

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА
ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Мобільний додаток розпізнавання автомобільних номерів

Освітній ступінь “бакалавр”
Спеціальність 123 – “комп’ютерна інженерія”

Керівник проекту:

(підпис)

Рязанцев О.І.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Капран О.Ю.

(ініціали, прізвище)

Група:

КІ-15з

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітній ступінь бакалавр
Напрямок підготовки _____
(шифр і назва)
Спеціальність 123 – комп'ютерна інженерія
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри КНІ
І.С. Скарга-Бандурова
« _____ » _____ 20 ____ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Капран Олені Юріївні
(прізвище, ім'я, по батькові)

1. Тема роботи Мобільний додаток розпізнавання автомобільних номерів

керівник проекту (роботи) Рязанцев Олександр Іванович, д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "13" 05 2019 р. № 84/15.15

2. Термін подання студентом роботи 16.06.2019

3. Вихідні дані до роботи Тестові файли для завантаження: 2 файли типу jpeg,
2 файли типу png, 2 файли типу bmp

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Розпізнавання образів та його застосування, система контролю і розпізнавання автомобільних номерів, реалізація та аналіз результатів, охорона праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання 30.04.2019

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз завдання та робота з літературою	05.05.2019 - 13.05.2019	
2	Аналіз технічних засобів	14.05.2019 - 22.05.2019	
3	Розробка алгоритму	22.05.2019 - 02.06.2019	
4	Програмна реалізація	02.06 .2019- 11.06.2019	
5	Оформлення пояснювальної записки та електронних плакатів	11.06.2019 - 16.06.2019	

Здобувач вищої освіти

_____ (підпис)

Капран О.Ю.

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

Рязанцев О.І.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 82 с., 31 рис., 17 джерел.

Робота присвячена вирішенню проблеми розпізнавання зображень та ідентифікації автомобільних номерних знаків для подальшого їх застосування у різних сферах.

У результаті атестаційної роботи здійснено огляд, розбір та аналіз існуючих методів розпізнавання тексту на зображеннях. Розроблено мобільний додаток для тестування однієї з технологій детектування номерних знаків. Був проведений аналіз та обґрунтування отриманих результатів, а також був зроблений висновок щодо доцільності використання даної системи.

На даний момент існують різні методи розпізнавання і системи розпізнавання номерних знаків сьогодні використовуються у різних програмах дорожньої безпеки, таких як паркування, фіксування номерів при перевищенні швидкості, доступ до закритих територій, а також прикордонний контроль або відстеження викрадених автомобілів. У деяких країнах системи розпізнавання, встановлені на кордонах країни, автоматично виявляють і контролюють пункти перетину кордону. Кожен транспортний засіб може бути зареєстрований у центральній базі даних та порівнювати з чорним списком викрадених транспортних засобів.

Ключові слова: РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, НОРМАЛІЗАЦІЯ ЗОБРАЖЕНЬ, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ, МЕТОДИ РОЗПІЗНАВАННЯ, XAMARIN.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєверодонецьк, 93400с.

ЗМІСТ

ВСТУП.....	7
1 РОЗПІЗНАВАННЯ ОБРАЗІВ ТА ЙОГО ЗАСТОСУВАННЯ.....	8
1.1 Теорія розпізнавання образів.....	8
1.1.1 Теоретичні відомості.....	8
1.1.2 Підходи до розпізнавання образів.....	11
1.1.3 Перцептрон як метод розпізнавання образів.....	12
1.2 Сфери застосування.....	13
1.3 Постановка задачі.....	14
2 СИСТЕМА КОНТРОЛЮ І РОЗПІЗНАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ	16
2.1 Система розпізнавання номерів та її характеристики.....	16
2.2 Алгоритми і технології розпізнавання номерних знаків.....	17
2.3 ПЗ для системи розпізнавання.....	19
2.4 Технологія оптичного розпізнавання символів.....	21
2.5 Типові рішення.....	27
2.6 Огляд існуючих бібліотек для розпізнавання номерів авто.....	29
2.6.1 Проект Opos.....	29
2.6.2 Проект JavaANPR.....	31
2.6.3 Проект Automatic License Plate Recognition.....	32
2.6.4 Інші проекти.....	33
2.7 Процедура розпізнавання.....	33
2.7.1 Система розпізнавання.....	34
2.7.2 Найпоширеніші методи розпізнавання.....	35
3 РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	37
3.1 Обґрунтування вибору середовища програмної реалізації.....	37
3.1.1 Особливості Xamarin Framework.....	37
3.1.2 Середовище розробки.....	43
3.1.3 Оптичне розпізнавання символів в .NET.....	44
3.2 Програмна реалізація.....	45

	5
3.2.1 Алгоритм реалізації	45
3.2.1 Ідентифікація регіону номерного знаку	51
3.3 Інструкція користувача	52
3.4 Тестування розробленої моделі	53
3.4.1 Тестування на локальному ПК.....	53
3.4.2 Мобільне тестування	59
4 ОХОРОНА ПРАЦІ	61
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектованого об'єкту, що мають вплив на персонал	61
4.2 Заходи щодо техніки безпеки	62
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці	65
4.4 Рекомендації по пожежній безпеці.....	69
ВИСНОВКИ	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	76
ДОДАТОК А. Електронні плакати.....	79

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПК – персональний комп'ютер

UI – User interface – інтерфейс користувача

VS – Microsoft Visual Studio

ОС – операційна система

ПЗ – програмне забезпечення

LPR – License Plate Recognition - система автоматичного розпізнавання номерів

ANPR – Automatic number plate recognition system – система автоматичного розпізнавання цифрових таблиць

OCR – Optical Character Recognition – оптичне розпізнавання символів

ВСТУП

У наш час комп'ютеризація розвивається швидкими темпами і відіграє значну роль в суспільстві. Комп'ютерні технології допомагають автоматизувати широке коло процесів, які в недалекому минулому покладалися на людину. Інформаційні технології використовуються в промисловості, в транспорті, в побуті та ін.

Сьогодні важливим аспектом безпеки дорожнього руху та контролю є ідентифікація автомобілів по їх реєстраційному номерному знаку. Системи розпізнавання номерних знаків мають різні сфери застосування, такі як автотранспортні підприємства, контроль в'їзду на територію підприємства і переміщення транспортних засобів на об'єктах з обмеженим доступом, заправні станції, контроль швидкості руху, автомобільні стоянки.

Актуальність завдання розпізнавання номерних знаків транспортних засобів полягає в тому, що з кожним днем збільшується потреба автоматизації контролю в'їзду на територію підприємств, контролю швидкості руху і визначення порушення правил дорожнього руху.

В даний час існує досить велика кількість систем визначення номерних знаків, але не всі з них є якісною і надійною продукцією. Системи з високою швидкістю і точністю розпізнавання є комерційними, засекреченими і дорогими, що не дозволяє здійснити їх масове впровадження.

Важливим є визначення найбільш ефективних методів обробки зображень і розпізнавання автомобільних номерних знаків, а також створення нових комбінованих або модифікованих методів для якісного розпізнавання системи в реальному часі.

Мета роботи дати користувачеві можливість розпізнавати зображення у мобільному додатку, та отримувати інформацію про регіон автомобілю без додаткового пошуку. Додаток розроблений для операційної системи Android за допомогою технології Xamarin. Користувач зможе завантажувати зображення, отримувати автомобільний номер та додаткову інформацію.

1 РОЗПІЗНАВАННЯ ОБРАЗІВ ТА ЙОГО ЗАСТОСУВАННЯ

1.1 Теорія розпізнавання образів

1.1.1 Теоретичні відомості

Equation Section (Next) (не видаляти зробити білим) (рядок)

Теорія розпізнавання образів — розділ кібернетики, що розвиває теоретичні основи й методи класифікації і ідентифікації предметів, явищ, процесів, сигналів, ситуацій і т. п. об'єктів, які характеризуються скінченним набором деяких властивостей і ознак. Такі задачі вирішуються досить часто, наприклад, при переході або проїзді вулиці за сигналами світлофора. Розпізнавання кольору лампи світлофора, що засвітилася, і знання правил дорожнього руху дозволяє прийняти правильне рішення про те, можна, чи не можна переходити вулицю в цей момент.

У більшості випадків, сприймаючи явища навколишнього світу, людина здійснює їх класифікацію, тобто розбиває ці явища (предмети, ситуації) на групи схожих явищ (саме схожих, а не тотожних). По тим чи іншим причинам необхідно віднести в одну групу у чомусь "подібні" явища чи предмети, які можуть при цьому значно відрізнятися один від одного. Наприклад, усі фігури, зображені на рисунку 1.1, ми називаємо "літерою А", незважаючи на їх значну відмінність у їх написанні.



Рисунок 1.1 – Літера «А» у різних варіаціях

Суттєвим є той факт, що виділивши такі групи (множини) об'єктів, ми отримуємо здатність "упізнавати", тобто встановлювати належність до вже відомої множини, нові об'єкти, які ще не зустрічалися нам раніше, наприклад впізнавати букви, написані новим для нас почерком.

Отримавши уявлення про те, що являє собою буква "А" на основі деякої, зазвичай невеликої кількості екземплярів цієї букви, ми спроможні упізнати як завгодно велику кількість інших її екземплярів.

Проте, далеко не всі множини об'єктів дають змогу на основі невеликої частини множини упізнати як завгодно багато інших невідомих нам її представників. Наприклад, фотографії студентів деякого вузу утворюють множину. Проте неможливо після ознайомлення зі скажімо десятьма фотографіями студентів визначити по новій фотографії є людина студентом цього вузу чи ні.

Таким чином існують множини деякого особливого типу. Ці множини мають характерну властивість, яка виявляється у тому, що після ознайомлення із скінченною частиною об'єктів цих множин, можна упізнавати як завгодно велику кількість інших їх представників. Множини такого типу будемо називати образами.

Прикладами образів можуть бути такі множини: чоловіки, дитячі портрети, ссавці, картини Пікассо, цифри 5, зображення винищувачів МІГ. Застосовувати до них термін "образ" ми можемо тому, що ознайомлення з образом не пов'язано із запам'ятовуванням окремих об'єктів, а упізнавання нового об'єкта відбувається без безпосереднього порівняння із кожним раніше відомим.

Характерна властивість образів об'єктивна у тому сенсі, що різні люди (живі істоти), які навчалися на різних групах об'єктів образа, у переважній більшості однаково і незалежно один від одного класифікують одні і ті самі нові об'єкти. Саме об'єктивність цієї властивості образів дає змогу людям, які вчилися у різних школах, успішно розпізнавати раніше невідомий їм почерк.

Проте, самі образи у той самий час є у певній мірі "розпливчастими" і питання про належність об'єкта до даного образу не завжди може бути вирішене однозначно. На рисунку 1.2 показано, як за допомогою невеликих послідовних змін можна перетворити цифру 5 у цифру 3. Рисунок 1.2

ілюструє важливу особливість об'єктів кожного образу. Можна в деяких межах змінювати об'єкт, але він усе одно залишиться об'єктом того самого образу.

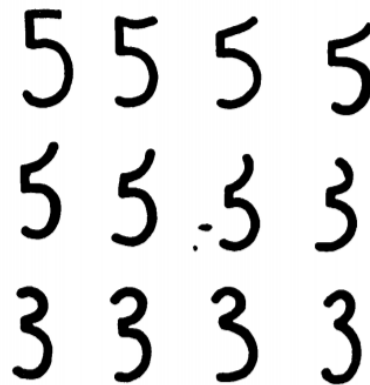


Рисунок 1.2 – Перетворення цифри 5 у 3

Створення штучних систем розпізнавання образів залишається складною теоретичною й технічною проблемою. Необхідність у такому розпізнаванні виникає в самих різних областях — від військової справи й систем безпеки до оцифрування різних аналогових сигналів.

Сприйняття явищ у формі образів відіграє надзвичайно важливу роль у процесах пізнання зовнішнього світу. У процесі біологічної еволюції багато тварин за допомогою зорового й слухового апарата вирішили задачу розпізнавання образів досить добре. Як впливає із самого означення образа "розпізнаванню" нових для нас об'єктів передуює процес навчання. Під час навчання істоти ознайомлюються із деякою кількістю об'єктів і, крім цього, із якогось джерела (наприклад від батьків, старших і т.д.) отримують інформацію про те, до якого образу відноситься кожний із цих об'єктів. Цей процес отримав назву "навчання з учителем".

Більш загальний характер має "навчання без учителя", у процесі якого система вчиться спонтанно виконувати поставлене завдання без втручання з боку "вчителя". Навчання машин без вчителя формулюється як задача кластерного аналізу. Вибірка об'єктів розбивається на кластери (множини, що мають порожній перетин), таким чином, що кожний кластер складається

із "схожих" об'єктів, а різні кластери "суттєво" відрізняються один від одного. Кластеризація часто використовується в якості допоміжного засобу розв'язування задач класифікації та регресійного аналізу. Деякі алгоритми розв'язування задач класифікації комбінують навчання з учителем та навчання без учителя (наприклад навчання мереж векторного квантування).

Традиційно задачі розпізнавання образів включають у коло задач штучного інтелекту. Можна виділити два основних напрямки:

- вивчення здібностей до розпізнавання, якими володіють живі істоти, їхнє пояснення й моделювання;
- розвиток теорії й методів побудови пристроїв, призначених для розв'язання окремих задач у прикладних цілях.

1.1.2 Підходи до розпізнавання образів

Для оптичного розпізнавання образів можна застосувати метод перебору вигляду об'єкта під різними кутами, масштабами, зсувами й т. д. Для букв потрібно перебирати шрифт, властивості шрифту й т. д.

Другий підхід — знайти контур об'єкта й досліджувати його властивості (зв'язність, наявність кутів і т. д.). Існує безліч алгоритмів для виділення контурів, такі як: оператор Собеля, оператор Лапласа, оператор Робертса, оператор Прюїтт і оператор Кенні. Детектування контурів необхідно в тому випадку, коли є досить складне зображення і використовуючи малі інструменти бібліотеки комп'ютерного зору необхідно виділити об'єкти на цьому зображенні для подальшої роботи з ними.

Ще один підхід — використовувати штучні нейронні мережі (багатошарові перцептрони, мережі квантування, мапи Кохонена, рекурентні мережі). Цей метод вимагає або великої кількості прикладів задачі розпізнавання (із правильними відповідями), або спеціальної структури нейронної мережі, що враховує специфіку даної задачі.

