

Всеукраїнський конкурс студентських наукових робіт

Комп'ютерні науки

СТУДЕНТСЬКА НАУКОВА РОБОТА

На тему: «Розробка програмного засобу розпізнавання автомобільних номерів»

Шифр: «Автомобільні номери»

ЗМІСТ

Вступ.....	3
1 Аналіз предметної області і постановка задачі.....	5
1.1 Опис та аналіз сучасного стану проблеми розпізнавання автомобільних номерів	5
2 Опис та аналіз підходів до розробки програмних засобів для розпізнавання автомобільних номерів.....	8
2.1 Виділення контурів на зображенні.....	8
2.2 Попередній пошук області з зображенням номеру методом Віюлі-Джонса.....	10
2.3 Нормування і сегментація.....	10
2.4 Розпізнавання символів з використанням нейромереж	13
3 Проектування і розробка програмного засобу.....	15
3.1 Проектування системи.....	15
3.2 Вибір і обґрунтування технічного і програмного забезпечення.....	17
3.2.1 Вибір і обґрунтування технічного забезпечення.....	17
3.2.2 Вибір і обґрунтування мовних і програмних засобів.....	17
3.3 Розробка компонентів програмного засобу розпізнавання автомобільних номерів.....	19
3.3.1 Опис створення проекту та контейнеризації додатків	19
3.3.2 Розробка серверної частини	23
3.3.3 Розробка додатку веб-інтерфейсу користувача	24
4 Аналіз результатів, отриманих за допомогою програмного засобу.....	26
Висновки.....	28
Перелік посилань.....	29

ВСТУП

Щоденно мільйони людей використовують автомобіль як засіб постійного переміщення. Автомобільна промисловість активно розвивається, і у неї постійно збільшується кількість користувачів. Статистичні дані, зібрані компанією Wards Auto говорять що світі в 1986 році кількість зареєстрованих автомобілів складала близько 500 млн, в 2009 році близько 950 млн, а в 2010 році ця цифра перейшла за 1 млрд. За інформацією AUTO-Consulting станом на 2016 рік, в Україні 202 автомобіля на 1000 жителів. Вроховуючи приблизну кількість населення до 45 млн, в нашій країні близько 8,5 млн зареєстрованих автомобільних засобів. Тому проблема розпізнавання і класифікування автомобільних номерів є більш ніж актуальною.

У даний час бурхливого розвитку комп'ютерних технологій усе більш актуальною задачею є розробка і використання систем розпізнавання образів для різних потреб: визначення і розпізнавання обличчя, знаходження координат місця по зображенню, знаходження конкретних образів на відео та зображенні та обробка отриманих даних. Найбільш широкою галуззю в даній тематиці є розпізнавання текстової інформації, оцифровування та обробка таких даних має величезне значення в теперішньому світі. Одна з підзадач розпізнавання тексту є розпізнавання автомобільних номерів.

Сьогодні такі системи користуються великою популярністю не тільки для правоохоронних органів, але і в різноманітних охоронних системах, підприємствах, які займаються перепродажем автомобілів та просто для будь-кого, хто має бажання визначити правопорушника на дорозі чи знайти володаря автомобільного засобу. Головною перевагою таких систем є можливість отримання необхідної інформації про автомобільний засіб максимально зручним способом.

Інформатизація та автоматизація, як визначна умова переходу до інформаційного суспільства, не лише надає додаткові можливості отримання

інформації про зареєстровані автомобільні засоби, а й суттєво трансформує традиційні процеси й уявлення про можливості отримання та обробки цієї інформації.

Мета виконання дипломної роботи полягає у проведенні досліджень та створенні програмного засобу розпізнавання автомобільних номерів, вивчення існуючих методів розпізнавання та їх оптимізація для використання всім доступними засобами такими як смартфон або персональний комп'ютер.

Розроблюваний програмний засіб для розпізнавання автомобільних номерів повинен виконувати наступні задачі:

- надання веб-інтерфейсу для відправки зображення на сервер;
- прийом через веб-ендпоінт зображення та обробка його на серверній частині, повернення зображення з обведеним знайденим номером на зображенні а також розпізнаний номер текстом;
- можливість паралельно оброблювати декілька зображень.

Актуальність проведення досліджень методів розпізнавання автомобільних номерів та створення спеціалізованого програмного засобу для цього обумовлена актуальністю даної проблеми, а також бурхливим розвитком технологій обробки медійної інформації, з метою удосконалення існуючих методів рішення проблеми, які знаходяться в публічному доступі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Опис та аналіз сучасного стану проблеми розпізнавання автомобільних номерів

Зараз в Україні існує 12 видів реєстраційних знаків транспортних засобів. В них можуть використовуватися українські літери, літери латиниці, цифри а також малюнки[1].

Перший тип можна розділити на 2 підтипи. До першого підтипу належать номерні знаки для легкових та вантажних автомобілів, немаршрутних автобусів та мікроавтобусів, причепів, напівпричепів, причепів-розпусків. Їх розмір складає – 520x112мм, тіло – біле, символи – чорні. Літери ліворуч позначають регіон, праворуч – серію номерного знака. До другого підтипу відносяться номерні знаки для маршрутних автобусів та мікроавтобусів і таксі. Розмір такий же як і у першого підтипу, колір – жовтий, літери – чорні.

Номерні знаки другого виду можна розділити на 3 підтипи. Першому належать знаки для тимчасового використання, до 2-х місяців, на легкових і вантажних автомобілях, автобусах, мікроавтобусах, причепах, напівпричепах, причепах-розпусках. До другого підтипу відносять реєстраційні знаки транспортних засобів, видані торгівельним підприємством, для тимчасового використання (до 10 діб) на легкових і вантажних автомобілях, автобусах та мікроавтобусах, причепах, напівпричепах і причепах-розпусках. Вони мають позначку яка свідчить що знак видається торгівельними організаціями. Такі знаки мають серію в діапазоні Т0 ÷ Т9. Розмір – 520x112мм, тіло – червоне, символи – білі.

