ідеї;

– визначаємо попереднє коло проектів-конкурентів, що вже існують на ринку, та проведено збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

– проведено порівняльний аналіз показників: для власної ідеї визначено показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (табл. 5.2).

Таблиця 5.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	потенційні товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Системи спостереження	Системи розпізнавання (звичайні)			
1.	Програма для Windows (керувальна)	+	+	+		+	
2.	Сучасна технологія	+	-	-			+
3.	Висока швидкість роботи	+	-	-			+

На основі визначених сильних та слабких сторін стартап проекту можна провести подальший аналіз та оцінку його конкурентоспроможності. Таких як технологічний аудит ідеї проекту, аналіз ринкових можливостей запуску стартап проекту та розроблення ринкової стратегії проекту.

5.2 Технологічний аудит ідеї проекту

В межах даного підрозділу проводимо аудит технології (мови програмування), за допомогою якої можна реалізувати ідею створення проекту.

Визначення технологічної здійсненності ідеї проекту передбачає аналіз складових які вказані в таблиці 5.3.

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Модуль пошуку об'єкта	Cascade classifiers	+	+
2.	Програма керування	WPF (C#)	+	+
Обрана технологія реалізації ідеї проекту: <i>WPF, Cascade classifiers</i>				

Проаналізувавши таблицю можна зробити висновок що технології які використовуються для реалізації проекту добре підходять для використання та застосування в даному проекті а саме каскадні класифікатори та мова програмування C# за рахунок її кроссплатформеності та універсальності.

5.3 Аналіз ринкових можливостей запуску стартап проекту

Визначимо ринкові можливості, які можна використати під час ринкового впровадження проекту, та ринкові загрози, які можуть перешкодити його реалізації.

Це дозволяє урахувати більшість проблем та шляхів їх вирішення для подальшого корегування та виправлення напрямку розвитку проекту для його наступної його реалізації.

Спочатку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.4).

Таблиця 5.1. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	Не відомо
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	-
5	Специфічні вимоги до стандартизації та сертифікації	-
6	Середня норма рентабельності в галузі (або по ринку), %	Не відомо

Проаналізувати таблицю з характеристиками потенційного ринку важко, так як інформація яку можна отримати в вільному доступі дуже обмежена.

Надалі визначаємо потенційні групи клієнтів, їх характеристики, та формуємо орієнтовний перелік вимог до товару для кожної групи (табл. 5.5).

Таблиця 5.2. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Необхідність виявлення та розпізнавання об'єктів на зображеннях без участі людини	Підприємства, охоронні підприємства, державні органи контролю дорожнього руху, системи розпізнавання особи	Інтеграція в системи розпізнавання особи потребує наявності відкритого виховного коду системи, а державні органи в наявності даної системи тільки у них	Висока швидкість роботи Помірно низький ступінь помилок Невеликі вимоги до застосування

Проаналізувавши таблицю потенційних клієнтів можна зробити висновок що відмінності в потребах різних цільових групах клієнтів виключають широку направленість на всі галузі застосування та краще сконцентруватись лише на декількох з них.

При застосуванні даної технології існують певні загрози у зв'язку з тим вона передбачає що система пошуку може видавати помилки то при її реалізації існують певні фактори загроз (таблиця 5.6).

Таблиця 5.3. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Поява помилок	Складність вимірювання можливих помилок	Проведення додаткових досліджень по вдосконаленню системи
2.	Конкуренція	Поява нових технологій у даній сфері	Аналіз та застосування нових технологій

Таким чином виходячи з даної таблиці розроблена в ході виконання роботи система помірно схильна до ризиків що робить її розробку та використання перспективними.

Але поряд із колом загроз існують і певні можливості (таблиця 5.7).

Таблиця 5.4. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Інноваційна технологія	Відносно невелика кількість конкурентів	Можливість вдосконалення технології новими дослідженнями
2.	Нові шляхи вирішення задач	Автоматизація нових типів задач	Застосування технології новим способом для вирішення існуючих задач

Проаналізувавши можливості які відкриваються із застосуванням даної технології можна зробити наступний висновок: подальші дослідження та розробка на основі технологій застосованих в даній роботі має велике практичне значення.