1.1.3 Перцептрон як метод розпізнавання образів

Ф. Розенблатт уводячи поняття про модель мозку, завдання якої полягає в тому, щоб показати, як у деякій фізичній системі, структура й функціональні властивості якої відомі, можуть виникати психологічні явища та описав найпростіші експерименти з розрізнення. Дані експерименти цілком стосуються до методів розпізнавання образів, але відрізняються тим, що алгоритм розв'язання не детермінований.

Найпростіший експеримент, на основі якого можна одержати психологічно значиму інформацію про деяку систему, зводиться до того, що моделі пред'являються два різних стимули й потрібно, щоб вона реагувала на них різним чином. Метою такого експерименту може бути дослідження можливості спонтанного розрізнення стимулів системою при відсутності втручання з боку експериментатора, або, з навпаки, вивчення примусового розрізнення, при якому експериментатор прагне навчити систему здійснювати необхідну класифікацію. У досвіді з навчанням перцептрону зазвичай пред'являється деяка послідовність образів, у яку входять представники кожного із класів, що підлягають розрізненню. Відповідно до деякого правила модифікації пам'яті правильний вибір реакції підкріплюється. Потім перцептрону пред'являється контрольний стимул і визначається ймовірність одержання правильної реакції для стимулів даного класу. Залежно від того, збігається чи не збігається обраний контрольний стимул з одним з образів, які використовувалися в навчальній послідовності, отримують різні результати:

- якщо контрольний стимул не збігається з жодним із навчальних стимулів, то експеримент пов'язаний не тільки з чистим розрізненням, але містить у собі й елементи узагальнення;

- якщо контрольний стимул збуджує деякий набір сенсорних елементів, цілком відмінних від тих елементів, які активізувалися при впливі

раніше пред'явлених стимулів того ж класу, то експеримент є дослідженням чистого узагальнення.

Перцептрони не мають здатності до чистого узагальнення, але вони цілком задовільно функціонують в експериментах із розрізнення, особливо якщо контрольний стимул досить близько збігається з одним з образів, щодо яких перцептрон уже має певний досвід.

1.2 Сфери застосування

)

Приклади задач розпізнавання образів:

- розпізнавання літер;
- розпізнавання штрих-кодів;
- розпізнавання автомобільних номерів;
- розпізнавання осіб;
- розпізнавання мови;
- розпізнавання зображень;
- розпізнавання локальних ділянок земної кори, у яких знаходяться

родовища корисних копалин.

Розпізнавання образів застосовується в наступних областях:

- біоінформатика: пошук шаблонів в ДНК;
- бази даних: пошук і класифікація;
- обробка текстів: тематична класифікація;
- аналіз зображень: розпізнавання символів, робота з картами, розпізнавання осіб, поділ об'єктів;
- виробництво: контроль якості (візуальна перевірка коректності мікросхем);
- пошук по мультимедіа: визначення жанрів;
- біометрія: ідентифікація людини за відбитками пальців, по райдужній оболонці ока;

- прогнозування: погода, сейсмологія, геологія;
- обробка мови: переклад аудіо-сигналів в текст;

1.3 Постановка задачі

Основною метою атестаційної роботи являється аналіз існуючих алгоритмів та методів розпізнання символів на зображеннях у реальному масштабі часу та розробка мобільного застосунку для реалізації одного з алгоритмів детектування номерних знаків для їх подальшої ідентифікації та використання у різних напрямках. В процесі виконання даної роботи необхідним є аналіз методів розпізнання та бібліотек, що їх реалізують.

У даній роботі будуть детально розглядатися процес розробки програмних продуктів, зокрема мобільних, створених з використанням бібліотек для обробки зображень. Для досягнення мети необхідно вирішити такі завдання:

- вибрати найбільш актуальну бібліотеку для розпізнавання номера на зображенні;
- проаналізувати та вибрати середу розробки мобільного додатку;
- розробити та реалізувати алгоритм для вирішення поставленої задачі у реальному часі;
- дослідити якість програмної реалізації, визначити оптимальні параметри його функціонування;
- реалізувати головні задачі. А саме: завантаження зображення, розпізнання автомобільного номеру, пошук додаткової інформації по номеру;
- розробити мобільний додаток, що у подальшому може інтегруватися на різні типи та версії ОС мобільних пристроїв;
- розробити зручний для користування інтерфейс додатку;
- протестувати програму на різних версіях мобільних пристроїв.

Визначимо основні завдання, які необхідно вирішити в атестаційній роботі:

- аналіз існуючих методів розпізнання;
- вибір актуального середовища для мобільної розробки;
- детальний розбір бібліотек для обробки зображень;
- виявлення переваг та недоліків кожної з бібліотек;
- реалізація .

Таким чином необхідно розробити програмний продукт для розпізнавання номерних знаків на зображенні у реальному часі, а також провести аналіз результатів роботи та швидкодії мобільного застосунку і зробити висновки щодо доцільності даної системи.

2 СИСТЕМА КОНТРОЛЮ І РОЗПІЗНАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ

2.1 Система розпізнання номерів та її характеристики

Equation Section (Next) (рядок)

В даний час в світі на дорогах знаходиться понад півмільярда автомобілів. Всі ці транспортні засоби мають унікальний ідентифікаційний номер в якості основного розпізнавального знаку. Ідентифікаційний номер транспортного засобу фактично є реєстраційним номером, що дає законне право на участь у дорожньому русі.

Проблема ідентифікації автомобіля за реєстраційним номерним знаком є важливим аспектом контролю і забезпечення безпеки дорожнього руху. Продукти, здатні вирішувати це завдання, затребувані у самих різних сферах. Прикладом можуть служити автотранспортні підприємства, автомобільні стоянки, гаражні кооперативи, котеджні селища, заправні станції, пункти контролю в'їзду на територію об'єкта і т.п.

Коли йдеться про систему автоматичного розпізнання номерів (License Plate Recognition, LPR), мається на увазі програмний або апаратно-програмний комплекс, який реалізує алгоритми автоматичного розпізнавання номерних знаків для реєстрації подій, пов'язаних з переміщенням автомобілів, тобто для автоматизації введення даних і їх подальшої обробки.

Строго кажучи, LPR-система - це пристрій, який реєструє проїзд транспортного засобу, зчитує його реєстраційний номер і виводить його в ASCII-систему обробки даних.

В даний час існує досить багато LPR-систем з різним рівнем якості розпізнавання, швидкодії і спектром послуг додаткових функцій. Продукти, що володіють високою швидкістю і точністю розпізнавання, як правило, дуже дорогі. Їх висока вартість не дозволяє здійснити масове впровадження. Розглянемо загальні принципи, покладені в основу розпізнавання автомобільних номерів, щоб зрозуміти причини високої вартості таких систем.

2.2 Алгоритми і технології розпізнавання номерних знаків

Безперечно, основою будь-якої LPR-системи є використовувані алгоритми розпізнавання. Кваліфікація розробників в області сучасної вищої математики, обробки зображень, програмуванні і технологіях оптимізації програм, а також наявність істотного досвіду роботи - всі ці фактори визначають характеристики LPR-системи, такі як:

- ймовірність розпізнавання;
- швидкість обробки;
- здатність розпізнавати різні типи номерних знаків;
- здатність працювати з зображеннями різної якості.

Розпізнавання державних реєстраційних знаків є нетривіальним завданням з області технічного зору та штучного інтелекту. Використовувані алгоритми локалізації номерного знаку і його розпізнавання, як правило, є комерційною таємницею і, природно, не публікуються. Лише деякі компанії називають їх типи і публікують послідовність дій.

Нижче послідовно наведені ключові етапи розпізнавання автомобільного номера:

- приведення вихідного зображення до виду, який не залежить від умов реєстрації зображення (ступінь освітленості, нерівномірність розподілу яскравості від джерел світла, розмитість, зашумленість і т.п.);
- виділення на отриманому зображенні областей-кандидатів, потенційно містять пластину з номером;
- проведення детального аналізу областей-кандидатів на основі формального уявлення масштабних характеристик номерний пластини і скорочення простору для подальшого пошуку;
- приведення до стандартного розміру графічного зображення номерний пластини з корекцією якості зображення;
- попереднє визначення типу номерний пластини (в прив'язці до діючих стандартів);

– витяг окремих символів і їх розпізнавання (аналіз символів за ключовими характеристиками, незалежним від масштабу, використовуваного шрифту, геометричних спотворень і розривів);

– уточнення результатів розпізнавання на основі інформації про тип номера і за результатами з попередніх кадрів.

Результатом роботи алгоритму є інформація про проїзд транспортного засобу, що містить рядок з розпізнаним номером, стоп-кадр з найкращим зображенням транспортного засобу, інформацією про час проїзду автомобіля і т.д.

З представленої послідовності кроків видно, що вихідні дані для розпізнавання номера не обмежуються тільки візуальним зображенням. У світі існує велика кількість видів номерних знаків, що розрізняються:

- використовуваними шрифтами (знаки з символами різного розміру, латинські, кириличні та інші шрифти);
- кольором фону і символів (чорні символи на світлому тлі або білі символи на темному тлі);
- кількістю рядків у номері (однорядкові, дво- або три- рядкові);
- наявністю або відсутністю коду позначення регіону або спеціальної позначки і т.п.

Існуючі типи автомобільних знаків України зображено на рисунку 2.1.

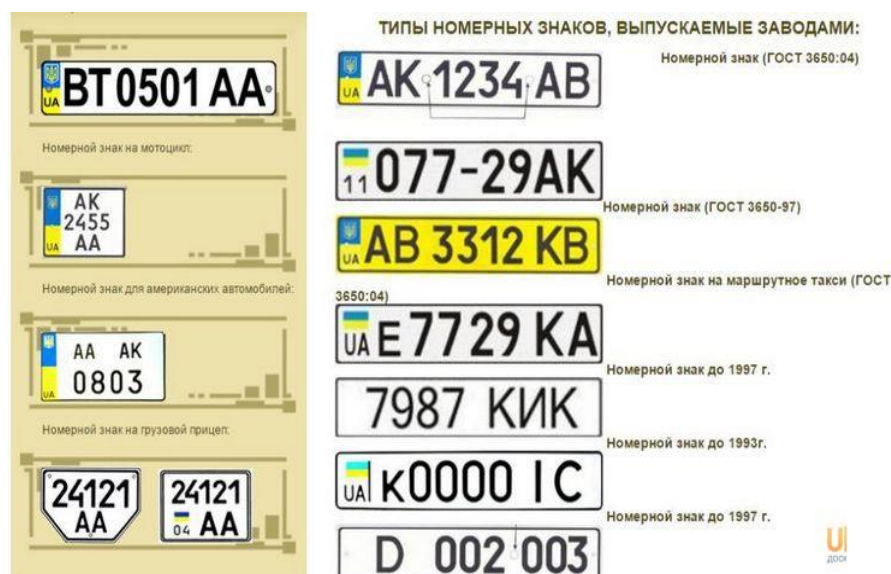


Рисунок 2.1 – Різні види автомобільних реєстраційних знаків

Облік цих відмінностей в номерних знаках дає суттєву перевагу тим розробникам LPR-систем, які використовують цю додаткову інформацію у логіці своїх алгоритмів розпізнавання. Інформація про структуру знаку та його синтаксисі дозволяє істотно підвищити ймовірність правильного розпізнавання, одночасно знижуючи вимоги до якості алгоритмів вилучення і розпізнавання окремих символів.

2.3 ПЗ для системи розпізнавання

Завдання розробки перетворення фотографії автомобільного номерного знаку в текст може бути вирішена різними способами. У разі системи розпізнавання пропонується використовувати додаткове програмне забезпечення, вільне для використання в академічних і комерційних цілях, що дозволяє значно зменшити вартість розроблюваного програмного забезпечення.

Процес розпізнавання складається з двох основних етапів:

- визначення області номера автомобіля (рис.2.2);
- розпізнавання тексту в даній області.
-



Рисунок 2.2 – Визначення області, що містить номер автомобілю

На першому етапі використовується зображення, отримане з камери спостереження. Для початку необхідно виділити область, що містить

зображення номерного знаку. Оскільки співвідношення сторін автомобільного номерного знаку регламентовано ГОСТ, можна визначити шукану область.

Після виділення необхідної нам області зображення можна приступати власне до розпізнавання тексту, що міститься в ній (рис.2.3). Ефективним способом перекладу зображень в текстові дані є оптичне розпізнавання символів (OCR).



Рисунок 2.3 – Розпізнавання тексту

Існує вільний у використанні програмний модуль Tesseract OCR, що виконує функцію розпізнавання тексту. Дана програма в даний час є кращою відкритою бібліотекою для розпізнавання символів, має гарну швидкість роботи і добре документована. Бібліотека Tesseract підтримує російську мову тексту і працює під управлінням операційних систем Windows і Linux.

Однак, бібліотека Tesseract чутлива до шумів. Тому фрагмент зображення, відповідний області номера автомобіля, вимагає додаткової обробки перед виконанням оптичного розпізнавання тексту. Наприклад, зменшення можна домогтися за допомогою операцій ерозії і дилатації.

Отже, для повного циклу розпізнавання зображення необхідний інструмент визначення меж об'єкта. Одним з кращих детекторів кордонів є оператор Кенні (рис.2.4).

