

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Баданін Роман Владленович

**Модель і програмно-апаратні засоби системи
автоматичного розпізнавання автомобільних номерів
для "розумного" міста / Model and software-hardware of
the automatic recognition system of car numbers for the
"smart" city**

спеціальність:123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Випускна кваліфікаційна робота

Виконав студент групи КІм-21
Р.В. Баданін

Науковий керівник:
к.т.н., Г. М. Мельник

ТЕРНОПІЛЬ - 2019

РЕЗЮМЕ

Випускна кваліфікаційна робота на тему «Модель і програмно–апаратні засоби системи автоматичного розпізнавання автомобільних номерів для “розумного” міста» освітнього ступеня магістр зі спеціальності 123 «Комп’ютерна інженерія» написана обсягом 92 сторінок і містить 42 ілюстрацій, 7 таблиць, 3 додатки та 51 джерел за переліком посилань.

Метою роботи є розроблення алгоритму виділення символів номерного знака автомобіля на цифровому зображенні на основі моделі потенціалів.

Методи дослідження включають методи: теорію алгоритмів і систем, методи аналізу зображень, методи об’єктно–орієнтованого програмування та базуються на використанні методів гістограмного аналізу зображень, методів комп’ютерного зору, положень теорії алгоритмів і аналітичної геометрії.

Розроблено алгоритм пошуку та розпізнавання зображень на основі уже існуючої моделі потенціалів, що дозволило збільшити шанс розпізнавання автомобільного номеру.

Здійснено реалізацію програмної системи розпізнавання автомобільних номерів із застосуванням динамічної нетипізованої мови Python та бібліотеки алгоритмів комп’ютерного зору OpenCV.

Можливими напрямками подальших досліджень є продовження робіт по аналізу розпаралелення даних та процесів під час розпізнавання зображення.

КЛЮЧОВІ СЛОВА: НОРМАЛІЗАЦІЯ, КОНТУР, СЕГМЕНТАЦІЯ.

RESUME

Master's thesis on the topic «Model and software–hardware of the automatic recognition system of car numbers for the “smart” city» from the specialty 123 «Computer engineering». Thesis contains 92 pages, 42 figures, 7 tables, 3 appendixes and 51 references sources.

The purpose of the work is to develop an algorithm for numbers recognition from automobile number plate, based on potential model.

Methods of research include methods: the theory of algorithms and systems, image analysis methods, methods of object–oriented programming and based on usage of image analysis, computer optical character recognition and algorithm theory.

Developed algorithm of search and number plate recognition from images based on potential model, what gives us a higher chance of number recognition.

Software implementation of the number plate recognition system with the use of dynamic non–typed Python language and OpenCV computer vision algorithms library is implemented.

Possible ways for further research are the analysis of parallelization data and process while image is got recognized.

KEYWORDS: NORMALIZATION, CONTOUR, SEGMENTATION.

ЗМІСТ

Вступ.....	7
1 Поняття «розумного» міста і його систем.....	10
1.1 Основні ознаки «розумного» міста.....	10
1.2 Аналіз основних систем розумного міста.....	13
1.3 Аналіз алгоритмів систем «розумного» міста.....	20
1.4 Апаратні засоби та алгоритми розпізнавання зображень.....	27
1.5 Постановка задач магістреської роботи.....	30
2 Апаратні засоби та моделі розпізнавання автомобільних номерів.....	31
2.1 Модель розпізнавання зображень.....	31
2.2 Модель січних площин.....	33
2.3 Модель потенціалів.....	41
2.4 Модель Хаара.....	48
3 Програмно-апаратна реалізація системи автоматизованого розпізнавання автомобільних номерів.....	51
3.1 Програмна структура системи автоматизованого розпізнавання автомобільних номерів.....	51
3.2 Апаратна структура системи автоматизованого розпізнавання автомобільних номерів для.....	57
3.3 Реалізація системи автоматизованого розпізнавання автомобільних номерів для «Розумного» міста.....	64
3.4 Експериментальні дослідження ефективності системи автоматизованого розпізнавання автомобільних номерів для «Розумного» міста.....	70
Висновки.....	73
Список використаних джерел.....	74
Додаток А Лістинг коду системи розпізнавання автомобільних номерів.....	79
Додаток Б Світлокопії виданих публікацій.....	85
Додаток В Довідка про використання.....	91

ВСТУП

Актуальність теми. В сучасному світі майже у кожної людини є автомобіль, вони використовуються у багатьох галузях людської діяльності. На даний час автомобіль це вже давно не розкіш, а необхідність. Адже на даний момент майже всі жителі нашої планети мають доступ до автотранспорту, адже це дуже комфортно, з'являються компанії які надають автомобілі в оренду на деякий час, всюди поширений міський, міжміський транспорт. Стрімке поширення автомобілів необхідно контролювати, для цього використовують номерні автомобільні знаки, вони є унікальними і дозволяють ідентифікувати автомобіль та його власника в разі необхідності. В Україні існують дванадцять типів номерних знаків, всі вони мають українські літери, деякі з них навіть латинські та певний числовий набір. Для того, щоб оптимізувати процес пошуку та порівняння автомобільних номерів, створюють системи автоматизованого розпізнавання автомобільних номерів, які за допомогою алгоритмів роботи з комп'ютерним зором мають змогу розпізнати необхідний об'єкт на цифровому зображенні. Ці технології тільки розвиваються на даний момент, вдосконалюються вже існуючі алгоритми та розробляються нові, для більш ефективного розпізнавання номерного знаку автомобіля на цифровому зображенні. Випускна кваліфікаційна робота оформлена відповідно до [1,2]

Мета і завдання дослідження. Метою роботи є розроблення алгоритмів виділення символів номерного знака автомобіля на цифровому зображенні на основі моделі потенціалів. Спочатку використовується серія прийомів маніпулювання зображеннями для виявлення, нормалізації та покращення зображення номерного знака, а потім оптичного розпізнавання символів (OCR) для вилучення буквено-цифрових знаків номерного знака. Системи ANPR, як правило, розгортаються в одному з двох основних підходів: один дозволяє весь процес виконуватись у місці розташування смуги руху в режимі реального часу, а інший передає всі зображення з багатьох смуг до віддаленого комп'ютера та виконує процес OCR там в якийсь пізній момент часу.

Об'єкт дослідження – процеси обробки зображень автомобільних номерів.

Предмет дослідження – алгоритми розпізнавання автомобільних номерів.

Програмний аспект системи працює на стандартному апаратному забезпеченні домашнього комп'ютера і може бути пов'язаний з іншими програмами або базами даних.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- здійснити огляд систем розпізнавання автомобільних номерів;
- проаналізувати моделі і алгоритми розпізнавання;
- обрати алгоритм який відповідає поставленим задачам;
- обрати модель розпізнавання зображення;
- розробити фізичну модель реалізації;
- програмно реалізувати алгоритм розпізнавання автомобільних знаків;
- провести порівняння алгоритмів детекції номерних знаків.

Методи досліджень базуються на використанні методів гістограмного аналізу зображень, методів комп'ютерного зору, положень теорії алгоритмів і аналітичної геометрії.

Наукова новизна одержаних результатів. Розроблено алгоритми розпізнавання автомобільних номерів для систем автоматичного розпізнавання на основі моделі потенціалів, що підвищило імовірність розпізнавання номерних знаків та швидкість розпізнавання.

Практичне значення отриманих результатів. Розроблено систему автоматизованого розпізнавання автомобільних номерних знаків на основі моделі потенціалів, об'єктну модель програмного засобу та фізичну модель системи. Практичне значення розробленої системи:

- управління секцією для вимірювання середньої швидкості руху автомобіля на більшій відстані;
- прикордонний перехід;
- автомобільні репозиції;
- АЗС для реєстрації, коли автомобіліст їде далеко, не платячи за пальне;

– системи управління дорожнім рухом, які визначають потік руху, використовуючи час, який транспортним засобам потрібно пройти два майданчики;

– аналіз поведінки подорожей (вибір маршруту, місце початку походження тощо) для цілей планування транспорту;

– проїжджайте через розпізнавання клієнтів, щоб автоматично розпізнавати клієнтів на основі їх номерних знаків та пропонувати їм предмети, які вони замовляли в останній раз, коли вони користувалися послугою;

– допомога системам управління відвідувачами в розпізнаванні автомобілів для відвідувачів;

– поліція та допоміжна поліція;

– компанії, що займаються парковкою автомобілів;

– підняти або опустити автоматичні кронштейни;

– готелі;

– забезпечення переходу законів щодо екстрених транспортних засобів;

– автоматизовані випробування на викиди.

Публікації та апробація результатів. Отримані результати апробовані в межах II науково–практичної конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі Тернопільського національного економічного університету та опубліковано дві тези доповіді по темі роботи [3, 4]:

Впровадження практичних результатів планується на підприємстві «ФОП Степаненко».

В першому розділі магістерської роботи розкриваються та описуються основні поняття, функції та ознаки. Проводиться аналіз основних систем та алгоритмів.

В другому розділі описані існуючі апаратні засоби та моделі які використовуються для вирішення поставленого завдання.

Третій розділ являє собою практичну реалізацію системи автоматизованого розпізнавання автомобільних номерів для "розумного" міста.

1 ПОНЯТТЯ «РОЗУМНОГО» МІСТА І ЙОГО СИСТЕМ

1.1 Основні ознаки «розумного» міста

Розумне місто – це концепція, в основі якої лежить місто, що використовує різноманітні інформаційні технології задля більш ефективного функціонування та відповідності потребам його жителів.

Ідея такого міста полягає в тому, щоб завдяки збору інформації в режимі реального часу усі ресурси міста можна було використовувати більш продуктивно. Це дозволяє економити кошти, раціональніше діяти та надавати сервіс вищого гатунку – тобто поліпшувати рівень життя населення.

Щоб побудувати місто, яке буде максимально комфортним для його жителів, необхідно створити певні умови для цього. Основною рушійною силою у побудові розумного міста є збір та обробка великої кількості даних (Big Data). Саме управління даними дозволяє муніципальним службам підвищувати якість життя населення.

Дані охоплюють такі сфери життя жителів міста як: безпека, транспорт, медичні послуги, комунальне господарство, благоустрій тощо. Джерелами даних служать відеокамери, різні датчики, сенсори, інформаційні системи, які впроваджуються у повсякденне життя. Найбільш розвинутими і «розумними», на сьогодні вважаються Барселона, Амстердам, Лондон, Нью-Йорк [5].

Ключові ознаки розумного міста сказані нижче.

Інтелектуальні системи управління дорожнім рухом – передбачають підвищення безпеки та ефективності транспортного процесу, комфортності для водіїв і користувачів транспорту. Приміром, у багатьох країнах світу, і в Україні зокрема, встановлюється відеонагляд на автошляхах, що дозволяє контролювати порушення правил дорожнього руху. Крім того, важливу роль у цій концепції відіграє інформація про стан доріг, завантаженість парковок у місті, інформування пасажирів про час прибуття громадського транспорту, зміни у напрямках руху тощо. Ця інформація дозволяє економити власний час та правильно

розпоряджатися ним.

Розумний підхід до вуличного освітлення – особливої популярності набувають датчики руху, які вмикають світло лише коли фіксують певні рухи чи присутність людини, і вимикають його, коли ви, наприклад залишаєте приміщення. Так само працює і вуличне освітлення. Крім того значним попитом користуються LED-лампи (Light Emitting Diode – світлотехнічні вироби для побутового, промислового та вуличного освітлення, у яких джерелом світла є світлодіоди), оскільки вони зменшують використання електроенергії до 80% у порівнянні зі звичайними лампами накаливання, до яких ми звикли.

Залучення жителів міста до управління – це є чи не найвагомішою складовою тут є електронний уряд та управління завдяки впливу на місцеву владу. У всьому світі ці напрями вже давно набули розмаху і чимало громадян спілкується з керівництвом муніципалітетів за допомогою електронних звернень. В Україні за останні роки цей напрямок також почав активно розвиватися. Найбільшим попитом на сьогодні у нас користуються електронні петиції на сайті президента, електронна система держзакупівель «ProZorro» і волонтерський проект iGov, де зібрано усі можливі засоби отримання інформації від держави в онлайні [6].

Розумний будинок – ця технологія передбачає використання системи високотехнологічних пристроїв в оселі для найбільш комфортного проживання людей. Зокрема виділяють кілька основних напрямків впровадження технологій розумного будинку: безпека (датчики руху, присутності, вібрації, розбиття скла, відкриття вікна або двері, відеоспостереження, електронні замки і модулі управління воротами, сирени), управління освітленням (розумні вимикачі, модулі управління шторами та ролетами, контролери для управління світлодіодними світильниками, датчики руху і присутності), управління кліматом (датчики вологості і температури, термостати для підтримки постійної температури або її автоматичного регулювання, терморегулятори для управління потужністю засобів опалення, гігостати для підтримки постійної вологості або її регулювання).

Впровадження міської мережі Wi-Fi – на сьогодні у більшості європейських міст доступ до мережі Wi-Fi є нормою. Україна також прямує до цього і вже почала

надавати бездротовий доступ до інтернету у громадському транспорті (метро, трамваї, тролейбуси), у центральних кварталах міст, публічних місцях тощо. А от у Барселоні вже важко знайти куточок, де Wi-Fi не працює.

Розумний громадський транспорт – передбачає контроль за усім, що відбувається в салоні та зовні протягом руху і здатен передавати інформацію про порушення в органи влади. Слід зауважити, що така технологія уже розроблена в Києві: у розумному автобусі працює Wi-Fi, встановлено камери внутрішнього та зовнішнього відеоспостереження. Водій зможе оцінювати ситуацію за допомогою датчиків: наприклад, якщо загорілося червоне світло – це означає, що в салоні сталася надзвичайна подія. Також є можливість використовувати геолокацію: це дозволить відслідковувати рух транспорту та в разі потреби – швидко з'ясувати, де знаходиться автобус [7].

Сповіщення про надзвичайні ситуації (НС) – персональна мережа оповіщення, що відправляє звичайні SMS-повідомлення на телефони абонентів, які знаходяться в зоні НС, надзвичайно ефективно працює у багатьох країнах. Таким чином вдається запобігти збільшенню кількості жертв, а також нападам паніки, яка у таких ситуаціях нерідко завдають багато шкоди.

Кнопки екстреного реагування – такі засоби допомагають правоохоронним органам швидше зреагувати на певні і події і прибути на місце оперативніше. Така практика має місце у США і показує позитивні результати.

Використання сонячних батарей – у багатьох містах світу, де це дозволяє клімат, така практика є невід'ємною складовою розумного міста. На дахах будівель встановлюються сонячні батареї, які можуть забезпечити автономне електропостачання для окремих квартир чи будинків загалом, в залежності від їх площі.

Безготівкові платежі – на сьогоднішній день у розвинених країнах паперовими грошима майже не користуються – їм на заміну прийшли банківські платіжні картки. Але і це не є остаточним рішенням. У концепції розумного міста передбачається повсякчасне використання мобільних платежів, щоб жителям не доводилося носити з собою великі суми чи безліч карток, встановлюються

спеціальні прилади, які можуть проводити оплати за допомогою мобільного телефону та спеціального застосунку до нього [8].

1.2 Аналіз основних систем розумного міста

Автоматизовані системи керування дорожнім рухом є однією з основних систем розумного міста. Вони виконують наступні функції:

- керування транспортними потоками;
- забезпечення транспортною інформацією;
- управління в особливих ситуаціях та керування безпекою.

Такі системи можуть бути представлені як сукупність пристроїв дорожньої тематики, контролерів та автоматизованих робочих місць, підключених до мережі. Такі структури мають ієрархічну будову [9].

На нижньому рівні дорожні контролери кожного з перехресть забезпечують керування світлофорами всіх напрямків та смуг руху. До контролерів можуть бути під'єднані додаткові інформаційні табло, детектори транспорту, табло пішоходів. Контролери перехресть працюють або за власною програмою керування, локально, або отримують програми з верхнього рівня керування. У більшості малих та середніх міст локальний режим керування дорожнім рухом є основним.

Для забезпечення режиму "зелена хвиля" дорожні контролери перехресть під'єднуються до зонального контролера, програма якого розраховує керуючі програми кожного з контролерів, перехрестя яких підключені до цього режиму. Зональні контролери можуть отримувати всю інформацію, що надходить на дорожні контролери, а також можуть коригувати програми керування за інформацією з верхнього, центральноміського рівня [10].

Міський центр керування забезпечує в основному контролюючу функцію та реалізує регулюючу функцію лише у випадках збоїв в керування дорожнім рухом або для забезпечення проїзду спеціального транспорту.

Розумне вуличне освітлення забезпечує економію енергії та підвищення безпеки дорожнього руху.

Складність транспортної інфраструктури вимагає застосування різних типів освітлення, це необхідно для забезпечення безпечного руху транспорту, при цьому люди відчуватимуть себе в безпеці, а міста економлять на витратах. Розумне світлодіодне вуличне освітлення – економічний і стабільний вибір для міст сьогодні – і в майбутньому. Світлодіодні вуличні світлодіодні системи підключені до датчиків руху є інтелектуальними і універсальними, тому ви можете просто і ефективно управляти, підтримувати і контролювати всю систему.

Видобуток альтернативної енергії, а саме використання сонячних батарей є однією з вагомих ознак розумного міста [11].

Найбільша перевага використання сонячної енергії – її доступність, бо навіть взимку на вулиці сяють сонячні промені, які зможуть виробити хоча б мінімальну кількість необхідної енергії. До того ж, дійсно великий термін експлуатації, внаслідок чого – бездоганна окупність. Якщо розділити вартість встановлення фотоелементів на кількість років, які вони можуть простояти, економія стає відчутною. Автономна сонячна батарея може виробляти достатню кількість енергії, аби задовольнити побутові потреби. Крім того, батарея забезпечує повну незалежність енергосистеми у майбутньому. Також встановлення колекторів не потребує ретельного догляду: лише періодична чистка. До того ж, не потрібно платити за рахунками, а вкладені гроші окупляться приблизно через 5 років. Сонячні панелі є легко масштабованими.

Безготівкові розрахунки – розрахунки, що проводяться без участі готівки, тобто в сфері безготівкового грошового обороту. Організація безготівкових розрахунків повинна відповідати конкретним вимогам, які обумовлені інтересами розвитку економіки.

Головна з них – забезпечувати своєчасне отримання кожним підприємством грошових коштів за поставлену ним продукцію та надані послуги, чим сприяти прискоренню обігу оборотних коштів у розрахунках [12].

У сучасних умовах застосовуються такі основні форми безготівкових

розрахунків:

- платіжними дорученнями;
- платіжними вимогами–дорученнями;
- чеками;
- акредитивами;
- векселями;
- платіжними вимогами;
- інкасовими дорученнями (розпорядженнями).

Набір систем розумного будинку забезпечують захист будинку від вторгнення злодіїв, але це не єдина причина. Сучасні системи безпеки можуть не тільки завадити проникненню незваних гостей в будинок а я здійснювати контроль за іншими частинами будинку. Вони можуть попереджати мешканців про різні небезпеки швидше і більш ефективно, ніж старі моделі [13].