Таблиця 5.5. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Визначення типу конкуренції - монополія/олігополія/ монополістична/чиста	На досліджуваному ринку конкуренція є чистою, у кожного підприємства є можливість самому обирати, яким з великого різноманіття програмного забезпечення користуватися	Доцільним буде намагатись підлаштувати систему під потреби конкретного користувача (підприємства)
За рівнем конкурентної боротьби – локальний/національний/світовий	В даному випадку конкуренція є світовою, адже конкурентом, як і користувачем, може бути особа з будь-якої країни	Вивчення конкурентів, можливостей, що надається їх системами, та можливостей стати більш затребуваними
За галузевою ознакою конкуренція може бути міжгалузевою та внутрішньогалузевою	Міжгалузева, подібні системи можуть використовуватися у різних галузях за різним призначенням	Конкретизувати функціонал для більш ефективного його використання в різних галузях
На досліджуваному ринку конкуренція за особливостями конкурентних переваг може бути ціновою і неціновою	Можливе існування обох типів конкуренції – цінової та нецінової	Надання особливої уваги типу монетизації та динамічна його зміна як реакція на змінювані умови ринку; доповнення функціоналу як спосіб робити продукт більш затребуваним

Таким чином, як можна зрозуміти з таблиці, що демонструє ступеневий аналіз конкуренції на ринку, у системи, що розробляється, є можливість підлаштуватися під динамічний розвиток ринку задля підвищення конкурентоспроможності продукту шляхом внесення певних змін в функціонал та маркетингове позиціонування системи.

Після аналізу конкуренції проведемо більш детальний аналіз умов конкуренції в галузі.

Таблиця 5.6. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Інші компанії, що розробляють програмне забезпечення	Нові компанії-розробники програмного забезпечення	Відсутні	Підприємства різних галузей	Можна замінити виключно системами з подібним функціоналом
Висновки:	Попри затребуваність, інтенсивність конкуренції не надто висока	Наявні можливості виходу на ринок і нових компаній	-	Потенційні споживачі можуть висувати власні вимоги до функціоналу розроблюваної системи	Необхідний постійний й усебічний моніторинг ринку для своєчасного виявлення конкуруючих систем

Підсумовуючи дані, викладені в попередній таблиці, можна зробити висновок про те, що для забезпечення конкурентоспроможності розроблюваного програмного забезпечення необхідно постійно тримати руку на пульсі відповідного ринку та його новинок.

На основі здійсненого аналізу конкуренції, конкурентного середовища на ринку, з урахуванням усіх загроз та можливостей, що надаються розроблюваним програмним забезпеченням, можна виділити фактори конкурентоспроможності, що впливатимуть на затребуваність та популярність продукту, та здійснити більш детальний їх аналіз, що й буде зроблено в наступній таблиці (табл. 5.10).

Таблиця 5.7. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Наявність патентів	Можливість безперешкодно, легально та з мінімальною кількістю ризиків користуватись програмним забезпеченням
2	Велика кількість розробників	Командна робота над проектом, що дозволяє швидше й ефективніше його допрацьовувати
3	Висока якість	Фактор, котрій буде визначальним при покупці системи.
4	Технічна підтримка	Важливість розкривається через можливе розширення функціоналу, індивідуальну допомогу користувачам, усунення неполадок та допомогу користувачам
5.	Ціна	Дуже важливий фактор при виборі серед представлених подібних систем

За визначеними факторами конкурентоспроможності був проведений порівняльний аналіз сильних та слабких сторін проекту.

Таблиця 5.8. Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з проектом							
			-3	-2	-1	0	+1	+2	+3	
1	Наявність патентів	16	+							
2	Велика кількість розробників	15				+				
3	Висока якість	17	+							
4	Технічна підтримка	19		+						
5.	Ціна	14							+	

З таблиць 5.10 та 5.11 бачимо, що фактори конкурентоспроможності суттєві та мають великий позитивний внесок при впровадженні нового програмного забезпечення для розрахунку його ефективності та конкурентоспроможності. Основною перевагою та головним досягненням є висока якість продукту та технічна підтримка на протязі всього терміну його використання споживачем.

