

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Допущено до захисту»

Завідувач кафедри
комп'ютерних наук,
доктор технічних наук, професор

_____ А. П. БОНДАРЧУК
(підпис)
« ____ » _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «Магістр»

Спеціальність 122 Комп'ютерні науки

Освітня програма 122.00.02 Інформаційно-аналітичні системи

Тема роботи: Визначення ефективності застосування апаратних платформ на прикладі задачі розпізнавання автомобільних номерів

Виконав
студент групи ІАСм-1-24-1.4д
(шифр академічної групи)

Яковенко Вадим Андрійович
(прізвище, ім'я, по батькові)

(підпис)
Науковий керівник
Кандидат технічних наук,
старший науковий співробітник
(науковий ступінь, наукове звання)

БАГАЦЬКИЙ О. В.
(підпис)

Київ – 2025

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»
Завідувач кафедри
комп'ютерних наук
канд.техн.наук, доцент

(підпис) **І. В. МАШКІНА**

ЗАВДАННЯ НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

студент групи **ІАСм-1-24-1.4д**

Яковенко Вадим Андрійович

Тема роботи: Визначення ефективності застосування апаратних платформ на прикладі задачі розпізнавання автомобільних номерів

1. Вихідні дані: Наукові праці та публікації з тематики автоматичного розпізнавання номерних знаків; технічні характеристики сучасних апаратних платформ типу SoC; документація OpenCV, Tesseract, OpenALPR; матеріали з комп'ютерного зору та систем відеоаналізу.
2. Основні завдання: Провести огляд сучасних методів і засобів автоматичного розпізнавання автомобільних номерів. Класифікувати апаратні платформи, що можуть бути застосовані для ALPR-систем. Виділити ключові характеристики, які впливають на ефективність обробки відеоданих. Розробити методику тестування ефективності платформ. Провести експериментальні дослідження на Raspberry Pi 3 Model B, Orange Pi Zero 2W та NVIDIA Jetson Nano. Порівняти швидкість обробки, точність та економічну доцільність платформ. Сформулювати рекомендації щодо вибору оптимальної платформи для ALPR. Оформити пояснювальну записку та підготувати презентацію.
3. Пояснювальна записка: Обсяг – до 62 сторінок формату A4 комп'ютерного набору з дотриманням вимог стандарту та методичних рекомендацій кафедри.
4. Графічні матеріали: презентація.
5. Додатки: Код програми, приклад результатів розпізнавання на Orange Pi
6. Строк подання роботи на кафедру: *«1» грудня 2025 р.*

Науковий керівник

кандидат технічних наук, старший
науковий співробітник

(науковий ступінь, наукове звання)

_____ Багацький О. В.

Завдання прийняв до виконання

« ____ » _____ 20__ р.

(підпис студента)

Яковенко В.А.

ІНДИВІДУАЛЬНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

студента групи ІАСм-1-24-1.4д

Яковенка Вадима Андрійовича

Тема роботи: Визначення ефективності застосування апаратних платформ на прикладі задачі розпізнавання автомобільних номерів

№	Назва етапу дипломної роботи	Строк виконання етапу роботи	Примітка
1	Вибір напрямку дослідження та формування теми дипломної роботи магістра	до 10 жовтня 2024р	Виконано
2	Ознайомлення з методичними рекомендаціями до дипломної роботи	до 10 жовтня 2024р	Виконано
3	Складання змісту та календарного плану роботи	до 15 листопада 2024р	Виконано
4	Підбір літератури, складання завдання	до 18 грудня 2024р	Виконано
5	Написання реферату та вступу	до 20 грудня 2024р	Виконано
6	Написання першого розділу	до 30 грудня 2024р	Виконано
7	Написання другого розділу	до 15 травня 2025р	Виконано
8	Написання третього розділу	до 15 жовтня 2025р	Виконано
9	Написання висновків	до 25 жовтня 2025р	Виконано

10	Виправлення зауважень	до 31 листопада 2025р	Виконано
12	Створення презентації	до 15 листопада 2025р	Виконано
13	Захист матеріалів дипломної роботи на передзахисті	за 1 місяць до захисту	Виконано
14	Офіційний захист матеріалів дипломної роботи на засіданні екзаменаційної комісії	згідно графіку захистів	

«Затверджую»

Індивідуальний план склав

Науковий керівник

_____ (підпис)

_____ (підпис студента)

« ____ » _____ 20__ р.

« ____ » _____ 20__ р.

Анотація кваліфікаційної роботи

Дипломна робота: 60 с., 5 рис., 5 табл., 44 посилань.

Актуальність: Автоматичне розпізнавання номерних знаків (ALPR) є ключовим елементом сучасних систем контролю доступу, моніторингу руху та безпеки. Ефективність таких систем значною мірою залежить від апаратної платформи, що виконує обробку відеоданих. З огляду на різноманітність доступних SoC-рішень, актуальною є оцінка їх ефективності з точки зору швидкості, точності та економічної доцільності.

Об'єкт дослідження: сучасні апаратні платформи, що можуть бути використані для розв'язку задачі розпізнавання автомобільних номерів в оптичному діапазоні.

Предмет дослідження: особливості та ефективність застосування апаратних платформ для задачі розпізнавання автомобільних номерів.

Мета роботи: визначити та оцінити ефективність застосування різних апаратних платформ для задачі розпізнавання автомобільних номерів, з урахуванням їх швидкості обробки даних, точності одержаних даних та економічної доцільності, для обґрунтування вибору оптимальної платформи в певних умовах використання.

Завдання роботи:

- Огляд методів ALPR; Аналіз платформ;
- Розробка методики тестування;
- Експериментальне порівняння Raspberry Pi 3 Model B, Orange Pi Zero 2W та Jetson Nano;
- Формування рекомендацій щодо вибору апаратної платформи.

Методи дослідження: Аналіз наукової літератури, аналіз продуктивності, порівняльний аналіз, експериментальні вимірювання.

Наукова новизна дослідження полягає у порівняльному аналізі ефективності застосування обраних апаратних платформ для задачі розпізнавання автомобільних номерів.

Практичне значення дослідження: розроблені та обґрунтовані рекомендації до вибору оптимальної платформи для бюджетних систем ALPR; застосування отриманих результатів у проєктах IoT, безпеки та відеоаналітики.

Ключові слова: ALPR, комп'ютерний зір, апаратні платформи, Raspberry Pi, Jetson Nano, Orange Pi, OCR, OpenCV, Python, Tesseract.

ЗМІСТ

АНОТАЦІЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ	6
ЗМІСТ	7
ВСТУП	8
1 АНАЛІЗ СУЧАСНИХ ІСНУЮЧИХ МЕТОДІВ І ЗАСОБІВ	12
1.1 Існуючі методи і способи обробки відеоданих, які використовуються для задачі автоматичного розпізнавання номерів	12
1.1.1 Класичні методи комп'ютерного зору (rule-based)	15
1.1.2 Глибоке навчання (Deep Learning)	16
1.2 Аналіз існуючих систем	18
1.2.1 Система Kobi Software (ZetPro / LPR BOX)	18
1.2.2 Система Rekor Scout (OpenALPR)	19
1.3 Аналіз апаратних платформ для систем ALPR	22
1.3.1 Raspberry PI	22
1.3.2 NVIDIA Jetson Nano	24
1.3.3 Orange Pi	26
1.3.4 Порівняння розглянутих платформ	27
2 ОГЛЯД ТА ОБҐРУНТУВАННЯ ОБРАНИХ МЕТОДІВ ТА ЗАСОБІВ	29
2.1 ОБРАНИ МЕТОДИ ALPR	29
2.2 ОБРАНИ АПАРАТНІ ПЛАТФОРМИ	29
2.2.1 Raspberry PI 3 Model B	30
2.2.2 NVIDIA Jetson Nano	30
2.2.3 Orange Pi Zero 2W 4GB	31
2.3 ОБРАНИ ПРОГРАМНІ ЗАСОБИ	32
2.3.1 Мова програмування Python	32
2.3.2 Бібліотека OpenALPR	32
2.3.3 Бібліотека OpenCV	32
2.3.4 Бібліотека Tesseract	32
2.4 КРИТЕРІЇ ОЦІНКИ ОТРИМАНИХ РЕЗУЛЬТАТІВ	33
2.5 ПРОГРАМА ДЛЯ ЕКСПЕРИМЕНТАЛЬНИХ ЗАПУСКІВ ALPR	33
3 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ ТА ОЦІНКА РЕЗУЛЬТАТІВ	36
3.1 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ	36
3.2 ФОРМУВАННЯ РЕКОМЕНДАЦІЙ	39

ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
Додаток А	45
Додаток Б	48

ВСТУП

Розпізнавання автомобільних номерів є важливим елементом сучасних систем автоматизації транспортних і безпекових процесів, включаючи контроль доступу, облік транспортних засобів, моніторинг дорожнього руху та виявлення правопорушень. Ефективність таких систем значною мірою залежить від апаратного забезпечення, яке дозволяє обробку великих обсягів даних у реальному часі.

Зараз на ринку присутня велика кількість апаратних платформ, які побудовані на “System on Chip” (SoC) і мають різні характеристики швидкості обробки даних, енергоспоживання, вартості та простоти інтеграції у існуючі системи. Однак існує невелика кількість праць,[1][2][3] в яких порівнюється ефективність цих платформ у контексті розпізнавання автомобільних номерів, враховуючи специфічні вимоги до точності, швидкості роботи та ресурсоемності.

Таким чином, дослідження цієї теми дозволить знайти оптимальні рішення для реалізації таких систем, що є важливим як з наукової, так і з прикладної точки зору.

Мета роботи: визначити та оцінити ефективність застосування різних апаратних платформ для задачі розпізнавання автомобільних номерів, з урахуванням їх швидкості обробки даних, точності одержаних даних та економічної доцільності, для обґрунтування вибору оптимальної платформи в певних умовах використання.

Об'єктом дослідження є сучасні апаратні платформи, що можуть бути використані для розв'язку задачі розпізнавання автомобільних номерів в оптичному діапазоні.

Предметом дослідження є особливості та ефективність застосування апаратних платформ для задачі розпізнавання автомобільних номерів.

Завдання дослідження

1. Огляд літератури та існуючих рішень

Вивчити сучасний стан розробок у сфері розпізнавання автомобільних номерів, приділивши увагу використовуваним апаратним платформам і алгоритмам.

2. Класифікація апаратних платформ

Визначити основні типи апаратних платформ, які побудовані на “System on Chip” (SoC) та їх характеристики, які впливають на швидкість обробки даних у задачах комп’ютерного зору.

3. Вибір критеріїв оцінки ефективності

Сформулювати критерії для оцінки ефективності платформ, такі як швидкість обробки даних, ймовірність вірного результату розпізнавання, можливість масштабування рішення і її вартість.

4. Розробка методики тестування

Розробити тестову методику, яка передбачає використання однакових алгоритмів розпізнавання номерів на різних платформах, а також створити набір тестових даних.

5. Експериментальна перевірка

Виконати серію експериментів з використанням обраних апаратних платформ, виміряти швидкість обробки даних і ймовірність вірного результату розпізнавання, а також інші показники.

6. Порівняння продуктивності платформ

Проаналізувати отримані результати експериментів, порівнявши платформи за визначеними критеріями.

7. Аналіз впливу параметрів

Дослідити вплив обсягу даних та умов експлуатації на ефективність роботи різних платформ.

8. Оцінка економічної доцільності

Визначити співвідношення «вартість – ефективність» для кожної проаналізованої платформи та оцінити їх доцільність у різних сценаріях використання.

9. Вивчення реальних прикладів впровадження

Проаналізувати приклади реальних систем розпізнавання номерів, які використовують різні платформи.

10. Формування рекомендацій

Розробити рекомендації щодо вибору апаратних платформ для задачі розпізнавання номерів залежно від конкретних умов і вимог.

11. Аналіз перспектив розвитку

12. Узагальнення та висновки

Сформулювати узагальнені висновки щодо ефективності різних платформ і представити результати.