Причини встановлення вдома систем безпеки:

– захист від впливу небезпечних речовин – деякі системи, якщо виявляють потенційну небезпеку за допомогою таких пристроїв як димові датчики, датчики чадного газу або датчики затоплення, навіть відправляють оповіщення на смартфон господарів. Це означає, що у них з'являється можливість запобігти потенційно дорогому катастрофу, перш ніж ситуація вийде з–під контролю;

– гарантія порядку – охоронні системи дозволяють переглядати кадри вашого будинку, поки ви знаходитесь далеко і знати напевно, що все в порядку. Тепер ви зможете використовувати свою домашню систему безпеки, щоб протягом свого робочого дня віддалено наглядати за літнім родичем або дивитися, як нова няня поводить з вашою дитиною. Використовуючи відеоспостереження, ви ніколи не будете сумніватися. За допомогою монітору стаціонарного ПК або екрану смартфона ви будете знати все напевно;

– догляд за дітьми – багато систем безпеки для будинків видають звуковий сигнал, якщо відчиняються двері або вікно. Якщо у вас є маленькі діти або підлітки, які можуть вислизнути непоміченими, це відмінна функція. Якщо двері

або вікно відкриваються, ви будете миттєво попереджені і зможете швидко відреагувати, щоб переконатися, чи все в порядку;

– віддалене управління будинком – якщо ви поставите систему безпеки з функціями домашньої автоматизації, ви отримаєте можливість керування будинком віддалено. Особливості домашньої автоматизації включають в себе можливість за допомогою смартфона контролювати освітлення, термостати і побутові прилади. Ви також можете запрограмувати пристрої на функціонування в певний час. Наприклад, можна запрограмувати свою систему освітлення на включення в потрібний час і встановити термостат на потрібну температуру. Можливості безмежні і можуть сприяти економії енергії [14].

Розглянемо алгоритми роботи та схеми системи охорони які застосовуються в «розумних» містах для забезпечення безпеки майна та людей які знаходяться на об'єктах які охороняються.

Засоби виявлення зломисників класифікують за: принципом дії; за призначенням; за видом зони виявлення; за кількістю зон виявлення; дальністю дії; за конструктивним виконанням [5].

Принципи дії засобів виявлення:

- електроконтактні, омічні;
- ударноконтактні;
- п'єзоелектричні;
- оптично–електронні;
- індуктивні;
- радіохвильові;
- ультразвукові.

Види призначення:

- сповіщувачі для закритих приміщень;
- сповіщувачі для відкритих площ.

Види зон виявлення:

- точкові;

- лінійні;
- поверхневі;
- об'ємні.

Кількість зон виявлення:

- багатозонні сповіщувачі;
- однозонні сповіщувачі.

Класифікація сповіщувачів за дальністю дії:

- сповіщувачі великої дальності – понад 200 метрів;
- сповіщувачі середньої дальності – від 50 до 200 метрів;
- сповіщувачі малої дальності – до 50 метрів.

Поділ за конструктивним виконанням:

- однопозиційні сповіщувачі (випромінювач і приймач поєднані в одному блоці);
- багатопозиційні сповіщувачі (більше двох блоків випромінювачів і приймачів в будь-якій комбінації);
- двохпозиційні сповіщувачі (випромінювач і приймач виконані у вигляді окремих блоків).

Більшість систем охорони працюють за наступним алгоритмом: коли магнітний датчик роз'єднано запускається таймер (60 секунд), якщо пароль до пульта охорони введено правильно, сигналізація вимикається і на пульт охорони приходить повідомлення, що на об'єкті все гаразд, якщо за 60 секунд пароль не введено або введено невірно тричі на пульт охорони приходить повідомлення про те, що на об'єкті знаходяться сторонні [15].

Перевага даного алгоритму роботи полягає в швидкому сповіщенні охоронних органів які приходять на пульт керування системи хорони. Від цього також залежить швидкодія охоронних органів адже вони отримують сповіщення в реальному часі та можуть одразу ж відреагувати на повідомлення з об'єкту який охороняється.

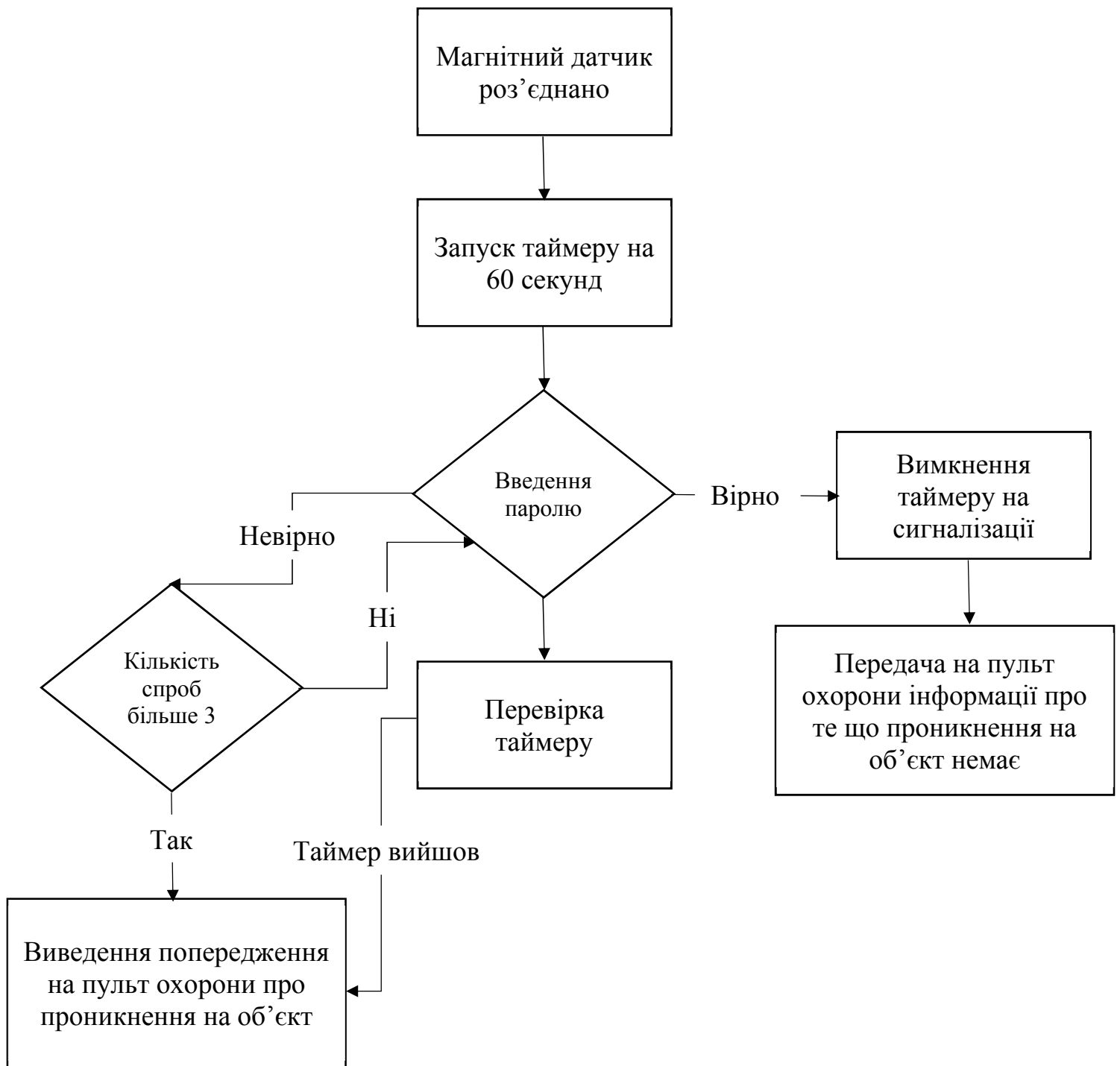


Рисунок 1.1 – Процес роботи системи охорони

Система керування дорожнім рухом працюють на основі ієрархічної систем зображеної на рисунку 1.2 [16].

Розвиток сучасної ієрархічної структури автоматичної системи керування дорожнім рухом відбувався поступово – від нижнього рівня локального керування вручну до комп'ютеризованих зональних і централізованих систем, тому за своїм

складом, архітектурою, функціональними можливостями, способом перепрограмування на дорогах сьогодні використовують АСКДР декількох поколінь, які умовно поділяють на чотири за рівнем розрахунку керуючих параметрів і введення їх до дорожніх контролерів [17].

Перше покоління – розрахунок керуючих параметрів і введення їх до дорожніх контролерів, а пізніше і до зональних контролерів АСКДР, виконуються вручну.

Друге покоління – розрахунок керуючих параметрів автоматизований на комп'ютерах зональних контролерів, проте введення їх до дорожніх контролерів виконується вручну.

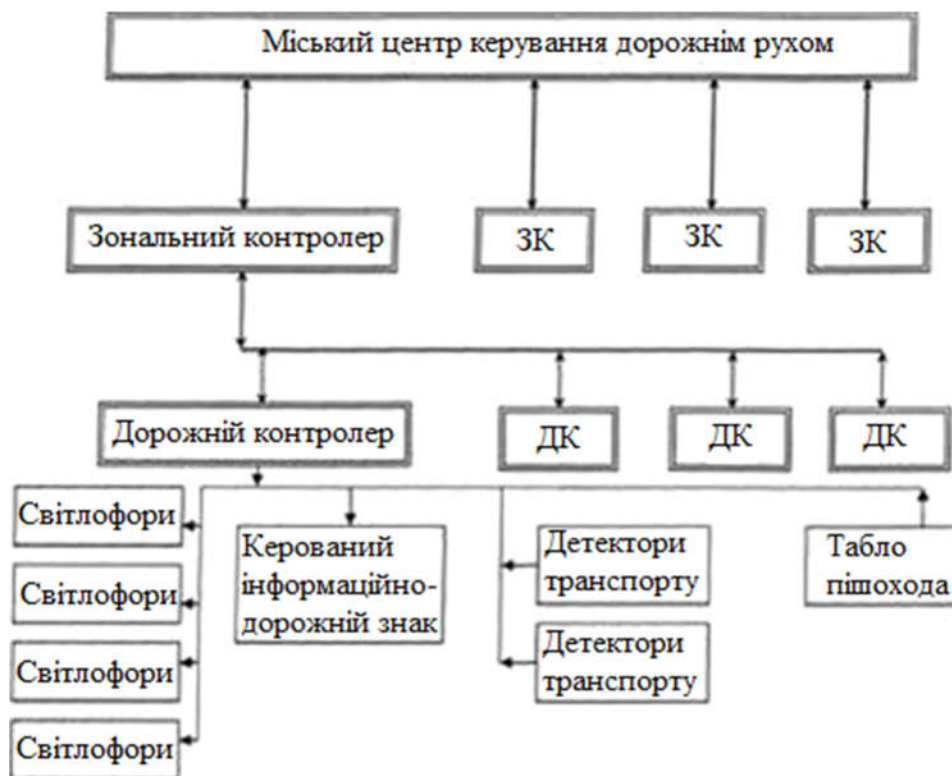


Рисунок 1.2 – Ієрархічна система керування дорожнім рухом

Третє покоління – розрахунок керуючих параметрів і введення їх до контролерів АСКДР автоматизовані, також можлива реалізація керування з прогнозом динаміки транспортних потоків.

Четверте покоління – керування дорожнім рухом автоматичне у реальному

часі, коли за допомогою детекторів транспорту забезпечується збір інформації на контролери, а адаптивні керуючі програми перемикають світлофори перехресть, залежно від реального стану транспортних і пішохідних потоків [18].

1.3 Аналіз алгоритмів систем «розумного» міста

«Розумне місто» допомагає з організацією транспорту, управлінням міськими комунальними службами, а також керує безпекою. Комп'ютери та датчики «розумного міста» включають і вимикають світлофори, аналізують "прямий ефір" з камер над дорогами і вулицями, сигналізують про прориви труб і затори на магістралях.

З великою силою, як відомо, приходить велика відповідальність. І, само собою, великі ризики. За прогнозами PwC, ринок технологій для "розумних міст" в найближчі 20 років добереться до позначки майже в два трильйони доларів США. Але взаємопов'язаність всього – як реальної, так і віртуальної інфраструктури – через Інтернет відкриває дорогу і нові небезпеки. Аналітики відзначають, що Smart City схильні до загроз з боку всього арсеналу кіберзлочинців – починаючи від "традиційного" шкідливого програмного забезпечення (ПО), malware і DDoS-атак, і закінчуючи таємним проникненням з метою розкрадання даних або втручання в роботу інформаційних систем.

«Також кіберзлочинці цілком можуть використовувати в своїх атаках елементи IT-систем Smart City. Досить згадати, що в одній з найбільших DDoS-атак останнього часу було задіяно близько 500 тисяч веб-камер, підключених до мережі Інтернет. У подібних нападах можуть використовуватися і інші підключені до мережі пристрої міських інформаційних систем, наприклад, датчики вимірювання тиску», – пояснює директор департаменту інженерних рішень компанії «IT-Інтегратор» Максим Жданов. Тому, каже він, місто має інвестувати в кіберзахист "розумних" систем, тому що ціна успішної атаки висока. Будь-яка

серйозна атака – і, наприклад, особисті дані мільйонів людей знищуються, або ж витікають і стають загальнодоступними. Або хакери виводять з ладу важливі для роботи міста системи, наприклад, світлофори або водопровід.

Поки ж хакери не робили по–справжньому масштабних атак на міста, але свої можливості вже показали. Українцям це відомо по виведенню з ладу підстанцій на Прикарпатті або під Києвом в останні два роки. В американському Далласі кіберзлочинці просто "пожартували" над містом, включивши в опівночі всі тривожні сирени [19].

У звіті "Захист розумних міст: на шляху до утопії з урахуванням вимог безпеки" компанія Trend Micro Incorporated прогнозує, що серед атакуючих інформаційні системи "розумного" міста будуть бажані «просто» випробувати свої хакерські здібності.

Іншими факторами мотивації будуть крадіжка грошей і особистих даних користувачів, а також корпоративної інформації, шпигунство або "хактивізм" (кібератака з політичною метою).

Протилежний табір – захисників цифрових фортець – теж не дримає і готує захисні і профілактичні заходи.

Експерти пропонують муніципальним органам влади брати на озброєння досвід великих компаній і співпрацювати з профільним бізнесом. Цілком очевидно також, що Smart City будуть немислимі без підрозділів кіберполіції.

Ще одним важливим елементом "іммунітету" цифрового міста називають особливе планування інтелектуальних систем управління. Попросту кажучи, потрібно «не класти всі яйця в одну корзину»: необхідно розподіляти "розумні" елементи Smart City на більше число незалежних одна від одної ділянок, а також створювати дублюючі схеми.

Аналітики компанії Trend Micro Incorporated пропонують кілька важливих кроків, які повинні забезпечити захист "розумних" міст. Йдеться як про регулярний контроль і тестування системи, так і постійну пильність при виборі постачальників і провайдерів послуг. Крім національних центрів реагування на комп'ютерні інциденти (так звані CERT), говорять аналітики, повинні бути і муніципальні

команди такого роду. Вони додають, що менеджмент "розумного міста" повинен постійно пам'ятати про необхідність регулярно оновлювати системи безпеки програмного забезпечення, контролювати фізичну безпеку обладнання, а також ретельно контролювати всі використовувані канали зв'язку [20].

На допомогу приходять математика зі статистикою, втілені в так звані SIEM, Security Information and Event Management (системи збору, аналізу та управління подіями інформаційної безпеки). Сама по собі ця система ще не цифровий щит, але вона збирає інформацію, що надходить від різних інформаційних систем Smart City і здатна виявляти відхилення від нормального функціонування. Як приклад застосування SIEM–системи в концепції Smart City можна розглянути наступний сценарій: підключення різнорідних джерел подій (таких як камери контролю дорожнього руху, системи реєстрації заяв в поліцію, системи реєстрації надходжень до травм пунктів та іншого) дає можливість отримати хорошу аналітику і побудувати автоматизовані ефективні алгоритми реагування, а в перспективі – і зперебачити багато інцидентів [21].

Фахівці з кібербезпеки переконані, що рішення для забезпечення безпеки «розумних» міст не можуть бути точковими. Навпаки, вони повинні бути частиною єдиної стратегії кібербезпеки, які доповнюють і підтримують один одного. Само собою, таке планування включає і розробку детальних регламентів та інструкцій для сервісних служб Smart City про те, що і як потрібно робити на випадок кібератаки.

Однак кібербезпека "розумних міст" не повинна лягати важким тягарем на муніципальні чи державні бюджети – багато питань повинні вирішуватися (і вирішуються!) в партнерстві з бізнесом і державою. У Лос–Анджелесі, наприклад, місцева влада запустила партнерську ініціативу разом з міністерством внутрішньої безпеки і декількома фірмами з різних сфер діяльності. Разом вони створюють інтегрований аналітичний центр, який постійно відстежує загрози, збирає воедино дані з різних джерел і готує аналітику.

Продумана організація цифрової інфраструктури та захисту її елементів спростить і запуск нових проектів. Але при цьому, їх безпека також повинна бути

ретельно спланована, реалізована і протестована. Особливо це важливо, коли мова йде про об'єкти критичної інфраструктури або персональні дані громадян [22].

Блокчейн–технології дають принципово нові можливості для розвитку електронного уряду (eGovernment). Гранична захищеність інформації робить дані, занесені в систему, найвірогідніше будь – якого паперу з підписами і печатками. Смарт–контракти дозволяють автоматизувати операції з даними і задавати алгоритми їх виконання відповідно до нормативних актів. Реалізація таких проектів поки що ускладнюється з огляду на деяких юридичних аспектів, але, на думку експертів, це лише питання часу. Наприклад, в Дубаї планується впровадити блокчейн–технології в державний сектор вже до 2020 році [23].

Smart Governance – це майбутнє суспільних послуг, лідерство громади, мобільна робота та постійне вдосконалення через інновації. Smart Governance використовує технології для полегшення та підтримки кращого планування та прийняття рішень. Мова йде про вдосконалення демократичних процесів і перетворення способів надання державних послуг. Вона включає електронне урядування, програму ефективності та мобільну роботу. В Україні старт розвитку «розумного» управління ще попереду, але вже досить активно впроваджується е–урядування. Можливо, на це вплинуло приєднання до Міжнародної хартії відкритих даних.

Цілями зазначеної хартії є поширення демократії, боротьба з корупцією та сприяння економічному зростанню. Головною метою хартії є покращення та сприяння співпраці під час прийняття та реалізації спільних принципів, стандартів та кращих практик відкритих даних. Державному агентству з питань електронного урядування доручено забезпечити: подання в установленому порядку заяви про приєднання України до Міжнародної хартії відкритих даних; координацію реалізації принципів Міжнародної хартії відкритих даних; розроблення проекту плану дій з реалізації принципів хартії [24].

Тому нині зростають ідеї трансформації робочого середовища. Наприклад, хмарні обчислення, а саме – GovCloud – продукт віртуалізації рішень, розроблений спеціально для державних організацій і установ. GovCloud пропонує новий спосіб

вивчення того, як уряди організують і відкривають нові горизонти спілкування з громадою [25].

Створення адаптованої урядової робочої сили потребує безпрецедентного рівня гнучкості. Урядова робоча сила, заснована «на хмарі», або GovCloud, може включати працівників, які займаються творчою, проблемно–орієнтованою роботою. Замість того, щоб існувати в одному агентстві, ці працівники можуть перебувати «в хмарі». «Хмарні команди» можуть бути спрямовані більш спеціалізованими агенціями, ніж ті, які існують сьогодні. Агенції та «хмарні команди» можуть бути підтримані загальнодержавними спільними службами, які запобігають створенню нових, постійних структур, шляхом надання допомоги в поточній, рутинній роботі. GovCloud повністю включає нові концепції робочого місця, які використовуються в усьому світі, як у приватному, так і в державному секторах – моделі, такі як гнучкі ресурси, спільні послуги та дистанційна робота [26].

