

ВСТУП

В даний час комп'ютеризація в нашому суспільстві розвивається дуже швидкими темпами і відіграє величезну роль у житті людини. За допомогою комп'ютерних технологій автоматизується широкий спектр процесів, які в недалекому минулому покладалися на людину. Інформаційні технології використовуються всюди: у промисловості, у транспорті, в побуті та ін. Програмісти усього світу розробляють нові й удосконалюють вже існуючі алгоритми автоматизації [1].

Рішення проблеми ідентифікації автомобіля за реєстраційним номерним знаком є важливим аспектом безпеки і контролю. Використовувати такий продукт можна в різних сферах застосування, що стосуються автотранспорту. Прикладом можуть служити автотранспортні підприємства, заправні станції, контроль швидкості руху, автомобільні стоянки, контроль в'їзду на територію підприємства і т.п.

В даний час існує не так багато систем визначення номерних знаків, не всі з яких є по-справжньому якісною продукцією [2].

Системи, що володіють високою швидкістю і точністю розпізнавання, як правило, дуже дорогі. Висока вартість існуючих продуктів не дозволяє здійснити їх масове впровадження.

Задачу ідентифікації автомобіля можна умовно розділити на дві підзадачі: локалізація номерного знака і розпізнавання символів. Ця робота присвячена реалізації розпізнавання номерних знаків з використанням Raspberry Pi, та реалізацією веб-інтерфейсу для подальшої обробки результатів розпізнавання. За останні десятиріччя веб-сайти перетворились з простих текстових сторінок у могутні сервіси, які можна використати для нашої задачі.

А використання Raspberry Pi надає зробити більш-менш реальний пристрій розпізнавання номерних знаків. Звісно, інструментом зйомки

					РБ101.17010201.072.007 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

виступає відеокамера. Принцип розпізнавання по відео не сильно відрізняється від розпізнавання по зображенню, так як відеозйомка — це потік зображень із певною частотою.

У загальному випадку розпізнавання реалізується в три етапи: попередня обробка зображення, сегментація та власне розпізнавання символів. Але в нашому випадку, треба додати ще один етап — кадрування зображення з відео.

					РБ101.17010201.072.007 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІТИЧНИЙ ОГЛЯД

Розпізнавання номерного знака (LPR) є формою автоматичної ідентифікації транспортного засобу. Ця технологія обробки зображення використовується, щоб ідентифікувати транспортні засоби лише за їх номерними знаками. У режимі реального часу LPR грає головну роль в автоматичному контролі правил дорожнього руху і підтримці правопорядку на громадських дорогах. Істотна перевага системи LPR полягає в тому, що система може вести відеозапис транспортного засобу, що корисно для того, щоб боротися з злочином і шахрайством. Ранні системи LPR страждали від низької швидкості розпізнавання, нижче, ніж необхідні практичні системи. Зовнішні ефекти (сонце і фари, погані номерні знаки, велике число типів номерних знаків), обмежений рівень програмного забезпечення розпізнавання і апаратні засоби бачення привели до низькопробних систем. Проте недавні поліпшення програмного і апаратного забезпечення зробили системи LPR більш надійними і широко поширеними.

Л.Н. Ясницький виділяє три основні класи методів розпізнавання символів [3]:

- шаблонний метод — заснований на порівнянні зображення символу з наявним в пам'яті шаблоном. Вимагає складних перетворень і обчислень для компенсації деформації символів;
- структурний метод — символ представляється графом, вузлами якого є деталі символу (прямі лінії, ділянки кіл), а дугами — просторові відносини між ними. Використовує методи теорії графів;
- ознаковий метод — заснований на виділенні деяких ознак зображення символу. Набір ознак представляє з себе n -мірний вектор, який порівнюється з еталонними векторами для визначення класу символу.

Якщо розглядати проблему глибше, то вона складається з двох частин: виділення характерних рис і класифікації. У шаблонному методі обидві

					РБ101.17010201.072.007 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

частини об'єднані в одну, але в інших вони чітко розділені. Для вирішення обох частин проблеми зазвичай застосовуються різні методи. Методи виділення характерних рис:

- моменти — використовує такі поняття як контур, центр мас, момент обертання. Використовуються геометричні моменти, моменти Церніке, моменти Лежандра, моменти Чебишева;
- гістограми — це графік статистичного розподілу елементів цифрового зображення з різною яскравістю, в якому по горизонтальній осі представлена яскравість, а по вертикалі — відносна кількість пікселів з конкретним значенням яскравості;
- спрямовані риси — використовує геометричні перетворення для виділення в зображенні штрихів, що мають схожі кути нахилу. Безліч штрихів стає безліччю характерних рис;
- перетворення Хафа;
- скелетизація — перетворення ліній зображення у векторний скелет. Довжини і відносні кути нахилу векторів є характерними рисами;
- дескриптори Фур'є — представлення контуру зображення в частотній області;
- апроксимація — аналог скелетизації, але замість векторів використовуються криві або контури;
- топологічні риси — виділення кінців ліній, перехресть, розвилок;
- лінійні перетворення — метод головних компонент, лінійний дискримінантний аналіз;
- ядра — методи перетворення складних нелінійних задач в низькорівневі лінійні задачі за допомогою ядер відображень.

Методи класифікації символів:

- статистичні — засновані на теорії ймовірності;
- нейромережеві — використовують нейромережі різної топології для класифікації набору виділених характерних рис;

					РБ101.17010201.072.007 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- метод опорних векторів — переведення вихідних векторів у простір більш високої розмірності і пошук розділяє гіперплощини з максимальним зазором в цьому просторі;
- структурні — засновані на порівнянні структури зображення з шаблоном;
- змішані — комбінація з декількох методів.

У даній дипломній роботі буде використовуватися такі бібліотеки, як OpenALPR, OpenCV, TesseractOCR та Leptonica. Після компіляції цих бібліотек на Raspberry Pi, напишемо маленький скрипт для перевірки роботи. А далі реалізуємо веб-інтерфейс з результатами. В наступних розділах будуть описані кожна з бібліотек, робота з ними, технічні характеристики Raspberry Pi та відеокамери.

1.1 Структура ALPR

На сьогоднішній день у світі вже існує багато систем автоматичного розпізнавання автомобільних номерних знаків. Всі вони далекі від досконалості і постійно модифікуються. Проте, загальна структура даної задачі вже сформована (рисунок 1.1).

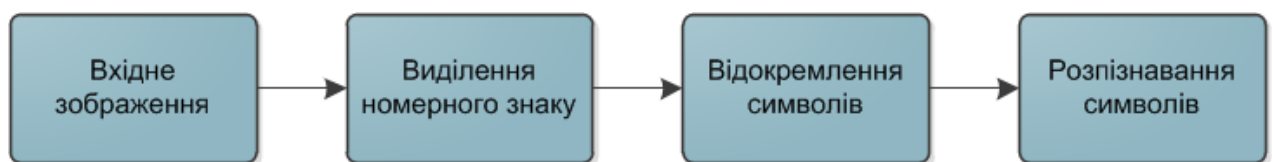


Рисунок 1.1 — Загальна структура системи розпізнавання номерних знаків

Алгоритми виділення номерного знака, відділення символів і розпізнавання символів розглядатимуться в наступних розділах більш детально. Виходом системи розпізнавання номерних знаків є сукупність символів — розпізнаний номерний знак.

					РБ101.17010201.072.007 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Актуальність, предмет та об'єкт роботи

Нині системи розпізнавання та обробки зображень отримують повсюдне поширення, так як з кожним днем збільшується потреба автоматизації контролю в'їзду на територію підприємств, контролю швидкості руху і визначення порушення правил дорожнього руху. У той час, існує декілька бібліотек, які дозволяють реалізувати таку функціональність. Серед них немає явного лідера, і для різних спеціальних задач, створюється безліч їх модифікацій. В даний час існує досить велика кількість систем визначення номерних знаків, але не всі з них є якісною і надійною продукцією. Системи з високою швидкістю і точністю розпізнавання є комерційними, засекреченими і дорогими, що не дозволяє здійснити їх масове впровадження.

Ця робота є однією з таких модифікацій. Для віддаленого управління та моніторингу, використовується веб-інтерфейс, заснований на сучасному фреймворку.

Предметом дослідження виступає процес використання готових бібліотек на Raspberry Pi для ALPR (Automated License Plate Recognition).

Об'єкт дослідження — автоматична система ALPR на основі бібліотеки OpenALPR.

1.3 Мета та задачі роботи

Метою роботи є використання готових рішень у вигляді бібліотек, дослідження існуючих методів, алгоритмів і програм розпізнавання знаків, символів, цифр і букв, які забезпечують аналіз та обробку інформації на зображенні з метою виявлення, локалізації та ідентифікації номерного знака автомобілів.

Досягнення поставленої мети вимагає вирішення наступних завдань:

- аналіз бібліотек для ALPR;

					РБ101.17010201.072.007 ПЗ	Арк.
						1 1
Змн.	Арк.	№ докум.	Підпис	Дата		

- компіляція бібліотек з #1 на Raspberry Pi;
- дослідження методу розпізнавання образів бібліотеки OpenALPR;
- розробка веб-інтерфейсу для віддаленого управління та моніторингу, заснований на Twitter Bootstrap;
- оцінка ефективності та порівняння часу обробки для зображень а відео;
- динамічно оновлювати результати розпізнавання.

Завдання розпізнавання номерних знаків розбивається на три етапи: попередня обробка зображення, сегментація, розпізнавання символів.

Попередня обробка зображення необхідна для поліпшення візуальної якості зображення і, в кінцевому підсумку, для локалізації номерного знака.

Сегментація проводиться з метою виділення символів для подальшого розпізнавання обраним методом.

У рамках роботи спеціаліста планується налагодити відіокамеру, Raspberry Pi, а також реалізувати веб-інтерфейс, який буде в режимі Real Time оновлювати інформацію щодо розпізнаних номерних знаків транспортних засобів.

1.4 Плановані практичні результати

Після закінчення роботи планується отримати працездатний програмний продукт, призначений для розпізнавання номерних знаків, символів, запису в текстовому вигляді на сайті.

1.5 Сфери застосування

Основні сфери застосування систем розпізнавання номерних знаків представлені нижче [3]:

- паркування;

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2
Змн.	Арк.	№ докум.	Підпис	Дата		

- стоянки;
- торгові центри;
- залізничні вокзали;
- станції ДПС;
- системи управління та обробки інформації;
- охоронні підприємства;
- банківські служби;
- аеропорти і т.д.

1.6 Огляд досліджень і розробок по даній темі

У світі вже існують розробки, що стосуються подібних програмних систем. Найбільш успішні з них, описані нижче.

1.6.1 Система «Авто-Инспектор»

Це програмно-апаратний комплекс, що забезпечує розпізнавання номерів рухомих автомобілів, надійно працює в широкому діапазоні зовнішніх умов, легко інтегрований з охоронним обладнанням, виконавчими пристроями і зовнішніми базами даних [3].

Ефективний для вирішення завдань реєстрації, ідентифікації та забезпечення безпеки автомобілів, контролю транспортних потоків.

Система дозволяє:

- здійснювати відеоспостереження за зоною зважування, записувати відеозображення, фіксуючи проходження автомобілів;
- автоматично детектувати проходження автомобіля. Детектування проводиться апаратними засобами (промінь), що гарантує стовідсоткову достовірність;

					РБ101.17010201.072.007 ПЗ	Арк.
						1 3
Змн.	Арк.	№ докум.	Підпис	Дата		

- автоматично розпізнавати номери автомобілів по відеозображенню. Розпізнавання ведеться в реальному часі, результати розпізнавання негайно відображаються на моніторі оператора. Для підвищення достовірності результату і у випадку використання автопричепів розпізнавання ведеться за двома телекамерами (спереду й ззаду);
- здійснювати зважування автомобілів;
- виконувати прив'язку маси автомобіля до його номера;
- автоматично оцінювати рівень наливання рідини в автоцистернах (за наявності тепловізора);
- записувати результати розпізнавання і зважування в спеціалізовану базу даних — протокол;
- виставляти інформацію в протоколі з натурним листом або іншим джерелом інформації і виводити відповідну відмітку в разі виявлення невідповідностей;
- здійснювати оперативний пошук автомобілів по протоколу з можливістю відтворення відповідного відеофрагменту;
- формувати звіти за даними з протоколу, в тому числі зведені, з урахуванням зіставлення інформації з протоколу з натурним листом.

На національному рівні розробляються зчитувачі автомобільних номерів Київською фірмою «Аллан». Вони забезпечують висококонтрастні зображення автомобільних номерів в межах всього спектра умов навколишнього освітлення, від повної темряви до сліпучого сонячного світла і світла автомобільних фар. Зчитувачі оптимальні для роботи з програмним забезпеченням для розпізнання автомобільних номерів.

1.6.2 Система розпізнавання номерних знаків CARMEN

CARMEN — система розпізнавання номерних знаків, яка володіє всіма можливостями для забезпечення високоякісної парковки, охорони і контролю доступу по номерному знаку, розпізнаному системою. Характеризується як

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4
Змн.	Арк.	№ докум.	Підпис	Дата		

дуже точна, відмовостійка, стабільна і швидка система. Система CARMEN спеціально розроблена для паркувальних систем, систем безпеки та систем контролю доступу. Вона має унікальний рівень інтеграції розпізнавання знака, технології отримання відео та обробки зображень з відео потоків, а в той же час вона як і раніше пропонує користувачеві найвищу технологічну гнучкість.

Також вона забезпечує високий рівень технологічної продуктивності серед інших систем розпізнавання, система CARMEN пропонує величезну гнучкість — схвалена виробниками та інтеграторами.

Система CARMEN поставляється з якісним драйвером для захоплення зображень і потокового відео, забезпечуючи швидку установку і використання системи для паркування транспортних засобів та систем контролю доступу.

Також до складу системи CARMEN входить Software Development Kit (SDK) — набір модулів, який спрощує процес інтеграції. Легко, швидко і економічно ефективно. Паркувальні системи, системи контролю доступу, системи безпеки працюють під контролем системи CARMEN забезпечують високу точність розпізнавання, відмовостійкості, високу швидкість роботи.

Особливості системи CARMEN. Дана система пропонує гнучкі, низькі експлуатаційні витрати, представляючи ефективні та інтелектуальні технології для експлуатації парку автомобілів. Система зменшує витрати, а також підвищує ефективність і безпеку. Система для контролю паркування є однією з основних технологій, і не є повноцінною. Вона була спеціально розроблена і створена, щоб легко бути інтегрованою в комплекс парковки і контролю доступу. Така гнучка система може бути адаптована для задоволення особливих вимог замовника.

Система розпізнавання номерних знаків CARMEN містить наступні компоненти:

- движок / ядро системи автоматичного розпізнавання номерних знаків;
- FXVD4mc_s карта захоплення;

					РБ101.17010201.072.007 ПЗ	Арк.
						1 5
Змн.	Арк.	№ докум.	Підпис	Дата		

- функції бібліотеки (ANSI C/C++, MSVC /C + +, Delphi і MSVB, ActiveX, OCX);
- демо- і тест-додатки;
- підручник і приклади програм, як у виконуваному вигляді, так і у вигляді вихідного коду.

1.6.3 Камера «Spike +» з системою розпізнавання номерних знаків

«Spike+» являє собою компактну, міцну і повністю інтегровану систему розпізнавання номерного знака в камері, також виробляє підсвічування і процесор для системи розпізнавання в одному герметичному корпусі. «Spike+» є унікальним продуктом, і виграв престижну нагороду Queens Award for Innovation. Різноманітність об'єктів доступних для різних фокусних вимог довжини. Способи зберігання Compact Flash дозволяє зберігати і буферизувати дані. «Spike +» може використовуватися через веб-протокол, і пропонує різноманітні можливості підключення, включаючи бездротові мережі. Додаткові продукти, такі як Spike + Interface Box дозволяють швидко і легко розгортати Spike +.

Камера «Spike +» проста в установці, що робить її ідеальним для установки у важкодоступних або прихованих умовах. Незважаючи на свій невеликий розмір, «Spike +» дуже надійний і призначений для забезпечення безперебійної роботи в будь-яких погодних умовах 24 години на добу.

Програми для камери «Spike +». Talon Mobile — додаток розпізнавання автомобільних номерних знаків, що володіє функціями SpeedPlate і VeriPlate, які призначені для використання «на ходу» патрулюванням поліції, надаючи начальству і користувачам гнучкість і оперативну «досяжність» для визначення інформації про область, які виходять за рамки основних систем розпізнавання автомобільних номерів. Крім того, система може працювати, коли автомобіль нерухомий або припаркований на шосе, на блокпостах і контрольно-пропускних пунктах.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 6
Змн.	Арк.	№ докум.	Підпис	Дата		

Типові мобільні додатки для автоматичного розпізнавання номерних знаків складається з системи і движка ядра Talon, використовуваних для процесу визнання номерного знака. Користувач контролює і управляє системою за допомогою функції SpeedPlate або Veriplate систем прикладного програмного забезпечення розпізнавання автомобільних номерів, що дозволяє використовувати дані, як місцевих, так і національних баз даних.

Додаткові функції:

- точність читання більше 97%;
- підключення до VISCE Back Office Facility;
- запис дати і часу.

Області застосування:

- поліцейський патруль автомобілів;
- судові пристави і транспортні засоби цивільних агентств;
- установка на мобільний дата-термінал.

1.6.4 CarDetector Software

Компанія Vigilant video досліджувала динаміку використання розпізнавання номерних знаків, починаючи від потреб кінцевого користувача до безпосередньо самого виробничого процесу. Результати надали запропонувати продукт, який може вважатися найбільш просунутим за сьогоденніми мірками. Система CarDetector дозволяє користувачеві будь-якого рівня управляти можливостями програмного пропозиції. У комплекті системи йде короткий курс навчання.

Можливості:

- інфрачервоне і кольорове відображення номерів;
- запис пропущених знаків;
- інтеграція з різними камерами відеоспостереження;
- централізована система управління системою;

					РБ101.17010201.072.007 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- можливість установки серверної і клієнтської частини.

Підтримка до 35 камер відеоспостереження для розпізнавання знака, підключених до одного сервера. Сучасна технологія оптичного розпізнавання символів, яка управляється системою через бездротові мережі. Система CarDetector дає максимально активні можливості, які забезпечують захист необхідних територій.

1.6.5 Система «ALPR»

Це програмно-апаратний комплекс, що забезпечує розпізнавання номерів рухомих автомобілів, надійно працює в широкому діапазоні зовнішніх умов, легко інтегрований з охоронним обладнанням, виконавчими пристроями і зовнішніми базами даних.

У системі «ALPR» реалізовані функціональні можливості необхідні для ефективного вирішення завдань на різних об'єктах: від забезпечення збереження автомобілів в межах автостоянки, до контролю за пересуванням транспорту в масштабах підприємства, окремої магістралі, цілого міста.

Основні функціональні можливості:

- розпізнавання реєстраційного номера автомобіля виявлення появи автомобіля в кадрі;
- розпізнавання всіх видів українських номерів;
- можливість адаптації до стандартів номерів будь-якої країни світу;
- успішна робота модуля в будь-яких погодних умовах;
- вибір кадру з оптимальним розміром і чіткістю реєстраційного номера;
- одночасне розпізнавання в зоні контролю декількох номерів (переклад відеозображення в текстовій формат);
- ефективна взаємодія з різними охоронними системами (охоронного телебачення, контролю доступу);
- запис кожного проїзду, в'їзду / виїзду транспортних засобів;

					РБ101.17010201.072.007 ПЗ	Арк.
						1 8
Змн.	Арк.	№ докум.	Підпис	Дата		

- створення баз даних.

1.6.6 Система «Трафік-КОНТРОЛЬ»

Це система зважування автомобілів і розпізнавання номерів. Програмне забезпечення «Трафік-КОНТРОЛЬ» призначено для автоматизованого контролю транспортного потоку і володіє наступними функціональними можливостями:

- одночасне зважування і розпізнавання номера автомобіля;
- розпізнавання державного номера транспортного засобу;
- пошук відповідності між визначеним державним номером і наявними номерами в БД;
- визначення швидкості руху транспортних засобів;
- централізована реєстрація та обробка подій, генерація оповіщень і керуючих впливів згідно з гнучко налаштованим алгоритмами;
- формування фото і відеоархіву.

1.6.7 АПК «Автоураган»

Це апаратно-програмний комплекс автоматичного розпізнавання зображень державних реєстраційних знаків транспортних засобів [4].

За відеозображенню, вступнику на комп'ютер, відбувається розпізнавання реєстраційних знаків автомобілів, перевірка їх по підключеним баз даних з видачею повідомлення оператору та збереженням інформації про проїзд або з виконанням іншого призначеного дії.

Гнучкі налаштування ПО дозволяють використовувати АПК «Автоураган» для вирішення численних завдань пов'язаних з розпізнаванням автомобільних номерів.

Переваги модульної побудови системи:

					РБ101.17010201.072.007 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

- модулі можуть працювати як на різних комп'ютерах, утворюючи розподілену мережеву клієнт-серверну систему, так і одночасно на одному комп'ютері, формуючи її локальний варіант;
- при необхідності завжди можна додати нові функції або кількість оброблюваних каналів розпізнавання.

1.6.8 Представники локального рівня

Полтава Сергій Олександрович, «Розпізнавання зображень» (джерело: газета «Комп'ютер-Інформ»). Розглядаються основні процедури і методи розпізнавання зображень, структура методів розпізнавання, особливості при проектуванні роботизованих систем. На відміну від моєї роботи, в системі не використовуються нейронні мережі, що підвищують швидкість розпізнавання [6].

Афанасенко Андрій Вікторович, «Розробка гібридної спеціалізованої системи розпізнавання образів на базі нечітких нейронних мереж». У роботі Андрія Вікторовича використовується нейромережа в задачі розпізнавання образів. Однак не розглядається архітектура нейромережі, що розпізнає зображення з заданим відсотком розпізнавання [7].

1.6.9 Представники глобального рівня

Lotufo, Морган і Джонсон представили автоматичну систему розпізнавання автомобільних номерів, використовуючи методи оптичного розпізнавання символів. S.K. Kim, D.W. Kim та H.J. Kim використали генетичний алгоритм для сегментації локалізації областей з номерним знаком. Hontana запропонував спосіб вилучення символів без знання їх положення і розмірів зображення.

					РБ101.17010201.072.007 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

1.7 Висновки

Даний розділ був присвячений загальній характеристиці роботи. У ньому розкривається актуальність і доцільність роботи шляхом аналізу та порівняння з відомими рішеннями наукової задачі. Сформульована мета роботи і завдання, які необхідно вирішити для досягнення поставленої мети. Містяться: обґрунтування необхідності нової розробки або вдосконалення (модернізації) існуючого об'єкта на основі аналізу сучасного стану проблеми за даними вітчизняної та зарубіжної науково-технічної літератури, патентного пошуку та досвіду роботи підприємств, установ, провідних фірм у відповідній галузі виробництва, економіки чи науки та обґрунтування основних прийнятих рішень або напрямків досліджень.

					РБ101.17010201.072.007 ПЗ	Арк.
						2 1
Змн.	Арк.	№ докум.	Підпис	Дата		

2 КОНФІГУРАЦІЯ RASPBERRY PI ТА ІНСТАЛЯЦІЯ OPEN-SOURCE БІБЛІОТЕК

2.1 Загальні відомості

Для початку опишемо Raspberry Pi B+, його технічні характеристики, різницю між Raspberry та Arduino, далі надамо описання таких open-source бібліотек, як OpenCV, OpenALPR, TesseractOCR, їх інсталяцію на ПО комп'ютера, а також перевіримо роботу за допомогою написання скрипта (попередньо без веб-інтерфейсу).

2.2 Raspberry Pi

Raspberry Pi — одноплатний комп'ютер, розроблений британським фондом Raspberry Pi Foundation. Його головне призначення — стимулювати навчання базовим комп'ютерним наукам у школах.

У жовтні 2013 розробники одноплатного комп'ютера Raspberry Pi оголосили про продаж більше двох мільйонів цих плат, що закріпило за Raspberry Pi звання найпопулярнішої платформи для ентузіастів. Примітно, що за прогнозами двохмільйонного рубежу планувалося досягти в кінці лютого 2014 року, але популярність платформи перевищила очікування. Причиною успіху проекту можна назвати низьку ціну, відкритий характер розробки та орієнтацію на навчання та експерименти.

2.2.1 Опис

Raspberry Pi побудований на системі-на-чипі (SoC) Broadcom BCM2835, яка включає в себе процесор ARM із тактовою частотою 700 МГц, графічний процесор VideoCore IV, і 512 чи 256 мегабайт оперативної пам'яті. Твердий диск відсутній, натомість використовується SD карта. Така апаратна начинка

					РБ101.17010201.072.007 ПЗ	Арк.
						2 2
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє відтворювати відео формату H.264 в роздільній здатності 1080p, і запускати комп'ютерні ігри на зразок Quake III Arena.

Комп'ютер замислювався як пристрій для навчання дітей програмуванню, однак здобув популярність і в інших сферах — зокрема, на його основі роблять домашні медіацентри. Наддешевий Raspberry Pi поставляється без корпусу і має вигляд плати розміром з кредитну карту. Плата важить 45 грамів. У комп'ютері задіяний 700-мегагерцевий процесор на архітектурі ARM; має роз'єм для навушників і слот для карт пам'яті. Молодша (A) і старша (B) моделі Raspberry Pi відрізняються об'ємом оперативної пам'яті (256 мегабайт проти 512 мегабайт) і кількістю USB-портів (один проти двох). Крім цього, у старшої моделі є роз'єм Ethernet 10/100, а молодша споживає на третину менше енергії.

Для Raspberry Pi випущені спеціалізований дистрибутив Linux, Raspbian OS (він заснований на дистрибутиві Debian) і магазин застосунків Pi Store, де є як платні, так і безплатні програми. Серед іншого Raspberry Pi використовує веб-браузер Iceweasel та KOffice, які поставляються в комплекті. Офіційна мова програмування для навчання — Python.

2.2.2 Розробка

Розробка пристроїв здійснюється Фондом Raspberry Pi Foundation, благодійною організацією зареєстрованою Комісією з благодійності. Його метою є «заохочення вивчення інформатики та суміжних наук, особливо на рівні школи, і знову зробити вивчення комп'ютерної науки цікавим». Прототип плати зробленої в 2006 році використовував Atmel ATmega644. Мікроконтролер Wong, George (2011-10-24). «Build your own prototype Raspberry Pi minicomputer». Плата працює на Linux ядрі.

					РБ101.17010201.072.007 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2.3 Характеристики

Надамо характеристик моделей –А та –В у таблиці 2.1.

Таблиця 2.1 — Характеристики двох моделей

	Model A	Model B
SoC	BCM2835	
CPU	700 МГц ARM	
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 H.264 high-profile decode	
Пам'ять (SDRAM)	256 MB	512 MB
USB 2.0 порти	1	2
Відео вихід	HDMI	
Аудіо вихід	3.5 mm jack, HDMI	
On-board storage	SD/MMC/SDIO memory card slot	
On-board network	None	10/100 wired Ethernet
Low-level peripherals	Up to 16 GPIO pins, SPI, I ² C, UART	
Real-time clock	None	
Властивості живлення	500 мА, (2.5 В)	700 мА, (3.5 В)
Джерело живлення	5В micro USB	
Програмне забезпечення	Debian GNU/Linux, Fedora, Arch Linux	
ПО, що підтримується	Other FLOSS software (Iceweasel, KOffice, Python), RISC OS (shared source)	

2.2.4 Подібності та відмінності В+ від В-моделі

Подібності:

- той же чіпсет Broadcom BCM2835;

					РБ101.17010201.072.007 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

- той же об'єм (512 Мб) оперативної пам'яті;
- той же повнорозмірний HDMI порт;
- той же порт 10/100 Ethernet;
- ті ж порти CSI для камери і DSI для дисплея;
- те ж живлення завдяки micro-USB.

Відмінності:

- тепер поставляється з 4 портами USB, що дозволяє підключити до Raspberry Pi ще більше пристроїв;
- інтерфейс GPIO збільшений до 40 пін, що дозволяє зробити більше проектів, ніж будь-коли раніше. Перші 26-контактів ідентичні моделі B, що забезпечує 100-відсоткову зворотну сумісність для проектів;
- тепер в пристрої використовується слот для карти пам'яті Micro-SD для зберігання інформації та завантаження ваших операційних систем;
- вдосконалена система:
- тепер ви можете забезпечити до 1,2 AMP на 4 USB порту — це дозволяє підключати більшу кількість пристроїв USB, вимогливих до живлення, без необхідності зовнішнього концентратора USB;
- модель B + більш енергоефективна (600 мА) в порівнянні з Моделлю B (750 мА);
- комбінований 4-полюсний роз'єм для підключення стерео, аудіо вихід і композитний відео-вихід.

2.2.5 Raspberry та Arduino

Їх часто сприймають як аналоги і навіть як конкуруючі апаратні платформи, які призначені для вирішення одних і тих же завдань. Насправді, Arduino і Raspberry Pi мають багато відмінностей.

Перше на що варто звернути увагу — це те, що Raspberry Pi являє собою повнофункціональний комп'ютер, у той час як Arduino назвати комп'ютером дуже складно.

					РБ101.17010201.072.007 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Розглянемо докладніше, у чому відмінності між Arduino і Raspberry Pi. Таке порівняння допоможе зробити вибір, яка з цих двох платформ найкращим чином відповідає потребам і буде ефективною базою для роботи.

Raspberry Pi родом з Великобританії. Винахідник Ебен Аптон і його колеги з комп'ютерної лабораторії Кембриджського університету створювали Raspberry Pi як дешевий комп'ютер для освітніх цілей. Аптон працював на прототипах з 2006 року, перша партія Raspberry Pi надійшла в продаж тільки в квітні 2012 року. Нещодавно вийшла нова версія Raspberry Pi — Model B +.

Arduino народився в Італії. Банзая, викладач проектного інституту Івреї, хотів простий і дешевий апаратний прототип для студентів. Існує кілька модифікацій плати (Arduino Uno, Nano, Mega та ін.). Крім того в силу відкритості платформи проводиться безліч практично ідентичних аналогів Arduino (DCCduino, FreDuino, XDRuino та ін.).

Як Arduino, так і Raspberry Pi підходять для початківців. Але якщо придивитися до апаратної частини і програмного забезпечення цих плат, стає очевидно, що вони призначені для проектів різних типів.

Для початку звернемо увагу на деякі характеристики з специфікацій плат, які показують найбільші відмінності між Raspberry Pi і Arduino.

Розміри цих двох пристроїв співставні (якщо брати найбільш відому Arduino Uno). Але це практично все, що робить їх схожими один на одного.

По тактовій частоті Raspberry Pi в 40 разів швидше, ніж Arduino. Ще більше розходження в оперативній пам'яті: Raspberry Pi має в 128000 разів більше оперативної пам'яті ніж Arduino.

Raspberry Pi є комп'ютером, на якому може бути запущена операційна система Linux, що підтримує багатозадачність. До USB-портів можна підключати різні пристрої, наприклад для бездротового підключення до мережі Інтернет. Загалом ця крихітна плата є досить потужною і може функціонувати в якості повноцінного комп'ютера (хоча, звичайно з

					РБ101.17010201.072.007 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

сучасними персональними комп'ютерами і ноутбуками конкурувати за потужністю Raspberry Pi не може).

Може здатися, що Raspberry Pi перевершує Arduino. Але ця перевага скоріше в програмному забезпеченні. Для чисто апаратних проєктів набагато краще підходить Arduino. Така перевага Arduino має за рахунок своєї простоти.

Arduino здатна краще ніж Raspberry Pi і дійсно в реальному часі зчитувати аналогові сигнали. Ця гнучкість дозволяє Arduino працювати практично з будь-яким видом датчиків або чіпів. Raspberry Pi не така гнучка, наприклад, для читання аналогових датчиків потрібні додаткові апаратні засоби.