1 АНАЛІЗ СУЧАСНИХ ІСНУЮЧИХ МЕТОДІВ І ЗАСОБІВ

1.1 Існуючі методи і способи обробки відеоданих, які використовуються для задачі автоматичного розпізнавання номерів

Системи автоматичного розпізнавання номерних знаків (ALPR — Automatic License Plate Recognition) базуються на послідовному виконанні ряду етапів, кожен з яких виконує специфічну функцію у загальному процесі. Якість і ефективність роботи всієї системи залежить від коректного виконання кожного етапу, від захоплення зображення до роботи з результатами або керуванням зовнішніми пристроями. Нижче наведено детальний опис основних етапів, які зазвичай реалізуються в таких системах.

1) Захоплення відео

Це перший та базовий етап системи ANPR, на якому здійснюється отримання відеопотоку або окремих кадрів за допомогою відеокамери. Якість даних, отриманих на цьому етапі, безпосередньо впливає на точність та швидкість подальшої обробки. У системах розпізнавання номерів зазвичай використовуються спеціалізовані LPR-камери (License Plate Recognition), які мають високу роздільну здатність (1080p і вище), високу частоту кадрів (до 60 fps і більше), інфрачервоне підсвічування (IR) для нічної зйомки, а також функції компенсації засвічення (BLC, HLC) та широкого динамічного діапазону (WDR) для роботи у складних умовах освітлення. (Наприклад, ZKTeco BL-852Q38A-LP, TVT TD-9423A3-LR, Provision-ISR I6-320LPR-MVF1)

Правильне встановлення камери також має вирішальне значення: важливо забезпечити оптимальний кут огляду, мінімізацію спотворень і стабільну фокусну відстань до об'єкта. Особливо це критично на шосе або на територіях зі складними погодними умовами. Для практичної реалізації часто використовують IP-камери з підтримкою RTSP-протоколу, що дозволяє передавати відео у реальному часі на обчислювальний модуль і, таким чином,

створювати системи «хмарної» або «крайової» обробки цих даних. Такі камери, як Hikvision Dahua IPC-HFW5541E-ZE, Uniview IPC2128SB-ADF28KM,[4][5] підтримують режими «License Plate Mode», налаштування зон детекції та оптимізацію експозиції під номерні знаки.

2) Попередня обробка зображення

Цей етап спрямований на підвищення якості зображення та виділення корисної інформації з потоку пікселів. Зокрема, застосовуються методи фільтрації шуму (медіанний, фільтр Гауса), нормалізації яскравості, підвищення контрасту (наприклад, CLAHE – Adaptive Histogram Equalization), а також перетворення зображення у відтінки сірого або бінаризація (порогова обробка). Мета цієї обробки — привести вхідне зображення до оптимального вигляду для детекції номерного знака.

Такі перетворення дозволяють зменшити обсяг зайвих деталей, виділити контури об'єктів і підготувати дані для сегментації. Важливо, щоб обробка була достатньо швидкою й оптимізованою під апаратну платформу — наприклад, на Raspberry Pi може використовуватись спрощений алгоритм з попередньою зменшеною роздільною здатністю кадру.

3) Виділення області номерного знака (детекція)

На цьому етапі виконується пошук та визначення конкретної області на зображенні, де ймовірно знаходиться номерний знак. У класичних підходах застосовують виявлення контурів, фільтрацію за геометричними ознаками (прямокутна форма, співвідношення сторін), а також перетворення Хафа для визначення паралельних ліній.

У більш сучасних реалізаціях використовуються алгоритми глибокого навчання, наприклад YOLO (You Only Look Once)[6], які можуть точно виявляти номерні знаки в різних умовах — при нахилі, частковому перекритті, у русі. Особливо ефективно працює комбінація YOLO + OpenCV,

де YOLO визначає координати об'єкта, а OpenCV забезпечує попередню обробку.[7]

Для практичного впровадження на апаратних платформах на кшталт Jetson Nano використовують оптимізовані моделі у форматі TensorRT[8] або ONNX[9], що дозволяє забезпечити високу продуктивність при низькому енергоспоживанні.

4) Сегментація символів

Після того як номерний знак було знайдено, необхідно «вирізати» з нього окремі символи для подальшого розпізнавання. Це завдання ускладнюється різними шрифтами, фоном, кутами нахилу, наявністю зайвих елементів (рамки, наклейки). Сегментація зазвичай базується на вертикальних гістограмах піксельної щільності, виявленні пробілів між символами та виділенні прямокутних областей з потрібною висотою й шириною.

Сучасні алгоритми також здатні виконувати сегментацію на основі глибоких нейронних мереж, уникаючи попередньої бінаризації. Це дозволяє зменшити втрати інформації при складних зображеннях (наприклад, при поганій погоді).

5) Розпізнавання символів (OCR)

Оптичне розпізнавання символів (OCR) — це процес перетворення зображень символів у текст. Для класичних систем використовується бібліотека Tesseract, яка може бути попередньо натренована на певний шрифт номерних знаків. У більш складних випадках застосовують глибокі мережі: CRNN (Convolutional Recurrent Neural Network)[10] або CNN+LSTM[11], які можуть обробляти увесь рядок без необхідності попередньої сегментації.

Нейромережі краще адаптуються до шрифтів, дефектів та умов освітлення. Для прискорення обробки моделі можна оптимізувати й запускати у середовищах з підтримкою GPU або спеціалізованих чипів.

б) Аналіз, логування, реакція системи

На заключному етапі обробки відеоданих система приймає рішення на основі розпізнаних даних. Наприклад:

- Зберігає інформацію про автомобіль у базу даних.
- Порівнює з "білим" або "чорним" списком для надання/заборони доступу.
- Активує пристрої контролю (відкриття шлагбаума, подача сигналу).
- Відправляє дані в систему правопорядку (фіксація порушення).

У цьому етапі важливо забезпечити надійність логування, таймстемпування, додавання міток часу до кожного запису, та синхронізацію з відео. Можлива реалізація — створення локального («крайового») сервера або використання «хмарного» сервера та API для централізованої обробки запитів, використання протоколів передачі даних MQTT або HTTP для зв'язку між IoT-пристроями та сервером.

Після розгляду основних етапів, які проходить відеодані в системі автоматичного розпізнавання номерів, важливо звернути увагу на способи реалізації кожного з цих етапів на програмному рівні. На практиці використовується декілька підходів до розробки таких систем, кожен з яких має свої переваги, обмеження та вимоги до апаратної платформи. Залежно від умов використання (наявність ресурсів, складність середовища, потрібна точність), розробник може обрати один із підходів або їхню комбінацію.

Нижче докладно описано два найпоширеніші методи — класичний комп'ютерний зір та глибоке навчання, які активно використовуються у проектах ALPR.

1.1.1 Класичні методи комп'ютерного зору (rule-based)

Цей підхід базується на використанні «класичних» алгоритмів обробки зображень без залучення штучного інтелекту або навчальних моделей. Його перевага — простота реалізації, що робить його популярним для вбудованих систем із обмеженими ресурсами, наприклад Raspberry Pi або ESP32-CAM.

На практиці такий підхід включає:

- Попередню обробку зображення: приведення до відтінків сірого, фільтрація шуму, покращення контрасту.
- Виділення контурів за допомогою алгоритмів Кенні (Canny)[12] або Собела (Sobel)[13], що дозволяє ідентифікувати області з високою різницею інтенсивності пікселів — типову ознаку рамки номерного знака.
- Перетворення Хафа (Hough transform)[14] застосовується для виявлення прямих ліній, які формують прямокутну область знака.
- Сегментація символів[15] — аналіз вертикальної та горизонтальної гістограми пікселів для виявлення меж символів.
- OCR (оптичне розпізнавання символів) — часто реалізується за допомогою бібліотеки Tesseract[16] або її легковажних аналогів, натренованих під конкретний шрифт номерів.

Цей підхід не потребує складного налаштування апаратно-програмної платформи, легко масштабується, однак його головним недоліком є низька стійкість до зовнішніх факторів: забруднення, нахили, нестандартні шрифти, погане або, навпаки, сильне освітлення об'єктів.[17] У реальних умовах,

особливо на дорогах із інтенсивним рухом або при швидкісній зйомці, такі системи демонструють нижчу точність, у порівнянні з статичними розпізнаваннями.[18]

Проте, для закритих територій, паркінгів із контрольованим середовищем або прототипів систем це рішення успішно використовується і сьогодні.

1.1.2 Глибоке навчання (Deep Learning)

Глибоке навчання є сучасним і найефективнішим підходом до реалізації систем ALPR. На відміну від класичних методів, які базуються на жорстко заданих правилах, моделі глибокого навчання самостійно навчаються виявляти ознаки номерів і розпізнавати символи на основі великої кількості прикладів. Це дозволяє системі бути значно стійкішою до зовнішніх умов, таких як різні кути зйомки, шрифти, освітлення, погодні умови, тощо.

Основні компоненти реалізації ALPR з використанням Deep Learning:

- YOLO (You Only Look Once)[6] — одна з найпопулярніших моделей для детекції об'єктів. Використовується для виявлення області номерного знака на зображенні. Завдяки своїй швидкодії, підходить для обробки відео в реальному часі.
- CRNN (Convolutional Recurrent Neural Network)[10] або CNN+LSTM[11] — архітектури для розпізнавання символів без явної сегментації. Вони вміють працювати з рядками тексту як із цілісними послідовностями, що значно покращує точність при обробці зображень номерів.
- EfficientNet, MobileNet, ResNet [19][20][21] — легкі або глибокі базові архітектури, які можуть бути використані для задач класифікації, сегментації або як основа для інших моделей.

Такі системи можуть працювати як на потужних серверах із GPU так і на спеціалізованих пристроях, таких як NVIDIA Jetson Nano, Xavier NX чи Google Coral, які підтримують апаратне прискорення inference (виконання попередньо навченої моделі для передбачення).

Головна перевага «глибокого навчання» — висока точність і адаптивність до змін обставин (Світло, погода і т.п.), у порівнянні з «класичними» підходами, що дозволяє застосовувати таку систему навіть у несприятливих умовах. Недолік такого типу систем – високі вимоги до ресурсів і потреба у великій навчальній вибірці (датасеті), а також складність початкової підготовки моделей.[22][23]

На практиці часто використовується підхід «transfer learning» — використання вже натренованої моделі, яка адаптується (донавчається) під локальні умови.

Під час розробки системи автоматичного розпізнавання номерів важливо враховувати не лише точність, а й обчислювальні можливості апаратної платформи, стабільність роботи в реальних умовах та гнучкість системи до змін. «Класичні» методи комп'ютерного зору є простішими у реалізації та менш вимогливими до ресурсів, однак показують обмежену ефективність у складних умовах. Натомість системи, побудовані на основі глибокого навчання, демонструють високу точність і стійкість, проте потребують більш потужного обладнання та складнішої початкової підготовки.

Тому у випадках, коли пріоритетом є швидкість розробки та економія ресурсів, доцільним є використання «класичних» методів комп'ютерного зору. Проте якщо проєкт вимагає високої якості отриманих даних, стабільної роботи в динамічних умовах і масштабованості — найкращим вибором стане підхід на основі глибокого навчання.[24] Враховуючи особливості IoT-проєктів і можливість використання сучасних енергоефективних

платформ типу NVIDIA Jetson, саме глибоке навчання можна вважати перспективнішим для систем розпізнавання номерів у промислових або муніципальних умовах.

У цьому дослідженні використовуються «класичні» методи комп'ютерного зору, що обумовлюється можливістю їх розгортання на дешевих апаратних платформах та наявністю практичних реалізацій обробки відеоданих. Всі проведені дослідження й сформовані на їх основі рекомендації будуть відноситись до систем, які використовують «класичні» методи комп'ютерного зору.

1.2 Аналіз існуючих систем

1.2.1 Система Kobi Software (ZetPro / LPR BOX)

Система автоматичного розпізнавання номерних знаків від української компанії Kobi є прикладом локального рішення, орієнтованого на інтеграцію в сучасні комплекси контролю доступу. Продукти цієї лінійки активно використовуються на промислових об'єктах, приватних територіях, а також для організації платного паркування та шлагбаумів на житлових комплексах.

