

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

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ____ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: Інтелектуальна система розпізнавання елементів дорожнього руху

Виконав: студент другого курсу, групи ІТ-84мп
(шифр групи)

_____ Головацький Ігор Володимирович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к.т.н., доцент Корнага Я.І. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

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

Студент _____
(підпис)

Київ – 2019 року

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

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
І.Р. Пархомей
(підпис)

«__» _____ 2019 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Головацькому Ігорю Володимировичу
(прізвище, ім'я, по батькові)

1. Тема дисертації «Інтелектуальна система розпізнавання елементів дорожнього руху», _____
науковий керівник дисертації доцент, к.т.н., доцент Корнага Я.І., _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 2019 р. № _____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження – комп'ютерне бачення (зір) для розпізнавання елементів дорожнього руху за даними відео потоку.

4. Предмет дослідження – розпізнавання елементів дорожнього руху.

5. Перелік завдань, які потрібно розробити – аналіз ефективності та точності існуючих методів розпізнавання об'єктів; розробка архітектури та програмного забезпечення інтелектуальної системи; дослідження ефективності розробленого програмного забезпечення.

6. Орієнтовний перелік ілюстративного матеріалу – чотири плакати

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	12.09.2019 р.	
2	Постановка задачі	16.09.2019 р.	
3	Аналіз інформаційного забезпечення	21.09.2019 р.	
5	Аналіз алгоритмічного забезпечення	24.09.2019 р.	
6	Розробка алгоритмічного забезпечення	16.10.2019 р.	
7	Розробка програмного забезпечення	02.11.2019 р.	
8	Маркетинговий аналіз стартап-проекту	12.11.2019 р.	
9	Висновки	14.11.2019 р.	

Студент

(підпис)

І.В. Головацький

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Я.І. Корнага

(ініціали, прізвище)

АНОТАЦІЯ

У роботі розглянуто проблему розпізнавання елементів дорожнього руху у відео потоці, проведено аналіз наявних проблем та складнощів в існуючих методах розпізнавання елементів та порівняння їхніх характеристик точності та швидкодії, переваг та недоліків.

Розроблено інтелектуальну систему розпізнавання елементів дорожнього руху за допомогою алгоритмів машинного навчання та нейронних мереж. Система може бути використана у відео реєстраторах та у системах пасивної безпеки автомобіля.

Загалом в роботі розкрито питання призначення та доцільність використання нейронної мережі та представлена програмна реалізація системи за допомогою мови програмування C# та бібліотеки Accord.NET, основними вимогами якої є: прийнятна точність розпізнавання, можливість використання відео потоку в якості вхідних даних, знайдені елементи повинні бути інтуїтивно виділені серед інших елементів та простота в налагодженні. Окремо було приділено увагу локальним результатам експериментів, що дають уявлення про характеристики запропонованої системи.

Ключові слова: інтелектуальна система, нейронна мережа, машинне навчання, алгоритм, комп'ютерний зір, дорожній рух.

Розмір пояснювальної записки – 81 аркушів, містить 23 ілюстрацій, 28 таблиць, 6 додатків.

ABSTRACT

Examines the problem of recognition of traffic elements in the video stream, analyzes the existing problems and complexities in the existing methods of recognition of the elements and compares their characteristics of accuracy and speed, advantages and disadvantages.

An intelligent system for recognizing traffic elements is using machine learning algorithms and neural networks. The system can be used in video recorders and passive vehicle security systems.

In general, the paper addresses the purpose and feasibility of using a neural network and presents the software implementation of the system using the C# programming language and the Accord.NET library. The main requirements of which are: acceptable recognition accuracy, the ability to use video stream as input, found elements should be intuitive highlighted among other elements and simplicity in configuring. Special attention was paid to the local results of the experiments, which give an idea of the characteristics of the proposed system.

Keywords: intelligent system, neural network, machine learning, algorithm, computer vision, road traffic

Explanatory note size – 81 pages, contains 23 illustrations, 28 tables, 6 applications.

**Пояснювальна записка
до магістерської дисертації**

на тему: *Інтелектуальна система розпізнавання елементів
дорожнього руху*

Київ – 2019 року

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ	12
1.1 Об'єкт та предмет дослідження	12
1.2 Способи розпізнавання елементів дорожнього руху	14
1.2.1 Розпізнавання об'єкта за допомогою кольору	15
1.2.2 Розпізнавання об'єкта за допомогою форми	20
1.2.3 Розпізнавання об'єкта гібридними методами	23
1.3 Методи класифікації елементів дорожнього руху	23
1.4 Методи відстеження елементів дорожнього руху	25
1.5 Огляд існуючих рішень	26
1.5.1 Таһа Khan система розпізнавання знаків дорожнього руху	26
1.5.2 Boris Karamatic система розпізнавання знаків дорожнього руху. 27	
1.6 Постановка задачі	29
Висновки по розділу	29
РОЗДІЛ 2. АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЕЛЕМЕНТІВ ДОРОЖНЬОГО РУХУ	30
2.1 Фаза розпізнавання елементів дорожнього руху	30
2.1.1 Алгоритм гістограми орієнтованих градієнтів (HOG)	30
2.1.2 Метод опорних векторів (SVM)	36
2.2 Фаза класифікації елементів дорожнього руху	41
2.3 Фаза відстеження елементів дорожнього руху	47
Висновки по розділу	52

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	53
3.1 Реалізація машинного навчання системи	53
3.2 Архітектура програмного забезпечення	58
3.4 Вимоги до технічних характеристик комп'ютерного забезпечення	59
3.5 Вимоги до програмного забезпечення	60
3.6 Результати експериментів	60
Висновки по розділу	63
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ	64
4.1 Опис ідеї проекту	64
4.2 Технологічний аудит ідеї проекту	65
4.3 Аналіз ринкових можливостей запуску стартап-проєкту	66
4.4 Розроблення ринкової стратегії проекту	72
4.5 Розроблення маркетингової програми стартап-проєкту	74
Висновки по розділу	76
ВИСНОВКИ	77
ПЕРЕЛІК ПОСИЛАНЬ	79
ДОДАТКИ	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ADAS (advanced driver-assistance systems) – це електронні системи, що допомагають водію транспортного засобу під час руху;

NHTSA (National Highway Traffic Safety Administration) – американська національна адміністрація безпеки автомобільних доріг;

GPS (Global Positioning System) – сукупність радіоелектронних засобів, що дозволяє визначати положення та швидкість руху об'єкта на поверхні Землі або в атмосфері;

FPS (Frames per Second) – характеристика відео зйомки, кількість записаних кадрів в секунду.

ROI (Region of Interest) – область інтересу, термін використовується для позначення області на картинці де ймовірно знаходиться об'єкт, який шукається;

API (application programming interface) – програмний інтерфейс;

IT – інформаційні технології.

ВСТУП

Розширені системи допомоги водіям (ADAS) – це технології, спрямовані на те, щоб надати водіям критичну інформацію про середовище дорожнього руху, взяти під контроль деякі складні або повторювані завдання та підвищити загальну безпеку водіїв та пішоходів. За даними Американської національної адміністрації безпеки автомобільних доріг (NHTSA)[1], головна причина (94%) аварій транспортних засобів – це людська помилка. Серед усіх можливих типів помилок які залежать від водія, найчастішими та критичнішими є помилки прийняття рішень в аварійній ситуації, маленька швидкість реакції в стресовій ситуації та проблеми з розпізнаванням елементів дорожнього руху. На основі цих даних можна зробити висновок, що людство потребує технології, які б могли в автоматичному режимі уникати аварій або ж надавати водію достатньо інформації, щоб попередити аварію.

На сьогоднішній день, автомобілі із вдосконаленими системами допомоги водіям є дуже поширеними сьогодні. За останні два десятиліття було розроблено багато систем допомоги водіям. Наприклад, навігація по GPS (Глобальній системі позиціонування) – це найпопулярніша технологія, яка існує з 1990-х років. Існують також інші системи допомоги водіям, які були розроблені в останні роки, включаючи:

- адаптивний круїз-контроль,
- адаптивні фари ближнього світла з автоматичним регулюванням кута нахилу для уникання осліплення інших водіїв,
- автоматична система екстреного гальмування,
- автоматичні системи паркування автомобіля,
- системи уникнення зіткнень,
- системи виявлення автомобілів, які знаходяться в «сліпій зоні»,
- система виявлення сонливості водія,
- система контролю спуску з пагорба, та ін.

Мета цих систем – це зробити дороги безпечнішим місцем для транспортних засобів та пішоходів. Однак ці системи практично не приділяють уваги чинникам які впливають на дії та рішення водія, а саме – дорожнім знакам, та іншим елементам дорожнього руху. Дуже часто водії, рухаючись по, уже знайомими, ділянками дороги не приділяють уваги дорожнім знакам, що може спричинити аварію у випадку зміни схеми руху. У випадку, якщо водій не помітив деякі дорожні знаки, це сильно впливає на його стиль водіння, що може призвести до аварії.

Зазвичай автомобілі з досконалими системами допомоги водіям також є дуже дорогими, що не кожна людина може дозволити собі купити такий автомобіль.

Дані обставини підкреслюють необхідність в інтелектуальних системах розпізнавання елементів дорожнього руху, задля інформування водія.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Об'єкт та предмет дослідження

Розробка системи розпізнавання знаків дорожнього руху є складним завданням. Є дуже багато факторів, які можуть стати причиною низької точності та повноти системи розпізнавання знаків дорожнього руху.

В загальному проблеми будь-якої системи розпізнавання об'єктів можна розділити на такі групи:

1. Недостатня освітленість об'єкту – є одною з найбільш основних проблем в розробці систем розпізнавання об'єктів, в тому числі і системи розпізнавання елементів дорожнього руху. Різні дорожні знаки мають різні кольори, для того щоб дорожній знак був більш помітний. Більшість існуючих алгоритмів розпізнавання елементів дорожнього руху використовують інформацію про колір об'єкту для ідентифікації знаків. Інформація про колір об'єкту є дуже чутлива до умов освітлення та віку знаку, відомо, що на сонці дорожні знаки мають властивість



висвітлюватись (див рис. 1.1).

Рисунок 1.1. Справа – приклад висвітленого сонцем знаку на трасі E40 (Київ-Чоп), для порівняння по ліву частину дороги – новий знак

2. Деформовані або пошкоджені знаки – є проблемою для сучасних алгоритмів розпізнавання елементів дорожнього руху, оскільки в алгоритмах використовується інформація про форму об'єкту, що може призвести до неправильної класифікації або ж взагалі до того, що знак не буде розпізнано (див. рис. 1.2). Також бувають випадки коли знак частково закритий іншими об'єктами, наприклад – листям або деревом.



Рисунок 1.2. Приклад деформованого дорожнього знаку

3. Також важливою проблемою є оптичні ефекти, розмитість та висвітленість, що мають місце в поганих погодних умовах, що можуть зменшувати точність то повноту алгоритму розпізнавання та ідентифікації. Цей ефект також має місце в сучасних системах, оскільки для отримання зображення використовують камери, які під час руху автомобіля на високій швидкості можуть давати розмиті фотографії. Для зменшення цієї проблеми використовують камери з великою

характеристикою FPS (Frames per seconds). Наприклад камери з можливістю зйомки 240 кадрів в секунду.

4. Встановлення зони розміщення знаку на картинці. Деякі конструкції, які є дуже схожі на елементи дорожнього руху можуть спричинювати проблеми для сучасних алгоритмів при розпізнаванні дорожніх знаків. Наприклад рекламні щити схожої форми можуть розпізнаватись як дорожні знаки.

5. Погана видимість. Тінь від інших машин призводить до спотворення картинки при розпізнаванні, наприклад, лінії руху. Такі природні явища як дощ, сніг та туман також є серйозними проблемами для сучасних алгоритмів розпізнавання елементів дорожнього руху.

Багато існуючих методів пасивної безпеки автомобілів є застарілими, а алгоритми, які в них використовуються – неефективними. Існуючі системи пасивної безпеки є дуже дорогими, а їхні алгоритми запатентовані та у закритому доступі.

1.2 Способи розпізнавання елементів дорожнього руху

За останній час було запропоновано багато різних методів розпізнавання елементів дорожнього руху. Зазвичай вони складаються з двох послідовних процесів:

- Етап розпізнавання області інтересу, тобто області на картинці, де ймовірно може бути елемент дорожнього руху.
- Етап ідентифікації елемента, тобто встановлення типу елемента, наприклад дорожній знак, а згодом, встановлення конкретно який це знак.

Основною задачею розпізнавання області інтересу – це виділення певних регіонів з картинки, які найбільш ймовірно є дорожніми знаками. Для вирішення цієї задачі можуть використовуватись різні підходи, які використовують основні характеристики елементів дорожнього руху, такі

як форма і колір. В загальному методи розпізнавання області інтересу можна розбити на три групи:

- розпізнавання об'єкта за допомогою кольору;
- розпізнавання об'єкта за допомогою форми;
- розпізнавання об'єкта гібридними методами.

1.2.1 Розпізнавання об'єкта за допомогою кольору

Використання інформації про колір об'єкту є одним із найпоширеніших підходів до розпізнавання елементів дорожнього руху. Ця інформація знаходиться за допомогою порогових або розширених методів сегментації і використовується для пошуку областей, де найбільш ймовірно може бути дорожній знак. Однак головним недоліком цього методу – є висока чутливість до нестабільних умов освітлення, тому цей метод є дуже залежним від часу доби та погодних умов. У сучасних системах цей метод використовується з різними типами кольорових просторів, найбільш поширеними є RGB, HSI/HSV, YUV, YCbCr, CIE Lab та CIECAM97.

Методи розпізнавання об'єктів на основі характеристик мають високу швидкодію, хорошу стійкість та інші характеристики, які можуть певною мірою поліпшити продуктивність розпізнавання об'єктів на картинці. В таких методах розпізнавання об'єктів зазвичай використовуються кольорові картинки високої роздільної здатності, але, зазвичай, такі методи дають низьку точність на чорно-білих картинках.

В таких підходах до розпізнавання елементів, вхідні картинки розглядають як набір пікселів, в яких кожен піксель має певні характеристики, які визначають його колір, потім набір пікселів ділять на окремі ділянки в яких сусідні пікселі мають схожі характеристики кольору. Потім знаки дорожнього руху виявляють за допомогою порогування кольору та алгоритмом розпізнавання. В таких підходах велику роль відіграє вибір простору кольору, тобто рівняння, яке

визначає характеристики кольору пікселя. Вхідні картинки системи розпізнавання дорожніх знаків зазвичай трансформують у вибраний простір кольору, де елементи, що шукаються, більш виразні. Згідно з роботою [2], існуючі алгоритми розпізнавання дорожніх знаків використовують наступні простори кольорів:

- RGB (червоний, зелений, синій),
- HSV (відтінок, насичення, значення),
- HSI (відтінок, насичення, інтенсивність),
- та багато інших просторів кольорів.

Найбільш поширеними алгоритмами розпізнавання елементів дорожнього руху з використанням інформації про колір є:

- алгоритм порогування кольору,
- алгоритм region growing,
- алгоритм індексування кольору,
- алгоритм динамічної агрегації кольору.

Алгоритм порогування кольору є вже давно відомою технікою, яка використовується для сегментації зображення. Загальна ідея цієї техніки – це припущення, що суміжні пікселі, значення яких (рівень сірого кольору, значення кольору, текстури тощо) лежить у певному діапазоні, належать до одного класу. Кольорова сегментація була використана в роботі Varun et al. [3] на власному наборі тестових даних, близько 2000 зображень, що дало точність розпізнавання лише 82%. Ефективність даного підходу була вдосконалена в роботі [4] завдяки використанню сегментації кольору за якої слідувало посилення кольору.

Останнім часом, порогування кольору – є дуже поширеним алгоритмом для попередньої обробки зображення [5, 6]. В роботі [6] Vazquez-Reina використав перехід з простору кольорів RGB в простір HSI, в якості додаткового параметру для класифікації дорожніх знаків на білому фоні. В роботах [7, 8] простори кольорів HSI/HSV були використані для виявлення дорожніх знаків. Основна перевага простору

кольорів HSI над простору кольорів RGB – це те, що в просторі кольорів HSI є лиш два компонента, які задають колір – це відтінок і насичення, які дуже схожі на сприйняття людиною, і вони більш незалежні від умов освітлення, в порівнянні з RGB.

У роботі [7] перехід з простору кольору RGB в HSI використовується для цілей TSDR. У роботі [9] простір кольорів HSI був використаний для виявлення елементу, а згодом отримані характеристики використовували у для розпізнавання форми об'єкту, для виявлення форми для підвищення рівня точності. Середня точність склала приблизно 88,4%. Основне обмеження використання трансформації HSV – це сильна залежність відтінку від яскравості зображення. Відтінок – це лише фізична характеристика яскравості кольору, а не сприйнятої яскравості зображення. Таким чином, значення повністю насиченого жовтого та синього кольорів однакове.