До третього типу реєстраційних номерних знаків відносяться номерні знаки для транспортних засобів з об'ємом двигуна, що не перевищує 50 куб.см. Їх розмір складає 140x114мм. Розрізняють 3 підтипи.

Перший тип має біле тіло і чорні символи. До другого підтипу відносять номери для тимчасового використання. Третьому підтипу відносять номерні знаки видані торгівельним підприємством для тимчасового використання. Колір тіла – червоний, символи – білі. Мають серію в діапазоні Т0 ÷ Т9.

Четвертий вид знаків, це номери для автотранспорту дипломатичних представництв, консульств, представництв міжнародних організацій і членів родин їхнього персоналу.

До п'ятого типу номерних знаків відносять знаки для мотоциклів, мотоколясок і моторолерів. Розмір – 220x174мм, тіло біле, символи – чорні.

До шостого типу відносять номерні знаки для тимчасового використання на мотоциклах, мотоколясках і моторолерах.

Сьомий тип номерів розділяють на чотири підтипи. Це номерні знаки які створюють за особистим замовленням.

До восьмого типу відносять номери для тракторів (самохідних машин) та причепів до них, що використовуються в сільському господарстві. Видаються Держтехнаглядом Мінагрополітики.

Дев'ятий тип номерів розділяють на 2 підтипи. До першого належать номерні знаки для автомобілів і причепів до них та тракторів (самохідних машин), що належать ЗСУ, ДССТ, ДПСУ, ДСНС, ВВ МВС, ТСОУ, Національній Гвардії. Також існують номери особливого виду, наприклад номерні знаки посадових осіб ВМС мають зображення прапорця, на якому по кількості зірок можна визначити ранг посадової особи. До другого підтипу належать номерні знаки для добровольчих батальйонів та волонтерських організацій, що допомагають військовим.

Десятий тип належить номерним знакам для мотоциклів і причепів до тракторів (самохідних машин), що належать ЗСУ, ДССТ, ДПСУ, ДСНС, ВВ МВС, ТСОУ, Національній Гвардії.

До одинадцятого типу належать номерні знаки МВС України. Розмір – 520x112мм, тіло – блакитне, символи – білі. Вищі посадові особи МВС мають дещо відмінні номерні знаки. На них міститься ліворуч прапор України, а

праворуч символ МВС. Розмір – 520x112мм, тіло – блакитне, символи – білі. Окрім вищих посадових осіб МВС, вищі посадові особи української влади також мають спеціальні номерні знаки на яких ліворуч зображено герб України, праворуч прапор України. Їх розмір складає – 520x112мм, тіло – біле, символи – чорні. Посадові особи України також мають особливого формату номерні знаки. Ліворуч зображено прапор України, праворуч серія, яка виражає до якої установи належить посадова особа. Серії та діапазон номерів розподіляється наступним чином: 0-000ВР (народні депутати України), 1-000АП (Адміністрація Президента), 2-000ВР (автобаза Верховної Ради України), 3-000КМ (Кабінет Міністрів). Їх розмір складає – 520x112мм, тіло – біле, символи – чорні.

Номерні реєстраційні знаки останнього дванадцятого типу належт Держгірпромнагляду і поділяються на 2 підтипу. До першого належать номерні знаки для великотоннажного та технологічного транспорту (Видаються Держгірпромнаглядом). Розмір – 288x226мм, тіло – біле, символи – чорні. До другого підтипу належать номерні знаки для тимчасового використання на великотоннажному та технологічному транспорті (Видаються Держгірпромнаглядом). Розмір – 140x114мм, тіло – червоне, символи – білі. Де ТР — обов'язкова позначка, АА — код регіону.

Слід згадати що реєстраційні номерні знаки України з деякими кодами регіонів повністю повторюють болгарські та попередні іспанські. Тому неможливо швидко ідентифікувати до якої країни вони належать. Це стосується чотирьох регіонів Іспанії з кодами: ВА, ВІ, СА, ІВ; та п'яти регіонах Болгарії з кодами: ВТ, ВН, СА, СВ, СН, КН.

Така різноманітність автомобільних номерів потребує детального аналізу і чіткого планування системи розпізнавання.

2 ОПИС ТА АНАЛІЗ ПІДХОДІВ ДО РОЗРОБКИ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ

Загальний підхід розпізнавання автомобільних номерів на зображенні складається з 3 етапів: попередній пошук області з зображенням номеру, потім нормування номеру для знаходження чітких меж та регулювання контрасту, і остання частина це безпосередньо розпізнавання тексту на автомобільному номері[4].

Кожен з етапів має різні підходи до рішення проблеми, які мають свої характерні особливості, переваги і недоліки. Для вибору найбільш оптимального підходу необхідно детально розглянути кожен з них.

2.1 Виділення контурів на зображенні

Спочатку є зображення автомобіля, зняте під якимось кутом, часто забруднена, іноді змащене. Деякі з розглянутих в подальшому методів вимагають додаткових перетворень вхідний картинки.

Варто відзначити, що всі наступні методи працюють з чорно-білими зображеннями. Тобто, спершу вихідна картинка з RGB переводиться в відтінки сірого. Приклад розглядаємого зображення показано на рисунку 2.1.