Ініціатива GovCloud в деяких країнах відбувається у багатьох країнах на основі федеральних та місцевих законів, правил та загальної стратегії. Тим не менш, головною метою є створення основи та набору керівних принципів для пошуку та розгортання рішення для хмарних обчислень для урядів та їх контрольованих органів. Наприклад, програма GovCloud у США сприяє впровадженню рішень для хмарних обчислень за формальними стандартами та процедурами, причому основна увага приділяється безпеці та сумісності. Вони опублікували кілька керівних принципів у рамках цієї програми, таких як Федеральна стратегія хмарних обчислень, 25–точковий план федерального СІО і дорожня карта NIST Cloud Computing Technology. Крім того, що є специфічним для уряду, GovCloud також пропонується в якості фірмового продукту декількох приватних постачальників хмар, таких як Amazon AWS, що надає аналогічні рішення для державних інститутів [27].

Отже, як ми бачимо, потужні розвинені країни уже зараз сміливо втілюють нові способи управління, постійно постаючи перед новими викликами та небезпеками в даній сфері, що лише стимулює держави до вдосконалення своїх

інформаційно–комунікативних систем.

В наш час існує досить багато систем для ідентифікації номерних знаків, але не всі з них є якісними. Паралельно з розробкою алгоритмів, розробляються і апаратні засоби для розпізнавання номерних знаків.

Для вирішення задачі використовують наступні етапи:

- попередній пошук номера;
- нормалізація номера;
- розпізнавання тексту.

У більшості випадків складність обробки зображень полягає у неоднорідності освітлення та наявності затемнених або навпаки занадто освітлених областей. На етапі попередньої обробки часто застосовують коригування рівня яскравості, контрастності та морфологічні операції.

Бібліотека надає засоби для використання основних морфологічних перетворень: Erode – звуження заданої області з використанням ядра та Dilate – розтягування заданих областей [28].

Також для попередньої обробки використовують:

- дискретизація – поділ на пікселі з обраним кроком;
- зменшення шумів і підвищення яскравості зображення;
- сегментація – розділення зображення на складові частини, що мають спільні властивості за рахунок пошуку однорідних областей або методом виділення контурних ліній;
- отримання ознак зображення – здобуваються за допомогою вирішення задач індексації.

Нормалізація зображення номерного знака проводиться в два етапи. На першому етапі встановлюється кутовий номер в площині образу. На другому – виконується алгоритм отримання нормалізованого зображення номерів з вихідного зображення з врахуванням його нахилу [29].

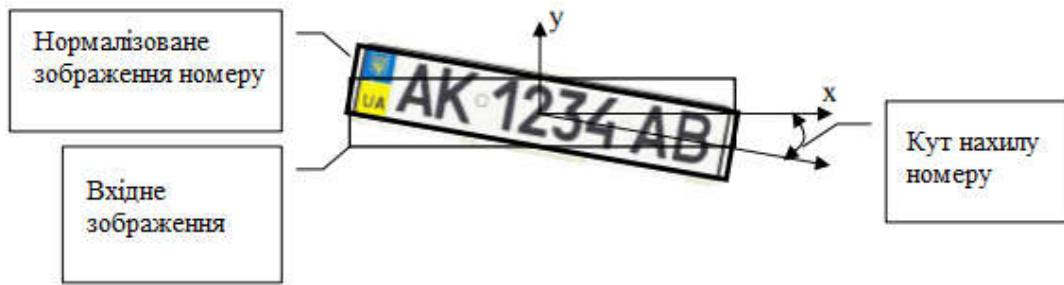


Рисунок 1.3 – Схема отримання нормалізованого зображення

Для повороту області зображення, що відповідає нормалізовано 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 знака 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 [30].

Наведений оператор більш чутливий до напрямів кордонів, близьких до горизонтального, тому дозволяє добре виділити на зображенні верхню і нижню частину номерного знака.

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

Наділяючи прями в просторі коефіцієнтів вагою, відповідним значенням яскравості зображення результату підкреслення меж, і проводячи їх в просторі коефіцієнтів з яскравістю рівною вазі [31].

Розпізнавання тексту з зображення надає можливість вирішення ряду наукових та прикладних задач при ідентифікації об'єктів.

1.4 Апаратні засоби та алгоритми розпізнавання зображень

Система розпізнавання автомобільних номерів складається з "механізму захоплення кадру", який має здатність фіксування зображення, пошука потрібного місця знаходження номера на зображенні та подальшого виділення символів за допомогою засобів оптичного розпізнавання символів (OCR), що переводять пікселі в цифри, які можна прочитати.

У число деяких помилок, з якими бореться нова технологія ANPR, входять розмивання кордонів символів номерного знака в результаті поганого фокусування або руху автомобіля, гострокутні зображення внаслідок поганого кута зору камери та низької контрастності. Нумери можуть вийти розмитими через яскраве світло автомобільних фар або відображень сонячного світла; окрім цього, на чіткість зображення можуть вплинути багато інших чинників.

Хоча в цих додатках часто можна використовувати звичайні камери відеоспостереження, вони зазвичай мають низьку роздільну здатність. Крім того, вони вимагають гарного освітлення, погано працюють вночі при світлі фар, мають досить низьку витримку затвора а, отже, можуть використовуватися лише в разі нерухомих автомобілів [32].

Як основа системи отримання зображень повинні використовуватися спеціалізовані зчитувачі. Камери REG-X засновані на відзначених нагородами технологіях та забезпечують абсолютну продуктивність при фіксуванні номерних знаків. Камери REG-X забезпечують бездоганні зображення як вдень, так і вночі, а технологія відкидання зайвих деталей гарантує надійне зчитування номерного знака. Потужний інфрачервоний масив з довжиною хвилі 850 нм забезпечує роботу REG-X в будь-яких умовах освітлення, включаючи повну темряву та сліпуче світло фар. Об'єктив з фіксованою відстанню 35 мм гарантує ефективне зображення з роздільною здатністю 400 ТВЛ на відстані від 11 до 18 м. REG-X легко інтегрується зі стандартними відеореєстраторами та сумісні зі стандартом IP66 для роботи в суворих зовнішніх умовах.



Рисунок 1.4 – Зчитувач зображень REG–X

Характеристики REG–X:

- швидкість реєстрації: 80 км/г;
- тип матриці: LXR;
- система відеосигналу: CCIR;
- ефективне розширення: 795 x 596;
- вхід відеосигналу: 1В;
- ступінь захисту: IP66;
- матеріал корпусу: алюміній;
- діапазон робочих температур: –30...+55 градусів за Цельсієм;
- вага: 1.2 кг.

Процес роботи системи показаний на наступному рисунку:

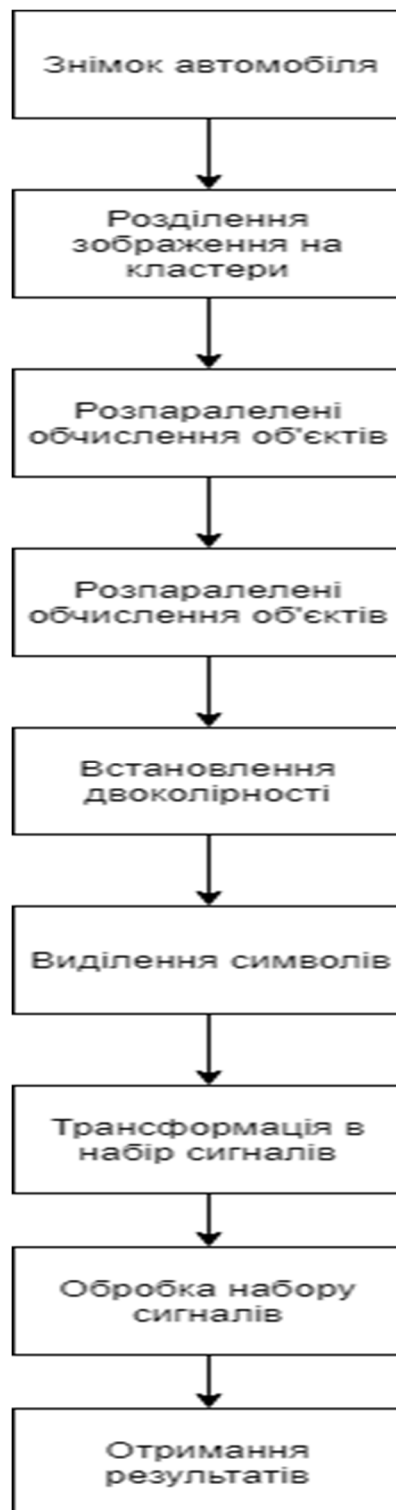


Рисунок 1.5 – Процес роботи системи розпізнавання автомобільних номерів

Опис алгоритму роботи системи: знімок автомобіля; розділення зображення на кластери за допомогою методу k -means, де вхідні дані координати пікселів; проведення розпаралелених обчислень для кожного об'єкту на фото; надання зображенню фіксованих розмірів та надання зображенню двоколірності; далі

небхідно виділити символи з отриманого зображення; трансформація зображення в набір сигналів; передання нейронній мережі набір сигналів на вхід, після обробки мережа поверне символи зображені на номерному знаку автомобіля [33].

1.5 Постановка задач магістреської роботи

Випускна кваліфікаційна робота включає програмну та апаратну реалізацію системи розпізнавання автомобільних номерів.

Мета роботи – програмна та апаратна реалізація автоматичної системи розпізнавання автомобільних номерів тобто автоматично виділяти символи номерного знака автомобіля з зображення з камер відеоспостереження для подальшої обробки системою безпеки. Це допоможе швидше відслідковувати необхідну інформацію та опрацьовувати заявки які виникають в ході керування рухом автотранспорту та в перспективі покращить продуктивність праці.

Об'єктом дослідження є процес реєстрації та обробки автомобільних номерів.

Предметом дослідження є створення продукту для програмного забезпечення та апаратної реалізації автоматичної системи розпізнавання автомобільних номерів, а також систематизації цієї інформації. Методи розробки базуються на основі бібліотеки OpenCV для мови програмування Java, середовище розробки IntelliJ IDEA Community Edition для розробки програмного продукту та Microsoft SQL Management Studio для баз даних [34].

2 АПАРАТНІ ЗАСОБИ ТА МОДЕЛІ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ

2.1 Модель розпізнавання зображень

Теорія розпізнає образи – розділяє кібернетичні, що розвиває теоретичні основи і методи класифікації і ідентифікація показників, явищ, процесів, сигналів, ситуацій і т. п. об'єкти, 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 завітують, і знання правил дорожнього руху дозволяють прийняти правильне рішення про те, можна, чи не можна переходити вулицю в цей момент [35].

У більшості випадків, що призводить до наближення світу, людина здійснює їх класифікацію, тобто розбиває ці явища (предмети, ситуації) на групи схожих, а не тотожних. По тим чи іншим причинам необхідна відмінність в одній групі у чотирьох "подібних" явищах, які можуть при цьому значно відрізняється один від одного.

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.

Основні моделі розпізнавання зображень [36]. Шаблонна модель – перетворюють зображення окремого символу в растрове, порівнюють його зі всіма шаблонами, наявними в базі і вибирають шаблон з найменшою кількістю крапок, відмінних від вхідного зображення. Модель в свою чергу поділяється на:

- 1) кореляційний – розпізнавання накладанням зображень;

2) модель допустимих перетворень об'єкта – масштабування, поворот, деформація, пошкодження. Виконання допустимих перетворень для знаходження максимального співпадіння;

3) просторово – частотна модель – на основі зображення формується спектр а потім порівнюється [37].

Структурні моделі розпізнавання зберігають інформацію не про поточкове написання символу, а про його топологію. Дана модель розділяється на:

- 1) синтаксичний – на основі математичної лінгвістики;
- 2) логічні моделі – які використовують дискретний аналіз.

Ознакова модель базується на тому, що зображенню ставиться у відповідність N -мірний вектор ознак:

1) кластерний аналіз – обрані точки зображення об'єднуються в кластери. Відповідно до цього класу відповідає деяка множина точок в кластері, невелике число граничних точок, будь-які дві внутрішні точки з'єднані між собою, внутрішня точка має тільки точки даної множини;

2) потенціали – об'єкт розглядається як електричний потенціал – в просторі ознак точки класів A і B створюють потенціал певного знаку, клас точки C визначається найбільшим потенціалом в тоці C ;

3) модель опорного словника – виділяється сукупність ознак кожного символу певного слова, потім використовують вирішальні функції які розроблені для кожного символу, ці функції приймають найбільше значення яке найближче відноситься до певної літери;

4) модель зондів – використовується для розпізнавання цифр та літер, навіть якщо вони написані від руки та з деякими відхиленнями у розмірах та стилі написання. Зонд складається з двох електродів, розділених ізоляційним проміжком. Кожному зображенню одного класу (одній літері) відповідає одна комбінація збуджених зондів. Зонд вважається збудженим, якщо він пересікає лінії літери;

Нейронні мережі – один з напрямків наукових досліджень в галузі створення штучного інтелекту (ШІ), в основі якого лежить прагнення імітувати нервову систему людини.

Експертні системи – це методологія адаптації алгоритму успішних рішень однієї сфери науково–практичної діяльності в іншу.

Системи оптичного розпізнавання – технологія цілісного цілеспрямованого адаптивного розпізнавання. Активність – це основна властивість роботи системи розпізнавання по аналогії з живим організмом. Комп'ютер сприймає не тільки те, що прямо спостерігається на зображенні, але і те, що від зображення очікується. Робота такої системи стала можлива завдяки принципам цілісності, цілеспрямованості і використанню контексту. Цілісність допускає, що кожен просторовий об'єкт складається з елементарних частин, зв'язаних між собою визначеними геометричними відношеннями. Цілеспрямованість стверджує, що процес ефективного розпізнавання повинен виконуватися шляхом висунення і подальшої перевірки гіпотез. Адаптивність свідчить, що для надійного розпізнавання різнорідних об'єктів система повинна мати здібність до самонавчання [38].

2.2 Модель січних площин

Алгоритм навчання, заснований на моделі січних площин, полягає у відокремленні образів за допомогою частин гіперплощин. Алгоритм складається з наступних етапів.

Навчання (формування гіперплощин для відокремлення множин):

- 1) проведення січних площин;
- 2) вилучення зайвих гіперплощин;
- 3) вилучення зайвих частин площини.

Розпізнавання нових об'єктів – Припустимо, що потрібно навчити комп'ютер розпізнавати три образи a , b , та c . Необхідно провести січні площини. У ПК вводять коди двох точок, які належать різним образам та проводять довільну пряму, яка їх відокремлює.

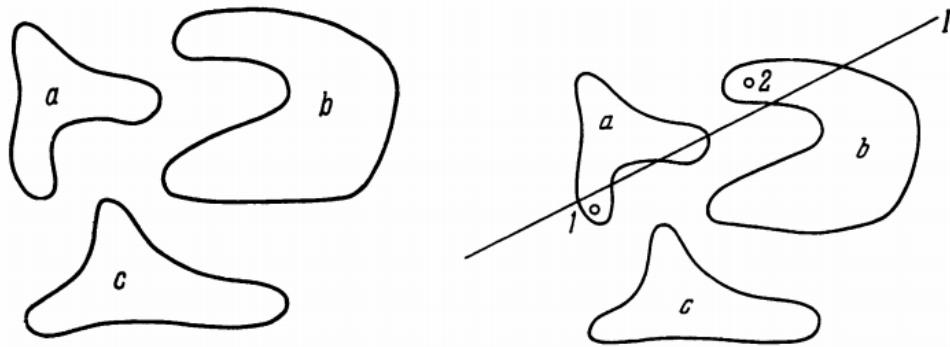


Рисунок 2.1 – Відокремлення січних площин

Потім береться ще один об'єкт на основі якого перевіряється правильність класифікації відносно проведеної прямої.

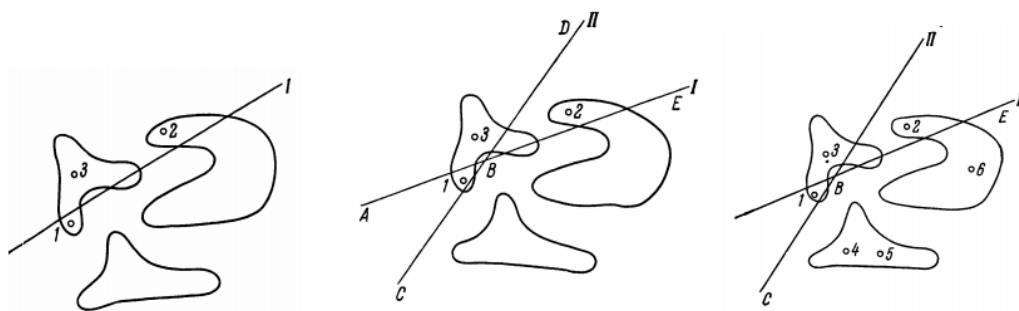


Рисунок 2.2 – Перевірка на основі третього об'єкту

Оскільки третя точка належить тій самій півплощині, що й друга, то приходимо до висновку, що поточний класифікатор працює невірно. Для відокремлення точок 2 та 3 проводиться пряма II. Після цього площина розбивається на 4 частини. Ті частини, у якій лежать об'єкти 1 та 3 (області ABC та ABD) відносимо до класу a, область DBE – до класу b.

При перевірці нової точки можливі три випадки:

- виникає протиріччя (як це було для точки 3);
- протиріччя не виникає, точка потрапляє у свою частину простору;
- протиріччя не виникає, точка потрапляє у вільну (непозначену) частину простору (точки 4 і 5).

У третьому випадку (точки 4 та 5 на рисунку 2.1) машина відносить вільну частину площини до відповідного образу.

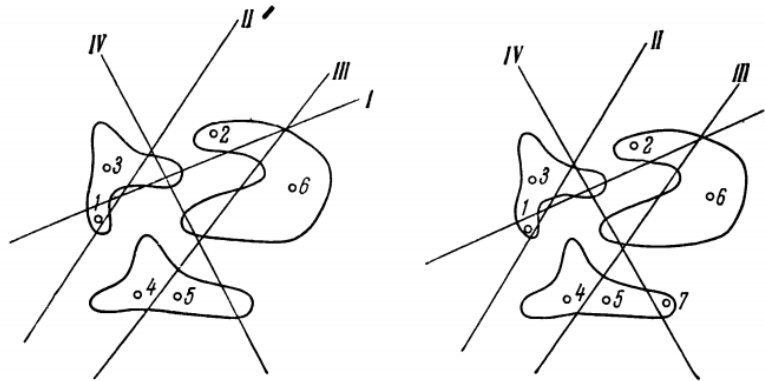


Рисунок 2.3 – Проведення нових прямих для відокремлення

Для відокремлення точки 6 (образ b) від точок 4 та 5 (образ c) потрібно провести дві нові прями (прямі III та IV на рисунку 2.3).