Region growing – це ще один простий і популярний алгоритм, що використовується для розпізнавання елементів в системах TSDR. Region growing – це метод сегментації зображення на основі пікселів, який починається з вибору початкової точки або початкового пікселя. Потім область розширюються шляхом додавання сусідніх пікселів, які є рівномірними, за певним критерієм відповідності, збільшуючи крок за кроком розмір області. Цей метод використовували Nicchiotti et al. [10] та Priese et al. [11] для TSDR. Його ефективність була не дуже високою, приблизно 84%. Оскільки ефективність цього методу залежить від початкових даних, а саме від вибору початкового пікселя. Проблеми можуть виникнути, коли початкові пікселі лежать на краях (якогось об'єкту), невизначеність навколо країв сусідніх областей може бути усунена неправильно.

Алгоритм індексації кольорів – це ще один простий метод, який ідентифікує об'єкти повністю на основі кольору. Він був розроблений Суеном і Баллардом [12] і був використаний дослідниками на початку

1990-х. У цьому методі порівняння будь-яких двох кольорових зображень проводиться шляхом порівняння їх гістограми кольорів. Для заданої пари гістограм I і M , кожна з яких містить n бінів, тоді перетини гістограми визначаються як (формула 1.1):

$$\sum_{j=1}^n \min(I_j, M_j). \quad (1.1)$$

Значення відповідності тоді:

$$H(I, M) = \frac{\sum_{j=1}^n \min(I_j, M_j)}{\sum_{j=1}^n M_j}. \quad (1.2)$$

Перевагою використання кольорових гістограм є їх надійність щодо геометричних змін проєктованих об'єктів. Однак індексація кольорів залежить від сегментації, і повна, ефективна та надійна сегментація не може бути виконана до розпізнавання. Таким чином, кольорова індексація характеризується як ненадійний метод.

Іншим підходом до сегментації кольорів є – алгоритм динамічної агрегації пікселів. У цьому способі процес сегментації здійснюється шляхом введення динамічного порогу до процесу агрегації пікселів у кольоровому просторі HSV. Застосований поріг незалежний з точки зору лінійності і його значення визначається як:

$$a = k - \sin(S_{seed}), \quad (1.3)$$

де, k – параметр нормалізації, а S_{seed} – насичення пікселів. Основна перевага такого підходу – це зменшення нестабільності відтінку. Однак це не вирішує інші проблеми сегментації зображення, такі як згасання та освітлення. Цей метод був випробуваний на власній базі даних з 620

зображеннями, в результаті чого рівень точності був приблизно від 86,3 до 95,7%.

Таблиця 1.1. Порівняння технік розпізнавання елементів за кольором

Техніка	Метод сегментації	Переваги	Точність
Порогування кольору	RGB сегментація кольору	Простота	~82%
	RGB сегментація з посиленням кольору	Швидкодія та відносно висока точність розпізнавання	
HSI/HSV трансформація	HSI порогування як додаткова характеристика для білих знаків	Сегментує погано освітлені знаки	~88.4 %
	HSI кольорова сегментація	Простота та швидкодія	
	Перехід від RGB до HSI простору кольорів	Простота та висока точність розпізнавання	
Region growing	Розширення області від початкових пікселів в залежності від фактору подібності		84%
Індексування кольору	Порівняння двох зображень на основі гістограми кольорів	Швидкодія	
Алгоритм динамічної агрегації кольору	Динамічне порогування кольору в HSV просторі кольорів	Зменшена нестабільність відтінку	~86.3 % - 95.7%

1.2.2 Розпізнавання об'єкта за допомогою форми

Ще одною важливою характеристикою елементів дорожнього руху – є форма (кругла, трикутна, восьмикутна або ж прямокутна). В порівнянні з методом розпізнавання об'єкту за допомогою кольору, даний метод не має залежності від погодній умов і змін яскравості завдяки тому, що не використовує інформацію про колір об'єкту, тому багато сучасних систем використовують інформацію про форму об'єкту для його ідентифікації.

Для розпізнавання знаків дорожнього руху за формою знаку використовують різні алгоритми розпізнавання форми, наприклад пошук контурів і їх наближення для отримання остаточного рішення, виходячи з кількості контурів. Даний метод також може використовуватись в якості детектора дорожніх знаків, завдяки тому, що він дозволяє знаходити області інтересу з вхідної картинки, а потім їх окремо розглядати для подальшої ідентифікації знаку. Проте, такий підхід може бути затратним по пам'яті, якщо вхідна картинка має велику роздільну здатність.

Існує багато різних алгоритмів розпізнавання форми елемента, такі як:

- Трансформація Хафа;
- Similarity Detection;
- Distance Transform Matching;
- Виявлення характеристик контурів;
- Haar-Like Features.

Найбільш поширеним алгоритмом є – Трансформація Хафа. Алгоритм Хафа може використовуватись для розпізнавання ліній, кіл, прямокутників чи інших кривих. Цей метод вперше був введений в 1962 році, і його першим застосуванням було пошук ліній в послідовності зображень. Основна перевага Трансформації Хафа – це його стійкість до

шумів, масштабування та обертання. Автори в [13] використовували Трансформацію Хафа для кіл для розпізнавання круглих дорожніх знаків, а для розпізнавання трикутних знаків автори використали Трансформацію Хафа для розпізнавання прямих ліній.

Алгоритм базується на основі представлення елемента, якого ми шукаємо у вигляді деякого параметричного рівняння. Масив параметрів рівняння називають простором Хафа, для кожної точки вхідного зображення розраховується параметричне рівняння і отримуються параметри, які зберігаються в просторі Хафа. Фінальним кроком алгоритму є обхід простору Хафа і вибір максимальних значень, за яке «проголосувало» найбільша кількість пікселів картинки, що дає нам параметри для рівняння пошукового елемента. В основі алгоритму Хафа лежить ідея, що будь-яка точка зображення може бути частиною деякого набору можливих ліній.

Similarity Detection – це один алгоритм розпізнавання форми об'єкту. Алгоритм базується на пошуку фактору подібності між набором вже відомих зображень з вхідним зображенням. При цьому вхідна картинка повинна бути добре сегментована та мати таку ж роздільну здатність. Цей метод був використаний в одній праці, де на своєму наборі зображень знаків дорожнього руху, де автор отримав точність розпізнавання 86.3%. Основна перевага даного методу це простота і пряmolінійність, в той час як недолік цього методу це строгі обмеження вхідного зображення, яке повинне бути правильно сегментовано та зжато до потрібного розміру, який зазвичай такий же, як зображення в навчальному наборі даних. Рівняння функції подібності можна записати у вигляді:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \cdot n, \quad (1.4)$$

$$y' = \frac{y - y_{min}}{y_{max} - y_{min}} \cdot n, \quad (1.5)$$

де, x_{max} , y_{max} , x_{min} , y_{min} – координати вершин прямокутника

Distance Transform Matching – ще один алгоритм розпізнавання елементу за допомогою форми. В алгоритмі, кожному пікселю, який не є граничним, розраховують відстань до найближчого кутового пікселю (який ймовірно є границею об'єкту), далі за допомогою обчислених параметрів отримують параметри для параметричного рівняння. Алгоритм є схожим до Similarity Detection, бо отримані відстані від не граничних пікселів до найближчих кутових пікселів є обернено пропорційні відповідності вхідного зображення та уже відомим шаблонам (навчальному набору).

Виявлення характеристик контурів – одна з основних технік обробки зображень для знаходження меж різних предметів на зображенні. Даний метод найбільше застосовують в системах обробки зображення та в комп'ютерному зорі для сегментації, коли зображення розділяють на ділянки відповідні різним об'єктам. Наприклад, детектор Кенні був використаний в роботі [4], оскільки цей метод зберігає контури, що є необхідним для детекторів дорожнього руху.

Ще одним методом виявлення форми об'єкту – є метод виявлення кутових точок. Пауло і Коррея [6] використовували кутовий детектор Харріса для виявлення дорожніх знаків. Вони розпізнавали трикутні та квадратні форми знайшовши кути для кожної області інтересу (ROI), використовуючи кут Харріса. Потім існування кутів перевіряється в шести різних областях управління ROI. Кутовий детектор Харріса вперше був введений у 1988 році Крісом Харрісом та Майком Стівенсом [5]. Цей кутовий детектор став популярним, тому що він простий і швидкий. Більше того, кутовий детектор Харріса не залежить від масштабу, обертання та освітлення.

1.2.3 Розпізнавання об'єкта гібридними методами

Як відомо методи розпізнавання елементів дорожнього руху за допомогою кольору та форми мають свої переваги та недоліки. Тому багато систем використовують різні комбінації цих методів для досягнення максимальної точності розпізнавання.

Багато систем розпізнавання елементів дорожнього руху включають в себе крок сегментації кольору, за який слідує етап розпізнавання форми елемента. Фанга в своїй роботі [9] використав характеристику відтінку кольору та метод виявлення кутових точок для розпізнавання форми елемента. [9] є ще одним прикладом використання композиції методів розпізнавання елементів за допомогою кольору та форми.

1.3 Методи класифікації елементів дорожнього руху

Алгоритми класифікації елементів дорожнього руху використовуються після локалізації ROI для подальшого визначення вмісту виявлених дорожніх знаків. Розуміння правила дорожнього руху, що задається дорожнім знаком, досягається за допомогою зчитування внутрішньої частини виявленого дорожнього знаку за допомогою методу класифікатора. Алгоритми класифікації не ґрунтуються ні на кольорі, ні на формі. Класифікатор зазвичай приймає певний набір параметрів на вхід, за якими відрізняє кандидатів один від одного. Для класифікації дорожніх знаків використовуються різні алгоритми в яких є свої недоліка та переваги.

Методи класифікації об'єктів:

- метод відповідності шаблонів,
- дерево вибору,
- генетичний алгоритм,
- нейроні мережі,
- AdaBoost класифікатор,

- Support Vector Machine (SVM).

Метод відповідності шаблонів є загальним методом обробки зображень та розпізнавання шаблонів. Це підхід низького рівня, який використовує заздалегідь задані шаблони для пошуку зображення піксель за пікселем. Основна перевага даного підходу – це швидкодія та точність (максимально отримана точність було 90% на власному наборі даних). Однак недоліком цього методу є те, що він дуже чутливий до шуму та оклюзій. Крім того, для даного методу потрібен окремий шаблон для різних розмірів зображення об'єкта та орієнтації.

Інший поширений метод класифікації – дерево вибору. Це метод машинного навчання, який функціонує шляхом побудови безлічі дерев рішень протягом навчального. Цей метод порівнювали з класифікаторами на основі SVM, MLP, класифікаторами на основі гістограми орієнтованого градієнта (HOG), показуючи найвищу швидкість точності та найменший час обчислення. На основі власного набору даних точність становила приблизно 94,2%, тоді як точність SVM становить 87,8%, а MLP – 89,2%. Щодо обчислювального часу для однієї класифікації, SVM займає 115,87 мс, MLP займає 1,45 мс, а дерево рішень займає 0,15 мс. Незважаючи на високу точність та низький час обчислення, головне обмеження методу дерева рішень полягає в тому, що велика кількість дерев може зробити алгоритм повільним і неефективним для прогнозування в реальному часі.

Генетичний алгоритм – це ще один метод класифікації об'єктів. Він заснований на природному процесі відбору, що імітує біологічну еволюцію, яка була використана на початку цього століття. У цих дослідженнях було доведено, що цей метод ефективний для виявлення дорожнього знаку, навіть якщо дорожній знак має певні деформації форми чи недостаток освітленості. Недоліком генетичного алгоритму є недетермінований час роботи та негарантоване знаходження найкращого рішення.

Інший найпоширеніший метод класифікації – це використання штучної нейронної мережі (ANN). Цей метод завоював все більшу популярність в останні роки завдяки просуванню в обчисленні загального призначення технологій графічної обробки (GPGPU). Крім того, він популярний завдяки своїй надійності, більшій адаптивності до змін, гнучкості та високій швидкості та точності. Ще однією ключовою перевагою цього методу є його здатність розпізнавати та класифікувати об'єкти одночасно, зберігаючи високу швидкість та точність. В одному експерименті, швидкість потрапляння становила 97,6%, а обчислювальний час $\sim 0,2$ с. Однак у методах на основі ANN були описані деякі обмеження, такі як їх повільність та нестабільність у навчанні NN через занадто великий крок. Цей метод порівнювали з методом відповідності шаблонів, роблячи висновок, що для NN потрібна велика кількість навчальних зразків для реальних програм.

1.4 Методи відстеження елементів дорожнього руху

Для надійного виявлення та для підвищення точності інформації, що використовується для ідентифікації дорожніх знаків, знаки відстежуються за допомогою простої моделі руху та часового поширення інформації. Цей процес відстеження є дуже важливим для програм, які працюють у режимі реального часу, за допомогою яких система TSDR перевіряє правильність знаку дорожнього руху та продовжує відстежувати знак, щоб уникнути повторної обробки одного й того ж раніше виявленого знаку. Процес відстеження здійснюється постійно при подачі відео від системи TSDR, записаним камерою, закріпленою на транспортному засобі. Відео розбивається на кадри, які є вхідними зображеннями для відповідних алгоритмів. Прийнятими кандидатами на знак є лише ті, хто з'являвся не один раз. Якщо об'єкт не є дорожнім знаком або знаком, який відображається лише один раз, його можна позбутися якнайшвидше, для

зменшення часу обчислення. Найпоширенішим адаптованим трекером – є фільтр Кальмана.

1.5 Огляд існуючих рішень

Схожі інтелектуальні системи використовуються в більшості автомобільних виробників, таких як Volkswagen, Mercedes, BMW, Tesla Motors, але вони є в закритому доступі, оскільки є власними розробками та є дуже дорогими. Тому зробити детальний аналіз та огляд існуючих рішень неможливий для таких комерційних систем.

Оскільки одна з цілей даної роботи це здешевлення інтелектуальної системи допомоги водіям, то в цьому розділі брались до уваги лише системи, які є у відкритому доступі.

1.5.1 Таһа Khan система розпізнавання знаків дорожнього руху

Дана система була запропонована в 2013 році та складається з декількох фаз:

- етап створення відео-потoku (з використанням камери);
- етап попередньої обробки зображення, який обов'язковий для наступних фаз
- Перехід з простору кольорів RGB в простір HSV;
- застосування фільтру для усунення шумів;
- масштабування зображення до потрібних розмірів;
- фаза розпізнавання елементів, яка використовує метод відповідності шаблонів;
- фаза відстеження елементів.

В загальному робота системи представлена на рис. 1.3.

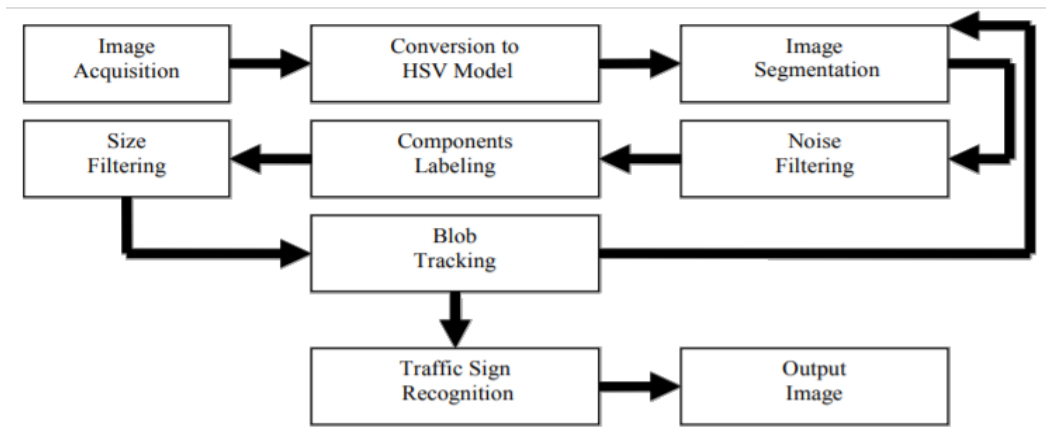


Рисунок 1.3. Робота системи розпізнавання знаків Тааа Кхан

Переваги:

- дуже висока швидкість роботи системи, завдяки дуже простому алгоритму класифікації – методом відповідності шаблонів;
- легка реалізація.

Недоліки:

- досить низька точність розпізнавання елементів на картинці де є інші об'єкти;
- нестійкість до проблем з освітленням та інших оптичних ефектів;
- нестійкість до проблем з деформованими знаками;
- кожен екземпляр даних потрібно включати для тренування системи, обертаючи його, для забезпечення інваріативності до розпізнавання повернутих знаків;
- обов'язковість попередньої обробки зображення, яке подається на вхід системі.