Рисунок 2.1 – Вхідне зображення

Розглянемо способи виділення оператором Собеля. Результатом роботи даного диференціального оператора в точці з постійною яскравістю є нульовий вектор, а якщо ж вона лежить на кордоні областей з різною яскравістю, то вектор, що перетинає кордон спрямований в бік збільшення яскравості[10]. Його роботу можна описати наступними формулами:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \quad (2.1)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (2.2)$$

$$G = \sqrt{G_x^2 + G_y^2}, \quad (2.3)$$

де А – вхідне зображення.

На рисунку 2.2 показано результат виконання оператора.



Рисунок 2.2 – Результат використання оператора Соболя

Як видно з рисунку, оператор виділив кордони. Варто відзначити, що на області автомобіля явно видно контур автомобільного номера і навіть його символи.

2.2 Попередній пошук області з зображенням номеру методом Віоли-Джонса

Попередній метод нікуди не годиться в умовах реальних доріг. Бруд, сніг, пил – ті супутники, без яких автомобільні номери рідко зустрічаються.

У 2001 році Паул Віола і Майкл Джонс запропонували алгоритм для знаходження обличчя з використанням примітивів Хаара[6]. Суть методу полягає в знаходженні ознак на зображенні, характерних для обличчя людини - очі, ніс, рот. Примітиви Хаара на рисунку 2.3 якраз і відображають в собі дані ознаки. Суть даного алгоритму в тому, що по зображенню рухається вікно, до якого додаються дані примітиви по черзі. У разі знаходження ознак, що підходять під умову задачі, дане вікно фіксується.

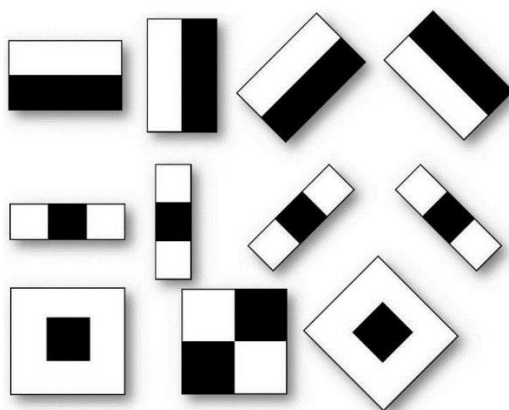


Рисунок 2.3 – Примітиви Хаара

За розрахунками[5] результати алгоритму повинні бути близько 90% вірного визначення.

2.3 Нормування і сегментація

Так як більшість алгоритмів знаходить номер не точно, то проводять нормування та сегментацію зображення номеру.

Після попереднього етапу на вихід подається рамка зображення автомобільного номера, наприклад, як на рисунку 2.4. Невідомий градус нахилу, можлива бруд на номері, відзняти розмиття це лише частина проблем, які необхідно вирішити для подальшої роботи системи.



Рисунок 2.4 – Зображення після попередньої обробки

Для визначення кута нахилу зображення найбільш простий в реалізації є перетворення Хафа. Цей метод знаходить прямі лінії на зображенні. Результат виконання показано на рисунку 2.5.



Рисунок 2.5 – Результат перетворення Хафа

Далі виконуємо поворот на знайдений в попередньому етапі кут. Крім цього, можна відразу за отриманими після перетворення Хафа лініях обрізати зображення. Приклад зображено на рисунку 2.6.



Рисунок 2.6 – Результат повороту та обрізання зображення

Після виділення, необхідно зробити зображення більш контрастним. Для цього можна лінійно розтягнути гістограму зображення. Тепер є правильно орієнтований чорно-білий номер. Наступним етапом буде накладення фільтра LoG (Laplacian of Gaussian), який задається наступною формулою:

$$LoG(x, y) = -\frac{1}{\pi * \sigma^4} * \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) * e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (2.15)$$

де σ – значення в діапазоні 0..1, в даному випадку $\sigma = 0.3$.

Результат виконання оператора LoG показано на рисунку 2.7.



Рисунок 2.7 – Результат виконання оператора LoG

Після цього етапу контури букв чітко видно і читаються на зображенні. Далі потрібно сегментувати номерний знак на символи.

Після нормалізації зображення, сегментацію найлегше провести шляхом побудови графіка проекції на ОХ і виділення на графіку максимумів[2].

Приклад роботи можна побачити на рисунку 2.8.

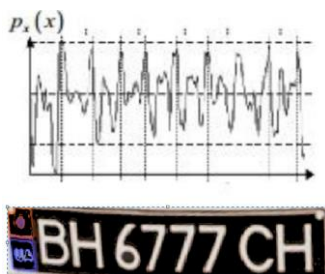


Рисунок 2.8 – Результат побудови графіка проекції

Нажаль, у даного способу є слабкі місця. Наприклад, сильно зашумлений номер розділити на символи буде вкрай проблематично, адже наведений метод сегментації залежить від кольорів[11]. Отже, при наявності чорного бруду на білому номері крім контуру символів номера виділиться і бруд. Крім того, існують ще автомобілі з увігнутими номерами, у яких обробка буде дуже ускладнена.

2.4 Розпізнавання символів з використанням нейромереж

Результатом минулого етапу є певна кількість зображень цифр і букв. Дані зображення необхідно розпізнати. Завдяки стандартному шрифту використання тих чи інших алгоритмів буде спрощено, адже основні проблеми розпізнавання у рукописних шрифтів. Але з огляду на те, що це цифри і літери номерного знаку автомобіля, часто інформація може бути частково зашумлена брудом, пилом, снігом чи просто витерта. Штучна нейронна мережа - це математична модель, а також її програмна реалізація, яка будується за принципом функціонування нервових систем живих об'єктів. З точки зору машинного навчання, нейромережа являє собою окремий метод розпізнавання образів або класифікації[3]. Найпростіша модель нейронної мережі зображена на рисунку 2.9.