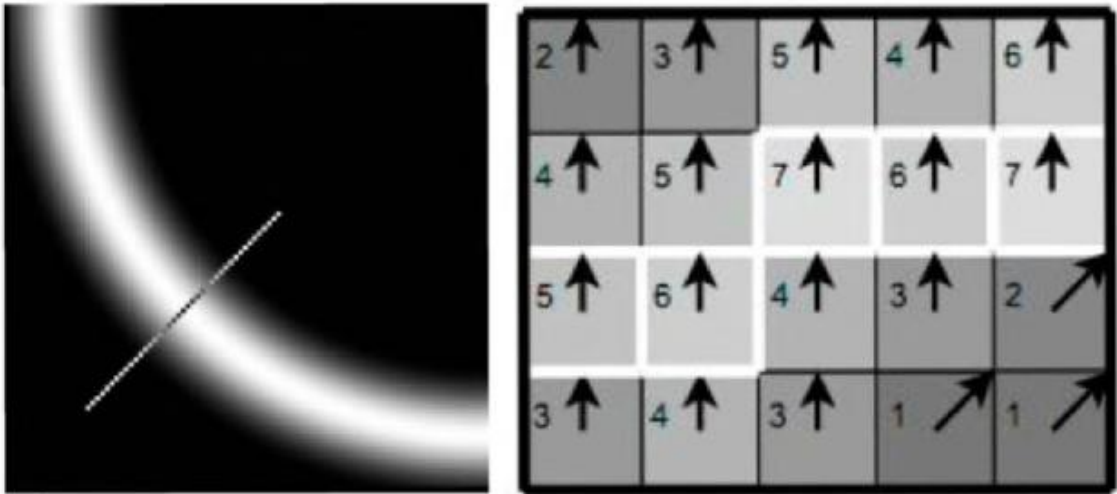


Рисунок 2.4 – Оператор Кенні

Реалізацію даного методу містить бібліотека алгоритмів комп'ютерного зору і обробки зображень OpenCV. Ця бібліотека доступна для будь-якого використання, реалізує також алгоритми дилатації і ерозії, а також підтримує всі популярні формати зображень.

Для виконання завдання розпізнавання зображення від початку до кінця досить інструментарію двох бібліотек: OpenCV і Tesseract. Обидві бібліотеки є прикладними, проте їх можна використовувати не безпосередньо, а за допомогою графічної оболонки (wrapper), що дозволить користуватися ними в керованому коді. Більш того, існує проект Emgu CV, що представляє собою оболонку обох бібліотек (OpenCV і Tesseract) для технологій «.Net». Таким чином, Emgu CV ідеально підходить для вирішення поставленого завдання: всі необхідні функції доступні через єдиний інтерфейс. При цьому з'являється можливість використовувати переваги технології «.Net»: керована пам'ять, гнучкий мову програмування C #, потужні інструменти LINQ, MVVM та інші.

2.4 Технологія оптичного розпізнавання символів

Точність оптичного розпізнавання символів (Optical Character Recognition, OCR) вносить істотний внесок в якість роботи LPR-системи в

цілому. Щоб усвідомити складність розв'язуваної на цьому етапі завдання, розглянемо наступний простий приклад. Припустимо, в розробляється LPR-системі потрібно забезпечити ймовірність правильного розпізнавання номерного знака на статичному зображенні 95%. Визначимо, яка повинна бути ймовірність розпізнавання окремого символу номера.

Нехай безпосередньому розпізнаванню символів передують три алгоритму:

- алгоритм, що локалізує на зображенні номерний знак. Ймовірність $P_{FindLP} = 98,5\%$;

- алгоритм попередньої обробки, нормалізує контраст і яскравість, який коригує зображення. Ймовірність $P_{Preprocessing} = 99.7\%$;

- алгоритм виділення символів, який відповідає за перебування та виділення окремих символів на знаку, і передачу їх алгоритму розпізнавання символів. Ймовірність $P_{ExtractSymbols} = 99.0\%$.

Повна ймовірність правильного розпізнавання символів обчислюється множенням всіх перерахованих вище ймовірностей і повинна становити не менше 0,997.

Наприклад, на основних російських державних реєстраційних знаках нового зразка (з тризначним кодом регіону) 9 символів. Якщо загальна точність оптичного розпізнавання номерного знаку повинна бути не менше 97,7%, то точність розпізнавання окремого символу повинна бути не менше 0,997 тобто допускається, що з 1000 символів, поданих на вхід OCR-модуля, тільки 3 може бути не розпізнано або розпізнано невірно.

Сьогодні відомо досить багато OCR-методів. У структурних методах об'єкт описується як граф, вузлами якого є елементи вхідного об'єкта, а дугами - просторові відносини між ними. Методи, що реалізують подібний підхід, звичайно працюють з векторними зображеннями. Структурними елементами є складові символ лінії.

В першу чергу символ, що розпізнається піддається процедурі отримання скелета, для чого може використовуватися будь-який із

загальновідомих алгоритмів, описаних в тематичній літературі. Далі для кожної особливої точки отриманого скелетного подання символу обчислюється безліч топологічних ознак, основними з яких є:

- нормовані координати особливої точки (вершина графа);
- довжина ребра до наступної вершини у відсотках від довжини всього графа;
- нормоване напрямок з даної точки на наступну особливу точку;
- нормоване напрямок входу в точку, виходу з точки;
- кривизна дуги, точніше «ліва» і «права» кривизна дуги, що з'єднує особливу точку з наступною вершиною (кривизна обчислюється як відношення максимальної відстані від точок дуги до прямої, що з'єднує вершини, до довжини відрізка, що з'єднує ті ж вершини).

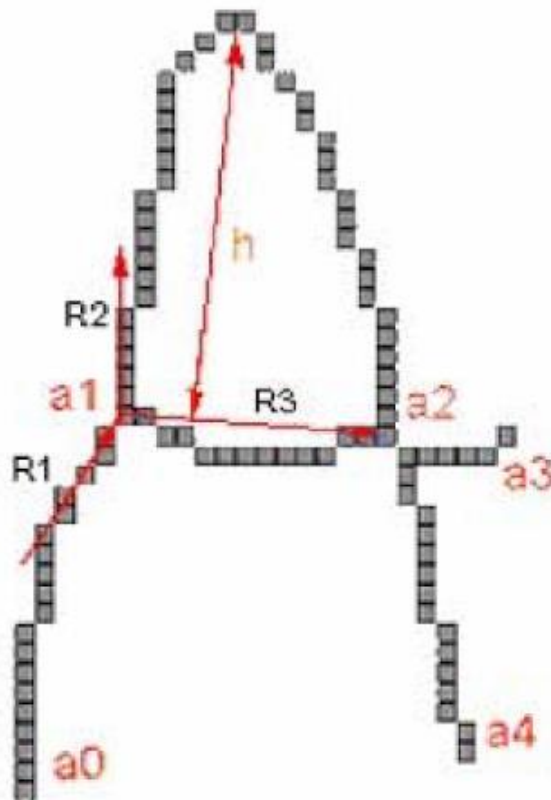


Рисунок 2.5 – Дугова модель

На рисунку 2.5 умовно показані деякі з топологічних ознак. Граф має п'ять особливих точок - a_0 , a_1 , a_2 , a_3 , a_4 . При обході графа по маршруту $a_0 \rightarrow$

$a_1 \rightarrow a_2 \dots$ в вершині a_1 умовно показані наступні ознаки: вектор R_1 - напрямок входу в точку, вектор R_2 - напрямок виходу з точки, вектор R_3 - глобальний напрям на наступну особливу точку. Двонаправлений вектор h показує величину «лівого» відхилення дуги (a_1, a_2) від прямої; «праве» відхилення дорівнює нулю.

Для деяких кодів число особливих точок i , відповідно, число топологічних ознак занадто мало. Так, для коду, що відповідає символу «0», топологічних ознак взагалі немає, тому що немає жодної особливої точки. Тому можуть обчислюватися і використовуватися такі додаткові ознаки:

- розміри і положення компонент і «дірок»;
- «чорна» і «біла» ширина верхньої половини символу;
- модифіковані прямі прогини.

Прогини обчислюються як відстані від точок скелетного подання до опуклої оболонки побудованого уявлення. Додатково запам'ятовується положення точок максимального прогину. Для деяких топологічних кодів число топологічних ознак може бути досить велике, що може зажадати занадто великого набору еталонів для навчання, тому в ряді випадків в розпізнаванні використовується тільки частина ознак.

Символ визначається після порівняння його опису з кодами з бази даних, при цьому вибирається найближчий топологічний код.

Якщо символ після проходу циклу розпізнавання залишився нерозпізнаним, робиться спроба поліпшення зображення за допомогою наступних операцій:

- склеювання кінців ліній за напрямками;
- склеювання точок скелета, що знаходяться на мінімальній відстані одна від одної;
- відкидання найкоротшої лінії (рис.2.6).



Рисунок 2.6 – Спроба поліпшення зображення символу:
 а) вихідне зображення символу; б) символ зі склеєними лініями.

Розглянутий метод не є оптимальним. До його недоліків слід віднести високу чутливість до дефектів зображення, що порушує складові елементи. Також після векторизації можуть з'явитися додаткові дефекти, що викривляють зображення. Крім того, для цих методів (на відміну від шаблонних і ознакових методів) до сих пір не створені ефективні автоматизовані процедури навчання, тому структурні описи найчастіше доводиться створювати вручну.

У реальних LPR-системах найчастіше використовуються комплексні OCR-алгоритми, які представляють собою синтез декількох методів. З представленого опису видно, що ефективна робота OCR-алгоритму істотно залежить від якості зображення, що подається на вхід.

На рисунку 2.7 представлені найбільш типові проблеми зображень, що може погіршувати та викривляти результати розпізнавання через високу чутливість розглянутого методу до дефектів.

Низька роздільна здатність	
Змазане зображення	
Забруднене зображення	
Перетримка	
Нерівномірне освітлення	
Сильне викривлення	

Рисунок 2.7 – Типові проблемні зображення

Технологія отримання зображення визначається середньою якістю зображення, над яким алгоритму розпізнавання доведеться працювати. Очевидно що чим вище якість зображення, тим в кращих умовах працює алгоритм розпізнавання номерів, і тим більша точність може бути досягнута LPR-системою.

Щоб отримати задовільні результати від алгоритму розпізнавання номерів, оброблені зображення повинні містити в собі номерні знаки:

- із задовільним просторовим дозволом;
- із задовільною високою чіткістю;
- із задовільним високим контрастом;

- у задовільних умовах освітлення;
- у задовільній позиції і під правильним кутом.

Звичайно, «задовільним» досить умовне визначення, хоча воно має цілком точне значення.

Технологія отримання зображення вважається задовільною, якщо вона забезпечує стійку, збалансовану, досить хорошу якість зображення при всіх робочих умовах. Якщо LPR-система повинна працювати цілий рік протягом 24 годин на добу, 7 днів на тиждень на території України, вона повинна бути пристосована до будь-яких погодних умов.

2.5 Типові рішення

Мабуть, одним з наймасовіших і затребуваних застосувань LPR-систем є контроль доступу на територію об'єкта і автостоянки. В рамках даних програм ми вже можемо виділити основний тип систем розпізнавання номерів з типовою конфігурацією технічного забезпечення і системним обладнанням.

Прикладом систем розпізнавання номерів може послужити система розпізнавання номерів контролю доступу. Варто відзначити, що в більшості випадків система розпізнавання номерів є лише частиною інтегрованої системи контролю доступу.

Автомобіль наближається до воріт зони, що охороняється, на територію якої він хоче проїхати (рис.2.8). На контрольно-пропускному пункті встановлений шлагбаум і світлофор, палаючий червоним, інформуючи про те, що проїзд заборонений. Крім того, на в'їзді встановлена CCTV камера відеоспостереження. Шлагбаум, світлофор і камера підключені до комп'ютера, на якому встановлено LPR-систему, яка координує операції системи контролю доступу.

Якщо система реєструє наближення транспортного засобу, LPR-модуль намагається «вважати» номерний знак в межах зони контролю (рис.2.9).

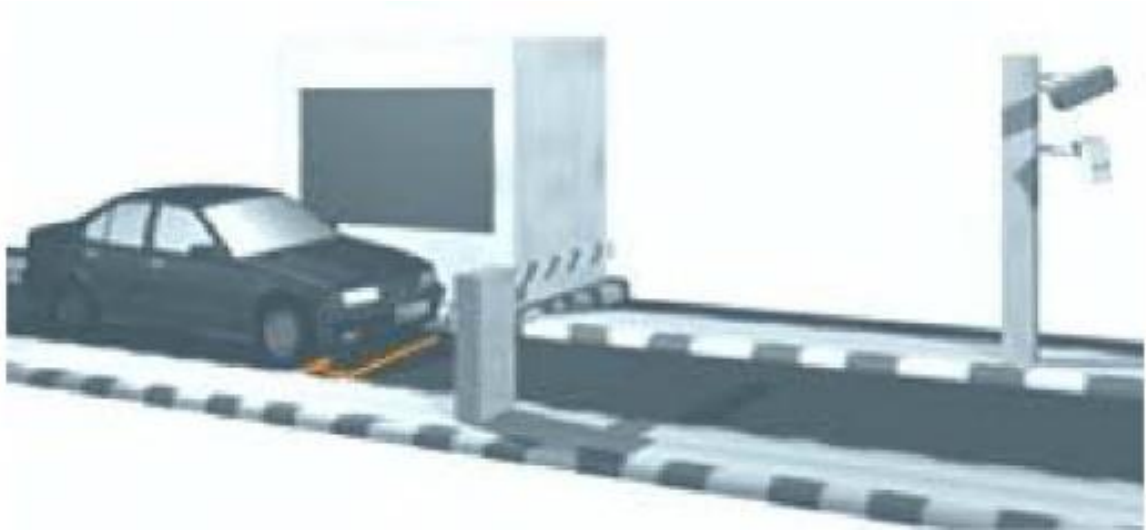


Рисунок 2.8 – Наближення авто до воріт зони, що охороняється

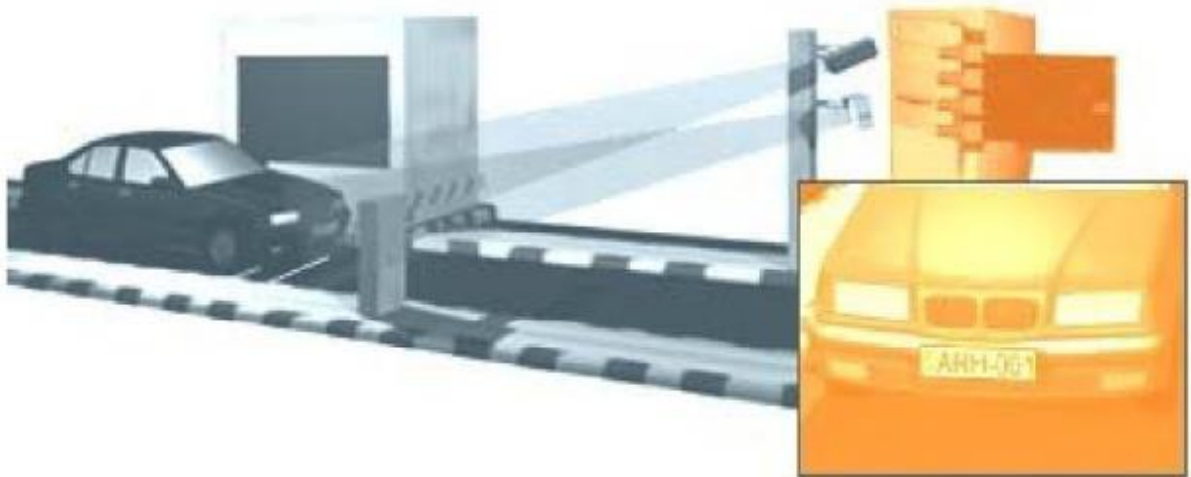


Рисунок 2.9 – Спроба зчитування автомобільного реєстраційного знака

Після зчитування номера, LPR-модуль передає розпізнаний номер для подальшого прийняття рішень.