1.5.2 Boris Karamatic система розпізнавання знаків дорожнього руху.

Основною особливістю запропонованої системи є – використання нейронних мереж. Система складається з 12 шарів, вихідні параметри кожного шару є вхідними параметрами для наступного шару. Ще одною особливістю системи є те, що фаза розпізнавання та класифікації елементів є частинами нейронної мережі та виділені в окремі шари. Система складається з наступних шарів:

- вхідних шар, який приймає картинку 32*32 пікселів, тобто вектор 32*32*3, де 3 – значення RGB кольору пікселя;
- convolutional 2D шар;
- maximum pooling 2D шар, основна ціль якого це зменшення кількості вхідних параметрів (для пришвидшення роботи алгоритму);
- шар нормалізації;
- dropout шар, який використовується тільки при навчанні мережі;
- Fully connected шари, де останній шар складається з кількості нейронів, що рівна кількості дорожніх знаків в тестовому наборі даних.

Загалом робота системи представлена на рис. 1.4.

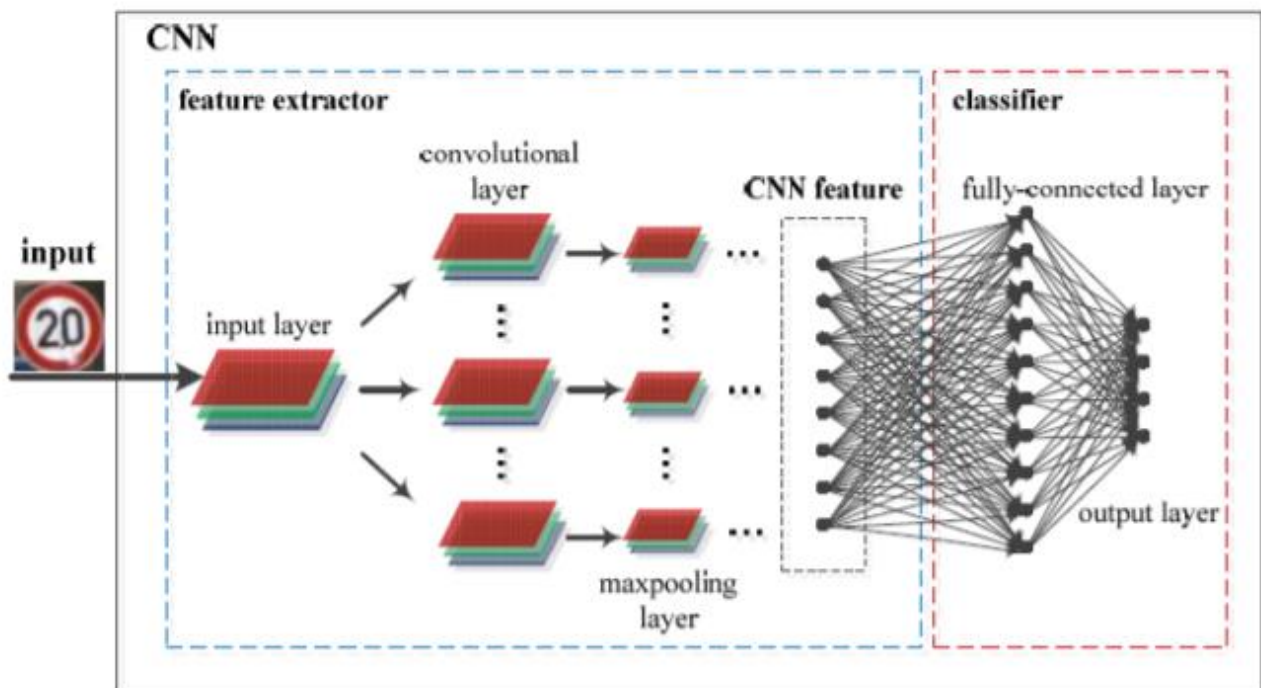


Рисунок 1.4. Робота системи розпізнавання знаків Boris Karamatic

Переваги:

- досить висока точність розпізнавання об'єктів;
- можливість розширення та додавання нових шарів в мережу;
- стійкість до оптичних ефектів завдяки використанні інформації про форму об'єкту.

Недоліки:

- середня швидкість роботи системи;
- сильна залежність від даних, на яких система була навчена.

1.6 Постановка задачі

Метою роботи є покращення точності та оптимізація розпізнавання елементів дорожнього руху, а саме дорожніх знаків в режимі реального часу. Для досягнення мети необхідно вирішити наступні задачі:

1. Аналіз існуючих алгоритмів для розпізнавання елементів на зображенні.
2. Розробка архітектури інтелектуальної системи розпізнавання елементів дорожнього руху в режимі реального часу.
3. Програмна реалізація інтелектуальної системи.

Створена система повинна відповідати наступним вимогам:

- точність розпізнавання елементів дорожнього руху повинна бути більше 95%;
- знайдені елементи повинні бути інтуїтивно виділені серед інших елементів;
- можливість використання відео файлу в якості вхідного відео для системи;
- простота у налагодженні.

Висновки по розділу

У першому розділі проведено аналіз існуючих рішень та предметної області, проаналізовані основні сучасні алгоритми та підходи для розпізнавання елементів на картинці, визначено переваги та недоліки цих рішень. В результаті проведеного аналізу сформульована постановка задачі, наведене призначення, цілі та задачі розробки.

РОЗДІЛ 2. АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЕЛЕМЕНТІВ ДОРОЖНЬОГО РУХУ

У цій главі ми детально розглянемо архітектуру запропонованої системи, яка складається з трьох основних етапів. Та розглянемо основну концепцію, яка була взята за основу. Як було сказано раніше, більшість систем розпізнавання елементів складається з трьох основних етапів:

1. Розпізнавання елементів на зображенні та встановлення ROI.
2. Класифікація дорожніх знаків.
3. Відстеження раніше класифікованих об'єктів.

2.1 Фаза розпізнавання елементів дорожнього руху

Запропонована інтелектуальна система розпізнавання елементів дорожнього руху буде використовувати алгоритм HOG (Гістограма орієнтованих градієнтів) та класифікатор SVM.

У цьому розділі ми надаємо більше інформації щодо алгоритмів, що використовуються на етапі виявлення, таких як функції HOG та класифікатор SVM.

2.1.1 Алгоритм гістограми орієнтованих градієнтів (HOG)

Загальновизнано, що для виявлення об'єктів дуже добре підходить алгоритм гістограми орієнтованих градієнтів (HOG). Характеристики алгоритму засновані на оцінці добре нормалізованих локальних гістограм орієнтацій градієнта зображення в щільній сітці. Цей метод вперше застосовували Навнеет Далал та Білл Триггс у 2005 році. Вони використали його для завдання виявлення пішоходів. З тих пір, багато дослідників використовували HOG дескриптори для різних додатків в області обробки зображень та комп'ютерного зору. Такі види характеристик мають відмінні показники порівняно з іншими алгоритмами. Ми можемо розділити реалізацію алгоритму дескрипторів HOG на чотири різні етапи, а саме: нормалізацію зображення, розрахунок

градієнта, встановлення орієнтації елемента та блоки нормалізації. На рис. 2.1 представлена схема реалізації алгоритму.

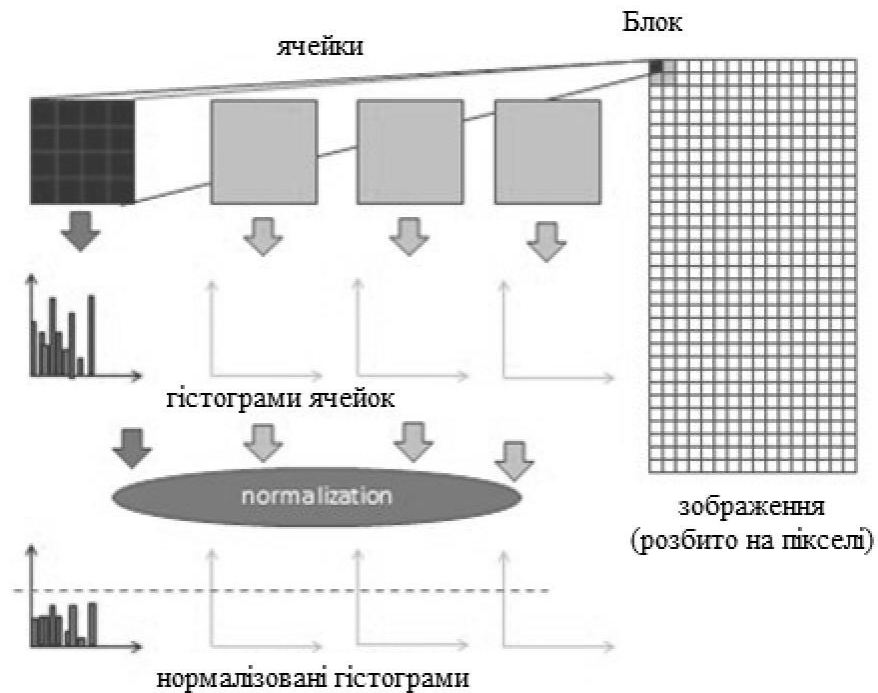


Рисунок 2.1. Схема реалізації HOG алгоритму

Першим кроком вилучення характеристик об'єкту за допомогою HOG є нормалізація зображення. Для нормалізації зображення використано вирівнювання зображення, щоб зменшити вплив непослідовного освітлення. Це вирівнювання виконується за допомогою гамма корекції.

Гамма корекція, або часто, просто гамма, – це нелінійна операція, що використовується для кодування та декодування значень яскравості у системах відео або нерухомих зображень. Корекція гамми в найпростіших випадках визначається наступним виразом:

$$V_{out} = A * V_{in}^{\gamma} \quad (2.1)$$

де V_{in} – вхідне значення підноситься до степені γ і множиться на деяку константу A , яка в більшості випадках рівна 1, вхідні значення та вихідні зазвичай в межах від 0 до 1.

На практиці гамма корекція використовується для висвітлення або затемнення зображення, при застосування даного рівняння з параметром

$\gamma > 1$ – вихідне зображення бути більш затемненим, а при $\gamma < 1$ – більш висвітленим, як показано на рис. 2.2.



Рисунок 2.2. Приклад використання гамма компресії

Другий етап складається з обчислення градієнтів зображення першого порядку. Мета цього кроку – захоплення деякої інформації про контури та текстури, а також надання більшої стійкості до невідповідних умов освітлення. Найпоширенішим методом обчислення градієнтів зображення є використання одновимірного пікселя та розрахунок градієнту опираючись на сусідні пікселі по вертикалі та горизонталі. Наприклад, градієнт пікселя (x, y) можна визначити наступним чином:

$$G(x) = H(x + 1, y) - H(x - 1, y), \quad (2.2)$$

$$G(y) = H(x, y + 1) - H(x, y - 1), \quad (2.3)$$

де, $H(x, y)$ величина сірого кольору пікселя (x, y) і $G(x)$ та $G(y)$ – відповідно градієнти. Розрахунки роблять для всіх каналів кольору зображення, наприклад якщо піксель задається простором кольору RGB, то розрахунки проводяться для трьох каналів кольору: червоний, зелений, синій. Приклад розрахунку градієнтів на рис. 2.3.

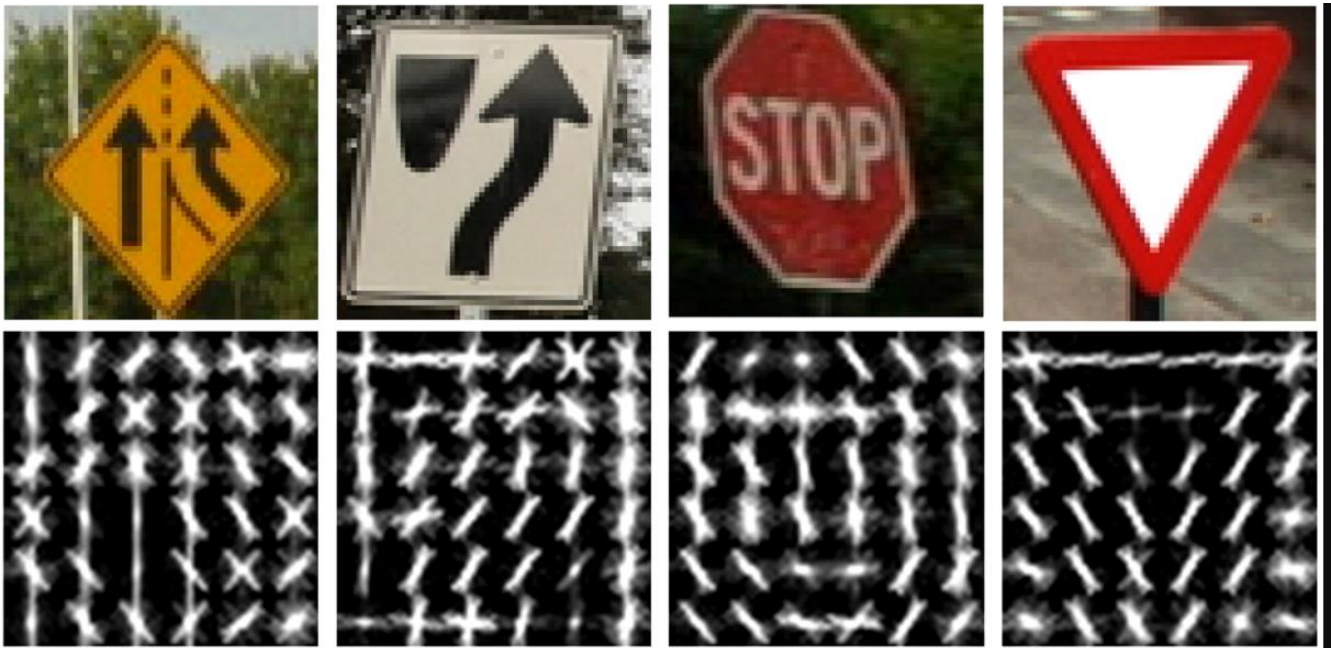


Рисунок 2.3. Приклад порашований градієнтів на зображені 36x36 пікселів

На кожному пікселі градієнт має величину та напрямок. Величина градієнта в пікселі – це максимальна величина градієнтів трьох каналів, а кут – це кут, відповідний максимальному градієнту.

Третій крок – це створення гістограм пікселів. На цьому етапі зображення розділяється на 8×8 комірок і розраховується гістограма градієнтів для кожної клітини 8×8 . Гістограма зазвичай є вектором (або масивом) з 9 елементів (цифр), що відповідають кутам 0, 20, 40, 60... 160. На рис. 2.4 показано приклад розрахунку гістограми пікселів, на середній картинці стрілки показують напрямок градієнта, а довжина стрілки – його величину. Напрямок порашованих значень напрямів градієнтів становлять від 0 до 180 градусів. Такий спосіб називають "непідписаними" градієнтами, тому що градієнт і його протилежне значення представлені

однаковими числами. Іншими словами, стрілка градієнта та її протилежне, на 180 градусів, значення вважаються однаковими.

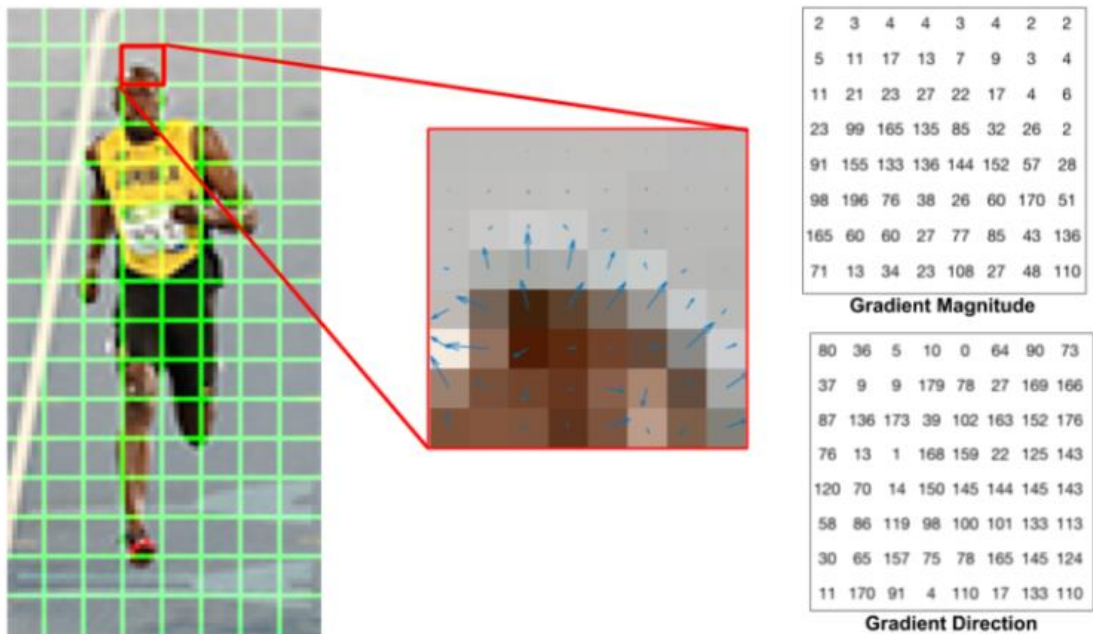


Рисунок 2.4. Приклад гістограми клітинок

Останній крок полягає в нормалізації гістограм. Нормалізація приносить більше інваріантності змінам освітлення, затіненню та контрастності. Оскільки діапазон варіацій градієнта великий, ефективна локальна нормалізація контрасту є критичною для підвищення продуктивності системи. Запропонована схема нормалізації ґрунтується на групуванні комірок у більші блоки, з наступним кроком нормалізації контрасту кожного блоку окремо.