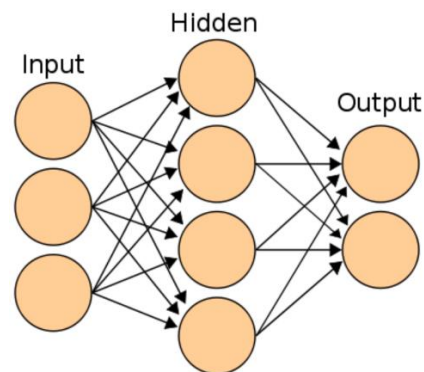


Рисунок 2.9 – Модель нейронної мережі

Нейромережі навчаються – це їхня відмінність від більшості інших алгоритмів. Технічно, навчання - це процес знаходження коефіцієнтів зв'язків між нейронами. Завдяки узагальненню, нейромережа може зробити висновок про об'єкт, який не був присутній в навчальній вибірці[9]. Крім цього, проблема зашумленості також не є суттєвою. Нейронні мережі ідеально підходять для задачі розпізнавання автомобільних номерів.

Бінаризація зображення допомагає сегментувати номер, але непередбачувано спотворює самі символи. Для побудови Нейромережі необхідно враховувати цей факт.

Як висновок, даний метод краще підходить для рішення поставленої задачі, для навчання мережі потрібна вибірка даних менше ніж для попереднього методу. До недоліку наданого методу можна віднести неможливість відстежити аномальну поведінку в роботі мережі[8].

3 ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

Виходячи з мети роботи, програмний засіб, що розробляється, повинен розпізнавати автомобільні номери на зображеннях, що передаються на обробку.

Програмний засіб повинен мати логічно декомповану структуру, налаштування та запуск програмного засобу з усіма використаними сторонніми допоміжними програмними модулями повинен виконуватися кросплатформово, без доповнення та зміни конфігурації проекту. Процедура запуску додатків має бути простою. Також інтерфейс користувача доступу до системи повинен бути інтуїтивно зрозумілим і простим у використанні.

3.1 Проектування системи

Для проектування на основі предметної області була використана методологія IDEF0 для створення функціональної декомпозиції процесів.

На рисунку 3.1 представлена контекстна.

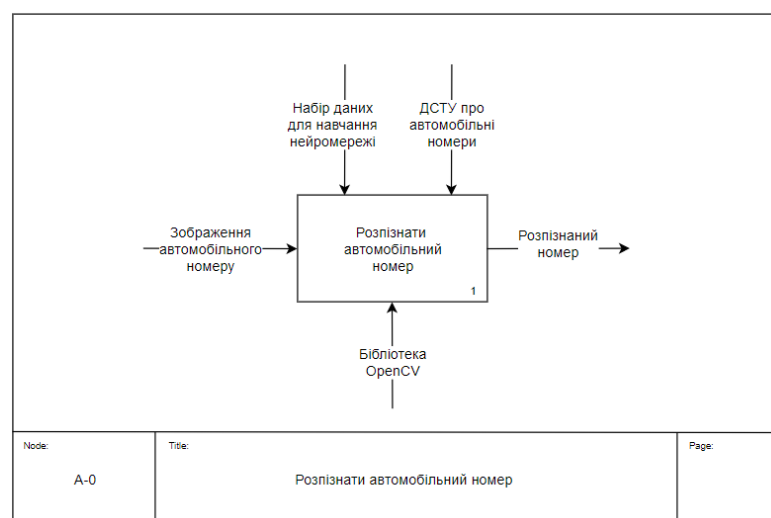


Рисунок 3.1 – Контекстна діаграма

На рисункові 3.2 представлена діаграма декомпозиції першого рівня.

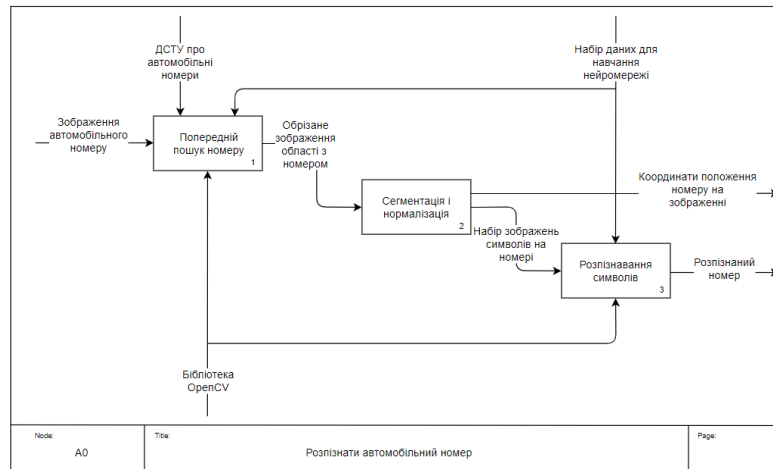


Рисунок 3.2 – Декомпозиція основного бізнес-процесу

Для того, щоб описати, як буде використовуватись розроблюваний програмний засіб на концептуальному рівні слід скласти діаграму прецедентів.

Діаграма варіантів використання системи, представлена на рисунку 3.3, була складена на підставі вимог до програмного засобу.