В основі апаратної частини рішення лежать два ключові компоненти — ZetPro-камери та окремі пристрої LPR BOX, які можуть працювати як самостійні обчислювальні вузли. Камери ZetPro здатні здійснювати повну обробку зображень безпосередньо на борту, тобто без потреби в зовнішньому сервері. Такі камери підтримують розпізнавання номерів транспортних засобів, що рухаються зі швидкістю до 170 км/год, і можуть зберігати оброблені дані на SD-карті. Окрему увагу в цій системі приділено функції розпізнавання марки, моделі та кольору автомобіля, що значно підвищує точність при виявленні конкретного транспортного засобу серед потоку машин.[25]

У випадках, коли використовуються звичайні IP-камери, до системи підключаються окремі блоки LPR BOX: це такі модулі, як Orthus, Cyclops або

Hydra. Вони дозволяють інтегрувати функцію ALPR в існуючу систему відеоспостереження, навіть якщо самі камери не мають вбудованого інтелектуального функціоналу. Наприклад, пристрій Cyclops має підтримку 3G/4G-зв'язку, що дозволяє йому працювати повністю автономно та передавати дані у хмару або локальний сервер без участі інших компонентів. Система здатна розпізнавати номери транспортних засобів навіть на швидкості до 190 км/год, що робить її придатною для встановлення біля швидкісних трас.[26][27]

Програмне забезпечення, що використовується разом із цими рішеннями, має назву ZetPro VMS. Це центральна керуюча система, яка обробляє відеопотоки з RTSP-камер, аналізує їх у реальному часі, розпізнає номери автомобілів, і виконує автоматичні дії на основі отриманих результатів. Наприклад, при розпізнаванні номерного знаку з білого списку програма може самостійно подати сигнал на відкриття шлагбаума. У системі реалізовані додаткові аналітичні функції, як-от підрахунок кількості проїздів, генерація звітів, зберігання історії подій, а також можливість інтеграції з месенджером Telegram для віддаленого сповіщення адміністратора або користувача.[28]

Загалом, рішення від Kobi Software має високу точність розпізнавання (до 95%), підтримує роботу в автономному режимі (edge-модель), а також характеризується широкими можливостями адаптації під конкретні потреби користувача. Такий підхід робить систему ефективною для локального розгортання з обмеженим бюджетом, коли важливо забезпечити надійну ідентифікацію без необхідності підключення до інтернету чи зовнішніх сервісів.[27]

1.2.2 Система Rekor Scout (OpenALPR)

Система Rekor Scout, яка базується на платформі OpenALPR, є міжнародним лідером серед рішень для автоматичного розпізнавання номерів. Це рішення активно використовується в Сполучених Штатах, Канаді, країнах Європи та Азії. Його можна охарактеризувати як масштабовану та гнучку систему, розроблену для інтеграції у великі інфраструктурні проєкти — включаючи муніципальні системи відеоспостереження, паркінгові комплекси, аеропорти, порти та магістралі.

В апаратному плані компанія Rekor пропонує лінійку пристроїв під брендом Edge Max, які забезпечують обробку відеопотоку безпосередньо на місці встановлення. Такі пристрої здатні виявляти та ідентифікувати транспортні засоби на відстані до 30.48 метрів (100 футів) і при швидкості до 225 км/год. Це дозволяє встановлювати їх на об'єктах з високою інтенсивністю руху, наприклад, на автомагістралях або в пунктах контролю периметру. Завдяки підтримці GPU-прискорення, Edge-пристрої можуть обробляти відео в реальному часі, навіть при значному навантаженні.[29]

Програмне ядро цієї системи представлено рішенням Rekor Scout — це веб-інтерфейс для роботи з IP-камерами, що надає користувачу повний контроль над процесом розпізнавання: збереження відео, журнал подій, пошук за номерним знаком, створення сповіщень, формування аналітичних звітів. Програмна платформа підтримує понад 70 країн світу і здатна розпізнавати не лише номерний знак, а й марку, модель та колір автомобіля. Це робить систему придатною не лише для контролю доступу, а й для допомоги у кримінальних розслідуваннях та пошуку викрадених авто.

Окрім Scout, компанія також пропонує API-сервіс CarCheck, що дозволяє завантажувати окремі кадри або фотографії і отримувати результат у вигляді JSON-файлу. Цей сервіс особливо зручний для інтеграції у сторонні

додатки та мобільні системи, які потребують функції розпізнавання «на вимогу», без потреби у відеопотоці.[30]

Особливістю OpenALPR є те, що система базується на відкритих бібліотеках з підтримкою нейромережевої обробки зображень (зокрема, використовуються OpenCV і Tesseract), що дозволяє досягти високої точності та гнучкості. Усі обчислення можуть виконуватись як у хмарному середовищі, так і на локальних пристроях, а завдяки модульному підходу та контейнеризації її можна легко масштабувати та впроваджувати в архітектуру вже існуючих ІТ-систем.

Таким чином, Rekor Scout — це рішення, яке орієнтоване насамперед на масштабованість, гнучкість та розширену аналітику, і ідеально підходить для реалізації в умовах розподіленої інфраструктури або державних проєктів.

Таблиця 1.1 - Порівняння описаних систем

Система	Kobi Software (ZetPro/LPR BOX)	OpenALPR / Rekor Scout & Edge Max
Апаратна складова	Вбудоване LPR-процесування в камерах (до 190 км/год); LPR BOX (до 190 км/год, 3G/4G)	Edge Max – до 140 mph (~225 км/год), дальність до 30 метрів; оптимізовані для edge-середовища
Програмна складова	ZetPro VMS: RTSP, аналітика, інтеграція, off-line/edge	Rekor Scout: web-інтерфейс, оповіщення, аналітика; CarCheck API для зображень
Технологія розпізнавання	Власні нейромережі + класичні методи	Бібліотека OpenALPR (C++), GPT/Tesseract; GPU-оптимізація через Metropolis

Продовження таблиці 1.1

Масштабованість	Невеликі мережі – 1–20 камер; інтеграція через RTSP	Підтримка масштабних установок; container/GPU рішення; Edge та Cloud
Інтеграція	API, Telegram-інтеграція	JSON API, SDK (C#, Java, Python тощо)

Заявлена точність	~95%, розпізнавання марок/моделей	«industry leading», >90%, розпізнавання марок/моделей
Вартість	Локальні рішення, вартість залежить від установки	SaaS підписка (від \$10/місяць), Edge пристрої
Цільовий ринок	Паркінги, будівлі, безпека, український ринок	Глобальні рішення(~70 країн), органи правопорядку бізнеси й домовласники

Отже, на основі розгляду двох систем ALPR, можна зробити висновок, що Kobi Software ідеально підходить для локальних проєктів зі стабільним середовищем і обмеженим числом камер, де важлива автономність і локальна обробка. В той час як Rekor / OpenALPR це більш гнучке рішення, що підтримує великі інсталяції, використовуючи edge та cloud підходи, маючи широку географію підтримки і інтеграцію з існуючими ІТ-системами.

1.3 Аналіз апаратних платформ для систем ALPR

Ключову роль у побудові системи ALPR відіграє вибір апаратної платформи. Її вибір залежить від сфери застосування створюваної системи, фінансових можливостей та апаратних вимог обраного методу. У цьому розділі буде розглянуто популярні апаратні платформи, що можна використовувати для реалізації систем ALPR. Розглядатимуться апаратні реалізації, які мають формальну підтримку виробника, станом на час проведення дослідження. Також, буде проведено порівняння цих платформ на основі характеристик, що впливають на швидкість обробки даних у задачах комп'ютерного зору .

1.3.1 Raspberry PI

Raspberry PI – це серія одноплатних комп'ютерів на основі ARM мікроконтролера, створені британською благодійною організацією Raspberry Pi Foundation. Ці плати мають функціональні можливості порівнянні з персональним комп'ютером.

Мікрокомп'ютер Raspberry Pi 3 Model B – суперкомпактний високопродуктивний одноплатний комп'ютер, розміром трохи більше кредитної картки. Це найперша модель третього покоління Raspberry PI. Вона замінила модель Raspberry Pi 2 Model B лютого 2016. Основним призначенням Raspberry Pi є можливість доступного навчання програмуванню. Але також, її можливо використовувати для проектів різних ступенів складності.

Ключовими змінами в Raspberry PI 3 є новий 4-х ядерний процесор BCM2837 з тактовою частотою 1.2 ГГц. Це забезпечує вищу швидкодію при виконанні обчислювальних задач та роботі системи вцілому. Також, нова модель отримала кращий графічний процесор 400 MHz Dual Core VideoCore IV, у порівнянні з одноядерним Broadcom VideoCore IV 250 МГц

Не менш важливим нововведенням в цій моделі, у порівнянні з попередніми, є поява Wi-Fi та Bluetooth 4.1 модулів. Це дозволяє реалізовувати бездротове підключення пристроїв та підключення плати до мережі, що розширює можливості використання RPI3, у порівнянні з RPI1/2.

Основні характеристики:

- Процесор: Quad Core 1.2GHz Broadcom BCM2837 (SoC)
64bit ARM Cortex-A53
- Графічний прискорювач: 400 MHz Dual Core VideoCore IV GPU
- Графічні можливості: OpenGL ES 2.0, Hardware-Accelerated OpenVG, 1080p30 H.264 high-profile decode
- Оперативна пам'ять: 1 GB LPDDR2
- Слот для MicroSD карти
- Сумісність з операційними системами: NOOBS, Raspbian, LibreElec, Open Elec, OSMC, Pinet, RISC, OS, Snappy Ubuntu Core, Ubuntu Mate, Weather Station, Windows IOT Core, XBia
- Порти і роз'єми: 40-піновий роз'єм GPIO, 15-pin інтерфейс камери MIPI Camera Serial Interface, HDMI (rev 1.3 & 1.4), порт дисплея Display Serial Interface (DSI), 4 порти USB 2.0, RCA (PAL і NTSC), 3.5 мм аудіо вихід
- Вивід зображення: HDMI
- Мережа: Ethernet 10 / 100BA SE-T під RJ-45,
- Бездротовий зв'язок:
 - 802.11n WiFi wireless Networking; IEEE 802.11a / g / b / n compatible,

- Bluetooth 4.1
- Живлення: Джерело живлення Micro USB до 2,5 A[31]

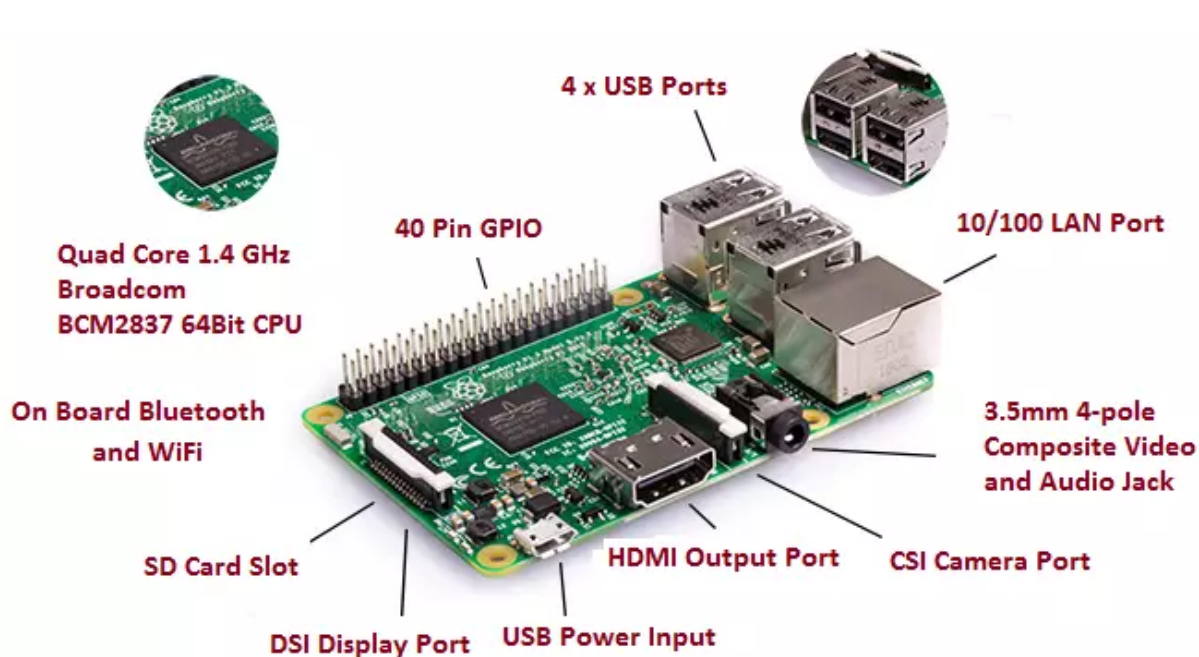


Рисунок 1.1 – Raspberry Pi 3 Model B

1.3.2 NVIDIA Jetson Nano

NVIDIA Jetson Nano – частина сімейства Jetson від NVIDIA, розробленого для периферійного штучного інтелекту — обробки даних локально на пристроях для зменшення затримки та підвищення конфіденційності. На відміну від хмарно-залежних систем, Nano поєднує можливості штучного інтелекту з компактним обладнанням, що робить його ідеальним для проектів робототехніки, Інтернету речей та комп'ютерного зору. Його доступність виділяла його з першого моменту запуску, конкуруючи з такими платами, як Raspberry Pi, пропонуючи обчислення з прискоренням CUDA для завдань машинного навчання.