Таблиця 5.9. SWOT-аналіз стартап-проекту

<p>Сильні сторони:</p> <ol style="list-style-type: none"> 1. Висока якість; 2. Постійна підтримка продукту; 3. Зручність користування. 	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> 1. Обмеженість функціоналу; 2. Необхідність розширення можливостей для адаптації в використанні на підприємствах різних галузей.
<p>Можливості:</p> <ol style="list-style-type: none"> 1. Збільшення продаж; 2. Отримання державних замовлень на отримання послуг; 3. Розширення ринку за рахунок іноземних замовників; 4. Зменшення податкового тиску, отримання тендерів на послуги. 	<p>Загрози:</p> <ol style="list-style-type: none"> 1. Цінова конкуренція в зв'язку з появою нових гравців на ринку. 2. Різка зміна курсу гривні може привести до зменшення попиту, особливо з боку малих фірм. 3. Політичні та економічні ризики ведення бізнесу; 4. Втрата потенційних клієнтів через недостатню технічну підтримку; 5. Зменшення продажів через несвоєчасне виконання замовлень.

На основі SWOT-аналізу розробляємо альтернативи ринкової поведінки (конкретизуємо перелік необхідних заходів) для успішного й грамотного виведення продукту на ринок та забезпечення йому високого місця в конкурентному середовищі, а також орієнтовний час ринкової реалізації розроблених заходів з урахуванням можливого виникнення складнощів в ході впровадження проекту.

Таблиця 5.10. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Додавання нового функціоналу	середня	Залежить від функціоналу
2	Виправлення помилок	висока	Важко виміряти
3	Впровадження нових технологій	висока	Від одного місяця
4	Рекламна кампанія	середня	Місяць

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.11. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Промислові компанії	Готові у разі наявності необхідного функціоналу та здорової цінової політики	Високий	Мала	Досить просто
2	Розробники програмного забезпечення	Готові у разі необхідності використання напрацювань для власних проектів	Невисокий	Велика	Необхідний продукт високої якості з найновішими напрацю-

					ваннями та досягненнями
3	Індивідуальні підприємці	Готові у разі можливості максимізації прибутку за умови мінімізації витрат	Невисокий	Мала	Досить просто
Які цільові групи обрано: Промислові компанії.					

За результатами аналізу потенційних груп споживачів ми обрали цільові групи промислових компаній, адже за сукупністю усіх критеріїв вибору дана група споживачів буде найвдалішою для успішного запровадження проекту та забезпечить найбільш раціональне й ефективне охоплення ринку спеціалізованого програмного забезпечення.

Для роботи в обраному сегменті ринку необхідно сформувати базову стратегію розвитку.

Таблиця 5.12. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
	Система пошуку антропометричних точок обличчя для систем розпізнавання особи	Орієнтація на системи ідентифікації особи	Не велика ціна, нові технології розробки, зручність інтегрування	Стратегія спеціалізації

Виходячи з проведеного аналізу стратегій розвитку було визначено альтернативні шляхи розвитку проекту.

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 5.16).

Таблиця 5.13. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
	Частково	Шукати нових споживачів	Система надає унікальні можливості які налаштовуються під певну задачу	Стратегія розширенні первинного попиту

На основі вимог споживачів з обраного сегменту до постачальника і продукту (таблиця 5.5), а також в залежності від стратегії розвитку (таблиця 5.15) та стратегії конкурентної поведінки (таблиця 5.16) розробляємо стратегію позиціонування (таблиця 5.17) яка визначається у формування ринкової позиції, за яким споживачі мають ідентифікувати проект.

Таблиця 5.14. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувавши комплексну позицію власного проекту (три ключових)
	Висока швидкість роботи, простота інтеграції, налаштування під певну задачу	Стратегія спеціалізації	Низька ціна, налаштування під задачу, можливість інтегрування в інші системи	Технічна підтримка, нові технології, універсальність

Результатом даного підрозділу є прийнята система рішень щодо ринкової поведінки компанії, яке потім буде впливати на напрям роботи компанії на ринку.

5.5 Розроблення маркетингової програми стартап-проекту

Під час розроблення маркетингової програми першим кроком є розробка маркетингової концепції товару, який отримає споживач. У таблиці 5.18 підсумовуємо результати аналізу конкурентоспроможності товару.

Таблиця 5.15. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Автоматизація пошуку об'єктів на зображенні	Пошук заданих об'єктів	Пристаювання до особливостей підприємства
	Збільшення продуктивності	Швидкість роботи	Система працює швидше що дає збільшення продуктивності.

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту, його фізичні складові, особливості процесу його надання (табл. 4.19).