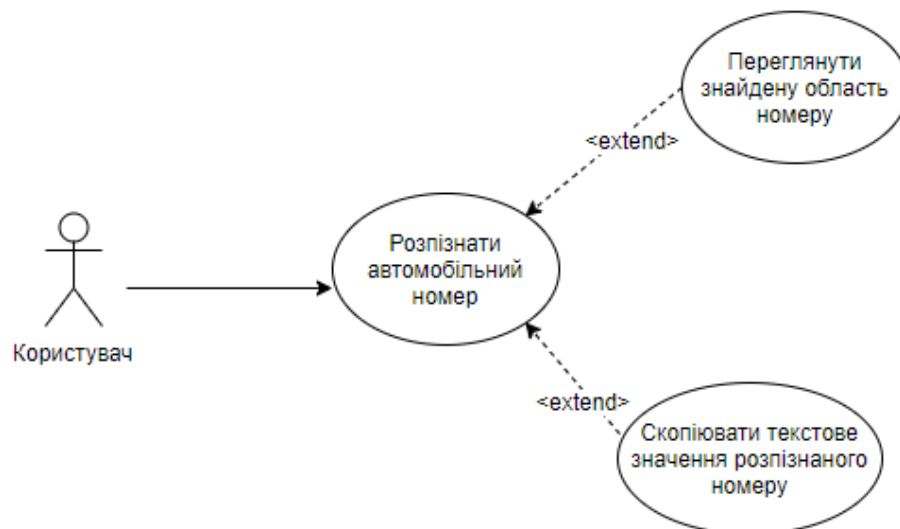


Рисунок 3.3 – Діаграма варіантів використання системи

3.2 Вибір і обґрунтування технічного і програмного забезпечення

3.2.1 Вибір і обґрунтування технічного забезпечення

Для забезпечення необхідного рівня продуктивності сервера слід використовувати комп'ютер з тактовою частотою процесора не менше 2.4 ГГц і обсягом оперативної пам'яті не менше 8 Гб, відео карта з обсягом оперативної пам'яті не менше 4 Гб. При даних значеннях основних характеристик процесора та відео пам'яті навчання нейронної мережі і обробка зображення буде відповідати вимогам до системи(див. розділ 1.3). В роботі середовища серверної частини не повинно бути збоїв.

Для забезпечення необхідного рівня продуктивності компонента, який відповідає за надання веб-інтерфейсу для користувача слід використовувати комп'ютер з тактовою частотою процесора не менше 2 ГГц і обсягом оперативної пам'яті не менше 4 Гб. Таких характеристик достатньо для роботи даного компоненту з кількістю одночасних запитів близької до 500, що задовольняю вимоги.

Ще однією необхідною умовою запуску тестів для веб-орієнтованого додатка являється висока пропускна здатність каналу Internet.

У тому випадку, якщо система буде використовуватися в промислових масштабах, слід задуматися про використання розподіленої архітектури, де буде присутні декілька серверних частин, над ними стоїть балансувальник навантаження, який буде приймати рішення якому з серверів віддавати на обробку запит.

3.2.2 Вибір і обґрунтування мовних і програмних засобів

Враховуючи основні характеристики та поширеність використання мов програмування у сфері розробки серверів, вдалим вибором для розробки серверної частини є мова програмування Java. Для даної мови програмування

написана величезна кількість бібліотек, що спрощує процес створення програмних засобів. Не менш важливим є те що Java кросплатформова мова програмування, тому додаток можна буде запускати на різних платформах. Серед основних характеристик Java можна виділити наступні: простий об'єктно-орієнтований синтаксис, сурова типізація, незалежність від архітектури, висока продуктивність виконання [7].

Однією з найбільш зручних середовищ розробки на Java вважають IntelliJ IDEA. IntelliJ IDEA має механізми аналізу коду, який спрощує простежування зв'язаності коду, перехід до шуканої реалізації, знаходження помилок та здійснення рефакторингу.

Серед інструментів складання(будування) проекту був обраний Apache Maven – це засіб автоматизації роботи з програмними проектами. Цей засіб має простий вигляд build-налаштувань, яке надається в форматі XML. XML-файл описує проект, його зв'язки з зовнішніми модулями і компонентами, порядок будування(build), папки та необхідні плагіни. Виконання певних, чітко визначених задач – таких, як компіляція коду та пакетування відбувається шляхом досягнення заздалегідь визначених цілей(targets). Двигун ядра може динамічно завантажувати плагіни з репозиторію, що забезпечує доступ до багатьох версій різних Java-проектів з відкритим кодом, від Apache та інших організацій та окремих розробників.

Для розробки додатку, який надає веб-інтерфейс користувача було вибрано мову програмування JavaScript та фреймворк VueJs, що працює на платформі NodeJS. Даний фреймворк спрямований на створення динамічних веб-сторінок. Слід зазначити, що платформа NodeJS є кросплатформовою, що є значною перевагою, оскільки дозволяє запускати додаток на різних платформах. Не слід забувати, що на відміну від мови Java, платформа NodeJS не має зворотної сумісності, тому не слід змінювати чи оновлювати версію платформи.

Серед інструментів складання(будування) проекту був обраний NPM(Node Package Manager) це засіб автоматизації збірки проекту, а також дозволяє легко

працювати з залежностями проекту. Конфігурації описуються з допомогою JSON-файлів.

Кожен з двох додатків залежить від роботи інструментів будівництва проектів, а також від платформ на яких працюють. Таким чином кожен компонент має певне цільне середовище роботи. Для створення, перенесення, та запуску такого середовища максимально простим чином було прийняте рішення використати засіб контейнеризації Docker, та розширення docker-compose для створення та підтримки компонентної архітектури компонентів однієї системи.

3.3 Розробка компонентів програмного засобу розпізнавання автомобільних номерів

Розробка програмного засобу розпізнавання автомобільних номерів складається з декількох етапів: створення проектів та налаштування середовища для серверної частини та додатку, який надає веб-інтерфейс користувача, контейнеризація даних додатків, розробка серверної частини, розробка веб-інтерфейсу користувача. Кожен з цих етапів буде розглянуто далі.

3.3.1 Опис створення проекту та контейнеризації додатків

Для розробки програмного засобу був проведений аналіз основних вимог до його функціональності, було розроблено його проект, був обраний оптимальний підхід до розробки серверної частини – комбінація підходів під управлінням даними та поведінкою, а також був здійснений вибір програмних та апаратних засобів, які максимально спрощують процес розробки.