У 2025 році Jetson Nano залишається актуальним для освітніх цілей, застарілих проектів та розробників, які цінують бюджет. Хоча NVIDIA випустила потужніших наступників (наприклад, Jetson Orin Nano та Orin

Nano Super), сумісність Nano з NVIDIA JetPack SDK та його активна спільнота підтримують його життя.

Jetson Nano має чотирьохядерний 64-бітний ARM процесор і 128-ядерний інтегрований графічний процесор NVIDIA на архітектурі Maxwell™ з 4 ГБ оперативної пам'яті формату LPDDR4, що забезпечує продуктивність 472 GFLOPS для швидкої роботи сучасних алгоритмів AI. Ці характеристики дозволяють паралельний запуск декількох нейронних мереж та одночасну обробку декількох камер високої роздільності.

Jetson Nano також підтримується NVIDIA JetPack, який включає в себе пакет підтримки плат (BSP), програмні бібліотеки CUDA, cuDNN і TensorRT для Deep Learning, комп'ютерного зору, обчислень на GPU, обробки мультимедіа та багато чого іншого. JetPack SDK також включає в себе можливість власної установки популярних платформ машинного навчання (ML) з відкритим вихідним кодом, таких як TensorFlow, PyTorch, Caffe / Caffe2, Keras і MXNet, що дозволяє розробникам швидко і легко інтегрувати свою улюблену AI модель.

Технічні характеристики:

GPU: 128-ядерний графічний процесор на архітектурі Maxwell 921MHz

Процесор: чотирьохядерний процесор на архітектурі ARM Cortex-A57 1.43GHz

Оперативна пам'ять: 4 ГБ, 64-бітна LPDDR4

Сховище: Слот для карти Micro SD (потрібна зовнішня карта на, мінімум, 16 ГБ)

Відео:

- Кодування: 1 x 4K30 (H.265), 2 x 1080p60 (H.265)

- Декодування: 1 x 4K60 (H.265), 4 x 1080p60 (H.265)

Інтерфейси:

- Ethernet: 10/100 / 1000BASE-T з автоматичним узгодженням
- Камера: 12-канальна (3x4 АБО 4x2) MIPI CSI-2 DPHY 1.1 (1,5 Гбіт / с)
- Дисплей: HDMI 2.0, DP (DisplayPort)
- USB: 4x USB 3.0, USB 2.0 (Micro USB)
- Інші: GPIO, 4 x I2C, 2 x I2S, 3 x SPI, 2 x UART

Живлення:

- Micro USB (5В 2А)
- Гніздо для подачі постійного струму (5 В, 4 А)

Елементи мікрокомп'ютера:

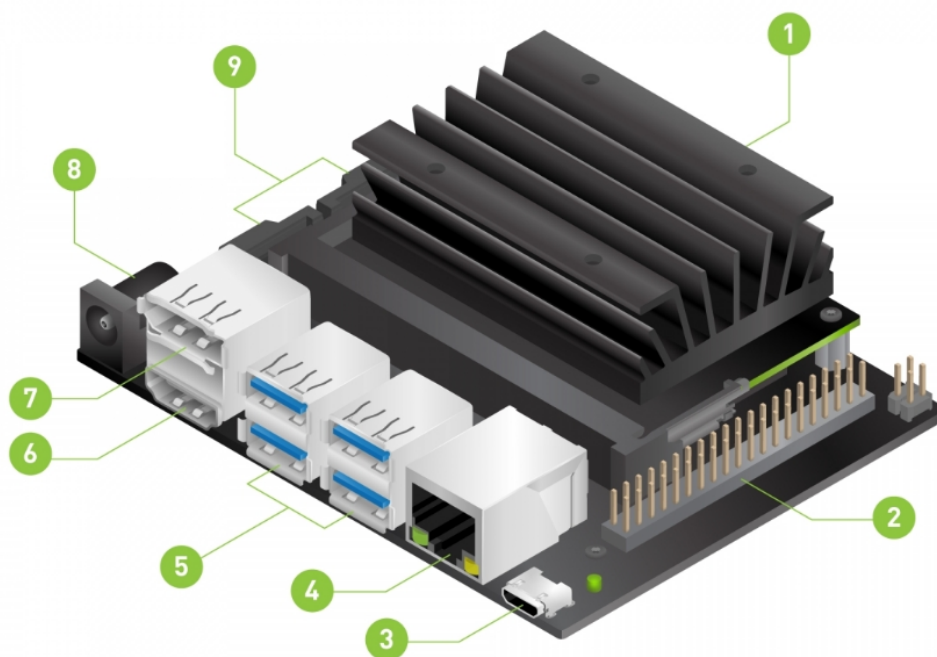


Рисунок 1.2 – NVIDIA Jetson Nano

1. Слот для карти microSD для основного сховища
2. 40-контактний роз'єм розширення GPIO
3. Порт Micro-USB для входу живлення 5В або для передачі даних
4. Порт Gigabit Ethernet
5. Порти USB 3.0 (4 од.)
6. Вихідний порт HDMI
7. Роз'єм DisplayPort
8. Роз'єм DC Barrel для входу живлення 5В
9. Роз'єми камери MIPI CSI-2[32]

1.3.3 Orange Pi

Orange Pi — це серія одноплатних комп'ютерів (SBC), що виробляються китайською компанією Shenzhen Xunlong Software. Ці пристрої призначені для різноманітних застосувань та проектів.

Мікрокомп'ютер Orange Pi Zero 2W 4GB — це потужний одноплатний комп'ютер, який надає широкі можливості для розробки різних проектів. Ця модель представляє собою аналог популярної Raspberry Pi Zero 2W.

Одним із головних його переваг, є потужний чотирьохядерний процесор Allwinner H618 Cortex-A53 з тактовою частотою до 1.5GHz та графічний процесор Mali G31 MP2 з частотою 650 МГц. Також, ця модель оснащена 4 ГБ оперативної пам'яті LPDDR4 й підтримує флеш-пам'ять SPI 16 МБ. В комбінації з наявністю Wi-Fi5.0+Bluetooth 5.0 антени та компактних розмірів ця модель є ефективним варіантом для проектів Інтернету речей.

Технічні характеристики:

- Процесор: Allwinner H618 чотирьохядерний Cortex-A53 1.5GHz
- Відеокарта: Mali G31 MP2 650 МГц
- Сумісність: OpenGL ES 1.0/2.0/3.2, OpenCL 2.0, Vulkan 1.1
- Оперативна пам'ять: LPDDR4:1GB/1.5GB/2GB/4GB
- Пам'ять: SPI Flash 16MB
- Слот для MicroSD картки
- USB: Type-C USB2.0 x 2
- Відеовихід: 1*Міні HDMI TX 2.0 порт, який підтримує 4K@60FPS
- Бездротовий зв'язок: Wi-Fi5.0+ Bluetooth 5.0
- Роз'єм розширення: 24Pin: USB2.0 x 2, 100M Ethernet інтерфейс, інтерфейс ІЧ-приймача, вихід аудіо, TV-вихід, кнопка живлення, визначені користувачем кнопки x 2.
- Джерело живлення: Type-C 5V2A, Type-C 5V3A
- Підтримувані ОС: Android 12 TV, Debian11, Debian12, Ubuntu22.04, Ubuntu20.04, Orange Pi OS (Arch)[33]

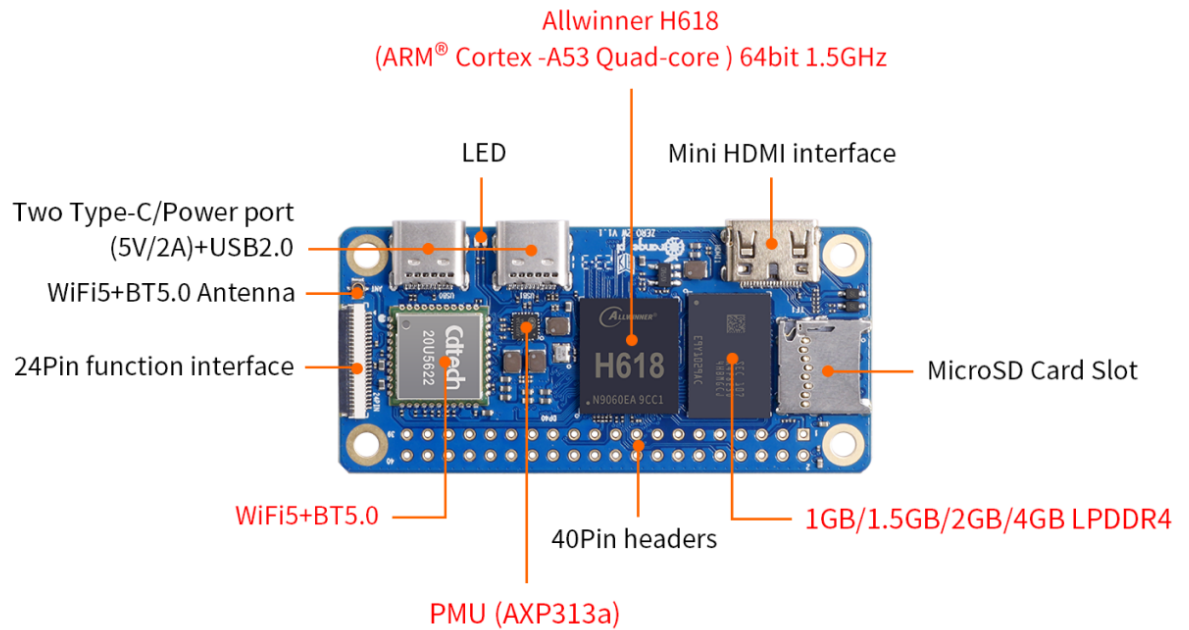


Рисунок 1.3 – Orange Pi Zero 2W 4GB

1.3.4 Порівняння розглянутих платформ

На основі розглянутих апаратних платформ можна сформувати таблицю, в якій порівняти характеристики цих платформ. Порівнюватимуться характеристики, які впливають на швидкість обробки даних у задачах комп'ютерного зору, такі як тактова частота ядер, кількість ядер та оперативної пам'яті. Також, буде порівняна ціна розглянутих платформ.

Кожна розглянута платформа має 4 ядра з тактовими частотами більше 1 ГГц, що дозволяє реалізацію багатопотокової обробки кадрів. Також, платформи, окрім Jetson Nano, яка потребує підключення USB-адаптера, мають можливість бездротового з'єднання за допомогою WiFi або Bluetooth, що спрощує роботу з ними.