Таблиця 5.16. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом			
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Пошук заданого об'єкту на зображенні	Нм	Тх
	2. Використання машинного навчання	Нм	Тл
	3. Знаходження антропометричних точок обличчя	Нм	Тх

	Якість: необхідний функціонал, швидкодія, інтуїтивно зрозумілий інтерфейс
	Пакування: програмне забезпечення записане на компакт диск.
	Марка: назва організації-розробника та назва товару
III. Товар із підкріпленням	До продажу с
	Після продажу
Захищено від копіювання буде за рахунок закритого коду	

Наступним кроком є визначення цінових меж, якими необхідно керуватися при встановленні ціни на потенційний товар, це передбачає аналіз цін товарів конкурентів, та доходів споживачів продукту (табл. 5.20). Аналіз проводиться експертним методом.

Таблиця 5.17. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	8000 грн місяць	1000 грн на місяць	20000 грн на місяць	5000 грн – 7000 грн

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 5.21):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 5.18. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Після покупки товару проводиться завантаження програмного забезпечення та консультація по його використанню	Встановлення програмного забезпечення та сервісна підтримка		

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 5.19. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікації, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Можлива необхідність в навчанні персоналу користування розробленим програмним забезпеченням	Інтернет, телефон, приватні зустрічі		Проінформувати та заохотити клієнта придбати програмне забезпечення	Продемонструвати функціонал системи та показати його вигідність застосування

Так як одною із цільових клієнтів є промислові підприємства, тому існує необхідність провести навчання персоналу користуванню системою для більш точного та коректного пошуку об'єктів тому цю інформацію можна використовувати при проведенні рекламної кампанії.

Проаналізувавши дані можна зробити висновок, що проект має можливість ринкової комерціалізації. Цьому сприяє зростаюча динаміка ринку в сучасному світі, наявний попит на програмне забезпечення для розвитку підприємства та низька собівартість. Також бар'єр входження невеликий і обумовлений він наявністю необхідних знань або наявністю коштів на формування команди розробників. Кількість конкурентів націлених на потенційних клієнтів низька, що сприяє конкурентоспроможності проекту. Для ринкової реалізації проекту доцільно обрати систему пошуку об'єкту на зображенні з відеокамер, із можливістю пошуку антропометричних точок обличчя. Наступна розробка проекту є доцільною..

ВИСНОВКИ ПО РОБОТІ

1. Досліджено існуючі підходи та методи виявлення об'єктів на зображеннях. Обрано для подальших досліджень машинне навчання з використанням каскадних класифікаторів.

2. Проведено порівняння існуючих методів пошуку об'єктів на зображеннях з використанням каскадних класифікаторів. Визначено програмне забезпечення для реалізації каскадних класифікаторів

3. Розроблено програмний прототип для пошуку об'єктів на зображеннях та визначення антропометричних точок обличчя;

4. Проведено обчислювальні експерименти для визначення більш ефективних класифікаторів для задачі пошуку заданого об'єкту. Визначено, що класифікатор `lbpcascade_frontalface_improved` оптимальніший за класифікатор `haarcascade_frontalface_alt` за критеріями частки коректного виокремлення заданого об'єкту та часу розв'язання задачі.

5. Проведено обчислювальні експерименти для визначення більш ефективних класифікаторів для задачі визначення антропометричних точок обличчя. Визначено, що для роботи з детектором антропометричних точок обличчя класифікатор `haarcascade_frontalface_alt` дає кращі результати ніж `lbpcascade_frontalface_improved` завдяки тому, що він навчений знаходити більшу частину обличчя та голови.