Проект із кодом програмного засобу – стандартний за структурою Maven проект за архетипом для веб додатків із окремими директоріями для основного коду, тестів, допоміжних інструментів та конфігураційних файлів. Також було визначено структуру пакетів проекту де роздільно знаходяться класи для моделей даних, класів виключних ситуацій для окремих випадків виникнення

помилки в додатку, шар сервісів, який відповідний за логіку виконання програмного засобу, модуль з допоміжними класами для роботи основної частини, модуль відповідний за роботу прийом запитів по HTTP протоколу, передачу обробки до шару сервісів та повернення результату обробки. Така структура представлена у ієрархічному вигляді на панелі файлів проекту у середовищі розробки на рисунку 3.4.

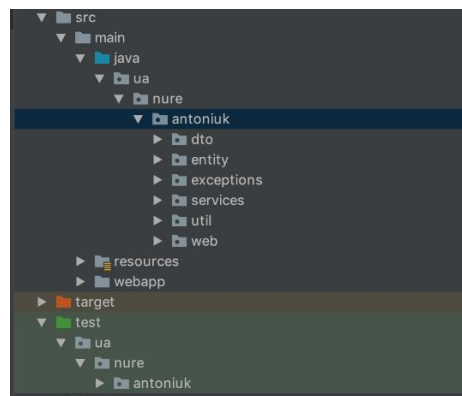


Рисунок 3.4 – Структура проекту серверної частини

Файли пакету `src/main` становлять програмний засіб для розпізнавання автомобільних номерів, а все, що знаходиться у `test`, становить модульні тести для покриття функціоналу.

У процесі розробки було вирішено дотримуватись наступної структури проекту:

- `src/main/java` – класи, що відповідають за основну роботу програмного засобу;
- `src/main/resources` – інструменти для конфігурації і налаштувань сервісів;
- `test` – класи із тестовими методами, з модульними тестами, які покривають функціонал.

Для створення проекту додатку веб-інтерфейсу користувача було використано `vue cli` – допоміжна програма яка створює основну структуру

проекту. Також було створено два Vue компонента: перший для рисування основного вікна, другий відповідає за відправку зображення та рендерінг результату. Таким чином структура проекту веб-інтерфейсу користувача зображено на рисунку 3.5.

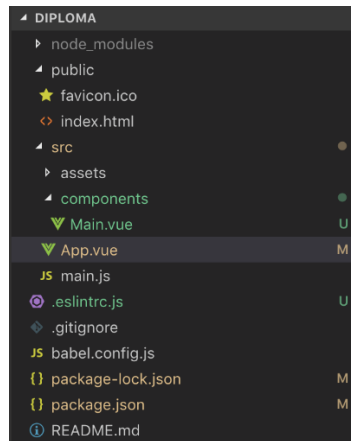


Рисунок 3.5 – Структура проекту додатку веб-інтерфейсу користувача

Для компіляції використовується компілятор мови JavaScript `babel-compiler`, який підтримує компіляцію конкретних модулів додатку, без перезапуску програмного засобу, що значно спрощує і прискорює процес розробки і налаштування додатка, адже зміни в шаблонах робляться постійно. Також використовується аналізатор коду `eslint`, який допомагає уникнути помилок та слідкує за коректним написанням коду в рамках проектного стилю коду, що покращує читаємість коду.

Для контейнеризації було використано систему `Docker`, яка створює оточення в окремій віртуальній машині, та надає зручний засіб роботи з нею. Тож поряд з кожним проектом було додано файли з назвою «`Dockerfile`», що сигналізує для програми `Docker` що необхідно робити для створення контейнера.

Для контейнеризації серверної частини було обрано за основу операційну систему `CentOS 7`. Потім прописано встановлення `OpenJDK8`, для роботи Java-додатків. Також було додано в змінних середовища шлях для віртуальної машини. Після цього було встановлено систему збірки `Maven`. Для роботи

алгоритмів попередньої обробки зображення та розпізнавання символів використовується бібліотека OpenCV 3.4, для її інтеграції з нашим додатком необхідно було скомпілювати вихідні файли бібліотеки для обраної операційної системи, а також вказати Java Virtual Machine використовувати їх як нативні бібліотеки. Для роботи додатку як веб-сервера було встановлено до контейнеру контейнер сервлетів Apache Tomcat 8.1. Запуск контейнера ініціює запуск нового коду в контейнері сервлетів.

Для контейнеризації серверної частини було обрано за основу операційну систему CentOS 7. Було добавлено встановлення середовища NodeJS 9.0, для роботи додатків з використанням VueJS. Прописане встановлення систему збірки і управління пакетами для платформи NodeJS – NPM (Node Package Manager). Для комфортної роботи з фреймворком було додано модуль vue cli 3.4. Запуск контейнера ініціює збірку проекту і його запуск.

Було прийняте рішення використання розширення для системи Docker – docker-compose, що дозволяє управляти декількома контейнерами в одному середовищі. Для роботи цього розширення потрібно створити файл docker-compose.yml, який управляє контейнерами. Було створено наступний конфігураційний файл:

```
version: "2.2"
services:
  server-side:
    image: "diploma-docker/server-side"
    build:
      context: ./server-side
    volumes:
      - "/usr/maksym_antoniuk/Projects/diploma:/app/tomcat"
    expose:
      - "8000"
  ui-side:
    image: "diploma-docker/ui-side"
    build:
```

```

context: ./ui-side
volumes:
  - "/usr/maksym_antoniuk/Projects/NodeJS/diploma:/etc"
expose:
  - "8080

```

3.3.2 Розробка серверної частини

Перш за все для попередньої обробки зображення та знаходження області з номером було навчено каскад Хаара знаходити автомобільні номери, для цього було зібрано колекцію зображень автомобілів з номерами, вказані параметри пошуку, ваги нейронів мережі. Зображення були зібрані власноруч, фотографуючи автомобілі на вулиці. Алгоритм навчання каскаду Хаара для визначення області номеру на зображенні показано на рисунку 3.6.