Таблиця 1.2 - Порівняння характеристик апаратних платформ

Характеристика	Raspberry Pi 3 Model B	NVIDIA Jetson Nano	Orange Pi Zero 2W
Процесор (CPU)	Broadcom BCM2837 ARM Cortex-A53	MPCore Arm Cortex-A57	Allwinner H618 Arm Cortex-A53
Кількість ядер	4	4	4
Тактова частота	1.2ГГц	1.43 ГГц	1.5ГГц
Графічний процесор (GPU)	VideoCore IV (400 МГц)	128-core Maxwell (CUDA) (921 МГц)	Mali G31 MP2(650 МГц)
Оперативна пам'ять	1 GB LPDDR2	LPDDR4 (4 ГБ)	LPDDR4 (4 ГБ)
Інтерфейс камери	15-pin інтерфейс камери MIPI Camera Serial Interface	12-lanes MIPI CSI-2	Необхідний спеціальний модуль
Ціна	2300грн.~	9000грн.~	1500грн.~

На основі цієї таблиці можна зробити висновок, що найперспективнішими платформами для «класичних» методів комп'ютерного зору є Orange Pi Zero 2W та NVIDIA Jetson Nano. Їх потужні процесори, більша кількість оперативної пам'яті та вища її версія дозволяють швидко виконувати задачу розпізнавання. Raspberry Pi 3 Model B має слабший процесор та менший обсяг оперативної пам'яті, тож буде показувати повільніші результати у порівнянні з іншими платформами. Але її перевагою є наявність вбудованого інтерфейсу камери, на відміну від Orange Pi.

2 ОГЛЯД ТА ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ ТА ЗАСОБІВ

Для визначення і обрахування критерію ефективності необхідно визначити програмні та апаратні засоби, використані для проведення дослідження та обґрунтувати їх вибір. Також було описано обрані критерії, для оцінювання отриманих в ході експериментів результатів.

2.1 Обрані методи ALPR

При дослідженні матеріалів за обраною темою, було знайдено невелику кількість актуальних праць,[34] в яких би порівнювалась ефективність різних апаратних платформ, для вирішення задачі розпізнавання автомобільних номерів. Більшість знайдених робіт, у свою чергу, були основані на методах із застосуванням нейромереж та глибокого навчання. Також, розглянуті в цих роботах апаратні платформи є представниками сучасних платформ(на момент проведення вищезгаданих досліджень), що в свою чергу робить їх не найдешевшим варіантом. Тому було прийнято рішення в цій роботі провести дослідження на основі «класичних» методів комп'ютерного зору, через їх нижчі вимоги до апаратних платформ. Також, їхній вибір обумовлений низькою кількістю актуальних робіт за цими методами.

2.2 Обрані апаратні платформи

Вибір апаратних платформ базувався на технічних вимогах „класичних методів” комп'ютерного зору при застосуванні для задачі розпізнавання автомобільних номерів. Не менш важливими вимогами є економічна доцільність обраної платформи для поставленої задачі. Для цього було сформовано наступні критерії вибору платформи:

- Тактова частота ядра CPU – вища тактова частота дозволяє швидше проводити всі необхідні обчислення, чим пришвидшує процес розпізнання.

- Кількість ядер CPU – наявність декількох доступних ядер дозволяє пришвидшити обробку, за рахунок паралельного розпізнавання різних кадрів.
- Швидкість оперативної пам'яті – при обробці відеопотоку, відбувається постійний обмін великих масивів даних (розмір залежить від роздільної здатності оброблюваного відео) між ОЗП і процесором. Тож швидкість ОЗП напряду впливає на швидкість виконання всієї задачі в цілому.
- Кількість оперативної пам'яті – визначає максимальний об'єм інформації, який можна зберігати та передавати при роботі системи. Найбільше впливає на швидкодію у випадку паралельної обробки кадрів на різних ядрах, але також важлива при роботі з відео з високою роздільною здатністю.
- Ціна – є важливим критерієм вибору платформи, так як застосування „класичних методів” комп'ютерного зору є доцільним в системах, з обмеженим бюджетом.

Для цього дослідження обраними апаратними платформами на яких проводитимуться всі експерименти є Raspberry Pi 3 Model B, Orange Pi Zero 2W 4GB та NVIDIA Jetson Nano.

2.2.1 Raspberry PI 3 Model B

Raspberry Pi 3 Model B — це ультракомпактний одноплатний мінікомп'ютер, з високою продуктивністю за невисокої вартості. Наявність процесора Quad Core ARM Cortex-A53 з тактовою частотою 1.2ГГц надає достатню обчислювальну потужність для реалізації простої ALPR. Хоча 1 Гб оперативної пам'яті LPDDR2 не є сильною перевагою, але вона повністю задовольняє вимоги обраного методу ALPR. Не менш важливим є повня підтримка всіх обраних програмних засобів(OpenALPR, Tesseract, OpenCV).

У підсумку, Raspberry Pi 3 це компактна, енергоефективна та недорога платформа, яка добре підходить для реалізації простої системи комп'ютерного зору. Саме тому ця платформа була обрана в цьому дослідженні.

Обрана платформа має наступні специфікації:

Операційна система: Debian GNU/Linux 13(Trixie)

Ядро: Linux 6.12.47+rpt-rpi-v8

Архітектура: ARM64

Версія Python: 3.13.5

Версія OpenCV: 4.10

MicroSD: SanDisk microSDHC 16Gb C10 UHS-I U

2.2.2 NVIDIA Jetson Nano

NVIDIA Jetson Nano – це невеликий, але потужний мікрокомп'ютер призначений для роботи з ШІ. Він має потужний чотирьохядерний процесор MPCore Arm Cortex-A57 з тактовою частотою 1.43ГГц та 4 ГБ ОЗП LPDDR4. Ці технічні характеристики та підтримка обраних програмних засобів дозволяють реалізувати на цій платформі ALPR за обраними методами.

Вибір цієї апаратної платформи обумовлений подібними технічними характеристиками до інших обраних апаратних платформ, що дозволить краще оцінити отримані результати з усіх платформ.

Обрана платформа має наступні специфікації:

Операційна система: Ubuntu 18.04.6

Ядро: Linux 4.9.337-tegra

Архітектура: ARM64

Версія Python: 3.6.9

Версія OpenCV: 4.9

MicroSD: SanDisk microSD 64Gb C10 UHS-I U3

2.2.3 Orange Pi Zero 2W 4GB

Orange Pi Zero 2W – це компактний потужний одноплатний мікрокомп'ютер. Він позиціонується як аналог популярної Raspberry Pi Zero 2W, але за технічними характеристиками може конкурувати навіть з Raspberry Pi 3.

На відміну від аналогу, Orange Pi має потужніший процесор Allwinner H618 Arm Cortex-A53 з чотирма ядрами та тактовою частотою 1.5ГГц. Він має варіанти з 1/1.5/2/4 ГБ оперативної пам'яті LPDDR4. В цьому дослідженні розглядатиметься варіант з 4 ГБ ОЗП. Вагомою перевагою є й невисока ціна даної платформи.

Високі технічні характеристики та підтримка обраних програмних засобів, дозволяють реалізувати ALPR за обраними методами. Цим обумовлюється вибір цієї апаратної платформи для дослідження.

Обрана платформа має наступні специфікації:

Операційна система: Ubuntu 22.04.5 LTS

Ядро: Linux 6.1.31-sun50iw9

Архітектура: ARM64

Версія Python: 3.10.13

Версія OpenCV: 4.10

MicroSD: SDCS2/64GB-2P1A

2.3 Обрані програмні засоби

2.3.1 Мова програмування Python

Python — це потужна мова програмування, з великою кількістю підтримуваних бібліотек. [35] У цьому дослідженні цією мовою написано програму, яка захоплює відео, визначає номерний знак, проводить спробу розпізнання, розраховує швидкість обробки та виводить відповідні результати в консоль. Вибір цієї мови програмування обумовлений її простотою, підтримкою на обраних апаратних платформах та підтримкою обраних бібліотек.

2.3.2 Бібліотека OpenALPR

OpenALPR — це бібліотека з відкритим кодом для автоматичного розпізнавання номерних знаків, написана на C++ з прив'язками на C#, Java, Node.js, Go та Python. Бібліотека може аналізувати зображення або відео для ідентифікації номерних знаків. Результатом є текстове представлення будь-яких символів номерного знака. Для роботи цієї бібліотеки необхідні встановлені бібліотеки OpenCV та Tesseract.[36] Для дослідження було обрано цю бібліотеку за рахунок її доступності, простоти, підтримки на обраних апаратних платформах та використання „класичних методів” комп'ютерного зору. Використовувалась остання доступна версія 2.3.0.

2.3.3 Бібліотека OpenCV

OpenCV — це відкрита бібліотека комп'ютерного зору, яка призначена для аналізу, класифікації та обробки зображень. Вона підтримує мови програмування C, C++, Python і Java.[37] За допомогою цієї бібліотеки здійснюється захоплення відео, виділення з нього кадру та його преобробка перед розпізнанням. Цю бібліотеку було обрано за рахунок її доступності, простоти й підтримки на обраних апаратних платформах. Також, вона є необхідною для використання бібліотеки OpenALPR.

2.3.4 Бібліотека Tesseract

Бібліотека Tesseract – є відкритою бібліотекою для оптичного розпізнавання символів, написана на C/C++ й має обгортки для Java, Python, R і т.п. Це потужний інструмент для комп'ютерного зору, який найчастіше застосовується для розпізнавання символів із зображень. [38][39] Ця бібліотека була обрана для даного дослідження за рахунок її доступності, підтримці на обраних платформах, використання „класичних методів” для розпізнавання символів та необхідність для бібліотеки OpenALPR. Використовувалась версія 3.05.02, яка є останньою працюючою з OpenALPR версією та без використання нейромережових методів.

2.4 Критерії оцінки отриманих результатів

Для оцінювання ефективності роботи певної апаратної платформи було обрано наступні критерії.

Основним критерієм оцінки результатів є **"швидкість обробки, кадр/с"(FPS)**. Цей критерій відображає, наскільки швидко обрана платформа розв'язує поставлену задачу. Чим вище показник, тим краще результат. Так як данне дослідження орієнтується на системи для місць з невисокою швидкістю руху, то всі результати вище 0.5 кадр/с є прийнятними.

Значним, є наступний обраний критерій – **співвідношення ціни платформи до швидкості обробки 1 кадр/с**. Даний критерій дозволить оцінити економічну доцільність використання платформи для вирішення задачі розпізнавання автомобільних номерів. Чим вище отримане значення відношення, тим вигідніше використання платформи.

Не менш важливим критерієм є **точність розпізнавання**. Цей критерій дозволить оцінити доцільність використання платформи для поставленої задачі. Чим більше кількість успішно розпізнаних номерів до фактично присутніх на відео, тим ефективніше застосування платформи для обраної задачі. Важливу роль в результатах грає швидкість руху автомобіля. Для авто,

що рухаються швидше 15 км/год, прийнятною точністю розпізнання є результат вище 75% розпізнаного знаку.

2.5 Програма для експериментальних запусків ALPR

За допомогою описаних програмних засобів, було написано програму для розпізнавання автомобільних номерів з відеофайлу.