Далі система контролю і управління доступом відправляє номер в модуль бази даних. Модуль бази даних звіряє отриманий номер зі списками прав доступу і повертає прапор «доступ дозволено» або «доступ заборонено». Залежно від типу прапора додаток контролю доступу відкриває шлагбаум і запалює зелене світло чи ні (рис.2.10).



Рисунок 2.10 – Передача отриманої інформації

Крім того, додаток контролю доступу може відправляти модулю бази даних такі додаткові відомості як дата або час доступу для збереження в журналі подій. Після того як автомобіль проїжджає (або їде), система повертається на вихідну позицію і чекає наступного.

2.6 Огляд існуючих бібліотек для розпізнавання номерів авто

Крім платних програм, призначених для розпізнавання автомобільних номерів, існують безкоштовні програми і бібліотеки, які не поступаються платним аналогам у плані функціоналу. Нижче були розглянуті існуючі бібліотеки для аналізу зображень, їх особливості при використанні, а також переваги та недоліки.

Пошук в мережі показав, що існують кілька вільних проектів бібліотек для розпізнавання номерів авто реалізованих для різних мов програмування.

2.6.1 Проект Opos

Реалізація – C#. Аналіз вихідного коду показав, що для розпізнавання номера авто використовується порт бібліотеки комп'ютерного зору opencv на C#, а для, власне, розпізнавання символів номера використовується обгортка на C# над бібліотекою OCR CuneiFrom - Puma.NET.

Puma.NET - це бібліотека обгортки когнітивних технологій CuneiFrom, яка полегшує інтеграцію функцій оптичного розпізнання символів (OCR - optical character recognition) у будь-якій програмі .NET Framework 2.0 (або вище). API надається через ряд простих класів. Висока продуктивність і точні результати розпізнавання можуть бути досягнуті через пару рядків коду.

Доступна довідкова документація та зразок програми Windows Forms, що полегшує початок розробки в Puma.NET.

Можливості OCR:

- технологія Omni Font - розпізнавання майже всіх друкованих шрифтів;

- підтримуються 27 мов (англійська, німецька, хорватська, польська, датська, португальська, голландська, цифрові, чеська, французька, румунська, угорська, болгарська, словенська, латиська, литовська, естонська, турецька, російська, шведська, іспанська, італійська, російсько-англійська, українська, сербська);

- перевірка орфографії та автоматичні виправлення;

- визначення форматування шрифтів (розміри, підкреслення і т.д.);

- фрагментація та збереження структури документів: абзаци, пробіли, зображення, таблиці тощо;

- покращене розпізнавання спотворених і плямистих зображень, спеціальні режими сканування точкових матриць та факсимільних документів;

- формати вхідних зображень: BMP, GIF, EXIF, JPG, PNG та TIFF;

- вихідні формати: TXT, RTF, HTML.

- SDK доступна за ліцензією BSD. Ви можете завантажити та запустити пакет установки, який буде витягувати необхідні збірки, документацію, зареєструвати COM-сервер, або ви можете завантажити ці вихідні коди.

До додаткових можливостей Puma.NET можна віднести завантаження

одного растрового зображення, його розпізнання та збереження результатів у файл (розпізнання однієї сторінки) або у вигляді рядка (string).

Дана бібліотека не надає можливостей завантажувати кілька зображень та зберегти результати розпізнання в одному файлі (необхідне розпізнання кількох сторінок).

Головним недоліком даної реалізації можна вважати той факт, що зображення для подальшої передачі в Puma.NET спочатку зберігається на диск, що ускладнює використання цієї бібліотеки в системах реального часу.

2.6.2 Проект JavaANPR

Реалізація – Java. JavaANPR - це програма для автоматичного розпізнавання цифрових знаків, яка реалізує алгоритмічні та математичні принципи з області штучного інтелекту, машинного зору та нейронних мереж.

Система автоматичного розпізнавання цифрових таблиць (Automatic number plate recognition system - ANPR) - це спеціальний набір апаратного та програмного забезпечення, який передає вхідний графічний сигнал, як статичні зображення або відеопослідовності, і розпізнає символи номерного знака з нього. Як правило, апаратна частина системи ANPR складається з камери, процесора зображень, тригера камери, пристрою зв'язку та зберігання даних. Система ANPR призначена для перетворення даних між реальним середовищем та інформаційними системами.

Основна перевага цієї бібліотеки в її багатоплатформності. Крім цього, всі алгоритми написані на Java без використання нативних бібліотек, що сильно спрощує використання. Також цю бібліотеку з невеликою доопрацюванням можна використовувати на пристроях під управлінням OS Android. Швидкість розпізнавання однієї картинки з автомобільним номером порядку 02 - 0,8 секунд, що дозволяє використовувати її в системах реального часу. Бібліотека дуже добре документована. Один з недоліків цієї

бібліотеки це те, що бібліотека добре працює на прикладах, які йдуть з нею в комплекті. У разі зашумлених картинок або картинок з поганою освітленістю бібліотека іноді дає не дуже хороший результат.

2.6.3 Проект Automatic License Plate Recognition

Реалізація – C #. Це універсальна технологія, яка в даний час використовується у багатьох різних додатках для дорожнього руху. Це, включає, наприклад, системи контролю доступу, високошвидкісні програми, такі як безкоштовний зворотний зв'язок та правоохоронні програми. Якщо обладнання камер містить відповідні датчики, вони можуть забезпечувати необхідну чутливість для зйомки високоякісних зображень під час короткого періоду експозиції.

Для цієї бібліотеки мають враховуватися важливі певні регіональні особливості, такі як розміри номерних знаків; в Європі рекомендується горизонтальний дозвіл 800 x 1000 пікселів на смугу. Для США, Близького Сходу та Китаю цей параметр збільшується до 1600 x 2000 пікселів на смугу. Крім того, в залежності від властивостей цифрових пластин, в Європі кращим варіантом є камери з підтримкою світлодіодів, тоді як кольоровий варіант з підтримкою білого світла є більш оптимальним на Близькому Сході та в Китаї.

Даний проект використовує дві бібліотеки - порт бібліотеки opencv на C # - Emgu для пошуку номера і порт бібліотеки Tesseract OCR - tessnet для розпізнавання автомобільного номера. Серед переваг даної бібліотеки те, що вона працює з кириличними символами. Серед недоліків можна відзначити, що відсутні приклади для використання бібліотеки в системах реального часу.

2.6.4 Інші проекти

Існують також ще кілька проектів в мережі в тій чи іншій мірі завершеності. Багато з них так чи інакше використовують бібліотеку комп'ютерного зору `opencv` і бібліотеку `Tesseract OCR`. Хоча деякі бібліотеки використовують свій алгоритм розпізнавання символів наприклад, `JavaANPR`. Як правило, ці алгоритми засновані на нейронних мережах, або на аналізі контурів символів. Варто зазначити, що алгоритми, засновані на нейронних мережах бувають чутливі до вибору шрифту. Найефективнішим рішенням, очевидно, було б використання в своєму проекті бібліотеки `opencv` для локалізації номера і бібліотеки `Tesseract OCR` для розпізнавання номера. Цей підхід дозволить найбільш гнучко використовувати весь потенціал цих бібліотек. `Opencv` написана на C і добре оптимізована для використання в системах реального часу. `Tesseract OCR` в даний час є найкращою відкритою бібліотекою для розпізнавання символів, володіє хорошою швидкістю роботи і добре документована.

2.7 Процедура розпізнавання

Розпізнавання образів — процес віднесення об'єкта з фіксованою групою його властивостей до одного об'єкту з множини образів за заздалегідь обумовленим правилом. Наприклад, рибу у тенетах треба поділити на окунів і лососів. Припустимо, що це робиться по довжині риби. Тобто у нас є об'єкт "риба", і за значенням властивості "довжина" ми відносимо рибу або до образу "лосось", або до образу "окунь".

Виділимо найбільш важливі кроки у процедурі розпізнавання:

– сприйняття образу. На цьому етапі проводять отримання значень характеристичних властивостей об'єкта (вимірювання лінійних вимірів, фотографування, оцифровка звуку);

- попередня обробка. Нормалізація. Видалення шумів, представлення зображення в чорно-білому варіанті, обрізання непотрібних частин зображення;
- виділення характеристик (індексація). На цьому етапі вимірюються характеристичні властивості об'єкта (вимірюємо довжину риби та її колір);
- сегментація. Виділення головних областей для розпізнавання;
- класифікація (прийняття рішення).

2.7.1 Система розпізнавання

Розробка системи розпізнавання виконується за наступним кроками:

- отримати навчальну вибірку (тренувальну колекцію) (навчальна вибірка – множина об'єктів, для яких відомі їхні образи. Наприклад колекція аудіо записів для кожного звуку, або колекція зображень кожної букви латинського алфавіту);
- вибрати модель представлення об'єктів;
- вибрати значущі характеристики. Це один з найважливіших етапів розробки системи розпізнавання. Наприклад, якщо у випадку ідентифікації риби окунь / лосось в якості характеристики вибрати тільки довжину риби, то ніяке правило класифікації не зможе точно визначити тип риби, оскільки можна зустріти лосося і окуня однакової довжини;
- розробити правило класифікації. Правило класифікації — правило, яке за значеннями характеристичних властивостей об'єкта відносить його до одного з образів;
- навчання розпізнавальної системи. На цьому етапі алгоритм навчання "збирає досвід" на основі розпізнавання навчальної вибірки. Для того, щоб правильно виставити коефіцієнти (параметри) розпізнавальної системи, алгоритм навчання застосовують на навчальній вибірці, контролюючи результат роботи алгоритму;

– перевірка якості навчання. Повернення назад до кроків 2 (3, 4) ... Якщо частота помилок алгоритму не влаштовує, то необхідно повернутися до п. 2, 3, 4. Інтуїтивно зрозуміло, що збільшення кількості характеристичних властивостей, збільшення тренувальної колекції покращують якість роботи алгоритму;

– оптимізація розпізнавальної системи. Після того, як якість роботи системи розпізнавання підходить під умову розглянутої задачі, іноді доводиться провести його оптимізацію. Початковий алгоритм розпізнавання може бути занадто довгим або ресурсоємним. Прискорити процедуру розпізнавання можна зменшивши кількість характеристичних властивостей об'єкта, вибравши інші характеристичні властивості, використовуючи інше правило класифікації.

2.7.2 Найпоширеніші методи розпізнавання

Виділяють 4 групи методів розпізнавання:

– порівняння із зразком. Застосовуємо геометричну нормалізацію і вважаємо відстань до прототипу. Найбільш наочно застосування цього методу в розпізнаванні тексту. Завдання. У нас є зображення відсканованого символу і колекція зображень зразків (усіх букв абетки), ми хочемо визначити, якій букві алфавіту відповідає відскановане зображення. Розв'язок. Масштабуємо зображення символу до розмірів зразків і виберемо той, відстань до якого мінімальна;

– нейронні мережі. Вибираємо вид мережі і налаштовуємо коефіцієнти. На вхід нейронної мережі подається об'єкт для розпізнавання. Група рецепторів мережі відповідає за прийом своєї характеристичної властивості;

– статистичні методи. В основі лежить уявлення про клас розпізнаваних об'єктів як про реалізацію деякої випадкової величини. Цю

випадкову величину з визначеними статистичними характеристиками називають статистичною моделлю класу розпізнаваних об'єктів;

– структурні та синтаксичні методи. Розбираємо об'єкт на елементи. Будуємо правило, в залежності від входження окремих елементів та їх послідовностей.

3 РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Обґрунтування вибору середовища програмної реалізації

3.1.1 Особливості Xamarin Framework

У сфері розробки ПЗ досить широку популярність здобули ряд фреймворків (наприклад PhoneGap), які пропонують розробку кросплатформових мобільних додатків на HTML5 з використанням JavaScript. Ідея полягає в тому, що програма розробляється як звичайний сайт для мобільних пристроїв з використанням відповідних js-бібліотек, наприклад, JQuery Mobile. Наступний крок - запакування у деякий контейнер, який для користувача виглядає як прикладний додаток. Мінуси цих фреймворків очевидні: по-перше, розробник не має доступу до прикладних елементів UI. По-друге, ви отримуєте урізаний і узагальнений API для роботи з платформою. Таким чином, ті чи інші особливості, властиві для окремої платформи будуть недоступні. Та найважливішим мінусом є те, що такий додаток фізично запускається всередині браузера телефону. Це означає низьку продуктивність та проблеми з відображенням. Однак, звичайно, в певних випадках ці фреймворки можуть виявитися дуже доречні.

Xamarin — влаштований інакше. Він заснований на open-source реалізації платформи .NET - Mono. Ця реалізація включає в себе власний компілятор C #, середу виконання, а також основні .NET бібліотеки. Мета проекту - дозволити запускати програми, написані на C #, на операційних системах, відмінних від Windows - Unix-системах, Mac OS і інших. Важливо, що розробкою Xamarin займаються ті ж люди, що і розробкою Mono.

Xamarin - це фреймворк для кросплатформової розробки мобільних додатків (iOS, Android, Windows Phone) з використанням мови C #. Ідея полягає у наступному: пишеться програмний код із можливим застосуванням всіх можливостей SDK платформи і рідного механізму створення UI. На виході отримуємо додаток, який нічим не відрізняється від прикладних додатків і майже не поступається їм у продуктивності.