Отже, кінцевий дескриптор HOG – це вектор, що складається з усіх клітинних гістограм. Існують основні геометрії блоку: прямокутна (R-HOG) і кругла (C-HOG). Блоки R-HOG, як правило, представлені трьома параметрами: кількістю комірок у кожному блоці, кількістю пікселів у кожній комірці та кількістю каналів кольору у гістограмі кожної комірки. Dalaland Triggs виявив, що клітини 8x8 пікселів на блок з 9 каналами гістограми є оптимальними параметрами для завдання виявлення пішоходів. Незважаючи на деяку схожість між блоками R-HOG та дескрипторами SIFT, вони використовуються по-різному. Блоки R-HOG

обчислюються в щільних сітках в одній шкалі без домінуючого вирівнювання орієнтації, тоді як дескриптори SIFT обчислюються за наперед встановленим набором ключових точок інваріантних масштабів, та різних варіантів повернутих зображень, та використовуються окремо.

Інша форма блоку – кругла. Дескриптори C-HOG можна знайти у двох різних варіантах: в одній центральній комірці круглої форми та у центральній комірці з кутовим поділом. Блоки C-HOG представлені чотирма параметрами: кількістю кутових та радіальних комірок, радіусом центральної комірки та коефіцієнтом розширення радіусу додаткових радіальних комірок. Для досягнення оптимальної продуктивності необхідні дві радіальні комірки та чотири кутові комірки.

Існує 4 методи для нормалізації R-HOG і C-HOG блоків:

$$L_{1-sqrt}: f = \sqrt{\frac{v}{(|v|_1 + e)}}, \quad (2.4)$$

$$L_{1-norm}: f = \frac{v}{(|v|_1 + e)}, \quad (2.5)$$

$$L_{2-norm}: f = \frac{v}{\sqrt{(|v|_1^2 + e^2)}}, \quad (2.6)$$

$$L_{2-hys}: f = \frac{v}{\sqrt{(|v|_1^2 + e^2)}}, \text{ де } v \leq 0.2, \quad (2.7)$$

де v – не нормалізований вектор гістограми, наприклад R-HOG, e – константа, f – фактор нормалізації.

Після блокової нормалізації всі гістограми можуть бути об'єднанні в один вектор, який використовується для подальшого розпізнавання елементу.

Рис. 2.5 ілюструє приклади HOG характеристик, витягнутих із кількох знаків руху.



Рисунок 2.5. Приклад HOG характеристик порохованих з вхідних зображень дорожніх знаків

2.1.2 Метод опорних векторів (SVM)

Метод опорних векторів (SVM) входить до складу найкращих контрольованих алгоритмів машинного навчання. Контрольоване навчання – це один з способів машинного навчання, в ході якого випробовувана система примусово навчається за допомогою деякої наявної множини помічених даних, наприклад зображення та його тип. Такий спосіб відноситься до завдання машинного навчання, в знаходженні функції, на основі вхідних мічених даних. Алгоритм контрольованого навчання, оцінює вхідні дані навчання, щоб зробити модель функції, яку потім можна використати для класифікації нових, небачених прикладів. SVM – це дискримінаційна класифікація, формально визначена роздільною лінією. Тобто, для деякого навчального набору даних, цей метод дає оптимальну лінію, яка розділяє різні класи елементів, тим самим може класифікувати нові елементи.

Цей метод класифікації є високоточним та надзвичайно швидким для великої кількості навчальних даних, що і обумовило вибір в даній роботі.

Точніше, SVM – це бінарний класифікатор, який розділяє два різних класи елементів за допомогою підмножини зразків даних, які називаються опорними векторами. Виявлення пішоходів – це добре відоме застосування SVM в області комп'ютерного зору та машинного навчання. Наведемо приклад класифікації двох типів зразків у двовимірному просторі, як це проілюстровано на рис. 2.6. Квадрати та кола представляють два різних типи зразків.

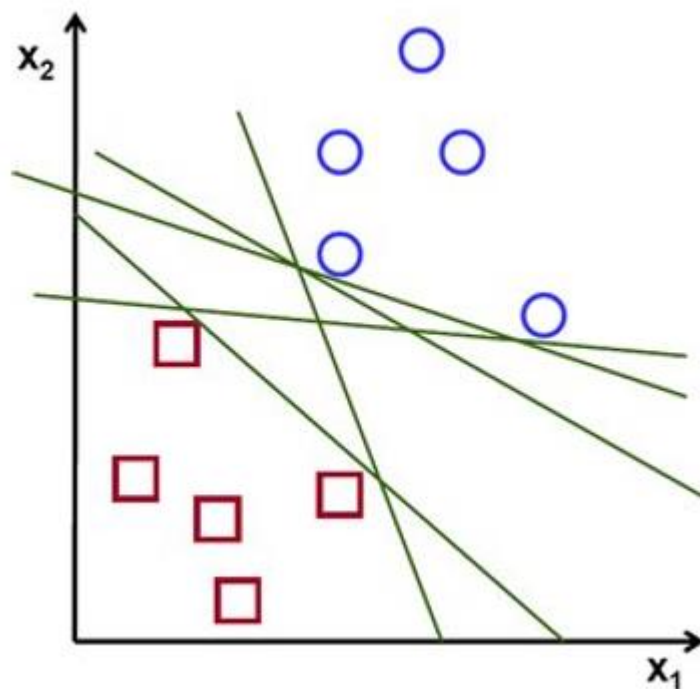


Рисунок 2.6. Приклад класифікації елементів у двовимірному просторі

Припустимо, що наші дані мають вигляд (\vec{x}_1, y) , ..., (\vec{x}_n, y_n) . Кожна пара представляє мітки -1 або +1; -1 для негативних зразків і +1 для позитивних зразків. Лінія поділу, яка використовується як функція прийняття рішення, може розділити ці два класи. Будь-яку лінію поділу можна визначити як набір точок \vec{x} , які задовільняють формулу 2.8:

$$\vec{w} * \vec{x} + b = 0, \quad (2.8)$$

де \vec{w} – вектор нормалі до лінії розподілу, і b – деяка константа, яка називається – упередженість.

Приклад з рис. 2.6 показує, як позитивні та негативні елементи розділені за декількома лініями поділу (або площинами, якщо розглядати у трьох вимірному просторі). Ми можемо вибрати дві паралельні прямі, щоб розділити наші дані. Відстань між двома граничними прямими робиться максимально можливою. Ці прямі можна визначити наступним чином (формули 2.9 та 2.10):

$$\vec{w} * \vec{x} + b = 1, \quad (2.9)$$

$$\vec{w} * \vec{x} + b = -1. \quad (2.10)$$

Відстань між двома граничними прямими є $\frac{2}{\|\vec{w}\|}$. Для того щоб максимально збільшити відстань між граничними лініями, ми повинні мінімізувати $\|\vec{w}\|$. Нам також потрібно знайти лінію, що лежить посередині між двома визначеними граничними лініями. Ми також можемо визначити критерій, щоб визначити, яка лінія є оптимальним. Якщо лінія проходить занадто близько до точок, вона буде чутливою до шуму, і це не буде вдалим вибором для класифікації двох класів. Тому наша мета – знайти лінію, яка є максимально далекою від усіх точок. Алгоритм SVM заснований на визначенні прямої, який дає найбільшу мінімальну відстань до навчальних даних. У теорії SVM ця відстань називається запасом. Отже, оптимальне розміщення прямої максимально збільшує відстань від тренувальних зразків. На рис. 2.7 показана ця оптимальна пряма.

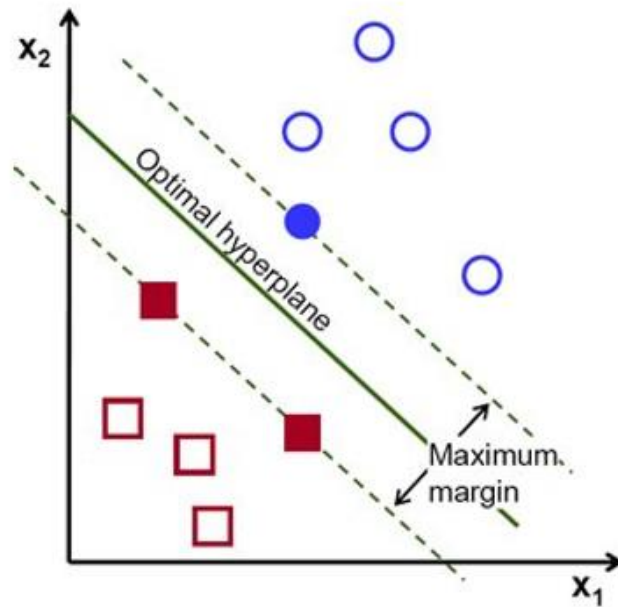


Рисунок 2.7. Оптимальна пряма, яка найкраще розділяє позитивні та негативні класи

Ще одна важлива річ у визначенні оптимальної прямої – це запобігання потрапляння точок даних у межу. Використовуючи наступні обмеження, ми переконуємося, що точки даних лежать на правильній стороні поля:

$$\vec{w} * \vec{x} + b \geq 1, \text{ якщо } y_i = 1, \quad (2.11)$$

$$\vec{w} * \vec{x} + b \leq -1, \text{ якщо } y_i = -1. \quad (2.12)$$

Ці обмеження можна об'єднати в одне рівняння (2.13):

$$y_i * (\vec{w} * \vec{x} + b) \geq 1 \text{ для всіх } 1 \leq i \leq n, \quad (2.13)$$

з якого видно наступну проблему оптимізації:

$$\begin{aligned} \text{Мінімізувати } \|\vec{w}\|, \text{ так щоб } y_i * (\vec{w} * \vec{x} + b) \geq 1 \text{ для } i & \quad (2.1) \\ = 1, \dots, n. & \quad 4) \end{aligned}$$

Наведені вище обмеження гарантують, що пряма класифікує кожен елемент вибірки правильно, коли дані лінійно відокремлюються. Але в тих випадках, коли дані не є лінійно відокремленими, потрібно ввести поняття функції втрат. Тобто, більший запас може бути досягнутий, завдяки тому, що класифікатор буде неправильно класифікувати деякі точки даних. Щоб дозволити певний ступінь відмово стійкості, вводяться люзові змінні.

$$y_i * (\vec{w} * \vec{x} + b) \geq 1 - \varepsilon_i, \quad (2.15)$$

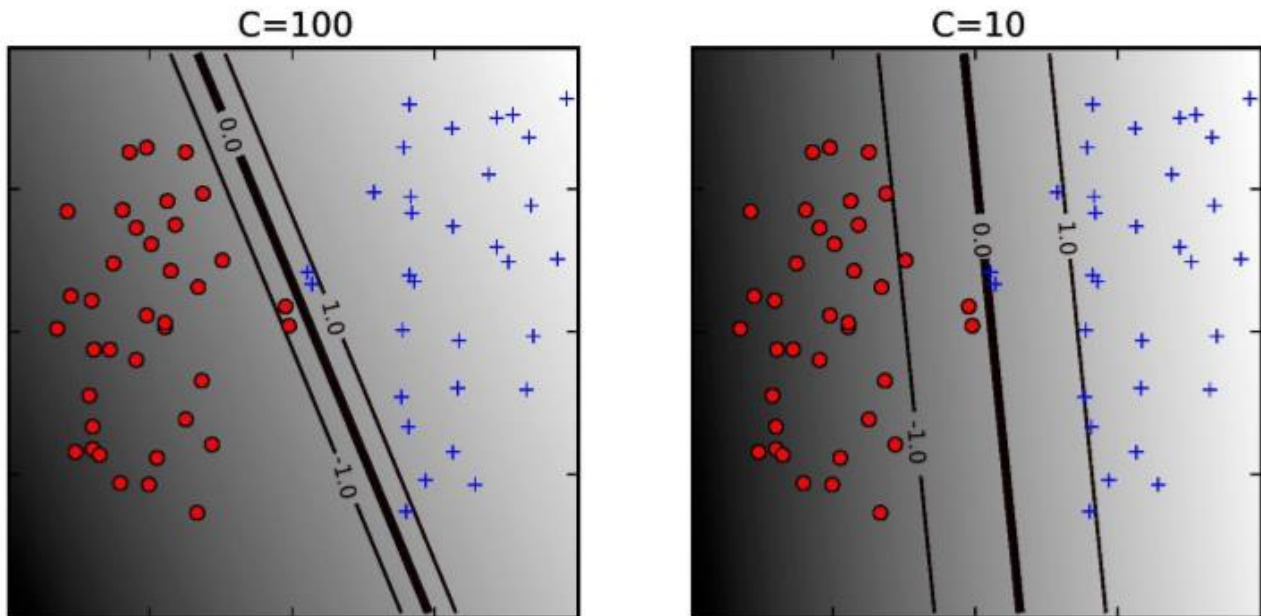
де $\varepsilon_i \geq 0$ для $i = 1, \dots, n$.

Якщо люзові змінні є в межах 0 та 1, це означає, що точки даних знаходяться в запасі (похибка поля), і якщо вони перевищують 1 ($\varepsilon_i > 1$), деякі точки даних неправильно класифікуються. Для того, щоб контролювати помилки класифікації та помилки поля, введемо нову константу C . Таким чином, проблема оптимізації стає:

$$\begin{aligned} \text{Мінімізувати } \frac{1}{2} \|\vec{w}\|^2 + C * \sum_i \varepsilon_i, \text{ щоб } y_i * (\vec{w} * \vec{x} + b) \geq 1 & \quad (2.16) \\ \text{для } i = 1, \dots, n. & \end{aligned}$$

Цю формулу вперше подали Кортес і Вапник. Константа C має важливий вплив на межу прийняття рішення, оскільки її можна використовувати для контролювання запасу та помилок класифікації. Якщо константа C буде мати велике значення, то алгоритм ігноруватиме

деякі помилки класифікації, тобто пряме не досить точно буде відокремлювати негативні і позитивні елементи. Як показано на рис. 2.8 (зліва), дві точки, які є найближчими до прямої, впливають на його орієнтацію. Це призводить до того що пряма поділу знаходиться близько



до інших точок даних. Коли C має менше значення (рис. 2.8 (праворуч)), ці точки знаходяться на однаковій відстані від прямої поділу.

Рисунок 2.8. Прикладу впливу константи C на алгоритм SVM

2.2 Фаза класифікації елементів дорожнього руху

Мета даної фази в інтелектуальній системі – це класифікація елементів дорожнього руху, тобто, встановлення конкретного класу знайденого об'єкту, які були знайдені у фазі розпізнавання. Для розпізнавання використовується алгоритм масштаб незалежного перетворення ознак SIFT (Scale-invariant feature transform), з додатковим етапом, який враховує інформацію про колір об'єкту, що в загальному результаті дає нам досить швидку систему, яку можна використовувати в режимі реального часу. Основними перевагами SIFT алгоритму є:

1. Локальність характеристик: знайдені характеристики об'єкту є частинами об'єкту, тобто локальними, що робить алгоритм стійким до оклюзій чи інших шумів.
2. Відмінність: індивідуальні характеристики об'єкту можуть порівнюватись з великою наявною базою об'єктів.
3. Повнота: навіть для маленьких об'єктів алгоритм може знаходити багато унікальних характеристик.
4. Швидкодія: швидкість алгоритму досить прийнятна для того, щоб використовувати його в режимі реального часу з частотою кадрів > 40 в секунду.
5. Розширюваність: можливість розширювати алгоритм різними видами характеристик, що може зробити систему більш стійку та незалежну до зовнішніх умов.

Цей розділ підсумовує технічні деталі алгоритму. SIFT алгоритм складається з декілька послідовних кроків, а саме:

- вибір пікового масштабу в просторі: знаходження потенційного місця знаходження особливостей об'єкту,
- локалізація ключових точок: встановлення точного розташування ключових точок об'єкту,
- встановлення орієнтації ключовим точкам,
- дескриптор ключових точок,
- узгодження ключових точок.

Об'єкти реального світу є змістовними лише в певному масштабі. На столі ви прекрасно можете побачити кубик цукру, але, якщо дивитися на стіл з точки зору сузір'я молочного шляху, то кубика цукру просто не існує. Ця багато масштабна природа предметів є досить поширеною в природі. Ідея масштабного простору повторюється на цифрових зображеннях.