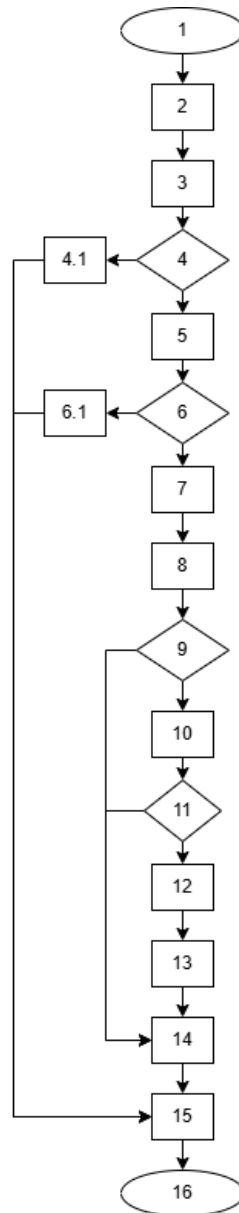


Рисунок 2.1 – Блок-схема програми

1. Початок
2. Імпорт необхідних бібліотек

3. Ініціалізація ALPR: встановлення країни, шляху до конфігураційного файлу та шляху для тимчасових файлів

4. Ініціалізація ALPR пройшла успішно? Так, перехід до пункту 5. Ні, перехід до пункту 4.1

4.1. Вивід повідомлення "Не вдалося завантажити OpenALPR" та перехід до пункту 15

5. Початок захоплення відео з відеофайлу

6. Початок захоплення успішний? Так, перехід до пункту 7. Ні, перехід до пункту 6.1

6.1. Вивід повідомлення "Камеру не вдалося відкрити" та перехід до пункту 15

7. Задавання змінних часу початку запуску та кількості кадрів.

8. Вивід повідомлення "Розпізнавання запущено. Натисни 'е' для виходу."

9. Клавішу 'е' натиснута? Так, перехід до пункту 14. Ні, перехід до пункту 10.

10. Захоплення та зчитування кадру з відео.

11. Захоплення та зчитування успішні? Так, перехід до пункту 12. Ні, перехід до пункту 14.

12. Розпізнавання номерного знаку на захопленому кадрі.

13. Вивід результатів розпізнавання та ймовірності у консоль. Збільшення кількості кадрів на один.

14. Розрахунок та виведення середньої швидкості обробки.

15. Зупинка захоплення відео. Закриття всіх відкритих після запуску програми вікон. Вивантаження ALPR.

16. Кінець

3 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ ТА ОЦІНКА РЕЗУЛЬТАТІВ

3.1 Проведення експериментів

Було проведено серію експериментів на апаратних платформах Raspberry PI 3 Model B, Orange PI Zero 2W та Jetson Nano. Суть експериментів полягала в запуску створеної програми на обраних апаратних платформах та фіксації отриманих результатів. Всього було проведено 4 експерименти, де в кожному експерименті були відмінні вхідні дані(Різні відеофайли). Кожен експеримент складається з 6 запусків програми на кожній платформі. Запуски проводилися послідовно й обробка відео була однопоточною.

Для проведення експериментів, було обрано 4 відеофайли:

- Відео день, мегаполіс: Відео формату .mp4(Кодек H. 264) з роздільною здатністю 720р. Його тривалість 6 секунд, від 0:28 до 0:35 в оригінальному відео. На цьому відео зйомка відбувається в хмарну погоду вдень.
- Відео день, місто: Відео формату .mp4(Кодек H. 264) з роздільною здатністю 1080р. Його тривалість 8 секунд, від 0:00 до 0:08 в оригінальному відео. На цьому відео зйомка відбувається в хмарну погоду вдень.
- Відео ніч: Відео формату .mp4(Кодек H. 264) з роздільною здатністю 2160р. Його тривалість 5 секунд, від 0:24 до 0:36 в оригінальному відео. На цьому відео зйомка відбувається в ясну погоду в сутінки.
- Відео дощ: Відео формату .mp4(Кодек H. 264) з роздільною здатністю 720р. Його тривалість 17 секунд, від 1:10 до 1:27 в оригінальному відео. На цьому відео зйомка відбувається в дощову погоду вдень.
 - З оригінального відео спочатку було обрано проміжок з 0:00 до 0:19, але при запуску програми на Raspberry Pi не було

жодних результатів. Саме тому було вирішено взяти інший фрагмент з відео.

За результатами проведених експериментів було отримано наступні данні.

```

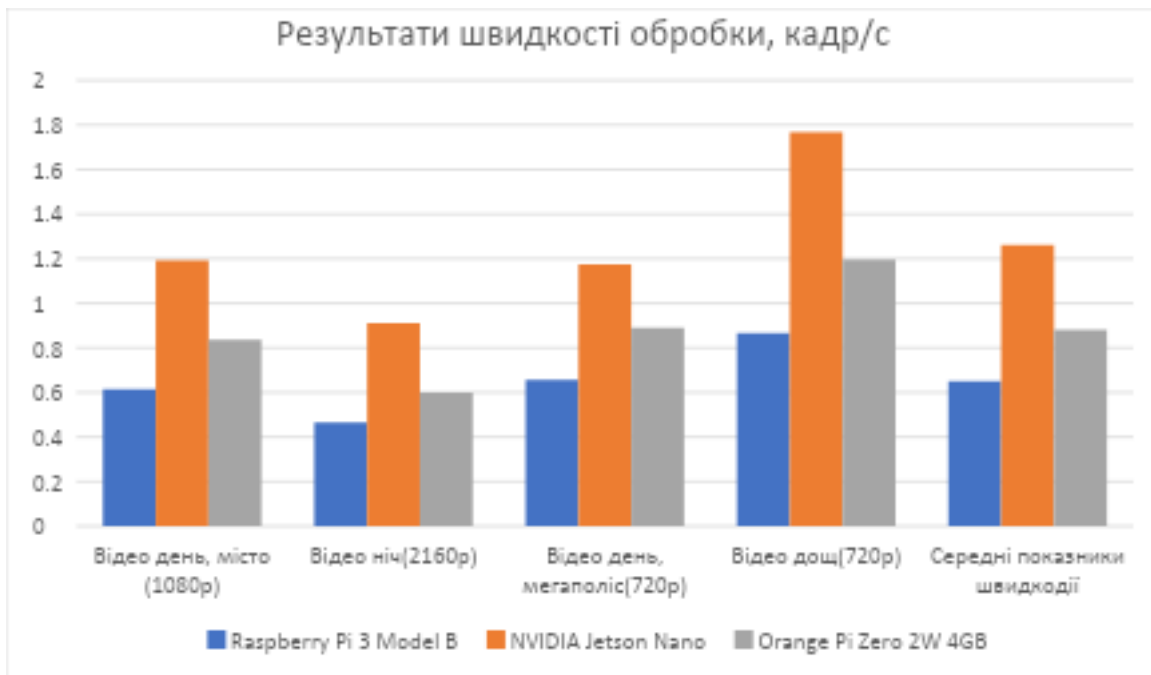
user@raspberrypi: ~/Desktop/sample
File Edit Tabs Help
user@raspberrypi:~ $ cd Desktop
user@raspberrypi:~/Desktop $ cd sample
user@raspberrypi:~/Desktop/sample $ python3 ALPR_test.py
Розпізнавання запущено. Натисни 'е' для виходу.
Номер: 0JUI3 (Ймовірність: 78.91%)
Номер: JHND (Ймовірність: 76.91%)
Номер: I11111 (Ймовірність: 89.79%)
Номер: W1111 (Ймовірність: 80.48%)
Номер: I0D11 (Ймовірність: 80.99%)
Номер: I111D (Ймовірність: 76.23%)
Номер: E1111 (Ймовірність: 76.87%)
Номер: 0K09B (Ймовірність: 83.57%)
Номер: GK09BJE (Ймовірність: 85.78%)
Номер: J9ILI (Ймовірність: 79.40%)
Неможливо зчитати кадр
Середня швидкість: 0.60 кадр/с
user@raspberrypi:~/Desktop/sample $

```

Рисунок 3.1 – Приклад отриманих результатів, після запуску програми на Raspberry Pi з Відео день, місто(1080p)

Таблиця 3.1 – Результати швидкості обробки, кадр/с (FPS)

Відео	Апаратна платформа	Raspberry PI 3 Model B	NVIDIA Jetson Nano	Orange Pi Zero 2W 4GB
Відео день, місто (1080p)		0,615	1,192	0,837
Відео ніч(2160p)		0,465	0,912	0,6
Відео день, мегаполіс(720p)		0,6567	1,175	0,89
Відео дощ(720p)		0,8667	1,768	1,195
Середні показники швидкодії		0,65085	1,26175	0,8805



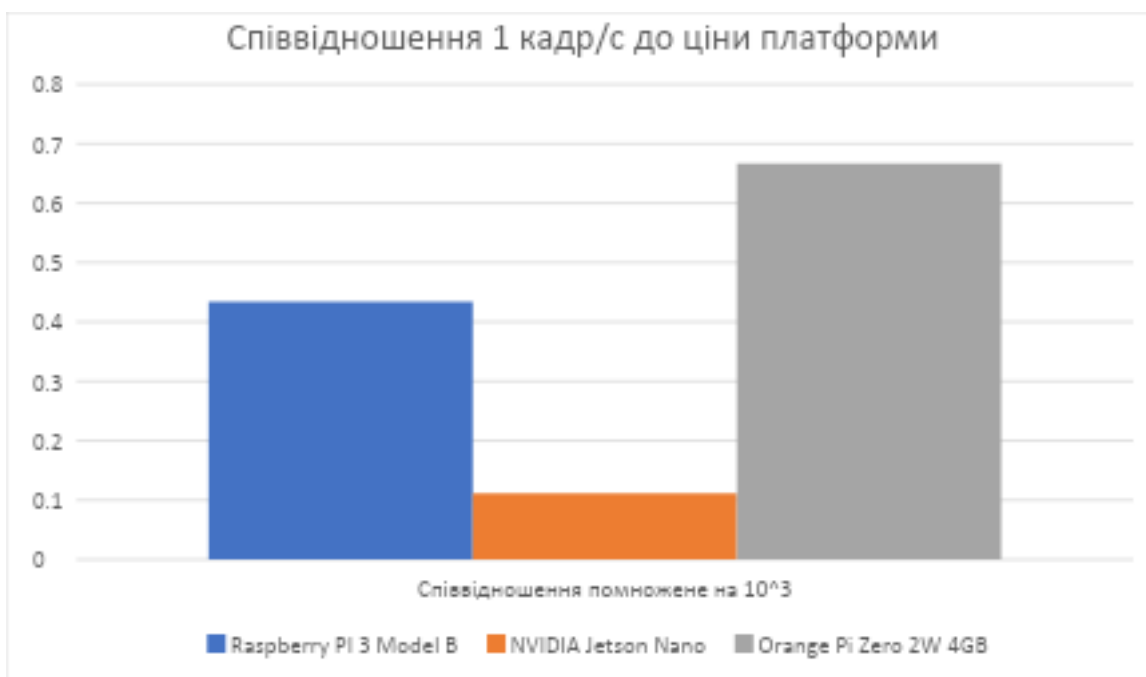
Таблиця 3.2 – Результати точності

Апаратна платформа	Raspberry PI 3 Model B	NVIDIA Jetson Nano	Orange Pi Zero 2W 4GB
Відео			
Відео день(1080p) 1 знак	Розпізнано 1 знак	Розпізнано 1 знак	Розпізнано 1 знак
Відео ніч(2160p) 2 знаки	Розпізнано 1 знак	Розпізнано 1 знак	Розпізнано 1 знак
Відео день, місто(720p) 1 знак	Розпізнано 1 знак частково(1 із 8 символів)	Розпізнано 1 знак частково(7 із 8 символів)	Розпізнано 1 знак частково(1 із 8 символів)
Відео дощ(720p) 5 знаків	Розпізнано 1 знак та 4 частково(4 із 7(1), 5 із 7(1), 6 із 7(2) символів)	Розпізнано 1 знак та 4 частково(6 із 7 символів)	Розпізнано 1 знак та 4 частково(6 із 7 символів)

Також, розраховано співвідношення швидкості обробки 1 кадр/с до ціни апаратної платформи:

Таблиця 3.3 – Відношення швидкості обробки 1 кадр/с до ціни платформи

Апаратна платформа	Raspberry PI 3 Model B	NVIDIA Jetson Nano	Orange Pi Zero 2W 4GB
Ціна	2300грн.	9000грн.	1500грн.
Співвідношення	$0,434 \cdot 10^{-3}$	$0,111 \cdot 10^{-3}$	$0,667 \cdot 10^{-3}$



Отримані в ході експериментів результати переважно задовольняють мінімальні вимоги обраних критеріїв оцінки. Тож можна стверджувати, що всі платформи успішно впорались із задачею розпізнавання автомобільних номерів за допомогою „класичних” методів комп’ютерного зору.