Фреймворк складається з декількох основних частин:

- Xamarin.iOS - бібліотека класів для C #, що надає розробнику доступ до iOS SDK;
- Xamarin.Android - бібліотека класів для C #, що надає розробнику доступ до Android SDK;
- компілятори для iOS і Android;
- IDE Xamarin Studio;
- плагін для Visual Studio.

З точки зору виконання додатків між iOS і Android є одна ключова відмінність - спосіб їхньої попередньої компіляції (рис.3.1). Як відомо, для виконання додатків в Android використовується віртуальна Java-машина Dalvik. Прикладні додатки, які пишуться на Java, компілюються у проміжний байт-код, який інтерпретується Dalvik`ом в команди процесора в момент виконання програми. Це так звана Just-in-time компіляція (компіляція у режимі реального часу). В iOS використовується інша модель компіляції - Ahead-of-Time (компіляція перед виконанням). Xamarin враховує цю різницю, надаючи окремі компілятори для кожної з цих платформ, які дозволяють на виході отримувати справжні, прикладні додатки, які виконуються поза контекстом браузера і можуть використовувати всі апаратні і програмні ресурси платформи.

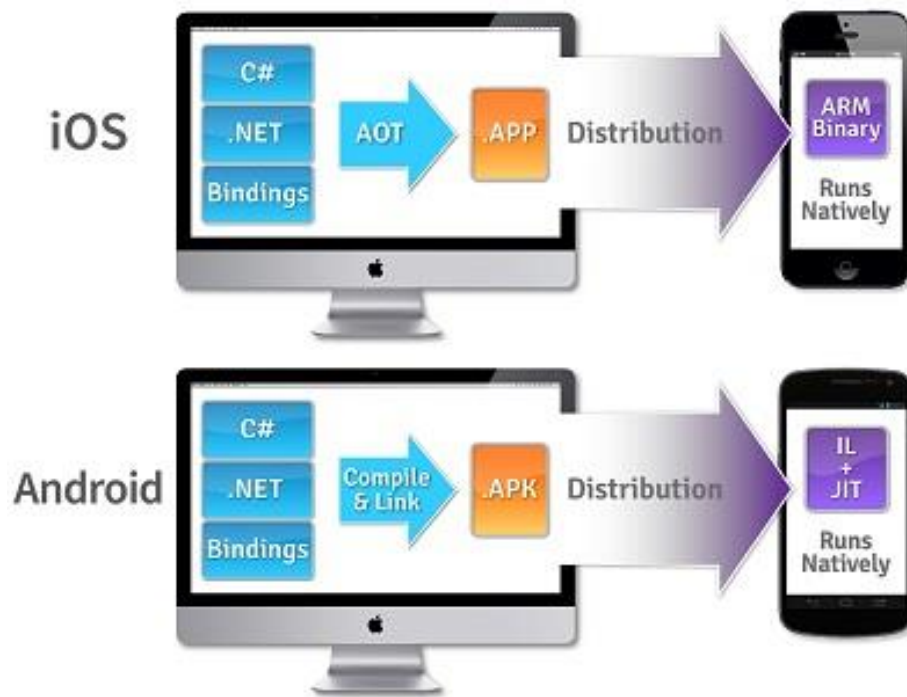


Рисунок 3.1 – Відмінність iOS та Android

Для iOS віртуальної машини немає і програмний код повинен бути заздалегідь скомпільований в машинний. Для цієї мети використовується AOT компілятор Mono.

Для Android інакше. При компіляції програми відбувається переклад коду на C # в проміжний байт-код, зрозумілий віртуальній машині Mono і сама вона також додається в заповнений додаток. І Mono і Dalvik написані на C і працюють поверх ядра Linux (Android заснована на Linux). При запуску програми на Android обидві віртуальні машини починають працювати взаємозв'язано і обмінюються даними через спеціальний механізм обгортки (рис.3.2).

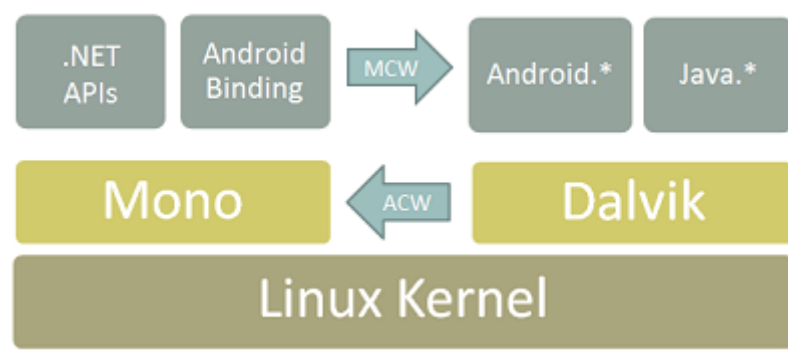


Рисунок 3.2 – Особливості Android

Бібліотека класів `Monotouch.dll` для `Xamarin.iOS` надає доступ до всіх можливостей `iOS SDK`. Для розробника це просто набір `C#`-класів з анотацією. У середині ці класи використовують розроблені інженерами `Xamarin` прикладні класи і методи. Більшість класів і методів називаються так само, як в оригінальному `iOS SDK`, хоча бувають винятки. У класах активно використовується механізм `C#` подій, що дозволяє написати досить компактний код обробників з використанням лямбда-виразів (рис.3.3).

```
button.TouchUpInside += (s, o) => {
    message.Text = "Hello!";
};
```

Рисунок 3.3 – Використання лямбда-виразів у `Xamarin`

Для асинхронної розробки `Xamarin` надає можливість використовувати як класи з простору імен `System.Threading.Thread` і `System.Threading.ThreadPool`, так і повний спектр можливостей, що надаються `Task Parallel Library`. Використання останньої, однак, вважається кращим. Крім того, після останніх оновлень можна використовувати ключові слова `async / await`.

Для кожної платформи `Xamarin` надає можливість використовувати прикладні застосунки розробки `UI` і прикладні елементи інтерфейсу користувача. Для `Android` створення `UI` може відбуватися безпосередньо в коді або ж за допомогою декларативного підходу з описом інтерфейсу в `XML`. Для `iOS` це також або код, або використання прикладних засобів проектування інтерфейсу - окремі `xib`-файли або ж один великий `Storyboard`. Редагування цих файлів відбувається в звичній для `iOS`-розробника середовищі `XCode`.

`Xamarin` є засобом кросплатформової розробки, тобто додаток, написаний один раз, може працювати на різних мобільних платформах. Але є

нюанс. Для кожної з платформи потрібно реалізувати власний шар UI. Тобто код, який відповідає за зовнішній вигляд програми, вам доведеться написати для кожної платформи окремо. Якщо розбивати додаток на шари, можна отримати:

- Data Layer (DL) - сховище даних, наприклад, база SQLite або xml-файли;
- Data Access Layer (DAL) - обгортка над сховищем для здійснення CRUD-операцій;
- Business Layer (BL) - шар, що містить бізнес-логіку додатка;
- Service Access Layer (SAL) - шар, який відповідає за взаємодію з віддаленими сервісами (Rest, Json, WCF);
- Application Layer (AL) - шар, що містить залежний від платформи код, іншими словами, це код, який залежить від бібліотек `monotouch.dll` або `monodroid.dll`;
- User Interface Layer (UI) - шар призначеного для користувача інтерфейсу (рис.3.4).



Рисунок 3.4 – Шари Xamarin-застосутку

Кросплатформовими являються усі шари, розташовані вище Application Layer. Частка переносимого коду досить сильно залежить від самого додатка. Інженери Xamarin це розуміють, тому прагнуть до збільшення цієї частки. Як досягнень у вирішенні цієї проблеми можна розглядати бібліотеку Xamarin.Mobile. Вона надає єдиний для різних платформ API для роботи з камерою, контактами та гео-локацією. Але використання цієї бібліотеки ніяк не обмежує розробника в застосуванні платформозалежного API, наприклад, за допомогою механізму делегатів.

У Xamarin існує власний магазин сторонніх компонентів Xamarin Components. Він інтегрується в IDE і дозволяє в підключати до проекту різні компоненти, написані як інженерами Xamarin, так і сторонніми розробниками. Кількість компонентів постійно зростає. Є як платні, так і безкоштовні. Всі компоненти можна розділити на дві частини. Одні надають додаткові елементи призначеного для користувача інтерфейсу, інші є бібліотеками класів. Наприклад варіант для Mono відомої бібліотеки для роботи з Json - Json.NET або ж бібліотека для взаємодії з Rest-сервісами - RestSharp. Не всі компоненти кросплатформові, багато таких що доступні тільки для конкретної платформи. Як я згадував вище, Xamarin використовує механізм «binding» (зв'язування) для зв'язування з прикладними бібліотеками класів, що дозволяє перенести на C # будь-які пикладні бібліотеки класів. Крім того, для Xamarin.iOS, існує спеціальна утиліта, яка вміє генерувати такі зв'язування автоматично. Власне це дозволяє інженерам Xamarin встигати за всіма нововведеннями iOS.

Xamarin має відмінну документацію, яка містить докладні керівництва, а також значну базу прикладів. Документація безпосередньо по всіх класах бібліотек Monotouch (для iOS) і Monodroid (для Android) є частиною загальної документації Mono. Але, на жаль, цього все одно недостатньо, щоб покрити усі питання, які можуть виникати в процесі розробки. У Xamarin існує спільнота розробників, який сконцентрована на офіційному форумі і на StackOverflow.

3.1.2 Середовище розробки

Розробники Xamarin як середовище розробки пропонують використовувати або власну IDE - Xamarin Studio, або Visual Studio.

Xamarin Studio - кросплатформова IDE, яка працює як на Mac OS X, так і на Windows. Вона достатньо проста у використанні, має ряд додаткових функцій, присутніх у таких поширених середовищах розробки як Visual Studio і ReSharper. Деякі з них:

- автодоповнення коду (включаючи можливість одночасного імпорту namespaces);
- зручний універсальний пошук за назвами файлів, типам, членам класів тощо;
- розвинені можливості навігації по проекту (швидкий перехід до опису класу, перехід до базового класу, список місць використання класу і т.д.);
- різні механізми рефакторінга і швидка підказка;
- досить розвинені механізми налагодження, включаючи стеження, перегляд поточного значення змінної при наведенні, візуалізацію потоків і аналог «Immediate window» в VS;
- вбудована інтеграція з системами контролю версій.

Xamarin пропонує можливість вести розробку в Visual Studio після установки спеціального плагіна, який доступний в business-ліцензії.

Використання Xamarin можливе після створення кросплатформового проекту. Ви маєте можливість використовувати усі особливості даної середовища розробки. Після установки плагіна для Visual Studio для розробки iOS вам буде потрібно налаштувати з'єднання з вашим Mac, яке буде використано при запуску проекту на виконання. Тобто після запуску, додаток автоматично пересилається на Mac, де компілюється і завантажується або на симулятор або на пристрій, при цьому сам процес налагодження, розстановка точок зупинок продукту і т.д. буде відбуватися в Visual Studio.

На поточний момент технологія Xamarin є серйозним інструментом для вирішення складних завдань в області розробки мобільних додатків. Незважаючи на це, команда розробників не зупиняється і продовжує його активний розвиток та поліпшення. На мій погляд у технології велике майбутнє і з кожним днем число розробників, які використовують її в якості основного фреймворка для розробки буде неухильно зростати.

3.1.3 Оптичне розпізнавання символів в .NET

Emgu CV - це кросплатформова .Net бібліотека обробки зображень OpenCV. Дозволяє викликати функції OpenCV з мов сумісних з .NET, таких як C #, VB, VC ++, IronPython тощо. Обгортку може бути складена у Visual Studio, Xamarin Studio та Unity. Продукт з її використанням може працювати на Windows, Linux, Mac OS X, iOS, Android і Windows Phone.

Emgu CV написано повністю у C #. Перевага в тому, що вона може бути скомпільована в Mono, і тому вона може працювати на будь-якій платформі Mono, включаючи iOS, Android, Windows Phone, Mac OS X і Linux. Багато зусиль було витрачено на виконання чистої C #, оскільки заголовки повинні бути перенесені, у порівнянні з керованою програмою C++, де заголовки можуть бути включені. Emgu CV може використовуватися у декількох різних мовах, включаючи C #, VB.NET, C ++ та IronPython.

Переваги:

- клас зображення із загальним кольором і глибиною;
- автоматичне збирання сміття;
- XML Документація та підтримка intellisense;
- вибір або використання класу Image або функції прямого виклику з OpenCV;
- загальні операції на пікселях зображення.

3.2 Програмна реалізація

3.2.1 Алгоритм реалізації

Розглянемо два головних етапи використаного алгоритму – нормалізацію та сегментацію.

Під нормалізацією у даній роботі розуміється поворот зображення автомобільного номера в площині зображення так, щоб рядок символів розташовувалася горизонтально. Для вирішення цього завдання використовується глобальний аналіз зображення за допомогою перетворення Хафа.

Сегментація символів ґрунтується на використанні моделей розташування символів на номері. В ході зіставлення різних моделей з реальним зображенням визначається модель, що має найкращу відповідність, параметри якої використовуються для отримання координат символів. Такий підхід дозволив також визначати тип номера і приналежність символу до букв або цифрах, що полегшує їх подальше розпізнавання.

Запропонований алгоритм нормалізації і сегментації символів дозволяє використовувати його в системах розпізнавання автомобільних номерів.

Нормалізація зображення номерного знаку проводиться у два етапи. На першому етапі визначається кут повороту номера в площині зображення. На другому - виконується алгоритм отримання нормалізованого зображення номера з вихідного зображення з урахуванням кута його повороту.