Arduino IDE значно простіше у використанні, ніж Linux. Наприклад, якщо ви хочете написати програму для миготіння світодиодом з Raspberry Pi, ви повинні встановити операційну систему і деякі бібліотеки коду — і це тільки початок. На Arduino ви можете запрограмувати миготіння світлодіодом, використовуючи всього вісім рядків коду. Оскільки Arduino не призначений для запуску операційної системи або безлічі програм.

Raspberry Pi дозволяє використовувати переваги багатозадачності, кілька програм у фоновому режимі можуть працювати одночасно. Наприклад, Raspberry Pi, може працювати і як сервер друку, і як сервер VPN одночасно.

Arduino працює з будь-яким комп'ютером і може працювати від батареї. Arduino можна включити і благополучно вимкнути в будь-який час. Операційна система на Raspberry Pi може бути пошкоджена, якщо відключити плату без належного завершення роботи.

З Raspberry Pi повинні знати дещо про Linux, а також про програмне середовище, наприклад, Python. З іншого боку, Raspberry Pi має вигоду від десятиліть програмного забезпечення Linux.

Таким чином, у той час як Raspberry Pi має колосальну перевагу в можливості програмного забезпечення, Arduino має перевагу в простоті

					РБ101.17010201.072.007 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

реалізації апаратних проектів. Доцільність використання Raspberry Pi або Arduino залежить від проекту. Сходячи з порівняння, був обраний Raspberry Pi так як нам необхідна ОС для підключення бібліотек (п.п 2.3 -).

2.3 Бібліотека OpenALPR

OpenALPR є open-source бібліотекою ALPR (Automatic License Plate Recognition — автоматичне розпізнавання номерних знаків), яка написана на C ++. Бібліотека аналізує зображення і визначає номерні знаки. Вихідом є текстове представлення будь-яких символів номерного знака, знайдених в зображенні. OpenALPR працює як pipeline. На вхід подається зображення, обробка відбувається в кілька етапів, і виходом є можливі номерні знаки в зображенні. Етапи pipeline відбуваються в наступному порядку (таблиця 2.2):

Таблиця 2.2 — Опис класів для виконання розпізнавання

Фаза pipeline	C++ класи	Опис
Детекція	regiondetector.cpp	Знаходить потенційні області номерного знака
Бінаризація	binarizewolf.cpp	Перетворює область зображення номерного знака на чорне і біле
Char Analysis	characteranalysis.cpp	Знаходить символи розміру ‘blobs’ в регіоні пластини
Края пластини	platelines.cpp і platecorners.cpp	Знаходить края/форму номерного знака
Вирівнювання	licenseplatecandidate.cpp	Перетворює перспективу, заснованої на ідеальному розміру номерного знака.
Сегментація символів	charactersegmenter.cpp	Відокремлює і очищає символи таким чином, що вони можуть бути оброблені окремо

Продовження таблиці 2.2

OCR	ocr.cpp	Аналізує кожен символ зображення і надає безліч можливих букв
Post Processing	postprocess.cpp	Створює список найімовірніших варіантів розпізнаного номерного знака. Також виконує і Regex match з шаблонами області, якщо потрібно.

Описання кожної ітерації більш докладніше в розділі 3 «Теоретичні дані з розпізнавання номерних знаків».

OpenALPR включає утиліту командного рядка. Для цього достатньо набрати *alpr [шлях до файлу]*, щоб почати розпізнавання зображення з номерним знаком.

Детальне використання *alpr* в *cmd*:

```
user@linux:~/openalpr$ alpr --help
```

USAGE:

```
alpr [-c <country_code>] [--config <config_file>] [-n <topN>] [--seek
<integer_ms>] [-t <region code>] [--clock] [-d] [-j] [--]
[--version] [-h] <image_file_path>
```

Where:

```
-c <country_code>, --country <country_code>
Country code to identify (either us for USA or eu for Europe).
Default=us

--config <config_file>
Path to the openalpr.conf file

-n <topN>, --topn <topN>
Max number of possible plate numbers to return. Default=10

--seek <integer_ms>
Seek to the specied millisecond in a video file. Default=0

-t <region code>, --template_region <region code>
Attempt to match the plate number against a region template (e.g., md
for Maryland, ca for California)
```

					РБ101.17010201.072.007 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		


```

--clock
  Measure/print the total time to process image and all plates.
  Default=off

-d, --detect_region
  Attempt to detect the region of the plate image.  Default=off

-j, --json
  Output recognition results in JSON format.  Default=off

--, --ignore_rest
  Ignores the rest of the labeled arguments following this flag.

--version
  Displays version information and exits.

-h, --help
  Displays usage information and exits.

<image_file_path>
  Image containing license plates
OpenAlpr Command Line Utility

```

Наприклад, якщо виконати цю команду для зображення *samplecar.png*, то отримаємо такі результати:

```

user@linux:~/openalpr$ alpr ./samplecar.png
plate0: top 10 results -- Processing Time = 58.1879ms.
- PE3R2X      confidence: 88.9371
- PE32X       confidence: 78.1385
- PE3R2       confidence: 77.5444
- PE3R2Y      confidence: 76.1448
- P63R2X      confidence: 72.9016
- FE3R2X      confidence: 72.1147
- PE32        confidence: 66.7458
- PE32Y       confidence: 65.3462
- P632X       confidence: 62.1031
- P63R2       confidence: 61.5089

```



Plate	Confidence (%)	Processing Time (ms)
PE3R2X	88.94%	152.94 ms

Рисунок 2.1 — Приклад розпізнаного зображення

					РБ101.17010201.072.007 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Попередньо скомпільовані бінарні пакети знаходяться тут:

<https://github.com/openalpr/openalpr/releases> . Вони як правило, збираються з кожним новим релізом. Встановити OpenALPR на Linux треба за допомогою наступних команд:

```
wget -O - http://deb.openalpr.com/openalpr.gpg.key | sudo apt-key add -
echo "deb http://deb.openalpr.com/master/ openalpr main" | sudo tee
/etc/apt/sources.list.d/openalpr.list
sudo apt-get update
sudo apt-get install openalpr openalpr-daemon openalpr-utils libopenalpr-dev
```

Але попередній варіант встановлення є найлегшим, тоді як найскладнішим варіантом є:

1. Переконавшись, що необхідні інструменти встановлені:

- `sudo apt-get install libpng12-dev libjpeg62-dev libtiff4-dev zlib1g-dev`
- `sudo apt-get install build-essential`
- `sudo apt-get install autoconf automake libtool`
- `sudo apt-get install git-core`
- `sudo apt-get install cmake`

2. Встановити *opencv*.

3. Завантажити і встановити *leptonica* та *tesseract-ocr*:

- *tesseract-ocr* необхідна *leptonica*, а також один мовний пакет
- <http://www.leptonica.org/source/leptonica-1.70.tar.gz>
- <https://tesseract-ocr.googlecode.com/files/tesseract-ocr-3.02.02.tar.gz>
- <https://tesseract-ocr.googlecode.com/files/tesseract-ocr-3.02.eng.tar.gz>
- перемістити завантажені архіви в будь-який каталог. Скоріш за все, що вони знаходяться в `/usr/local/src/openalpr/`.

4. Розпакувати архіви:

- `tar xf /usr/local/src/openalpr/tesseract-ocr-3.02.02.tar.gz`
- `tar xf /usr/local/src/openalpr/tesseract-ocr-3.02.02.eng.tar.gz`
- `tar xf /usr/local/src/openalpr/leptonica-1.70.tar.gz`

5. Скомпілювати *leptonica* :

- `cd /usr/local/src/openalpr/leptonica-1.70/`
- `./configure --prefix=/usr/local`
- `make`
- `make install`

6. Скомпілювати *tesseract*:

- `cd /usr/local/src/openalpr/tesseract-ocr/`
- `./autogen.sh`
- `./configure`
- `make`

					РБ101.17010201.072.007 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

- sudo make install
- sudo ldconfig

7. Клонувати репозиторій *openalpr* в */usr/local/src/openalpr/* директорію

- cd /usr/local/src/openalpr/
- git clone https://github.com/openalpr/openalpr.git

8. Оновлення *CMakeLists.txt*:

- cd /usr/local/src/openalpr/openalpr/
- gedit CMakeLists.txt &
 - o SET(OpenCV_DIR "/usr/local/lib")
 - o SET(Tesseract_DIR "/usr/local/src/openalpr/tesseract-ocr")
- cmake ./
- make

OpenALPR працює на Linux, Mac OSX і Windows. OpenALPR необхідні наступні додаткові бібліотеки (в наступних підрозділах будуть описані):

- Tesseract OCR v3.0.3;
- OpenCV v2.3.8+ .

2.3.1 Тренування сенсору за допомогою train-ocr

Репозиторій train-ocr забезпечує код і дані, які можуть бути використані для підготовки користувацьких шрифтів номерного знака на підтримку бібліотеки OpenALPR. Бібліотека OCR використовується OpenALPR Tesseract. Багато з утомливих аспектів навчання OCR були автоматизовані за допомогою Python скрипта. Тим не менш, введення даних як і раніше повинно бути в певному форматі, щоб задовольнити Tesseract. Для початку, зклонювати репозиторій та ознайомитися з вхідними файлами. У папці "eu/input", є ряд TIF файлів. Номерний знак тієї чи іншої країни може мати багато шрифтів.

Іменування є: I[код_країни].[імя_шрифту].exp[PageNumber].box.

Наприклад, Європейський шрифт німецького номерного знаку буде виглядати так:

leu.germany.exp0.box

Якщо планується навчання OCR для абсолютно нової країни, спочатку потрібно налаштувати розміри номерного знаку і символів. Додати новий

					РБ101.17010201.072.007 ПЗ	Арк.
						3 2
Змн.	Арк.	№ докум.	Підпис	Дата		

файл в runtime_data/Config/ с 2-значного коду країни. Ми повинні налаштувати наступні значення:

```
plate_width_mm = [width of full plate in mm]
plate_height_mm = [height of full plate in mm]
char_width_mm = [width of a single character in mm]
char_height_mm = [height of a single character in mm]
char_whitespace_top_mm = [whitespace between the character and the top of the
plate in mm]
char_whitespace_bot_mm = [whitespace between the character and the bottom of
the plate in mm]
template_max_width_px = [maximum width of the plate before processing. Should
be proportional to the plate dimensions]
template_max_height_px = [maximum height of the plate before processing.
Should be proportional to the plate dimensions]
min_plate_size_width_px = [Minimum size of a plate region to consider it
valid.]
min_plate_size_height_px = [Minimum size of a plate region to consider it
valid.]
ocr_language = [name of the OCR language -- typically just the letter l
followed by your country code]
```

На жаль, на сьогоднішній день можливе використання лише бельгійських, нідерландських та германських тренерованих скриптів. Тому було прийнято рішення тренувати для України. Використовується ДСТУ 3650:2004 [26]. Цей стандарт визначає правила виготовлення державних номерних знаків (далі — знаки) для автомобілів, автобусів, мотоциклів, моторолерів, мотоколясок, тракторів, автомобільних і тракторних причепів, а також знаків, що їх виготовляють за індивідуальним замовленням.



Рисунок 2.2 — Знак типу 1

					РБ101.17010201.072.007 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4 Бібліотека OpenCV

OpenCV (англ. *Open Source Computer Vision Library*, бібліотека комп'ютерного зору з відкритим вихідним кодом) — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека розроблена Intel і нині підтримується Willow Garage та Itseez. Реалізована на C / C++, також розробляється для Python, Java, Ruby, Matlab, Lua та інших мов. Може вільно використовуватися в академічних та комерційних цілях — поширюється на умовах ліцензії BSD.

Бібліотека містить більше ніж 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору. Алгоритми OpenCV застосовують у таких сферах:

- аналіз та обробка зображень;
- системи з розпізнавання обличчя;
- ідентифікації об'єктів;
- розпізнавання жестів у відео;
- відстежування переміщення камери;
- побудова 3D моделей об'єктів;
- створення 3D хмар точок з стерео камер;
- склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю;
- система взаємодії людини з комп'ютером;
- пошуку схожих зображень із бази даних;
- усунення ефекту червоних очей при фотозйомці із спалахом;
- стеження за рухом очей;
- аналіз руху;
- ідентифікація об'єктів;

					РБ101.17010201.072.007 ПЗ	Арк.
						3 4
Змн.	Арк.	№ докум.	Підпис	Дата		

- сегментація зображень;
- трекінг відео;
- розпізнавання елементів сцени і додавання маркерів для створення додаткової реальності;

та інші.

OpenCV написана на C++ і її основний інтерфейс також реалізовано на C++, але бібліотека і досі представляє старіший C інтерфейс. На даний момент реалізовано інтерфейс на мовах Python, Java і MATLAB/OCTAVE (починаючи з версії 2.5). API для цих інтерфейсів можна знайти в онлайн документації. Оболонки для інших мов, таких як C#, та Ruby були розроблені з метою охоплення ширшої аудиторії. Всі нові розробки та алгоритми OpenCV у даний момент розробляються у C++ інтерфейсі.

Підтримувані платформи та інструменти:

- Microsoft Windows: компілятори Microsoft Visual C++ (6.0, .NET 2003), Intel Compiler, Borland C++, Mingw (GCC 3.x);
- Linux: GCC (2.9x, 3.x), Intel Compiler: `./configure-make-make install`, RPM;
- Mac OS X: GCC (3.x, 3.x).
- Android;
- iOS — неофіційно;

Засоби GUI, захоплення відео:

- Microsoft Windows: DirectShow, Vfw, MIL, CMU1394;
- Linux: V4L2, DC1394, FFmpeg;
- Mac OS X: QuickTime.

Комп'ютерний зір застосовується всюди: в системах безпеки, в системах виробничого контролю, в медицині, в безпілотних літальних апаратах і т.д. Воно пов'язує Google Maps і Google Earth, перевіряє пікселі на ЖК екранах і засвідчується, що застібки на ваших сорочках пришиті належним чином.

					РБ101.17010201.072.007 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

OpenCV являє собою просту у використанні бібліотеку комп'ютерного зору з більш ніж 500-ми функціями, здатними працювати в реальному часі.

OpenCV заточена під виконання в реальному часі. Вона написана на оптимізованому C і може працювати з багатоядерними процесорами. Якщо Ви хочете отримати від бібліотеки більшу продуктивність то Вам доведеться придбати IPP (Integrated Performance Primitives). IPP представляє з себе ряд ретельно оптимізованих функцій для роботи з мультимедіа, криптографією, розпізнаванням мови, стисненням даних і багато чим ще. OpenCV автоматично використовує IPP, якщо вона встановлена.

Однією з цілей OpenCV є надання простий у використанні бібліотеки машинного зору, яка допомагає створювати досить складні додатки в короткі терміни. OpenCV містить в собі більше 500 функцій які охоплюють багато областей, включаючи контроль якості на виробництві, системи безпеки, обробку зображень в медицині, інтерфейси, стереоскопічний зір і роботобудування. Оскільки комп'ютерний зір і навчання машин йдуть "рука об руку", OpenCV також включає в себе бібліотеку загального призначення MLL (Machine Learning Library, Бібліотека Навчання Машин). Ця підбібліотека вирішує завдання статистичного розпізнавання образів і кластеризації. MLL дуже корисна для вирішення складних завдань, пов'язаних з комп'ютерним зором, але є досить загальною, для того щоб вирішувати більшість завдань навчання машин.

Основні модулі бібліотеки:

- `cxcore` — ядро містить базові структури даних і алгоритми:
 - Базові операції над багатовимірними числовими масивами
 - Матрична алгебра, математичні ф-ції, генератори випадкових чисел
 - Запис/оновлення структур даних в / з XML
 - Базові функції 2D графіки
- `CV` — модуль обробки зображень та комп'ютерного зору:

					РБ101.17010201.072.007 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

- Базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів і т.д.)
- Аналіз зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми)
- Аналіз руху, стеження за об'єктами
- Виявлення об'єктів, зокрема осіб
- Калібрування камер, елементи відновлення просторової структури
- Highgui — модуль для введення/виведення зображень і відео, створення користувацького інтерфейсу:
 - Захоплення відео з камер і з відео файлів, читання / запис статичних зображень.
 - Функції для організації простого UI (всі демо додатки використовують HighGUI)
- Cvaux — експериментальні і застарілі функції:
 - Просторів. зір: стерео калібрації, само калібрації
 - Пошук стерео-відповідності, кліки в графах
 - Знаходження і опис рис обличчя
- CvSam — захоплення відео: Дозволяє здійснювати захоплення відео з цифрових відео-камер (підтримка припинена і в останніх версіях цей модуль відсутній)

					РБ101.17010201.072.007 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

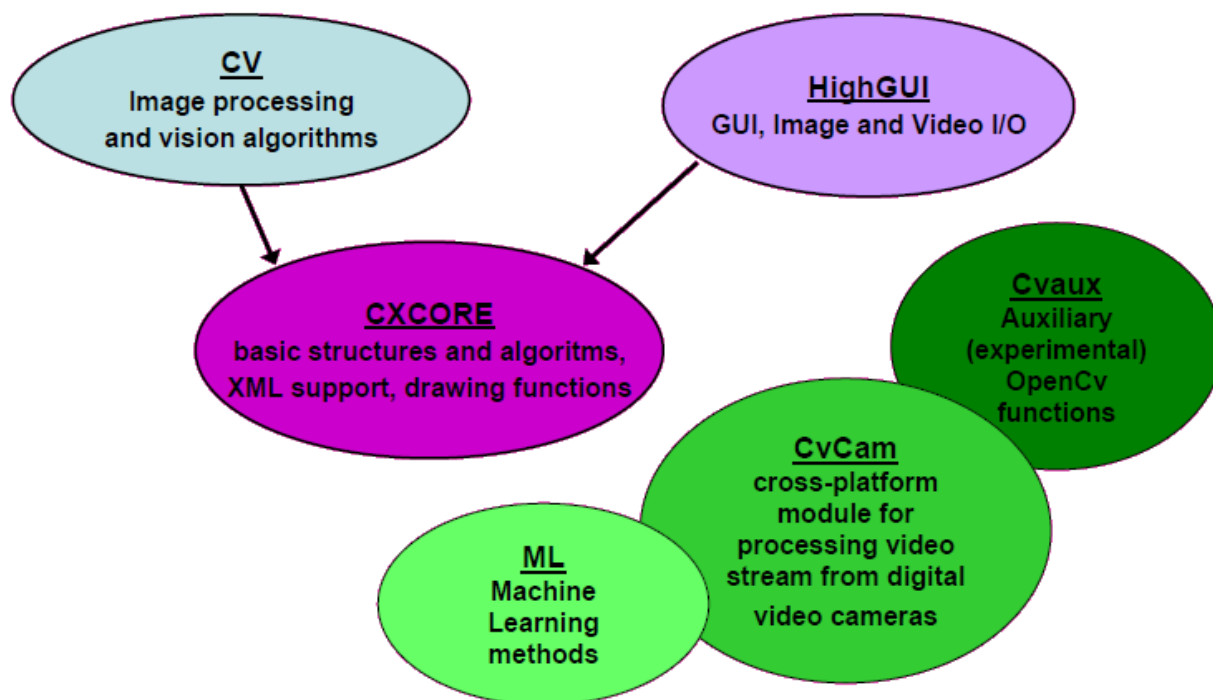


Рисунок 2.3 — Загальна структура системи розпізнавання номерних знаків

У версії 2.3 структура бібліотека реорганізована — тепер замість великих універсальних модулів (cxcore, Cvaux, Highgui, Cvaux) бібліотека OpenCV розділена на невеликі модулі по функціональному використанню:

- opencv_features2d — різні дескриптори (SURF).
- opencv_video — аналіз руху і відстеження об'єктів (оптичний потік, шаблони руху, усунення фону).
- opencv_objdetect — детектування об'єктів на зображенні (вейвлети Хаара, HOG і т. д.).
- opencv_calib3d — калібрування камери, пошук стерео-соответствующая і елементи обробки тривимірних даних.
- opencv_flann — бібліотека швидкого пошуку найближчих сусідів (FLANN).
- opencv_contrib — супутній код, ще не готовий для застосування.

- `opencv_legacy` — застарілий код, збережений заради зворотної сумісності.
- `opencv_gpu` — прискорення деяких функцій OpenCV за рахунок CUDA (NVidia).
- `opencv_core` — ядро: базові структури, обчислення (математичні функції, генерація псевдовипадкових чисел, DFT, DCT, введення / висновки в XML тощо).
- `opencv_imgproc` — обробка зображень (фільтри, перетворення і т. д.).
- `opencv_highgui` — простий UI, завантаження / збереження зображень і відео.
- `opencv_ml` — методи і моделі машинного навчання (SVM, дерева прийняття рішень і т. д.).

2.4.1 Комп'ютерний зір

Комп'ютерний зір — це перетворення даних, що надходять з фото- або відеокамери, в ухвалення рішення або інше уявлення. Всі подібні перетворення виконуються для досягнення певної мети. Вхідні дані можуть включати в себе якусь контекстну інформацію, начебто "камера встановлена в машині" або "лазер вказує на об'єкт, відстань до якого 1 метр". Новим поданням може бути перетворення кольорового зображення в чорно-біле або видалення руху камери з відеозапису.

Оскільки ми з вами є "візуальними" істотами, то ми можемо помилково вважати, що машинне зір — це дуже просто. Наскільки важко знайти, скажімо, автомобіль, розташований на картинці? Ваша інтуїція може ввести вас в оману. Людський мозок ділить зоровий сигнал на кілька різних потоків інформації. Ваш мозок має систему, яка знаходить важливі частини зображення і зосереджує увагу на них, в той же час пригнічуючи обробку інших частин. Це робиться способом, залежним від поточного завдання. Ми поки погано розуміємо пристрій цієї системи. Є також асоціативні входи від

					РБ101.17010201.072.007 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

усіх органів чуття, на підставі даних з яких мозок може використовувати асоціації, накопичені за роки життя в світі. Також людське око здатне автоматично звужувати або розширювати зіницю залежно від освітлення, і змінювати діаметр кришталика фокусуючи картинку на сітківці.

У системі ж машинного зору комп'ютер отримує всього-на-всього масив чисел з камери або диска — і більше нічого. Як правило, немає ні вбудованого розпізнавача образів, ні автоматичного управління фокусом і об'єктивом, ні асоціацій, накопичених за роки життя. Здебільшого системи машинного зору все ще дуже примітивні.

На рисунку 2.4 зображений автомобіль з виділеним бічним дзеркалом. Однак камера побачить всього лише масив чисел. Будь-яке число в цьому масиві має дуже велику шумову складову, і саме по собі дає нам дуже мало інформації, але цей масив — все, що здатний "побачити" комп'ютер. Наше завдання полягає в тому, щоб перетворити цей масив у сприйняття "дзеркала".

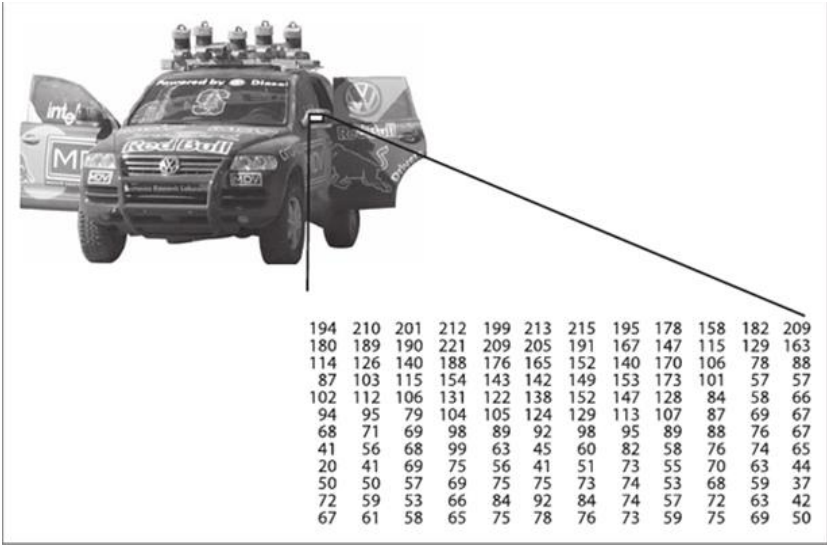


Рисунок 2.4 — Загальна структура системи розпізнавання номерних знаків

Рисунок 2.4 ще більше показує, чому машинне зір — це так складно. Насправді проблема ще складніше, ніж ми її представили спочатку, оскільки формально її неможливо вирішити. Маючи двовимірне (2D) подання

тривимірного (3D) світу, немає єдиного способу відтворення тривимірного сигналу.

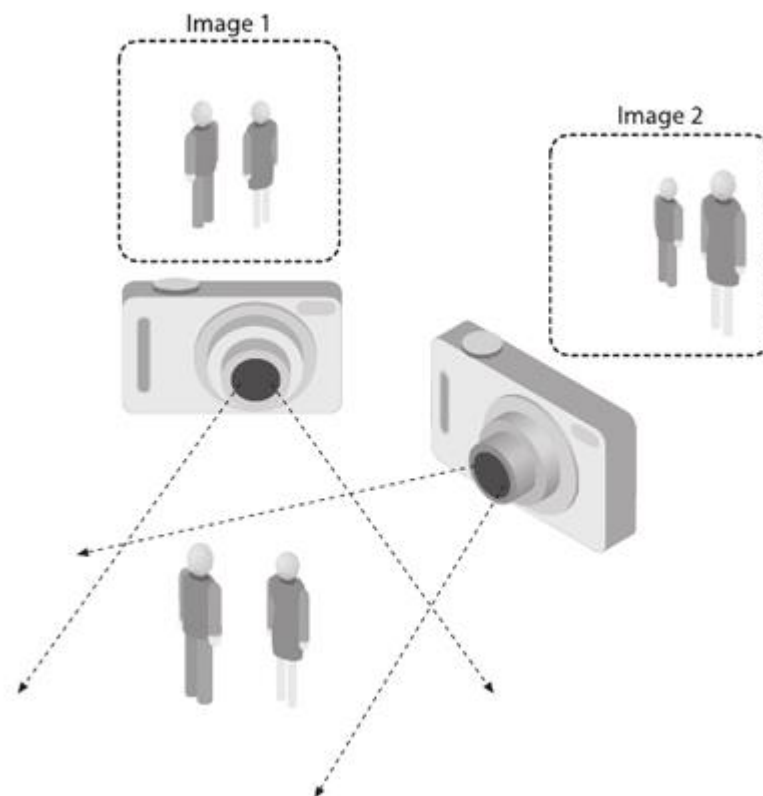


Рисунок 2.5 — Стереозір

Формально, така некоректна задача не має однозначного або остаточного рішення. Одна і та ж 2D-картинка може бути представленням нескінченної кількості 3D-сцен, навіть якщо якість даних було ідеальним. Насправді ж, як уже згадувалося, дані зазвичай пошкоджені шумами і спотвореннями. Ці проблеми пов'язані із змінами у світі (погода, освітлення, відображення, рух), недоліками об'єктивів, кінцевим часом витримки, електричними шумами в камері, артефактами стиснення зображення. З урахуванням всіх цих проблем, як ми можемо добитися хоч якогось небудь успіху?

При розробці практичної системи ми можемо використовувати знання про контекст, щоб обійти накладаються на нас обмеження. Розглянемо приклад мобільного робота, який повинен знайти і зібрати всі степлери з

					РБ101.17010201.072.007 ПЗ	Арк.
						4 1
Змн.	Арк.	№ докум.	Підпис	Дата		

будівлі. Робот може використовувати той факт, що столи зазвичай знаходяться всередині будівлі, а степлери зазвичай лежать на столах. Це дає неявну посилання на розмір; степлер повинен міститися на стіл. Це також допомагає усунути помилкові розпізнання, оскільки степлер не може лежати, наприклад, на вікні або стелі. Робот також може благополучно проігнорувати 200-футовий рекламний дирижабль у формі степлера, тому що під ним немає текстури дерева. Навпаки зображення степлера в базі даних може мати дуже великий розмір або мати незвичайну конфігурацію, так як це пов'язано з тим, хто робив фото. Фотограф міг сфотографувати тільки степлери однакових розмірів. Люди також мають тенденцію знімати об'єкти так, щоб вони були в центрі зображення, і були повернені «по-звичайному» — скажімо, навряд чи хтось стане фотографувати людини догори ногами. Таким чином у фотографіях, зроблених людьми, є багато «зайвої інформації» включеної туди ненавмисно.

Контекстна інформація також може бути змодельована за допомогою алгоритмів навчання машин. Приховані параметри, такі як розмір і орієнтація на гравітацію, можуть бути позначені в безлічі, на якому проводиться навчання. Крім того можна спробувати виміряти приховані параметри за допомогою додаткових датчиків. Застосування лазерного далекоміра дозволяє точно виміряти розміри об'єкта.

Наступна проблема, з якою стикаються при вивченні комп'ютерного зору — це шум. Скажімо, завдання про визначення кордонів у зображенні може виявитися неймовірно складною, якщо ми будемо вирішувати її, порівнюючи точки зі своїми найближчими сусідами. Однак, якщо збирати інформацію з околиці точки, завдання істотно спрощується. Шум також можна видаляти, збираючи інформацію протягом проміжку часу. Інші методи будують моделі шуму на підставі доступних даних. Наприклад, так як спотворення, пов'язані з лінзами, добре вивчені, достатньо лише знайти

					РБ101.17010201.072.007 ПЗ	Арк.
						4 2
Змн.	Арк.	№ докум.	Підпис	Дата		

параметри простого многочлена, щоб описати — і, таким чином, майже повністю видалити — подібні спотворення.

Дії або рішення, що виконуються комп'ютером на основі даних з камери, зазвичай виконуються в контексті конкретного завдання. Ми можемо видалити із зображення перешкоди і спотворення так, щоб наша система безпеки була тривогу якщо хтось перелазить через паркан, або зробити систему підрахунку людей, що пройшли повз камери. Програмне забезпечення для роботів, які їздять по офісним будівлям, буде сильно відрізнятися від ПО для стаціонарних камер, тому ці дві системи мають суттєво різні цілі. Як правило, чим більше обмежень має умова конкретного завдання, тим простіше і надійніше буде рішення.

Метою бібліотеки OpenCV є надання основних інструментів, необхідних для вирішення завдань комп'ютерного зору. Можливо, для вирішення навіть досить складної задачі функціональності бібліотеки буде достатньо. Навіть якщо це не так, базові компоненти бібліотеки досить просунуті, щоб можна було створити власне рішення практично будь-якої задачі комп'ютерного зору. Для останнього випадку є кілька перевірених методів використання бібліотеки; всі вони починаються з вирішення задачі з використанням найбільшої кількості компонентів бібліотеки. Як правило, після того, як розробите це перше чорнове рішення, ви побачите його недоліки і зможете їх виправити, використовуючи свій код і розум (також відоме як "вирішувати проблему, яка перед вами стоїть, а не ту, що ви придумали"). Після цього ви зможете використовувати ваше чорнове рішення в якості тесту, щоб оцінити зроблені вами поліпшення. Нарешті, які б недоліки не залишалися, з ними можна справитися, використовуючи обмеження більшої системи, що включає проблему.

					РБ101.17010201.072.007 ПЗ	Арк.
						4 3
Змн.	Арк.	№ докум.	Підпис	Дата		

2.5 OpenALPR демон (alprd)

OpenALPR демон дозволяє контролювати потік з камери для номерних знаків у фоновому режимі. Alprd працює як процес демона на Linux. Цифри знаків можуть передаватися на інший сервер (за допомогою HTTP повідомлень) або можна вживати програмно за допомогою beanstalkd. Alprd працює таким чином:

1. Потік зображення постійно витягується з IP-камери за допомогою MJPEG через HTTP.
2. alprd обробляє потік так швидко, як це може шукати номерні знаки на зображенні. Демон автоматично пропускає кадри, щоб залишитися в синхронізації з часом годин.
3. Коли один або більше знаків детектовані, інформація записується в локальну чергу beanstalkd як JSON файл.
4. Можливо, alprd також зберегати зображення як JPEG
5. Можливо, alprd також працює як окремий процес, який виснажує дані, поміщені в чергу beanstalkd і завантажує на віддалений HTTP-сервер через POST.

Alprd може бути використаний у двох режимах:

1. Результати розпізнавання передаються на веб-сервер.
2. Результати розпізнавання можна прочитати з черги beanstalkd.