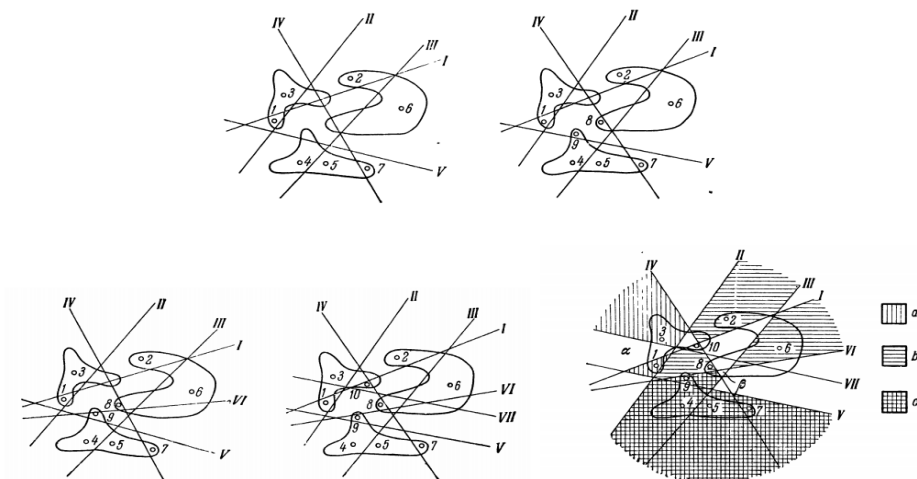


Рисунок 2.4 – Продовження розділення зображення

Подальші кроки вищенаведеного процесу навчання наведені на рисунку 2.4, де вказано проміжкові та кінцевий результат стадії. Незафарбовані ділянки не відносяться розпізнавальною системою до жодного із трьох наведених образів [39].

Видалення зайвих площин. На рисунку 2.4 видно, що гіперплощини (прямі)

I, III та V можуть бути вилученні цілком, оскільки вони не містять частин, відкидання яких призводить до появи протиріч. Наприклад, пряма I використовується лише для відокремлення об'єктів 1 та 2 – представників різних образів а та б. Але ці об'єкти відокремлює також пряма IV (або пряма VI). Розбиття після відкидання зайвих площин наведено на рис. 11. Слід зазначити, що у результаті частка вільних частин площини зменшилися.

Проте не можна вважати, що процес навчання завершений, оскільки ще залишилися "порожні" частини площини. Для їх вилучення використовується гіпотеза компактності. Природнім є віднести "порожні" області до того образу, що і яка-небудь суміжна зафарбована частина. Цей процес називають процесом вилучення зайвих частин. Для деяких вільних частин площини цей процес однозначний (область α на рисунку 2.4), для інших – ні (усі області на рисунку 2.5). Рекомендується проводити це вилучення поступово проглядаючи усі гіперплощини (прямі на рисунку 2.5) та "порожні частини", які до них прилягають [40].

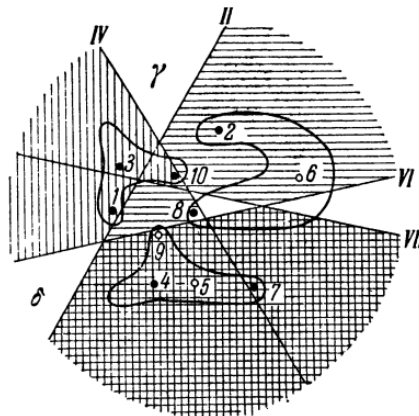


Рисунок 2.5 – Перегляд усіх прямих

Тому спочатку виконуємо перевірку усіх частин площини II, потім усіх частин площини IV і т.д. Кінцевий результат наведений на рисунку 2.6.

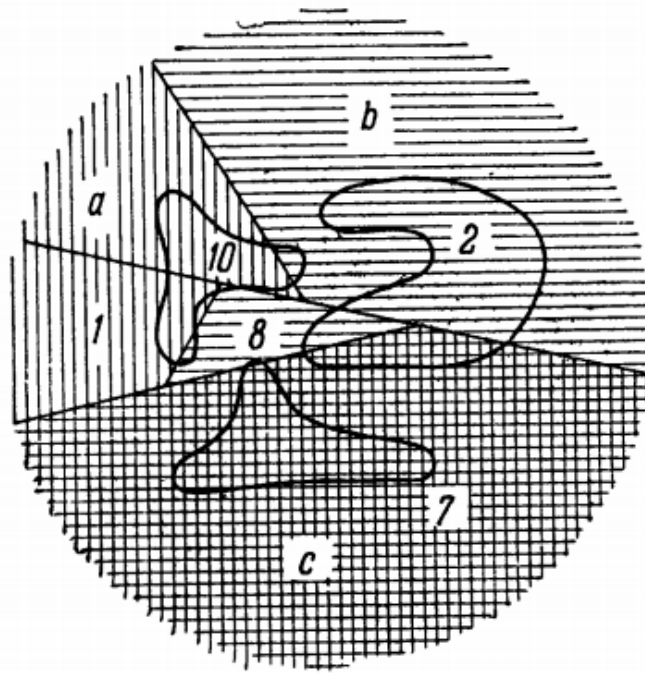


Рисунок 2.6 – Кінцевий результат

Робота алгоритму починається з того, що випадковим чином вибирається n чисел $\lambda \dots \lambda$. Далі для цих точок обчислюємо значення сум.

$$\sigma^{(1)} = \lambda_i x_i^{(1)};$$

$$\sigma^{(2)} = \lambda_i x_i^{(2)}.$$

Після чого вибирається число за допомогою формули:

$$\lambda_{n+1} = (\sigma^{(1)} + \sigma^{(2)})/2,$$

потім виконується рівняння:

$$\lambda_1 X_1 + \dots + \lambda_n X_n - \lambda_{n+1} = 0.$$

Будемо позначати положення точки відносно площини знаком відхилення

$$\lambda(x) = \lambda_1 X_1 + \dots + \lambda_n X_n - \lambda_{n+1}.$$

Далі опис процесу відбувається за допомогою таблиць знаків:

Таблиця 2.1 – Таблиця знаків №1

Таблиця знаків			
Номер точки	Образ	Номер площини	
		1	
		Знак точки	
1	a	0	
2	b	1	

Таблиця 2.2 – Таблиця знаків №2

Таблиця знаків			
Номер точки	Образ	Номер площини	
		1	
		Знак точки	
1	a	0	
2	b	1	
3	a	1	

Таблиця 2.3 – Таблиця знаків №3

Таблиця знаків			
Номер точки	Образ	Номер площини	
		I	II
		Знак точки	
1	a	0	1
2	b	1	0
3	a	1	1

Таблиця 2.4 – Таблиця знаків №4

Таблиця знаків			
Номер точки	Образ	Номер площини	
		I	II
		Знак точки	
1	a	0	1
2	b	1	0
3	a	1	1
4	c	0	0
5	c	0	0
6	b	0	0

Таблиця 2.5 – Таблиця знаків №5

Таблиця знаків					
Номер точки	Образ	Номер площини			
		I	II	III	IV
		Знак точки			
1	a	0	1	1	1
2	b	1	0	1	0
3	a	1	1	1	1
4	c	0	0	1	1
5	c	0	0	0	1
6	b	0	0	0	0

В ході всіх розрахунків кінцевий результат зображений в таблиці 6.

Таблиця 2.6 – Таблиця знаків №6

Таблиця знаків		
		Номер площини

Продовження таблиці 2.6

Номер точки	Образ	I	II	III	IV	V	VI	VII
		Знак точки						
1	a	0	1	1	1	0	1	1
2	b	1	0	1	0	0	1	0
3	a	1	1	1	1	0	1	0
4	c	0	0	1	1	1	0	1
5	c	0	0	0	1	1	0	1
6	b	0	0	0	0	0	1	0
7	c	0	0	0	0	1	0	1
8	b	0	0	1	1	0	1	1
9	c	0	0	1	1	0	0	1
10	a	0	0	1	1	0	1	0

Далі відбувається перевірка можливості видалення стовпців. Виконується перевірка того, чи залишаються унікальними рядки після відкидання стовпчика. Якщо так, то відповідну площину можна відкинути. Результат зображений у таблиці 2.7.

Таблиця 2.7 – Таблиця знаків №7

Таблиця знаків					
Номер точки	Образ	Номер площини			
		II	IV	VI	VII
		Знак точки			
1	a	1	1	1	1
2	b	0	0	1	0
3	a	1	1	1	0
4	c	0	1	0	1
5	c	0	1	0	1

Продовження таблиці 2.7

6	b	0	0	1	0
7	c	0	0	0	1
8	b	0	1	1	1
9	c	0	1	0	1
10	a	0	1	1	0

У цій таблиці є однакові рядки знаків. Зайві точки також можна виключити. Результат наведений у останній таблиці. Розпізнавання нових об'єктів. Обчислюються знаки об'єкту відносно усі гіперплощин і отриманий код порівнюється з усіма рядками таблиці знаків. При співпаданні рядків об'єкт відноситься до відповідного образу [41].

2.3 Модель потенціалів

Точковий електричний заряд у однорідному середовищі створює електричне поле, зображене на рисунку 2.6. Радіальні лінії – це силові лінії поля, концентричні кола – лінії однакового потенціалу.

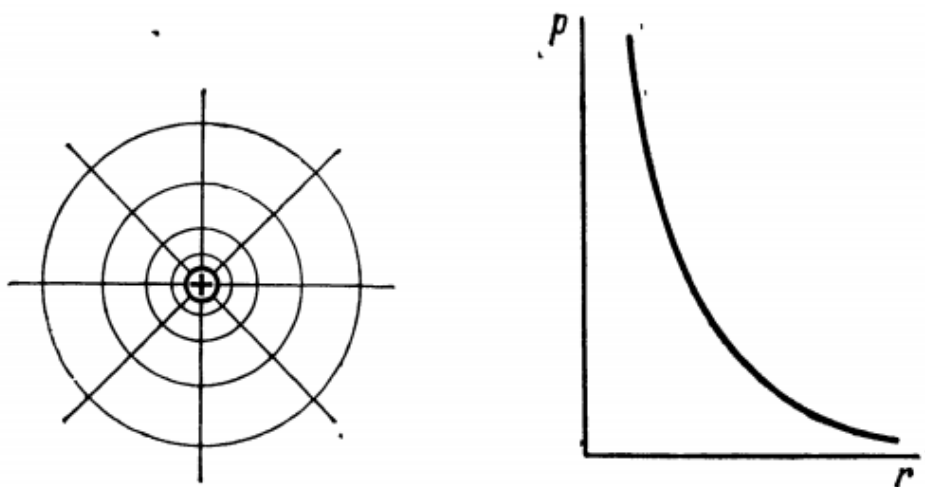


Рисунок 2.6 – Електричне поле

Потенціал p у кожній точці простору визначається співвідношенням

$$p = a \frac{q}{r^2},$$

де a – деяка стала,

q – величина заряду,

r – відстань від заданої точки до заряду.

Крива зміни потенціалу як функції відстані наведена на рисунку 2.6. Потенціал зменшується по мірі віддалення від його джерела. Якщо поле утворене кількома зарядами, то потенціал в кожній точці рівний сумі потенціалів, які створюються кожним із зарядів.

Припустимо, що у просторі розташовані дві компактні групи зарядів. У одній групі – позитивні, у другій – негативні. На рисунку 2.7 показаний розподіл потенціалів у околі цих зарядів, на рисунку 2.8 ці потенціали алгебраїчно просумовані.

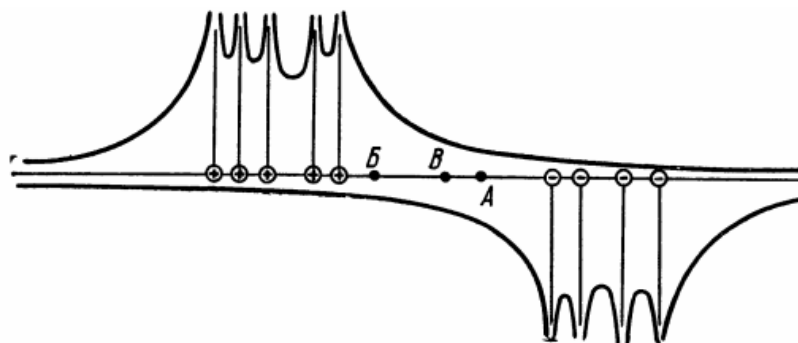


Рисунок 2.7 – Розподіл потенціалів

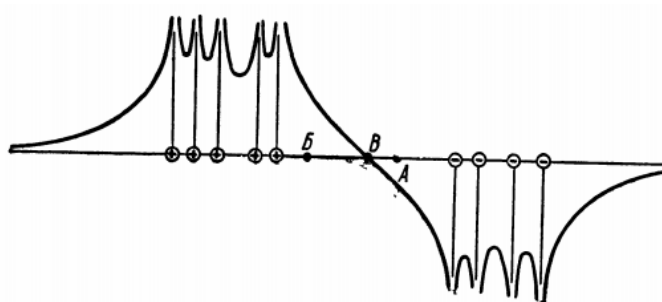


Рисунок 2.8 – Алгебраїчно просумовані потенціали

Точку можна віднести до тієї чи іншої множини точок у залежності від того, який знак має сумарний потенціал поля у цій точці.

Вищенаведені міркування по аналогії можна перенести на точки простору рецепторів. Кожній точці, яка з'являється у процесі навчання, поставимо у відповідність деяку функцію, аналогічну по формі до електричного потенціалу. Такою функцією може бути, наприклад, функція:

$$\varphi(D) = \frac{1}{1+\alpha D^2},$$

де α – деякий коефіцієнт,

D – відстань (у деякій метриці) між точкою–джерелом потенціалу та точкою, у якій обчислюється потенціал.

Наприклад, у якості D можна використовувати евклідову відстань, віддаль Хеммінга, манхеттенську метрику, тощо.

Нехай джерелами потенціалів є група точок, які відповідають деякому образу a . Тоді можна вважати, що середній потенціал, які створюють у деякій точці простору джерела цього образу, характеризує віддаль від цієї точки до усього образу у цілому. Спираючись на гіпотезу компактності, можна запропонувати наступне правило розпізнавання: точку відносимо до того образу, середній потенціал якого у цій точці є максимальним.

Алгоритм розпізнавання на основі моделі:

– у процесі навчання запам'ятовуються координати усіх точок та належність їх до відповідних образів $a \dots a$;

– для точки x , яка підлягає розпізнаванню, обчислюється потенціали кожного образу, тобто суми

$$\phi_i(x) = \frac{1}{n_i} \sum_{a \in a_i} \varphi(x), \quad i=1,2,\dots, m,$$

де m – кількість різних образів, n_i – кількість точок відповідного образу, використаних у процесі навчання,

$$\varphi_a(x) = \frac{1}{1+\alpha D^2(a,x)},$$

потенціал який створює точка a у точці x .

Порівнюються величини $\Phi_1(x)\dots\Phi_m(x)$ і точка x відноситься до того образу, потенціал якого є найбільшим.

У випадку двох образів a_1 та a_2 рішення можна приймати на основі значення знаку функції

$$\Delta\phi(x) = \phi_{a_1}(x) - \phi_{a_2}(x),$$

Застосування вищенаведеного алгоритму для розпізнавання цифр, зображених за допомогою рецепторного поля 6×10 у середньому дало змогу досягнути правильного розпізнавання у 85% випадків за умови, що

$$n_1 = n_2 = \dots = n_{10} = 12,$$

причому подальше збільшення обсягу навчальної вибірки практично не впливає на якість розпізнавання.

Недоліки найпростішого алгоритму виявляються у тому випадку, коли точки навчальної вибірки розподілені нерівномірно. Відповідний приклад наведений на рисунках 2.9 – 2.11.

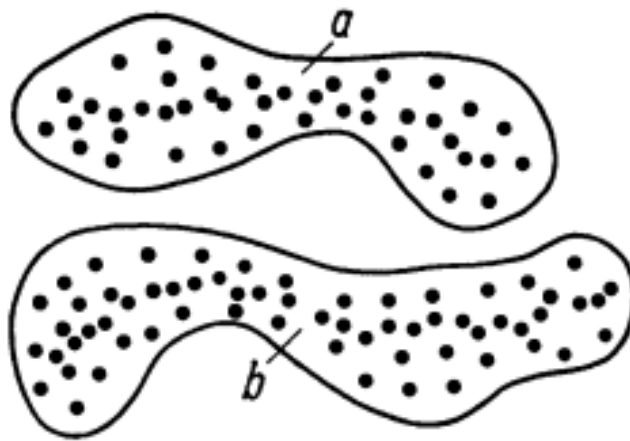


Рисунок 2.9 – Образи

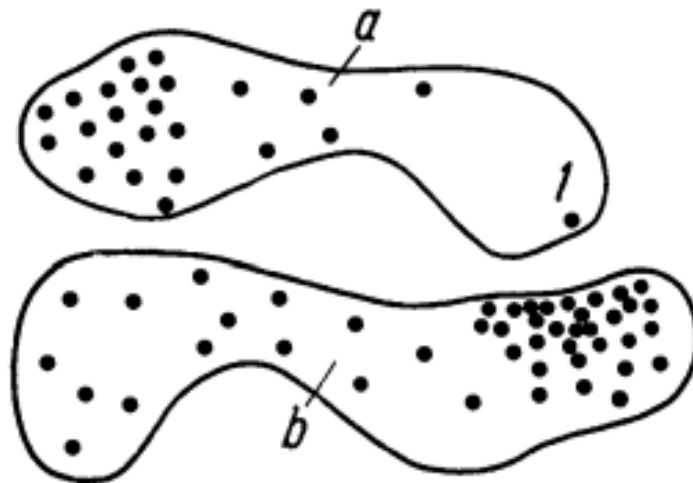


Рисунок 2.10 – Вибірка

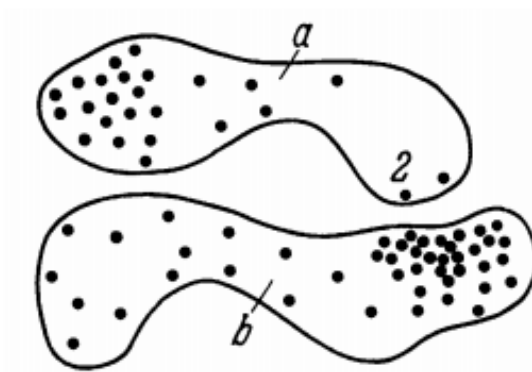


Рисунок 2.11 – Результат

Навчальна вибірка для образів а та b, зображених на рисунку 2.9, наведена на рисунку 2.10. У цьому випадку у зв'язку із «невдалою» навчальною вибіркою, нова точка 2 (рисунок 2.11) буде віднесена до образу b. Більш того, машина може не впізнати навіть відомий їй об'єкт, який зустрічався у процесі навчання, наприклад, точку 1, яка належить до образу а.

Алгоритм розпізнавання може бути покращений, шляхом введення поняття «ваги точки» та доповненням кроку навчання наступною операцією. Після запам'ятовування усіх об'єктів їм присвоюють початкову вагу 1. Далі до елементів навчальної вибірки застосовують крок 2 алгоритму навчання, причому потенціали середні потенціали обчислюють за формулою

$$\phi_i(x) = \frac{1}{n_i} \sum_{a \in a_i} \phi^* w(a), \quad i=1,2,\dots, m,$$

де $w(a)$ – вага точки а.

Якщо при розпізнаванні об'єкту а відбувається помилка, то вагу $w(a)$ відповідного об'єкту збільшують на деяку величину, наприклад на одиницю. Потім застосовується ще один такий самий цикл перевірки та корекції ваг і т.д. Цикли повторюються до тих пір, поки усі відомі фігури не будуть розпізнані правильно.

Застосування покращеного алгоритму до розпізнавання дало змогу підвищити відсоток правильного розпізнавання нових зображень цифр до 89%.

У випадку бінарних зображень традиційні функції відстані Хеммінга або Евкліда не завжди точно відображають відмінність між зображеннями. Відповідний приклад наведений на рис. 20. Шляхом зсуву вертикальної лінії із п'ятірки (рисунок 2.12, а), можна отримати «трохи гіршу» п'ятірку (рисунок 2.12, б) та трійку (рисунок 2.12, в), кожна з яких відрізняється від початкової фігури у шести розрядах. Тобто, елемент «свого» образу та елемент «чужого» образу знаходяться на однаковій відстані від початкової фігури. Схожий приклад наведений на рисунку 2.13 для випадку зображень одиниць та двійок, які відрізняються у десяти розрядах.