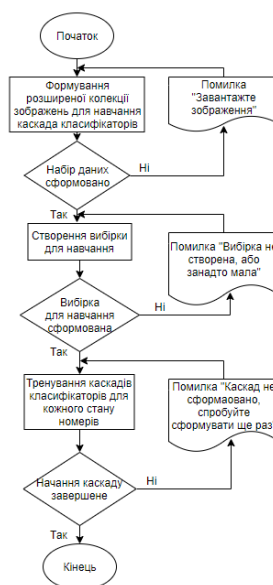


Рисунок 3.6 – Блок-схема алгоритму навчання каскаду Хаара

Після навчання каскаду Хаара, ще слід навчити нейронну мережу розпізнавати символи. Для цього було нарізано на зображення багато номерних знаків і їх символів, кожному зображенню поставлений оригінальний символ, і

далі за допомогою програмної реалізації алгоритму Tesseract OCR, з бібліотеки OpenCV була навчена нейронна мережа для розпізнавання символів. Алгоритм навчання схожий з навчанням каскаду Хаара.

Після підготовчих дій було розроблено чотири основні сервіси в додатку, які по черзі викликаються і виконують операції над зображенням після чого повертається результат користувачеві. Діаграма послідовності для розпізнавання автомобільного номери при умові успішного розпізнавання зображено на рисунку 3.7.

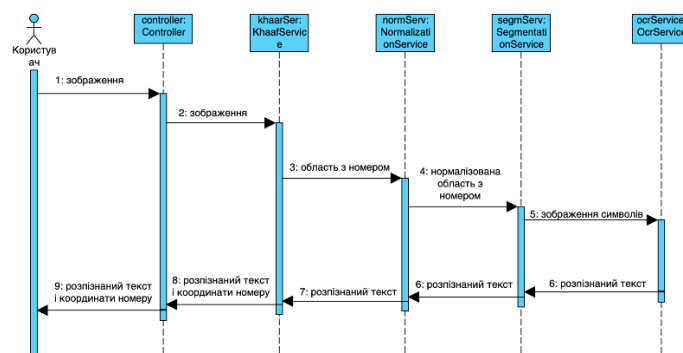


Рисунок 3.7 – Діаграма послідовності для оптимального сценарію розпізнавання номеру

Користувачеві буде повернена відповідь у вигляді json масиву об'єктів з полями number(номер автомобіля), coordinates(об'єкт з полями x та y).

У випадку якщо не вдалося розпізнати ніякий з символів, в сервісі OrcService згенерується об'єкт виключної ситуації OrcException. Така ситуація може виникнути коли було знайдено не автомобільний номер а якась прямокутна область з невідомими символами чи зображеннями, або автомобільний номер занадто забруднений чи пошкоджений через це символи розпізнати не вдалося. Помилка буде оброблена в класі Controller, та користувачеві буде повернуто результат у вигляді пустого json масиву.

Існує ситуація при якій алгоритм Віоли-Джонса не знайшов автомобільний номер, така ситуація може виникнути коли буде передане зображення яке не

містить зображення номеру або алгоритм не зміг його знайти на зображенні, в такому випадку буде створена виключна ситуація і сервіс KhaaService згенерує KhaaException, яке буде оброблене класом Controller, і користувачеві буде повернуто пустий json масив.

3.3.3 Розробка додатку веб-інтерфейсу користувача

Для надання веб-інтерфейсу користувача було створено Vue компонент App, який в своєму шаблоні визначає основну структуру веб-сторінки, а також підключає основні залежності на модулі, необхідні для роботи додатку. Також було створено компонент Main, в якому визначається структура та поведінка HTML елементів для завантаження зображення, відправки зображення та рендерінгу результатів обробки.

Після вибору зображення та натискання кнопки «Upload» відбувається завантаження зображення до серверної частини, під час очікування відповіді від сервера з'являється компонент Spinner, у вигляді кола з синім сегментом, який крутиться навколо центру доки відповідь від сервера не буде отримано.

Після успішного отримання результату створюється елементи canvas та img, завантажене зображення записується до елементу img, потім даний елемент рисується на елементі canvas, і по отриманим координатам номеру рисуються червоні лінії поверх зображення, також відображається знайдений номер і поряд з ним кнопка яка копіює текстове значення номеру до буферу обміну.

У випадку отримання пустого json масиву від серверної частини створюється елементи canvas та img, завантажене зображення записується до елементу img, потім даний елемент рисується на елементі canvas, як і у випадку успішного розпізнавання, але на зображенні не буде червоних ліній області номеру, на місці значення номеру буде відображено напис «UNFORTUNATELY, NOT RECOGNIZED», також кнопка для копіювання номера не буде відображено.

4 АНАЛІЗ РЕЗУЛЬТАТІВ, ОТРИМАНИХ ЗА ДОПОМОГОЮ ПРОГРАМНОГО ЗАСОБУ

У результаті виконання роботи було розроблено програмний засіб для розпізнавання автомобільних номерів.

Тобто розроблений додаток дозволяє автоматизувати процес розпізнавання номерних знаків з зображень та стоп-кадрів камер спостереження чи спеціальних пристроїв розпізнавання номерів.

Для аналізу результатів роботи програмного засобу була зібрана невелика кількість зображень автомобілів з номерами, які можна розпізнати людським оком, та мануально перевірена робота програмного засобу.