3.2 Формування рекомендацій

На основі отриманих даних можна зробити наступні спостереження:

- Найвищі показники швидкості обробки в усіх експериментах показувала Jetson Nano. Наступною, за показниками, є Orange Pi. Й останньою є Raspberry Pi.
- Найкращі показники точності в середньому мали Jetson Nano та Orange Pi. Raspberry Pi показала нижчі результати точності в Відео день, місто(720p) та Відео дощ(720p).
- Найвище співвідношення швидкості обробки 1 кадр/с до ціни має Orange Pi. Нижчі показники має Raspberry Pi. Й найнижчі результати має Jetson Nano.

Тож, можна зробити висновок, що для вирішення задачі розпізнання автомобільних номерів за допомогою „класичних” методів комп’ютерного зору з використанням мови програмування Python та бібліотек OpenALPR, Tesseract, OpenCV, ефективним та достатнім варіантом є Orange Pi марки Zero 2W з 4 Гб оперативної пам’яті, 4 ядрами тактової частоти 1,5 ГГц, який є найдешевшим з розглянутих апаратних платформ. Ця апаратна платформа показала хороші результати швидкодії й точності, у порівняння з іншими платформами, що у комбінації з низькою ціною робить її найкращим вибором.

Jetson Nano, хоч і продемонструвала найвищі результати швидкодії та високі результати точності з поміж розглянутих платформ, але є досить дорогою. Це робить її застосування з такими методами та засобами економічно не вигідним. Але, у випадку незначущості економічної доцільності, ця платформа буде кращим вибором за рахунок вищої результативності.

Raspberry Pi показала найнижчі результати швидкодії та точності, з поміж обраних платформ, хоча й має непогане співвідношення швидкодії до ціни. Її застосування у рамках розглянутої задачі є недоцільним.

ВИСНОВКИ

В цій роботі було проведено дослідження, спрямоване на визначення ефективності застосування апаратних платформ на прикладі задачі розпізнавання автомобільних номерів. В результаті, було обрано методи комп'ютерного зору, апаратні платформи, написано програму для розпізнавання автомобільних номерів, проведено експерименти та отримано результати. На основі аналізу отриманих результатів визначено найефективнішу платформу серед обраних та сформовано рекомендації щодо вибору апаратної платформи.

В ході дослідження було розглянуто етапи та сучасні методи, які застосовуються для ALPR. В результаті, визначено два основних методи, які застосовуються в сучасних системах ALPR: класичні методи комп'ютерного зору та глибоке навчання. Розгляд визначених методів дозволив виділити основні переваги та недоліки. Глибоке навчання демонструє кращі результати, вимагаючи більшої кількості ресурсів, в той час як „класичні” методи демонструють нижчі результати у порівнянні, але мають менші вимоги до потужності апаратної платформи. У ході дослідження цієї теми було знайдено переважно дослідження ALPR з використанням глибокого навчання, тому для цього дослідження було обрано використання „класичних” методів.

Було обрано та проаналізовано апаратні платформи, які сумісні з обраними методами та є недорогими. На основі розгляду платформ, сформовано таблицю, в якій проведено порівняння основних технічних характеристик, які впливають на швидкість обробки даних в задачах комп'ютерного зору. За результатами порівняння, обрано три найперспективніші платформи – Raspberry Pi, Jetson Nano та Orange Pi.

Для створення програми ALPR, обрано та обґрунтовано вибір відповідних програмних засобів, які є сумісними з обраними платформами та

„класичними” методами: мова програмування Python, бібліотеки OpenALPR, Tesseract та OpenCV. Визначено критерії оцінки отриманих в ході експериментів результатів. За допомогою обраних програмних засобів, написано програму для розпізнання автомобільних номерів, яка використовуватиметься в подальших експериментах.

Для проведення експериментів, було підібрано набір відеофайлів з різним розширенням та різними умовами зйомки даних відео; визначено критерій ефективності. Проведено серію експериментів на обраних апаратних платформах та отримано результати, які свідчать про те, що, згідно порівнянням ефективності найкращою платформою для задачі розпізнавання автомобільних номерів виявилася Orange Pi. Ця платформа поєднує високі показники точності та швидкодії з низькою ціною, з поміж розглянутих платформ.

За результатами експериментів, сформовано рекомендації, щодо вибору апаратної платформи для задачі розпізнавання автомобільних номерів.

Результати даного дослідження корисні, для вибору найефективнішої апаратної платформи в бюджетних системах ALPR, які можуть створюватись з метою навчання або використання для простих задач(Паркінг, пропускні системи), які не потребують високої швидкості обробки. Також, вони демонструють актуальність та ефективність „класичних” методів комп’ютерного зору у поєднанні з бюджетними апаратними платформами для ALPR.

В перспективі, дане дослідження можна розширити, за рахунок дослідження інших апаратних платформ для формування списку з ефективністю застосування різних апаратних платформ у комбінації з „класичними” методами для задачі розпізнавання автомобільних номерів. Це дозволить в подальшому визначити найефективнішу та/або найдешевшу платформу для реалізації простих систем ALPR.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jeffery J. H. W., Yu P. X. Number Plate Recognition System on an ARM-DSP and FPGA. *Hot Chips 25 Symposium*. 2013. URL: https://old.hotchips.org/wp-content/uploads/hc_archives/hc25/HC25-posters/HC25.26.p10-Number%20Plate%20Recognition%20System%20on%20an%20ARM-DSP%20and%20FPGA-Jeffery.pdf (дата звернення: 20.10.2024).
2. Alabdan A., Alawad H., Aldakhil H., Alqahtani A., Alshammari F. Latency and Throughput Benchmarking for Portable Automatic Number Plate Recognition Systems in Simulated Saudi Traffic Scenes. *ResearchGate*. Nov. 2024. URL: https://www.researchgate.net/publication/396741209_Latency_and_Throughput_Benchmarking_for_Portable_Automatic_Number_Plate_Recognition_Systems_in_Simulated_Saudi_Traffic_Scenes (дата звернення: 20.10.2024).
3. Ahmed S., Alshehri F., Alzahrani K., Alharbi S., Alanazi H. Robust Plate Localization under Motion Blur Using Hardware-Accelerated Image Pipelines. *ResearchGate*. Nov. 2024. URL: https://www.researchgate.net/publication/396741655_Robust_Plate_Localization_under_Motion_Blur_Using_Hardware-Accelerated_Image_Pipelines (дата звернення: 20.10.2024).
4. IPC-HFW5541E-ZE 5MP IR Vari-focal Bullet WizMind Network Camera. *Dahua Technology*. URL: <https://www.dahuasecurity.com/products/All-Products/Discontinued-Products/Network-Cameras/IPC-HFW5541E-ZE> (дата звернення: 13.11.2025).
5. 8MP WDR Fixed Bullet Network Camera IPC2128SB-ADF28KM-I0. *Uniview*. URL: https://www.uniview.com/Products/Network_Cameras/Prime_Series/PRIMEI_Series/IPC2128SB-ADF28KM-I0/ (дата звернення: 13.11.2025).
6. Siddiqui Y. A., Aftab M. A., Ullah M. F. A., Yaseen M. A., Javed A. A. Enhanced License Plate Recognition Using Deep Learning for Smart City Applications. *IEEE Access*. 2024. URL: https://scholar.google.com/citations?view_op=view_citation&hl=uk&user=TDk_NfkAAAAJ&citation_for_view=TDk_NfkAAAAJ:u-x6o8ySG0sC (дата звернення: 05.12.2024).
7. Tahir M. N., Tahir A. R., Muhammad F., Abbas S., Shah M. H. Robust license plate localization and recognition system using You Only Look Once (YOLO). *Review of International Economics*. Vol. 11, Iss. 2. 2021. P. 138–149. URL: https://www.riejournal.com/article_106905_afd0caf26202eb3ac3b605fd17894255.pdf (дата звернення: 05.12.2024).

8. NVIDIA TensorRT. *NVIDIA Developer*. URL: <https://developer.nvidia.com/tensorrt> (дата звернення: 10.10.2025).
9. What is ONNX and why is it used? *Milvus.io: AI Quick Reference*. URL: <https://milvus.io/ai-quick-reference/what-is-onnx-and-why-is-it-used> (дата звернення: 05.12.2024).
10. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. *arXiv*. 2015. URL: <https://arxiv.org/pdf/1507.05717> (дата звернення: 15.12.2024).
11. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv*. 2015. URL: <https://arxiv.org/pdf/1411.4555> (дата звернення: 15.12.2024).
12. Canny Algorithm. *Scribd*. URL: <https://www.scribd.com/doc/48641993/canny-algorithm> (дата звернення: 05.12.2024).
13. Vincent D. An Analysis of Canny Edge Detection. *Proceedings of Informing Science & IT Education Conference (InSITE) 2009*. URL: <https://proceedings.informingscience.org/InSITE2009/InSITE09p097-107Vincent613.pdf> (дата звернення: 05.12.2024).
14. Sahoo S. K. A Complete Guide on Hough Transform. *Analytics Vidhya*. June 20, 2022. URL: <https://www.analyticsvidhya.com/blog/2022/06/a-complete-guide-on-hough-transform/> (дата звернення: 05.12.2024).
15. Canny J. F. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. PAMI-8, No. 6. Nov. 1986. P. 679-698. URL: <https://perso.telecom-paristech.fr/elc/papers/pami96.pdf> (дата звернення: 05.12.2024).
16. Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005. P. 886-893. URL: <https://static.googleusercontent.com/media/research.google.com/uk//pubs/archive/33418.pdf> (дата звернення: 15.12.2024).
17. Tariq U., Farrukh W., Riaz H., Riaz S., Rehman A. A Comprehensive Literature Review of Vehicle License Plate Detection Methods. *ResearchGate*. June 2024. URL: https://www.researchgate.net/publication/381732317_A_Comprehensive_Literature_Review_of_Vehicle_License_Plate_Detection_Methods (дата звернення: 15.12.2024).
18. Khandekar G. G., Kulkarni V. C., Kulkarni R. Y. License Plate Recognition System for Indian Vehicles Using Deep Learning. *International Conference on*

- Computer Communication and Informatics (ICCCI)*. 2021. P. 1–5. URL: <https://ieeexplore.ieee.org/document/9310202> (дата звернення: 15.12.2024).
19. EfficientNet: Smarter, Not Just Bigger Neural Networks. *Medium: Towards Data Science*. 2019. URL: <https://medium.com/@kdk199604/efficientnet-smarter-not-just-bigger-neural-networks-94db3e2f8699> (дата звернення: 15.12.2024).
 20. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv*. 2017. URL: <https://arxiv.org/pdf/1704.04861> (дата звернення: 15.12.2024).
 21. Residual Neural Network. *ScienceDirect Topics*. URL: <https://www.sciencedirect.com/topics/computer-science/residual-neural-network> (дата звернення: 15.12.2024).
 22. Jamil M. Z., Ullah S., Asif A., Nawaz A. A Deep Learning Framework for Accurate Number Plate Recognition using OWL-V2 and PaddleOCR. *ResearchGate*. April 2024. URL: https://www.researchgate.net/publication/390878681_A_Deep_Learning_Framework_for_Accurate_Number_Plate_Recognition_using_OWL-V2_and_PaddleOCR (дата звернення: 20.10.2024).
 23. Srinivasan S., B. S. A. G., HCL. Image Processing vs Deep Learning for Computer Vision. *White Paper*. URL: https://www.imveurope.com/sites/default/files/content/white-paper/pdfs/HCL_IMVE_WP-ImageProcessing_vs_DL.pdf (дата звернення: 20.10.2024).
 24. Niall O' Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Traditional Image Processing vs. Deep Learning. *arXiv*. August 2020. URL: <https://arxiv.org/pdf/1910.13796> (дата звернення: 20.10.2024).
 25. Камери для розпізнавання номерів автомобілів. *Kobi*. URL: <https://kobi.ua/solutions/rozpiznavannya-nomeriv-cameras/> (дата звернення: 25.11.2024).
 26. LPR BOX “Orthus” / “Hydra”. *Kobi*. URL: <https://kobi.ua/solutions/reshenie-6/> (дата звернення: 25.11.2024).
 27. Розпізнавання номерів (LPR). *Kobi*. URL: <https://kobi.ua/technological-solutions/plate-recognition/> (дата звернення: 25.11.2024).
 28. Kobi LPR 2.0. *Kobi*. URL: <https://kobi.ua/product/1226/> (дата звернення: 25.11.2024).
 29. Edge Max. *Rekor*. URL: <https://www.rekor.ai/systems/edge-max#specifications> (дата звернення: 25.11.2024).