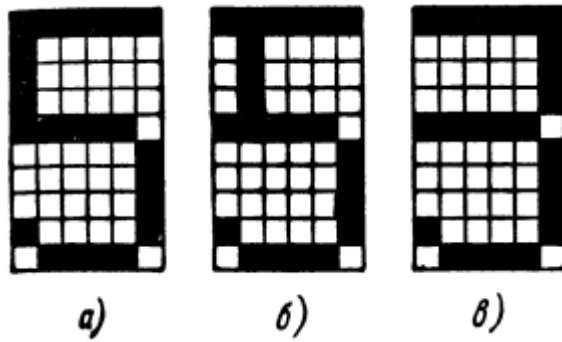


Рисунок 2.12 – Приклад п'ятірки

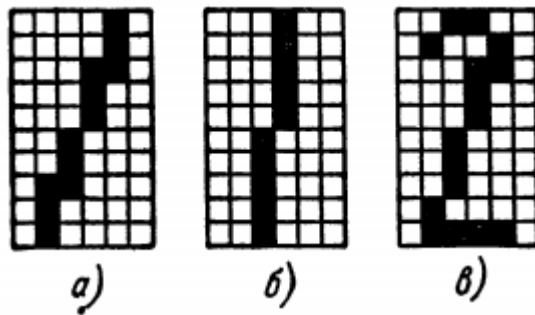


Рисунок 2.13 – Приклад для одиниці та двійки

Ці приклади показують, що значні зміни елементів зображень, які значно спотворюють зображення, приводять тих самих змін у відстані, що і незначні. Тому застосований модель кодування недостатньо добре передає відмінність фігур. Надійність розпізнавання може бути підвищена шляхом удосконалення методу кодування. Поставимо у відповідність кожному рецептору функцію, яка рівна одиниці на цьому рецепторі, спадаючу по усім напрямкам від нього, тобто функцію, аналогічну до потенціалу. Ця функція може бути апроксимована ступінчастими функціями [42].

Застосування удосконаленого методу кодування дозволило збільшити середню надійність розпізнавання букв до 94%.

2.4 Модель Хаара

Ознаки Хаара – ознаки цифрового зображення які використовуються в розпізнаванні образів. Історично склалося так, що алгоритми, які працюють з інтенсивністю зображення наприклад RGB, мають велику складність вирахування. Віола і Джонс адаптували ідею використання вейвлетів Хаара і розробили ряд ознак Хаара. Вони складаються з суміжних прямокутних областей і позиціонуються на зображенні, далі сумується інтенсивність пікселів в конкретних областях, після цього вираховується різниця між сумами. Ця різниця і буде значення певної ознаки та розміру. Розбиття зображення на прямокутники є основою даної моделі.

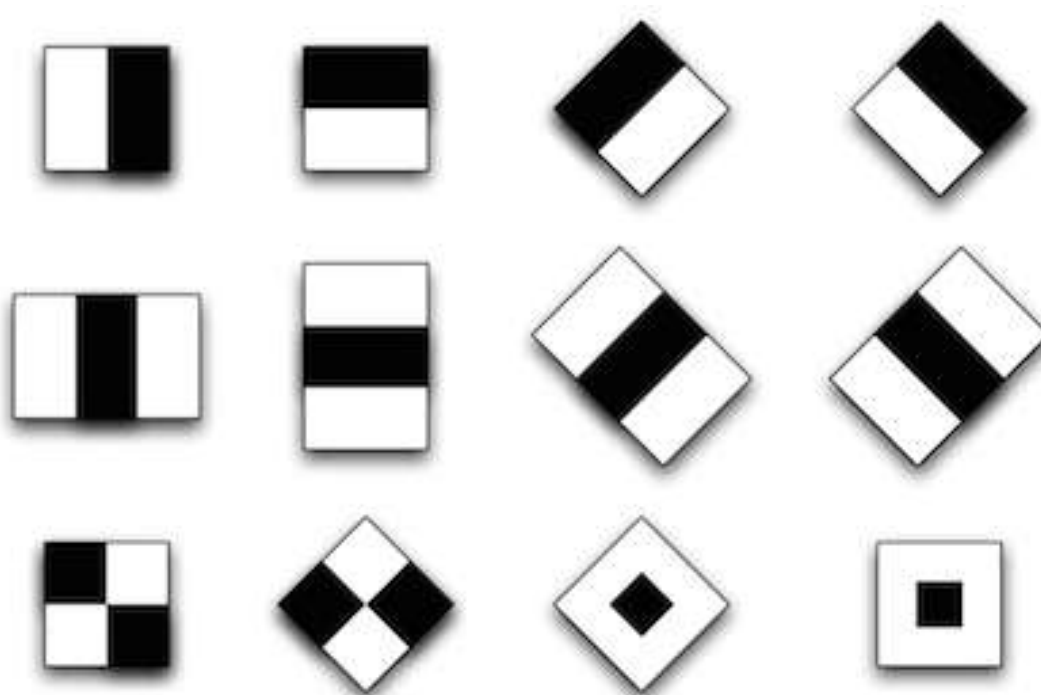


Рисунок 2.14 – Прямокутники в методі Хаара

В оригінальній версії алгоритму Віоли–Джонса використовувалися тільки примітиви без поворотів, а для обчислення значення ознаки сума яскравостей пікселів одній підобласті віднімалася з суми яркостей іншої підобласті.

У розвитку методу були запропоновані примітиви з нахилом на 45 градусів і несиметричних конфігурацій. Також замість обчислення звичайної різниці, було запропоновано приписувати кожній підобласті певну вагу і значення ознаки обчислювати як зважену суму пікселів різнотипних областей.

Чому в основу методу лягли примітиви Хаара? Основною причиною була спроба піти від піксельного уявлення зі збереженням швидкості обчислення ознаки. З значень пари пікселів складно винести якусь осмислену інформацію для класифікації, в той час як з двох ознак Хаара будується, наприклад, перший каскад системи з розпізнавання осіб, який має цілком осмислену інтерпретацію.

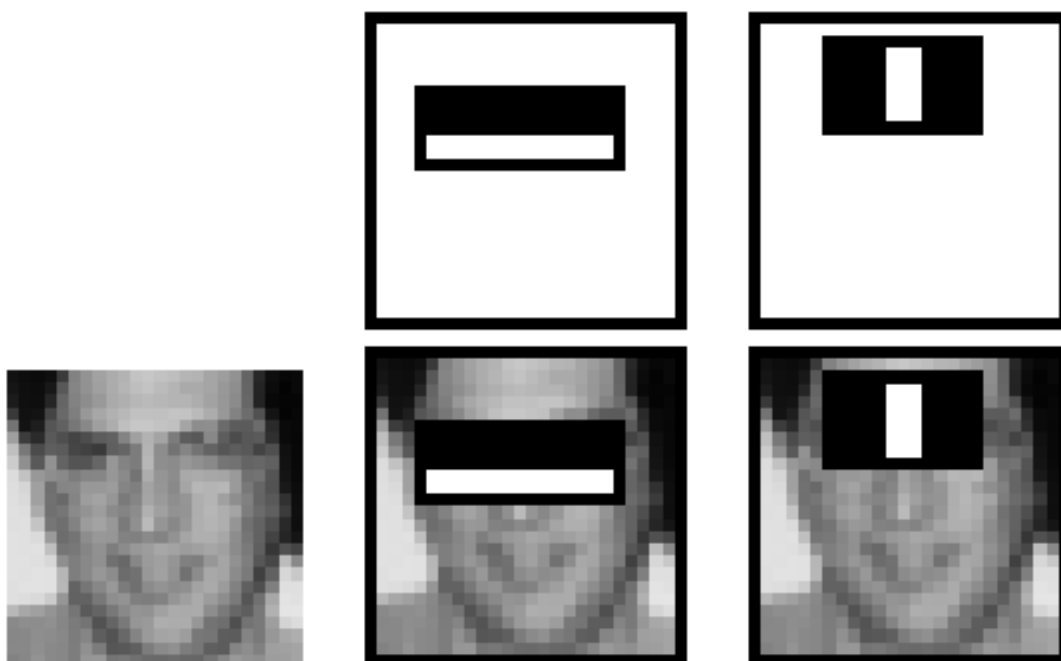


Рисунок 2.15 – Приклад визначення обличчя

Складність обчислення ознаки так само як і отримання значення пікселя залишається $O(1)$: значення кожної підобласті можна обчислити скомбінувавши 4 значення інтегрального уявлення (Summed Area Table – SAT), яке в свою чергу можна побудувати заздалегідь один раз для всього зображення за $O(n)$, де n – число пікселів в зображенні.

Це дозволило створити швидкий алгоритм пошуку об'єктів, який

користується успіхом вже більше десятиліття. Але повернемося до наших ознак. Для визначення приналежності до класу в кожному каскаді, перебуває сума значень слабких класифікаторів цього каскаду. Кожен слабкий класифікатор видає два значення в залежності від того більше або менше заданого порогу значення ознаки, що належить цьому класифікатором. В кінці сума значень слабких класифікаторів порівнюється з порогом каскаду і виносяться рішення знайдений об'єкт даними каскадом чи ні.

По суті всі ці ознаки в якійсь мірі є самими звичайними детекторами кордонів. На основі цього базису будується рішення про те чи каскад розпізнав об'єкт на зображенні чи ні.

Другий за важливістю момент в методі Віола–Джонса – це використання каскадної моделі або виродженого дерева прийняття рішень: в кожному вузлі дерева за допомогою каскаду приймається рішення може на зображенні містяться об'єкт чи ні. Якщо об'єкт не міститься, то алгоритм закінчує свою роботу, якщо він може містяться, то ми переходимо до наступного вузла. Навчання побудовано таким чином, щоб на початкових рівнях з найменшими витратами відкидається більшу частину вікон, в яких не може міститися об'єкт. Коли необхідно розпізнати осіб – перший рівень містить всього 2 слабких класифікатора, у разі розпізнавання автомобільних номерів – 6.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ

3.1 Програмна структура системи автоматизованого розпізнавання автомобільних номерів

В даному розділі дипломної роботи описано детально структуру системи автоматизованого розпізнавання автомобільних номерів для «Розумного» міста. Працівники які будуть використовувати систему для розпізнавання автомобільних номерів будуть запускати його в ручну при необхідності, за допомогою ярлика на робочому столі.

Діаграма варіантів використання системи автоматизованого розпізнавання автомобільних номерів для «Розумного» міста зі сторони користувача який буде працювати з системою.

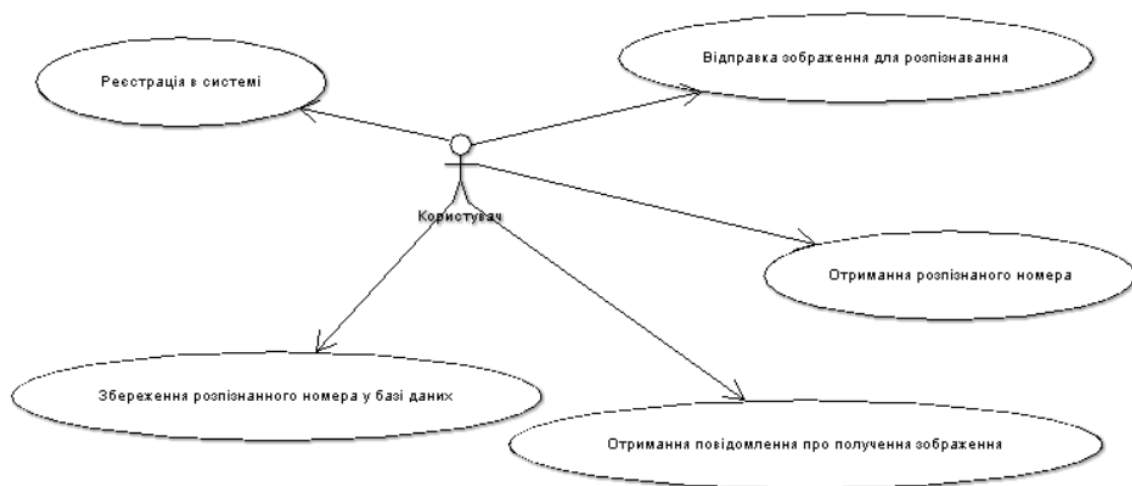


Рисунок 3.1 – Діаграма варіантів використання системи автоматизованого розпізнавання автомобільних номерів для «Розумного» міста

Варіанти використання системи:

– реєстрація в системі – додавання нового користувача в систему;

– відправка зображення для розпізнавання – зображення обране користувачем відправляється алгоритму для розпізнавання автомобільного номерного знаку;

– отримання розпізнаного номера – після відправлення зображення повертається клієнту разом з результатом опрацювання алгоритму;

– збереження розпізнаного номера у базі даних – результат розпізнавання алгоритмом зберігається в базі даних;

– отримання повідомлення про отримання зображення – після обробки зображення виводиться повідомлення про завершення обробки з результатом.

Окрім користувача в модуля розпізнавання також є варіанти використання, зображені на рисунку 3.2.

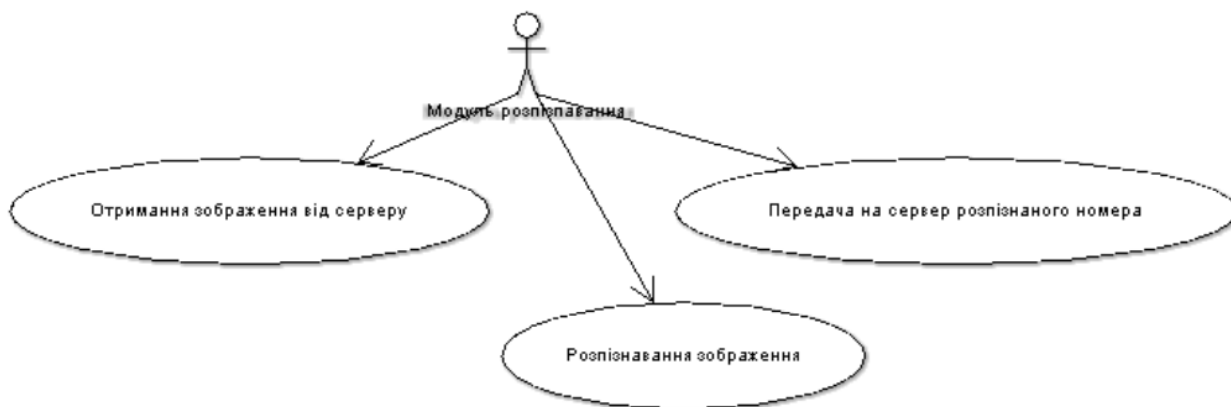


Рисунок 3.2 – Діаграма варіантів роботи модуля розпізнавання автомобільних номерів

Робота модуля розпізнавання автомобільних номерів:

– отримання зображення – прийняття для розпізнавання зображення яке відправив користувач;

– розпізнавання зображення – застосування алгоритму розпізнавання автомобільного номерного знаку для розпізнавання;

– передача на сервер розпізнаного номера – повернення розпізнаного автомобільного номерного знаку.

Для вирішення задач по організації даних використовується модуль обробки.



Рисунок 3.3 – Діаграма варіантів роботи модуля обробки даних

Варіанти використання модуля:

- реєстрація користувача – повернення даних про реєстрацію;
- відправлення повідомлення про отримання зображення – виведення повідомлення користувачу;
- отримання зображення – можливість завантаження зображень користувачем;
- відправлення зображення модулю розпізнавання – після отримання зображення від користувача воно відправляється модулю розпізнавання автомобільних знаків;
- відправка розпізаного номера користувачу – повернення результатів роботи алгоритму;
- отримання розпізаного номера – отримання результатів роботи;
- ведення журналу статистики – додавання результатів розпізнавання зображення в базу даних.

Для розробки системи автоматизованого розпізнавання автомобільних номерів використовувалася бібліотека для роботи з зображеннями OpenCV – основана на мові програмування Java, бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального

призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях. Середовище розробки IntelliJ IDEA Community Edition – для різних мов програмування включаючи Java, для розробки програмного продукту та Microsoft SQL Management Studio для створення баз даних та управління ними. Основною мовою для формування та використання запитів являється мова Transact-SQL яка є реалізацією стандарту ANSI/ISO [43].

В проєкті реалізовані наступні класи:

- WorkSession(робоча сесія);
- DWorkSession(робоча сесія структури);
- UserInfo(інформація про користувача);
- Registrator(реєстрація);
- DBManager(клас для роботи з базою даних);
- CarInfo(інформація про машину);
- BitmapConverter(конвертер);
- RcgEventArgs(клас який реалізує аргументи для розпізнавання);
- NumRecognizer(розпізнавання номерів);
- Recognizer(реалізація функції розпізнавання).

Для відображення класів я використав UML діаграму класів – це статичне представлення структури моделі. Відображає статичні елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

У своєму проєкті я використав агрегацію для поєднання класів, адже вона включає в себе поєднання рівноправних сутностей тобто два класи знаходяться на одному концептуальному рівні і вони всі однаково важливі. Інколи в таких

відношення необхідно моделювати частину і ціле, де ціле має вищий ранг і складається з менших частин.