Простір масштабу зображення – це деяка функція $L(x, y, \sigma)$ (формула 2.17), яка є результатом застосування ядра Гауса (Розмивання) в різних

масштабах за вхідним зображенням. Простір масштабу розділено на октави, а кількість октав і масштаб залежить від розміру вихідного зображення. Таким чином ми генеруємо кілька октав вихідного зображення. Розмір зображення кожної октави наполовину перевищує попереднє. У межах октави зображення розмиваються за допомогою оператора Розмиття Гауса. Математично "розмиття" називають згорткою оператора Гауса та зображення. Розмиття Гауса має певний вираз або "оператор", який застосовується до кожного пікселя. Як результат, розмите зображення.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad (2.17)$$

де, G – оператор розмиття Гауса, а I – вхідне зображення, x та y – координати пікселя та σ – коефіцієнт розмиття.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \quad (2.18)$$

Далі використовуємо ці розмиті зображення для створення іншого набору зображень, Різниця Гаусів (Difference of Gaussians). Різниця Гаусів чудово підходить для пошуку цікавих ключових точок на зображенні. Різниця Гауса виходить як різниця розмиття Гауса зображення з двома різними σ , нехай це буде σ і $k\sigma$. Цей процес робиться для різних октав зображення в Гауссовій піраміді. Діаграма представлена на рис. 2.9.

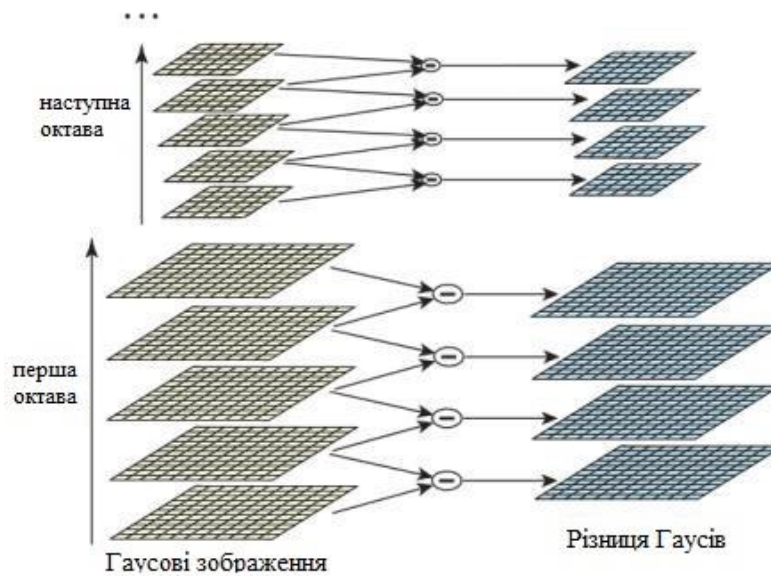


Рисунок 2.9. Розрахунок різниці Гаусів (Difference of Gaussians)

На рис. 2.9 зображено зліва – піраміда Гаусів, а праворуч – їхня різниця. Схематично показано, що кожна різниця розраховується з двох сусідніх Гаусів, кількість різниць на одиницю менше кількості гаусів, та при переході до наступної октави розмір зображень зменшується вдвічі.

Далі кожен піксель на зображенні порівнюється з його 8 сусідами, а також 9 пікселів у наступній шкалі та 9 пікселів у попередніх масштабах. Таким чином робиться загалом 26 перевірок. Якщо точка має локальний екстремум, то це потенційна ключова точка. В основному це означає, що ключова точка найкраще представлена в цій шкалі.

Зазвичай кількість генерованих на попередньому кроці ключових точок досить багато, але не всі вони дійсно є ключовими. Деякі з них лежать уздовж краю об'єкту, або їм не вистачає контрасту (знаходяться в місцях де різко змінюється контраст). В обох випадках вони не такі корисні, як реальні характеристики об'єкту. Таким чином ми повинні їх позбутися. Підхід аналогічний тому, який використовується в детекторі кутів Харріса для усунення крайових особливостей. Для особливостей низької контрастності ми просто перевіряємо їх інтенсивність, і за деяким пороговим значенням (визначеним емпірично) відкидаємо точки, які нас менше всього цікавлять.

В даній роботі використовується розширення масштабного простору серії Тейлора, щоб отримати більш точне розташування екстремумів, і якщо інтенсивність у цій крайності менша за порогове значення 0,03, його відхиляють. Різниця Гаусів має більш високу реакцію для країв, тому край також потрібно видалити.

Для кожної ключової точки встановлюється орієнтація, щоб забезпечити інваріантність до обертання зображення. Величина та напрямок градієнта обчислюється, взявши сусідні пікселі навколо місця розташування ключової точки. Потім створюється гистограма орієнтацій з 36 комірками, що охоплюють пікселі з всіх сторін навколо ключової точки. Гистограма зважується як на величину градієнта, так і на напрям Гауса, зосереджене на ключовій точці. Для розрахунку орієнтації вибираються лише найвищі вершини. Глобальний максимум і будь-який пік зі значенням понад 80% від найвищого піку також приймаються для обчислення орієнтації (див. рис. 2.10).

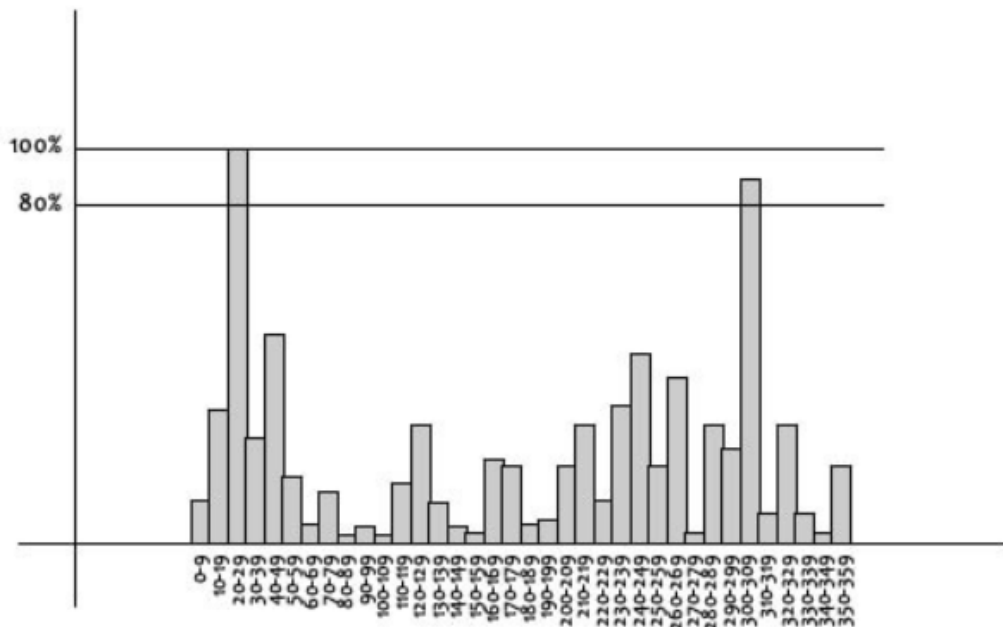


Рисунок 2.10. Розрахунок орієнтації градієнта

Далі слід обчислити дескриптор для локальної області зображення щодо кожної ключової точки, яка є максимально відмітною та інваріантною для варіацій, таких як зміни точки зору та освітленості.

Для цього береться вікно розміром 16x16 навколо ключової точки. Він розділений на 16 під блоків розміром 4x4 (див. рис. 2.11).

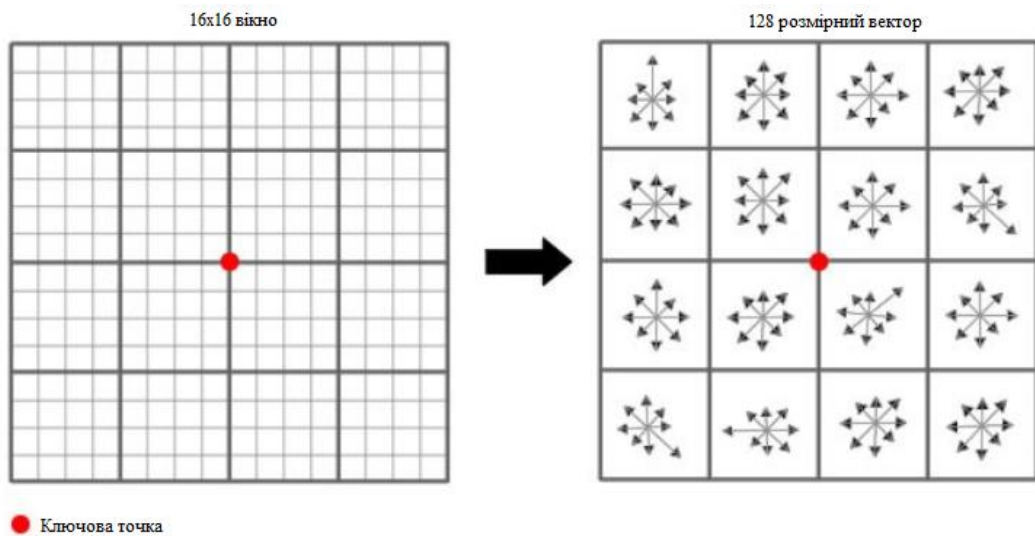


Рисунок 2.11. Обчислення дескриптора ключової точки

Для кожного під блоку створюється 8 бін орієнтації гістограми (див. рис. 2.12).

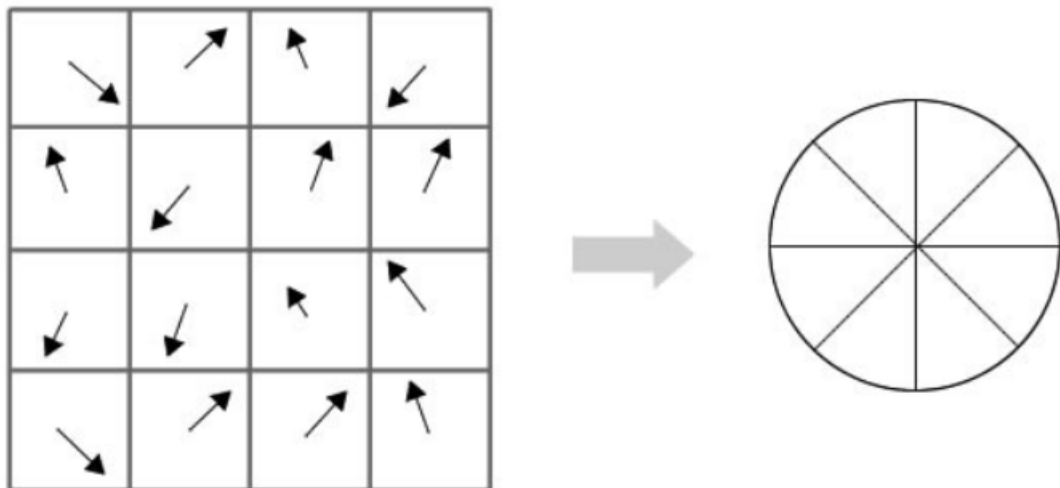


Рисунок 2.12. Представлення комірки ключової точки

В прикладі ми використовуємо $4 * 4$ дескриптори що дає нам матрицю $16 * 16$. Дескриптори $4 * 4$ та 8 напрямків дають нам значення 128 бінів. Він представлений у вигляді функціонального вектору для

формування дескриптора ключових точок. Цей функціональний вектор створює кілька ускладнень:

- Залежність від обертання. Вектор функції використовує градієнтні орієнтації. Ясно, що якщо ви обертаєте зображення, все змінюється. Всі орієнтації градієнта також змінюються. Для досягнення незалежності повороту, обертання ключової точки віднімається від кожної орієнтації. Таким чином, кожна орієнтація градієнта є відносно орієнтації ключової точки.
- Залежність від освітлення. Якщо ми будемо порогувати великі числа, ми можемо досягти незалежності освітлення. Отже, будь-яке число (із 128), яке більше 0,2 змінюється на 0,2. Цей отриманий вектор функції знову нормалізується.

Ключові точки між двома зображеннями узгоджуються шляхом визначення їх найближчих сусідів. Але в деяких випадках друга найближча відповідність може бути дуже близькою до першої. Це може статися через шум чи інші причини. У цьому випадку приймається відношення найближчої відстані до другої найближчої відстані. Якщо вона перевищує 0,8, вони відхиляються. Це виключає близько 90% помилкових збігів.

2.3 Фаза відстеження елементів дорожнього руху

Відстеження об'єктів – це звичайне завдання у багатьох програмах машинного зору на основі відео, наприклад. вуличне спостереження або відстеження військових цілей. Це важливо для підтримання стану об'єкта, змінюючи його 3D-положення, видимий розмір, колір, текстуру або коли він тимчасово невидимий. Слід зазначити, що такі програми, природно, вимагають роботи трекера в режимі реального часу, що дозволяє йому фіксувати ціль, оскільки нові спостереження додаються в режимі реального часу з вхідного відео. Природним питанням є те, чи потрібно підтримувати спеціалізований компонент відстеження в динамічній

системі розпізнавання об'єктів. Іншими словами, чи не може сам детектор досягти тих же цілей, що і окремий детектор і трекер. З кількох причин це не може.

По-перше, можливість досягнення інваріантності загальних змін у позі та зовнішності на рівні представленості функцій обмежена. Тому одна з можливих ролей трекера передбачає ймовірнісне моделювання цих змін. У цьому сценарії параметричний опис цільового стану поширюється з часом, що включає чергування прогнозів параметрів стану та їх оновлення, керованих послідовними кадрами спостережень.

Друга критична роль трекера – це моделювання тимчасових залежностей між цими спостереженнями. Оскільки за відсутності трекера виявлення повинно бути виконано незалежно у кожному кадрі вхідного відео, ці взаємозв'язки безповоротно втрачаються, а значить, оцінка цільового стану з часом менш стабільна і менш точна.

Нарешті, просторове відстеження рухомих об'єктів різко зменшує обчислення, що виконуються, порівняно з окремими кадровими виявленнями. Залежно від проблеми та прийнятої схеми відстеження, це вдосконалення може бути досягнуто двома способами. По-перше, трекер може бути використаний для обмеження меж області пошуку, яка локально сканується детектором у кожному зображенні вхідної послідовності для реєстрації нового положення та масштабу цілі. У другому сценарії використання спеціального детектора для просторової локалізації існуючих об'єктів є непотрібним, оскільки їх фактичне розташування та тривимірне представлення на сцені можна отримати з параметричних оцінок стану трекера. У цьому випадку роль детектора зводиться лише до реєстрації спостережуваних особливостей у фіксованому місці зображення, передбаченому трекером.

Відслідковувати дорожні знаки з автомобільної камери складно з кількох причин. Транспортний засіб, як правило, рухається не з постійною швидкістю, наприклад він може швидко розганятися,

гальмувати або змінювати напрямок руху на перехресті або під час обгону.

Це неправильно вважати, що автомобіль рухається з постійною швидкістю і швидкість. Подальші труднощі у відстеженні дорожніх знаків пов'язані з нестабільністю кріплення мобільної камери. Кожного разу, коли автомобіль рухається по нерівній дорожній поверхні, його коливання поширюються на камеру, що відображається у розмитості, зменшуючи контрастність у знятому відео. Це явище, як правило, важко моделювати. Іншим фактором, який сильно погіршує ефективність роботи дорожніх знаків, є недостатня частота освітленості. Дорожні знаки дуже чіткі, завдяки характерним формам та яскравим кольорам. Однак при несприятливому освітленні обидві властивості неможливо використовувати повною мірою, оскільки контраст навколо межі знаку та набагато менш різкий, а нормально відмінні кольори здаються блідими.

Крім того, тіні та відбивання світла, що часто трапляються під час руху, можуть впливати на спотвореність зображення. У багатьох реалістичних ситуаціях, особливо на міських сценах, дорожні знаки повинні бути виявлені та розпізнані за наявності зазначених труднощів. Хоча сам зовнішній вигляд знаку не сильно змінюється під час наближення автомобіля, але сусідні об'єкти на зображенні можуть швидко змінитися. Це збільшує ризик плутанини між справжніми знаками та іншими предметами чи фоном, а отже, ускладнює тимчасове відстеження знаків, особливо їх контурів.

Нарешті, слід визначити обчислювальний аспект відстеження, оскільки це найбільш інтенсивний крок у системі, припускаючи, що повне дослідження сцени в пошуках нових кандидатів областей, де ймовірно є знак, не потрібно проводити в кожному кадрі вхідного відео. Маючи на сцені потенційно до декількох різних ознак, загальний час, необхідний для оцінки стану кожного кандидата у кожному кадрі, не повинен перевищувати декількох мілісекунд. Тому трекер, замість того,

щоб обмежуватися лише скороченням локальної області пошуку для детектора, також повинен працювати над представленням цілі з низькою розмірністю, щоб оновлення стану треку на основі спостереження поточного кадру могло робити досить швидко.