30. What is CarCheck? *Rekor Support Center*. URL: <https://help.rekor.ai/what-is-carcheck> (дата звернення: 25.11.2024).
31. Мікрокомп'ютер Raspberry Pi 5 Board 4GB. *EVO*. URL: <https://evo.net.ua/mikrokompiuter-raspberry-pi-5-board-4gb/> (дата звернення: 24.12.2024).
32. NVIDIA Jetson Nano Developer Kit. *EVO*. URL: <https://evo.net.ua/nvidia-jetson-nano-developer-kit/> (дата звернення: 24.12.2024).
33. Orange Pi Zero 2W. *Orange Pi*. URL: <http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-Zero-2W.html> (дата звернення: 24.12.2024).
34. Shahin M., Zhang X., Zhu H., S. W. M. Enhanced YOLOv5-Based ANPR System for Complex Urban Environments. *Sensors*. Vol. 22, Iss. 14. 2022. P. 5283. URL: <https://www.mdpi.com/1424-8220/22/14/5283> (дата звернення: 20.10.2024).
35. About Python. *The Python Software Foundation*. URL: <https://www.python.org/about/> (дата звернення: 10.10.2025).
36. OpenALPR. *GitHub: openalpr*. URL: <https://github.com/openalpr/openalpr> (дата звернення: 10.10.2025).
37. Introduction to OpenCV. *OpenCV Documentation*. URL: <https://docs.opencv.org/4.x/d1/dfb/intro.html> (дата звернення: 10.10.2025).
38. Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005. P. 886-893. URL: <https://static.googleusercontent.com/media/research.google.com/uk//pubs/archive/33418.pdf> (дата звернення: 20.10.2024).
39. Tesseract Technical Documentation. *GitHub: tesseract-ocr/tessdoc*. URL: <https://github.com/tesseract-ocr/tessdoc/blob/main/tess3/Technical-Documentation.md> (дата звернення: 10.10.2025).
40. Abramov, V., Astafieva, M., Boiko, M., Bodnenko, D., Bushma, A., Vember, V., Hlushak, O., Zhyltsov, O., Ilich, L., Kobets, N., Kovaliuk, T., Kuchakovska, H., Lytvyn, O., Lytvyn, P., Mashkina, I., Morze, N., Nosenko, T., Proshkin, V., Radchenko, S., ... Yaskevych, V. (2021). *Theoretical and practical aspects of the use of mathematical methods and information technology in education and science*. URL: <https://doi.org/10.28925/9720213284km> (дата звернення: 11.10.2025).
41. Андрій Петрович Бондарчук, Ірина Юріївна Мельник, Євгеній Іванович Суханевич, Вадим Олексійович Абрамов. Підвищення ефективності систем відеоспостереження за рахунок гібридного методу відбору ключових кадрів і інтерпретації рішень Телекомунікаційні та інформаційні технології 2025 , №3

- c101-105 URL:<https://doi.org/10.31673/2412-4338.2025.038710> (дата звернення: 11.11.2025).
42. Яковенко В. А. Визначення ефективності застосування апаратних платформ на прикладі задачі розпізнавання автомобільних номерів. *Комп'ютерні науки та інформаційні технології*. 2025. Вип. 12. С. 217–218.
URL: <https://zcit.kubg.edu.ua/index.php/journal/issue/view/13/23> (дата звернення: 17.02.2025).
43. Яковенко В. А. Визначення ефективності застосування апаратних платформ на прикладі задачі розпізнавання автомобільних номерів. *Студентський науковий пошук* – 2025, Том 9, № 9 URL: <https://zcit.kubg.edu.ua/index.php/journal/article/view/35> (дата звернення: 22.10.2025).
44. Бушма, Олександр Володимирович та Машкіна, Ірина Вікторівна та Носенко, Тетяна Іванівна та Яскевич, Владислав Олександрович (2024) *Кваліфікаційна робота магістра: Навчально-методичний посібник для спеціальності «Комп'ютерні науки»* Київський столичний університет імені Бориса Грінченка, Україна. URL:<https://elibrary.kubg.edu.ua/id/eprint/50205/> (дата звернення: 10.10.2024).

Додаток А

Код програми

```
import cv2
import time
from openalpr import Alpr

# Ініціалізація ALPR
COUNTRY = "eu" # Країна
CONFIG_PATH = "/etc/openalpr/openalpr.conf"
RUNTIME_DATA_PATH = "/usr/share/openalpr/runtime_data"

alpr = Alpr(COUNTRY, CONFIG_PATH, RUNTIME_DATA_PATH)
if not alpr.is_loaded():
    print("Не вдалося завантажити OpenALPR")
    exit(1)

# Відкриття камери
cap = cv2.VideoCapture("/usr/share/samples/us-vid3.mp4")
# cap = cv2.VideoCapture(0) # При використанні камери

if not cap.isOpened():
    print("Камеру не вдалося відкрити")
    exit(1)

# Підрахунок FPS
```

```

frame_count = 0

start_time = time.time()

print("Розпізнавання запущено. Натисни 'є' для виходу.")

while True:

    ret, frame = cap.read()

    if not ret:

        print("Неможливо зчитати кадр")

        break

    results = alpr.recognize_ndarray(frame)

    # Вивід розпізнаних номерів

    for plate in results['results']:

        best_candidate = plate['candidates'][0]

        print(f"Номер: {best_candidate['plate']} (Ймовірність: {best_candidate['confidence']:.2f}%)")

        frame_count += 1

    # Розрахунок FPS кожні 60 кадрів(Опціонально)

    if frame_count % 60 == 0:

        passed = time.time() - start_time

        fps = frame_count / passed

        print(f"Поточна швидкість обробки: {fps:.2f} кадр/с")

```

```
cv2.imshow("Frame", frame)# Демонстрація захопленого  
відео(Опціонально)
```

```
if cv2.waitKey(1) & 0xFF == ord('e'):
```

```
    break
```

```
# Завершення
```

```
passed = time.time() - start_time
```

```
avg_fps = frame_count / passed
```

```
print(f"Середня швидкість: {avg_fps:.2f} кадр/с")
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

```
alpr.unload()
```

Додаток Б

Результат експерименту на Orange PI з Відео день, місто(1080p)

Розпізнавання запущено. Натисни 'q' для виходу.

Номер: 0JUI3 (Ймовірність: 78.91%)

Номер: JUNID (Ймовірність: 76.91%)

Номер: IIII (Ймовірність: 89.79%)

Номер: WIII (Ймовірність: 80.48%)

Поточна швидкість обробки: 0.79 кадр/с

Номер: IODII (Ймовірність: 80.99%)

Номер: LIUII (Ймовірність: 80.83%)

Номер: E11II (Ймовірність: 76.87%)

Номер: OK09B (Ймовірність: 83.57%)

Номер: GK09BJE (Ймовірність: 85.78%)

Поточна швидкість обробки: 0.79 кадр/с

Номер: J9ILI (Ймовірність: 79.40%)

Поточна швидкість обробки: 0.82 кадр/с

Поточна швидкість обробки: 0.83 кадр/с

Середня швидкість: 0.83 кадр/с

Номер: 0JUI3 (Ймовірність: 78.91%)

Номер: JUNID (Ймовірність: 76.91%)

Номер: IIII (Ймовірність: 89.79%)

Номер: WIII (Ймовірність: 80.48%)

Поточна швидкість обробки: 0.82 кадр/с

Номер: IODII (Ймовірність: 80.99%)

Номер: ЛІІІ (Ймовірність: 80.83%)

Номер: Е11ІІ (Ймовірність: 76.87%)

Номер: 0K09B (Ймовірність: 83.57%)

Номер: GK09BJE (Ймовірність: 85.78%)

Поточна швидкість обробки: 0.81 кадр/с

Номер: J9ІІІ (Ймовірність: 79.40%)

Поточна швидкість обробки: 0.83 кадр/с

Поточна швидкість обробки: 0.84 кадр/с

Середня швидкість: 0.84 кадр/с

Номер: 0JUI3 (Ймовірність: 78.91%)

Номер: JUHID (Ймовірність: 76.91%)

Номер: ІІІІ (Ймовірність: 89.79%)

Номер: WIII (Ймовірність: 80.48%)

Поточна швидкість обробки: 0.82 кадр/с

Номер: І0DII (Ймовірність: 80.99%)

Номер: ЛІІІІ (Ймовірність: 80.83%)

Номер: Е11ІІ (Ймовірність: 76.87%)

Номер: 0K09B (Ймовірність: 83.57%)

Номер: GK09BJE (Ймовірність: 85.78%)

Поточна швидкість обробки: 0.81 кадр/с

Номер: J9ІІІ (Ймовірність: 79.40%)

Поточна швидкість обробки: 0.83 кадр/с

Поточна швидкість обробки: 0.83 кадр/с

Середня швидкість: 0.83 кадр/с

Номер: 0JUI3 (Ймовірність: 78.91%)

Номер: JUHID (Ймовірність: 76.91%)

Номер: IIII (Ймовірність: 89.79%)

Номер: WIII (Ймовірність: 80.48%)

Поточна швидкість обробки: 0.82 кадр/с

Номер: I0DII (Ймовірність: 80.99%)

Номер: LIUII (Ймовірність: 80.83%)

Номер: E11II (Ймовірність: 76.87%)

Номер: 0K09B (Ймовірність: 83.57%)

Номер: GK09BJE (Ймовірність: 85.78%)

Поточна швидкість обробки: 0.82 кадр/с

Номер: J9ILI (Ймовірність: 79.40%)

Поточна швидкість обробки: 0.84 кадр/с

Поточна швидкість обробки: 0.84 кадр/с

Середня швидкість: 0.84 кадр/с

Номер: 0JUI3 (Ймовірність: 78.91%)

Номер: JUHID (Ймовірність: 76.91%)

Номер: IIII (Ймовірність: 89.79%)

Номер: WIII (Ймовірність: 80.48%)

Поточна швидкість обробки: 0.82 кадр/с

Номер: I0DII (Ймовірність: 80.99%)

Номер: LIUII (Ймовірність: 80.83%)

Номер: E11II (Ймовірність: 76.87%)

Номер: 0K09B (Ймовірність: 83.57%)

Номер: GK09BJE (Ймовірність: 85.78%)

Поточна швидкість обробки: 0.81 кадр/с

Номер: J9ILI (Ймовірність: 79.40%)

Поточна швидкість обробки: 0.83 кадр/с

Поточна швидкість обробки: 0.84 кадр/с

Середня швидкість: 0.84 кадр/с

Номер: 0JUI3 (Ймовірність: 78.91%)

Номер: JUNID (Ймовірність: 76.91%)

Номер: IIII (Ймовірність: 89.79%)

Номер: WIII (Ймовірність: 80.48%)

Поточна швидкість обробки: 0.82 кадр/с

Номер: I0DII (Ймовірність: 80.99%)

Номер: LIUII (Ймовірність: 80.83%)

Номер: E11II (Ймовірність: 76.87%)

Номер: 0K09B (Ймовірність: 83.57%)

Номер: GK09BJE (Ймовірність: 85.78%)

Поточна швидкість обробки: 0.81 кадр/с

Номер: J9ILI (Ймовірність: 79.40%)

Поточна швидкість обробки: 0.83 кадр/с

Поточна швидкість обробки: 0.84 кадр/с

Середня швидкість: 0.84 кадр/с