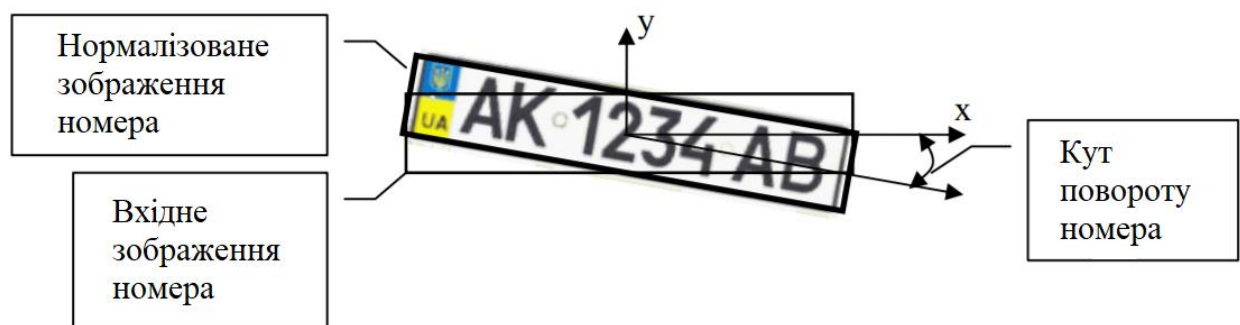


Рисунок 3.5 – Схема отримання нормалізованого зображення номера

Для повороту області зображення, що відповідає нормалізованому номеру, використовується алгоритм, заснований на відповідному афінному перетворенні координат. Для зменшення спотворень зображення при повороті, пов'язаних з його дискретним характером, використовується метод, заснований на білінійній інтерполяції по найближчих чотирьох пікселях.

Визначення кута повороту зображення номерного знаку виконується з використанням декількох етапів обробки та аналізу зображень.

На першому етапі виконується операція підкреслення кордонів на зображенні на основі лінійного оператора Собеля для горизонтальних кордонів, має маску згортки:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

Наведений оператор більш чутливий до напрямів кордонів, близьким до горизонтального, тому дозволяє добре виділити на зображенні верхню і нижню частину номерного знаку, як показано на рисунку 3.6.



а)



б)



в)

Рисунок 3.6 – а) фрагмент вихідного зображення зі знайденим положенням номера; б) вирізане зображення номерного знаку з розширенням на 40% у вертикальному напрямку; в) результат підкреслення кордонів

На другому етапі виконується розрахунок карти щільності знайдених точок меж у просторі коефіцієнтів лінійних залежностей просторових координат згідно перетворенню Хафа. Метою другого етапу є визначення рівняння прямих, що відповідають верхній і нижній межі номерного знаку. Кожна точка карти кордонів, отриманої на попередньому етапі, породжує ціле сімейство прямих, що проходять через неї:

$$y_{ij} = ax_{ij} + b,$$

що в просторі лінійних коефіцієнтів також відповідає прямій:

$$b = x_{ij} - y_{ij}.$$

Наділяючи прямі в просторі коефіцієнтів вагою $v(x_{ij}, y_{ij})$, відповідним значенням яскравості зображення результату підкреслення меж (рис.3.6(в)), і проводячи їх в просторі коефіцієнтів a і b з яскравістю, рівною вазі, отримаємо зображення, подібні до наведеного на рисунку 3.7.

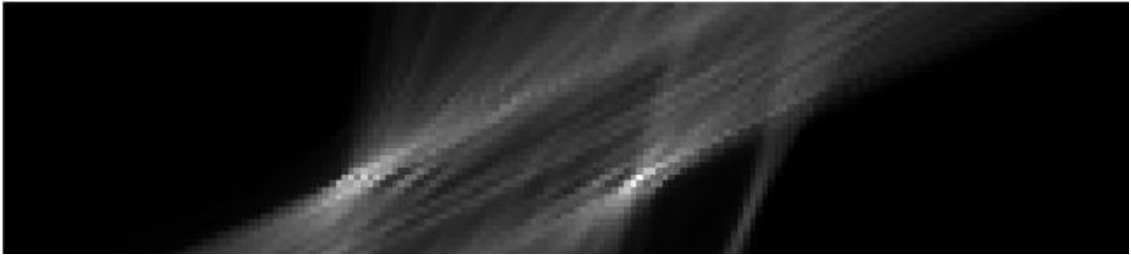


Рисунок 3.7 – Карта результатів перетворення Хафа

Вертикальний напрямок карти, наведеної на рисунку 3.7, відповідає змінам значень коефіцієнта a , горизонтальне - b . Діапазон зміни коефіцієнтів розраховується виходячи з допустимих поворотів зображення номера в площині зображення і розмірів знайденого зображення номера відповідно до формул:

$$a \in [-0,2; 0,2] (0,2 \approx \text{tg}(10^\circ)),$$

$$b \in [-a_{\max} W; H + a_{\max} W],$$

де a_{\max} - максимальне значення a , W - ширина зображення номера, H - висота зображення номера. Найбільш яскраві точки зображення карти результатів перетворення Хафа (рис.3.7) відповідають найбільш яскравим прямим вихідного зображення кордонів (рис.3.6(в)), що дозволяє визначити рівняння цих прямих і, відповідно, кути їх нахилу. Так як шуканими є прямі, відповідні до верхньої та нижньої меж номерного знаку, то для визначення найбільш вірогідного кута повороту номера на зображенні карти результатів перетворення Хафа шукаються дві точки, що мають одне значення a і різні

значення b , з найбільшою сумарною яскравістю. Значення параметра a знайдених точок повністю визначає кут нахилу зображення номерного знаку ($a = \text{tg}(\varphi)$).

Для сегментації символів на автомобільному номері пропонується використовувати підхід, заснований на підгонці під реальне зображення різних моделей розташування символів на номері. Кожна з моделей відповідає певному стандарту розташування символів. У даній роботі розглядаються дві моделі розташування символів в однорядкових номерних знаках, які можна представити наступними типами: {ЛЛ ЦЦЦЦ ЛЛ} і {ЦЦЦЦ-ЦЦ ЛЛ}, де Л - літера, Ц - цифра. Кожну модель можна представити у вигляді зображення темних прямокутників, відповідних символів, на світлому тлі, як показано на рисунку 3.8.

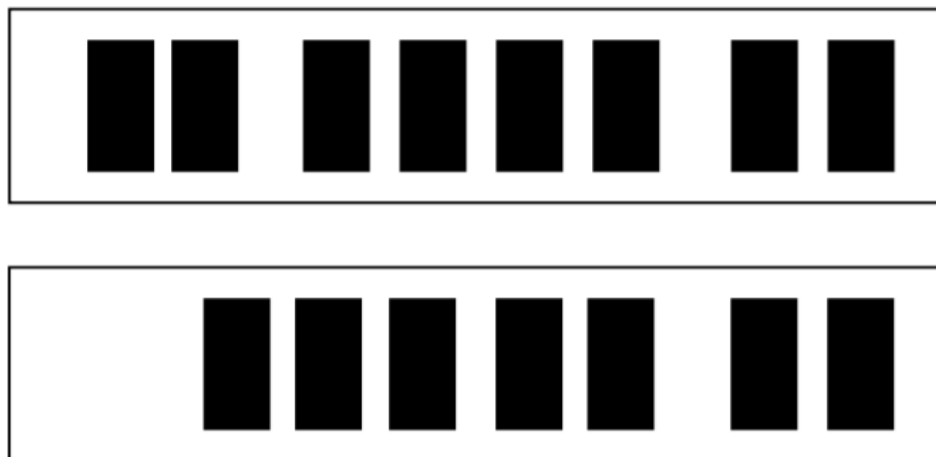


Рисунок 3.8 – Приклади використовуваних моделей розташування символів

Якщо номери мають інший тип, наприклад, білі символи на червоному фоні, або на чорному, то перед зіставленням моделі із зображенням останнім можна інвертувати по яскравості.

Як критерій відповідності моделі зображенню номера використовується величина:

$$K(x, y, W, H) = \frac{S_w - S_b}{\sigma_w}.$$

Тут S_w - середня яскравість зображення під світлою областю, S_b - середня яскравість зображення під чорної областю, σ_w - середній квадратичний розкид яскравості зображення під світлою областю, x і y - координати моделі всередині зображення номера, W і H - горизонтальний і вертикальний розміри моделі. Чим більше значення критерію $K(x, y, W, H)$ тим більше модель відповідає зображенню номера. Підгонка моделі під зображення полягає у виборі згідно з прийнятим критерієм найкращого положення і розмірів моделі. Після цього, згідно з тим ж критерієм, приймається рішення про найкращий тип моделі для поточного зображення номера. Якщо враховувати можливі похибки при нормалізації номера, на основі запропонованого критерію можна незалежно визначити більш точне положення для кожного символу окремо поблизу знайденого його положення з використанням всієї моделі.

Обчислення значення критерію $K(x, y, W, H)$ в кожному можливому положенні моделі всередині зображення номерного знаку може бути ефективно виконано на основі використання інтегральних зображень для вихідного зображення номера і зображення квадратів значень яскравості його пікселів.

Приклади спільної роботи послідовних етапів нормалізації зображення номера і сегментації символів наведено на рисунку 3.9. У верхньому рядку рисунку 3.9 показані вихідні зображення номерів, у другому рядку - результат їх нормалізації, в нижньому рядку - результат сегментації символів на номері (кожен символ являє собою окреме зображення).



Рисунок 3.9 – Приклади роботи алгоритмів нормалізації зображення номера і сегментації символів

Отримані після сегментації зображення символів дозволяють використовувати їх для вирішення задачі розпізнавання. Запропонований алгоритм сегментації дозволяє також визначити тип номерного знаку, а отже, з'ясувати, чи є кожен із символів буквою або цифрою, що дозволить полегшити аналіз зображень на наступному етапі розпізнавання. Запропоновані досить прості моделі дозволяють виконувати операцію сегментації з високою ефективністю за якістю і швидкості, а також дозволяють в разі потреби без значних додаткових зусиль розширити число використовуваних моделей номерного знаку шляхом введення в розгляд інших можливих розташувань символів.

3.2.1 Ідентифікація регіону номерного знаку

Після отримання розпізнаного номерного знаку у вигляді символного поля проводиться операція ідентифікації для визначення додаткової інформації, а саме – регіону номера автомобілю та його дати реєстрації.

На рисунку 3.10 зазначено регіони та відповідні їм серії автомобільних номерів і дати реєстрації.

№ Региона	Регион(область)	С 2004	1997 - 2004	До 1997
01	Республика Крым	АК	КР, КО, КТ	ЦС, КР,МЯ
02	Винницька область	АВ	ВІ, ВТ, ВХ	ВІ, ВІ
03	Волынська область	АС	ВО, ВК, ВМ	ВН, ВО
04	Дніпропетровська область	АЕ	АА, АВ, АЕ, АК, АН	ДН, ДП, АЕ, ЯА
05	Донецька область	АН	ЕА, ЕВ, ЕК, ЕН, ЕО, ЕС	ДО, ДЦ, СЛ, ЯН
06	Житомирська область	АМ	ВА, ВВ, ВЕ	ЖІ, ІО
07	Закарпатська область (Ужгород)	АО	РЕ, РР, РТ	ЗА, ІК
08	Запорожська область	АР	НА, НЕ, НО, НР, НС	ЗП, ЗР, ЯТ
09	Івано-Франківська область	АТ	ІВ, ІС	ІВ, СЯ, ІФ, ІВ
10	Київська область	АІ	КК, КХ, КМ	КІ, КХ, МІ
11	місто Київ	АА	КА, КІ, КВ, КЕ, КН	КІ, КІ, ХТ
12	Кіровоградська область	ВА	ОМ,ОС,ОН	КД, ЕІ
13	Луганська область	ВВ	АМ, АО, АР, АТ, АХ	ВГ, ЛУ, АІ
14	Львівська область	ВС	ТА, ТВ, ТН, ТС	ЛВ, ІН
15	Ніколаївська область	ВЕ	НІ, НК, НТ	НІ, НІ
16	Одеська область	ВН	ОА, ОВ, ОЕ, ОК	ОД, ОІ
17	Полтавська область	ВІ	СК, СН, СС	ПО, ІХ
18	Рівненська область	ВК	РВ, РА, РО	РО, РВ
19	Сумська область	ВМ	СА, СВ, СЕ	СУ, СІ
20	Тернопільська область	ВО	ТЕ, ТІ, ТК	ТЕ,ТІ
21	Харківська область	АХ	ХА, ХЕ, ХІ, ХК, ХК	ХА, ХК, ХІ
22	Херсонська область	ВТ	ХО, ХН	ХО
23	Хмельницька область	ВХ	ХМ, ХТ, ХІ	ХМ
24	Черкаська область	СА	МА, МВ, МЕ	ЧК, ЧН, РК
25	Чернігівська область	СВ	МК, ММ, МН	ЧН, ЧТ, ЧГ, РМ
26	Чернівницька область	СЕ	МО, МР, МС	ЧВ, ІС
27	місто Севастополь	СН	КС	ЦС, КР, МЯ

Рисунок 3.10 – Таблиця серій автомобільних номерів

3.3 Інструкція користувача

У даній роботі було розроблено мобільний додаток для операційної системи Android, який також можливо розгорнути за запустити на локальному комп'ютері.

Мобільний пристрій користувача повинен мати версію Android не менш 4.0, та розмір оперативної пам'яті не менше 500 МБ для успішного запуску та роботи програми.

Для запуску розробленого додатку користувачу потрібно завантажити на мобільний пристрій файл програми і запустити його. Вказаний файл міститься на диску або флеш-карті.

Програма в цілому і функції, які необхідно реалізувати, має на увазі, що користувач не має стабільного підключення для виходу в мережу Інтернет або по локальній мережі для з'єднання з сервером, проте періодично здійснює підключення до мережі для оновлення та актуалізації даних.

Головна проблема, з якою стикаються користувачі в початковій версії системи - неточність розпізнання тексту або ж її неможливість. Надалі будуть модифіковані методи обробки зображень, а також розширено спектр можливих причин відмови додатку. Також планується офіційне оформлення логотипу та назви продукту, розміщення програми на мобільному порталі для більш зручного завантаження та інсталювання.

3.4 Тестування розробленої моделі

3.4.1 Тестування на локальному ПК

Тестування проводиться на ПК з операційною системою Windows 10, обсягом оперативної пам'яті 8,00 ГБ.

Для запуску додатку було відкрито «Пуск» - «Diploma.UWP» (рисунок 3.11) та запущено. Початковий екран після запуску додатку зображений на рисунку 3.12.