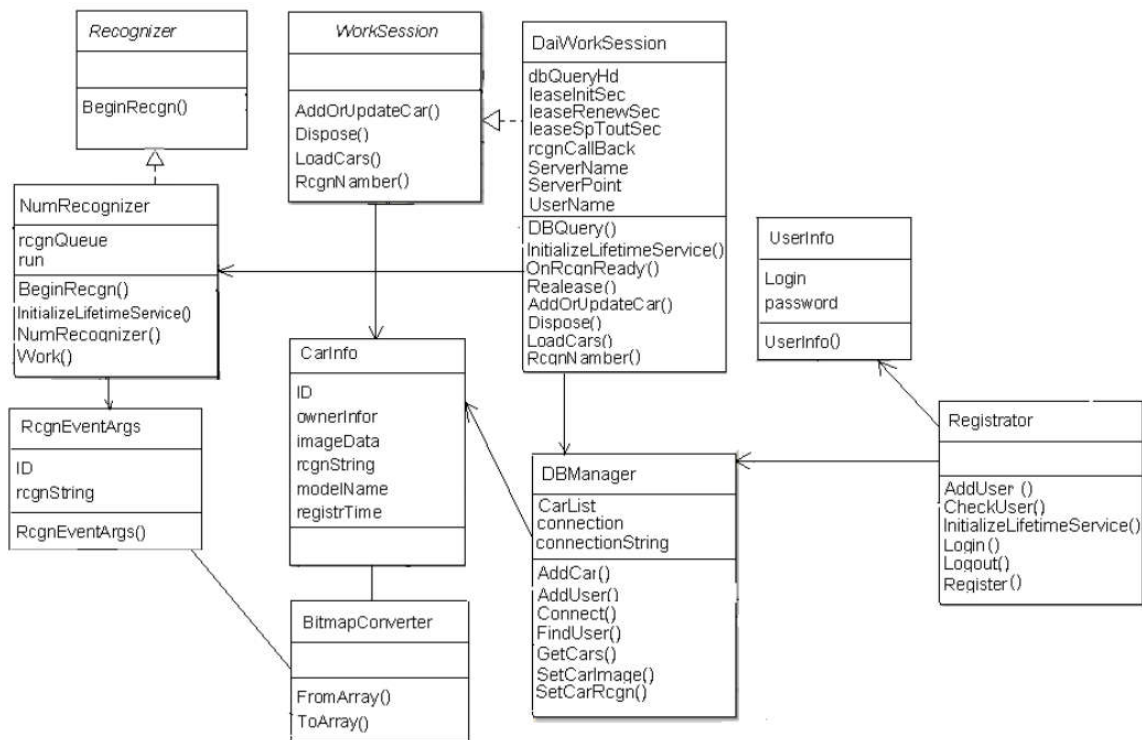


Рисунок 3.4 – UML діаграма класів

Кожен з класів має свій певний набір полів та функцій:

- Login() – функція логізації в систему, яка перевіряє правильність введених даних;
- Register() – реєстрація нових користувачів;
- AddUser() – додавання нового користувача в базу даних;
- Logout() – вихід з акаунту користувача;
- UserInfo() – інформація про користувача;
- AddCar() – додавання нового автомобіля в базу даних;
- Connect() – підключення до бази даних;
- FindUser() – знаходження необхідного користувача на основі введеного логіну та паролю;
- GetCars() – знаходження необхідних машин;

- SetCarImage() – збереження зображення для обраного автомобіля;
- SetCarRcgn() – збереження розпізнаного номерного знаку в базу даних;
- FromArray() – вибірка даних з масиву;
- ToArray() – запис даних в масив;
- DBQuery() – запит до бази даних;
- OnRcgnReady() – перевірка чи зображення автомобільного номеру яке завантажив користувач готове до розпізнавання;
- Release () – випуск зображення;
- AddOrUpdateCar() – додавання нової машини або оновлення уже існуючих записів в базі даних;
- LoadCars() – завантаження існуючих автомобілів за вказаними параметрами;
- RcgnNumber() – розпізнавання автомобільного номерного знаку у відповідності до завантаженого зображення;
- BeginRcgn() – початок розпізнавання автомобільного номерного знаку у відповідності до завантаженого зображення;
- NumRecognizer() – функція для розпізнавання номерного знаку автомобіля;
- Work() – робоча сесія;
- RcgnEventArgs() – аргументи необхідні для розпізнавання зображення завантаженого користувачем;
- Login – поле в якому зберігається логін користувача, тип даних – стрічковий;
- Password – поле в якому зберігається пароль користувача, тип даних – стрічковий;
- CarList – поле в якому зберігається список автомобілів які підійшли по певному критерію, тип даних – список;
- connection – поле в якому зберігаються дані для об'єднання, тип даних – стрічковий;

- connectionString – поле в якому зберігаються дані для під'єднання до бази даних, тип даних – стрічковий;
- UserName – ім'я користувача, тип даних – стрічковий;
- ID – ідентифікаційний номер користувача для простішого розпізнавання його в базі даних, тип даних – числовий;
- rcgnString – стрічкове представлення даних розпізнаних алгоритмом.

3.2 Апаратна структура системи автоматизованого розпізнавання автомобільних номерів для

В даному розділі представлені апаратні елементи які використовуються в системі автоматизованого розпізнавання автомобільних номерів. Всі структурні елементи зображені на схемі фізичної реалізації системи.

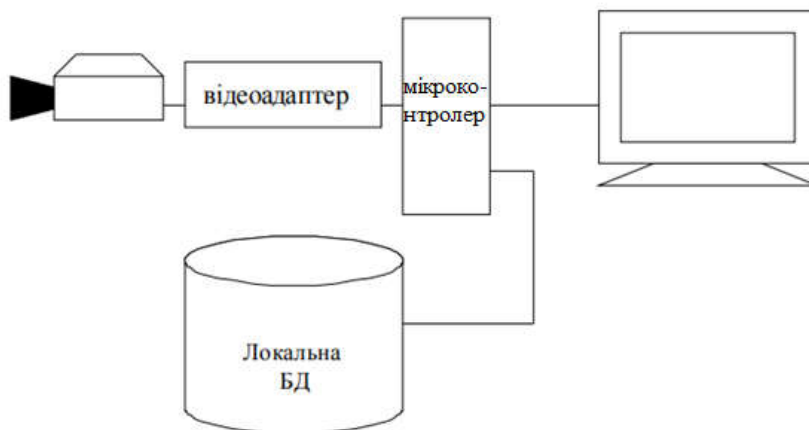


Рисунок 3.5 – Схема фізичної реалізації системи

В даному комплексі містяться відеокамери, що підключаються до персонального комп'ютера, локальної бази даних та монітора. Система може працювати в двох режимах: пасивному і активному.

Під час роботи в пасивному режимі система автоматизованого розпізнавання

просто зчитує всі номерні знаки автомобілів які проїжджають та і записує їх в базу даних.

Активний режим роботи системи заключається в тому, що кожен автомобільний номер який система ідентифікує звіряється з існуючими номерами в базі даних [44].

Алгоритм апаратної роботи автоматизованої системи розпізнавання номерних знаків автомобілів.

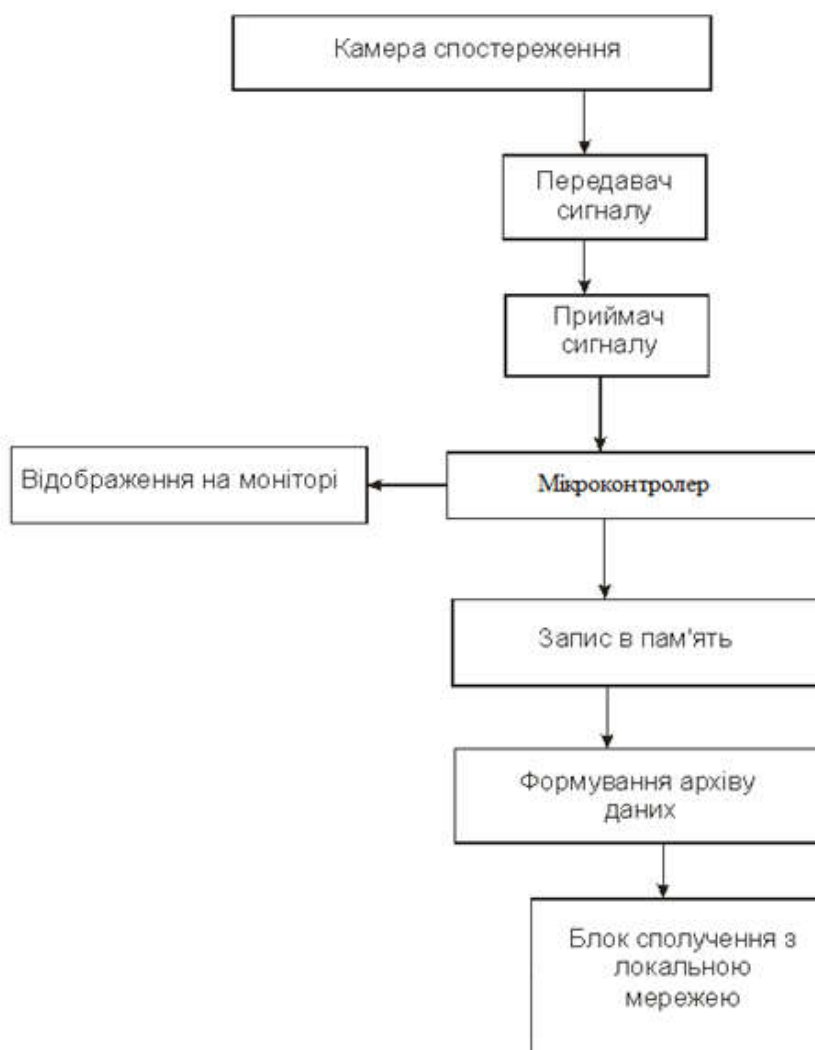


Рисунок 3.6 – Алгоритм роботи системи

Зображення з камери потрапляє на передавач сигналу, який в свою чергу відправляє його на приймач після чого зображення потрапляє на мікроконтролер який за допомогою відеоадаптеру його обробляє та показує зображення на

моніторі. Паралельно з цим відбувається запис отриманих зображень в пам'ять, додавання їх в локальну базу даних.

Для зчитування даних з камери використовується відеоадаптер – плата призначена для обробки зображень та їх генерації з подальшим виведенням на екран.

Мінімальні характеристики відеоадаптера влаштованого в мікроконтролер на основі VideoCore V який використовується в системі:

- частота пам'яті – впливає на кількість операцій записування/зчитування за такт (5010 МГц);
- частота ядра – вимірюється в мегагерцах, чим вища частота ядра, тим більше відеопроцесор може обробити даних за одиницю часу (954 МГц);
- об'єм пам'яті – використовується для зберігання частин зображення яке виведене на монітор (2 Гб);
- входи для підключення моніторів: VGA – аналоговий роз'єм(1), HDMI – інтерфейс для передачі цифрових відео/аудіо даних(1);
- інтерфейс – шина через яку відеокарта підключається до материнської плати (PCI-Express x8 2.0);
- активне охолодження – охолоджує відеопроцесор таким чином надає змогу йому коректно працювати.



Рисунок 3.7 – Приклад відеоадаптера

Для управління всіма приладами використовується мікроконтролер, він

містить в собі всі функції процесора, має оперативну пам'ять, відео ядро для підключення та роботи з адаптерами. Використовується мікроконтролер Raspberry Pi 3 Model B+ – одноплатний комп'ютер від Raspberry Pi Foundation, який працює на базі оновленого 4х-ядерного 64-бітного SoC Broadcom BCM2837B0 з збільшеною тактовою частотою 1,4 ГГц. Модуль також має: Wi-Fi встановлений двоякий діапазон даних IEEE 802.11ac, а Bluetooth – 4.2 BLE. Gigabit Ethernet, яка працює через USB 2.0 шину, що забезпечує швидкість передачі даних до 300Mbps. Об'єм оперативної пам'яті – 1 Гб, зручний формат-фактор і розміщення всіх портів, що забезпечує сумісність більшості існуючих аксесуарів. Дані мікроконтролери мають високий рівень надійності.

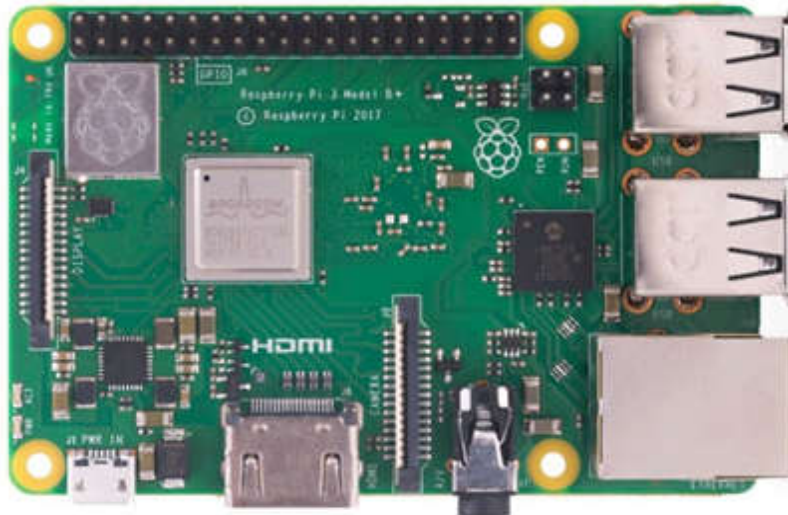


Рисунок 3.8 Мікроконтролерна плата

Для зберігання даних використовується SSD (solid-state drive) – твердотілий накопичувач на основі мікросхем на яких зберігається інформація та контролера який ними керує.

Переваги твердотілих дисків:

- відсутність рухомих частин;
- висока швидкість читання і запису, що нерідко перевершує пропускну здатність інтерфейсу твердого диска (SATA II – 3 ГБ/с, SATA III – 6 ГБ/с, SCSI тощо);

- низьке енергоспоживання;
- повна відсутність шуму (немає рухомих частин і вентиляторів охолодження);
- висока механічна стійкість;
- широкий діапазон робочих температур;
- стабільність часу зчитування файлів, незалежно від їх розташування або фрагментації;
- малі габарити і вага;
- великий модернізаційний потенціал, як самих накопичувачів, так і технологій їх виробництва;
- набагато менша чутливість до зовнішніх електромагнітних полів [45].



Рисунок 3.9 – Приклад твердотілого накопичувача

Мінімальні характеристики твердотілого накопичувача для локальної бази даних:

- об'єм – кількість даних які можуть зберігатися на диску одночасно (250 Гб);
- швидкість запису – швидкість запису даних на диск (550 Мб/сек);
- швидкість зчитування – швидкість зчитування даних з диску (520 Мб/сек);
- інтерфейс – інтерфейс підключення твердотілого накопичувача до материнської плати (SATA3).

Зображення з відеокамери після обробки відеоадаптером виводиться на

монітор – пристрій для виведення інформації. Мінімальні характеристики монітора:

- діагональ – розмір екрану (20 дюймів);
- частота оновлення – кількість кадрів показана за одиницю часу 60 Гц;
- тип матриці – технологія передачі зображення (AH-IPS);
- інтерфейс – цифровий тип підключення до відеоадаптера (HDMI);
- підсвітка – світлодіодна підсвітка (WLED).



Рисунок 3.10 – Приклад дисплею

Для відображення фізичного підключення я обрав фізичну модель – фізичне уявлення системи, об'єкта, явища або процесу з метою їхнього дослідження, тобто представлення за допомогою іншого фізичного («реального») об'єкта, що має «аналогічну» динаміку «поведінки», це означає, що математичні моделі об'єкта дослідження та об'єкта–моделі є аналогічними (якщо не тотожними). Відповідно вимірювання параметрів об'єкта–моделі дозволяє отримати значення параметрів об'єкта дослідження. Якщо установки, пристрої, пристосування на яких проводиться дослідження зберігають фізичну подібність процесів моделі тим процесам які цікавлять дослідника в системі – це фізичні моделі [46].

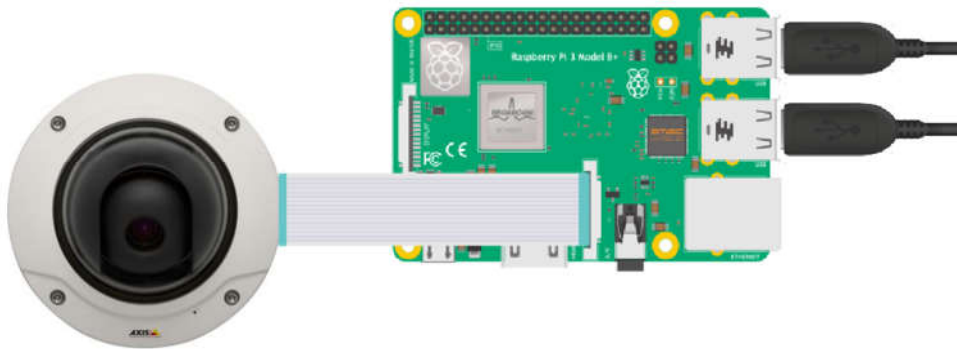


Рисунок 3.11 – Фізична модель підключення пристроїв

Для відображення повного складу елементів та зв'язків між ними обрано принципову схему.

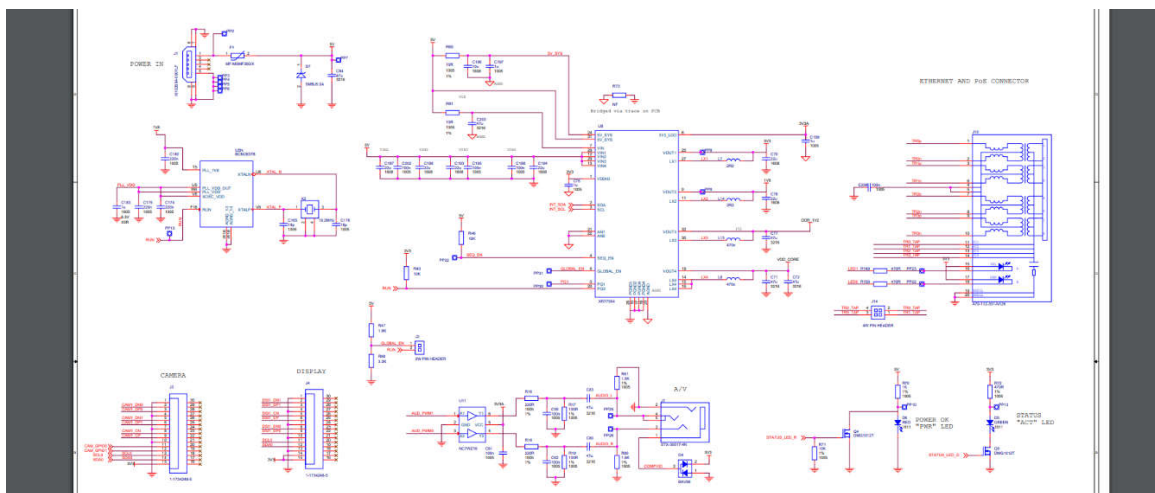


Рисунок 3.12 – Принципова схема підключень пристроїв

Вона дає повне уявлення про принципи роботи пристрою. Дані схеми виконують без дотримання масштабу, елементи позначають за допомогою умовних графічних позначень, розташування елементів не враховується. Всі елементи на схемі мають своє позначення, зазвичай для позначень використовують літери латинського алфавіту та арабські цифри. Позиційне позначення елемента в загальному випадку складається з трьох частин, які вказують вид елемента, його номер і функцію. Вид і номер – обов'язкова частина позначень. Вони повинні бути

надані всім елементам [47].

3.3 Реалізація системи автоматизованого розпізнавання автомобільних номерів для «Розумного» міста.

Алгоритм автоматизованого розпізнавання автомобільних номерів було реалізовано в середовищі для розробки IntelliJ IDEA Community Edition на мові програмування Java – це об'єктно–орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. В 2009 році компанія «Oracle» купила «Sun Microsystems» і почала займатися мовою Java.

Переваги мови програмування Java:

- синтаксис мови «простий, об'єктно–орієнтований та звичний» – всі дані та дії групуються в об'єкти класів;
- реалізація є «безвідмовною та безпечною» – можливість виконувати операції над елементом масиву який знаходиться за його межами або є порожнім елементом, є можливість зчитування з пуского каталогу або некоректної URL адреси, захист від вводу некоректних даних користувачем;
- зберігається «незалежність від архітектури та переносність» – тобто будь–яка програма написана на мові програмування Java буде працювати на будь–якій апаратній чи системній платформі, адже дана мова використовує байт–код, основною його перевагою є портативність. Для того, щоб зменшити витрати часу при роботі використовується JIT(just in time) компіляція яка під час першого запуску перетворює байт–код в машинний та додає його в кеш;
- висока продуктивність виконання – використання статичної компіляції, яка передбачає компіляцію програми в машинний код;
- мова є «інтерпретованою, багатонитевою, і з динамічним зв'язуванням модулів» – використання автоматичного збирача сміття для простого керування

пам'яттю та керування життєвим циклом об'єкту, якщо до певного об'єкту немає посилань, збирач сміття автоматично видаляє його з пам'яті, це може відбуватися в будь-який час але найкраще виконувати це коли програма бездіяльна.