Діаграма вище показує, що alprd використовується для потокової передачі даних на HTTP-сервер. Alprd налаштований за допомогою дистанційного HTTP адреса. Як номерні знаки визначені, сервер відправляє дані в форматі JSON з віддаленим сервером по протоколу HTTP. Beanstalkd і процес alprd спільно розміщені на одному сервері.

					РБ101.17010201.072.007 ПЗ	Арк.
						4 4
Змн.	Арк.	№ докум.	Підпис	Дата		

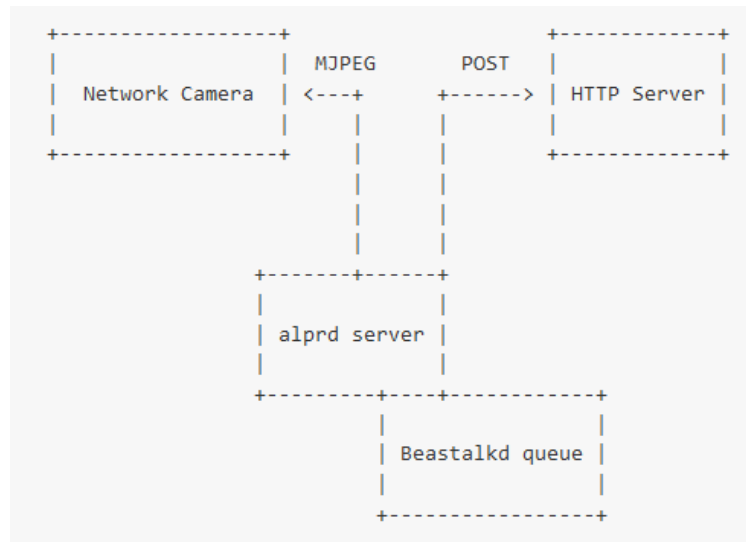


Рисунок 2.6 — Демон alprd

alprd потребує принаймні один визначений потік. Це просто URL для потоку MJPEG. Можна використовувати декілька потоків на одному сервері — кожен потік породжує окремий процес, який намагається використовувати ядро процесора.

Приклад JSON-файла:

```

{
  "uuid": "e11e0acc-6aaf-4817-9299-9e6773043b8e",
  "camera_id": 1,
  "site_id": "watchtower-hq",
  "img_width": 640,
  "img_height": 480,
  "epoch_time": 1402161050,
  "processing_time_ms": 138.669163,
  "results": [
    {
      "plate": "S11FRE",
      "confidence": 77.130661,
      "matches_template": 0,
      "region": "",
      "region_confidence": 0,
      "coordinates": [
        {
          "x": 218,
          "y": 342
        },
        {
          "x": 407,
          "y": 325
        },
        {
          "x": 407,
          "y": 413
        }
      ]
    }
  ]
}

```

					РБ101.17010201.072.007 ПЗ	Арк.
						4 5
Змн.	Арк.	№ докум.	Підпис	Дата		


```

    {
      "x": 218,
      "y": 431
    }
  ],
  "candidates": [
    {
      "plate": "S11FRE",
      "confidence": 77.130661,
      "matches_template": 0
    },
    {
      "plate": "S11ERE",
      "confidence": 75.496307,
      "matches_template": 0
    },
    {
      "plate": "S11RE",
      "confidence": 75.440361,
      "matches_template": 0
    },
    {
      "plate": "S11CRE",
      "confidence": 75.340179,
      "matches_template": 0
    },
    {
      "plate": "S11FHE",
      "confidence": 75.240974,
      "matches_template": 0
    },
    {
      "plate": "EJLESSIE",
      "epoch_time": 1402158050,
      "confidence": 78.167984,
      "matches_template": 0,
      "region": "",
      "region_confidence": 0,
      "processing_time_ms": 51.650604,
      "coordinates": [
        {
          "x": 226,
          "y": 369
        },
        {
          "x": 348,
          "y": 348
        },
        {
          "x": 355,
          "y": 406
        },
        {
          "x": 231,
          "y": 429
        }
      ]
    }
  ],
  "candidates": [
    {
      "plate": "EJLESSIE",
      "confidence": 78.167984,

```

					РБ101.17010201.072.007 ПЗ	Арк.
						4 6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        "matches_template": 0
    },
    {
        "plate": "VI5A",
        "confidence": 58.167984,
        "matches_template": 0
    },
    {
        "plate": "EDLESSIE",
        "confidence": 77.61319,
        "matches_template": 0
    }
]
}
]
}

```

2.6 Бібліотека Tesseract OCR

Tesseract — вільна програма для розпізнавання текстів, розроблялася Hewlett-Packard з 1985 до 1994 року. Не так давно Google купив її та відкрив джерельні тексти під ліцензією Apache 2.0 у 2006 році для продовження розробки. У цей час програма вже працює з UTF-8, розпізнає багато мов, серед яких і українська.

Доки HP мав незалежно один від одного технологію, розроблена компонування сторінки аналізу, який був використаний в продуктах (і, отже, не призначені для відкритого джерела) Tesseract ніколи не потрібен був свій аналіз шаблонів. Тому його вхід являє собою бінарне зображення з додатково визначеними полігональними текстовими областями. Обробка проводиться pipeline крок за кроком, але деякі з етапів були незвичайними в свій час, і, можливо, залишаються такими до сих пір. Першим кроком є зв'язний аналіз компонентів, в якому контури компонентів зберігаються.

Обчислювально це було дорогим рішенням в той час, але мало суттєву перевагу: розпізнати контури само легко, як чорним по білому тексту. Tesseract був, ймовірно, першим OCR Engine в змозі впоратися з чорно-білим текстом як з тривіальною задачею. На даному етапі, контури зібралися разом, чисто Верстка, в краплі. Blobs організовані в текстових рядків, а лінії та регіонів аналізуються для фіксованого кроку і пропорційного тексту. Текстові

					РБ101.17010201.072.007 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

рядки розбиті на слова по-різному залежно від виду інтервал символу. Фіксований текст тони нарізаний відразу осередках символів. Пропорційний текст розбивається на слова, використовуючи певний простору і нечіткі простору. Визнання потім переходить в процесі два проходи. У першому проході, робиться спроба розпізнати кожне слово в свою чергу. Кожне слово, що є задовільним передається до адаптивного класифікатору якості навчальних даних.

Адаптивний класифікатор при цьому отримує можливість більш точно розпізнавати текст нижче по сторінці адаптивної класифікатор, можливо, дізнався дещо корисне занадто пізно, щоб внести свій внесок у верхній частині сторінки, другий прохід виконується по сторінці, в якому слова, які не були визнані досить добре розпізнаються знову. Заключний етап вирішує нечіткі простору, і перевіряє альтернативні гіпотези для x висота, щоб знайти текст з малою капіталізацією.

2.7 Результати використання бібліотек

Підключивши веб-камеру та Raspberry Pi (використовували ОС Raspbian), скомпілюємо всі необхідні бібліотеки, як було написано в підрозділі 2.3. Загальний час, затрачений на компіляцію — 4.5 годин. Після компіляції всіх необхідних інструментів та бібліотек, я написала простий скрипт на Python, який приймає фото з веб-камери і обробляє його за допомогою OpenALPR. Виконання Python-скрипта і представляє результати розпізнавання номерних знаків.

Скористались такою інструкцією:

```
# Інсталяція передумов
sudo apt-get install libopencv-dev libtesseract-dev git cmake build-essential
libpython3-dev
sudo apt-get install liblog4cplus-dev libcurl3-dev

# Інсталюємо демон beanstalkd
sudo apt-get install beanstalkd
```

					РБ101.17010201.072.007 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

```
# Клонуємо останню версію з GitHub
git clone https://github.com/openalpr/openalpr.git

# Директорія build
cd openalpr/src
mkdir build
cd build

# setup the compile environment
cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr -DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..

# Компілюємо бібліотек
make

# Встановила виконавчі файли/бібліотеки для локальної системи (/usr)
sudo make install

# Тестуємо бібліотеку
wget http://easy-clan.com/ski/pics/license_plate.JPG -O lp.jpg
alpr lp.jpg
```

Веб-камера робить знімки розміром 640×480. Розпізнавання займає деякий час, близько 5,1 секунда. Я тестувала з 10 номерними знаками і результати розпізнавання вельми добрі. Всі номерні знаки були успішно розпізнані.

Після компіляції всіх необхідних бібліотек, необхідно налаштувати веб-камеру. По-перше, як увійшли по ssh через putty, виконати команду:

```
sudo apt-get update && sudo apt-get upgrade
```

Для того, щоб ми могли працювати з нашою веб-камерою, нам необхідно встановити з репозиторію програмне забезпечення motion. Робиться це за допомогою команди:

```
sudo apt-get install motion
```

Після інсталяції програми motion, необхідно здійснити її настройку. Відкриваємо файл *motion.conf* за допомогою команди:

```
sudo nano /etc/motion/motion.conf
```

					РБ101.17010201.072.007 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

Цей файл є достатньо великим, але нас цікавлять тільки дві строчки:

```
DAEMON = OFF (змінити на ON)
Webcam_localhost = ON (змінити на OFF)
```

Далі використаємо:

```
sudo nano /etc/default/motion
start_motion_daemon = no (змінимо на yes)
```

Запустити трансляцію відео:

```
sudo service motion start
```

Зупинити трансляцію:

```
sudo service motion stop
```

Тепер наш стрім доступний по локальній мережі за адресою *http://IP_адреса_raspberri_pi: 8081* (8081 — це порт для трансляції відео).

Зараз необхідно зробити розпізнавання у реальному часі з відеопотоку. А для цього ми розробили скрипт, який реалізує вище поставлену задачу. Створимо *class PlateReader*, який реалізується методами *webcam_subprocess*, *alpr_subprocess*, *alpr_json_results* та *read_plate*. По-перше, необхідно встановити *fswebcam*, так як при інтерпретації буде таке повідомлення:

```
Traceback (most recent call last):
File "PyALPR.py", line 64, in
plate_reader.read_plate()
File "PyALPR.py", line 40, in read_plate
alpr_json, alpr_error = self.alpr_json_results()
File "PyALPR.py", line 25, in alpr_json_results
self.webcam_subprocess().communicate()
File "PyALPR.py", line 17, in webcam_subprocess
return subprocess.Popen(self.webcam_command_args, stdout=subprocess.PIPE)
File "/usr/lib/python2.7/subprocess.py", line 710, in __init__
errread, errwrite)
File "/usr/lib/python2.7/subprocess.py", line 1335, in _execute_child
raise child_exception
OSError: [Errno 2] No such file or directory
```

					РБ101.17010201.072.007 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Тому необхідно виконати такий алгоритм:

```
sudo apt-get install fswebcam
cd /home/pi
touch fswebcam.conf
sudo nano /home/pi/fswebcam.conf
```

І в файлі *fswebcam.conf* прописати:

```
device /dev/video0
pid 9990
input 0
loop 10
skip 20
resolution 640x480
top-banner
title "ALPR@RPi"
timestamp "%d-%m-%Y %H:%M:%S (%Z) "
palette YUYV
jpeg 100
save /home/pi/webcam/capture.%Y%m%d%H%M%S".jpg
save /usr/share/nginx/www/wc/capture.jpg
```

де *device /dev/video0* — відеопристрій, *loop 10* — захоплення зображення кожні 10 секунд, *skip 20* — перед захопленням зображення пропускати 20 кадрів.

Результат буде у вигляді:

```
"Guess {0:d}: {1:s} {2:.2f}%".format(ordinal, candidate["plate"],
candidate["confidence"])
```

Наприклад, для зображення нижче, результат буде таким:



Рисунок 2.7 — Вхідне зображення для подальшої обробки

					РБ101.17010201.072.007 ПЗ	Арк.
						5 1
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Guess 1: OS802HN 82.70%
Guess 2: DS802HN 82.03%
Guess 3: QS802HN 81.50%
Guess 4: GS802HN 79.99%
Guess 5: BS802HN 77.11%
Guess 6: CS802HN 77.04%
Guess 7: SS802HN 75.12%

```

Рисунок 2.8 — Ймовірні результати розпізнавання

Усі експерименти будуть надані в главі 4 «Практичні результати».

2.8 Висновки

В даному розділі була розглянута конфігурація пристрою, різницю з Arduino, описання використаних бібліотек та приклади команд, які необхідні для їх інсталяції, а також наведено приклад роботи сучасної бібліотеки OpenALPR, який і потребував OpenCV, TesseractOCR, і т.д. Взято тестове зображення, і оброблене за допомогою OpenALPR. В наступних розділах буде модернізуватись вивід інформації не в командному рядку, а у веб-інтерфейсі.

					РБ101.17010201.072.007 ПЗ	Арк.
						5 2
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ОСНОВНІ ТЕХНІКИ, ВИКОРИСТАНІ ПРИ РОЗПІЗНАВАНІ НОМЕРНОГО ЗНАКУ

Розпізнавання державних реєстраційних знаків транспортних засобів (number plate reading, NPR або license plate recognition, LPR) відноситься до технологій, що стали обов'язковими в сучасних вимогах безпеки [11].

Алгоритм, запропонований у цій роботі, розроблений для розпізнавання номерних знаків транспортних засобів вже в бібліотеці OpenALPR. Вхідними даними системи є зображення транспортного засобу, захоплене відеокамерою. Захоплене зображення, зняте приблизно з 3-5 метрів спочатку перетворюється в півтонове (у відтінки сірого, градації сірого). Застосовуємо алгоритм вертикального виявлення меж і морфологічну операцію, відкритий і близький для вилучення номерного знака. Після застосування морфологічних операцій зображення фільтрується, щоб отримати точну область номерного знака. Сегментація рядків відокремлює рядок у номерному знаку, і поділ стовпця відокремлює символи з рядка. Зрештою, розпізнавання OCR розпізнає символи, що дають результат як номерний знак у форматі ASCII (є можливість копіювання розпізнаних номерів у текстові редактори).

У загальному випадку, розпізнавання тексту складається з наступних процедур і методів (рисунок 3.1):

- попередня обробка;
- сегментація;
- розпізнавання.

Під сегментацією розуміється процес поділу зображення на окремі символи.

Кінцевий етап обробки — розпізнавання. Для цього етапу вхідними даними є зображення, отримані в результаті процесу сегментації.

Нижче в підрозділах буде докладніше описаний даний алгоритм.

					РБ101.17010201.072.007 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1 Перетворення зображення у відтінки сірого

Відтінки сірого (градації сірого, [англ. Grayscale]) — кольоровий режим зображень, який відображається у відтінках сірого кольору, розміщені у вигляді таблиці як еталони яскравості білого кольору. Найчастіше використовують ступеневе зображення рівномірного ряду оптичної щільності нейтрально-сірих полів. Широко застосовується в кольорознавстві і колористиці, для оцінки і вимірів якості тонопередачі при фотографічній зйомці, скануванні, при копіювальних та друкованих процесах (поліграфія) [14].

Сіра шкала — це контрольне зображення рівномірного ряду оптичних густин нейтрально-сірих полів, виготовлених на прозорій чи непрозорій основі і призначене для оцінки та вимірювань якості тонопередачі при фотографічній зйомці, скануванні, копіювальних та друкованих процесах. Значення яскравості сірої шкали нерідко виражають у відсотках, при цьому 0% представляє білий колір (відсутність чорного пігменту на білому фоні), 100% — чорний колір (плашка глибокого чисто-чорного пігменту).

У комп'ютерному уявленні широко поширена сіра шкала використовує на кожен піксель зображення один байт (8 біт) інформації. Така шкала передає 256 відтінків (градацій) сірого кольору, або яскравості (значення 0 представляє чорний колір, а значення 255 — білий).

3.1.1 Перетворення кольорового зображення у відтінки сірого

У колірних просторах яскравість Y' обчислюється таким чином:

$$Y' = 0.299R + 0.587G + 0.114B. \quad (3.1)$$

Для врахування особливостей сприйняття зображення людським оком (чутливість до зеленого й синього кольору) використовують інші коефіцієнти:

					РБ101.17010201.072.007 ПЗ	Арк.
						5 4
Змн.	Арк.	№ докум.	Підпис	Дата		

$$Y' = 0.2126R + 0.7152G + 0.0722B. \quad (3.2)$$

Нижче етап попередньої обробки для вилучення номерного знака:

$$I(i, j) = 0.114 \cdot A(i, j, 1) + 0.587 \cdot A(i, j, 2) + 0.299 \cdot A(i, j, 3), \quad (3.3)$$

де $I(i, j)$ є масивом сірого зображення, $A(i, j, 1)$, $A(i, j, 2)$, $A(i, j, 3)$ — R , G , B значення вихідного зображення відповідно. Іноді зображення може бути дуже темним, містити розмиття, таким чином, роблячи завдання вилучення номерного знака важким. Для того, щоб розпізнати номерний знак навіть у нічних умовах, поліпшення контрастності важливо перед подальшою обробкою [15].

3.2 Математична морфологія

Одним з порівняно нових напрямків в аналізі зображень є застосування апарату математичної морфології. Початок математичної морфології, що використовує уявлення теорії множин і інтегральної геометрії, було покладено роботами французьких дослідників Ж.Матерона і Дж.Серра, що займалися проблемами мінералогії і петрографії. Мета їх досліджень полягала в кількісному описі фізичних і механічних властивостей матеріалів за допомогою аналізу їх геометричної структури. За наступне час математична морфологія досягла стану серйозного інструменту обробки зображень з основним застосуванням в матеріалознавстві, дослідженні цитологічних препаратів, аналізі медичних зображень.

Математична морфологія (ММ) — теорія і техніка аналізу та обробки геометричних структур, заснована на теорії множин, топології і випадкових функціях. В основному застосовується в обробці цифрових зображень, але

					РБ101.17010201.072.007 ПЗ	Арк.
						5 5
Змн.	Арк.	№ докум.	Підпис	Дата		

також може бути застосовна на графах, полігональній сітці, стереометрії та багатьох інших просторових структурах.



Рисунок 3.2 — Форма (синя) і її морфологічне розширення (зелене) і звуження (жовте) ромбічним структурним елементом

Морфологія в контексті аналізу і обробки зображень описує властивості форми і структури його областей. Тобто морфологічні перетворення пов'язані з аналізом зображень на основі форми.

Морфологічні перетворення здійснюються за допомогою фільтрів:

- дилатація;
- ерозія;
- розмикання;
- замикання;
- градієнтне;
- перетворення Top Hat;
- перетворення Black Hat.

Дані перетворення проводяться за допомогою структурообразуючого елемента (примітиву), накладення якого на зображення вирішує певні завдання, і засновані на операціях над множинами. Множинами в морфології

					РБ101.17010201.072.007 ПЗ	Арк.
						5 6
Змн.	Арк.	№ докум.	Підпис	Дата		

іменуються об'єкти на зображенні: наприклад, безліч точок в двовимірному просторі. Так, всі чорні пікселі чорно-білого зображення є одним із способів його морфологічного опису [16].

Перераховані фільтри використовуються для вилучення складових зображення, потрібних для його представлення та опису (кордонів, кістяків, опуклих оболонок і т.д.). У контексті дипломної роботи, математична морфологія використана для представлення кордонів номерного знака (license plate).

За допомогою морфології можливий пошук країв об'єктів на зображенні, їх аналіз та спостереження присутніх текстур. Таким чином, використання даних перетворень можливо для стиснення зображень, аналізу складових компонент, а також для придушення шуму. В основному вони застосовуються для підвищення якості, сегментації (наприклад, для спрощення подання) і відновлення зображень.

Морфологічні операції, найчастіше, застосовуються до двійковим зображенням, які виходять після порогового перетворення, і до напівтоновим (відтінки сірого) зображень.

Дилатація («розширення» або «нарощування») — поряд з перетвореннями ерозія, розмикання і замикання, одна з основних операцій математичної морфології.

Фільтр дилатація збільшує область зображення, розширюючи його пікселі і тим самим сприяючи об'єднанню областей зображення, які були розділені шумом та ін.

Зображення після фільтра стає світліше і злегка розмитим. Тобто темні деталі послаблюються або взагалі зникають, що залежить від співвідношення їх розмірів і яркостей із заданими параметрами фільтра.

Це операція визначення локального максимуму за деякою округи, яка задається структуроутворюючим елементом.

					РБ101.17010201.072.007 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

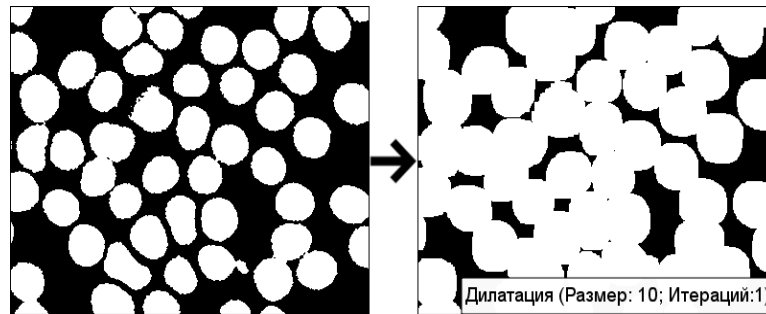


Рисунок 3.3 — Приклад операції дилатації

Параметрами фільтра є:

- Розмір (size);
- Ітерації (iteration).

Розмір задає околиця дії фільтра навколо кожної точки. Ітерація задає кількість дій.

Морфологічні оператори можуть бути застосовані до greyscale зображенням, наприклад, для зменшення шуму або збільшення яскравості зображення.

3.3 Локальні бінарні шаблони (LBP)

Локальний бінарний шаблон (LBP) є простий, але дуже ефективний оператор, який помічає пікселі зображення шляхом порогового сусідства кожного пікселя і видає результат у вигляді бінарного числа.

Завдяки дискримінаційного впливу та обчислювальній простоті LBP оператор став популярним підходом в різних програмах. Це можна розглядати як єдиний підхід до традиційно розходячи з статистичних та структурних моделей текстури аналізу. Можливо, найбільш важливою властивістю LBP оператора в реальних програмах є стійкість до монотонних змін градацій сірого, викликаних, наприклад, змінами освітленості. Ще однією важливою властивістю є його обчислювальна простота, що робить можливим аналіз зображень в складних налаштуваннях в режимі реального часу.

					РБ101.17010201.072.007 ПЗ	Арк.
						5 8
Змн.	Арк.	№ докум.	Підпис	Дата		

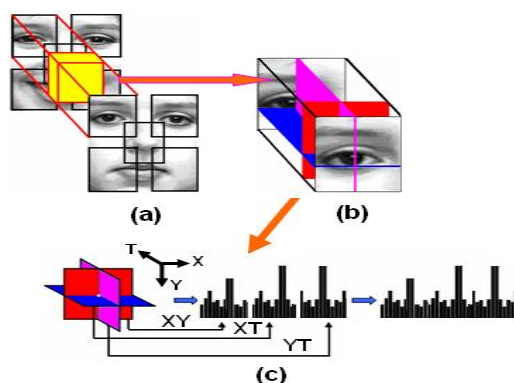


Рисунок 3.4 — Приклад LBP

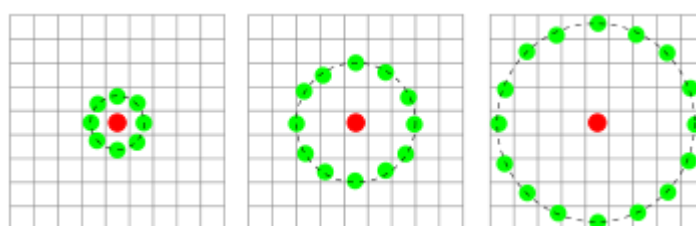


Рисунок 3.5 — Приклад LBP

Основна ідея розробки оператора LBP, що двовимірні текстури можуть бути описані двома додатковими заходами: локальні просторові паттерни і контраст шкали сірого. Оригінальний оператор LBP (Ojala, 1996) утворює позначки для пікселів зображення шляхом порогової інтерполяції 3x3 методом найближчого сусіда кожного пікселя з центрального значенням і враховує результат у вигляді двійкового числа. Гістограма з цих $2^8=256$ різних позначок можуть бути використані в якості дескриптора текстури. Цей оператор використовується спільно з простим локальним заходом контрасту за умови дуже гарної продуктивності без нагляду текстурної сегментації (Ojala and Pietikäinen 1999). Після цього, багато спорідненні методи були розроблені для текстури та кольору текстурної сегментації.

Оператор LBP був розширений для використання найближчого сусіда різних розмірів (Ojala та ін. 2002). Використання найближчого сусіда і білінійну інтерполяцію значень в нецілих координатах пікселів допускати

					РБ101.17010201.072.007 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

радіус і кількість пікселів по сусідству. Дисперсія сірої шкали локального найближчого сусіда можуть бути використані в якості додаткового контрастного заходу. Надалі, позначення (P, R) будуть використовуватися для піксельних сусідів, які означають точки P по колу радіуса R. Рисунок 3. для прикладу розрахунку LBP.

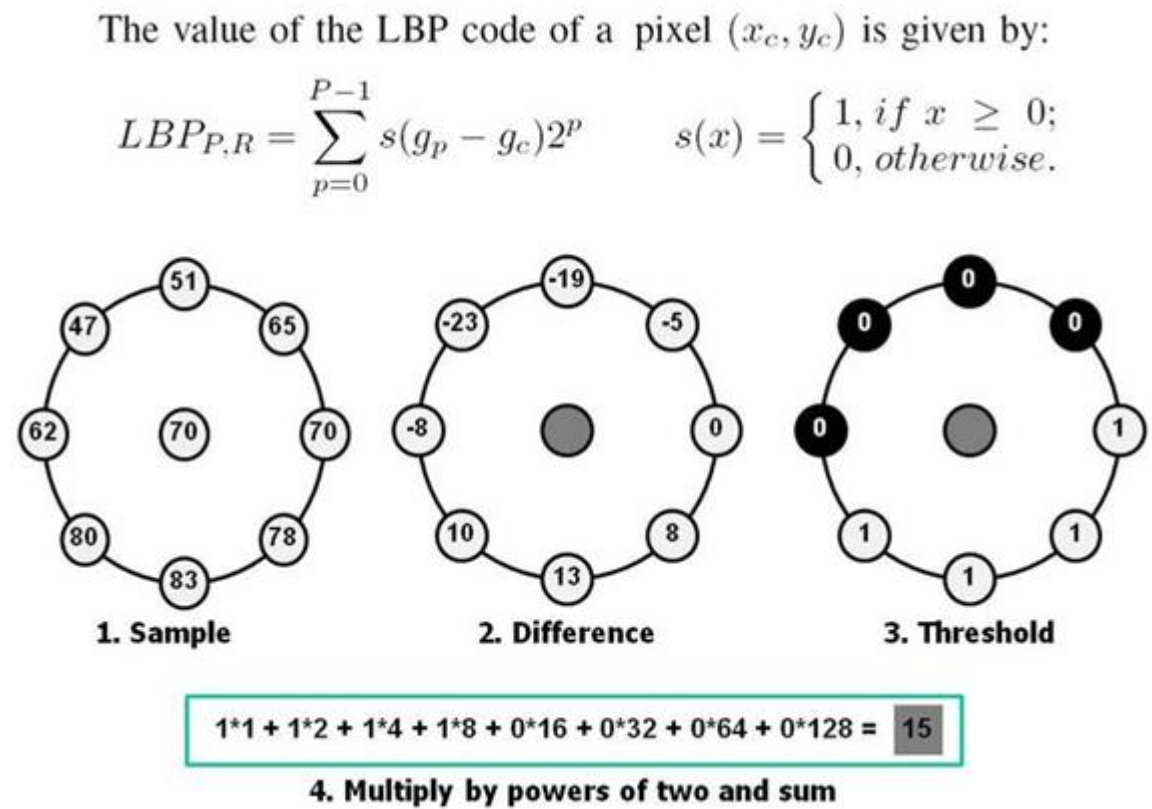


Рисунок 3.6 — Приклад розрахунку LBP

Інше розширення вихідного оператора є визначення так званих єдиних шаблонів, які можуть бути використані для зменшення довжини вектора і здійснювати простим інваріантним дескриптором. Це розширення було викликано тим, що деякі бінарні структури виникають частіше в текстурних зображеннях, ніж в інших. Локальний бінарний шаблон називається однорідним, якщо двійковий код містить не більше двох бітових переходів від 0 до 1 або навпаки, коли бітовий шаблон проходиться по колу. Наприклад, шаблони 00000000 (0 переходів), 01110000 (2 переходи) і 11001111 (2 переходи) є однорідними, тоді як зразки 11001001 (4 переходи) і 01010010 (6

переходів) немає. У розрахунку на LBP позначок, однорідні структури використовуються так, що існує окрема позначка для кожного рівномірного малюнка, і всі неоднорідні зразки позначені однієї мітці. Наприклад, при використанні (8, R) інтерполяції найближчим сусідом, тобто в цілому 256 шаблонів, 58 з яких є однорідним, що дає 59 різних позначок.

Ojala та ін. (2002) зауважив, у своїх експериментах з текстурних зображень, що однакові моделі припадає трохи менше 90% від усіх форм при використанні (8,1) і близько 70% в (16,2) інтерполяції методом найближчого сусіда. Кожен LBP код можна розглядати як мікро-текстон. Локальні примітиви, які кодуються цими кодами включає різні типи вигнутих країв, плями, плоскі ділянки і т.д.

Використовуються такі позначення для оператора LBP: $LB_{P,R}^{U2}$. Індекс використовує оператор (P, R) інтерполяції методом найближчого сусіда. верхній індекс U2 варто використовувати тільки один шаблон та помічати всі інші моделі з однієї позначки. Після маркування LBP зображення $f_l(x, y)$ була отримана гістограма LBP, яка може бути визначена як:

$$H_i = \sum_{x,y} I\{f_i(x, y) = i\}, i = 0, \dots, n-1, \quad (3.4)$$

в якому n — число різних міток, отриманих за допомогою оператора LBP, і $I\{\}$ дорівнює 1, якщо A істинно і 0, якщо A помилково.

Коли ділянки зображення, чії гістограми в порівнянні мають різні розміри, гістограми повинні бути нормалізовані, щоб отримати узгоджений опис:

$$N_i = \frac{H_i}{\sum_{j=0}^{n-1} H_j}. \quad (3.5)$$

					РБ101.17010201.072.007 ПЗ	Арк.
						6 1
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3.1 Просторово-часова LBP

Оригінальний оператор LBP був визначений тільки з просторової інформації. Пізніше, він був продовжений в просторово-часовому понятті для динамічного аналізу текстури. Для цього, так званий Volume Local Binary Pattern (VLBP) оператор був запропонований (Zhao і Pietikäinen 2007). Ідея VLBP складається з поглядом на динамічні текстури, як набір в (X, Y, T) просторі, де X та Y позначимо просторові координати і T позначає індекс кадру (часу). Інтерполяція методом найближчого сусіда для кожного пікселя визначається, таким чином, як в тривимірному просторі. Потім, так само, як LBP в просторової області, обсяг textons можуть бути визначені і отримані гістограми. Таким чином, VLBP поєднує в собі рух і зовнішній вигляд разом, щоб описати динамічні текстури.

Для того, щоб VLBP обчислювати просто і легко, також був введений оператор спільний з LBP з трьома взаємно перпендикулярними площинами (LBP-TOP). LBP-TOP розглядає три ортогональних площі: XY , XT і YT , і об'єднує статистичні дані локального бінарного шаблону в цих трьох напрямках, як показано на рисунку 3.7.