У запропонованій системі представлений базовий метод відстеження дорожніх знаків. Він заснований на техніці фільтрації Кальмана-Бюсі. Запропонований трекер робить припущення прямолінійного руху постійного простору автомобіля. Він намагається моделювати лише геометрію цілі з часом і складається з двох частин: фільтра Кальмана для прогнозування положення та масштабу, а також оцінки локальної області пошуку та фільтра частинок, який використовується для виправлення анізотропного масштабування виявленої ознаки.

Комбінований трекер елементів дорожнього руху був реалізований та інтегрований із штатними багатокутними детекторами. Щоб спостерігати за ефективністю цього трекера в реалістичних сценаріях руху, ми протестували його на майже 200 коротких відео, записаних за допомогою передньої відеокамери DV, встановленої прямо перед лобовим склом автомобіля. У цьому експерименті встановлено звичайний багатокутник-детектор для зйомки початкових знаків радіусу між 15 і 25 пікселями. Трекер повинен був взяти на себе процес виявлення після встановлення кожного з цих початкових кандидатів. Відстеження здійснювалося до тих пір, поки ціль не досягла межі видимого фрагмента сцени. На рис. 2.13 показано кілька послідовностей зображень, що ілюструють процес відстеження. У більшості тестових послідовностей точність трекера була високою. У випадках, коли відбувається ненульове обертання поза площиною цілі, один звичайний детектор багатокутника не в змозі пристосувати точний контур до спостережуваного дорожнього знаку. Ця проблема викликана тим, що форма спотвореного знаку вже не радіально-симетрична. В результаті детектор може видавати лише частково правильні пристосування. Наприклад, лише 3 з 4 сторін

видимого прямокутника можуть добре відповідати квадратному детектору. Крім того, це не сильно погіршує обчислювальну ефективність трекера. Це тому, що для пошуку значень лише двох параметрів масштабу в дуже малому діапазоні значень потрібно відносно мало частинок.

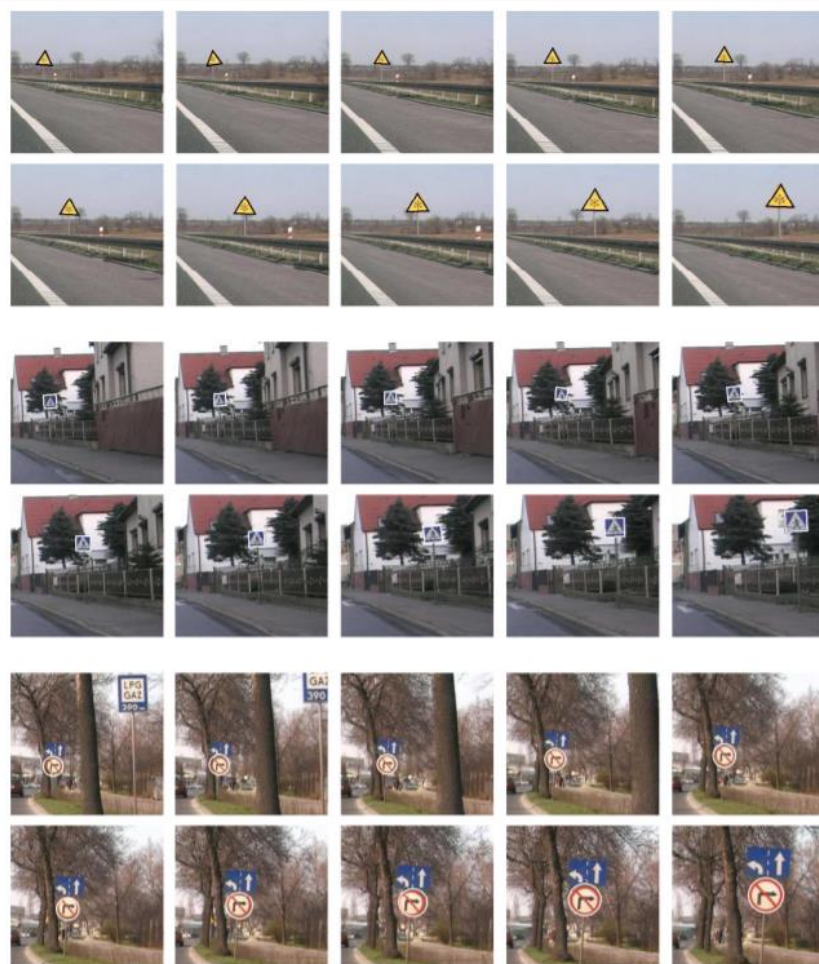


Рисунок 2.13. Процес відстеження елементів дорожнього руху

Представлений фільтр Кальмана, разом із дискримінаційними детекторами об'єктів, можна використовувати для ефективного відстеження дорожніх знаків у часі у вхідному відео. Однак жоден з цих детекторів не може отримати повну 3D-геометрію цілі. Каскад підсилених класифікаторів лише надає інформацію про положення центроїда об'єкта та його масштаб. Звичайний багатокутний детектор може додатково отримати внутрішнє обертання дорожніх знаків, що мають некруглу форму. Для кругових знаків це неможливо. В обох

випадках недостатньо знань для визначення очевидного обертання дорожніх знаків навколо осей перпендикулярно до оптичної осі камери, що може бути наслідком зміни точки зору камери відносно цілі. Підхід на основі фільтрування частинок, розглянутий у цьому розділі, вирішує цю проблему тим, що дозволяє виправити анізотропне масштабування, яке є найпоширенішим типом спотворень дорожніх знаків спотворення. Хоча оцінка повного набору параметрів, можлива за допомогою цієї методики, ми обмежимося корекцією масштабу, яку можна обчислити. Запропонований фільтр ініціалізується в момент виявлення нового ймовірного дорожнього знаку на місці події. У цей момент детектор надає інформацію про його центроїд і масштаб, будь то каскад уважного класифікатора або детектор форми на основі Хафа. Стан у будь-який час t задається парою змінних. Вони кодують горизонтальний та вертикальний коефіцієнти масштабування афінної трансформації, яка має бути застосована до симетричної форми, виявленої детектором навколо попереднього центроїда та шкали часу t .

Висновки по розділу

В даному розділі розглянуто запропоновану інтелектуальну систему розпізнавання елементів дорожнього руху, наведено її основні фази, детально описано кожен крок. Розкрито доцільність використання алгоритмів на всіх етапах розпізнавання та наведено характеристики інтелектуальної системи, з чого можна зробити висновки про основні переваги та недоліки запропонованої системи.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У цьому розділі представлені передові методи навчання та інші прийоми, пов'язані з нашим детектором елементів дорожнього руху.

3.1 Реалізація машинного навчання системи

Будь-яка система виявлення потребує набору позитивних та негативних зразків, які будуть використані у навчальному процесі. Ми вибрали 1000 зображень як позитивні навчальні зразки. Крім того, ми збільшили кількість позитивних зображень, додавши розроблені, обернені та перекладені версії оригінальних зразків, що призводить до кращої ефективності виявлення.

На рис. 3.1 показана візуалізація зображень повного переліку об'єктів навчальних прикладів та їх середнього значення.



Рисунок 3.1. Візуалізація прикладів дорожніх знаків

Початкову кількість негативних зразків (2000) вибирають із навчальних зображень із задалегідь викресленими регіонами. Для того, щоб підвищити ефективність навчання класифікатора, ми використовуємо передовий і популярний метод навчання, який

називається методом опорних векторів (SVM). Поки що SVM було вивчено за допомогою невеликої вибірки негативних зображень. Тим не менше, по суті, кожна окрема область зображення, яка не містить елементу дорожнього руху, може розглядатися як негативний приклад. Існує дуже багато зразків, які можна використовувати на практиці, але ми шукаємо лише ключові негативні зразки. Ми тренуємо SVM в ітераційним способом. У кожній ітерації детектор застосовується до нового зображення без дорожніх знаків. Тоді ми додаємо неправильно розпізнанні картини, де дійсно був дорожній знак до вибірки на наступну ітерацію. Цей процес повторюється протягом 5 ітерацій. В результаті, класифікатору надається більше дійсно негативних прикладів, що допомагає зробити систему більш стійкою рис. 3.2.



Рисунок 3.2. Позитивні зразки (вгорі), середнє зображення (внизу)

Після того, як були проведені операції з добування негативних прикладів та навчання, ми оцінюємо модель на тестових даних. Для усунення фальшивих розпізнавань використовується алгоритм не максимального придушення (Non-Maximum Suppression - NMS). NMS зберігає виявлення найвищої шкали та видаляє будь-яке інше виявлення,

перекриття якого перевищує поріг. Ми використовували показник перекриття Паскаля, щоб встановити коефіцієнт перекриття між двома обмежувальними полями. Він обчислюється як:

$$\alpha_0 = \frac{area(B_1 \cap B_2)}{area(B_1 \cup B_2)}, \quad (3.1)$$

де α_0 – коефіцієнт перекриття. B_1 і B_2 – це два перекриваючі об'єкти.

Спершу, кандидати, знайдені у фазі виявлення, масштабуються до розміру знаків шаблону. Попередньо був зібраний повний набір шаблонів знаків і об'єктів, які потрібно використовувати на етапі розпізнавання.

Наступним кроком є розробка способу інтеграції кольорової інформації з метою підвищення продуктивності для SIFT алгоритму. Ми підраховуємо різницю показників кольору вхідного зображення та шаблонних. Емпіричним шляхом було встановлено, що найбільш ефективним і кращим простором кольору є HSV. Більше того, простір кольору HSV має більш послідовну різницю кольорів між вхідним кандидатом та знаками шаблону. На рис. 3.3 відображаються пораховані значення відтінку (Hue), насичення (Saturation) та значення кольору для шаблону та визначених знаків.

Елементи дорожнього руху включають в себе різноманітні кольори і при представленні кольору в просторі HSV більш ефективно працює. Після отримання характеристик кольору, ми розраховуємо середнє значення всіх значень на основі визначеної маски. Потім створюємо дельта зображення:

$$\begin{aligned} \delta H &= H_{channel} - H_{standard}, \\ \delta S &= S_{channel} - S_{standard}, \\ \delta V &= V_{channel} - V_{standard}, \end{aligned} \quad (3.2)$$

де $H_{channel}$, $S_{channel}$, $V_{channel}$ – середні значення характеристик кольору зображення кандидата. При цьому остаточно середнє значення розраховується за формулою 3.3:

$$\delta f = \sqrt{\delta H^2 + \delta S^2 + \delta V^2}, \quad (3.3)$$

де значення δf знаходиться в межах між 0 та 1. Чим ближче значення до 1, тим більша різниця характеристик кольору між кандидатом і уже відомими об'єктами.

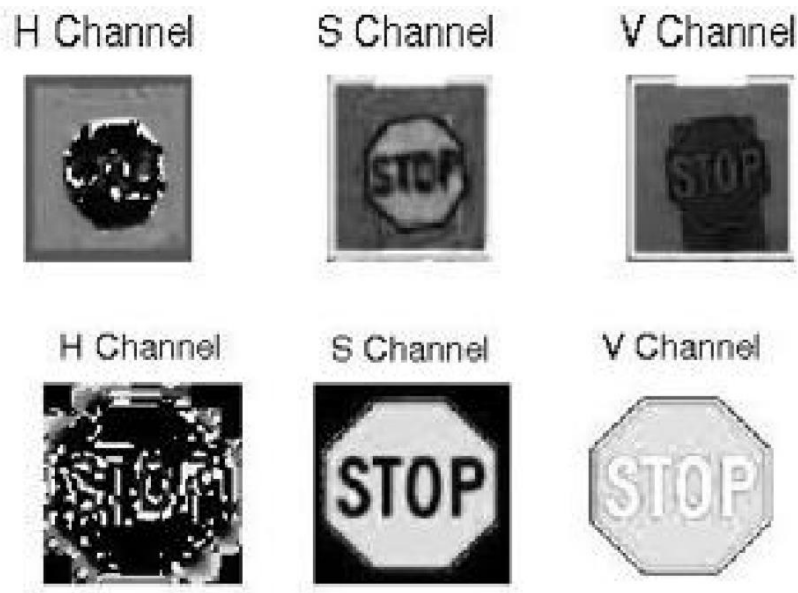


Рисунок 3.3. HSV пороховані зображення, зверху – кандидат, знизу – уже відомий знак

Порогове значення чи є ці два зображення кольоровими подібними чи ні, було визначено експериментальним шляхом, для забезпечення максимальної точності та швидкодію. Якщо δf менше визначеного порогового значення, ми вибираємо відповідний шаблон з бази для подальшої роботи SIFT алгоритму.

Отже, ми виконуємо SIFT алгоритм лише між зображенням кандидата та тими зображеннями в базі даних шаблонів, кольори яких схожі на виявлений знак. Цей метод видаляє деякі можливі помилкові

збіги та суттєво покращує продуктивність. Після того, як будуть знайдені всі шаблонні зображення, які мають подібний колір до виявленої цілі, ми виконуємо відповідність між тими зображеннями та зображеннями кандидата. Алгоритм SIFT використовується для цих цілей. Рис. 3.4 ілюструє різницю Гаусів для різних октав та масштабів зображення. Після встановлення ключових точок ми обчислюємо дескриптори кожної ключової точки. Потім ми приступаємо до уточнення дескрипторів кандидатів. Для цього ми визначаємо поріг: дескриптор d_1 узгоджується з дескриптором d_2 , якщо відстань між ними, помножене на поріг, не більше відстані d_1 до всіх інших дескрипторів. Значення цього порогу було отримано експериментальним шляхом і воно дорівнює 1,5.

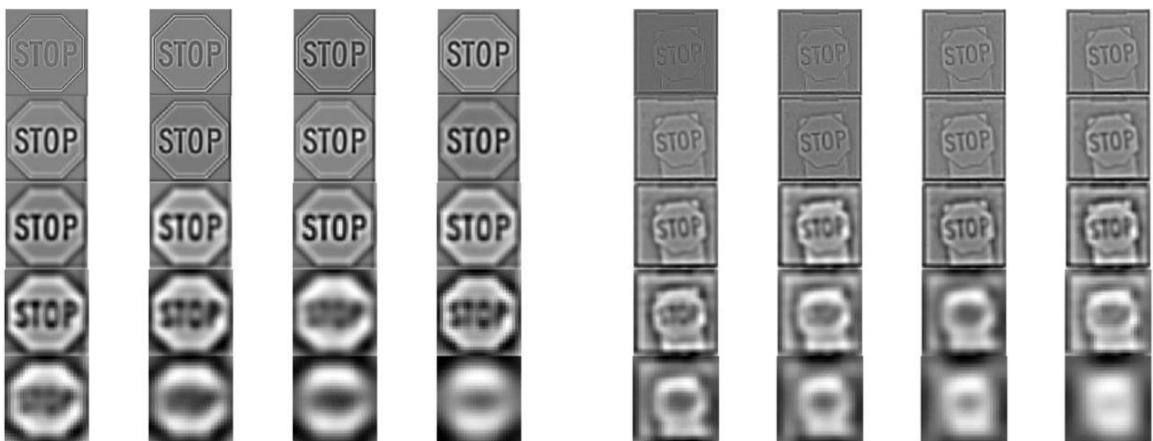


Рисунок 3.4. Різниця гаусів для різних октав і масштабів, зліва – шаблон, справа – кандидат

Алгоритм вибіркового прикладу RANdom SAmple (RANSAC) був використаний, для встановлення ключових точок об'єкту. Це ітеративний метод оцінки математичної моделі з набору даних, що містить зайві параметри. Основні етапи цього алгоритму узагальнені наступним чином:

- випадково вибирається мінімальну кількість точок, необхідну для визначення параметрів моделі,
- параметри моделі розв'язуються,
- визначається, скільки точок із набору вибраних точок є параметрами моделі із заданим допуском ϵ ,

- переоцінка параметрів моделі, якщо частка аномальних точок від загальної кількості точок у наборі нижча за встановлений поріг τ ,
- повторення цих кроків, поки не буде досягнуто належного рівня впливу для оцінюваних параметрів моделі.

Для того, щоб виявити аномальні приклади, що вижили, під час тренінгу був використаний невеликий набір зразків для тренування гомографічної моделі. Потім визначаються зразки, які мають толерантність до похибки моделі гомографії. Ці зразки розглядаються як нормальні. Якщо кількість аномальних зразків є найбільшим знайденим досі, ми зберігаємо поточний набір даних. Цей процес повторюється протягом ряду ітерацій і повертає модель з найменшою середньою помилкою серед створених моделей.

3.2 Архітектура програмного забезпечення

Для розробки програмного забезпечення було використано бібліотеку Accord.NET Framework, з загальною реалізацією підходів до машинного навчання. Для розробки інтерфейсу користувача було використано Windows Forms технологію. Accord.NET Framework – це система машинного навчання .NET, поєднана з бібліотеками обробки аудіо та зображень, повністю написаними на C #.

Архітектура системи представлена на рис.3.5.