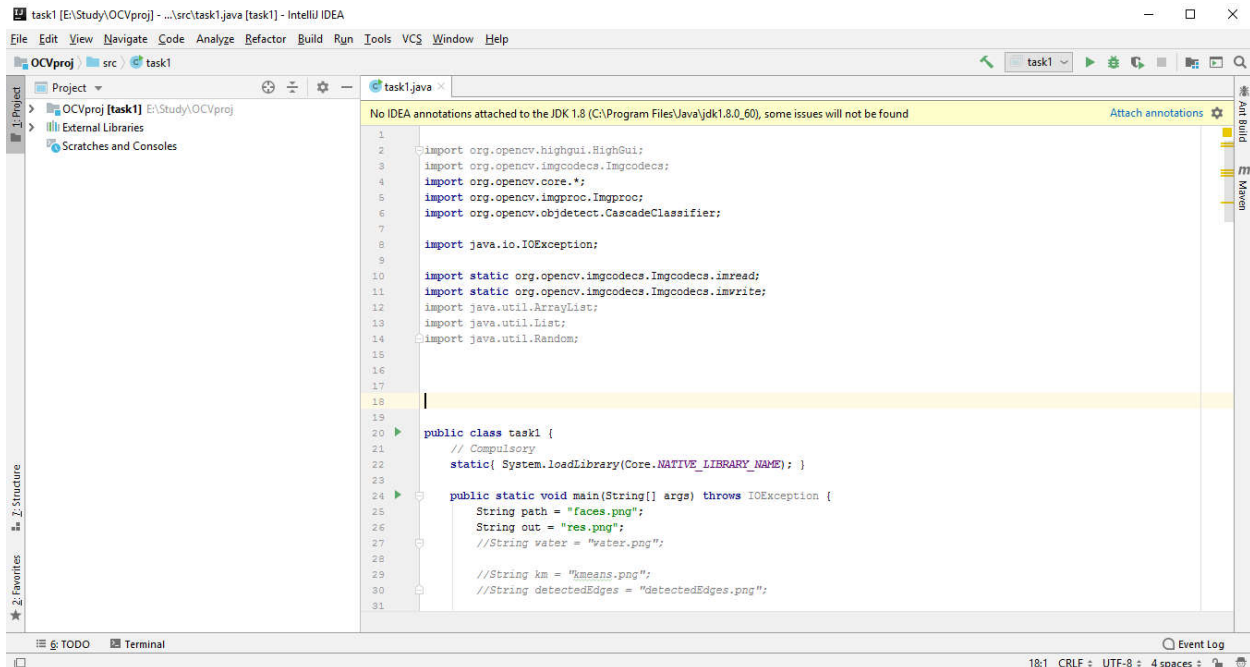


Рисунок 3.13 – Реалізація алгоритму в IntelliJ IDEA Community Edition

Для коректної роботи алгоритму автоматизованого розпізнавання автомобільних номерів використовується бібліотека комп'ютерного зору з відкритим кодом OpenCV. В ній містяться функції, алгоритми обробки зображень та комп'ютерного зору. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях, відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях.

Основні цілі бібліотеки:

- розвивати дослідження у напрямку комп'ютерного зору, забезпечуючи добре оптимізований та відкритий код бібліотеки;
- поширювати знання у сфері комп'ютерного зору, забезпечуючи загальну інфраструктуру, яку б могли розвивати розробники, таким чином код ставатиме більш легким для сприйняття та обміну;

– розвивати засновані на роботі з комп'ютерним зором комерційні додатки, створюючи не залежну від платформи, оптимізовану та безкоштовну бібліотеку. Для цього використовувалася ліцензія, яка не вимагала від таких комерційних додатків бути відкритими.

Напрямки застосування бібліотеки:

– аналіз та обробка зображень – математичні і алгоритмічні основи аналізу та класифікації зображень;

– системи з розпізнавання обличчя – технології здатні вирахувати обличчя людини на цифровому зображенні та проаналізувати його;

– побудова 3D моделей об'єктів – має сукупність інструментів та функцій які призначені для відображення об'ємних об'єктів;

– створення 3D хмар точок зі стерео камер – збір набору даних про точки в деякій системі координат;

– склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю;

– система взаємодії людини з комп'ютером – рішення які створюються для взаємодії користувача з машиною якою він керує;

– пошук схожих зображень із бази даних – техніка машинного зору яка вирішує проблему пошуку зображення за вмістом;

– усунення ефекту червоних очей при фотозйомці зі спалахом;

– аналіз руху – використання моделі оптичних потоків і метод оцінки руху по різницям зображень забезпечує досить ефективний і гнучкий апарат для аналізу видимого руху об'єктів на цифрових зображеннях;

– ідентифікація об'єктів – розпізнавання необхідних об'єктів на основі параметрів заданих користувачем;

– сегментація зображення – можливість розділення цифрового зображення на сегменти на основі заданих критеріїв;

– трекінг відео – алгоритми аналізу відеокadrів та визначення рухомих об'єктів на ньому;

– розпізнавання елементів сцени і додавання маркерів для створення доповненої реальності.

Система автоматичного розпізнавання автомобільних номерів запускається за допомогою ярлика “Toolkit”.

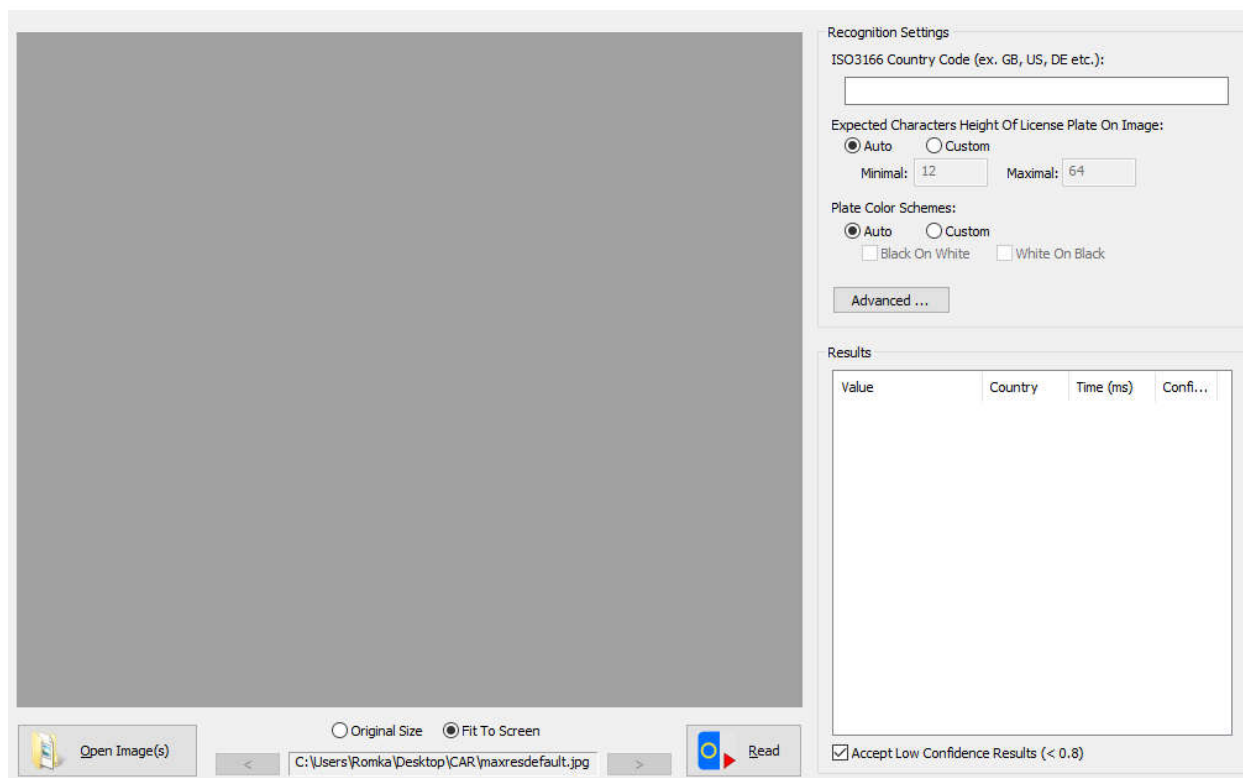


Рисунок 3.14 – Головний екран системи

На рисунку 3.14 показані основні можливості системи:

– recognition settings – додавання коду країни для підвищення шансу розпізнавання зображення;

– expected characters height – очікуваний розмір символів на автомобільному номерному знаку;

– plate color scheme – кольорова схема номерного знаку, є дві стандартні це білі символи на чорному фоні і навпаки чорні цифри на білому фоні, також є можливість обрати необхідну користувачеві кольорову схему;

– results – таблиця з результатами роботи алгоритму, показує країну час роботи, якість та цифри номерного знаку;

- image size – розмір відображення картинки в системі яку завантажує користувач;
- open image(s) – відкриває каталоги системи для пошуку необхідних цифрових зображень;
- read – зчитування даних з цифрового зображення автомобільного номерного знаку завантаженого користувачем;
- accept low confidence results – опція прийняття мало імовірних результатів роботи алгоритму;
- advanced – додаткові користувацькі налаштування які можуть підвищити можливість розпізнавання автомобільного номерного знаку.

Натиснувши кнопку «Open Image(s)» відкриється список папок з даними в якому користувач може знайти необхідні йому цифрові зображення.

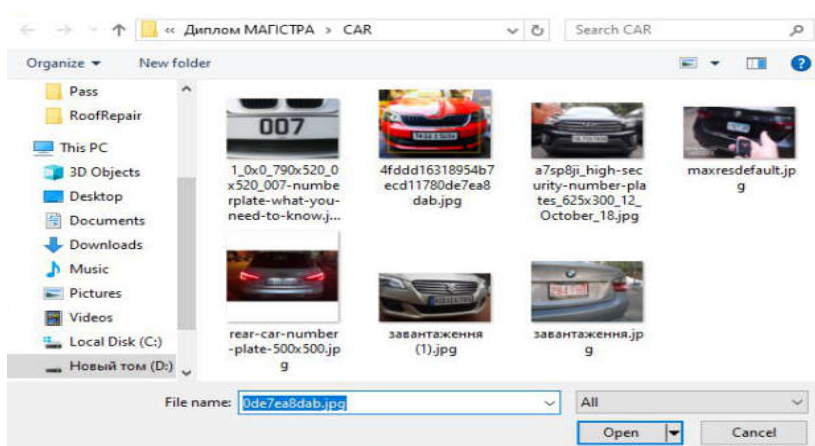


Рисунок 3.15 – Перегляд каталогів

Після вибору необхідного зображення необхідно натиснути кнопку «Read» для того, щоб запустити алгоритм розпізнавання автомобільних номерів. Після завершення результати про роботу алгоритму виведуться в таблиці з назвою «Results» та запишуться в базу даних.

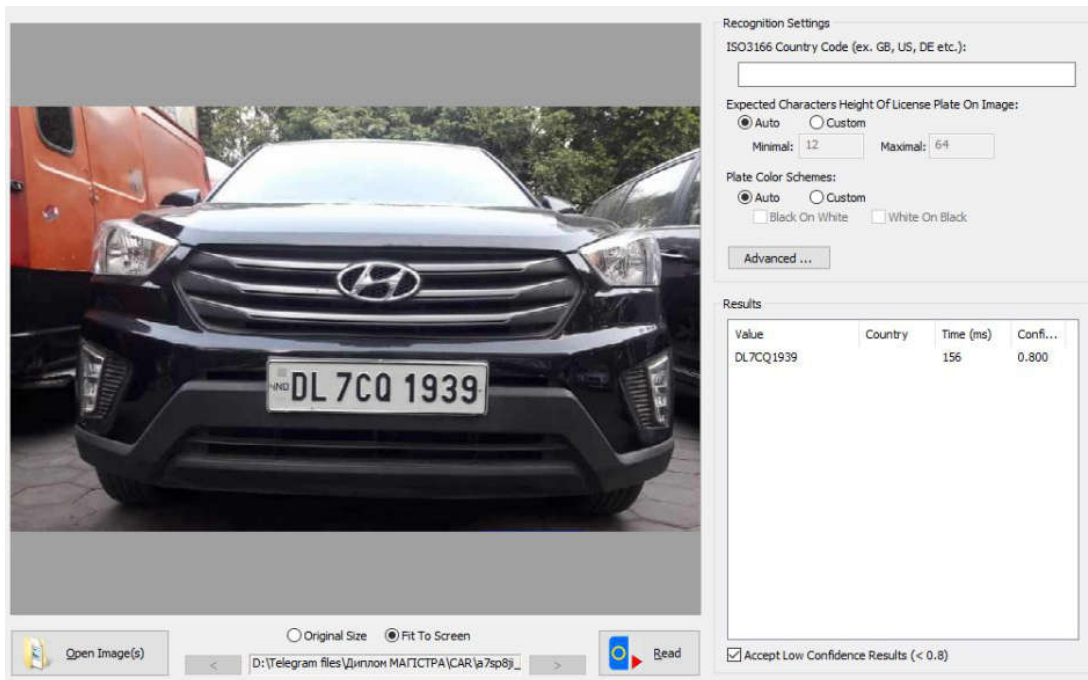


Рисунок 3.16 – Приклад роботи системи розпізнавання автомобільних номерів

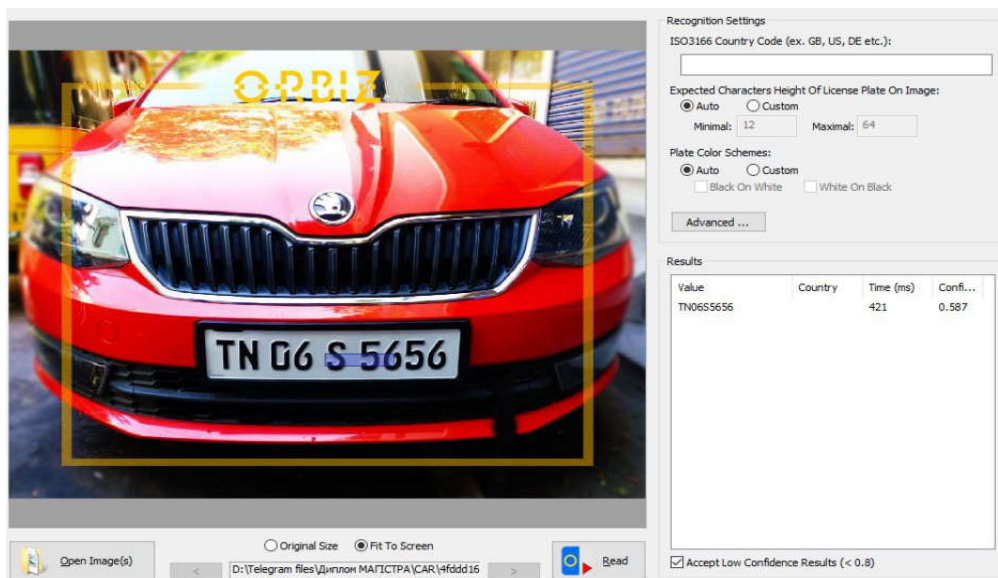


Рисунок 3.17 – Приклад роботи системи розпізнавання автомобільних номерів

На рисунках 3.16 та 3.17 показано приклади роботи системи автоматизованого розпізнавання автомобільних номерів. Результати відображені в таблиці «Results». У обох випадках номерні знаки розпізнало без похибок. Весь процес розпізнавання залежить від якості фото, дистанції номерного знаку та інших факторів поданого цифрового зображення. За допомогою налаштувань системи які можна зробити вручну можна збільшити можливість розпізнавання автомобільного знаку.

Дану систему можна просто реалізувати використовуючи всі доступні можливості IntelliJ IDEA Community Edition та бібліотеки для роботи з комп'ютерним зором OpenCV.

3.4 Експериментальні дослідження ефективності системи автоматизованого розпізнавання автомобільних номерів для «Розумного» міста.

Розроблену систему було досліджено на ефективність роботи для визначення алгоритму на якому буде реалізовано дану систему.

Основні параметри алгоритмів для коректної роботи системи:

- коректне зчитування зображення;
- можливість виділення основного об'єкту від шумів які присутні на цифровому зображенні;
- врахування діапазону фільтрації;
- відлагодження цифрових зображень за рахунок існуючого функціоналу;
- час виконання алгоритму.

Результати аналізу алгоритмів зображено на рисунку 3.18. На даній діаграмі зеленим прямокутником показано хороший результат, тобто алгоритм зміг виконати всі необхідні задачі, жовтий – функціональні можливості виправдані частково та червоний прямокутник – алгоритм показав найгірший результат з усіх перевірених [48].

	Потенціали	Хаар	Січні площини
Коректне зчитування зображення	Зелений	Жовтий	Жовтий
Виділення основного об'єкту	Зелений	Зелений	Червоний
Врахування діапазону фільтрацій	Зелений	Жовтий	Жовтий
Відлагодження зображення	Жовтий	Червоний	Жовтий
Час виконання алгоритму	Зелений	Жовтий	Червоний

Рисунок 3.18 – Діаграма порівняння роботи алгоритмів

Модель «Потенціалів» показав найкращі результати під час аналізу та тестування алгоритмів на відповідність вимогам, які ставить перед собою підприємство для повного та комфортного використання системи автоматизованого розпізнавання автомобільних номерів [49].

Для проходження всіх етапів алгоритму розпізнавання цифрового зображення, тобто повного його завершення, необхідно в середньому 300 мс, результат показано на рисунку 3.19.

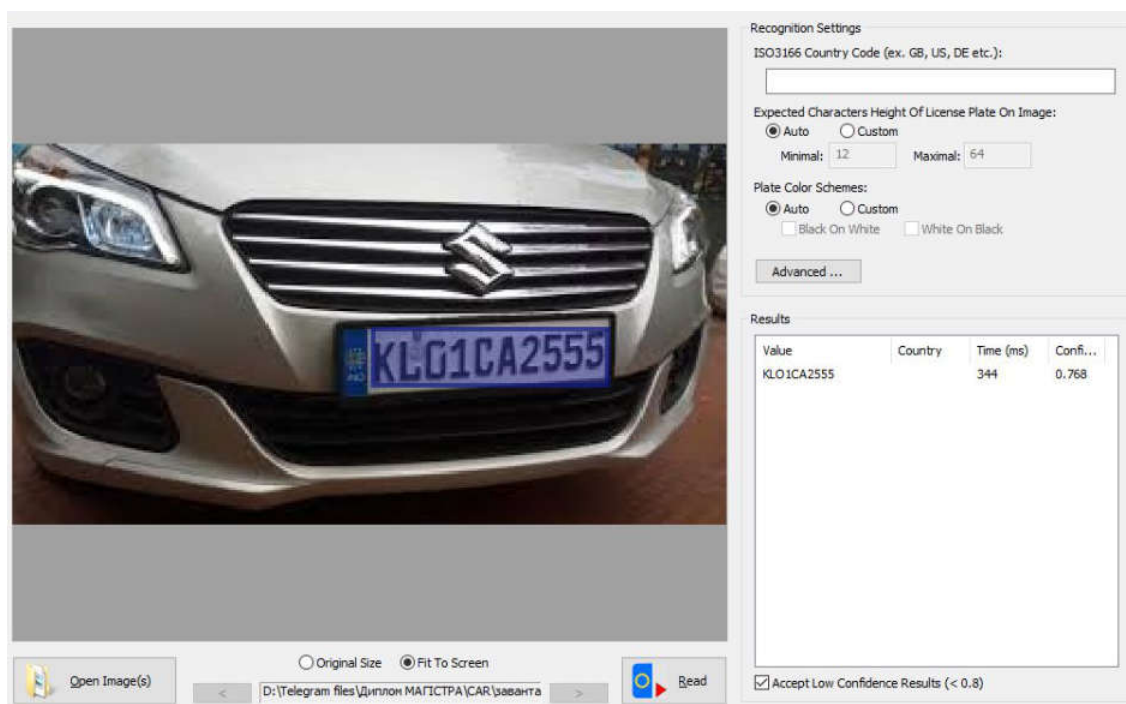


Рисунок 3.20 – Результат проходження моделі «Потенціалів»

Для порівняння алгоритм роботи «Січних площин» зображений на рисунку 3.21, який був використаний для перевірки швидкості виконання та складності роботи з виконується в середньому 400 мс хвилин [50].