6. Проведено обчислювальні експерименти для визначення найбільш ефективних наборів параметрів для роботи каскадних класифікаторів. Визначено оптимальні значення для наступного набору параметрів: `Scale` – масштаб вікна пошуку об'єкту, `Min neighbors` – кількість областей, які перетинаються при пошуку об'єкту, `Min size` – мінімальний розмір вікна пошуку об'єкту, `Max size` – максимальний розмір вікна пошуку об'єкту.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. S. Nagabhushana, "Introduction," in *Computer Vision and Image Processing*, New Age International (P) Ltd., Publishers, 2005, p. 3.
2. Papageorgiou, Oren, Poggio, "A general framework for object detection", *International Conference on Computer Vision*, 1998.
3. Viola and Jones, "Rapid object detection using a boosted cascade of simple features", *Computer Vision and Pattern Recognition*, 2001.
4. U.S. Patent 6,711,293, "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image", David Lowe's patent for the SIFT algorithm, March 23, 2004.
5. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005.
6. Bengio, Y. Courville, A. Vincent "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (2013) .
7. LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey (28 May 2015). "Deep learning".
8. J.R.R. Uijlings Selective Search for Object Recognition / J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers and A.W.M. Smeulders // Technical Report 2012, submitted to IJCV.
9. R. Girshick, J. Donahue, T. Darrell, J. Malik / Rich feature hierarchies for accurate object detection and semantic segmentation // UC Berkeley.
10. R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015
11. S. Ren, K. He, R. Girshick, J. Sun / Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

12. W. Liu, D. Anguelov, D. Erhan «SSD: Single Shot MultiBox Detector»
University of Michigan, Ann-Arbor
13. J. Redmon, S. Divvala, R. Girshick, A. Farhadi You Only Look Once: Unified,
Real-Time Object Detection
14. R. Padilla Evaluation of Haar Cascade Classifiers Designed for Face Detection
/ R. Padilla, C. F. F. Costa Filho and M. G. F. Costa // International Journal of
Computer, Electrical, Automation, Control and Information Engineering Vol:6, No:4,
2012, pp.466-469.
15. OpenCV: Open Source Computer Vision Library — Режим доступа:
<https://github.com/opencv/opencv/blob/master/data/>
16. Jo Chang-yeon Face Detection using LBP features / Jo Chang-yeon //CS 229
Final Project Report, December 12, 2008.
17. P. Viola, M. Jones Rapid Object Detection using a Boosted Cascade of Simple
Features ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN
RECOGNITION 2001
18. X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression.
In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.
IEEE, 2012
19. EmguCV: a cross platform .Net wrapper to the OpenCV image processing
library — Режим доступа: http://www.emgu.com/wiki/index.php/Main_Page
20. M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured
output regression. In Computer Vision– ECCV 2008, pages 2–15. Springer, 2008.
21. B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Simultaneous detection
and segmentation. In Computer Vision– ECCV 2014, pages 297–312. Springer, 2014.
22. R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object
detection. In Image Processing. 2002. Proceedings. 2002 International Conference on,
volume 1, pages I–900. IEEE, 2002.

23. D. G. Lowe. Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150–1157. Ieee, 1999.
24. P. Viola and M. J. Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154, 2004
25. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)
26. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: ECCV 2012. (2012)
27. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014) 8. Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. (2015)
28. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
29. Рихтер Д. Программирование на платформе Microsoft .NET Framework / Д. Рихтер., 2003.
30. Нейгел К. С# 4.0 и платформа .NET 4 / К.Нейгел, Б. Ивсен, Д. Глинн, Д. Уотсон., 2011.
31. Валерий Ф. Создание приложение с помощью С# / Ф. Валерий., 2008.
32. Колот С.С. Розпізнавання об'єктів з навмисним маскуванням / С.С. Колот, С.І. Шаповалова // Сучасні проблеми наукового забезпечення енергетики: матеріали XVI Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів, Київ, 24-27 квітня 2018 р. у 2 т. - К: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2018.- Т.2.- С.191.
33. Колот С.С. Системи моделювання Convolutional Neural Networks / С.С. Колот, С.І. Шаповалова // Сучасні аспекти розробки програмного забезпечення: Збірник наукових праць V науково-практичної дистанційної конференції молодих вчених і фахівців з розробки програмного забезпечення, 15 травня 2018. – Черкаси: видавець Чабаненко Ю.А., 2018. – С. 209-213

ДОДАТОК А

Виявлення об'єкту на зображенні з відеокамер

Апробація

УКР.НТУУ"КПІ".ТВ3262_18М

Аркушів 7

РОЗПІЗНАВАННЯ ОБ'ЄКТІВ З НАВМИСНИМ МАСКУВАННЯМ

Колот С.С., магістрант

Шаповалова С.І., к.т.н., доцент

НТУУ «Київський політехнічний інститут імені Ігоря Сікорського»

Останні досягнення в розвитку нейронних мереж, такі як "глибоке навчання", значно покращили ефективність найсучасніших систем візуального розпізнавання, але існує проблема розпізнавання об'єктів з навмисним маскуванням.