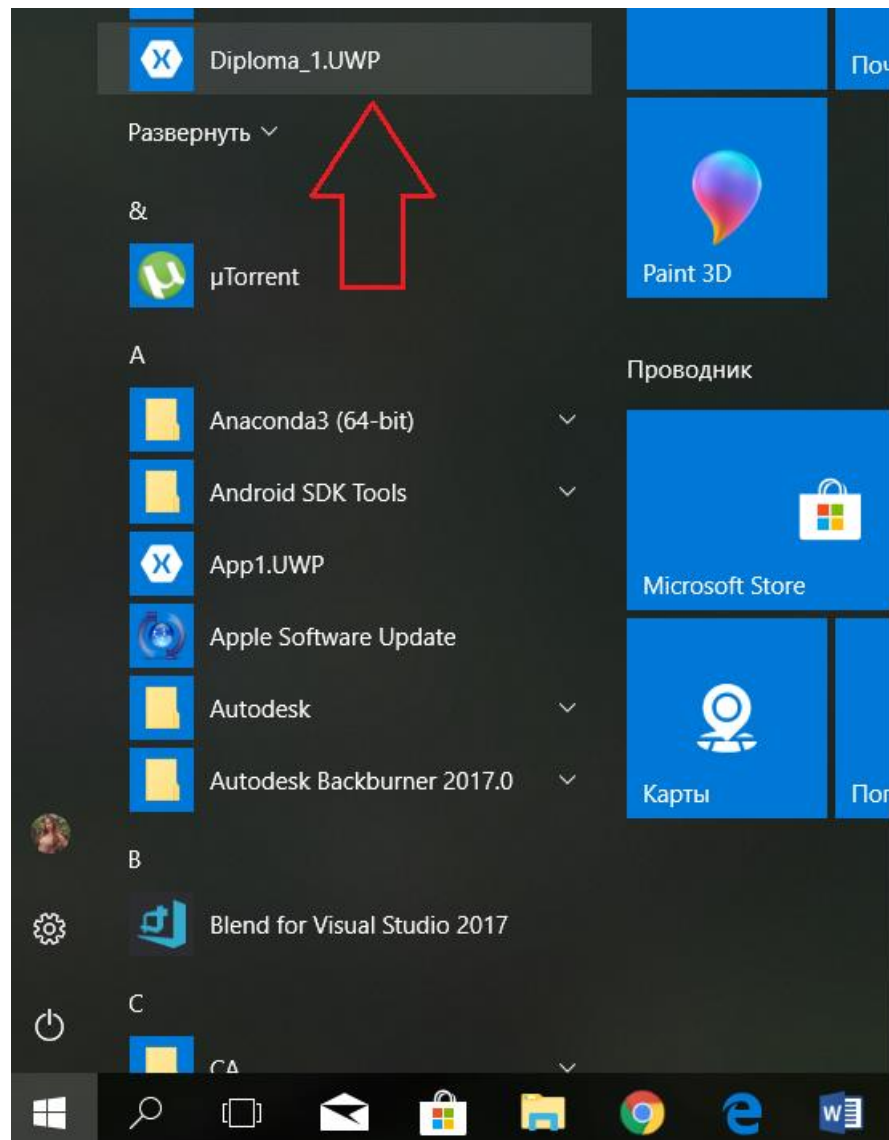


Рисунок 3.11 – Додаток у панелі управління Windows

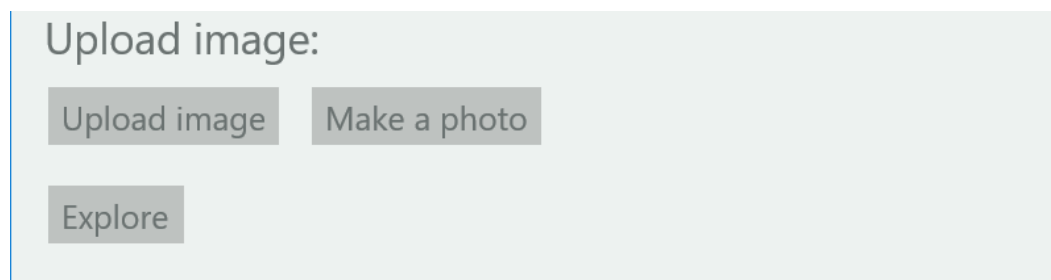


Рисунок 3.12 – Початковий екран при запуску додатку

Після натискання на кнопку «Upload» завантажуються зображення для подальшої обробки та розпізнавання та відображається на екрані користувача (рисунок 3.13). Швидкість завантаження залежить від розміру файлу: чим

більший файл, тим довше він завантажується. Також було протестовано завантаження таких форматів зображень як jpeg, png та bmp і виявлено, що файли формату jpeg завантажуються швидше за інші.

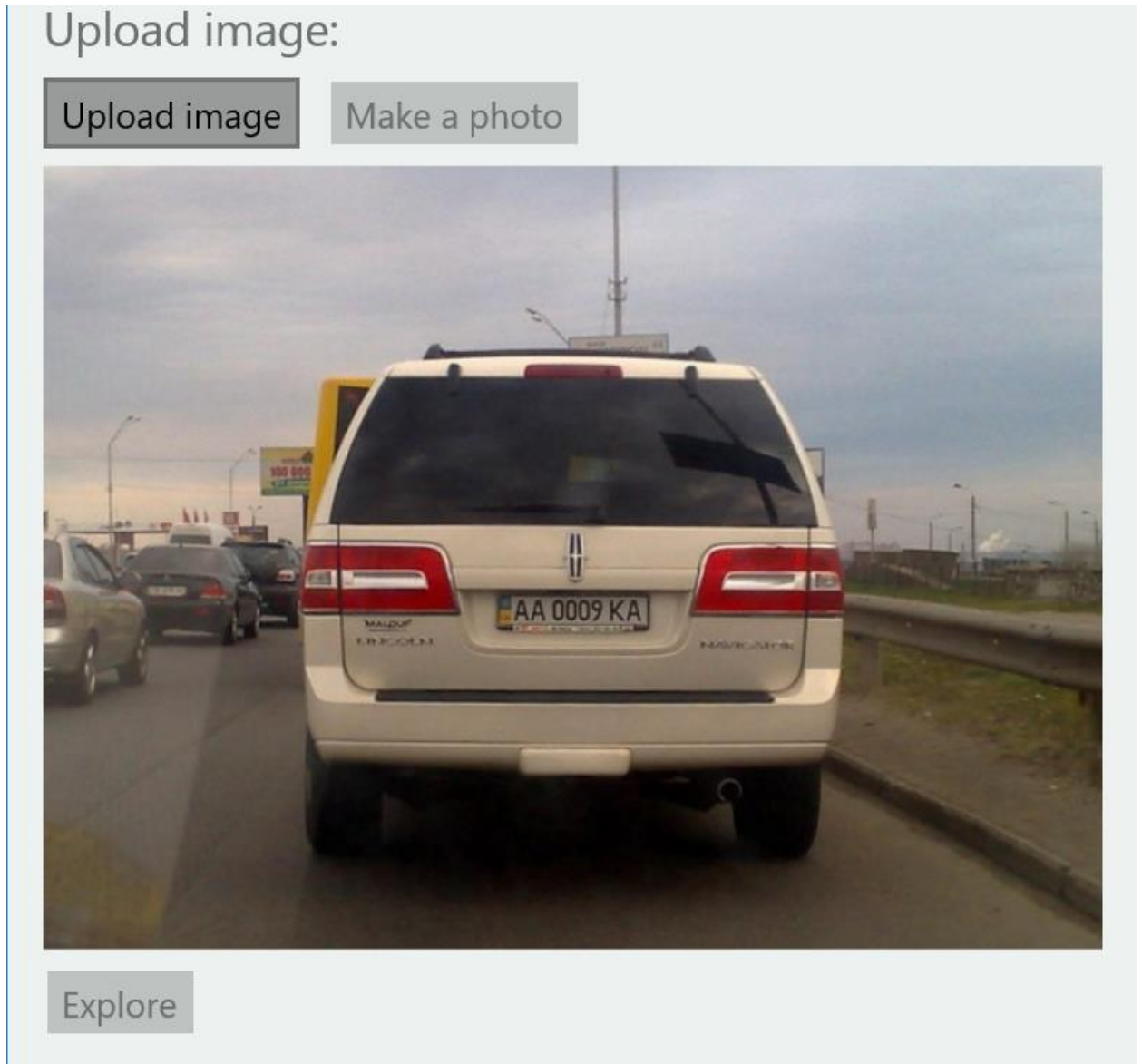


Рисунок 3.13 – Завантаження зображення для розпізнання

Кнопка «Explore» активує алгоритм розпізнавання завантаженого зображення, виводить детектовані символи, а також надає додаткову інформацію щодо регіону та дати реєстрації даного номера (рисунок 3.14).

Upload image:

Upload image

Make a photo



Explore

Number: AA 0009 KA

Info: г.Киев - 2004

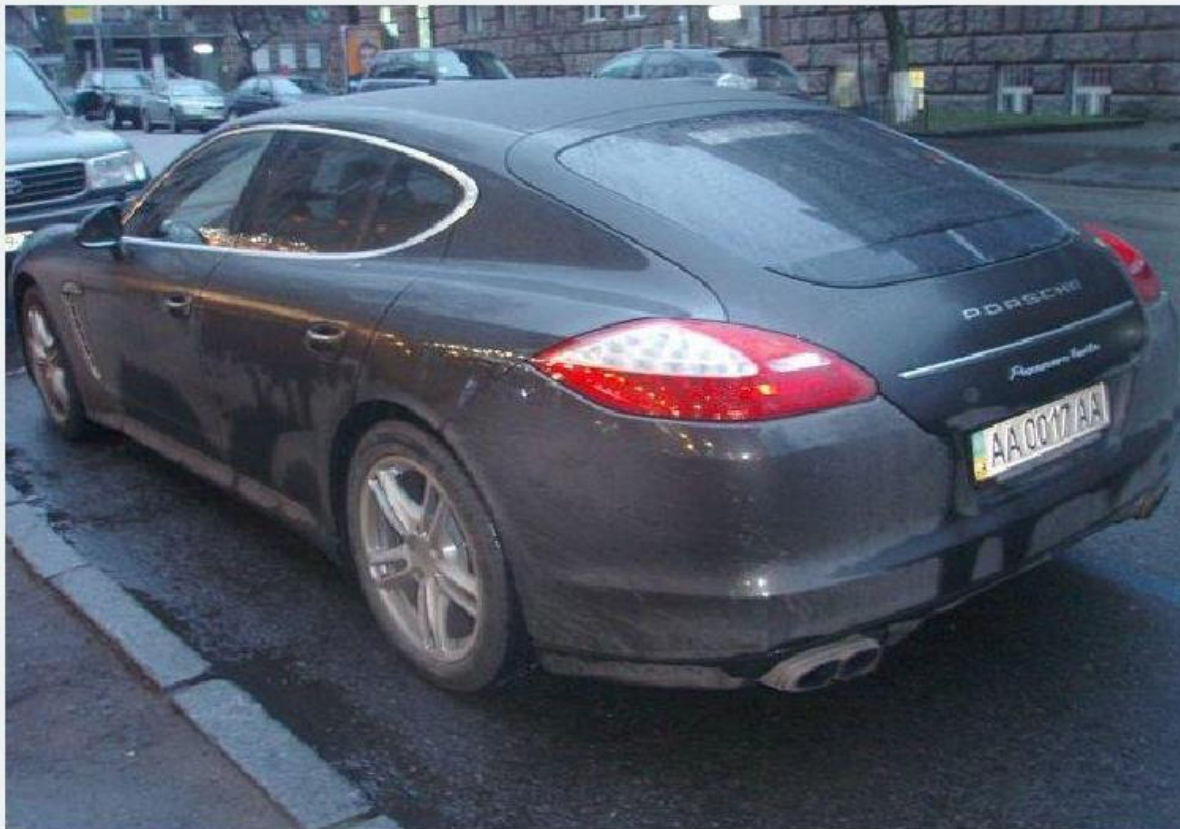
3.14 – Результати розпізнання після натиску кнопки «Explore»

Протестуємо детектування на зображенні з сильним викривленням номерного знака. У декількох випадках програма видає не зовсім точні результати, наприклад замінює один символ на інший. Звідси маємо неможливість подальшої ідентифікації знаку, у зв'язку з викривленням номеру. Один із випадків розпізнавання викривленого зображення зображено на рисунку 3.15.

Upload image:

Upload image

Make a photo



Explore

Number: AA 0017 AA

Info: г.Киев - 2004

Рисунок 3.15 – Детектування символів на викривленому зображенні

На зображеннях з низькою роздільною здатністю алгоритм працює досить непогано, якщо зображення не містить значних дефектів або викривлень (рисунок 3.16). Але зображення з великою кількістю дефектів майже неможливо розпізнати коректно.

Upload image:

Upload image

Make a photo



Explore

Number: A4 5005 MC

Number is not found!

Рисунок 3.16 – Детектування символів на зображенні з низькою роздільною здатністю

Якщо завантажене зображення не містить автомобільного номеру, або даного номеру не має у базі застосунку, то програма видає результат «Number not found!», зображений на рисунку 3.17.

Upload image:

Upload image

Make a photo



Explore

Number: CV 142 AD

Number is not found!

Рисунок 3.17 – Зображення з номерним знаком, відсутнього у базі даних застосунку

3.4.2 Мобільне тестування

Тестування проводиться на трьох мобільних пристроях з операційною системою Android.

Перший мобільний пристрій, що використовується у тестуванні має такі характеристики:

- операційна система Android 7.0 NRD90M;
- обсяг оперативної пам'яті 3,00 ГБ;
- роздільна здатність екрану 1920x1080 (Full HD).

Другий Android-пристрій:

- операційна система Android 7.1.2 N2G47H;
- обсяг оперативної пам'яті 2,00 ГБ;
- роздільна здатність екрану 1280x720.

Пристрій з мінімально дозволеною версією ОС:

- операційна система Android 4.0.4;
- обсяг оперативної пам'яті 1,00 ГБ;
- роздільна здатність екрану 1280x720.

З проведеного тестування можна зробити висновки, що швидкодія напряму залежить від обсягу оперативної пам'яті. Від роздільної здатності екрану залежить лише якість відображення завантажених зображень. Мобільний додаток найбільш ефективно працює на першому пристрої. Це впливає із-за деякої несумісності більш нової версії операційної системи Android для другого пристрою, та для третього за рахунок застарілої ОС та малого обсягу оперативної пам'яті.