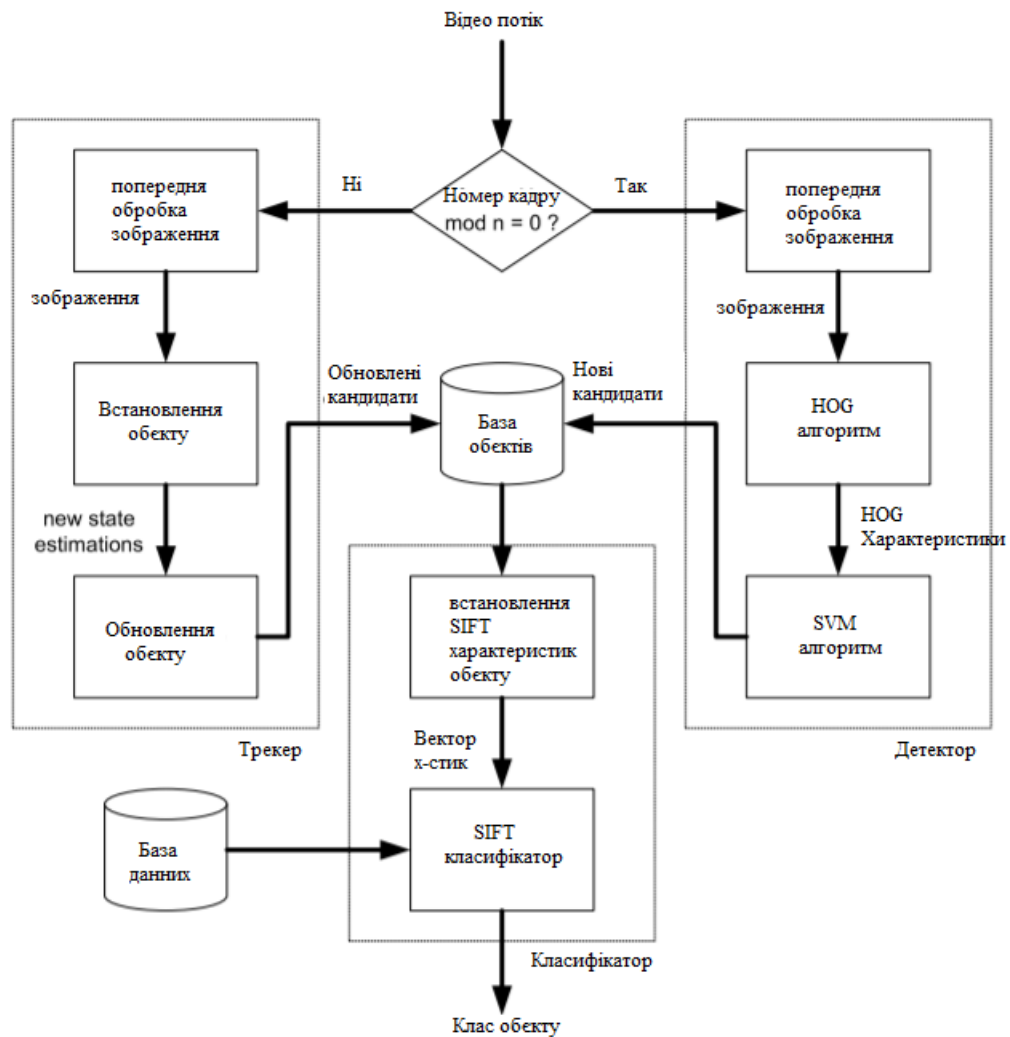


Рисунок 3.5. Архітектура системи

3.4 Вимоги до технічних характеристик комп'ютерного забезпечення

- процесор – 1.5 ГГц, 1 ядро ЦП або краще;
- оперативна пам'ять не менш ніж 512 МБ;
- не менше ніж 32 ГБ ПЗУ.

Загалом технічні характеристики впливають на швидкодію системи, і дуже сильно залежать від FPS параметра відео потоку, тобто з якою частотою буде виконуватись алгоритм розпізнавання. Для пришвидшення роботи можна виконувати алгоритм лиш на кожному N кроці, проте є ризик втратити характеристики точності.

Проте вище викладені мінімальні характеристики, які задовільняють нормальну роботу системи, забезпечуючи високу точність розпізнавання елементів дорожнього руху за прийнятний час.

Збільшуючи технічні характеристики комп'ютерного забезпечення можна добитись кращих результатів точності, завдяки можливості запускати алгоритм частіше.

3.5 Вимоги до програмного забезпечення

Загальні вимоги до програмної системи:

- .NET Framework 4.5 або більше;
- операційна система: Windows 7/8/10.

Загалом система може запускатись на будь-якій операційній системі, але для цього потрібно додатково реалізовувати плагіни окремо для кожної операційної системи.

Одною з ідеєю було встановлення фреймворку .NET Core, який сам по собі вже є крос платформним, що спростило б реалізацію.

Для подальшої розробки системи потрібно використовувати мову програмування C# та будь-яке середовище розробки.

3.6 Результати експериментів

Інтелектуальна система була протестована на 3500 кадрах. Серед цих кадрів з'явилося 1806 знаків руху. Для обрахунку результатів експерименту було представлено два значення в табл. 3.1 та табл. 3.2, відсоток точності розпізнавання (DR) та FPPF (false positive per frame) – це кількість неправильно визначених знаків по відношенню до загальної кількості знаків, які визначенні наступною формулою:

$$DR = \frac{TP}{TP + FN}, \quad (3.4)$$

$$FPPF = \frac{FP}{F}, \quad (3.5)$$

де, TP – кількість правильно розпізнаних об'єктів (об'єкт був на картинці та був правильно розпізнаний), FN – кількість не розпізнаних об'єктів на картинці де не було об'єкту, FP – кількість нерозпізнаних об'єктів, коли об'єкт дійсно був присутній на картинці та F – кількість вхідних картинок.

Таблиця 3.1. Результати точності розпізнавання та характеристик системи

	DR	FPPF
Запропонована інтелектуальна система	0.96	0.01

Таблиця 3.2. Сумарні результати розпізнавання

Кількість знаків	1806
Кількість правильно розпізнаних знаків	1774
Кількість неправильно розпізнаних знаків	32

Фаза розпізнавання будується на фазі виявлення з метою підтвердження виявлених кандидатів та визначення точного типу елемента. Табл. 3.3 демонструє точність запропонованого нами способу розпізнавання знаків руху. Результати розпізнавання знаків також відображаються на рис 3.6.

Таблиця 3.3. Результати розпізнавання знаків дорожнього руху

Кількість встановлених знаків	1517
Кількість розпізнаних знаків	1475
Кількість неправильно розпізнаних знаків	42
Точність розпізнавання	97.2%

Інший стандартний метод оцінки розпізнавання знаків – це матриця плутанини. У полі машинного навчання та комп'ютерного зору матриця плутанини – це таблиця, яка зазвичай використовується для оцінки ефективності класифікаційної моделі на наборі даних тесту, коли відомі справжні значення. Кожен стовпець цієї таблиці представляє передбачувані значення, тоді як кожен рядок представляє справжні значення.

На рис. 3.6 наведено порохований приклад матриці плутанини для двокласного класифікатора.

Матриця плутанини системи показує які класи об'єктів система розпізнає неправильно та як саме помиляється система. З матриці плутанини можна зробити багато цікавих висновків, які можуть покращити точність системи.

Матрицю плутанини потрібно читати згідно з табл. 3.4, де:

- a – це кількість правильних прогнозів про те, що екземпляр є позитивним;
- b – це кількість неправильних прогнозів, що екземпляр є негативним;
- c – це кількість неправильних прогнозів, що екземпляр є позитивним;
- d – це кількість правильних прогнозів, що екземпляр негативний.

Таблиця 3.4. Приклад таблиці плутанини системи

	Результат розпізнавання	
	Позитивний	Негативний
Дійсно позитивний	a	b
Дійсно негативний	c	b

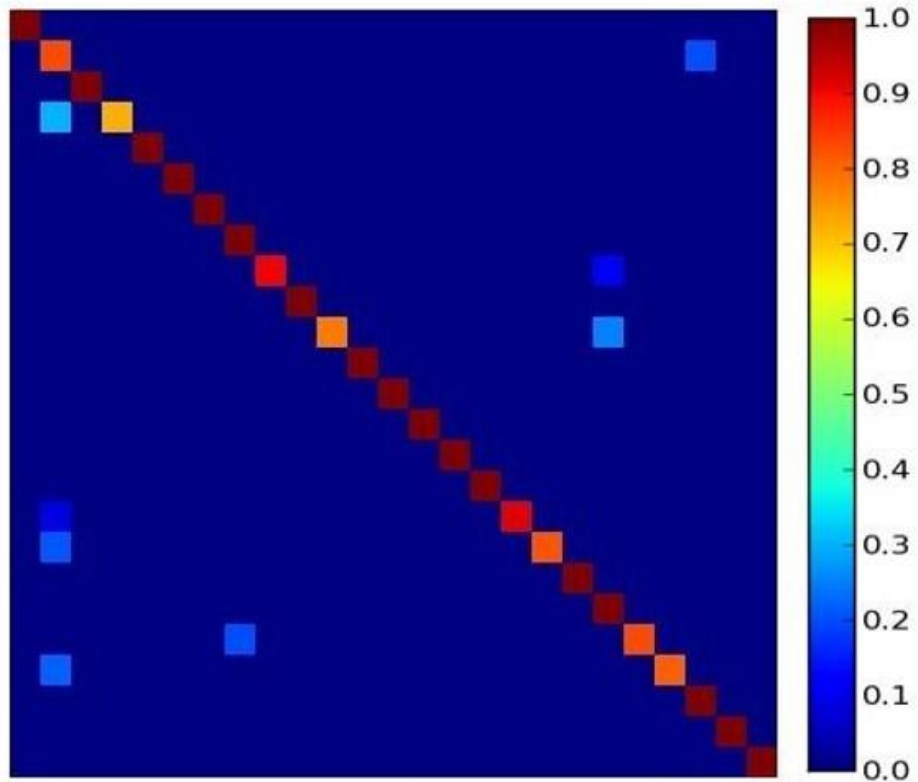


Рисунок 3.6. Матриця плутанини системи

Висновки по розділу

В третьому розділі описано етап розробки алгоритмічного та програмного забезпечення інтелектуальної системи розпізнавання елементів дорожнього руху. Наведено архітектуру програмного продукту та технічні вимоги, описано технічну реалізацію та підходи реалізації системи, також наведено результати експериментів.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Інтелектуальна система розпізнавання елементів дорожнього руху в режимі реального часу	Системи пасивної безпеки в автомобільному транспорті	Допомога водіям в отриманні інформації про дорожні знаки. Пасивна безпека. Розширення можливостей уже існуючих систем
	Вдосконалення існуючих відео реєстраторів	
	Використання в безпілотних автомобілях і вантажівок	

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Мій проект	Taha Khan	Boris Karamatic	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
1	Надійність (точність розпізнавання)	96%	~81%	97%	Неефективність використання при поганій точності розпізнавання	Джерело додаткової інформації водію	Ефективна допомога водіям в результаті зменшення аварій

Закінчення таблиці 4.2

2	Можливість запровадити високотехнолог	Є	Є	Є	Відсутність товарів-замінників	Можливість розширити	Можливість запровадити високотехнолог
---	---------------------------------------	---	---	---	--------------------------------	----------------------	---------------------------------------

	існує систему розпізнавання елементів дорожнього руху					функціональність і характеристики уже існуючих систем та побудова автоматизованих рішень	існує систему розпізнавання елементів дорожнього руху
3	Інтеграція в інші системи	Присутня	Відсутня	Відсутня	Складність інтеграції	Підтримка системи	Можливість розширення функціональності існуючих систем

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Інтелектуальна програмна система розпізнавання елементів дорожнього руху	Технології штучного інтелекту та глибокого машинного навчання	Технологія частково наявна	Технологічна реалізація проекту доступна
2	Інтелектуальна програмна система розпізнавання елементів дорожнього руху	OpenCV бібліотека з патентованими реалізаціями алгоритмів	Технологія частково наявна	Технологічна реалізація проекту частково доступна

Закінчення таблиці 4.3

3	Можливість інтеграції системи	Технології мови програмування C# та Accord.Net	Технологія наявна	Технологічна реалізація проекту доступна
<i>Обрана технологія реалізації ідеї проекту: №1</i>				

Висновок: технологічна реалізація продукту – є можливою, вибрана технологія №1.

4.3 Аналіз ринкових можливостей запуску стартап-проєкту

Таблиця 4.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Високо-технологічні компанії, держави
2	Загальний обсяг продаж, грн./ум.од	У залежності від початкового капіталу
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу	Складність технології
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі або по ринку, %	45%

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проєкту

Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару

	Потреба автоматизованих систем, для автоматизації та роботизації бізнесу/складів, логістики. Інноваційність	Власники високотехнологічних компаній, держави, компанії сфери послуг	Відмінності відсутні	- До продукції: висока точність розпізнавання елементів, надійність роботи системи, коректна поведінка в разі непередбачуваних подій. - До компанії: висока якість товару, підтримка товару
	Потреба у відносно дешевих інтелектуальних систем пасивної допомоги водіям	Держави, компанії сфери послуг	Відмінності відсутні	- До продукції: дешева вартість інтеграції та реалізації. - До компанії: висока якість товару, підтримка товару

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Відсутність інвестицій	Недостатня розвиненість інвестиційної діяльності у країні	Пошук грошових ресурсів на краудфандингових платформах
2	Складність реалізації	Відсутність досвіду у розробці аналогічних систем	Залучення експертів суміжних областей розробки
3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Зацікавленість	Ріст популярності	Інвестиції в розробку технологій

	інвесторів	систем штучного інтелекту та машинного навчання	
2	Відсутність аналогів	Наявність товарів-замінників, але не аналогів	Активна реклама продукції, захоплення якомога більшої кількості ніш

Таблиця 4.9. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: олігополія	Є компанії з продуктом-замінником	Пропозиція ідейно нового товару
2	За рівнем конкурентної боротьби: національний	Наявність груп клієнтів з однаковими потребами на всій території ринку	Робота на якомога ширшій території
3	За галузевою ознакою: внутрішньогалузева	Наявність галузі застосування товару	Запровадження нової технології
4	Конкуренція за видами товарів: товарно-видова	Конкуренція товарів-аналогів	Запровадження нової технології
5	За характером конкурентних переваг: нецінова	Конкуренція за якісними характеристиками	Запровадження нової технології
6	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку.

Таблиця 4.10. Аналіз конкуренції в галузі за М. Портером

ладові аналіз	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
---------------	---------------------------	-----------------------	---------------	---------	------------------

	Відсутні	Поява конкурентів з товаром - аналогом	Відсутні	Відсутність можливості самим впровадити систему	Є компанії, що надають товари - замітники
Висновки	Інтенсивність припустима через відсутність товарів - аналогів	Бар'єри прийнятні	Постачальники не диктують умови роботи	Клієнти не диктують умови	Необхідність реклами переваг нової ідеї

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має ряд недоліків та переваг, та в свою чергу, створювався не заради комерційної вигоди, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал що має кращі характеристики ніж у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.11. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Товар - замітник	Повна відсутність товарів - аналогів
2	Нова технологія	Сучасний підхід до організації логістики та сервісу
		Можливість побудови унікальних автоматизованих систем

Закінчення таблиці 4.11

3	Не повторюваність	Патентування розробленої системи та алгоритмів
4	Широкий спектр застосування	Можливість інтеграції в інші системи

Таблиця 4.12. Порівняльний аналіз сильних та слабких сторін інтелектуальної системи розпізнавання елементів дорожнього руху

J	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим							
			3 ⁻	2 ⁻	1 ⁻	0	1 ⁺	2 ⁺	3 ⁺	
1	Товар - замітник	1-20							X	
2	Можливість інтеграції	1-20	X							
3	Не повторюваність	1-20					X			
4	Широкий спектр застосування	1-20						X		

Таблиця 4.13. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> – Рекламні акції, ідея впровадження нових технологій. – Застосування новітніх технологій при виробництві. – Ефективна кадрова політика, професійність та кваліфікованість кадрів. – Основна сума статутного капіталу формується за рахунок власних фінансових ресурсів учасників, вкладень інвесторів, прибутку продажі 	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> – Недостатня усвідомленість про технологію на ринку. – Необхідність кваліфікованого персоналу для розробки та підтримки. – Виникнення додаткових витрат на навчання та перекваліфікацію персоналу. – Можливі додаткові витрати на навчання персоналу при розробках. – Можливі додаткові фінансові витрати при розробці та покращенні технології
<p>Можливості (O):</p> <ul style="list-style-type: none"> – Зростання попиту на продукцію. – Вихід на нові ринки або сегменти. – Послаблення позицій конкурентів. – Використання новітніх світових технологій. – Залучення висококваліфікованого персоналу. – Послуги даного типу надзвичайно популярні у наш час 	<p>Загрози (T):</p> <ul style="list-style-type: none"> – Поява товарів – аналогів – Нестабільна інвестиційна ситуація. – Неприйняття новизни проекту

Таблиця 4.14. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс - наявний	3-4 місяці
2	Реклама	Залучення власних коштів для реклами товару	1-3 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс - наявний	3-4 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	2-3 місяці

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.15. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Вітчизняні виробники засобів пересування	Готовий	Середній	Маленька	Середня
2.	Виробники відео реєстраторів	Готовий	Великий	Маленька	Складна
3.	Державні правоохоронні органи	Маленька	Маленький	Маленька	Дуже складна
Обрано цільові групи №1 та №2.					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного

продукту є вітчизняні виробники засобів пересування, та іноземні виробники загалом. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для підприємств та компаній у цілому надається стандартизований продукт з можливістю його інтеграції в інші системи.