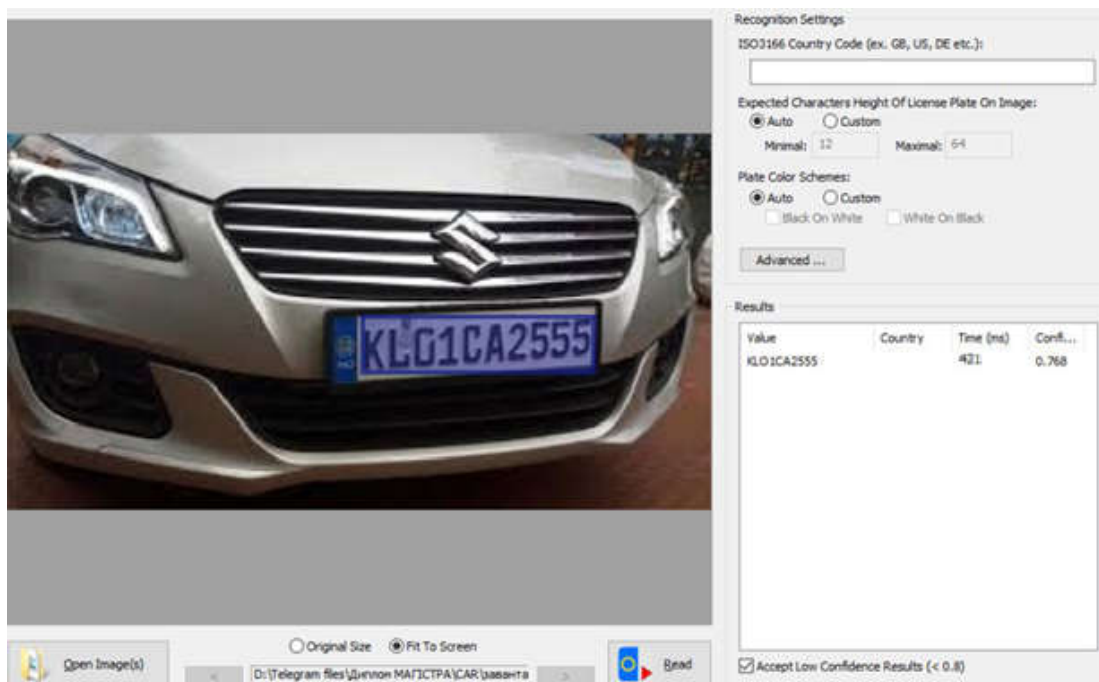


Рисунок 3.21 – Результат проходження моделі «Січних площин»

У ході аналізу алгоритмів було досліджено, що працювати з алгоритмом «Потенціалів» зручніше та виконується на сто мілісекунд швидше, результат виконання показано на рисунку 3.22 [51].

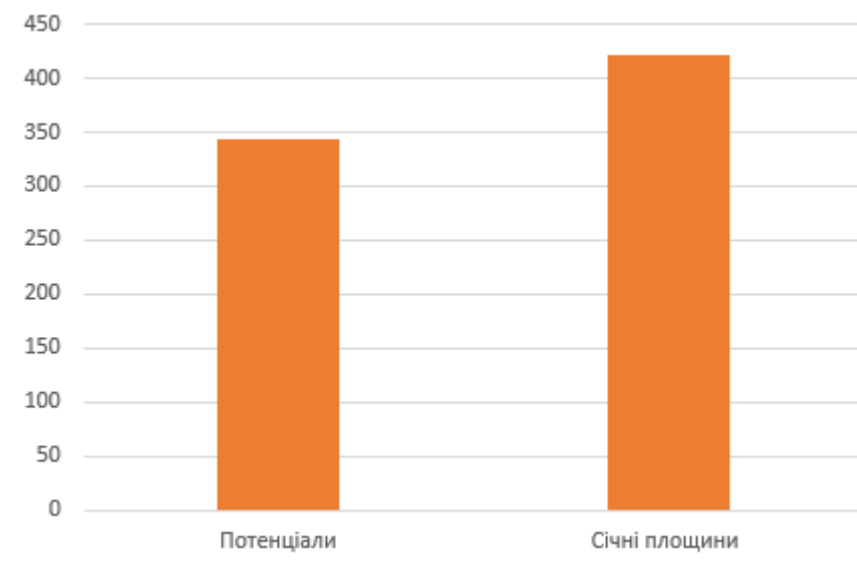


Рисунок 3.22 – Діаграма порівняння швидкості алгоритмів

З даних, поданих на рисунку 3.22, можна зробити висновок, що алгоритм «Потенціалів» є продуктивніший ніж алгоритм «Січних площин».

ВИСНОВКИ

1. В ході написання магістерської роботи було досліджено різні алгоритми нормалізації зображення номерних знаків по куту повороту, сегментації символів, якості та швидкості роботи, що дозволяє використовувати обрані алгоритми в якості етапів підготовки зображення до розпізнавання також було розглянуто безпосередньо алгоритми та методи розпізнавання об'єктів на цифровому зображенні – та обрано найефективніший з усіх доступних.

2. Досліджено системи автоматизованого розпізнавання зображень які уже існують на ринку та визначено програмні підходи, шаблони та модель які найкраще підійшли для даного типу системи.

3. Інформація яку ми отримуємо на етапі сегментації зображення дозволяє визначити тип номерного знаку, що спрощує подальше розпізнавання за рахунок знань про походження символу та приналежність його до букв чи цифр.

4. Розроблено алгоритми розпізнавання автомобільних номерів для систем автоматичного розпізнавання на основі моделі потенціалів, що підвищило імовірність розпізнавання номерних знаків та швидкість розпізнавання.

5. Під час написання системи автоматизованого розпізнавання автомобільних номерів було використано об'єктно–орієнтований підхід, який дозволяє легко вносити будь–які модифікації в систему.

6. Вказано інформацію про об'єкт практики, описано відділи на які поділено підприємство, їх призначення та роботу яку вони виконують, також досліджено обладнання за яким працюють працівники, та описано мережеве обладнання яким користуються на підприємстві.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Березький О.М. Методичні рекомендації до виконання магістерської роботи з освітнього ступеня “Магістр”. Спеціальність: 123 – Комп’ютерна інженерія. Магістерська програма – Комп’ютерна інженерія" / О.М. Березький, Л.О. Дубчак, Г.М. Мельник /Під ред. О.М. Березького – Тернопіль: ТНЕУ, 2018.– 41 с.
2. Гураль І.В. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп’ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. – Тернопіль: ТНЕУ, 2019. – 33 с.
3. Баданін Р.В. Алгоритм роботи системи автоматичного розпізнавання автомобільних номерів для «розумного» міста номерів // Науково–практична конференція молодих вчених і студентів «інтелектуальні комп’ютерні системи та мережі» Частина 1. 15 квітня 2019 р., с. 18;
4. Баданін Р.В. Основні етапи процесу розпізнавання автомобільних номерів // II науково–практична конференція молодих вчених і студентів «інтелектуальні комп’ютерні системи та мережі» Частина 1. 14 листопада 2019., с. 27.
5. Fischer I. Pattern Recognition Algorithms for Symbol Strings. Paperback, 2010. 100 с.
6. Karl Tombre. Graphics Recognition: Algorithms and Systems. Springer, 2015. 94 с.
7. Dietrich W.R.P., Joachim H. Applied Pattern Recognition. Vieweo Teusner, 2003. 55 с.
8. Дж. В Райзин. Классификация и кластер. М.: Мир, 1980. 399 с.
9. Li J.B., Chu S.C., Pan J.S. Kernel Learning Algorithms for Face Recognition. Communication, 2014. 40 с.
10. Розробка розподіленої системи розпізнавання автомобільних номерів: інформаційний портал. URL: <http://dSPACE.nbuV.gov.ua/bitstream/handle/Volkova.pdf>

(дата звернення: 13.10.2019).

11. Автоматичне розпізнавання автомобільних номерних знаків за допомогою нейронної мережі зі зворотнім поширенням помилки: інформаційний портал. URL: <https://nau.edu.ua/site/variables/news/2018/5/%D0%A2%D0%B5%9A%D0%BE%D0%BC%D0%BF%E.pdf>(дата звернення: 30.11.2019).

12. Навчальний ресурс з тестування програмного забезпечення: веб-сайт. URL: http://qalearning.com.ua/theory/about_qa(дата звернення: 15.10.2019).

13. Тестування додатків. Писати тести – це як писати код. Тільки тести: веб-сайт. URL: http://is.unicyb.kiev.ua/files/poliachenko_2015.pdf(дата звернення: 7.10.2019).

14. Нормалізація забраження автомобільного номеру і сегментація символів для подальшого розпізнавання: інформаційний портал. URL: <http://space.nbuv.gov.ua/bitstream/handle/123456789/56557/42-Murygin.pdf?see=1>(дата звернення: 25. 11.2019).

15. Гамаюн І. П., Безменова О. М. Оцінювання міри схожості між об'єктам, що характеризуються кількісними і номінальними ознаками. Інформаційні технології: наука, техніка, технологія, освіта, здоров'я. Тези доповідей XXI міжнародної науково-практичної конференції. Ч. 1 (29–31 травня 2013 р., Харків). – Харків: НТУ «ХП», 2013. – С. 9.

16. Теорія розпізнавання образів: інформаційний портал. URL: <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/16460/1/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F%20%D1%80%D0%BE%D0%B7%D0%BF%D1%96%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%D0%BD%D0%97.pdf>(дата звернення: 17.11.2019).

17. Метод допустимих перетворень: інформаційний портал. URL: <https://helpiks.org/7-88520.html>(дата звернення: 20.11.2019).

18. Loumos V., Anagnostopoulos C., Anagnostopoulos I., Kayafas E. Alicense plate-recognition algorithm for intelligent transportation system applications. Lesvos, 2006. 392 с.

19. Sirithinaphong T. The recognition of car license plate for automatic parking

system. Signal Processing and Its Applications. Lesvos, 1999. 457 с.

20. ДСТУ Р 50577–93. Знаки державні реєстраційні транспортних засобів. Типи і основні розміри. Технічні вимоги. [Чинний від 1993–29–06]. Видавництво стандартів, 2002. 33 с. (Інформація та документація).

21. Boreiko O.Y., Teslyuk V. M. Developing a controller for registering passenger flow of public transport for the "smart" city system. Eastern–European Journal of Enterprise Technologies. 2016. 46 с.

22. Boreiko, O. Y., Teslyuk V. M., Zelinsky A., Berezsky O. Development of models and means of the server part of the system for passenger traffic registration of public transport in the "smart" city. Eastern–European Journal of Enterprise Technologies. 2017. 47 с.

23. ДСТУ 3582–97. Скорочення слів в українській мові у бібліографічному описі. [Чинний від 1997–07–04]. Держстандарт України, 2012. 55 с. (Інформація та документація).

24. Sankar K.M. Pattern Recognition Algorithms for Data Mining. Prabira Charman & Hall, 2010. 10 с.

25. Робота каскада Хаара: веб–сайт. URL: <https://habr.com/ru/company/recognitor/blog/228195/>(дата звернення: 20.11.2019).

26. Бібліотека MSDN: інформаційний портал. URL: <http://msdn.microsoft.com/en-us/vstudio/default.aspx>(дата звернення: 20.12.2019).

27. Несен М.В. Віддалена система адаптивного розпізнавання образів. Херсон: Издательство Херсонского морского института, 2006. 114 с.

28. Давыдов В.О. Автоматизация технологического процесса регистрации номеров при управлении транспортными потоками. Одесса: ОНПУ, 2004. 155 с.

29. Гонсалес Р. Принципи розпізнавання образів. Мир, 1978. 22 с.

30. Шапіро Л. Комп'ютерний зір. Мир, 2006. 13 с.

31. Васильев В.І. Системи розпізнавання. Наукова думка, 1983. 4 с.

32. Анісімов Б.В., Курганов В.Д., Злобін В.К. Розпізнавання і цифрова обробка зображень. Мир, 1983. 25 с.

33. Wang P.P. Genetic Algorithms for Pattern Recognition. CRC Press,

2007. 55 c.

34. Bezdek C.J Pattern Recognition with Fuzzy Objective Function Algorithms. CRC Press, 2010. 20 c.

35. Xuejun T. Computational algorithms for fingerprint recognition. University of California, 2003. 41 c.

36. Szeliski R. Computer Vision: Algorithms and Applications. Springer, 2010. 11 c.

37. Pradipta M., Sushmita P. Scalable pattern recognition algorithms. DOL, 2014. 21 c.

38. Blostein D. Graphics Recognition Algorithms and Applications. Springer, 2001. 15 c.

39. Javidi B. Image Recognition and Classification: Algorithms, Systems, and Applications. CRC Press, 2002. 57 c.

40. Elden L. Matrix Methods in Data Mining and Pattern Recognition. Society for Industrial and Applied Mathematics, 2007. 91 c.

41. Treiber M.A. An Introduction to Object Recognition: Selected Algorithms for a Wide Variety of Applications. Springer, 2010. 29 c.

42. Kuncheva I.L. Combining Pattern Classifiers: Methods and Algorithms. Wiley, 2014. 66 c.

43. Flach P. Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press, 2012. 97 c.

44. Riesen K. Structural Pattern Recognition with Graph Edit Distance. Springer, 2015. 25 c.

45. Hornegger J. Applied pattern recognition: Algorithms and Implementation in C++. Vieweg, 2003. 14 c.

46. Bishop M.C. Neural Networks for Pattern Recognition. Oxford University Press, 1995. 32 c.

47. Hong C. Sparse Representation, Modeling and Learning in Visual Recognition. Springer, 2015. 23 c.

48. Duda O.R., Stork G.D., Pattern Classification. Wiley, 2000. 11 c.

49. Rafgerden T. Perfect Algorithm. Foundation. Springer, 2019. 9 c.

50. Downey B.A. Think Data Structures: Algorithms and Information Retrieval in Java. Green Tea Press, 2016. 16 c.

51. Morin P. Open Data Structures: An Introduction. Athabasca University Press, 2013. 77 c.

ДОДАТОК А

Лістинг коду системи розпізнавання автомобільних номерів

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;

namespace Calc
{
    class Logic
    {
        public static decimal[] Parser(string[] s)
        {
            decimal[] array = new decimal[s.Length];

            for (int i = 0; i < s.Length; i++)
            {
                if (!decimal.TryParse(s[i], out array[i]))
                    return array = null;
            }

            return array;
        }

        public static string[] ArraySplit(string s)
        {
            string[] Numbers = s.Split(new char[] { '+', '-' },
StringSplitOptions.RemoveEmptyEntries);

            return Numbers;
        }

        public static string Bounds(string s)
        {
            var openBound = s.LastIndexOf('(');
            var closeBound = s.IndexOf(')');

            var workString = s.Substring(openBound, closeBound
- openBound);

            workString = workString.Replace("(", "");

            var replaceResult =
Logic.SecondPriorityOperations(Logic.Parser(Logic.FirstOrderOperatio
ns(Logic.ArraySplit(workString))), workString, FindPlus(workString),
FindMinus(workString));
```

```

        s = s.Replace("(" + workString + ")",
replaceResult.ToString());

        return s;
    }

    public static string[] FirstOrderOperations(string[]
array)
    {
        for (int i = 0; i < array.Length; i++)
        {
            if (array[i].Contains("*") ||
array[i].Contains("/"))
            {
                decimal[] decimalWorkArray =
Parser(array[i].Split(new char[] { '*', '/' }));
                decimal repNumber = 0;
                int n = 0;
                char[] charArray = array[i].ToCharArray();

                for(int j = 0; j < charArray.Length; j++)
                {
                    if (charArray[j].Equals('*'))
                    {
                        //how to multiply
                        if (n == 0 && n + 1 <
decimalWorkArray.Length)
                            {
                                repNumber = decimalWorkArray[n] * decimalWorkArray[n +
1];

                                    n++;
                                }
                            else if (n + 1 <
decimalWorkArray.Length)
                                {
                                    repNumber *= decimalWorkArray[n +
1];

                                        n++;
                                }
                            }
                        else if (charArray[j].Equals('/'))
                        {
                            if (n == 0 && n + 1 <
decimalWorkArray.Length)
                                {
                                    if (decimalWorkArray[n + 1] ==
0)
                                        {
                                            UI.DivideByZero();
                                            break;
                                        }
                                    else
                                        {

```

```

        repNumber = decimalWorkArray[n] / decimalWorkArray[n +
1];
    }
    n++;
}
else if (n + 1 <
decimalWorkArray.Length)
{
    if (decimalWorkArray[n + 1] == 0)
    {
        UI.DivideByZero();
        break;
    }
    else
    {
        repNumber /= decimalWorkArray[n +
1];
    }
    n++;
}
}
}
array[i] = repNumber.ToString();
}
}
return array;
}

public static int FindPlus(string s)
{
    var plus = Regex.Matches(s, @"\+").Count;

    return plus;
}

public static int FindMinus(string s)
{
    var minus = Regex.Matches(s, @"\-").Count;

    return minus;
}

public static decimal
SecondPriorityOperations(decimal[] array, string s, int plus, int
minus)
{
    decimal number = 0;
    var i = 0;
    var charArray = s.ToCharArray();

```

```

bool check = false;

if (plus == 0)
{
    minus++;
}

for (int j = 0; j < charArray.Length; j++)
{
    if (charArray[j].Equals('+'))
    {
        check = true;

        if (plus != 0)
        {
            if(i == 0)
            {
                plus++;
            }

            for (; plus != 0; i++)
            {
                number += array[i];

                plus--;
            }
        }
    }

    if(charArray[j].Equals('-'))
    {
        check = true;

        if (minus != 0)
        {
            for (; minus != 0; i++)
            {
                if(i == 0)
                {
                    number = array[i];

                    if(minus == 1)
                    {
                        number -= array[i + 1];
                        i++;
                    }
                }
                else
                {
                    number -= array[i];
                }

                minus--;
            }
        }
    }
}

```



```

        }
    }
}

if (check)
    return number;
else
    return array[i];
}
}
}

```

-----UI-----

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Calc
{
    class UI
    {
        public static void EnterExpression()
        {
            Console.Write("Enter expression: ");
        }

        public static void WrongExpression()
        {
            Console.Write("Wrong expression. Enter new: ");
        }

        public static void Result(decimal res)
        {
            Console.WriteLine("Result: " + res);
        }

        public static void DivideByZero()
        {
            Console.WriteLine("You can't divide by zero!");
        }
    }
}

```

-----Program-----

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;
using System.Data;
using System.Text.RegularExpressions;

namespace Calc
{
    class Program
    {
        //multiply and divide
        static void Main(string[] args)
        {
            UI.EnterExpression();
            var expression = Console.ReadLine();
            int plusCount;
            int minusCount;

            while (expression.Contains('(') &&
expression.Contains(')'))
                {
                    expression = Logic.Bounds(expression);
                }

            while
(Logic.Parser(Logic.FirstOrderOperations(Logic.ArraySplit(expression
))) == null)
                {
                    UI.WrongExpression();
                    expression = Console.ReadLine();
                }

            plusCount = Logic.FindPlus(expression);
            minusCount = Logic.FindMinus(expression);

            UI.Result(Logic.SecondPriorityOperations(Logic.Parser(Logic.FirstOrd
erOperations(Logic.ArraySplit(expression))), expression, plusCount,
minusCount));

                Console.ReadLine();
            }
        }
    }
}

```