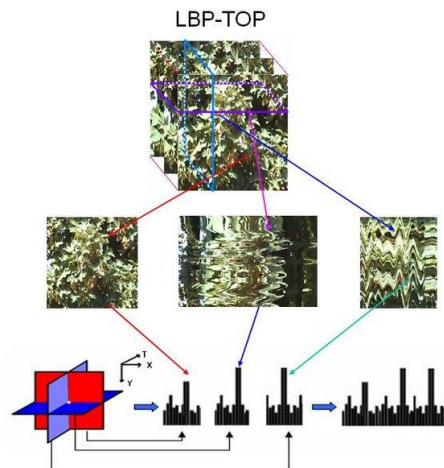


Рисунок 3.7 — Приклад розрахунку LBP

На рисунку 3.7 показаний приклад зображення з трьох площин. XY площину представляє інформацію зовнішній вигляд, в той час як XT дає

					РБ101.17010201.072.007 ПЗ	Арк.
						6 2
Змн.	Арк.	№ докум.	Підпис	Дата		

візуальне враження як змінюються криві по часу і YT описує рух одного стовпця в часовому просторі. У LBP коди витягуються для всіх пікселів XY , XT і YT графіків, що позначаються як XY -LBP, XT -LBP і YT -LBP і гістограми обчислюються і об'єднуються в єдину гістограми. В цілому, радіуси в осей X , Y і T , і кількості сусідніх точок в XY , XT і YT графіки можуть бути різними.

- Transition Local Binary Patterns(tLBP): бінарне значення кодованої LBP складається з порівнянь сусідніх пікселів по годинній стрілці для всіх пікселів, крім центрального;
- Direction coded Local Binary Patterns(dLBP): dLBP кодує зміну інтенсивності вздовж чотирьох основних напрямків через центральний піксель у двох бітах;
- Modified Local Binary Patterns(mLBP): mLBP порівнює значення сусідніх пікселів з середнім значенням інтенсивності у 3×3 пікселях;
- Multi-block LBP: зображення ділиться на безліч блоків, гістограма LBP обчислюється для кожного блоку, і об'єднуються в кінцевій гістограми;
- RGB-LBP: цей оператор отримується шляхом обчислення LBP по всім трьом каналам RGB незалежно один від одного, а потім результати конкатенуються разом.

3.4 Вирівнювання зображень

Вирівнювання (ректифікація) зображень являє собою процес перетворення, який використовується для проектування двох або більше зображень на загальній площині зображення. Це виправляє спотворення зображення, перетворюючи зображення в стандартній системі координат. Він використовується в комп'ютерному стереозору, який спрощує завдання знаходження відповідних точок між зображеннями. Він використовується в географічних інформаційних систем для об'єднання зображень, знятих з різних точок зору в загальну систему координат карти.

					РБ101.17010201.072.007 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

Але в нашому випадку, ректифікація являє собою вирівнювання зображення після вибору 4 точок і процесу гомографії, які будуть описані нижче.

3.3.1 Ректифікація зображень, у яких викривлена перспектива

Зображення перспективні в природі. Сьогоднішній світ вимагає перспективні ректифіковані зображення для додатків, наприклад, таких як image based rendering. Додатки залежать від паралельного огляду зображення для фото реалістичних результатів. Велика робота була виконана в напрямку ректифікації перспективних викривлень, на основі зображення, а також на основі декількох зображень. Останній акцент був зроблений на оновлення на основі одного зображення, має хороший потенціал застосування в зображенні, заснованій на image based rendering, image mosaicing, 3D реконструкції. Перспектива є проекцією реального зору [8].

Багато з додатків необхідна паралельна проекція, тому що паралельна проекція полегшує добування і застосування закономірностей. Однак вони не паралельні в перспективній проекції, отже, їх ідентифікація та застосування має велике значення. Це призводить до ректифікації перспективних спотворень.

3.3.2 Етапи ректифікації зображення

Цей підрозділ представляє собою блок-схему ректифікації зображень для корекції перспективи на зображеннях. Цей етап представлений на рисунку 3.8:

					РБ101.17010201.072.007 ПЗ	Арк.
						6 4
Змн.	Арк.	№ докум.	Підпис	Дата		

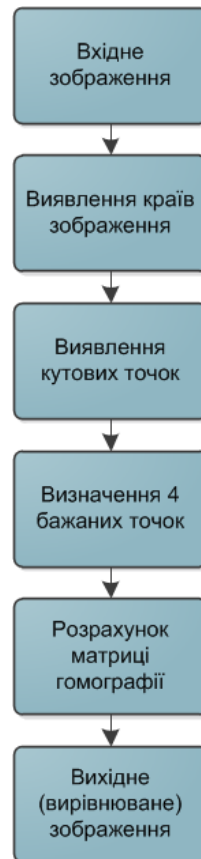


Рисунок 3.8 — Етапи ректифікації зображення

Гомографія — проєктивне перетворення, вид перетворення, використовуваного в проєктивній геометрії. Вона описує, що трапиться з сприймайними положеннями спостережуваних об'єктів, коли змінюється точка зору спостерігача. У більш формальних термінах: гомографія — це оборотне перетворення реальної проєктивної площини на проєктивну площину, що переводить прямі лінії в прямі лінії.

Використовуючи гомогенічні координати, можна уявити гомографію у вигляді матриці 3×3 з вісьмома ступенями свободи.

3.3.3 Проективна площина

Розглянемо проєктивностей простір розмірності 2 — проєктивна площина P^2 . Точка на цій площині представляється у вигляді вектора $m[x, y, w]^T$. Лінія теж представляється в вигляді вектора розмірності 3:

					РБ101.17010201.072.007 ПЗ	Арк.
						6 5
Змн.	Арк.	№ докум.	Підпис	Дата		

$l = [a, b, c]^T$. Точка m належить лінії тоді і тільки тоді, коли виконується рівняння $l^T m = 0$. Це рівняння також означає, що лінія l проходить через точку m . Можна говорити про дуалізм точок і прямих, зокрема про те, що «перетином» точок є пряма. Вектор прямої представимо у вигляді векторного добутку двох точок:

$$l \propto [m_1]_{\times} m_2; \quad [m_1]_{\times} = \begin{pmatrix} 0 & w_1 & -y_1 \\ -w_1 & 0 & x_1 \\ y_1 & -x_1 & 0 \end{pmatrix}. \quad (3.6)$$

У даному просторі точки виду $m[x, y, 0]^T$ є точками на нескінченності або нескінченно віддаленими точками. Вони визначають лінію на нескінченності, яка представлена у вигляді вектора $l_{\infty} = [0, 0, 1]^T$. Всі нескінченно віддалені точки належать їй:

$$l_{\infty}^T \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = 0 \times x + 0 \times y + 1 \times 0 = 0. \quad (3.7)$$

3.3.4 Паралельні прямі в проектному просторі

Паралельні прямі в проектному просторі перетинаються в нескінченно віддалених точках. Покажемо, що це так.

Для початку розглянемо співвідношення між нескінченно віддаленою точкою в декартових координатах і точкою виду $[x, y, 0]^T$ в однорідних координатах. Нехай у декартовій системі задана лінія, що проходить через початок координат і точку (X_d, Y_d) . Однорідні координати цієї точки будуть мати вигляд $(x, y, v)^T = (v X_d, v Y_d, v)^T$, де v є довільне ненульове число.

					РБ101.17010201.072.007 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

Межа відносин $\lim_{v \rightarrow 0} \frac{x}{y} = \frac{X_d}{Y_d}$, але так як $X_d = \frac{x}{v}, Y_d = \frac{y}{v}$, то декартові координати прагнуть до нескінченності $X_d \xrightarrow{v \rightarrow 0} \infty, Y_d \xrightarrow{v \rightarrow 0} \infty$. Отже, точка з однорідними координатами $[x, y, 0]^T$ задає в декартовій системі нескінченно віддалену точку на розглянутій прямій.

Дві пересічні прямі в декартовій системі координат задані системою рівнянь:

$$\begin{cases} a_1x + b_1y + c_1 = 0; \\ a_2x + b_2y + c_2 = 0. \end{cases} \quad (3.8)$$

Вирішуючи цю систему відносно x і y , знайдемо координати точки перетину:

$$x_{do} = \frac{c_1b_2 - c_2b_1}{a_1b_2 - a_2b_1}; y_{do} = \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1}, \quad (3.9)$$

які в однорідних координатах запишуться як $\left(\frac{c_1b_2 - c_2b_1}{a_1b_2 - a_2b_1}, \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1}, 1 \right)$

В силу довільності масштабного множника помножимо значення координат на $a_1b_2 - a_2b_1$, тоді, якщо позначити, $x_o = c_1b_2 - c_2b_1$, а $y_o = a_1c_2 - a_2c_1$ отримаємо $(x_o, y_o, a_1b_2 - a_2b_1)$.

В силу довільності масштабного множника помножимо значення координат на $a_1b_2 - a_2b_1$, тоді, якщо позначити $(x_o, y_o, 0)$.

3.3.5 Властивості перетворень в проектному просторі P^2

Взагалі кажучи, більшість перетворень над координатним простором зображень представлені гомографіями, а саме перетвореннями виду $P^2 \mapsto P^2$. Без обмеження спільності в даній роботі гомографію, або площинно-

					РБ101.17010201.072.007 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

проекційним перетворенням, будемо називати те перетворення, яке вирішує поставлене завдання. Іншими словами, площинно-проекційним називається перетворення, яке дозволяє з вихідного зображення об'єкта, знятого в перспективі, отримати зображення, на якому площину камери і деяка певна площину об'єкта паралельні [8].

Площинно-проекційне перетворення в просторі P^2 позначається H і представимо у вигляді матриці розмірністю 3×3 . Геометричні об'єкти відображаються за допомогою H по-різному. Точки з визначення відображаються шляхом добутку матриці перетворення на вектор координат: $\bar{w} = Hw$ (рисунок 3.9).

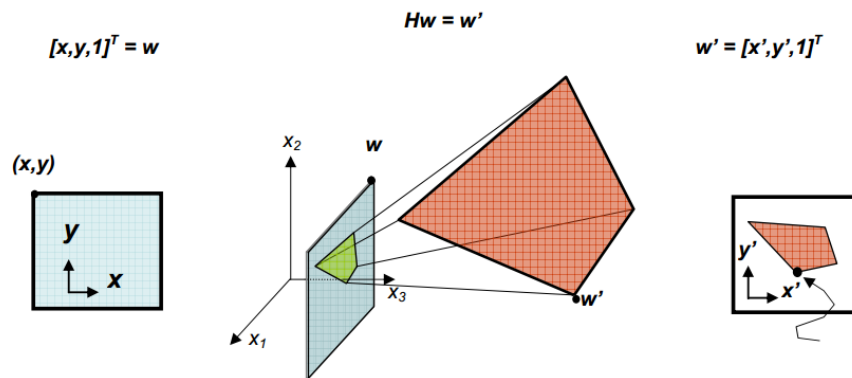


Рисунок 3.9 — Загальна схема застосування перетворення до зображення

Перетворення прямих може бути знайдено таким чином:

$$\begin{aligned} \bar{l}^T \bar{w} = 0 &= l^T w = l^T H^{-1} H w = l^T H^{-1} \bar{w}; \\ \bar{l}^T &= l^T H^{-1} \Rightarrow \bar{l} = H^{-T} l. \end{aligned} \quad (3.10)$$

3.3.6 Розкладання безлічі перетворень

Позначимо через T_P безліч всіх проектних перетворень, діючих над геометричними сутностями в проектному просторі P^2 , через T_A — всі афінні перетворення, а через T_M — всі метричні перетворення. Тоді отримаємо послідовність вкладених множин $T_M \subset T_A \subset T_P$. Важливим аспектом цих

					РБ101.17010201.072.007 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

множин є їх інваріанти, що і послужило основним критерієм для розбиття. Інваріантом в даному випадку є деяка властивість в конфігурації геометричної сутності, яка не змінюється, принаймні в цілому, під дією перетворення, що належить певній множині [9].

Афінні перетворення з множини T_A представимо у вигляді матриці:

$$T_A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.11)$$

За визначенням все інваріанти T_P є інваріантами і для T_A . Новим інваріантом для T_A є паралельність ліній. Зокрема, як уже зазначалося, паралельними є лінії, які перетинаються в одній точці на нескінченності. Ще одним новим інваріантом є відношення довжин вздовж певного напрямку. Слід зазначити, що афінні перетворення не змінюють лінію на нескінченності l_∞ . Хоча положення ідеальних точок може змінюватися, але вони все одно будуть належати l_∞ .

Метричні перетворення представлені наступним чином:

$$T_M = \begin{pmatrix} sR_{2 \times 2} & T_{2 \times 1} \\ 0_{1 \times 2} & 1 \end{pmatrix}, \quad (3.12)$$

де R — ортонормирована матриця. Зауважимо, що R є матрицею обертань, коли $RR^T = I$ і $\det(R) = 1$. Таким чином, T_M визначає орієнтацію, зсув і масштабування. Інваріанти цієї множини — відносні довжини і кути. Метричні перетворення також відображають спеціальну (абсолютну) форму в собі. Ця геометрична сутність ще більш абстрактна, ніж площа або лінія на

					РБ101.17010201.072.007 ПЗ	Арк.
						6 9
Змн.	Арк.	№ докум.	Підпис	Дата		

нескінченності. Її можна представити у вигляді кола в уявній частини комплексної площині лінії на нескінченності.

На основі розкладу, описаного вище, логічно площинно-проекційне перетворення представляти у вигляді композиції трьох основних перетворень з кожної групи: чистого проекційного, афінного і метричного. Чисте проекційне перетворення відтворить паралельність ліній на зображенні. Афінне перетворення дозволить відновити значення кутів і відношення довжин для непаралельних сегментів. Перетворення ізометрії і подоби забезпечує орієнтацію, положення і відносне масштабування. Відповідно матриця H розкладається в добуток декількох матриць, наприклад трьох:

$$H = H_s H_a H_p = \begin{bmatrix} s \cos \phi & -s \sin \phi & t_x \\ s \sin \phi & s \cos \phi & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ v_1 & v_2 & v_3 \end{bmatrix}. \quad (3.13)$$

3.3.7 Метод зникаючих точок для обчислення матриці H

Розв'язувана задача полягає в наступному: є зображення, на якому деякий об'єкт, відображений в перспективі — необхідно отримати нове зображення, на якому площину камери (місця, з якого виробляється зйомка) і деяка певна площина об'єкта паралельні. Для досягнення результату треба відновити паралельність ліній в заданій площині об'єкта, відтворити значення кутів, зокрема ортогональність основних напрямків, і відносні довжини непаралельних сегментів. Нове зображення виходить шляхом перерозподілу пікселів вихідного зображення, а значить шляхом застосування деякого перетворення над зображенням [10].

Метод, описаний тут, ще відомий як «гомографічне перетворення». Наведемо загальну схему застосування даного методу для визначення площинно-проекційного перетворення.

					РБ101.17010201.072.007 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Для початку задамо на вихідному зображенні дві пари ліній, які повинні бути паралельні в необхідному зображенні, причому без обмеження спільності вважаємо, що перша пара ліній відповідає горизонтальному напрямку, а друга — вертикальному.

2. Знайдемо точки перетину кожної пари ліній. Перша пара дасть точку p_1 , а друга — точку p_2 . Ці дві точки називаються зникаючими.

3. Обчислимо лінію l_h , яка з'єднає p_1 та p_2 (рисунок 3.10). Аналогічно ця лінія називається зникаючою.

3. Прямі у кожній із заданих пар повинні стати паралельними (рисунок 2.3), тоді точки їх перетину повинні відобразитися в точки на нескінченності, а лінія l_h відобразиться в лінію на нескінченності, тобто зникне. Іншими словами, $l_\infty = H^T l_h$, тоді, якщо l_h представлена у вигляді вектора $[l_{h1}, l_{h2}, l_{h3}]^T$, то:

$$H_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_{h1} & l_{h2} & l_{h3} \end{bmatrix}. \quad (3.14)$$

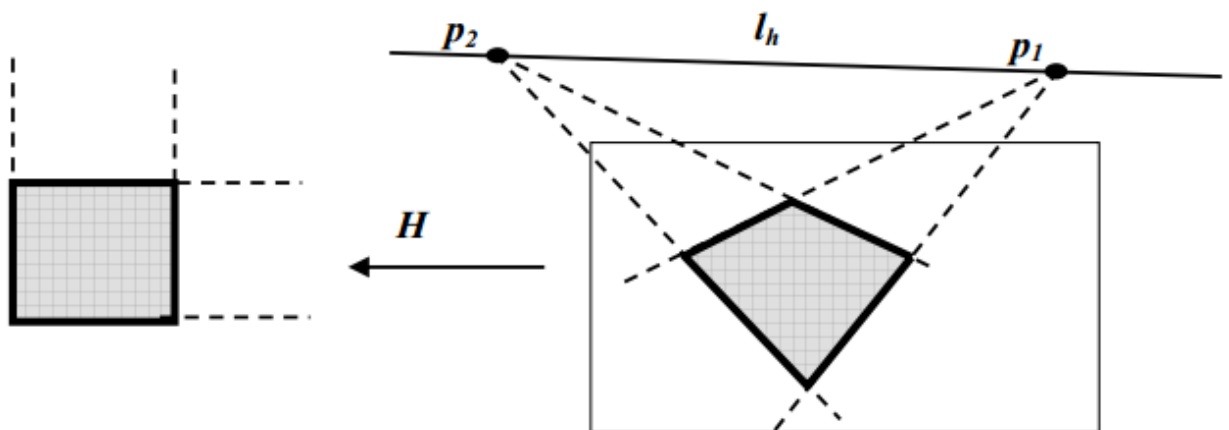


Рисунок 3.10 — Загальна схема дії методу зникаючих точок

5. Виправимо зникаючі точки p_1 та p_2 , використовуючи $H_p : p_{1A} = H_p p_1$, $p_{2A} = H_p p_2 * p_{1A}$; де p_{2A} є ідеальними точками, і представляють відповідні напрямки.

6. Повернемо зображення так, щоб p_{1A} було вирівняно відносно горизонтальної осі ($R(\varphi)$) є матриця повороту на кут φ).

7. Кут між напрямками p_{1A} та p_{2A} позначимо θ . Застосуємо афінне перетворення, яке відновлює метричну геометрію, зокрема ортогональність напрямків:

$$H_A = \begin{bmatrix} 1 & -\cot(\theta) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.15)$$

8. Отримаємо результуюче перетворення $H = H_A R H_p$



Рисунок 3.11 — Оригінальна фотографія для подальшого процесу ректифікації

					РБ101.17010201.072.007 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.12 — Зображення після процесу ректифікації

3.3.8 Результати алгоритму вирівнювання

Програма, написана у середовищі Matlab (з використанням 3-х функцій — `homography`, `normalize` і `rectify`), яка оцінює гомографію на площині (або викривленої перспективи) і деформує зображення (див. ДОДАТОК А). Для використання треба поетапно вибирати точки на зображенні строго зліва направо, знизу вгору.

Для зйомки використовувався фотоапарат Canon 1000d з КІТ-об'єктивом, із роздільною здатністю 3888x2592. На рисунку 3.7 представлена оригінальна фотографія автомобілю. На рисунках 3.13-3.16 представлено поетапне виділення точок для подальшої ректифікації зображення. Як видно, на рисунку 3.17 — ректифікація пройшла успішно, номерні знаки випрямлені.

					РБ101.17010201.072.007 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.13 — Оригінальне зображення



Рисунок 3.14 — Частка фотографії з 1 виділеною точкою



Рисунок 3.15 — Частка фотографії з 2 виділеною точкою

					РБ101.17010201.072.007 ПЗ	Арк.
						7 4
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.16 — Частка фотографії з 3 виділеною точкою



Рисунок 3.17 — Результат ректифікації зображення

Далі представимо результати на інших зображеннях 3.18-3.22.



Рисунок 3.18 — Результати роботи алгоритму ректифікації

					РБ101.17010201.072.007 ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.19 — Результаты работы алгоритму ректификації



Рисунок 3.20 — Результаты работы алгоритму ректификації

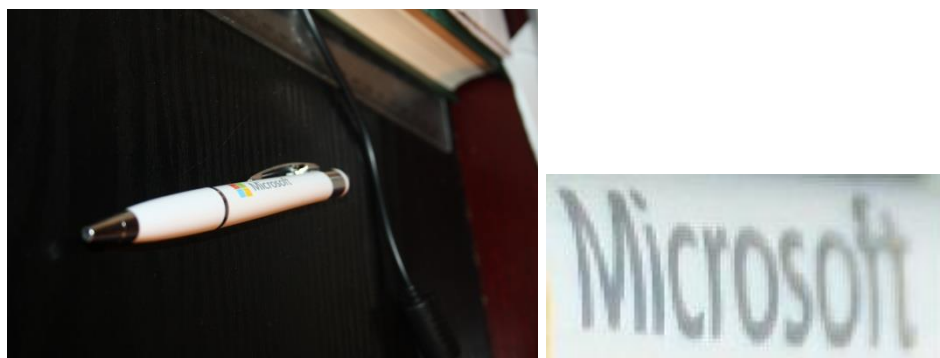


Рисунок 3.21 — Результаты работы алгоритму ректификації

					РБ101.17010201.072.007 ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.22 — Результати роботи алгоритму ректифікації

3.5 Сегментація символів номерного знака

Перед застосуванням OCR окремі символи в тексті розділяються за допомогою процесу поділу їх посимвольно [17-22].

Кроки для сегментації символів представлені в пунктах 3.5.1 та 3.5.2.

3.5.1. Бінарзація зображення платини

Процедури сегментації та розпізнавання працюють з бінарним зображенням, тобто тільки з чорними і білими пікселями. Тому перш ніж передавати роботу цих процедур, потрібно вихідне кольорове зображення привести до бінарного вигляду. Це завдання вирішується в два етапи.

На першому етапі кольорове зображення перетворюється в чорно-біле і представляється в градаціях сірого (рисунок 3.23). Для кожного пікселя обчислюється його яскравість в межах від нуля до 255. Рівню яскравості 0 відповідає чорний колір, рівню 255 — білий. Таким чином, для зберігання зображення припадає один байт на піксель. Яскравість пікселя обчислюється по одній з наступних формул:

$$I = \frac{R + G + B}{3}. \quad (3.16)$$

$$I = \frac{\max(R + G + B) + \min(R + G + B)}{2}. \quad (3.17)$$

					РБ101.17010201.072.007 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

де R, G, B — нормований на 256 (один байт) червоний, зелений та блакитний компонент кольору пікселя відповідно.



Рисунок 3.23 — а) кольорове зображення, б) чорно-біле зображення

Другим етапом є власне бінаризація. Результат бінаризації залежить від заздалегідь заданого параметра — співвідношення чорних пікселів і загальної їх кількості на зображенні. Далі наводиться докладний опис процедури бінаризації [23]. Приклади бінаризації зображення з різними параметрами наведені на рисунку 3.24.



Рисунок 3.24 — Бінаризація, а) вихідне зображення, б) 25% чорних пікселів, в) 50% чорних пікселів, г) 75% чорних пікселів

					РБ101.17010201.072.007 ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

Алгоритм бінарізації зображення [24].

1. Створюємо одновимірний масив I з 256 елементів (від 0 до 255). Заповнюємо його нулями.

2. Пробігаємо попіксельно все зображення. Збільшуємо на одиницю значення в комірці масиву I , відповідної яскравості пікселя $i(I[i]++)$. У підсумку, значення кожного осередку масиву $I[i]$ буде дорівнює кількості пікселів яскравості рівня i на всьому зображенні.

3. На цьому кроці визначається поріг яскравості a . Нехай N — загальна кількість пікселів (висота, помножена на ширину), k — коефіцієнт, що визначає кількість чорних пікселів. Тоді kN дорівнюватиме бажаній кількості чорних пікселів на бінарному зображенні. Підсумовуємо значення комірок масиву, починаючи з нульової до тих пір, поки значення цієї суми не перевищить kN . Індекс останньої сумовної комірки і буде порогом a .

4. Повторно пробігаємо попіксельно все зображення. Порівнюємо рівень яскравості кожного пікселя з порогом a . Якщо цей рівень менше або дорівнює a , то піксель стає чорним, інакше — білим.

Зображення піддається фільтрації. Фільтр підкреслення країв накладається на кожен піксель зображення, обчислюється сума добутків значень матриці і рівнів яскравості зображення.

Отримане число є яскравістю відповідного пікселя в новому зображенні (рисунок 3.25).

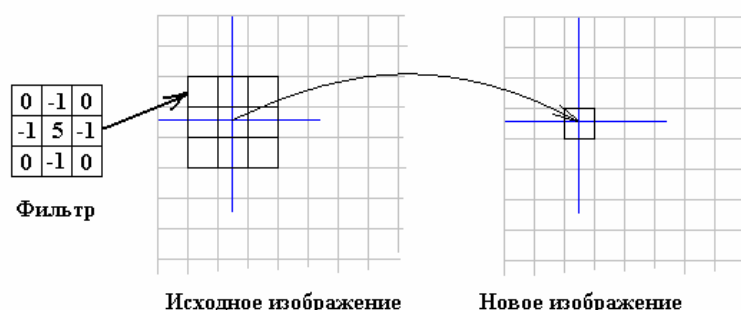
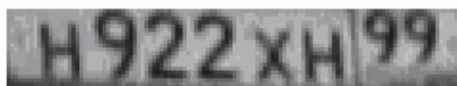


Рисунок 3.25 — Фільтрація зображення

Приклад фільтрації наведений на рисунку 3.26:



а)



б)

Рисунок 3.26 — а) вихідне зображення, б) оброблене фільтром підкреслення країв зображення

Профільтроване зображення піддається процедурі бінаризації (рисунок 3.27) і передається на сегментацію.



Рисунок 3.27 — Зображення, готове до сегментації

3.5.2. Вертикальна проекція для сегментації стовпців

Розглянемо обрізане зображення зліва направо по стовпцях після точного місця розташування верхньої та нижньої межі і підрахуємо загальну кількість чорних крапок в кожному стовпці [25-26]. Порогове значення встановлено в $h/10$. Визначимо кожне значення в масиві проекції. Якщо проекція $[i]$ більше, ніж $h/10$, проекція $[i]$ дорівнює одному. В іншому випадку, проекція $[i]$ встановлюється в нуль. Де h не модифікує рядок на бінарному зображенні після точної локалізації верхньої і нижньої меж. Потім символи обрізаються шляхом вибору частин, що мають проекцію $[i] = 1$.

Перед алгоритмом розпізнавання символи повинні бути уточнені в блок, який не містить додаткові пробіли (пікселі) з усіх чотирьох сторін символів.

					РБ101.17010201.072.007 ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.28 — Сегментація символів



Рисунок 3.29 — Символи після видалення зайвих пробілів з чотирьох сторін

Алгоритм сегментації [27-29].

1. Знайти рядок заголовка, знаходячи горизонтальну проекцію слова, тобто пошук рядків з максимальними чорними пікселями.
2. Знайти ширину заголовка.
3. Видалити рядок заголовка. Після видалення заголовка наше слово ділиться на три горизонтальні частини, відомі як верхня зона, середня зона і нижня зона.
3. Застосувати вертикальну проекцію після видалення рядка заголовка.
5. Витягти символи як у попередньому алгоритмі.

3.6 Оптичне розпізнавання символів (OCR)

Детектування об'єкта за шаблоном може стати в нагоді у безлічі випадків. Найпростіший приклад — пошук заздалегідь заданого об'єкта.

Оптичне розпізнавання символів (англ. optical character recognition, OCR) — механічний або електронний переведення зображень рукописного, машинописного або друкованого тексту в текстові дані — послідовність кодів, що використовуються для представлення символів в комп'ютері (наприклад, в текстовому редакторі). Розпізнавання широко використовується для

					РБ101.17010201.072.007 ПЗ	Арк.
						8 1
Змн.	Арк.	№ докум.	Підпис	Дата		

конвертації книг і документів в електронний вигляд, для автоматизації систем обліку в бізнесі або для публікації тексту на веб-сторінці. Оптичне розпізнавання тексту дозволяє редагувати текст, здійснювати пошук слова чи фрази, зберігати його в більш компактній формі, демонструвати або роздруковувати матеріал, не втрачаючи якості, аналізувати інформацію, а також застосовувати до тексту електронний переказ, форматування або перетворення в мову. Оптичне розпізнавання тексту є досліджуваної проблемою в областях розпізнавання образів, штучного інтелекту та комп'ютерного зору.

Кроки алгоритму OCR [30-31].

1. Вже сегментували номерний знак усередині функції OCR (рисунок 3.30, а).

2. Це зображення перетворюється у відтінки сірого для розрахунку порога (рисунок 3.30, б).

3. За допомогою порога зображення перетворюється в бінарне зображенні (рисунок 3.30, в).

3. Всі регіони сегментовані.

5. Середнє число чорних пікселів кожного регіону розраховується як поріг.

6. Якщо кількість чорних пікселів на одиницю площі менше, ніж порогове значення (середнє), то він буде видалений.

7. Всі регіони, в яких висота менше, ніж ширина ($\times 10\%$) видаляються. Остаточний результат показаний на рисунку 3.30, г.

8. Нарешті, ми порівнюємо всі залишилися регіони з кожною буквою в базі (рисунок 3.30).

Це порівняння виконується за допомогою методу кореляції, описаного нижче.

					РБ101.17010201.072.007 ПЗ	Арк.
						8 2
Змн.	Арк.	№ докум.	Підпис	Дата		

Порівняння за допомогою БД, у якій німецькі номерні символи, тому що саме вона була знайдена у відкритому доступі.

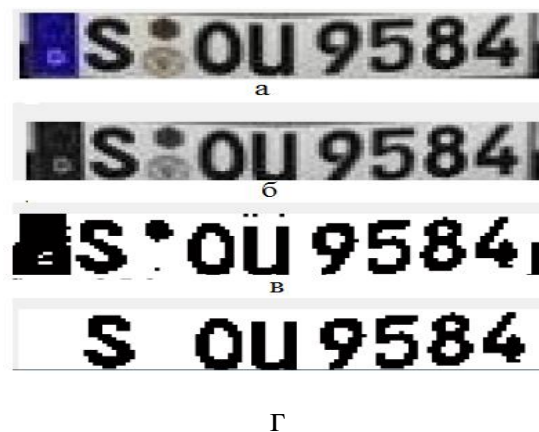


Рисунок 3.30 — Сегментація та виконання алгоритму OCR

OCR в даний час використовується для порівняння кожного окремого символу з повною буквено-цифровий базою даних, використовуючи відповідність шаблону. Шаблон відповідності є одним з методів оптичного розпізнавання символів. OCR фактично використовує метод кореляції, щоб відповідати окремому символу.

Зображення перетворюється у бітовий масив 12x12. Бітовий масив представлений 12x12-матрицею або векторами 144 з 0 і 1 координатами, рисунок 3.31.

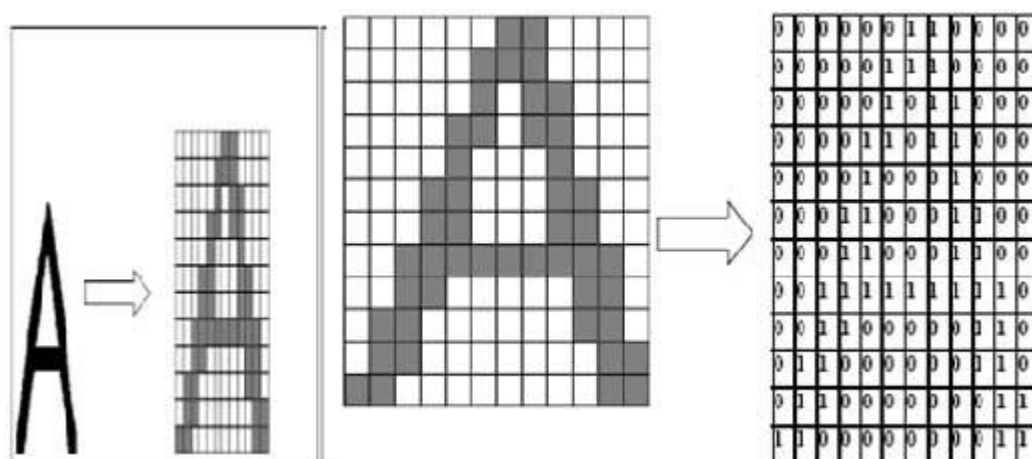


Рисунок 3.31 — Бітовий масив представлений матрицею 12x12

3.7 Висновки

Отже, що фаза детекції відбувається один раз для кожного вхідного зображення. Фаза детекції використовує алгоритм LBP (як правило, використовується для розпізнавання обличчя), щоб знайти можливі області номерного знака (X , Y , ширина, висота). Фаза, як правило, займає більшу частину обробки. GPU (графічний процесор) може бути прискорене для підвищення продуктивності.