Таблиця 4.16. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання функціональності що відсутня у товарів-замінників, підтримка клієнтів	Проведення реклами, освітлення унікальної функціональності через інтернет ресурси та інші канали, контакт на пряму з споживачами; формування лояльності і прихильності споживачів	Зниження ступеню заміненості товару; Прихильність клієнтів; Відмітні властивості товару; Відмітні характеристики товару;	Стратегія диференціації

Таблиця 4.17. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є товари-замінники, але дані товари замінники не мають деякого необхідного функціоналу	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх	Ні, оскільки система в цілому є закритою, і загалом неможливо вирахувати які алгоритми були використані	Стратегія виклику лідера

Таблиця 4.18. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Висока точність розпізнавання	Стратегія спеціалізації	Висока точність розпізнавання	Точно розпізнаваність
2	Швидкодія	Стратегія диференціації	Можливість налаштувати систему згідно з потрібними вимогами	Налагоджуваність
3	Інтуїтивно зрозумілий інтерфейс	Стратегія диференціації	Зрозумілий інтерфейс	Інтуїтивність

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.19. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Системи пасивної безпеки автомобіля	Дешева та точна система розпізнавання елементів дорожнього руху	Точність та швидкість розпізнавання
2	Інтегрованість	Можливість інтеграції в різні системи	Гнучкість використання
3	Інтуїтивно зрозумілий інтерфейс	Можливість налаштувати систему	Простота

Таблиця 4.20. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Інтелектуальна система розпізнавання елементів дорожнього руху		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Точність розпізнавання		
	Швидкодія		
	Налагоджуваність		
	Зрозумілий інтерфейс		
3. Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент			

Таблиця 4.21. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
~1,000,000,000\$	~7,000,000,000\$	~235,800,000,000\$	\$5,000,000 - \$50,000,000

Таблиця 4.22. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Вимоги до презентації системи	Презентувати систему, відповідати на запитання	Залежить від розмірів компанії замовника	Оптова

Таблиця 4.23. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Консервативність	Електронна пошта та зустрічі.	Технологічність системи	Донести інформацію про використані технології	Презентації та демо-показ системи.
2	Неприйняття новий ідей		Результати експериментів	Розкрити результати експериментів та їхнє значення	

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки по розділу

В розділі описано підходи та стратегії з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що є можливість ринкової комерціалізації проекту. Розглянувши потенційні категорії клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проекту було встановлено що проект є дійсно перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої реалізації проекту.

ВИСНОВКИ

У ході виконання магістерської дисертації було розглянуто питання розпізнавання елементів дорожнього руху, їхня важливість та роль для безпеки дорожнього руху. Був проведений аналіз існуючих алгоритмів, методів, підходів до розпізнавання елементів на картинці. Виокремлено основні проблеми та складнощі цієї галузі, на основі чого було запропоновано інтелектуальну систему розпізнавання елементів дорожнього руху.

На основі даних, отриманих в процесі аналізу, сформульовано задачу створення інтелектуальної системи розпізнавання елементів дорожнього руху. Описано вимоги до програмної реалізації системи та її основних практичних характеристик.

Робота та архітектура система поділена на окремі фази, на кожній з яких, детально обґрунтовано доцільність використання вибраних алгоритмів. Також було описано роботу системи, припущення та вибрані коефіцієнти емпіричним шляхом.

Для програмної реалізації інтелектуальної системи було використано мову програмування C#, середовище розробки Visual Studio 2017, технологію логування подій Serilog, фреймворк Accord.NET з програмними реалізаціями вже відомих алгоритмів та WindowsForms для реалізації інтерфейсу користувача. Для навчання й тестування системи використовувалась C# Console application. Все перераховане є широко вживаними та безкоштовними інструментами, що дозволить легко розширити функціонал системи в разі виникнення відповідної потреби.

Проведено експериментальні розрахунки, в результаті яких встановлено, що система має точність розпізнавання 96% на тестовому наборі даних. Завдяки високій швидкодії системи забезпечується її робота в режимі реального часу. Збільшувати точність розпізнавання системи можна за рахунок більшого вхідного набору даних, який використовувався для навчання системи.

Проведений маркетинговий аналіз стартап-проекту. Проведено опис ідеї проекту, технологічний аудит ідеї проекту, аналіз ринкових можливостей запуску стартап-проекту. Розроблено маркетингову програму та ринкову стратегію проекту та стартап-проекту. Дана інтелектуальна система може використовуватись у сферах пов'язаних з автомобільним транспортом.

Результати роботи над магістерською дисертацією опублікований у журналі Таврійських вісник.

Наукова новизна одержаних результатів магістерської дисертації полягає у наступному:

вперше:

- розроблено програмну реалізацію системи з можливістю її використання в інших програмах;

удосконалено:

- точність розпізнавання елементів в порівнянні з іншими системами;
- здешевлення пасивних систем допомоги водіям;

ПЕРЕЛІК ПОСИЛАНЬ

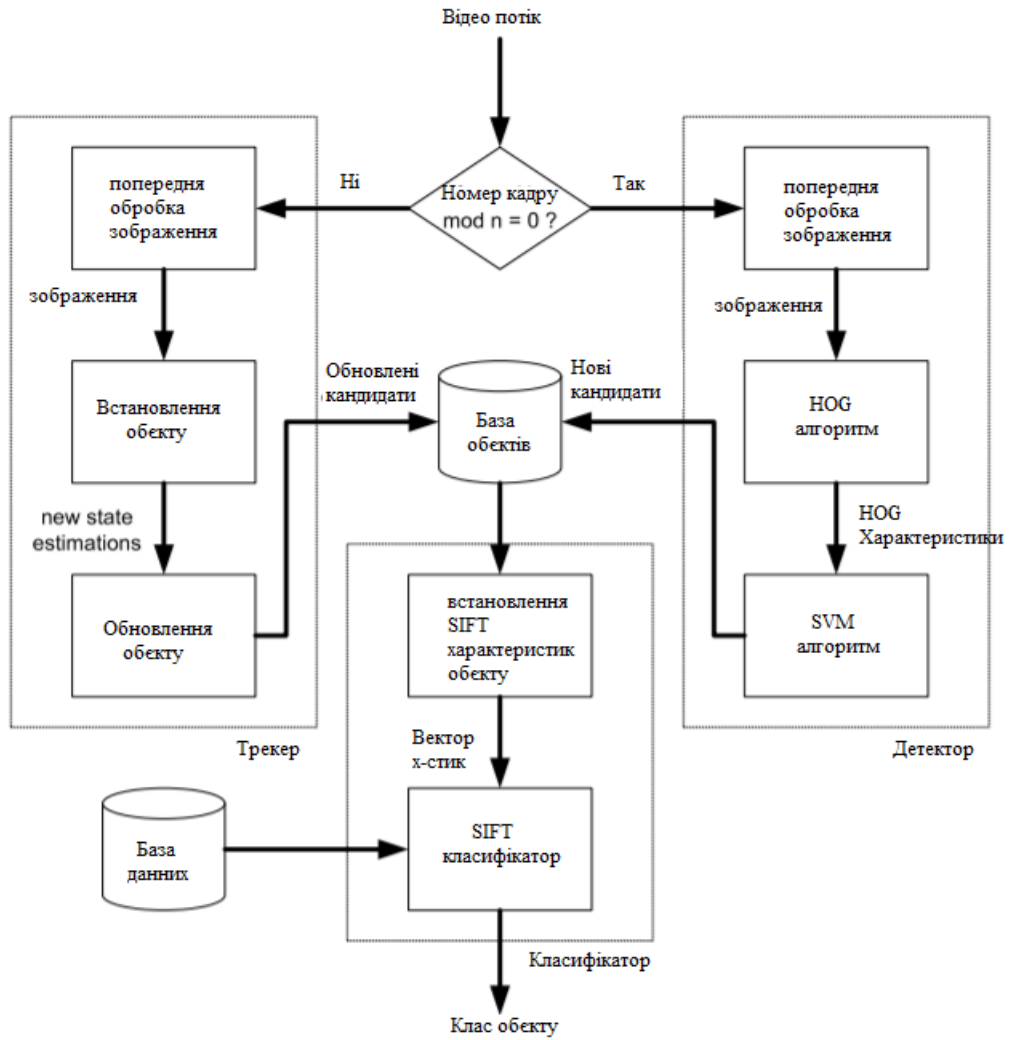
1. S. Singh, Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Chicago, 2015. 373 с.
2. Liu, C., Chang, F., Chen, Z., Liu, D. Fast traffic sign recognition via high-contrast region extraction and extended sparse representation. *IEEE Trans. Intell. Transp. Syst.* New York, 2016, 153 с.
3. Varun, S., Singh, S., Kunte, R.S., Samuel, R.S., Philip, B. A road traffic signal recognition system based on template matching employing tree classifier. *International Conference on Computational Intelligence and Multimedia Applications ICCIMA*. (Sivakasi, India, 13–15 December 2007). C. 360–365.
4. Ruta, A., Li, Y., Liu, X. Detection, tracking and recognition of traffic signs from video input. *11th International IEEE Conference on Intelligent Transportation Systems*. (ITSC 2008, Beijing, China, 12–15 October 2008). C. 55–60.
5. Jiang, Y., Zhou, S., Jiang, Y., Gong, J., Xiong, G., Chen, H. Traffic sign recognition using ridge regression and Otsu method. *2011 IEEE Intelligent Vehicles Symposium (IV)*. (Baden-Baden, Germany, 5–9 June 2011). C. 613–618.
6. Vázquez-Reina, A., Lafuente-Arroyo, S., Siegmann, P., Maldonado-Bascón, S., Acevedo-Rodríguez, F. Traffic sign shape classification based on correlation techniques. *5th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*. (Alcalá de Henares, Malta, 15–17 September 2005). C. 149–154.
7. Maldonado-Bascón, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gómez-Moreno, H., López-Ferreras, F. Road-sign detection and recognition based on support vector machines. *IEEE Trans. Intell. Transp. Syst.* 2007, c. 264–278.
8. Jiménez, P.G., Bascón, S.M., Moreno, H.G., Arroyo, S.L., Ferreras, F.L. Traffic sign shape classification and localization based on the normalized

- FFT of the signature of blobs and 2D homographies. *Signal Process.* 2008, c. 2943–2955.
9. Wu, J.-Y.; Tseng, C.-C.; Chang, C.-H.; Lien, J.-J.J.; Chen, J.C.; Tu, C.T. Road sign recognition system based on GentleBoost with sharing features. *2011 International Conference on System Science and Engineering (ICSSE)*, Macao, China, 8–10 June 2011; c. 410–415.
 10. Nicchiotti, G.; Ottaviani, E.; Castello, P.; Piccioli, G. Automatic road sign detection and classification from color image sequences. In *Proceedings of the 7th International Conference on Image Analysis and Processing*, 1994; pp. 623–626.
 11. Priese, L.; Rehrmann, V. On hierarchical color segmentation and applications. *1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 15–17 June 1993; c. 633–634
 12. Swain, M.J., Ballard, D.H. Color indexing. *Int. J. Comput. Vis.* 1991, C. 11–32.
 13. Fan, J., Yau, D.K., Elmagarmid, A.K., Aref, W.G. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Trans. Image Process.* 2001. C. 1454–1466.
 14. Розроблення стартап-проекту [Електронний ресурс]: Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.
 15. Головацький І.В., Корнага Я.І. Інтелектуальна система розпізнавання елементів дорожнього руху. *Вчені записки Таврійського Національного Університету Імені В.І. Вернадського.* Том 30(69) №6 2019 частина 2. С. 34-35.

ДОДАТКИ

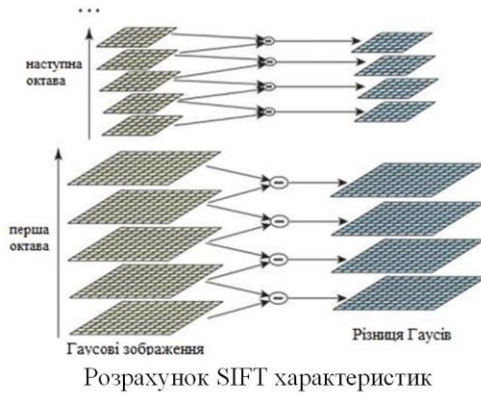
ДОДАТОК А

Схема роботи програмного додатку

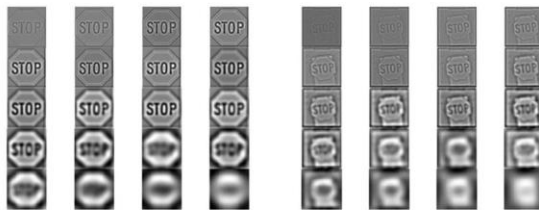


ДОДАТОК Б
Алгоритм роботи системи

Приклад обробки зображення



Розрахунок HSV простору кольору для
вхідного зображення

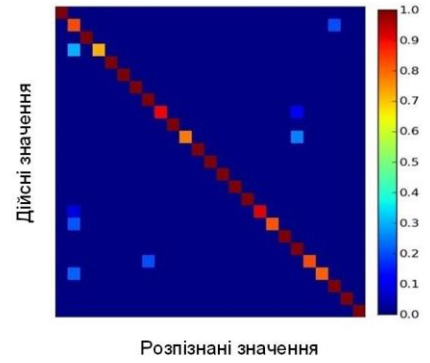
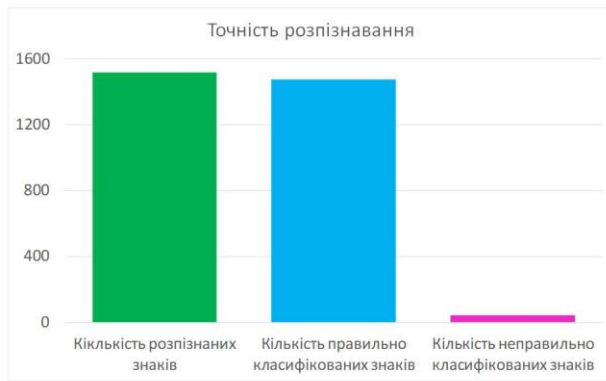


Різниця гаусів для різних октав і масштабів

Демонстраційний плакат № 2_
до магістерської дисертації на тему
Інтелектуальна система розпізнавання елементів дорожнього
руху
Розробив: Головацький І.В.
Приймав: Корнага Я.І.

ДОДАТОК В
Результати експериментів

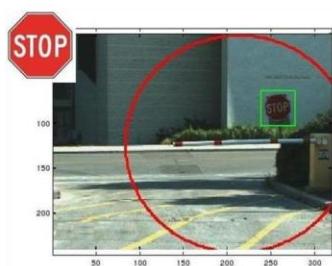
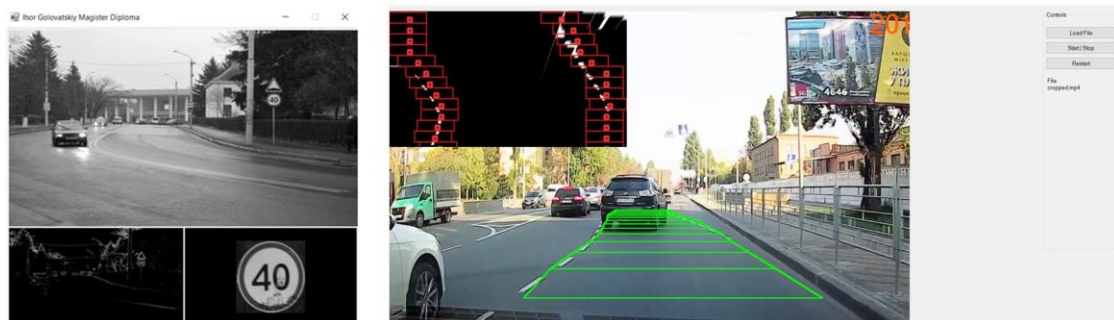
Результати експериментів



Демонстраційний плакат № 3
до магістерської дисертації на тему
Інтелектуальна система розпізнавання елементів дорожнього
руху
Розробив: Головацький І.В.
Приймав: Корнага Я.І.

ДОДАТОК Г
Результати експериментів

Інтерфейс користувача системи



Демонстраційний плакат № 4
до магістерської дисертації на тему
Інтелектуальна система розпізнавання елементів дорожнього
руху
Розробив: Головацький І.В.
Приймав: Корнага Я.І.