Найбільш відомими з прикладних систем розпізнавання облич є: FindFace, Pictriv, Betaface, пошук по картинках Google. Як правило, для розпізнавання особи використовують конволюційні нейронні мережі (CNN - Convolutional Neural Networks), але більшість таких систем легко ввести в оману, шляхом додавання аксесуарів, таких як окуляри, накладні вуса і т.п. В останній час з'явилися спеціальні дослідження з моделювання об'єктів, які CNN розпізнають невірно. Наприклад, авторам дослідження [1] вдалося ввести в оману класифікатор Google, який розпізнав черепаху як рушницю. Тому необхідно вдосконалити CNN для розпізнавання замаскованих об'єктів.

Архітектура конволюційних нейронних мереж CNN [2] відрізняється від традиційних багат шарових перцептронів (MLP - Multilayer Perceptron) [3], насамперед забезпеченням певного ступеню інваріантності зрушень та спотворень [4]. CNN об'єднують три архітектурні ідеї: поля локального сприйняття (Local Receptive Fields), загальні ваги, просторова та часова субспецифікація [5].

CNN порівнюють зображення частину за частиною. Частини, які мережа шукає, називаються рисами. Знайшовши приблизні риси, що збігаються у відповідних позиціях у двох зображеннях, CNN починає набагато краще знаходити подібність, ніж при схемах збігу цілого зображення.

В роботі:

- проведено огляд існуючих реалізацій алгоритмів CNN: Caffe [6], CNTK [7], Deeplearning4j [8], TensorFlow [9];
- проведено огляд прикладних систем на основі CNN: FindFace, Pictriv, Betaface;
- визначено тестові виборки та навчальні приклади для них;
- визначено напрям подальшого дослідження вдосконалення CNN.

Перелік посилань:

1. A. Athalye, E. Engstrom, SYNTHESIZING ROBUST ADVERSARIAL EXAMPLES, ICLR 2018.
2. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541-551, Winter 1989.
3. Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961
4. Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." The Journal of physiology 160.1 (1962): 106-154.
5. B. Ashwin, B. Maithili, G. Pranav, Applications of Convolutional Neural Networks, International Journal of Computer Science and Information Technologies, Vol. 7 (5), 2016, 2206-2215
6. Caffe – Access mode <http://caffe.berkeleyvision.org/>
7. CNTK – Access mode <https://github.com/Microsoft/CNTK>
8. Deeplearning4j – Access mode <https://deeplearning4j.org/>
9. TensorFlow – Access mode <http://www.tensorflow.org/>

Системи моделювання

Convolutional Neural Networks

Колот С.С., магістрант

Шаповалова С.І., к.т.н., доцент

НТУУ «Київський політехнічний інститут імені Ігоря Сікорського»

Задача розпізнавання об'єктів, зокрема облич, є затребуваною в прикладних програмних системах. Для її розв'язання запропоновані спеціальні нейронні мережі Convolutional Neural Networks (CNN) [1]. Поширенню та вдосконаленню CNN сприяла велика кількість даних для їх навчання, наприклад, фотографій із соцмереж. Успішне використання цих нейронних мереж сприяло тому, що в останній час з'явився новий напрямок досліджень з їх обману шляхом маскуванню об'єктів. Тому задача створення середовища моделювання перешкод розпізнавання об'єктів є актуальною та має практичне застосування.

На сьогоднішній день розроблено велику кількість середовищ моделювання та навчання нейронних мереж, в яких реалізовані усі основні концепції, зокрема CNN. Найбільш відомими середовищами моделювання нейронних мереж є: Tensorflow [2], Caffe [3], CNTK [4], MXNET [5]. Кожна з цих систем має особливості з точки зору вимог до комп'ютерної техніки.

Для моделювання маскуванню об'єктів необхідно визначити найбільш відповідну систему моделювання CNN.

Мета статті – вибір системи моделювання Convolutional Neural Networks для тренування розпізнавання замаскованих об'єктів.

Система моделювання Tensorflow - це система машинного навчання, яка може працювати як на локальному комп'ютері або в хмарі, так і на смартфоні або навіть на мікрокомп'ютерах, таких як Raspberry Pi. Система Tensorflow розроблена командою Google Brain. Метою розробників є створення гнучких моделей нейронних мереж, які здатні самостійно отримувати ознаки, а також ефективно використовувати дані та обчислювальні потужності.