Фаза бінаризації створює кілька бінарних зображень для кожного регіону пластини. Причина кількох бінарних зображень використовується, щоб дати нам найкращу ймовірність знайти всі символи. Перетворення в бінарну форму за допомогою методу Sauvola з різними параметрами. Кожне з бінарних зображень обробляється на наступних етапах.

Аналіз символів ґрунтується на знаходженні символів в номерному знаці. Спочатку знаходить всі підключені участки в регіоні номерного знака. Тоді він шукає участок, ширину і висоту номерного знака яких приблизно і має вершини і низи, які знаходяться в прямій лінії з іншими участками аналогічної ширини / висоти. Цей аналіз робиться кілька разів в регіоні. Вона починається з пошуку дрібних символів, а потім поступово працює і для великих символів. Якщо нічого не знайдено в регіоні, то регіон викидається і подальшої обробки не відбувається.

Наступний етап полягає в знаходженні краю номерного знака. Майте на увазі, що фаза виявлення відповідальна тільки для ідентифікації можливих регіонів, в якому номерний знак може існувати. Край пластини знаходиться точно (всі края зверху до низу, з ліва до права номерного знака).

Сегментація символів використовує вертикальну гистограму, щоб знайти пробіли між символами знаку.

Постобробка також обробляє перевірену область, якщо потрібно. Наприклад, якщо вказати OpenALPR, що це "Міссурі" пластина, то він

					РБ101.17010201.072.007 ПЗ	Арк.
						8 4
Змн.	Арк.	№ докум.	Підпис	Дата		

намагатиметься і відповідати результатам порівнянню шаблону, який відповідає формату Міссурі (наприклад, [символ][символ][число] - [символ][число][символ]). Так, наприклад, список топ 3 буде: CFOCIG, CF0CIG, CF0C1G. Третій результат відповідає шаблону, проте як два інших не відповідають. Таким чином, наступна обробка буде означати, що третя запис є найкраще збігається.

					РБ101.17010201.072.007 ПЗ	Арк.
						8 5
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ПРАКТИЧНІ РЕЗУЛЬТАТИ

4.1 Загальні відомості

Після компіляції всіх бібліотек, та їх конфігурації, почнемо проектування сайту для подальшого перегляду вихідної інформації. Для цього необхідно для початку дати визначення таким поняттям, як Front-end, Back-end, використання фреймворків для створення веб-сторінок і т.д.

4.1.1 Пояснення Front-end та Back-end

Front-end (фронтенд) і back-end (бекенд) використовуються в багатьох сферах і галузях, проте ми опишемо саме про ІТ, або, навіть більш конкретно, про Web-розробку.

Говорячи про front-end і back-end, програмісти зазвичай мають на увазі поділ інтерфейсної частини користувача від програмної логіки. У Web-розробці, наприклад, в якості front-end виступають HTML-верстка, стилі CSS і JavaScript, а в якості back-end — серверна частина, яку зазвичай програмують на PHP або ASP.net.

У середовищі розробників високонавантажених систем (highload-розробників) терміном front-end називають ту програмну частину, яка безпосередньо «віддає» контент. Наприклад, на великих проектах часто програмну серверну частину представляють 2 веб-сервера — Apache і nginx. Nginx приймає запити і, у разі статичного файлу, (зображення, файл css, js або xml) відразу ж віддає його вміст, а в разі PHP-скрипта, відправляє його до сервера Apache, який вже вміє обробляти PHP. Тут nginx — це front-end, а Apache — back-end. Звичайно, високонавантажені системи мають складну інфраструктуру, і часом представляють собою багато серверів, рознесених по різних континентах.

					РБ101.17010201.072.007 ПЗ	Арк.
						8 6
Змн.	Арк.	№ докум.	Підпис	Дата		

Також, коли говорять про CMS, адміністративну частину називають back-end, а «лицьову» частину сайту — front-end.

4.2 Twitter Bootstrap

Для front-end частини був обраний фреймворк Twitter Bootstrap.

Twitter Bootstrap — вільний набір інструментів для створення сайтів і веб-інтерфейсів. Включає в себе HTML і CSS шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсів, включаючи JavaScript розширення.

Bootstrap використовує найсучасніші напрацювання в галузі CSS і HTML, тому необхідно бути уважним при підтримці старих браузерів.

Основні переваги Twitter Bootstrap:

- Економія часу — Bootstrap дозволяє заощадити час і зусилля, використовуючи шаблони дизайну і класи, і сконцентруватися на інших розробках;
- Висока швидкість — динамічні макети Bootstrap масштабуються на різні пристрої і дозволу екрану без будь-яких змін в розмітці;
- Гармонійний дизайн — всі компоненти платформи Bootstrap використовують єдиний стиль і шаблони за допомогою центральної бібліотеки. Дизайн і макети веб-сторінок узгоджуються один з одним;
- Простота у використанні — платформа проста у використанні, користувач з базовими знаннями HTML і CSS може почати розробку з Twitter Bootstrap;
- Сумісність з браузерами — Twitter Bootstrap сумісний з Mozilla Firefox, Google Chrome, Safari, Internet Explorer і Opera;
- Відкрите програмне забезпечення — особливість Twitter Bootstrap, яка передбачає зручність використання, за допомогою відкритості вихідних кодів і безкоштовного завантаження.

					РБ101.17010201.072.007 ПЗ	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні інструменти Bootstrap:

- Сітки — наперед задані розміри колонок, які можна відразу ж використовувати, наприклад ширина колонки 140px відноситься до класу .span2, який можна використовувати в CSS опису документа;
- Шаблони — Фіксований або гумовий шаблон документа;
- Типографіка — Описи шрифтів, визначення деяких класів для шрифтів, таких як код, цитати і т. п.;
- Медіа — Представляє деяке управління зображеннями і відео;
- Таблиці — Засоби оформлення таблиць, аж до додавання функціональності сортування;
- Форми — Класи для оформлення форм і деяких подій, що відбуваються з ними;
- Навігація — Класи оформлення для табів, вкладок, сторінковому, меню і панелі інструментів;
- Алерти — Оформлення діалогових вікон, підказок і спливаючих вікон.

Сьогодні Twitter Bootstrap є одним з найпопулярніших html/css фреймворків. Цей інструмент дозволяє легко і швидко зверстати нескладний сайт, маючи мінімальні знання в області верстки html-сторінок. Однак, незважаючи на всю привабливість можливостей Bootstrap'a, далеко не всі розробники і ті, кому його можливості могли б стати в нагоді, вміють користуватися цією технологією.

По суті, Bootstrap містить готові стилі для оформлення основних елементів html, в тому числі кнопок, таблиць, графіків, картинок і відео, форм, елементів навігації, алертів, стилі для заголовків, списків, цитат, визначень, кодів, а також самі базові і часто використовувані динамічні елементи на javascript, такі як нескладний слайдер, що випадають списки, вкладені меню, лайтбокс, пагінацію і т.д. Цей фреймворк дозволяє створювати первинну структуру і стилізувати на базовому рівні більшість елементів. Тим, кому потрібний найпростіший сайт, для верстки, можливо, буде достатньо тільки

					РБ101.17010201.072.007 ПЗ	Арк.
						8 8
Змн.	Арк.	№ докум.	Підпис	Дата		

знання html і того, що робить кожен з класів Bootstrap — часто про це можна здогадатися виходячи вже з їх назви.

Величезний плюс до швидкості — це спосіб побудови сторінок, сама структура сайту, заснована на 12-колоночній grid системі, інакше кажучи — модульної сітці. Дизайн-макет повинен бути розкреслений на 12 вертикальних смужок, кожна з яких традиційно займає 60px і володіє відступом зліва в 20px. Блоки на дизайн-макеті, відповідно, орієнтуються на ці колонки — тобто займають відстань від початку однієї колонки до кінця однієї з наступних. Відповідно, один блок може займати від однієї до 12-ти колонок. По факту, в css можливі 4 варіанти розміру колонок, залежно від того, який сайт необхідно отримати — фіксований, гумовий, або адаптивний. Файл *bootstrap.css* відповідає за розмітку сторінки під монітори комп'ютера. Тут містяться стилі для фіксованих і гумових блоків. Файл *bootstrap-responsive.css* дозволяє зробити верстку адаптивної, тут також є можливість створення гумових і фіксованих блоків.

Для того щоб почати роботу, вам достатньо завантажити файли Bootstrap на офіційному сайті, створити html-сторінку, імпортувати туди css і javascript файли фреймворку і підключити свій стильовий файл, який буде розміщуватися нижче файлу *bootstrap.css* в списку підключень.

Після того, як створили стильовий файл, переглянемо файл *bootstrap.css*, щоб приблизно розуміти, що він може робити і які класи потрібно задіяти при верстці. Структура побудови блоків буде приблизно такою:

```
<span class="nottranslate" onmouseover="_tipon(this)"
onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr;
text-align: left"><div class="container"></span> <Div class =
"container"></span>
<span class="nottranslate" onmouseover="_tipon(this)"
onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr;
text-align: left">      <div class="row"></span> <Div class = "row"></span>
<span class="nottranslate" onmouseover="_tipon(this)"
onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr;
text-align: left">      <div class="span7"> </div></span> <Div class =
"span7"> </ div></span>
```

					РБ101.17010201.072.007 ПЗ	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<span class="nottranslate" onmouseover="_tipon(this)"
onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr;
text-align: left">      <div class="span2"> </div></span> <Div class =
"span2"> </ div></span>
<span class="nottranslate" onmouseover="_tipon(this)"
onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr;
text-align: left">      </div></span> </ Div></span>
<span class="nottranslate" onmouseover="_tipon(this)"
onmouseout="_tipoff()"><span class="google

```

де *.container* — це загальний блок, *.row* — один ряд блоків, а *.span ** — самі блоки, що займають кількість колонок, відповідне зазначеної після *.span* цифрі. Так як колонок всього 12, класів *.span ** теж 12. *.span1* дорівнює ширині однієї колонки, тобто 60px, *.span2* дорівнює двом колонкам, т.е. 140px (60px + 60px + 20px (відступ зліва)), *.span3* = 220px (60px + 60px + 60px + 20px + 20px) і т.д. до *.span12* = 940px (60 * 12 + 20 * 11).

Так як кількість колонок дорівнює 12-ти, сума цифр усередині одного ряду (першого рівня) не може перевищувати цього числа. Наприклад, в одному *div.row* можуть бути *span10* і *span2* (10 + 2 = 12), або *span6* і *span4* (6 + 4 = 10, відповідно відстань в 2 колонки (140px) залишиться вільним) і т.д. Наприклад, не можна вкласти в *.span5* (рівний 380px) елементи з класами *.span4* (300px) і *.span3* (220px), так як вони просто не помістяться в батьківському блоці.

Відступи задаються за допомогою додавання класів *.offset1* - *.offset11*. Додавання цього класу диву ставить йому відступ зліва, дорівнює кількості колонок, що передують цьому диву. Тобто якщо блок, який необхідно відсунути, відокремлюють від попереднього блоку або від початку контейнера 4 колонки, то цього блоку необхідно прописати *class = "offset4"*.

Фреймворк Bootstrap використовує найновіші технології верстки, найостанніші розробки. Тому необхідно, щоб в *head* нашого документа обов'язково був вказаний доктайп *html5* - *<!DOCTYPE html>*. Twitter Bootstrap підтримується Internet Explorer, починаючи з 7-ї версії.

					РБ101.17010201.072.007 ПЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

Найшвидший спосіб почати роботу: отримати компілюють і мінімізовані версії наших CSS, JS, і зображень. Ніяких документів або вихідних файлів. Після завантаження маємо таку структуру:

```
bootstrap/
├─ css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   └── bootstrap-theme.min.css
├─ js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└─ fonts/
    ├── glyphs-halflings-regular.eot
    ├── glyphs-halflings-regular.svg
    ├── glyphs-halflings-regular.ttf
    ├── glyphs-halflings-regular.woff
    └── glyphs-halflings-regular.woff2
```

Рисунок 4.1 — Структура Twitter Bootstrap

Це основна форма Bootstrap: компілюють файли для швидкого і легкого використання майже в будь-якому Web-проекті. Надаються компільовані CSS і JS (*bootstrap.**), і також компілюють і мінімізовані CSS і JS (*bootstrap.min.**). Файли зображень стиснуті за допомогою ImageOptim, додатки Mac для стиснення зображень в PNG. Всі плагіни JavaScript вимагають jQuery.

Основна сітка макета складається з 12 колонок, забезпечуючи ширину контейнерів в 940px без активних динамічних особливостей. При додаванні динамічного файлу CSS сітка може розтягуватися в ширину від 724px до 1170px, залежно від вашого монітора. На дисплеях, ширина яких менше 767px, колонки стають плаваючими і вибудовуються вертикально.

					РБ101.17010201.072.007 ПЗ	Арк.
						9 1
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 4.2 — Сітка макета

Для простого макета в дві колонки створюється клас `.row` і додається відповідне число колонок `.span *`. Так як це сітка в 12 колонок, кожен діапазон з двох `.span *` налічує 12 колонок і щоразу буде додавати 12 колонок в кожен рядок (або ту кількість колонок, яке задано в батьківському елементі).

```
<div class="row">
  <div class="span4">...</div>
  <div class="span8">...</div>
</div>
```

У даному прикладі у нас є `.span4` і `.span8`, створені і 12 колонок і однією суцільною рядка.

4.1.3 Порівняння класів Zurb Foundation та Twitter Bootstrap

Обидва фреймворку мають набір основних компонентів (сітка, кнопки, форми, таблиці), нижче представлена таблиця 4.1 порівняння класів для цих компонентів.

Таблиця 4.1 — Порівняння класів

Елемент	Bootstrap	Foundation
Alert	.alert .alert-success .alert-danger .alert-info	.alert-box .success .warning .info .round .radius .secondary

Продовження таблиці 4.1

Кнопки	.btn .btn-default .btn-primary .btn-success .btn-info .btn-warning .btn-danger .btn-link .btn-lg .btn-sm .btn-xs .btn-block	.button .tiny .small .large .secondary .success .alert .radius .round .disabled .expand
Списки	.list-unstyled .list-inline	.inline-list (також обнуляє стилізацію)
Labels	.label .label-default .label-primary .label-success .label-info .label-warning .label-danger	.label .success .alert .secondary .round .radius
Таблиці	.table .table-striped .table-hover .table-bordered .table-condensed .active .success .info .warning .danger	Таблиці стилізовані без можливості змінювати стилі за допомогою додаткових класів фреймворку.
Панелі	.panel .panel-default .panel-primary .panel-success .panel-info .panel-warning .panel-danger	.panel .callout .radius Немає інших кольорів і немає класів для додавання шапки панелі.
Прогрес-бари	.progress-bar .progress-bar-success .progress-bar-info .progress-bar-warning .progress-bar-danger .progress-bar-striped .active	.progress .small-# .large-# .secondary .success .alert .radius .round

Продовження таблиці 4.1

Утиліти для роботи з текстом	.text-left .text-center .text-right .text-justify .text-nowrap .text-lowercase .text-uppercase .text-capitalize .text-muted .text-primary .text-success .text-info .text-warning .text-danger	.text-left .text-right .text-center .text-justify .small-text-left (також працює для medium, large, xlarge) .small-only-text-left (також працює для medium, large, xlarge) .small-text-center (також працює для medium, large, xlarge) .small-only-text-center (також працює для medium, large, xlarge) .small-text-right (також працює для medium, large, xlarge) .small-only-text-right (також працює для medium, large, xlarge) .small-text-justify (також працює для medium, large, xlarge) .small-only-text-justify (також працює для medium, large, xlarge)
Класи для відображення елементів	.visible-*-block .visible-*-inline .visible-*-inline-block .hidden-xs .hidden-sm .hidden-md .hidden-lg	.show-for-small-only (medium, large, xlarge, xxlarge) .show-for-small-up (medium, large, xlarge, xxlarge) .hide-for-small-only (medium, large, xlarge, xxlarge) .hide-for-small-up (medium, large, xlarge, xxlarge) .show-for-landscape .show-for-portrait .show-for-touch .hide-for-touch .hidden-for-small-only (medium, large, xlarge, xxlarge) .hidden-for-medium-up (large, xlarge, xxlarge) .visible-for-small-only (medium, large, xlarge, xxlarge) .visible-for-medium-up (large, xlarge, xxlarge)

Таблиця 4.2 — Порівняння класів для побудови сітки

Bootstrap	Foundation
.container .container-fluid .row .col-xs-# .col-sm-#	.row .columns .small-# .medium-# .large-#

.col-md-#	.small-offset-#
.col-lg-#	.medium-offset-#
.col-xs-offset-#	.large-offset-#
.col-sm-offset-#	
.col-md-offset-#	
.col-lg-offset-#	

Цікаво бачити, як кожен фреймворк вирішує певні завдання. Foundation використовує універсальні класи, такі як *.secondary*, *.success*, *.alert*, *.radius*, і *.round*, в той час як Bootstrap додає до класів префікси з назвою елемента: *.btn*, *.alert-*, і *.panel-*. Foundation має багато класів для роботи з видимістю елемента, а Bootstrap більш гнучкий в плані побудови таблиць.

Так як в нас є задача виводу розпізнаних знаків у таблицю, то був обраний Bootstrap.

4.3 Bottle фреймворк

Bottle — це швидкий, простий і легкий WSGI мікро веб-фреймворк для Python. Він поширюється у вигляді одного файлу-модуля і не має жодних залежностей, крім стандартної бібліотеки Python. Ми використовуємо SQLite (див. розд.4.3) в якості бази даних. У системах Linux, більшість дистрибутивів мають SQLite3 встановленим за замовчуванням.

Крім того, необхідний Pysqlite, модуль Python для доступу до баз даних SQLite. Знову ж, у багатьох дистрибутивах Linux предумовлен цей модуль (їх часто називають "*Python-sqlite3*"), в іншому випадку можна просто встановити вручну або через *easy_install pysqlite*.

В принципі, кожна сторінка відображається в браузері після динамічної генерації, коли викликається адреса сторінки. Таким чином, немає статичного контенту. Це саме те, що називається "*route*" в Bottle: адреса на сервері. Так, наприклад, коли сторінка *http://localhost: 8080/todo* викликається з браузера, Bottle "перехоплює" виклик і перевіряє, чи є (Python) функція, визначена для

					РБ101.17010201.072.007 ПЗ	Арк.
						9 5
Змн.	Арк.	№ докум.	Підпис	Дата		

маршруту (route) "todo". Якщо так, то Bottle буде виконувати відповідний код Python і повертає результат.

Bottle — дуже легкий фреймворк і поміщається всього в один файл — *bottle.py*. Незважаючи на свою мінімалістичність, Bottle надає досить широкі можливості, яких на 100% вистачає для малих і середніх проєктів. Ось список основних можливостей:

1. Routing

Роутінг в bottle, як і в більшості фреймворків для Pythona, здійснюється за допомогою декораторів. наприклад:

```
@route('/hello/<name>')
def index(name):
    return name
```

Також динамічні url можна складати на основі регулярних виразів:

```
@route('/news/<number:re:[0-9]*>')
def show_news(number):
    pass
```

2. Templates

Одна з найсильніших сторін фреймворку — механізм шаблонів. Щоб скористатися шаблонізатором, достатньо написати таку легку конструкцію:

```
template('template_name', name=name, number=number, foo=bar)
```

Перший аргумент функції — назва файлу, в якому міститься текст шаблону (наприклад, шаблон буде називатися *template_name.tpl*). У самому ж файлі нам потрібно написати назву змінної в двох фігурних дужках:

```
Hello, {{name}}, glad to see you!
```

За замовчуванням зроблено так, що якщо в дужках вказано html-код, то він не виконається, щоб уникнути XSS-атак. Якщо це дуже необхідно, можна

					РБ101.17010201.072.007 ПЗ	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

написати `{{!name}}`. Також Bottle надає можливість: писати будь python-код усередині шаблону. Щоб викликати пітон, досить на початку рядка поставити `%`. наприклад:

```
%a = 100500
%for i in xrange(a):
    <div class="image_{{i}}"></div>
%end
```

Також можна включати шаблони з шаблонів, наприклад:

```
%include template_num2 foo=bar, blabla=qweqwe
```

3. POST-routing і обробка форм

Механізм для обробки POST запитів абсолютно такий же, як і для обробки GET запитів, просто слово `route` потрібно замінити на `post`:

```
@post("/url")
def foo():
    pass
```

Для доступу до форм використовуються атрибути полів «`name`». наприклад:

```
<input name="age" placeholder="Вік">
```

Щоб отримати вміст форми, потрібно використовувати наступну конструкцію:

```
request.forms.get("age") # Отримати зміст одного поля age
request.forms.getall("age") # Отримати зміст всіх полів age
```

Також можна звертатися і з файлами:

					РБ101.17010201.072.007 ПЗ	Арк.
						97
Змн.	Арк.	№ докум.	Підпис	Дата		

```
request.files.get("picture") # Отримати один файл з поля picture
request.files.getall("picture") # Отримати всі файли з поля (mult-upload)
```

4. Cookies

Звертатися з Cookies в bottle дуже просто, щоб встановити cookie:

```
response.set_cookie("name", value, max_age=100500)
```

Щоб взяти значення:

```
request.get_cookie("name")
```

5. Сервер

У bottle вшитий простий http-сервер, який придатний хіба що для дуже швидкого тестування однієї сторінки:

```
run(host='localhost', port=8080)
```

Зрозуміло, що для більш великих проектів використовувати його неможливо, тому треба пов'язати bottle з apache або nginx. \Bottle зв'язується з Apache через mod_wsgi. Для того, щоб це реалізувати, потрібно зробити наступне:

- Створити файл *adapter.wsgi* з ось таким вмістом:

```
import sys, os, bottle
sys.path.append(os.path.dirname(os.path.abspath(__file__)))
os.chdir(os.path.dirname(os.path.abspath(__file__)))
import index # Основной файл

application = bottle.default_app()
```

- Встановити і включити *mod_wsgi*
- Додати налаштування віртуального хоста:

					РБ101.17010201.072.007 ПЗ	Арк.
						98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<VirtualHost *:80>
    DocumentRoot /var/www/foo
</VirtualHost>

<Directory /var/www/foo>
    Options FollowSymLinks ExecCGI
    AddHandler wsgi-script .wsgi
    Order allow,deny
    AllowOverride All
    Allow from all
</Directory>

```

Часті помилки та їх вирішення:

– Якщо ваш сайт працює через apache, то потрібно бути дуже акуратним у роботі зі шляхами, потрібно завжди використовувати повні шляхи. На початку коду правильно визначимо робочий каталог, а далі просто його використовуємо. Наприклад, ось так:

```

sys.path.append(os.path.dirname(os.path.abspath(__file__)))
os.chdir(os.path.dirname(os.path.abspath(__file__)))
cwd = os.getcwd()

```

– Якщо шаблони беруться з якоїсь папки (наприклад, views), то обов'язково потрібно додати повний шлях до цієї папки в список *bottle.TEMPLATE_PATH*.

4.4 SQLite

SQLite — полегшена реляційна система керування базами даних, яка міститься у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92. Сирцевий код SQLite поширюється як суспільне надбання (англ. *public domain*), тобто може використовуватися без обмежень та безоплатно з будь-якою метою. Фінансову підтримку розробників SQLite здійснює спеціально створений консорціум, до якого входять такі компанії, як Adobe, Oracle,

					РБ101.17010201.072.007 ПЗ	Арк.
						99
Змн.	Арк.	№ докум.	Підпис	Дата		

Mozilla, Nokia, Bentley і Bloomberg. У 2005 році проект отримав нагороду Google-O'Reilly Open Source Awards.

Особливістю SQLite є те, що воно не використовує парадигму клієнт-сервер, тобто рушій SQLite не є окремим процесом, з яким взаємодіє застосунок, а надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми. Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується застосунок. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції весь файл, що зберігає базу даних, блокується; ACID-функції досягаються зокрема за рахунок створення файлу-журналу.

Кілька процесів або нитей можуть одночасно без жодних проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, коли жодних інших запитів у цей час не обслуговується; інакше спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

У комплекті постачання йде також функціональна клієнтська частина у вигляді виконуваного файлу sqlite3, за допомогою якого демонструється реалізація функцій основної бібліотеки. Клієнтська частина працює з командного рядка, і дозволяє звертатися до файлу БД на основі типових функцій ОС.

Завдяки архітектурі рушія можливо використовувати SQLite як на вбудовуваних (embedded) системах, так і на виділених машинах з гігабайтними масивами даних.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 0 0
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3.1 Особливості SQLite

- транзакції атомні, послідовні, ізольовані, і міцні (ACID) навіть після збоїв системи і збоїв живлення
- Встановлення без конфігурації — не потрібно ані установки, ані адміністрування
- Реалізує більшу частину стандарту SQL92
- Повна база даних зберігається в одному крос-платформовому файлі на диску
- Підтримка терабайтних розмірів баз даних і гігабайтного розміру рядків і BLOB.
- Малий розмір коду: менше ніж 350KB повністю налаштований, і менш 200KB з опущеними додатковими функціями
- Швидший за популярні рушії клієнт-серверних баз даних для найпоширеніших операцій
- Простий, легкий у використанні API
- Написана в ANSI C, включена прив'язка до TCL; доступні також прив'язки для десятків інших мов
- Добре прокоментований сирцевий код зі 100% тестовий покриттям гілок
- Доступний як єдиний файл сирцевого коду на ANSI C, який можна легко вставити в інший проект
- Автономність: немає зовнішніх залежностей
- Крос-платформовість: з коробки підтримується Unix (Linux і Mac OS X), OS/2, Windows (Win32 і WinCE). Легко переноситься на інші системи
- Сирці перебувають в суспільному надбанні
- Поставляється з автономним клієнтом інтерфейсу командного рядка, який може бути використаний для управління базами даних SQLite.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 0 1
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3.2 Технології, що підтримують SQLite

Сама бібліотека SQLite написана мовою C; існує велика кількість прив'язок до інших мов програмування, зокрема до C++, Java, Python, Perl, PHP, Tcl (засоби для роботи з Tcl включені в комплект постачання SQLite), Ruby, Haskell, Scheme, Smalltalk і Lua, а також до багатьох інших. Повний список наявних засобів можна знайти на сторінці проекту.

4.5 Результати роботи

Експеримент 1 — 10 фотографій (середня обробка) з різними кутами та відстанями.

Експеримент 2 — відео (реальні машини) під прямим кутом й різними кутами.

Як бачимо нижче, комплекс розпізнав номерні символи коректно. Нижче, в таблиці 4.3 ми протестували 10 фотографій, визначили середню кількість розпізнаних номерних знаків, а також відсоток успішного розпізнавання номерних знаків (букв і цифр).

Таблиця 4.3 — Проведення експерименту №1

	№ експерименту	Назва фотографії	Інформація(роздільна здатність)	Кут	Час на обробку, мс	Знайдений номерних знаків	Процент успішного розпізнавання, %	Ймовірність відносно еталона, %
	1.	IMG_3937	3888*2592 40mm	91°	10933.7 870.93	PASS	100	89.964
	2.	IMG_3939	3888*2592 45mm	45°	9058.48	FAIL	0	0
	3.	IMG_3944	3888*2592 35mm	110°	12243.7 798.04	PASS	100	84.5336
	4.	IMG_3950	3888*2592 22mm	135°	11895.2 827.54	PASS	83.4	86.5897
	5.	IMG_3959	3888*2592 50mm	120°	10585.1	FAIL	0	0

					РБ101.17010201.072.007 ПЗ	Арк.
						1 0 2
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 4.3

	6.	IMG_3960	3888*2592 24mm	112°	12137.8 3369.62	PASS	100	70.7267
	7.	IMG_8911	3888*2592 56mm	75°	7876.38 480.55	PASS	100	90.7512
	8.	IMG_9313	3888*2592 20mm	85°	8470.21 502,11	PASS	83.4	86.245
	9.	IMG_3957	3888*2592 49mm	160°	10254.5 881.781	PASS	100	88.4036
	10.	IMG_3938	3888*2592 35mm	88°	11076.1 814.795	PASS	100	87.9159
total	10	---	---	102°	10453,1	Співвідношення Pass/Fail 8/10	78,34	---

Виходячи з отриманих даних, видно, що 8 з 10 номерних знаків комплекс успішно розпізнав, і 78.34% успішного розпізнавання символів. Нижче на рисунках 4.3 і 4.4 показані діаграми з результатами дослідження.



Рисунок 4.3 — Результати розпізнавання номерних знаків

					РБ101.17010201.072.007 ПЗ	Арк.
						1 0 3
Змн.	Арк.	№ докум.	Підпис	Дата		

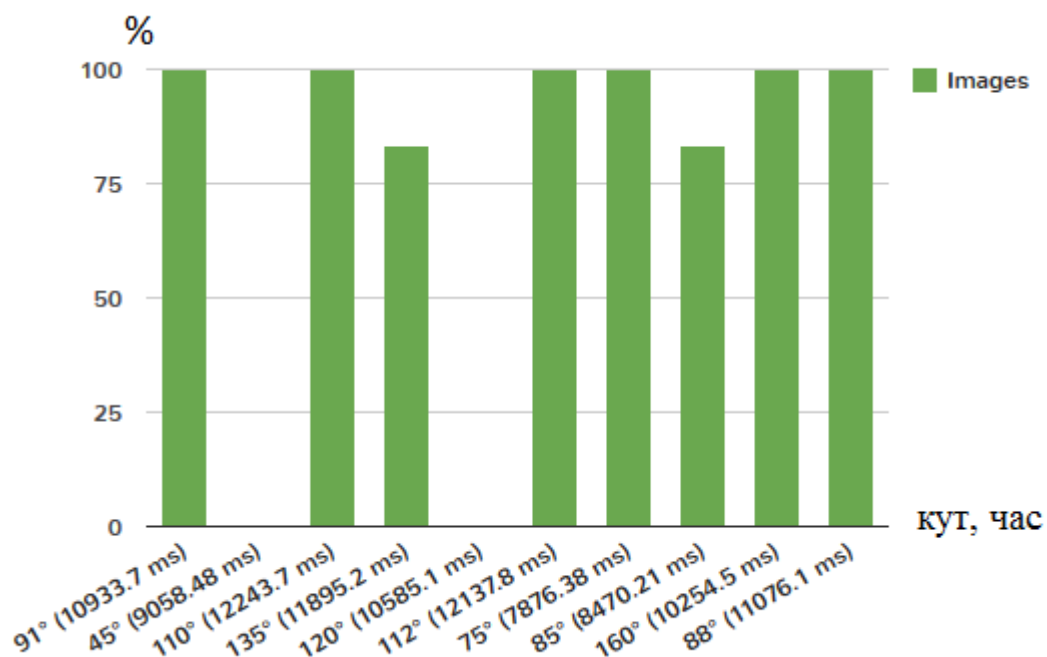


Рисунок 4.4 — Результати розпізнавання номерних символів

Проведемо експеримент № 2. Для нього нам знадобитися у вигляді вхідних даних взяті з камери зображення, з урахуванням різних кутів та відстані, як було вище.

Так як розмір номера на кадрі невідомий, також як і розташування площини номерів відносно фокальної площині камери, необхідно провести процедуру вирівнювання. Так як вже відомі чотири координати положення кутів номери і відомо, ідеальне розташування номера для розпізнавання, то можна за даними відповідностям знайти матрицю 3x3 гомографії. За допомогою знайденої матриці ми перетворимо кожен піксель знайденого спотвореного номера, в результаті чого отримуємо прямокутний номерний знак, готовий до подальшої обробки.