4 ОХОРОНА ПРАЦІ

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Відповідно до [9] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання

поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до [10], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання
- якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної

(роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом [11]:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення ятерів;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує година, протягом якого людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (4.1)$$

де U_ϕ - номінальна фазна напруга мережі, В;

Z_Π - повний опір петлі, створене фазними і нульовими дротами, Ом;

Z_T - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [4]: $Z_T / 3 = 0,1 \text{ Ом}$.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_{\Pi} = \sqrt{R_{\Pi}^2 + X_{\Pi}^2}, \quad (5.2)$$

де $R_{\Pi} = R_{\phi} + R_0$ - сумарний активний опір фазного R_{ϕ} і нульового R_0 дротів, Ом;

X_{Π} - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту $S = 2.5 \text{ мм}$, тоді згідно таблицям 5 і 6 [12], має такий опір:

$$X_{\Pi} = 0,11 \text{ Ом};$$

$$R_{\phi} = 7,55 \text{ Ом};$$

$$R_0 = 7,55 \text{ Ом}.$$

$$\text{Отже, } R_{\Pi} = 7,55 + 7,55 = 15,1 \text{ Ом}.$$

Тоді по формулі(4.2) знаходимо повний опір петлі :

$$Z_{\Pi} = \sqrt{15,1^2 + 0,11^2} \approx 15,1(\text{Ом}).$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (4.3)$$

де I_n - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму I_n , А.

Коефіцієнт кратності нелінійного струму I_n розраховується по формулі(4.4.) :

$$I_n = P / U, \quad (4.4)$$

де $P = 220$ Вт - споживана потужність;

$U = 220$ В - робоча напруга;

$k = 3$ А - для плавких вставок.

Отже, $I_n = 220 / 220 = 1$ А.

Підставивши значення у вираз(4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування (і захист) при підвищенні номінального струму.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [13], СНіП, відповідними ГОСТами і ОСТами з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами. Мікроклімат виробничих приміщень [9] визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [9] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу (категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й устанавленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створенню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в СНіП (30 м³ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи (струменеві, лазерні). Рекомендується

принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання (під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплону).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання: обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту [14].

Стандарт [14] пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання(електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [10].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [7] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(E_n) при загальному освітленні дорівнює 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки $a=12$ м, шириною $b=10$ м, заввишки $z=4$ м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м²;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використання освітлювальної установки = 0,6;

M - кількість ламп у світильнику = 2.

З формули(4.5) виразимо n (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.6)$$

Отже, $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$.

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

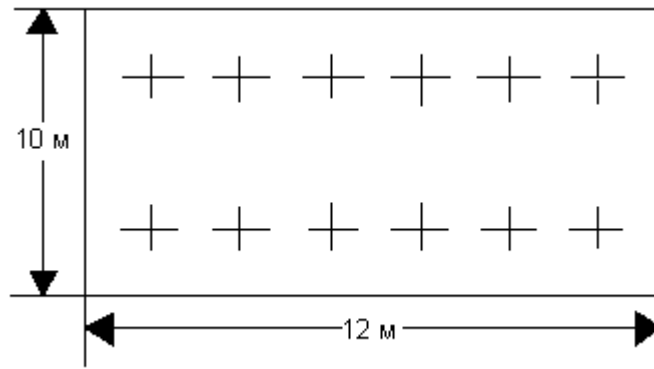


Рисунок 4.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 З ;

- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 З, температура самозаймання 530 З, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;

– стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

– пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;

– деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 З, схильна до самозаймання [12].

Згідно [17] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегрів від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є: оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акропеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В (жовта).

Пожежна безпека об'єктів народного господарства регламентується [18] і забезпечується системами запобігання пожежам і протипожежному

захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізолюване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потокова) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;

- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- організація автоматичного контролю параметрів, що визначають джерела запалення;
- ліквідація можливості самозаймання речовин і матеріалів.

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог:

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до [19] приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для

ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар(сокири, ломи, багри, шерстяну або азбестову ковдри) [20]. Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

ВИСНОВКИ

В атестаційній роботі були проведені дослідження щодо актуальності, доцільності та практичності систем розпізнання образів, а саме номерних знаків на зображеннях. Був проведений аналіз технологій для реалізації методу детектування даних із зображення, а також технологій для проектування кросплатформових мобільних застосунків.

Зокрема було проведений аналіз переваг та недоліків найпопулярніших бібліотек та методів обробки та розпізнавання зображень. В результаті були зроблені висновки, щодо можливостей використання вказаних технологій для рішення різних типів задач.

Мобільний додаток був розроблений за допомогою сервісів компанії Microsoft. Для програмної реалізації було застосовано середовище розробки Microsoft Visual Studio з використанням Xamarin Framework. Всі компоненти змогли вирішити всі поставлені задачі та реалізувати програму на практиці. Xamarin Framework виявився найкращим інструментом для розробки кросплатформових мобільних додатків. Мова програмування C# разом з бібліотекою оптичного розпізнавання символів Emgu CV утворюють гнучку платформу для реалізації методів детектування знаків на зображенні. Очевидно, що система детектування та ідентифікації автомобілів по їх реєстраційному номерному знаку дуже актуально у наш час і системи даного напрямку мають різні сфери застосування. У зв'язку з глобальною комп'ютеризацією системи розпізнавання будуть актуальні і надалі, завдяки їх можливості автоматизування багатьох процесів.

У ході виконання атестаційної роботи був розроблений мобільний додаток і реалізований метод розпізнавання номерних знаків на зображеннях. Вихідний продукт призначений для оптимізації та зручності пошуку інформації, орієнтований на використання у вигляді мобільного додатку. Був проведений аналіз результатів та зроблені висновки щодо доцільності та актуальності теми.

Під час четвертого розділу було проаналізовано умови праці, виявлені причини травматизму і захворювань, можливі небезпечні й шкідливі виробничі фактори. Також було проведено ряд розрахунків щодо виконання вимог охорони праці в приміщенні відділу програмного забезпечення. Дотримання цих вимог є важливим для збереження працездатності та здоров'я працівників.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) C# 4.0: полное руководство. [Текст] / : Пер. с англ. — М.: ООО "И.Д. Вильямс", 2011. — 1056 с.
 - 2) Бондарчук, А.С. Интеллектуальные робототехнические та мехатронні системи: Учебный посібник [Текст] / В.И. Сирямкіна. – Томск, 2017. – 256 с.
 - 3) Фисенко, В.Г. Компьютерная обработка и распознавание изображений. [Текст] — СПб.:, 2008.
 - 4) Анисимов Б.В., Курганов В.Д., Злобин В.К. Распознавание и цифровая обработка изображений. [Текст] — 1983 .
 - 5) Ту Дж., Гонсалес Р. Принципы распознавания образов. [Текст] — М.: Мир, 1978.
 - 6) Васильев, В.И. Распознающие системы. [Текст] — К.: Наукова думка, 1983.
 - 7) Шелехин, Андрей. Подробно о Xamarin [Электронный ресурс] / А. Шелехин. – Электрон. дані – 2013. – Режим доступу: <https://habr.com/post/188130>.
 - 8) Xamarin Documentation [Электронный ресурс] – Электрон. дані – Режим доступу: <https://docs.microsoft.com/en-us/xamarin/#pivot=platforms&panel=Android>.
- Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. Постанова N 42 від 01.12.99. Режим доступу: [www. URL:](http://www.url) <https://zakon.rada.gov.ua/rada/show/va042282-99>
- Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПН 3.3.2.007-98. Затверджено Постановою Головного державного санітарного лікаря України 10 грудня 1998 р. N 7. Режим доступу: [www. URL:](http://www.url) <https://zakon.rada.gov.ua/rada/show/v0007282-98>

НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів». Наказ від 09.01.98 №4. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0093-98](http://www.zakon.rada.gov.ua/laws/show/z0093-98)

9) ГОСТ 12.1.044-89 «Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения». Постанова від 12.12.1989 № 3683. Режим доступу: [www. URL: http://online.budstandart.com/ru/catalog/doc-page?id_doc=51048](http://online.budstandart.com/ru/catalog/doc-page?id_doc=51048)

10) ДСП 173-96 «Державні санітарні правила планування та забудови населених пунктів». Наказ від 19.06.1996 №173. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0379-96](http://www.zakon.rada.gov.ua/laws/show/z0379-96)

11) TCO'07 Certified Displays. © 2007 Copyright TCO Development AB. Режим доступу: [www. URL: https://tcocertified.com/files/2015/11/TCO-Certified-Displays-7.0.pdf](https://tcocertified.com/files/2015/11/TCO-Certified-Displays-7.0.pdf)

12) ДБН В.2.5-28:2018 «Природне і штучне освітлення». Режим доступу: [www. URL: http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf](http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf)

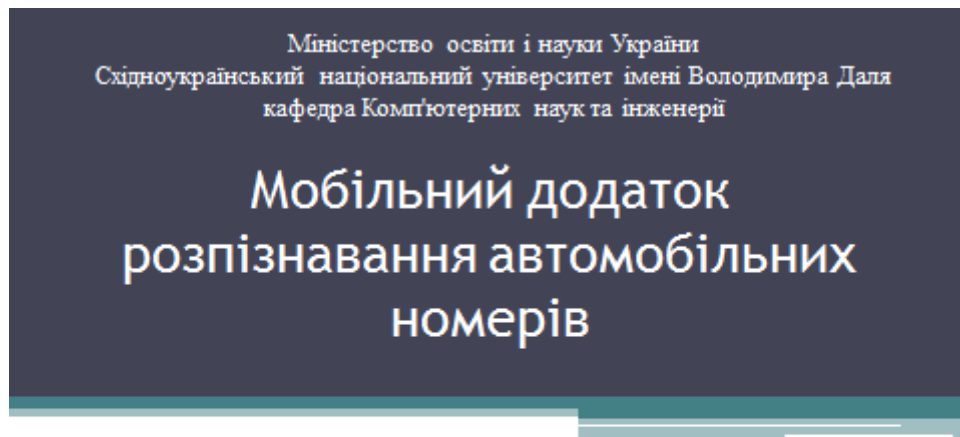
13) ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». Наказ від 15.06.2016 №158. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0158858-16](https://zakon.rada.gov.ua/rada/show/v0158858-16)

14) ГОСТ 12.1.004-91 «Система стандартов безопасности труда. Пожарная безопасность. Общие требования». Режим доступу: [www. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=48679](http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=48679)

15) НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні». Наказ від 30.12.2014 №1417. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0252-15](https://zakon.rada.gov.ua/laws/show/z0252-15)

16) НАПБ Б.01.008-2018 «Про затвердження правил експлуатації та типових норм належності вогнегасників». Наказ від 15.01.2018 №25. Режим доступу: [www. URL: http://search.ligazakon.ua/1_doc2.nsf/link1/RE31677.html](http://search.ligazakon.ua/1_doc2.nsf/link1/RE31677.html)

ДОДАТОК А. ЕЛЕКТРОННІ ПЛАКАТИ



Керівник
проф. Рязанцев О.І.

Виконав
ст.гр. КІ-153
Капран О.Ю.

Актуальність проблеми

- Сьогодні важливим аспектом безпеки дорожнього руху та контролю є ідентифікація автомобілів по їх реєстраційному номерному знаку. Системи розпізнавання номерних знаків мають різні сфери застосування, такі як автотранспортні підприємства, контроль в'їзду на територію підприємства і переміщення транспортних засобів на об'єктах з обмеженим доступом, заправні станції, контроль швидкості руху, автомобільні стоянки.
- Актуальність завдання розпізнавання номерних знаків транспортних засобів полягає в тому, що з кожним днем збільшується потреба автоматизації контролю в'їзду на територію підприємств, контролю швидкості руху і визначення порушення правил дорожнього руху.

Мета роботи

- Мета роботи дати користувачеві можливість розпізнавати зображення у мобільному додатку, та отримувати інформацію про регіон автомобілю без додаткового пошуку. Застосунок розроблений для операційної системи Android за допомогою технології Xamarin. Користувач зможе завантажувати зображення, отримувати автомобільний номер та додаткову інформацію.

Процес розпізнавання



На першому етапі використовується зображення, отримане з камери спостереження. Для початку необхідно виділити область, що містить зображення номерного знаку. Оскільки співвідношення сторін автомобільного номерного знаку регламентовано ГОСТ, можна визначити шукану область.

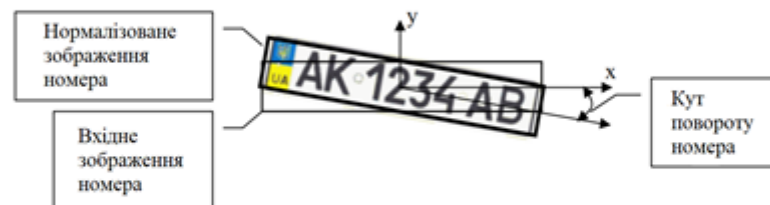
Процес розпізнавання



Після виділення необхідної нам області зображення можна приступати власне до розпізнавання тексту, що міститься в ній. Ефективним способом перекладу зображень в текстові дані є оптичне розпізнавання символів (OCR).

Існує вільний у використанні програмний модуль Tesseract OCR, що виконує функцію розпізнавання тексту. Дана програма в даний час є кращою відкритою бібліотекою для розпізнавання символів, має гарну швидкість роботи і добре документована. Бібліотека Tesseract підтримує українську мову тексту і працює під управлінням операційних систем Windows і Linux.

Програмна реалізація



Запропонований алгоритм нормалізації і сегментації символів дозволяє використовувати його в системах розпізнавання автомобільних номерів. Нормалізація зображення номерного знаку проводиться у два етапи.

На першому етапі визначається кут повороту номера в площині зображення. На другому - виконується алгоритм отримання нормалізованого зображення номера з вихідного зображення з урахуванням кута його повороту.

Програмна реалізація



Інтерфейс застосунку



Висновки

- Мобільний застосунок був розроблений за допомогою сервісів компанії Microsoft. Для програмної реалізації було застосовано середовище розробки Microsoft Visual Studio з використанням Xamarin Framework.
- Xamarin Framework виявився найкращим інструментом для розробки кросплатформових мобільних додатків.
- Мова програмування C# разом з бібліотекою оптичного розпізнавання символів Emgu CV утворюють гнучку платформу для реалізації методів детектування знаків на зображенні.
- Очевидно, що система детектування та ідентифікації автомобілів по їх реєстраційному номерному знаку дуже актуально у наш час і системи даного напрямку мають різні сфери застосування.

Дякую за увагу!