Середовище для глибокого навчання Caffe, розроблено Yangqing Jia в університеті Берклі. Середовище Caffe підтримує багато типів машинного навчання, націлених в першу чергу на вирішення задач класифікації та сегментації зображень. Caffe забезпечує моделювання Convolutional Neural Networks, Recurrent Convolutional Neural Networks, довгу та короткострокову пам'ять. Для прискорення навчання застосовується система графічних процесорів (GPU), підтримувана архітектурою CUDA, використовується бібліотека CuDNN від компанії Nvidia. До переваг Caffe можна віднести: велику кількість туторіалів, легкість освоєння, легкість розгортання.

Microsoft Cognitive Toolkit (CNTK) - безкоштовний загальнодоступний набір інструментів з відкритим кодом для моделювання нейронних мереж, розроблений компанією Microsoft. Це третій за популярністю спеціалізований пакет для глибокого навчання після TensorFlow і Caffe.

Apache MXNet - це середовище глибокого навчання, є портативним та легким, підтримує обчислення на декількох GPU. Бібліотека MXNet надає не тільки середовище для глибокого навчання, але також і колекцію прикладів і інструкцій для побудови глибоких навчальних систем.

Дослідники компанії Microsoft для порівняння систем моделювання нейронних мереж відокремили такі характеристики: швидкість, точність, структура API, масштабованість, скоринг, можливість розширення, вбудовані модулі зчитування. В статті [6] доведено, що CNTK в порівнянні з TensorFlow показує кращі результати за більшістю представлених характеристик.

Дослідник компанії Microsoft А. Trivedi в роботі [7] представив формальне порівняння систем навчання нейронних мереж: MXNET, TensorFlow, Theano.

На основі цього представлення в даному дослідженні було порівняно системи моделювання нейронних мереж: Tensorflow, Caffe, CNTK, MXNET (Таблиця 1).

Таблиця 1 Порівняння систем моделювання CNN

Характеристика	TensorFlow	Caffe	CNTK	MXNET
Автор	Команда Google Brain	Yangqing Jia	Microsoft	Distributed (Deep) Machine Learning Community
Ліцензія на програмне забезпечення	Apache 2.0	BSD	MIT	Apache 2.0
Відкритий вихідний код	Так	Так	Так	Так
Платформа	Windows, Linux, Mac OS X	Windows, Linux, Mac OS X	Windows, Linux	Ubuntu, OS X, Windows, AWS, Android, iOS
Мова програмування	C++, Python	Matlab, Python C++	C#, C++, Python, Java	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl
Підтримка CUDA	Так	Так	Так	Так
Підтримка паралельних і розподілених обчислень	Так	Так	Так	Так
Наявність попередньо навчених моделей	Ні	Так	Ні	Так
CNN	Так	Так	Так	Так

В дослідженні [6] перевагу віддано CNTK в порівнянні з TensorFlow однак CNTK має гіршу документованість ніж TensorFlow. Дослідження даного проекту показали, що з точки зору швидкості розв'язання задачі MNIST [8] система моделювання нейронних мереж Tensorflow має кращі показники ніж Caffe. Завдяки цьому для дослідницьких цілей було обрано TensorFlow.

Список літератури

1. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541-551, Winter 1989.
2. Caffe – Access mode <http://caffe.berkeleyvision.org/>
3. CNTK – Access mode <https://github.com/Microsoft/CNTK>
4. Deeplearning4j – Access mode <https://deeplearning4j.org/>
5. TensorFlow – Access mode <http://www.tensorflow.org/>
6. Reasons to switch from Tensorflow to CNTK. – Access mode <https://docs.microsoft.com/en-us/cognitive-toolkit/reasons-to-switch-from-tensorflow-to-cntk>
7. A. Trivedi. Deep Learning Part 1: Comparison of Symbolic Deep Learning Frameworks. – Access mode <http://blog.revolutionanalytics.com/2016/08/deep-learning-part-1.html>
8. THE MNIST DATABASE of handwritten digits. – Access mode <http://yann.lecun.com/exdb/mnist/>

ДОДАТОК Б

Виявлення об'єкту на зображенні з відеокамер

Акт впровадження

УКР.НТУУ"КПІ".ТВ3262_18М

Аркушів 2