Нижче представлена таблиця 4.4 за урахуванням відеозйомки.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 0 4
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.4 — Проведення експерименту №2

	№ експерименту	Назва відео	Інформація(роздільна здатність)	Кут	Час на обробку, мс	Відстань від камери до машини, м	Знайдення номерних знаків	Процент успішного розпізнавання номерних знаків(літер та чисел)
	1.	Відео_1 (AK1234AB)	640*480	90°	3463.68 1036.0	1	PASS	100%
	2.	Відео_2 (AK1234AB)	640*480	83°	3242.85 956.23	1	PASS	100%
	3.	Відео_3 (CB2558AK)	640*480	90°	3421.24 1057.21	1	PASS	100%
	4.	Відео_4 (CB2558AK)	640*480	115°	3712.5	1	FAIL	0%
	5.	Відео_5 (AA2080HH)	640*480	90°	3521.54 1002.47	1	PASS	100%
	6.	Відео_6 (AA2080HH)	640*480	80°	3412.0 982.12	1	PASS	100%
	7.	Відео_7 (BH9997AX)	640*480	90°	3125.64 1004.19	1	PASS	100%
	8.	Відео_8 (BH9997AX)	640*480	105°	3689.85 1203.0	1	PASS	87.5%
	9.	Відео_9 (BH0973EA)	640*480	90°	3356.25 1132.01	1	PASS	100%
	10.	Відео_10 (BH0973EA)	640*480	75°	3785.23 1253.12	1	PASS	87.5%
total	10			90.8	3473.078		Pass/ Fail 9/10	88%

Виходячи з отриманих даних, видно, що 9 з 10 номерних знаків комплекс успішно розпізнав, і 88% успішного розпізнавання символів. Нижче на рисунках 4.5 і 4.6 показані діаграми з результатами роботи.



Рисунок 4.5 — Результати розпізнавання номерних знаків

					РБ101.17010201.072.007 ПЗ	Арк.
						105
Змн.	Арк.	№ докум.	Підпис	Дата		

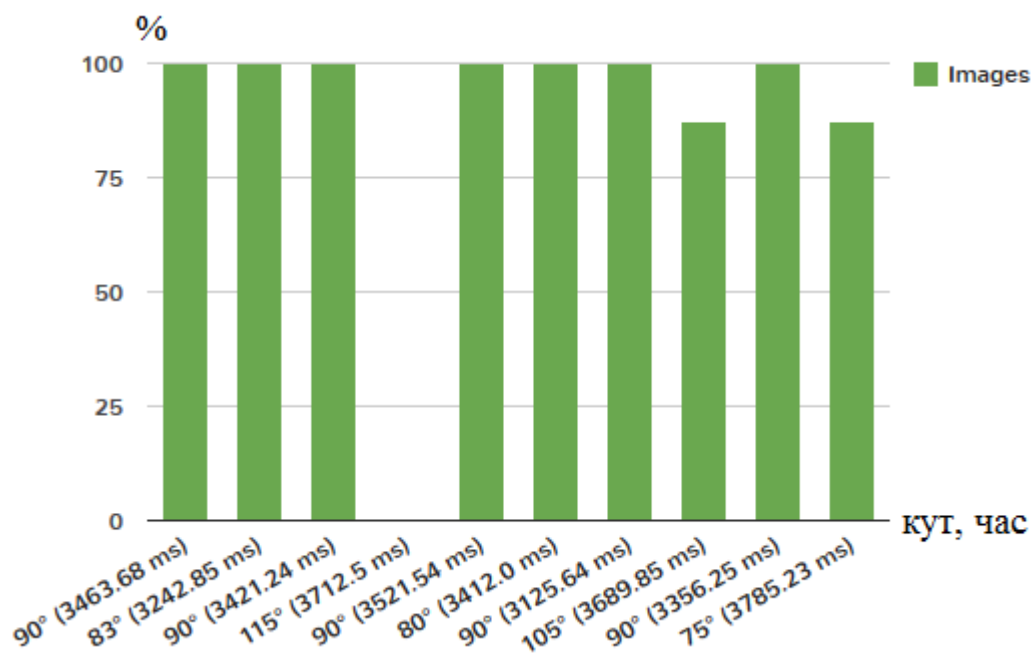


Рисунок 4.6 — Результати розпізнавання номерних символів

4.6. Висновки

Для реалізації системи з обробки та розпізнаванню зображень та відео з символічною інформацією був обраний одноплатний комп'ютер Raspberry Pi. Завдяки створеному сайту, інформація є дружній для користувача (user-friendly). Проаналізувавши систему на ефективність розпізнавання, отримали показник в 78.34% ймовірності правильного розпізнавання, що є задовільним результатом і 88 % з використанням відео.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 0 6
Змн.	Арк.	№ докум.	Підпис	Дата		

					РБ101.17010201.072.007 ПЗ	Арк.
						1 0 7
Змн.	Арк.	№ докум.	Підпис	Дата		

До невиробничих фондів відносяться фонди підприємства які безпосередньо не беруть участь у виробничому процесі і не створюють для нього умов, але числяться на балансі підприємства.

Залежно від ступеня участі у виробничому процесі основні виробничі фонди діляться на активні і пасивні.

Активні — це основні фонди, що безпосередньо беруть участь у виробничому процесі, видозмінюючи форму або стан предмету праці (машини, устаткування).

Пасивні — це основні виробничі фонди, що забезпечують нормальне функціонування активних.

Оборотні кошти — частина фондів підприємства, які одноразово беруть участь в процесі виробництва і повністю переносять свою вартість на готовий продукт.

Розробка програмного продукту є складним і тривалим процесом, що вимагає виконання значної кількості різноманітних операцій. Кожній операції відведено свій час та місце.

У цій главі приведений розрахунок економічних показників кваліфікаційної роботи спеціаліста, що присвячена розробці програмно-апаратного комплексу для розпізнавання номерних знаків (ALPR), як продовження попереднього проекту, зв'язного з комп'ютерним зором.

У якості початкових даних для розробки ALPR беруться зображення автомобілів. В якості сервера виступає одноплатний комп'ютер Raspberry Pi, а також після обробки динамічно оновлює на створеному сайті. Тобто розроблюється продукт для повноцінного розпізнавання відео та зображень.

До складу розрахунку економічних показників дипломного проекту включається як визначення трудомісткості і ціни, так і визначення її результативності. В даному розділі також будуть розраховані базова та планова собівартості і рентабельність. Крім цього необхідно побудувати

					РБ101.17010201.072.007 ПЗ	Арк.
						108
Змн.	Арк.	№ докум.	Підпис	Дата		

мережевий графік підготовки виробництва нового програмного продукту та розрахувати критичний шлях.

5.2 Складання етапів виконання

У технологічній структурі робіт можна виділити кілька самостійних етапів, а саме: розробка технічного завдання, огляд літератури, теоретичні й експериментальні дослідження, узагальнення й оцінка результатів. Дисперсія характеризує ступінь невизначеності виконання роботи за очікуваний час. Складемо перелік всіх етапів, які необхідно виконати (таблиця 5.1).

Таблиця 5.1 — Перелік всіх етапів

Етап проведення дипломної роботи	Вид робіт
Розробка технічного завдання (ТЗ)	Складання і затвердження ТЗ.
Література	Вибір літератури та складання огляду літератури.
Теоретичні та експериментальні дослідження	Проектування ALPR та програмна реалізація алгоритму функціонування ALPR, обробка отриманих даних. Зіставлення результатів експериментів з теоретичними дослідженнями. Проведення економічних розрахунків і досліджень по охороні праці. Складання проміжного звіту по охороні праці й економіці.
Узагальнення й оцінка результатів досліджень	Узагальнення результатів попередніх етапів роботи. Оцінка повноти рішення поставлених задач по програмній реалізації ALPR. Розробка рекомендацій з використанням отриманих результатів. Складання й оформлення звіту. Розгляд результатів проведеної роботи і приймання роботи в цілому.

Таблиця 5.2 — Перелік видів робіт

Вид робіт	Оцінки трудомісткості, Дні			Розрахункові величини, дні	
	$t_{\min i}$	$t_{н.ві}$	$t_{\max i}$	$t_{ожі}$	D_i
Складання і затвердження ТЗ	9	12	16	13	0,67
Вибір літератури та складання огляду	82	84	88	85	2,65
Проектування ALPR та програмна реалізація алгоритму функціонування ALPR, обробка отриманих даних	53	57	62	58	2,31
Зіставлення результатів експериментів з теоретичними дослідженнями	76	77	81	77,6	0,23
Проведення економічних розрахунків і досліджень по охороні праці	36	42	47	43	0,52
Перевірка економічних розрахунків	1	2	2	2	0,03
Проведення розрахунків і досліджень по охороні праці. Складання звіту	6	7	10	8	0,42
Перевірка розрахунків по охороні праці	1	2	2	2	0,03
Узагальнення результатів попередніх етапів роботи. Оцінка повноти рішення поставлених задач по програмній реалізації ALPR	15	19	22	19	0,25
Складання й оформлення звіту	25	29	33	29,3	0,38
Малий захист дипломної роботи	1	1	2	1	0,03
Захист роботи в цілому	1	1	1	1	0

У загальну трудомісткість дипломного проекту, крім трудомісткості розробки були включені трудомісткості конструкторського контролю і нормо контролю.

					РБ101.17010201.072.007 ПЗ	Арк.
						110
Змн.	Арк.	№ докум.	Підпис	Дата		

$$t_{kk} = 0.7 \cdot N_{стр} \quad (5.1)$$

$$t_{нк} = 0.15 \cdot N_{стр} \quad (5.2)$$

де 0.7, 0.15 — типові норми часу на проведення відповідно конструкторського контролю (контролю керівника) і нормоконтролю одного аркушу формату А4;

$N_{стр}$ — кількість аркушів, що підлягають контролю на кожному етапі.

Результати розрахунку трудомісткості кожного самостійного етапу розробки зведені в таблиці 5.3.

Таблиця 5.3 — Розрахунок собівартості і ціни виконання дипломної роботи

Найменування етапів	Трудомісткість, година		
	Розробка	Контроль керівника	Нормоконтроль
1.Розробка ТЗ	48	1,5	0,3
2.Вибір і огляд літератури	244	8	1,5
3.Теоритичні й експериментальні дослідження	428	24,5	5,75
4.Узагальнення й оцінка результатів досліджень	192	8	1,2
Разом, годин	912	42	9,25
Загальний час, годин	963,25		

Результатом виконання дипломного проекту є програмна продукція, що представляє собою закінчену роботу, виконані відповідно до вимог, передбачених в завданні на дипломне проектування.

Виходячи з особливостей створення програмної продукції будемо проводити по статтях: «Матеріали», «Спецобладнення», «Витрати на службові

					РБ101.17010201.072.007 ПЗ	Арк.
						1 1 1
Змн.	Арк.	№ докум.	Підпис	Дата		

відрядження», «Витрати на оплату праці», «Витрати на соціальні заходи», «Витрати на роботи, виконувані сторонніми підприємствами, установами й організаціями», «Накладні витрати» і «Інші витрати». Окремі елементи статей витрат знаходять різними методами, у залежності від їхнього виду.

5.3 Визначення витрат

Витрати на матеріали, покупні комплектуючі вироби і напівфабрикати визначаємо на основі розрахунку потреби в них (за винятком поворотних відходів) по діючим оптових цінах з урахуванням транспортно-заготівельних витрат, величина яких складає 7-10% від оптової вартості матеріалів, покупних напівфабрикатів і комплектуючих виробів. Результати розрахунку матеріальних витрат приведені в таблиці 5.4

Таблиця 5.4 — Результати розрахунку матеріальних витрат

Найменування ресурсів	Одиниця виміру	Необхідна кількість	Ціна за одиницю, грн.	Сума, грн.
Папір писальний	Лист	400	-	20
Папір для принтера	Лист	250	-	20
Диски	Шт.	3	2	6
Ручки	Шт.	4	1	4
Разом				50
ТЗР				5,3
Усього				55,3

До витрат за статтею «Спецобладнання» відносяться витрати на придбання чи виготовлення спеціальних приладів, стендів, апаратів і іншого спеціального устаткування, необхідного для виконання конкретної дипломної роботи. Оскільки дана робота носить суто програмний характер, то спеціальне устаткування не здобувається, а, отже, немає необхідності в даному

					РБ101.17010201.072.007 ПЗ	Арк.
						1 1 2
Змн.	Арк.	№ докум.	Підпис	Дата		

розрахунку. До складу статті витрат «Основна заробітна плата» відноситься оплата праці виконавців, безпосередньо залучених до її виконання. Вихідними даними для розрахунку основної заробітної платні є трудомісткість окремих етапів робіт з категорій працюючих відповідно до даних таблиці 5.5. Розмір основної заробітної плати виконавців визначаємо

Таблиця 5.5 — Розмір заробітної плати

Найменування робіт	Місячний посадовий оклад, грн.	Надбавки, 15 - 50%	Місячний посадовий оклад з урахуванням надбавок, грн.
Виконавець роботи	4000	700	4700
Науковий керівник	4000	700	4700
Консультант по охороні праці	2500	438	2938
Консультант по економічній частині	2500	438	2938
Сума основної заробітної платні			15276

Витрати на додаткову заробітну плату визначимо у відсотках від основної заробітної плати, що у наукових установах складає 10 – 12% від основної заробітної плати.

Додаткова зарплата дорівнює:

$$ЗП_{дон} = (0.1 \div 0.12) ЗП_{осн} \quad (5.3)$$

$$ЗП_{дон} = 0.1 \cdot 15270 = 1527 \text{ грн}$$

До складу собівартості дипломного проекту включаються податки, збори й інші обов'язкові платежі, установленні діючої системою оподаткування. З огляду на діючі в даний час ставки відрахувань, сума внесків у різні фонди складає:

1. Відрахування на обов'язкове державне пенсійне страхування:

					РБ101.17010201.072.007 ПЗ	Арк.
						1 1 3
Змн.	Арк.	№ докум.	Підпис	Дата		

$$З_{пенс.ф} = 0.318(ЗП_{осн} + ЗП_{дон}) \quad (5.4)$$

$$З_{пенс.ф} = 0.318 \cdot 16797 = 5341.44 \text{ грн}$$

2. Відрахування на державне (обов'язкове) соціальне страхування:

$$З_{с-с} = 0.029(ЗП_{осн} + ЗП_{дон}) \quad (5.5)$$

$$З_{с-с} = 0.029 \cdot 16797 = 487.11 \text{ грн}$$

3. Відрахування на державне (обов'язкове) соціальне страхування на випадок безробіття:

$$З_{безр} = 0.013(ЗП_{осн} + ЗП_{дон}) \quad (5.6)$$

$$З_{безр} = 0.013 \cdot 16797 = 218.36 \text{ грн}$$

4. Комунальний податок:

$$З_{ком} = Ч_{ср} \cdot C_{\min} \cdot 0.1, \quad (5.7)$$

де $Ч_{ср}$ — середньооблікова кількість працівників, що беруть участь у виконанні даної роботи:

$$Ч_{ср} = (ЗП_{осн} + ЗП_{дон}) / 3250, \quad (5.8)$$

де C_{\min} — неоподатковуваний податком мінімум заробітної плати (17грн.);

$$Ч_{ср} = 16797 / 3250 = 5 \text{ чол.}$$

					РБ101.17010201.072.007 ПЗ	Арк.
						1 1 4
Змн.	Арк.	№ докум.	Підпис	Дата		

$$З_{ком} = 0.1 \cdot 17 \cdot 5 = 8.5 \text{ грн}$$

Витрати на службові відрядження розраховуватися не будуть, тому що немає необхідності в цих витратах.

На статтю «Витрати по роботах, виконуваним сторонніми організаціями і підприємствами» відносять витрати по оплаті усіх видів робіт, у тому числі по виготовленню дослідних зразків і макетів, виконуваних безпосередньо для конкретного дипломного проекту сторонніми організаціями і підприємствами, а також підлеглими науковій установі дослідними виробництвами, що перебувають на самотійному балансі.

По даній статті витрат буде враховуватися тільки оплата роздруківки результатів обчислень, тому що інших робіт у даному дипломному проекті не передбачається. Результати розрахунку наведені у таблиці 5.6. У статті «інші прямі витрати» враховується оплата машинного часу ЕОМ. Витрати на машинний час визначаються по формулі (5.9):

Таблиця 5.6 — Оплата машинного часу ЕОМ

Організації і підприємства співвиконавці НИР	Виконувана робота	Витрати по роботах відповідно до договорів на їхнє виконання
М.П. «Копірка»	Роздруківка форматів А1	12шт. · 12грн. = 240грн

$$З_{м.у} = t_{м.у} \cdot Ц_{м.г} \quad (5.9)$$

де $t_{м.у}$ — фактичний час експлуатації ЕОМ при проведенні наукового дослідження; $Ц_{м.г}$ — вартість машинного часу роботи на ЕОМ, грн./година.

$$З_{м.у} = 160 \cdot 4 = 640 \text{ грн}$$

					РБ101.17010201.072.007 ПЗ	Арк.
						1 1 5
Змн.	Арк.	№ докум.	Підпис	Дата		

У статтю «Накладні витрати» включаються витрати на керування і господарське обслуговування, що відносяться до усіх виконуваних дипломних проектів. У наукових установах накладні витрати складають 25%:

$$З_{\text{н}} = 0.25(ЗП_{\text{осн}} + ЗП_{\text{дон}}) \quad (5.10)$$

$$З_{\text{н}} = 0.25 \cdot 16797 = 4199.25 \text{ грн}$$

Розрахуємо базову виробничу собівартість як суму по всіх статтях витрат:

$$C_{\text{пр}} = \sum_{i=1}^9 З_i \quad (5.11)$$

$$C_{\text{пр}} = 26777.9 \text{ грн}$$

На комерційні витрати:

$$З_{\text{ком}} = (0.15 \div 0.2) C_{\text{пр}} \quad (5.12)$$

$$З_{\text{ком}} = 0.15 \cdot 26777.9 = 4016.68 \text{ грн}$$

5.4 Калькуляція базової собівартості

Собівартість продукції виражена в грошовій формі, це текучі затрати на її виробництво і збут. В залежності від повноти обхвату затрат виробництва, собівартість буває різних типів: повна собівартість, виробнича собівартість, цехова собівартість, та технологічна собівартість (рисунок 5.2).

					РБ101.17010201.072.007 ПЗ	Арк.
						1 1 6
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 5.2 — Види собівартості

Технологічна — витрати на введення технологічного процесу виготовлення продукції. Виробнича — витрати підприємства на виробництво даного виду продукції. Повна — всі витрати підприємства по виробництву і реалізації продукції

В залежності від об'єкта калькуляції розрізняють:

- собівартість валової продукції — це витрати поточного періоду на її виробництво;
- собівартість реалізованої продукції — це витрати на виробництво і збут продукції на протязі усього циклу.

Калькуляція базової собівартості з використанням показників розраховується за формулами приведеними у таблиці 5.7.

Таблиця 5.7 — Калькуляція базової собівартості

№	Найменування статті витрат	Значення, грн.	Розрахунок
1	Сировина і матеріали	3054	$З_m = 0,2 \cdot З_o$
2	Заробітна плата		
2.1	Основна заробітна плата	15270	Див. табл. 5.5

Продовження таблиці 5.7

2.2	Додаткова заробітна плата	1527	$З_Д = 0,1 \cdot З_0$
3	Відрахування		
3.1	Відрахування на соціальне страхування	487.113	$З_{cc} = 0,029 \cdot (З_0 + З_Д)$
3.2	Витрати по роботах, виконуваним сторонніми організаціями і підприємствами	240	Див. табл. 5.6
3.3	Відрахування в пенсійний фонд	5341.44	$З_{пенс} = 0,318 \cdot (З_0 + З_Д)$
3.4	Відрахування на випадок безробіття	218.36	$З_{б} = 0,013 \cdot (З_0 + З_Д)$
4	Інші прямі витрати	640	$З_{м.у.} = t_{м.у.} \cdot Ц_{м.г.}$
5	Виробнича собівартість	26777.9	$C_{np} = \sum_1^9 З_i$
6	Комунальний податок	8,5	$З_k = Ч_{cn} \cdot C_{min} \cdot 0,1,$ де $Ч_{cn} = (З_0 + З_Д) / 2125$ де $Ч_{cn}$ — середньосписочна чисельність працюючих; H_{min} — мінімальний неоподаткований мінімум доходу
7	Комерційні витрати	4016.68	$З_{ком.} = (0,15 \div 0,2) C_{np}$
8	Накладні витрати	4199.25	$З_n = (0,25) \cdot (З_{Посн.} + З_{Пдоп.})$
9	Повна собівартість	34993.83	$C_{III} = C_{np} + \sum_{i=9}^{11} З_i$

5.5 Калькуляція планової собівартості

Калькуляція планової собівартості з використанням показників розраховується за формулами приведеними у таблиці 5.8:

					РБ101.17010201.072.007 ПЗ	Арк.
						1 1 8
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 5.8 — Калькуляція планової собівартості

№	Найменування статті витрат	Значення, грн.	Розрахунок
1	Сировина і матеріали	2900	$З_м = 0,2 \cdot З_0$
2	Заробітна плата		
2.1	Основна заробітна плата	14500	Див. табл. 5.5
2.2	Додаткова заробітна плата	1450	$З_д = 0,1 \cdot З_0$
3	Відрахування		
3.1	Відрахування на соціальне страхування	462,55	$З_{сс} = 0,029 \cdot (З_0 + З_д)$
3.2	Витрати по роботах, виконуваним сторонніми організаціями і підприємствами	240	Див. табл. 5.6
3.3	Відрахування в пенсійний фонд	5072,1	$З_{пенс} = 0,318 \cdot (З_0 + З_д)$
3.4	Відрахування на випадок безробіття	207,35	$З_б = 0,013 \cdot (З_0 + З_д)$
4	Інші прямі витрати	640	$З_{м.у.} = t_{м.у.} \cdot Ц_{м.г.}$
5	Виробнича собівартість	25472	$C_{np} = \sum_{i=1}^8 З_i$
6	Комунальний податок	10,37	$З_k = Ч_{cn} \cdot C_{min} \cdot 0,1$, де $Ч_{cn} = (З_0 + З_д) / 2600$ де $Ч_{cn}$ - середньосписочна чисельність працюючих; H_{min} - мінімальний неоподаткований мінімум доходу.

Продовження таблиці 5.8

7	Комерційні витрати	3820,8	$З_{\text{ком.}} = (0,15 \div 0,2) C_{\text{пр}}$
8	Накладні витрати	3987,5	$З_{\text{н}} = (0,25) \cdot (3П_{\text{осн.}} + 3П_{\text{доп.}})$
9	Повна собівартість	33290,67	$C_{\text{ПП}} = C_{\text{пр}} + \sum_{i=9}^{11} З_i$

За результатами калькуляції базової собівартості, ми отримали:
ПДВ дорівнює 20% від собівартості роботи з урахуванням прибутку.

$$ПДВ = 7000 \text{ грн}$$

Базова ціна програмного продукту із урахуванням ПДВ дорівнює 42000 грн. Базовий прибуток:

$$П_{\text{баз}} = Ц - C_{\text{баз}} = 42000 - 34993,83 = 7006,17 \text{ грн} \quad (5.13)$$

де Ц — ціна (грн.), $C_{\text{баз}}$ — базова собівартість (грн.), $П_{\text{баз}}$ — базовий прибуток (грн.).

Базова рентабельність:

$$R_{\text{баз}} = \frac{П_{\text{баз}}}{C_{\text{баз}}} \cdot 100 = \frac{7006,17}{34993,83} \cdot 100 = 20.02\%, \quad (5.14)$$

де $C_{\text{баз}}$ — базова собівартість (грн.), $П_{\text{баз}}$ — базовий прибуток (грн.), $R_{\text{баз}}$ — базова рентабельність (%). Уточнивши витрати на покупні вироби, ми отримаємо наступні значення планової собівартості виробу:

- планова ціна програмного продукту: 34000 грн.;
- ПДВ дорівнює 20% від собівартості роботи з урахуванням прибутку, тобто $ПДВ = 6800$ грн.;

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2 0
Змн.	Арк.	№ докум.	Підпис	Дата		

- планова ціна програмного продукту із урахуванням ПДВ: 40800 грн.;
- плановий прибуток отримаємо наступний:

$$\Pi_{пл} = Ц - C_{пл} = 40800 - 33290,67 = 7509,33 \text{ грн}, \quad (5.15)$$

де $Ц$ — (грн.), $C_{пл}$ — планова собівартість (грн.), $\Pi_{пл}$ — плановий прибуток (грн.).

Тоді розрахунок планової рентабельності матиме вигляд:

$$R_{пл} = \frac{\Pi_{пл}}{C_{пл}} \cdot 100 = \frac{7509,33}{33290,67} \cdot 100 = 22,557\% \quad (5.16)$$

де $R_{пл}$ — планова рентабельність.

5.6 Мережеві графіки

Мережні методи планування й керування це:

1. Методи, що використовують мережну модель як основну форму подання інформації про керований комплекс робіт.

2. Методи, що представляють собою апарат побудови, розрахунку, аналізу й оптимізації мережних моделей, які використовуються не тільки при рішенні досить складних завдань керування, але і є основою побудови спеціального класу систем організаційного керування, які називаються системами сіткового планування й керування (СПК).

Мережні методи планування й керування застосовують для підвищення якості різних робіт, спрямованих на скорочення строків, раціональне використання ресурсів тощо.

Параметри мережевого графіку:

- початкова подія J ;

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2 1
Змн.	Арк.	№ докум.	Підпис	Дата		

- закінчуюча подія C ;
- максимальні шляхи від початкової до даної події $L(J \div i)$;
- максимальні шляхи від даної до завершальної події $L(i \div c)$;
- тривалість даної події t_{ij} ;
- критичний шлях $t(L_{kp}) = \left[\sum_{i=J}^c t_{ij} \right]_{\max}$;
- ранні t_p й пізні t_n строки настання подій;
- ранні $T_{p1} = t \cdot [L(J \div i)_{\max}]$ й пізні $T_{n1} = t \cdot [L_{kp} - L(J \div C)_{\max}]$ кроки початку й закінчення робіт;
- резерв часу події $R_i = T_{n_i} - T_{p_i}$;
- тривалість максимального шляху, проходячого через дану роботу $t(L_{\max})$;
- тривалість відрізка шляху $t(L_{\max})$ обзначається $t(L_{kp})$ та співпадаючого з критичним;
- коефіцієнт напруженості робіт $K_{Hij} = \frac{t(L_{\max}) - t(L_{kp})}{t(L_{kp}) - t(L_{kp})}$;
- сума дисперсій робіт, лежачих на критичному шляху $\sum_{i=1}^i \sigma_{t_{ij}}^2$;
- ймовірність закінчення кінцевої події у заданий строк $Z = \frac{T_d - T_k}{\sqrt{\sum_{i=1}^n \sigma_{t_{ij}}^2}}$;
- строк кінця закінчуючої події T_k ;
- директивний строк кінця закінчуючої події T_d .

Ранній строк настання події — це мінімально можливий строк, необхідний для виконання всіх робіт, що передують даній події.

Розрахунок ранніх строків настання подій ведуть у порядку — від початкової події проекту (з номером 0) до завершальні. При розрахунку приймають, що ранній строк настання початкової події дорівнює 0.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2 2
Змн.	Арк.	№ докум.	Підпис	Дата		

Пізній строк настання події — це максимально припустимий строк настання розглянутої події, обумовлена з умови, що після настання цієї події у свій пізній строк залишається досить часу, щоб виконати наступні за ним роботи. Розрахунок пізніх строків настань подій ведуть у зворотному порядку — від завершальної події проекту до початкового (з номером 0). При розрахунку приймають, що пізній строк настання завершальної події збігається з його раннім строком настання.

Пізній строк закінчення роботи збігається з пізнім строком настання її кінцевої події, а пізній строк початку роботи менше на величину тривалості цієї роботи.

Повний резерв часу деякої роботи — це максимальний час, на яке можна відстрочити її початок або збільшити тривалість, не змінюючи директивного строку настання завершальної події селевого графіка.

Вільний резерв часу деякої роботи — максимальний час, на яке можна відстрочити її початок або збільшити її тривалість за умови, що всі події наступають у свої ранні строки.

Проведемо оцінку тривалості виконання робіт. При цьому задаємо три оцінки: мінімальна оцінка тривалості, що враховує найбільш сприятливі умови проведення роботи t_{min} ; максимальна оцінка тривалості, що враховує найбільш несприятливі умови проведення роботи t_{max} , і найбільше імовірна оцінка тривалості, що відбиває умови, які найчастіше зустрічаються, проведення роботи t_{nv} . По цих величинах оцінюється очікуване значення трудомісткості $t_{ожі}$.

$$t_{ожі} = \frac{t_{min} + 4t_{nv} + t_{max}}{6}, \quad (5.17)$$

$$t_{ожі} = \left(\frac{t_{max} - t_{min}}{5} \right)^2. \quad (5.18)$$

Підставляючи максимальний і мінімальний очікуваний час розробки у формулу (5.17) одержимо:

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2 3
Змн.	Арк.	№ докум.	Підпис	Дата		

$$T_{ож} = \frac{5 + 4 \cdot 4 + 6}{6} = 4,5 \text{ год.}$$

Або $T_{ож} = 4,5 \cdot 20,4 = 92$ дні, або $T_{ож} = 92 \cdot 8 = 734$ годин,
де 20,4 — середня тривалість робочого місяця (у днях),
8 — середня тривалість робочого дня (у годинах).

Побудувати мережевий графік підготовки виробництва нового програмного продукту, оптимізувати строки виконання графіка при обмеженому числі розробників, рівному 4 чоловікам, на основі наступного переліку подій і робіт по підготовці описаних у таблиці 5.9.

Складання простого мережевого графіка конструкторської підготовки на основі таблиці 5.9 та приведено на рисунку 5.3.

Таблиця 5.9 — Перелік подій і робіт

Події		Робота			$T_{оч}$, тижні	Кількість виконавців
Код	Формулювання	Початок і	Кінець j	Зміст		
0	Завдання отримано	0	1	Розробка умов	1	3
		0	3	Аналіз інформації	1	4
1	Початкові умови розроблені	1	2	Розробка документації	2	3
2	Дослідження ринку компонентів	2	6	Розробка проекту	3	2
3	Проект розроблено	3	4	Вибір середовища розробки	2	4
4	Тестування за допомогою ЕОМ	4	5	Створення тестового пакету на ЕОМ	2	3
		4	6	Підготовка для тестування ПП на ЕОМ	4	2
5	Виправлення помилок у програмному продукті (ПП)	5	7	Тестування ПП на ЕОМ	1	4

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2 4
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 5.9

6	Працездатність програмного продукту	6	7	Виправлення помилок на ЕОМ	3	4
7	Повноцінний ALPR створено	7	8	Доведення ALPR до кінцевого стану	4	4
8	ALPR готовий	8	9	Тестування готового програмного продукту на ЕОМ	3	4
9	Оформлення документації	9	10	Остаточне оформлення документації	2	4
10	Дослідний зразок налагоджений на ЕОМ	10	11	Вибір другого середовища розробки	2	4
11	Робочий проект складено	11	12	Оформлення результатів	1	4
12	Робота оформлена (закінчення робіт)					
Загальна трудомісткість і тривалість проведення розрахункової роботи						49

Для побудови повноцінного мережевого графіку задаємося такими графічними зображеннями, що на рисунку 5.3:

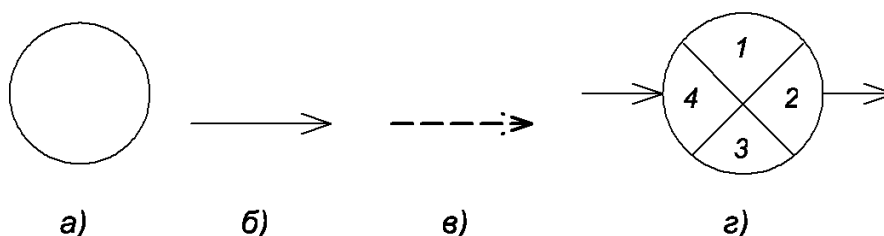


Рисунок 5.3 — Графічні зображення елементів мережевого графіку

На рисунку 5.3 зображено: а) подія; б) робота; в) фіктивна робота; г) вигляд події — тобто розбите коло на 4 сектори, що означають: 1 — номер події; 2 — пізній строк звершення події T_n ; 3 — номер події через котру для даної події відводиться максимальний за тривалістю шлях; 4 — ранній строк звершення події T_p .

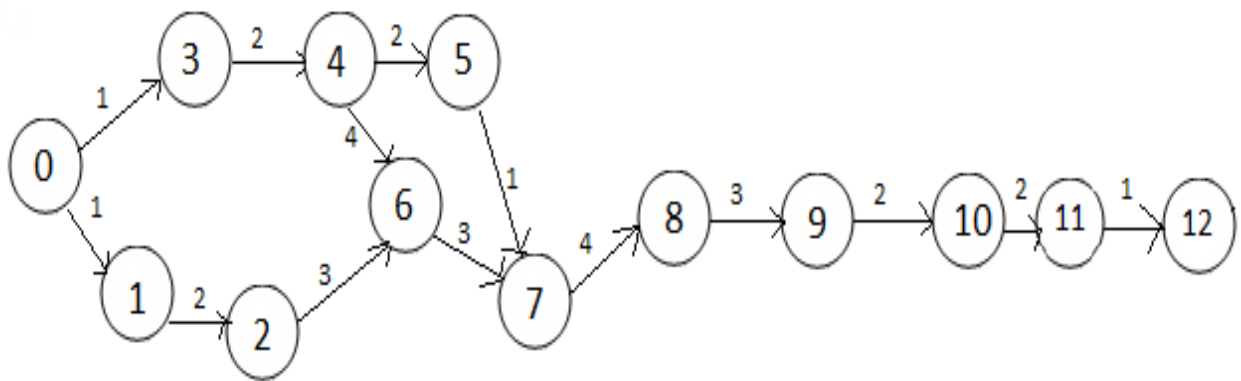


Рисунок 5.4 — Простий мережевий графік конструкторської підготовки

Визначення тривалості кожного шляху:

$$t(L_1) = t(0,1) + t(1,2) + t(2,6) + t(6,7) + t(7,8) + t(8,9) + t(9,10) + t(10,11) = 21 \text{ тиждень};$$

$$t(L_2) = t(0,3) + t(3,4) + t(4,5) + t(5,7) + t(7,8) + t(8,9) + t(9,10) + t(10,11) = 18 \text{ тижнів};$$

$$t(L_3) = t(0,3) + t(3,4) + t(4,6) + t(6,7) + t(7,8) + t(8,9) + t(9,10) + t(10,11) = 22 \text{ тижня};$$

$$t_{кр} = 22 \text{ тижня}$$

Побудуємо мережевий графік на рисунку 5.5:

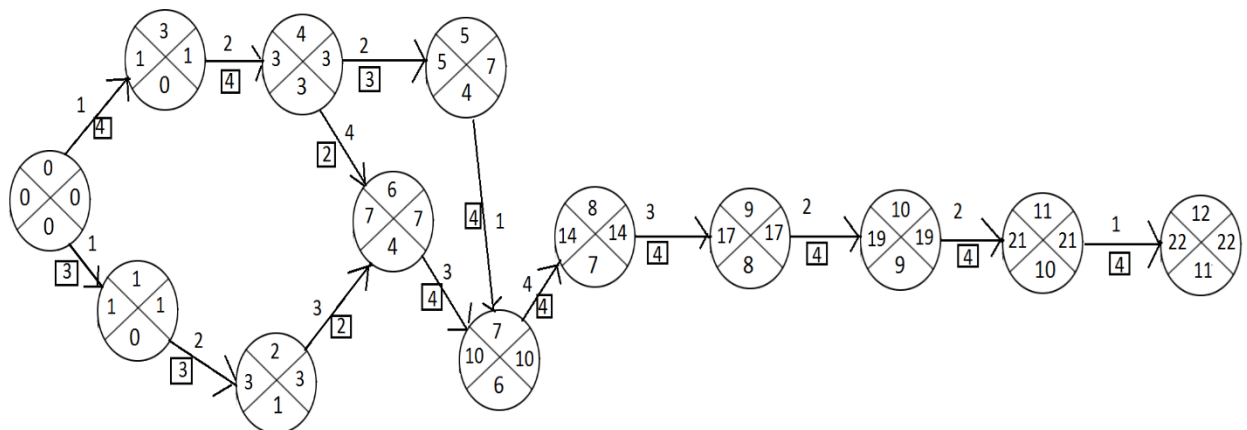


Рисунок 5.5 — Мережевий графік

5.7 Висновки

У даному розділі були проведені техніко-економічні розрахунки розробки програмно-апаратного комплексу ALPR. Провівши економічні розрахунки визначили в таблиці 5.10:

Таблиця 5.10 — Перелік подій і робіт

Показники	Одиниці виміру	Базова вир.	Планова вир.
1. Собівартість	грн	34993.83	33290,67
2. Ціна (з ПДВ)	грн	42000	40800
3. Прибуток	грн	7006,17	7509,33
4. Рентабельність	%	20,02	22,557

Також був побудований мережений графік конструкторської підготовки виробництва нового програмного продукту, були оптимізовані строки виконання графіка. Критичний шлях склав 22 тижня.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2 7
Змн.	Арк.	№ докум.	Підпис	Дата		

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2 8
Змн.	Арк.	№ докум.	Підпис	Дата		

6.2 Нормування виділених небезпечних та шкідливих факторів

Таблиця 6.1 — Нормування факторів

Фактор	Значення	Норма	Документ
Понижена температура в приміщенні лабораторії	Температура, t, °C		ГОСТ 12.1.005-88
	12	13–29	ДБН В.2.2-15-2005
Недостатня освітленість в приміщенні лабораторії	150 лк	300 лк	ДБН 13.2.5 – 28 – 2006
Підвищений рівень звукового тиску від роботи офісного обладнання	80дБ	65дБ	ГОСТ 12.1.003 – 83 ДБН В.1.2-10-2008
Перепади напруги при підключенні апаратури	< 180 В > 250 В	220 ± 10% В	ГОСТ 13109-97

Проаналізувавши дані, представлені в таблиці 6.1, зробимо висновок, що недостатнє освітлення при роботі з ПЕОМ для життя людини є найгіршим, тому зробимо розрахунок освітлення крапковим методом [34].

6.3 Розрахунок штучного освітлення

Приміщення, в якому знаходиться робоче місце оператора, має наступні характеристики:

- довжина приміщення — 10 м;
- ширина приміщення — 6 м;
- висота — $H = 4$ м;
- тип ламп у світильнику — ДРЛ;
- висота робочої поверхні — $h_{\text{пов}} = 0,8$ м;
- висота звису світильника — $h_{\text{зв}} = 0,7$ м;

					РБ101.17010201.072.007 ПЗ	Арк.
						1 2 9
Змн.	Арк.	№ докум.	Підпис	Дата		

Крапковий метод визначається за формулою 6.1:

$$E = \frac{I_a \cdot \cos^3 \alpha}{h_{cg}^2}, \quad (6.1)$$

де I_a — сила світла, для умовної лампи зі світловим потоком в 1000 лм, Кд;

α — кут між вертикальною площиною та напрямком світлового потоку;

$h_{св}^2$ — висота підвісу світильника, м.

Розміщення світильника показано на рисунку 6.1.

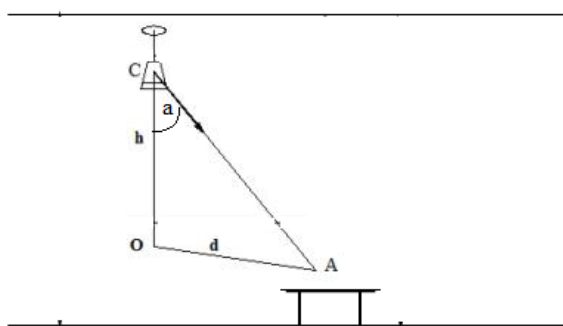


Рисунок 6.1 — Розміщення світильника

З рисунку 6.1 $h = H - h_{\text{поб}} - h_{\text{зб}} = 4 - 0,7 - 0,8 = 2,5 \text{ м}$; $d = 3 \text{ м}$.

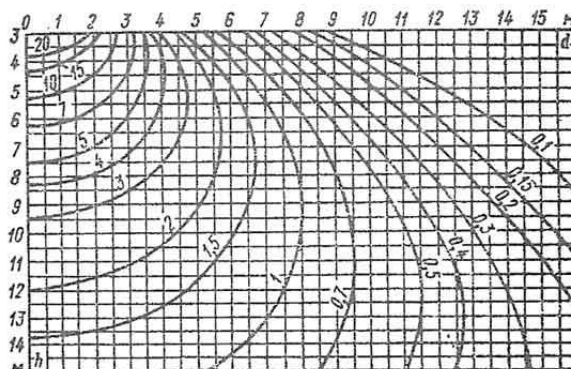


Рисунок 6.2 — Просторова ізолюкси умовної горизонтальної освітленості для світильника ДРЛ

					РБ101.17010201.072.007 ПЗ	Арк.
						130
Змн.	Арк.	№ докум.	Підпис	Дата		

За просторовими ізолюксами (рисунок 6.2) визначаємо $e = 18$, $\sum e = 18$, тому що обрана 1 лампа [35].

Визначаємо світловий потік за формулою 6.2

$$\Phi = 1000 \cdot \frac{E_{\text{норм}} \cdot K}{\mu \cdot \sum e}, \quad (6.2)$$

де $E_{\text{норм}} = 300 \text{ лк}$;

K — коефіцієнт запасу, $K = 1,4$;

μ — коефіцієнт додаткової освітленості, $\mu = 1,1$;

$$\text{Відповідно, } \Phi = 1000 \cdot \frac{300 \cdot 1,4}{1,1 \cdot 18} = 21212 \text{ лм}.$$

Обираємо лампу типа ДРЛ-400 з номінальною потужністю 400 Вт зі світловим потоком 24000 лм. Звідси, освітленість в контрольованій точці розраховується по формулі 6.3:

$$E = \frac{\Phi \cdot \mu \cdot \sum e}{1000 \cdot K}. \quad (6.3)$$

$$\text{Тобто } E = \frac{24000 \times 1,1 \times 18}{1000 \times 1,4} = 339 \text{ лк}, \text{ що відповідає нормі. Освітленість є}$$

достатньою із запасом у 13%.

6.4 Протипожежна безпека

6.4.1 Класифікація категорії приміщення

Одним із найважливіших виробничих факторів є пожежна безпека. Особливо його значення зростає в приміщенні з недостатнім вільним про-

					РБ101.17010201.072.007 ПЗ	Арк.
						131
Змн.	Арк.	№ докум.	Підпис	Дата		

стором, великою кількістю електронної техніки, корпуси яких більшістю зроблені з горючих матеріалів, велика кількість електричної проводки [36-37].

Характеристика категорій приміщень і будівель з вибухопожежної та пожежної небезпеки наведені у таблиці 6.2:

Таблиця 6.2 — Основні характеристики категорій приміщення за матеріалами, що знаходяться у приміщенні

Категорія приміщення	Характеристика речовин та матеріалів, які знаходяться у приміщенні
Категорія А	Горючі гази, легкозаймисті рідини з температурою не більше 28 °С в такій кількості, що можуть утворювати вибухонебезпечні парогазоповітряні суміші, розвивається розрахунковий надлишковий тиск вибуху в приміщенні, що перевищує 5 кПа Речовини і матеріали, здатні вибухати і горіти при взаємодії з водою, киснем повітря або одне з одним, що розрахунковий надлишковий тиск вибуху в приміщенні перевищує 5 КПа.
Категорія Б	Горючий пил або волокна, легкозаймисті рідини з температурою спалаху більше 28 °С Горючі рідини в такій кількості, що можуть утворювати вибухонебезпечні пилоповітряні або пароповітряні суміші, при спалахуванні яких розвивається розрахунковий надлишковий тиск вибуху в приміщенні, що перевищує 6 КПа.
Категорія В	Горючі і важкогорючі рідини, тверді горючі і важкогорючі речовини і матеріали (в тому числі пил і волокна), речовини і матеріали, здатні при взаємодії з водою, киснем повітря або один з одним горіти за умови, що приміщення, в яких вони утримуються (використовуються), не відносяться до категорій А та Б.
Категорія Г	Негорючі речовини і матеріали в гарячому, розжареному або розплавленому стані, процес обробки яких супроводжується виділенням променевого тепла, іскор, полум'я горючі гази, рідини, тверді речовини, які спалюються або утилізуються як паливо.

Продовження таблиці 6.2

Категорія Д	Негорючі речовини і матеріали в холодному стані. До категорії Д належать приміщення, в яких в системах машин охолодження і гідроприводу наявні ГР в системах смазки, охолодження і гідропривода обладнання при тиску не вище 0,2 МПа, а також у яких кабелі електропроводки до обладнання, окремі предмети меблі на місцях.
-------------	---

Обраний тип приміщення лабораторії відноситься до категорії – «Д», тобто знижена пожежна небезпека

Класифікація за категорією приміщень [38].

А — якщо сумарна площа приміщень категорії А перевищує 5% площі всіх приміщень або 200м². Допускається не відносити будівлі категорії А, якщо сумарна площа приміщення категорії А в будівлі не перевищує 25% сумарної площі всіх розміщених в ньому приміщень (але не більше 1000м²), і ці приміщення обладнуються установками автоматичного пожежогасіння.

Б — якщо не відноситься до категорії А і сумарна площа приміщень категорії А + Б перевищує 5% сумарної площі всіх приміщень (200м²). Допускається не відносити до Б, якщо А + Б менше або дорівнює 25% сумарної площі всіх приміщень (але не більше 1000м²)

В — сумарна площа А + Б + В більше 5% (10% якщо в будівлі відсутні приміщення категорії А + Б) сумарної площі всіх приміщень. Допускається не відносити до В, якщо А + Б + В не перевищує 25% (менше або дорівнює 3500м²).

Г — А + Б + В + Г більше 5%. Допускається не відносити до Г, якщо А + Б + В + Г менше або рівно 25% (але не більше 5000м²).

Д — якщо не відносяться ні до одного.

За цією класифікацією лабораторія відноситься до категорії Д.

Класифікація за технологічними процесами та горючими матеріалами що в них приймають участь.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 3 3
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 6.3 – Основні характеристики категорій технологічним процесом або горючих речовин, які використовуються

Категорія	Характеристика технологічного процесу або горючих речовин, які використовуються
А	Газо- та паровоздушні суміші з нижньою границею вибухонебезпечної концентрації < 10%. ЛВР $t < 25^{\circ}\text{C}$; тверді речовини, які здатні загорятися при взаємодії киснем та водою.
Б	Газо- та паровоздушні суміші з мінімальним границею > 10%, рідини $t = 28-61^{\circ}\text{C}$; горючі речовини, які створюють вибухонебезпечну концентрацію пилу в приміщенні.
В	Горючі рідини з $t > 61^{\circ}\text{C}$; тверді горючі речовини.
Г	Вогнева техніка, обробка металів в розколеному стані.
Д	Холодна обробка металів та негорючих матеріалів.

За цією класифікацією лабораторія відноситься до категорії Д.

Отже за трьома класифікаціями лабораторія відноситься до категорії Д.

6.4.2 Евакуація персоналу від робочого місця

У всіх будівлях і спорудах на випадок пожежі має бути передбачена й забезпечена евакуація людей з приміщень, що горять, через так звані евакуаційні виходи.

Таблиця 6.4 — Початкові дані для розрахунків

Кількість людей	Довжина приміщення, м	Ширина приміщення, м	Висота приміщення, м	Поверх
38	100	30	20	4

					РБ101.17010201.072.007 ПЗ	Арк.
						1 3 4
Змн.	Арк.	№ докум.	Підпис	Дата		

Згідно аудиторії:

$l_1 = 10$ м — парти

$l_2 = 0$ м — дверний отвір

$l_3 = 6$ м — відстань від дверей до сходів

$l_4 = 33$ м — сходи вниз

$l_5 = 10$ м — до входу

$l_6 = 0$ м — двері ІРТ

На рисунку 6.3 показана схема евакуації з приміщення.

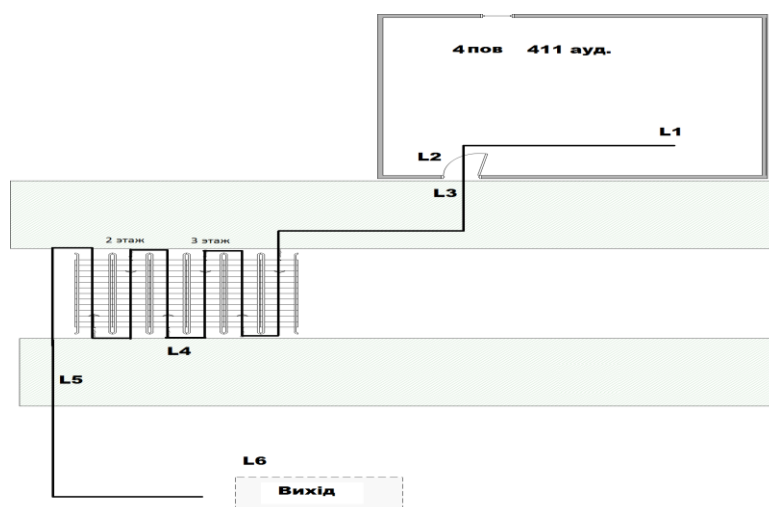


Рисунок 6.3 — Схема евакуації

Допустима тривалість евакуації визначається за формулою:

$$\tau_{\text{дон}} = m \cdot \tau_{n.k.}, \quad (6.4)$$

де $\tau_{n.k.}$ — критична тривалість пожежі при евакуації, хв.;

m — коефіцієнт безпеки, залежить від ступеня протипожежного захисту будівлі, його призначення й властивостей горючих речовин, що утворюються у виробництві або є предметом обстановки приміщень або їх обробки.

					РБ101.17010201.072.007 ПЗ	Арк.
						135
Змн.	Арк.	№ докум.	Підпис	Дата		

Значення коефіцієнта m рекомендується встановлювати в залежності від ступеня надійності засобів протипожежного захисту які розглядаються в будівлі. У виробничих будівлях при відсутності засобів автоматичного гасіння та оповіщення про пожежу $m = 1,0$.

Критична тривалість пожежі для будівлі в цілому встановлюється в залежності від часу проникнення продуктів горіння і можливої втрати видимості в комунікаційних приміщеннях, розташованих до виходу з будівлі .

Розрахунковий час евакуації людей (t_p) слід визначати як суму часу руху людського потоку по окремих ділянках шляху t_f [39]:

$$t_p = t_1 + t_2 + t_3 + \dots + t_i, \quad (6.5)$$

де $t_{н.е.}$ — час затримки початку евакуації;

t_1 — час руху людського потоку на першій ділянці, хв. Приймається 4,1 хв з урахуванням того, що будівля не має автоматичної системи сигналізації та оповіщення про пожежу;

t_2, t_3, t_i — час руху людського потоку на кожній з наступних після першої ділянках шляху, хв.

При розрахунку весь шлях руху людського потоку підрозділяється на ділянки (прохід, коридор, дверний проріз, сходовий марш, тамбур) довжиною і шириною b_j . Початковими ділянками є проходи між робочими місцями, обладнанням, рядами крісел і т.п [40].

Для визначення часу руху людей з урахуванням габаритних розмірів аудиторії , визначається щільність руху людського потоку на першій ділянці за формулою (6.6):

$$D_1 = \frac{N_1 \cdot f}{L_1 \cdot b_1}, \quad (6.6)$$

					РБ101.17010201.072.007 ПЗ	Арк.
						1 3 6
Змн.	Арк.	№ докум.	Підпис	Дата		

де N_1 — число людей на першій ділянці, чол.;

f — середня площа горизонтальної проекції людини, приймаємо згідно, $m^2/чол.$;

L_1, b_1 — довжина та ширина першої ділянки шляху, м.

Підставимо значення до формули (6.7) та отримуємо:

$$D_1 = \frac{38 \cdot 1}{10} = 3,8_{чол} / m^2. \quad (6.7)$$

Залежності швидкості та інтенсивності руху від щільності людського потоку, згідно з, представлені в таблиці 6.5.

Таблиця 6.5 — Залежність швидкості та інтенсивності руху від щільності людського потоку

Щільність потоку людей $D, чол$	Горизонтальний шлях		Шлях через двері $q,$ $м/хв$	Сходи вниз		Сходи вверх	
	$V,$ $м/хв$	$q>$ $м/хв$		$V,$ $м/хв$	$q,$ $м/хв$	$V,$ $м/хв$	$q,$ $м/хв$
0,01	100	1,0	1,0	100	1,0	60	0,6
0,05	100	5,0	5,0	100	5,0	60	3,0
0,1	80	8,0	8,7	95	9,5	53	5,3
0,2	60	12,0	13,4	68	13,6	40	8,0
0,3	47	14,1	15,6	52	16,6	32	9,6
0,4	40	16,0	18,4	40	16,0	26	10,4
0,5	33	16,5	19,6	31	15,6	22	11,0
0,6	27	16,2	19,0	24	14,4	18	10,6
0,7	23	16,1	18,5	18	12,6	15	10,5
0,8	19	15,2	17,3	13	10,4	10	10,0
0,9 і більше	15	13,5	8,5	10	7,2	8	9,9

Час руху людського потоку по першій ділянці шляху (t), хв, обчислюють за формулою:

$$t_1 = \frac{L_1}{V_1} = \frac{10}{15} = 0,66 \text{ хв.}, \quad (6.8)$$

де L_1 — довжина першої ділянки шляху, м.

V_1 — значення швидкості руху людського потоку по горизонтальному шляху на першій ділянці, визначається в залежності від відносної щільності D , чол, по таблиці 6.6.

Час руху в проході визначається за формулою (6.10):

$$D_2 = \frac{N_2 \cdot f}{L_2 \cdot b_2} = 0 \quad (6.9)$$

$$t_2 = \frac{L_2}{V_2} = 0 \text{ хв.} \quad (6.10)$$

Час руху по сходах вниз:

$$t_3 = \frac{L_3}{V_3} = \frac{6}{1,5} = 0,4 \text{ хв.} \quad (6.11)$$

Щільність людського потоку для поверху:

$$D_4 = \frac{N_4 \cdot f}{L_4 \cdot b_4} = \frac{38 \cdot 1}{66} = 0,57 \text{ м / хв.} \quad (6.12)$$

					РБ101.17010201.072.007 ПЗ	Арк.
						138
Змн.	Арк.	№ докум.	Підпис	Дата		

Швидкість руху по коридору першого поверху:

$$t_4 = \frac{L_4}{V_4} = \frac{83}{78} = 1,06 \text{ хв.} \quad (6.13)$$

$$t_5 = \frac{L_5}{V_5} = \frac{20}{23} = 0,869 \text{ хв.} \quad (6.16)$$

Розрахуємо час евакуації по формулі (6.17):

$$\begin{aligned} t &= t_1 + t_2 + t_3 + t_4 + t_5 + t_6 = \\ &= 0,66 + 0,4 + 1,06 + 0,869 + 1 + 1 = 4,9 \text{ хв.} \end{aligned} \quad (6.17)$$

В ході виконання розділу було виявлено, що розрахований час евакуації з кабінетів підприємства більше допустимого. Тому будинок, в якому розташовується підприємство, необхідно обладнати системою оповіщення про пожежу, засобами автоматичної сигналізації.

6.4.3 Близкавкозахист

Для попередження можливості проникнення в приміщення під час грози електричного потенціалу, що може привести для ураження людей, псування електрообладнання, виникнення пожежі необхідно використовувати блискавкозахист.

Захисна дія блискавкозахисту заснована на властивості блискавки вражати найбільш високі металевої споруди.

Так як будівля, для якої будуть проведені розрахунки, 6 поверхова, то блискавкозахист необхідний.

					РБ101.17010201.072.007 ПЗ	Арк.
						139
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 6.6 — вихідні дані для розрахунку

Висота приміщення, h_x	h - висота блискавковідводу, м
20	30

Висота конусу захисту розраховується за формулою:

$$h_0 = 0.85h = 0.85 \cdot 30 = 25.5 \text{ м} \quad (6.18)$$

Радіус зони захисту на поверхні землі:

$$r_0 = (1.1 - 0.002 \cdot h) \cdot h = (1.1 - 0.002 \cdot 30) \cdot 30 = 31.2 \text{ м} \quad (6.19)$$

Радіус зони захисту на висоті h_x :

$$r_x = (1.1 - 0.02 \cdot h) \cdot \left(h - \frac{h_x}{0.85}\right) = (1.1 - 0.02 \cdot 30) \cdot \left(30 - \frac{20}{0.85}\right) = 3.2 \text{ м} \quad (6.20)$$

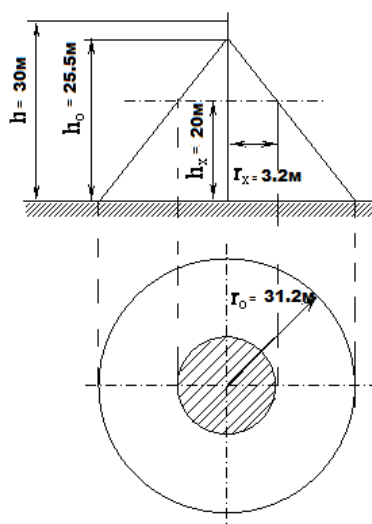


Рисунок 6.4 — Зона захисту одиночного стрижневого блискавковідводу

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 0
Змн.	Арк.	№ докум.	Підпис	Дата		

Де з рисунку 6.4:

h — висота блискавковідводу;

h_0 — висота конусу захисту;

h_x — висота споруди;

r_0 — радіус зони захисту на поверхні землі;

r_x — радіус зони захисту на висоті h_x .

З рисунку 6.6. видно, що радіусу r_0 нам вистачає, для забезпечення блискавкозахисту на рівні першого поверху. На даху недостатній блискавкозахист.

6.5 Заходи для усунення шкідливого впливу виявлених факторів

Стосовно пожежної безпеки по-перше має бути складений план евакуації з поверху (який має бути вивішений на видному місці для ознайомлення всіх працівників); по-друге мають бути розміщені на видному та легкодоступному місці вугільно-порошкові вогнегасники (не дивлячись на те, що в нас будинок категорії «Д»). А також бажане розміщення на стелі системи протипожежної безпеки, яка б могла в автоматичному режимі спостерігати та при необхідності загасити пожежу (ще на стадії займання) без втручання людини [41].

Умови праці на робочому місці аналізуються за наступними факторами:

1. Понижена температура в лабораторії через недостатню систему опалення лабораторії необхідно поліпшити систему опалення приміщень.
2. Недостатня освітленість на робочому місці внаслідок використання недостатньо потужних ламп.

Для створення сприятливих умов зорової роботи освітлення робочих приміщень повинне задовольняти таким умовам:

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 1
Змн.	Арк.	№ докум.	Підпис	Дата		

- рівень освітленості робочих поверхонь має відповідати гігієнічним нормам для даного виду роботи;
- мають бути забезпечені рівномірність та часова стабільність рівня освітленості у приміщенні, відсутність різких контрастів між освітленістю робочої поверхні та навколишнього простору, відсутність на робочій поверхні різких тіней (особливо рухомих);
- у полі зору предмета не повинно створювати сліпучого блиску;
- штучне світло, що використовується на підприємствах, за своїм спектральним складом має наближатися до природного.

3. Підвищений рівень звукового тиску, так як лабораторії працює кулер, принтер та кондиціонер, тобто офісне обладнання, можна змінити на моделі, що призводять менше шуму.

4. Можливим джерелом електричної небезпеки на робочому місці для людей може служити підключений до електричного ланцюга персональний комп'ютер. Необхідно визначити опір вертикальних електродів та їх кількість, що необхідні для заземлення ЕОМ на робочому місці [42].

Дотримання умов, що визначають оптимальну організацію робочого місця оператора ПК, дозволить зберегти хорошу працездатність протягом всього робочого дня, підвищить продуктивність праці.

6.6 Висновки

У даній роботі був виконаний розрахунок штучного освітлення виробничого приміщення, визначено коефіцієнт штучної освітленості, що забезпечує нормовану для даної роботи, величину коефіцієнта штучного освітленості. На основі отриманих розрахунків дійшли висновку, що за даних умов роботи необхідно встановити 1 світильник з лампою ДРЛ-400 для ефективного освітлення приміщення. Виявлено категорію даного приміщення, де розміщено ПК — «Д», тобто знижена пожежна небезпека. Проведено

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 2
Змн.	Арк.	№ докум.	Підпис	Дата		

розрахунок евакуації людей під час пожежі, в ході виконання якого було виявлено, що розрахований час евакуації з кабінетів підприємства більше допустимого.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 3
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Побудова розглянутої автоматичної системи фіксації порушень ПДР стало можливим в більшій мірі завдяки розвитку сучасних методів і алгоритмів комп'ютерного зору і зростання продуктивності персональних комп'ютерів. Зібрані воедино математичні методи і програмний код дозволяють створити систему забезпечення контролю над безпекою дорожнього руху.

Мета даної дипломної роботи — реалізувати комплекс для розпізнавання номерних знаків з подальшим виводом на створеному сайті.

Визначення найбільш відповідних до конкретного завдання методів обробки зображення і сегментації символів дозволить знизити вимоги до умов розпізнавання, що дуже важливо в задачі розпізнавання номерних знаків транспортних засобів.

В рамках проведених робіт виконано:

1. На підставі аналізу літературних джерел виділено основні етапи реалізації та алгоритми, які можуть бути використані при проектуванні заданої системи.

2. Оцінені вимоги до програмного забезпечення, виконаний пошук функціонально подібних програмних продуктів та проведено аналіз переваг і недоліків.

3. Проведений аналіз бібліотек для ALPR.

4. Досліджен метод детекції номерного знаку з бібліотеки OpenALPR.

5. Зкомпільовані всі необхідні бібліотеки.

6. Розроблен веб-інтерфейс для віддаленого управління та моніторингу, заснованого на Twitter Bootstrap.

7. Зроблено оцінку ефективності розпізнавання номерних символів для зображень та відео. Результати цілком задовольняють заданим вимогам, з

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 4
Змн.	Арк.	№ докум.	Підпис	Дата		

урахуванням можливого впливу навколишнього середовища на зображенні або при відеозйомці.

8. Результати розпізнавання динамічно оновлюються на сайті.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. J.L. Alty «computer science» (англ.) // International Encyclopedia of Information and Library Science. — 2002. — с. 98-101.
2. И.С. Личканенко. Методы обработки изображений и распознавания образов для задачи обнаружения номерных знаков транспортных средств // Информатика и компьютерные технологии — 2013 — Донецк: Донецкий национальный технический университет, 2013
3. Авто-Инспектор — система распознавания автомобильных номеров [Электронный ресурс]. — Режимдоступа:
<http://www.iss.ru/products/intelligent/auto/>
4. АПК «АВТОУРАГАН» [Электронный ресурс]. — Режим доступа:
<http://www.recognize.ru/node/23>
5. Fahmy M.M.M., 1994, Automatic Number plate Recognition : Neural Network Approach, Proceedings of VNIS'94 Vehicle Navigation and Information System Conference, 3 1 Aug-2 Sept, pp.291-296, 1994
6. Nijhuis J.A.G., Brugge Ter M.H., Helmholt K.A., Pluim J.P.W., Spaanenburg L., Venema L., Westenberg M.A., 1995, Car License PlateRecognition with Neural Networks and Fuzzy Logic, IEEE International Conference on Neural Networks, pp.2232-2236, 1995
7. Lotufo R.A., Morgan A.D., and Johnson AS., 1990, Automatic Number-Plate Recognition, Proceedings of the IEE Colloquium on Image analysis for Transport Applications, V01.035, pp.6/1-6/6, February 16, 1990.
8. Буземан Г., Келли П. Проективная геометрия и проективные метрики. М., 1957.
9. Бэр Р. Линейная алгебра и проективная геометрия. М., 1955.
10. Юнг Дж. В. Проективная геометрия. М.: ИЛ, 1949.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 6
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Kim S.K., Kim D.W., and Kim H.J., 1996, A Recognition of Vehicle License Plate Using a Genetic Algorithm Based Segmentation, Proceedings of 3rd IEEE International Conference on Image Processing, V01.2., pp.661-664, 1996

12. Hontani H., and Koga T., (2001), Character extraction method without prior knowledge on size and information, Proceedings of the IEEE International Vehicle Electronics Conference (IVEC'01), pp. 67-72.

13. Система распознавания автомобильных номеров «НомерОк» [Электронный ресурс]. — Режим доступа: <http://avtonomerok.com/>

14. Животченко О.В. Реферат по теме выпускной работы: Разработка компьютерной системы обработки изображений с камер видеонаблюдения для определения номерного знака транспортного средства. — Портал магистров ДонНТУ [Электронный ресурс]. — Режим доступа: <http://masters.donntu.edu.ua/2012/fknt/zhivotchenko/diss/index.htm>

15. Федоров А.В. Реферат по теме выпускной работы: Исследование методов контурной сегментации для построения системы оптического распознавания символов. — Портал магистров ДонНТУ [Электронный ресурс]. — Режим доступа: <http://masters.donntu.edu.ua/>

16. Грузман И.С., Киричук В.С. Цифровая обработка изображений в информационных системах: Учебное пособие. — Новосибирск: Изд-во НГТУ, 2002. — 352 с.

17. Воскресенский Е.М., Царев В.А. Моделирование и адаптация систем распознавания текстовых меток на видеоизображениях. — Череповец: ИНЖЭКОН-Череповец. — 2009. — 154 с.

18. Математическая морфология [Электронный ресурс]. — Режим доступа: <http://habrahabr.ru/post/113626/>

19. Путятин Е.П. Нормализация и распознавание изображений [Электронный ресурс]. — Режим доступа: <http://sumschool.sumdu.edu.ua/is-02/rus/lectures/pytyatin/pytyatin.htm>

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 7
Змн.	Арк.	№ докум.	Підпис	Дата		

20. Абраменко А. Компьютер читает [Электронный ресурс]. — Режим доступа: <http://www.ocrai.narod.ru/fr.html>

21. Брилюк Д.В., Старовойтов В.В. Нейросетевые методы распознавания изображений [Электронный ресурс]. — Режим доступа: http://rusnauka.narod.ru/lib/author/briluk_d_b/1/

22. Тельнов Ю.Ф. Интеллектуальные информационные системы / Московский государственный университет экономики, статистики и информатики. — М.: МЭСИ, 2003.

23. Шитиков В.К., Розенберг Г.С., Зинченко Т.Д. Количественная гидроэкология: методы системной идентификации. — Тольятти: ИЭВБ РАН, 2003. — 463 с.

24. Аль-Рашайда Хасан Хусейн. Исследование и разработка методов локализации, идентификации и распознавания арабских символов (на примере номерного знака автомобиля). — СПб.: ЛЭТИ, 2008 — 18 с.

25. Антощук С., Крилов В., Бабілунга О. Ієрархічна модель контурної сегментації зображень // Праці 8-ї Міжнародн. конф. «Оброблення сигналів і зображень (УкрОБРАЗ'2006)».- Київ: НАН України — Інститут кібернетики. — 2006. — с.109

26. ДСТУ 3650:2004 [Электронный ресурс]. — Режим доступа: http://vnsvu.com.ua/?page_id=336

27. Гонсалес Р., Вудс Р. Цифровая обработка изображений. — М.: Техносфера, 2005. — 1072 с.

28. Дуда Р., Харт П. «Распознавание образов и анализ сцен» — М.: Мир, 1976.

29. Копитчук М.Б., Олещук О.В. Попередня обробка зображень // Праці 6-ї Міжнародн. конф. «Оброблення сигналів і зображень (УкрОБРАЗ'2002)». — Київ: НАН України — Інститут кібернетики. — 2002. — с.127-130.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 8
Змн.	Арк.	№ докум.	Підпис	Дата		

30. Уидроу Б., Стирнз С. Адаптивная обработка сигналов. — М.: "Радио и связь", 1989.

31. Методы компьютерной обработки изображений / под ред. Сойфера В.А. — 2-е изд., испр. — М.: ФИЗМАТЛИТ, 2003. — 784 с.

32. Распознавание изображений (источник: газета «Компьютер-Информ»)/

<http://www.masters.donntu.edu.ua/2006/fvti/poltava/library/article5.htm>

33. Федяев О.И., Фёдоров А.В. Анализ методов контурной сегментации изображения // Сборник материалов четвёртой международной научно-технической конференции студентов, аспирантов и молодых ученых "Информатика та комп'ютерні технології".— Донецк — ДонНТУ — 2008. — 64-65 с.

34. Федяев О.И., Фёдоров А.В. Сравнительный анализ методов контурной сегментации в системах оптического распознавания символов» // Сборник материалов пятой всеукраинской научно-технической конференции студентов, аспирантов и молодых ученых "Комп'ютерний моніторинг та інформаційні технології".— Донецк — ДонНТУ, 2009. — 120-121 с.

35. Основы охраны праці: Навч. посіб. / В.В. Березуцький, Т.С. Бондаренко, Г.Г. Валенко та ін.; За заг. ред. В.В.Березуцького. — 2-ге вид., перероб. і доп. — Х.: Факт, 2007. — 480 с.

36. ГОСТ 12.1.005-88 ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны. Введен 01.07.89

37. СНиП II-4-79. Строительные нормы и правила. Естественное и искусственное освещение. Нормы проектирования — М.: Стройиздат, 1980. — 48 с

38. Баклашов Н.И. и др. Охрана труда на предприятиях связи и охрана окружающей среды. — М., 1989. — 288 с.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 4 9
Змн.	Арк.	№ докум.	Підпис	Дата		

39. Безопасность жизнедеятельности. Производственная безопасность и охрана труда / П.П. Кукин, В.Л. Лапин, Н.Л. Пономаренко и др. – М.: Высш. шк., – 2001. – 431 с.

40. ГОСТ 12.1.007-76 ССБТ. Вредные вещества. Классификация и общие требования безопасности

41. Пожежна безпека: Нормативні акти та інші документи: У 4 т. – К.: Основи, 1997–1998

42. ДБН В.1.1-7-2002. Захист від пожежі. Пожежна безпека об'єктів будівництва. – Діє з 01.01.03

					РБ101.17010201.072.007 ПЗ	Арк.
						150
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАДОК А

Лістинг програми Front-end частини сайту

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="shortcut icon" href="assets/img/favicon.jpg" />
  <title>ALPR Systems</title>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="">

  <!-- Bootstrap core CSS -->
  <link href="assets/css/bootstrap.css" rel="stylesheet">

  <!-- Custom styles for this template -->
  <link href="assets/css/style.css" rel="stylesheet">
  <link href="assets/css/font-awesome.min.css" rel="stylesheet">
  <link href="assets/js/fancybox/jquery.fancybox.css" rel="stylesheet" />
</head>

<body data-spy="scroll" data-offset="0" data-target="#theMenu">

  <!-- Menu -->
  <nav class="menu" id="theMenu">
    <div class="menu-wrap">
      <h1 class="logo"><a href="index.html#home">ALPR Systems</a></h1>
      <i class="fa fa-times menu-close"></i>
      <a href="#home" class="smoothScroll">Home</a>
      <a href="assets/login.html" class="smoothState">Login</a>
      <a href="assets/tracker.html" class="smoothScroll">Tracker</a>
      <a href="#about" class="smoothScroll">About</a>
      <a href="#services" class="smoothScroll">Services</a>
      <a href="#contact" class="smoothScroll">Contact</a>
      <a href="#"><i class="fa fa-facebook"></i></a>
      <a href="#"><i class="fa fa-twitter"></i></a>
      <a href="#"><i class="fa fa-vk"></i></a>
      <a href="#"><i class="fa fa-envelope"></i></a>
    </div>

    <!-- Menu button -->
    <div id="menuToggle"><i class="fa fa-bars"></i></div>
  </nav>

  <section id="home" name="home"></section>
  <div id="headerwrap">
    <div class="container">
      <div class="row">
        <div class="col-md-6 col-md-offset-3">
          <h1>ALPR SYSTEMS</h1>
        </div>
      </div><!--/row -->
    </div><!--/container -->
  </div><!--/headerwrap -->
```

					РБ101.17010201.072.007 ПЗ	Арк.
						151
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<section id="about" name="about"></section>
<div id="aboutwrap">
  <div class="container">
    <div class="row">
      <div class="col-lg-4 name">
        
        <p>X Z I B I T</p>
        <div class="name-label"></div>
      </div><!--/col-lg-4-->
      <div class="col-lg-8 name-desc">
        <h2>POWERFUL PRODUCT & <br/>YOUR'S PROTECTION <br/>WITH AWESOME ALPR
Systems </h2>
        <div class="name-zig"></div>

        <div class="col-md-6">
          <p>Know when unauthorized persons are on your property.</p>
          <p>The moment an unwelcome visitor drives past your camera we'll
send you an alert.se a more cost-effective and accurate approach to identify
suspicious activity.ALPR Systems monitors all your camera feeds 24 hours a
day, seven days a week.</p>

          <p>Every time someone drives past one of your cameras, ALPR
Systems records it to a database.With a simple search, you can see the full
history of a vehicle as it drives through your property.</p>
        </div>
        <div class="col-md-6">
          <p>Use the ALPR Systems and directly integrate ALPR Systems
leading license plate recognition technology into your applications.ALPR
Systems is cross-platform and operates in a variety of programming
language.</p>

        </div>

      </div><!--/col-lg-8-->
    </div><!-- /row -->
  </div><!-- /container -->
</div><!-- /aboutwrap -->

<!--ABOUT SEPARATOR -->
<div class="sep about" data-stellar-background-ratio="0.5"></div>

<!--Improve SECTION -->
<section id="portfolio" name="portfolio"></section>
<div id="portfoliowrap">
  <div class="container">
    <div class="row">
      <h1>IMPROVE SECURITY WITH ALPR SYSTEMS</h1>
      <center>
        <h3>Monitor license plates 24x7 at all of your facilities</h3>
      </center>
      <div class="col-md-6 col-md-offset-3 desc">
        <div class="project-wrapper">
          <div class="project">
            <div class="photo-wrapper">
              <div class="photo">
                <a class="fancybox" href="assets/img/portfolio/gg.jpg"></a>
              </div>
              <div class="overlay">
                </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

					РБ101.17010201.072.007 ПЗ	Арк.
						1 5 2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        </div>
    </div>
</div>
</div>
</div> <!--/container -->

<div class="container">
    <div class="row mt centered">
        <h1>OUR SEGGESTIONS</h1>
        <div class="col-lg-4 proc">
            <i class="fa fa-coffee"></i>
            <h3>The work</h3>
            <p>You could have rest, when the system is working. Minimum
strength, maximum joy!</p>
        </div>
        <div class="col-lg-4 proc">
            <i class="fa fa-cogs"></i>
            <h3>The Ideas</h3>
            <p>Everyday we have new improvments about our product.</p>
        </div>
        <div class="col-lg-4 proc">
            <i class="fa fa-heart"></i>
            <h3>The Product</h3>
            <p>Our product is very usefull on the market place, and we hope,
that we really could help YOU with your SECURITY</p>
        </div>

    </div><!--/row -->
</div><!--/container -->
</div><!--/Portfoliowrap -->

<!--SEPARATOR -->
<div class="sep portfolio" data-stellar-background-ratio="0.5"></div>

<!--SERVICE SECTION -->
<section id="services" name="services"></section>
<div id="servicewrap">
    <div class="container">
        <div class="row">
            <div class="col-lg-8-offset-2 centered">
                <h1>AN OVERVIEW</h1>
                <h3>We'll do all the work for you</h3>
            </div><!-- /col-lg-8 -->
        </div><!--/row -->

        <div class="row mt">
            <div class="col-lg-3 service">
                <i class="fa fa-star"></i><p>PREMIUM QUALITY<br/><small>ALPR
Systems</small></p>
                <p class="text">НАПИСАТЬ</p>
            </div>
            <div class="col-lg-3 service">
                <i class="fa fa-cloud"></i><p>CLOUD SERVICES<br/><small>ALPR
Systems</small></p>
                <p class="text">НАПИСАТЬ</p>
            </div>
            <div class="col-lg-3 service">
                <i class="fa fa-shield"></i><p>SECURED ACCOUNTS<br/><small>ALPR
Systems</small></p>
                <p class="text">НАПИСАТЬ</p>
            </div>
        </div>
    </div>
</div>

```

					РБ101.17010201.072.007 ПЗ	Арх.
						153
Змн.	Арх.	№ докум.	Підпис	Дата		

```

        </div>
        <div class="col-lg-3 service">
            <i class="fa fa-heart"></i><p>100% SATISFACTION<br/><small>ALPR
Systems</small></p>
            <p class="text">НАПИСАТЬ</p>
        </div>
    </div><!--/row -->
    <div class="row mt">
        <div class="col-lg-3 service">
            <i class="fa fa-trophy"></i><p>PREMIUM QUALITY<br/><small>ALPR
Systems</small></p>
            <p class="text">НАПИСАТЬ</p>
        </div>
        <div class="col-lg-3 service">
            <i class="fa fa-globe"></i><p>CLOUD SERVICES<br/><small>ALPR
Systems</small></p>
            <p class="text">НАПИСАТЬ</p>
        </div>
        <div class="col-lg-3 service">
            <i class="fa fa-lock"></i><p>SECURED ACCOUNTS<br/><small>ALPR
Systems</small></p>
            <p class="text">НАПИСАТЬ</p>
        </div>
        <div class="col-lg-3 service">
            <i class="fa fa-thumbs-up"></i><p>100% SATISFACTION<br/><small>ALPR
Systems</small></p>
            <p class="text">НАПИСАТЬ</p>
        </div>
    </div><!--/row -->

</div><!--/container -->
</div><!--/servicewrap -->

<div id="testimonials">
    <div class="container">
        <div class="row">
            <div class="col-lg-8 col-lg-offset-2 mt">

                <div id="carousel-example-generic" class="carousel slide" data-
ride="carousel">
                    <!-- Wrapper for slides -->
                    <div class="carousel-inner">
                        <div class="item active mb centered">
                            <h3>MARK WEBBER, Property Manager, Melville, NY</h3>
                            <p>We've opened up a new line of business that relies heavily
on ALPR Systems. Sales of the product have boosted our revenue by 8% in the
last six months.</p>
                            <p></p>
                        </div>

                        <div class="item mb centered">
                            <h3>PAUL LEVINGSTON, Senior Product Manager, London, UK</h3>
                            <p>ALPR Systems alerted us when a thief came back to steal
from one of our tenants. We called the police, and they were waiting for him
with handcuffs when he tried to leave.</p>
                            <p></p>
                        </div>

                        <div class="item mb centered">

```

					РБ101.17010201.072.007 ПЗ	Арх.
						154
Змн.	Арх.	№ докум.	Підпис	Дата		

```

        <h3>LUCY LENNIN, Facilities Director, Kansas City, MO</h3>
        <p>We had an employee show up hours late for work every day
for months. ALPR Systems gave us the proof we needed to correct that
behavior.</p>
        <p></p>
    </div>

</div>
<!-- Indicators -->
<ol class="carousel-indicators">
    <li data-target="#carousel-example-generic" data-slide-to="0"
class="active"></li>
    <li data-target="#carousel-example-generic" data-slide-
to="1"></li>
    <li data-target="#carousel-example-generic" data-slide-
to="2"></li>
</ol>
</div>

</div><!--/col-lg-8 -->
</div><!--/row -->
</div><!--/container -->
</div><!--/testimonials -->

<!--SERVICE SECTION -->
<section id="contact" name="contact"></section>
<!--CONTACT SEPARATOR -->
<div class="sep contact" data-stellar-background-ratio="0.5"></div>

<div id="contactwrap">
    <div class="container">
        <div class="row">
            <div class="col-lg-6">
                <p>CONTACT US RIGHT NOW!</p>
                <p>Every advice, complaint and improvment is important for us!</p>
                <p>
                    <small><a href="#"><i class="fa fa-home"></i></a> Deribasovka 1,
Odessa, Ukraine </small><br/>
                    <small><a href="#"><i class="fa fa-mobile-phone"></i></a> Tel.
+380482395641</small><br/>
                    <small><a href="#"><i class="fa fa-envelope"></i></a>
alprsystems@gmail.com</small><br/>
                    <small><a href="#"><i class="fa fa-skype"></i></a>
ALPRSystem</small>
                </p>
                <p>
                    <small><a href="#"><i class="fa fa-copyright"></i></a> 2015 Maryna
Nalbandian's project</small><br/>
                    ONPU
                </p>
            </div>

            <div class="col-lg-6">
                <form role="form">
                    <div class="form-group">
                        <label for="exampleInputName1">Your Name</label>
                        <input type="email" class="form-control" id="exampleInputName1"
placeholder="Enter name">
                        <label for="exampleInputEmail1">Email address</label>

```

					РБ101.17010201.072.007 ПЗ	Арк.
						155
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <input type="email" class="form-control" id="exampleInputEmail1"
placeholder="Enter email">
        <label for="exampleInputText1">Message</label>
        <textarea class="form-control" id="exampleInputText1"
rows="3"></textarea>
        </div>
        <button type="submit" class="btn btn-default">Submit</button>
    </form>
</div>

    </div><!--/row -->
</div><!-- /container -->
</div>

<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script type="text/javascript"
src="assets/js/fasw.transitions.min.js"></script>
<script type="text/javascript" src="assets/js/jquery.min.js"></script>
<script type="text/javascript" src="assets/js/jquery-latest.js"></script>
<script type="text/javascript"
src="assets/js/fancybox/jquery.fancybox.js"></script>
<script type="text/javascript"
src="assets/js/jquery.stellar.min.js"></script>
<script type="text/javascript" src="assets/js/classie.js"></script>
<script type="text/javascript" src="assets/js/bootstrap.min.js"></script>
<script type="text/javascript" src="assets/js/smoothscroll.js"></script>
<script type="text/javascript" src="assets/js/main.js"></script>
<script>
    $(function(){
        $.stellar({
            horizontalScrolling: false,
            verticalOffset: 40
        });
    });
</script>
<script type="text/javascript">
    $(function() {
        //    fancybox
        jQuery(".fancybox").fancybox();
    });
</script>

</body>
</html>

```

login.html

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <link rel="shortcut icon" href="img/favicon.jpg" />
        <title>Member Login</title>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <!-- Bootstrap -->

```

					РБ101.17010201.072.007 ПЗ	Арк.
						156
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<link
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
rel="stylesheet">

<body style="background:#eee;">

  <div class="container">
    <p><br/></p>
    <div class="row">
      <div class="col-md-8"></div>
      <div class="col-md-4">
        <div class="panel panel-default">
          <div class="panel-body">
            <div class="page-header">
              <h3>Login Area</h3>
            </div>
            <form role="form">
              <div class="form-group">
                <label
for="exampleInputEmail1">Email or Username</label>
                <div class="input-group">
                  <span class="input-
group-addon"><span class="glyphicon glyphicon-user"></span></span>
                  <input type="email"
class="form-control" id="exampleInputEmail1" placeholder="Enter email">
                </div>
              </div>
              <div class="form-group">
                <label
for="exampleInputPassword1">Password</label>
                <div class="input-group">
                  <span class="input-
group-addon"><span class="glyphicon glyphicon-star"></span></span>
                  <input type="password"
class="form-control" id="exampleInputPassword1" placeholder="Password">
                </div>
              </div>
              <hr/>
              <button type="button" class="btn
btn-success"><span class="glyphicon glyphicon-arrow-left"></span>
Back</button>
              <button type="submit" class="btn
btn-primary"><span class="glyphicon glyphicon-lock"></span> Login</button>
            <p><br/></p>
          </form>
        </div>
      </div>
    </div>
  </div>

  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></scri
pt>
  <!-- Include all compiled plugins (below), or include individual files as
needed -->
  <script
src="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></scr
ipt>
</body>

```

					РБ101.17010201.072.007 ПЗ	Арк.
						157
Змн.	Арк.	№ докум.	Підпис	Дата		

</html>

tracker.html

```
<!DOCTYPE HTML>
<html lang="en">

<head>
  <link rel="shortcut icon" href="img/favicon.jpg" />
  <title>Tracker</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="css/bootstrap.css" type="text/css"/>
  <link rel="stylesheet" href="css/dataTables.bootstrap.css" type="text/css">
  <link rel="stylesheet" href="css/dataTables.colVis.css" type="text/css">
  <link rel="stylesheet" href="css/jquery.dataTables.css" type="text/css">
  <link rel="stylesheet" href="css/dataTables.scroller.css" type="text/css">

  <script type="text/javascript" src="js/jquery-1.11.0.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.js"></script>
  <script type="text/javascript" src="js/alpr-common.js"></script>
  <script type="text/javascript" src="js/alpr-tracker.js"></script>
  <script type="text/javascript" src="js/jquery.dataTables.min.js"></script>
  <script type="text/javascript" src="js/dataTables.bootstrap.js"></script>
  <script type="text/javascript" src="js/dataTables.colVis.js"></script>
  <script type="text/javascript" src="js/dataTables.scroller.js"></script>

</head>

<body>
  <div class="well">
    <button class="btn btn-danger"><i class="glyphicon glyphicon-trash"></i>
Delete</button>
    <button class="btn btn-warning"><i class="glyphicon glyphicon-heart-
empty"></i> Leave Comment</button>
    <button class="btn btn-success"><i class="glyphicon glyphicon-
refresh"></i> Refresh</button>
    <div class="col-xs-1">
      
    </div>
  </div>
  <div id="alpr-results-add"></div>
</body>

</html>
```

alpr-tracker.js

```
var WS_PROTOCOL = 'http'
var WS_HOST = '10.0.0.1'
var WS_PORT = '9000'
var WS_API_VERSION = 'v1/'
var WS_ROUTE = '/api/'
var WS_URL = WS_PROTOCOL + '://' + WS_HOST + ':' + WS_PORT + WS_ROUTE +
WS_API_VERSION

var TBL_HEADER_ARR = [];
TBL_HEADER_ARR[0] = ['id', 'Id'];
TBL_HEADER_ARR[1] = ['date', 'Date'];
```

					РБ101.17010201.072.007 ПЗ	Арк.
						158
Змн.	Арк.	№ докум.	Підпис	Дата		

```

TBL_HEADER_ARR[2] = ['plate', 'Plate Number'];
TBL_HEADER_ARR[3] = ['proc_time_ms', 'Elapsed Time'];
TBL_HEADER_ARR[4] = ['confidence', 'Best Guess'];
TBL_HEADER_ARR[5] = ['hash', 'Photo'];

$(document).ready(function () {
    getResults();
});

function getResults() {
    $('#alpr-results-add').empty();
    var results = [];
    var tblTitle = 'ALPR Results';
    var tblId = 'alpr-results';

    var tblHeaders = '';
    for (var i = 0; i < TBL_HEADER_ARR.length; i++) {
        tblHeaders += '<th>' + TBL_HEADER_ARR[i][1] + '</th>';
    }

    var tblHtml = '';

    tblHtml += '<h3>' + tblTitle + '</h3>';
    tblHtml += '<div class="table-responsive" style="overflow-x: auto; overflow-y: auto;">';
    tblHtml += '<table id="' + tblId + '" cellpadding="0" cellspacing="0" border="0" width="100%" class="table table-striped table-bordered table-hover table-condensed">';
    tblHtml += '<thead>';
    tblHtml += '<tr>';
    tblHtml += tblHeaders;
    tblHtml += '</tr>';
    tblHtml += '</thead>';
    tblHtml += '<tfoot>';
    tblHtml += '<tr>';
    tblHtml += tblHeaders;
    tblHtml += '</tr>';
    tblHtml += '</tfoot>';
    tblHtml += '<tbody></tbody>';
    tblHtml += '</table>';
    tblHtml += '</div>';

    $('#alpr-results-add').html(tblHtml);

    $('##' + tblId + ' tfoot th').each( function () {
        var title = $('##' + tblId + ' tthead th').eq( $(this).index() ).text();
        $(this).html( '<input type="text" placeholder="Search '+title+'" />' );
    });

    var table = $('##' + tblId).DataTable({
        dom: 'C<"clear">lfrtip',
        stateSave: true,
        // save the state for the session only
        "stateDuration": -1,
        "pagingType": "full_numbers",
        "columns": [
            { "data": TBL_HEADER_ARR[0][0] },
            { "data": TBL_HEADER_ARR[1][0] },
            { "data": TBL_HEADER_ARR[2][0] },
            { "data": TBL_HEADER_ARR[3][0] },

```

					РБ101.17010201.072.007 ПЗ	Арк.
						159
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        "render": function (data, type, full, meta) {
            ret = parseInt(data).toFixed(0) + " ms";
            return ret;
        }
    },
    { "data": TBL_HEADER_ARR[4][0],
      "render": function (data, type, full, meta) {
          ret = data.toFixed(2) + "%";
          return ret
      }
    },
    { "data": TBL_HEADER_ARR[5][0],
      "render": function (data, type, full, meta) {
          ret = "<img src=/images/" + data[0] + "/" + data + "
width='100' height='100'>";
          return ret;
      }
    }
  ],
  colVis: {
    restore: "Restore",
    showAll: "Show all"
  }
});

var reqUrl = WS_URL + 'results'

function draw_data() {
    $.getJSON( reqUrl, function(json) {
        if (json.length) {
            table.clear();
            table.rows.add(json).draw();
        }
    })
}

setInterval(draw_data, 1000);

return table;
}

```

main.js

```

;(function(){

    // Menu settings
    $('#menuToggle, .menu-close').on('click', function(){
        $('#menuToggle').toggleClass('active');
        $('body').toggleClass('body-push-toleft');
        $('#theMenu').toggleClass('menu-open');
    });

})(jQuery)

```

Рисунок А.1 — Front-end частина сайту

					РБ101.17010201.072.007 ПЗ	Арк.
						160
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАДОК Б

Лістинг програми Back-end частини сайту

app.py

```
#!/usr/bin/env/python2

# Application Library imports
from bottle import Bottle, request, static_file
from bottle_sqlite import Plugin
import setup as setup_module

# Standard Library imports
from json import dumps, loads
from hashlib import sha1
from functools import partial
from imghdr import what
import logging

logging.basicConfig(filename='/tmp/log.txt', format=logging.BASIC_FORMAT)

app = Bottle()
database = './alpr.db'
plugin = Plugin(dbfile=database)
app.install(plugin)

# Handle main page and static
@app.route('/')
def main_page():
    return static_file('index.html', root='./')

@app.route('/assets/<filepath:path>')
def server_static(filepath):
    return static_file(filepath, root='./assets/')

@app.route('/images/<filepath:path>')
def server_static(filepath):
    return static_file(filepath, root='./images/')

@app.route('/setup')
def setup():
    setup_module.setup_database(database)
    setup_module.setup_directories()

# Handle API calls
@app.route('/api/v1/results')
def results_list(db):
    results = db.execute('''SELECT
        id, date, plate, proc_time_ms, confidence, hash
    FROM results''').fetchall()
    return dumps([dict(x) for x in results])
```

					РБ101.17010201.072.007 ПЗ	Арк.
						1 6 1
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@app.post('/api/v1/upload')
def upload_result(db):
    upload = request.files.get('img')

    if not what(upload.file):
        raise TypeError("Not an Image")

    # Break the file into multiple of 256-byte (8192, 32768 etc) chunks
    # and feed them to SHA256 consecutively using update()
    # SHA1 - 160
    # Useful for large files
    upload_hash = sha1()
    blocksize = 61440 # 160 * 384
    for chunk in iter(partial(upload.file.read, blocksize), b''):
        upload_hash.update(chunk)
    upload_hash = upload_hash.hexdigest()

    upload.file.seek(0)
    try:
        upload.save('./images/' + upload_hash[0] + '/' + upload_hash)
    except:
        raise
    upload.file.close()

    alpr_json = loads(request.forms.get('data'))
    results = alpr_json["results"]
    logging.error(type(results))

    for result in results:
        logging.error(result["matches_template"])
        if result["matches_template"] == 1:
            db.execute('''INSERT
                        INTO results (plate, proc_time_ms, confidence, hash)
                        VALUES (?, ?, ?, ?)''',
                        (result["plate"], result["processing_time_ms"],
                        result["confidence"], upload_hash, ))
            db.commit()

```

setup.py

```

#!/usr/bin/env python2

import os
import sqlite3

def setup_database(database):

    con = sqlite3.connect(database)

    # Create tables
    con.execute('''CREATE TABLE IF NOT EXISTS results
    (id INTEGER PRIMARY KEY,
    date DATETIME DEFAULT CURRENT_TIMESTAMP,
    plate CHAR(32) NOT NULL,
    proc_time_ms REAL NOT NULL,
    confidence REAL NOT NULL,
    hash CHAR(32) NOT NULL)''')

    con.commit()

```

					РБ101.17010201.072.007 ПЗ	Арк.
						1 6 2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

def setup_directories():
    if not os.path.exists('./images'):
        os.mkdir('./images')

    for h in range(0x10):
        h = format(h, '01x')
        path = './images/' + h
        if not os.path.exists(path):
            os.mkdir(path)

if __name__ == "__main__":
    setup_database('./alpr.db')
    setup_directories()

```

run.py

```

__author__ = 'jeronimo'
from app import app
app.app.run()

```

pyalpr.py

```

import json
import shlex
import subprocess
import requests

class PlateReader:
    def __init__(self):
        # webcam subprocess args
        webcam_command = "fswebcam -r 640x480 -S 20 --no-banner --quiet
alpr.jpg"
        self.webcam_command_args = shlex.split(webcam_command)

        # alpr subprocess args
        alpr_command = "alpr -c ua -p ua -n 3 -j /tmp/alpr.jpg"
        self.alpr_command_args = shlex.split(alpr_command)

    def webcam_subprocess(self):
        return subprocess.Popen(self.webcam_command_args,
                                stdout=subprocess.PIPE)

    def alpr_subprocess(self):
        return subprocess.Popen(self.alpr_command_args,
                                stdout=subprocess.PIPE)

    def alpr_json_results(self):
        # self.webcam_subprocess().communicate()
        alpr_out, alpr_error = self.alpr_subprocess().communicate()

        if not alpr_error is None:
            return None, alpr_error
        elif "No license plates found." in alpr_out:
            return None, None

```

					РБ101.17010201.072.007 ПЗ	Арк.
						1 6 3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        try:
            return json.loads(alpr_out), None
        except ValueError, e:
            return None, e

def read_plate(self):
    alpr_json, alpr_error = self.alpr_json_results()

    if not alpr_error is None:
        print alpr_error
        return

    if alpr_json is None:
        print "No results!"
        return

    # results = alpr_json["results"]
    #
    # ordinal = 0
    # for result in results:
    #     candidates = result["candidates"]
    #
    #     for candidate in candidates:
    #         if candidate["matches_template"] == 1:
    #             ordinal += 1
    #             print "Guess {0:d}: {1:s} {2:.2f}%".format(ordinal,
candidate["plate"], candidate["confidence"])

    url = 'http://localhost:9000/api/v1/upload'
    img = {'img': ('alpr.jpg', open('/tmp/alpr.jpg', 'rb'), 'image/jpeg',
{'Expires': '0'})}
    data = {'name': 'alpr.jpg', 'data': json.dumps(alpr_json)}

    response = requests.post(url, files=img, data=data)

if __name__ == "__main__":
    plate_reader = PlateReader()
    plate_reader.read_plate()

```

Рисунок Б.1 — Back-end частина сайту

					РБ101.17010201.072.007 ПЗ	Арк.
						1 6 4
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК В

Графічні матеріали

1. Реалізація програмно-апаратного комплексу для розпознавання номерних знаків
2. Взаємозв'язок між комп'ютерним зором та ін. галузями
3. Етапи розпізнавання номерних знаків
4. Схема комплексу ALPR
5. Результат проектування
6. Результати експерименту № 1
7. Результати експерименту № 2
8. Висновки

					РБ101.17010201.072.007 ПЗ	Арк.
						1 6 5
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК Г

CD та відомість електронних документів

№	Найменування файлу	Вміст
1	диплом.7z	Дипломна робота
2	source.7z	Код програми
3	Презентація.7z	Презентація диплома
4	плакати.7z	Плакати

Склала _____ ст. гр. РБ-101 Налбандян М.А.

Перевірів _____ к.т.н., доц. каф. ІБ Дмитренко В.С.

					РБ101.17010201.072.007 ПЗ	Арк.
						1 6 6
Змн.	Арк.	№ докум.	Підпис	Дата		