Спочатку було перевірено ряд зображень зі звичайними номерними знаками, чітко вираженими, не забрудненими, з чіткими чорними символами на білому тлі номеру. Програмний засіб відмінно розпізнав номерні знаки на них.

Було знайдене високоякісне зображення автомобіля з розділяючою здатністю 1920x1080 з забрудненим і зашумленим номером. Програмний засіб успішно розпізнав його. Результат розпізнавання зображено на рисунку 4.1.



Рисунок 4.1 – Результат розпізнавання високоякісного зображення з забрудненим номером

Потім якість зображення було зменшено вдвічі і запущено процес розпізнавання знову. На цей раз система не змогла розпізнати номер. В даному

випадку це сталося за двох причин. По-перше, автомобіль такого ж кольору як і тіло автомобільного номеру, що ускладнює процес знаходження номеру. По-друге, через забрудненість номеру і низьку якість зображення під час виділення контурів і фільтрування стало неможливим чітко визначити положення номерного знаку. Дана ситуація підкреслює необхідність використання якісних пристроїв оцифрування медійної інформації, і тому ця тема зараз є актуальною через появу якісних камер доступних будь-кому.

Потім було знайдене зображення автомобільного номеру з тимчасовим номером виданим торговою компанією, які мають червоне тіло номеру і білі літери (див. розділ 1.1). В результаті такий номер програмний засіб розпізнати не зміг. Така ситуація виникла з двох причин. По-перше, при навчанні каскаду Хаара було використано лише декілька зображень з номерами не білого кольору, тому алгоритм в деяких випадках не знаходив рамки номеру, але все ж система розпізнає такі номерні знаки, в чіткому ракурсі, з високою якістю зображення. По-друге, при навчанні нейромережі для розпізнавання символів не було білих символів на червоному фоні, тому OrcService не зміг розпізнати символи.

Окрім цього було проаналізовано швидкість обробки зображення серверною частиною. В залежності від якості зображення швидкість обробки запиту до серверної частини знаходиться в діапазоні 400мс–1900мс. Дані заміри проводилися на ноутбучі MacBook Pro з процесором Intel Core i7, та об'ємом оперативної пам'яті 16Гб.

ВИСНОВКИ

У ході виконання роботи був проведений аналіз предметної області і був розроблений програмний засіб розпізнавання автомобільних номерів.

В результаті проведеного аналізу особливостей різних алгоритмів, які дозволяють розпізнати автомобільні номери, була обрана зв'язка алгоритмів з оператору Соболя для контуризації, метод Віоли-Джонса для знаходження області номеру, перетворення Хафа для знаходження граней номеру і використання нейромереж для розпізнавання символів.

Розроблений програмний засіб надає користувачу можливість розпізнати автомобільний номер на основі зображення отриманого з фото чи стоп-кадру відеозапису. Система складається з двох компонентних складових: серверна частина яка розпізнає автомобільний номер, веб-додаток який надає інтерфейс користувача для зручної роботи з системою. Такий підхід дозволяє з легкістю створити декілька клієнтів, які будуть працювати з однією серверною частиною, наприклад мобільний додаток, чи інша інформаційна система.

Функціональність програмного засобу була перевірена на великій кількості зображень, і показала добрі результати, відмінно розпізнає більшість номерних знаків. Проте було виявлено що система не може розпізнати автомобільні номери які мають червоне тіло номеру та білі літери, також система погано розпізнає забруднені номери, якщо отримане зображення має низьку якість, що ускладнює процес розпізнавання. Алгоритми, які використовуються в дослідженні потребують великої матеріальної бази для навчання алгоритмів розпізнавання. Окрім зазначених, до них належать номери без обрамлень, які зливаються з фоном автомобіля, номери покриті сіткою, спеціально для ускладнення розпізнавання, що є протизаконним, брудні номери, номери, що втоплені в кузов автомобіля. Передбачення та обробка таких випадків вимагає удосконалення класичних алгоритмів обробки зображень.

ПЕРЕЛІК ПОСИЛАНЬ

1. ДСТУ 4278:2012 [Електронний ресурс] / Управління безпеки дорожнього руху – Режим доступу: [www / URL: http://www.sai.gov.ua/uploads/filemanager/file/zmina-1-dstu-4278.pdf](http://www.sai.gov.ua/uploads/filemanager/file/zmina-1-dstu-4278.pdf)
2. Стругайло В.В. «Огляд методів фільтрації і сегментації цифрових зображень» [Текст] / Стругайло В.В.
3. Арлазаров В. Л., Троянker В.В., Котович Н.В. Адаптивне розпізнавання символів. [Електронний ресурс]. - <http://www.ocrai.narod.ru/adaptive.html>.
4. Tamkang Journal of Science and Engineering, Vol. 13, No. 4, pp. 433-442 (2010) «A Real-Time Mobile Vehicle License Plate Detection and Recognition», Kuo-Ming Hung and Ching-Tang Hsieh
5. Messom, C.H. and Barczak, A.L.C., «Fast and Efficient Rotated Haar-like Features Using Rotated Integral Images», Australian Conference on Robotics and Automation (ACRA2006), pp. 1-6, 2006.
6. Hoiem, D., Chodpathumwan, Y., and Dai, Q. 2012. Diagnosing Error in Object Detectors. Computer Vision -ECCV 2012 Springer Berlin Heidelberg, 340-353.
7. «ALGORITHMIC AND MATHEMATICAL PRINCIPLES OF AUTOMATIC NUMBER PLATE RECOGNITION SYSTEMS» ONDREJ MARTINSKY, 2007, Brno University.
8. Shakhnarovich, Darrell, and Indyk, eds. (2005). Nearest-Neighbor Methods in Learning and Vision. MIT Press. ISBN 0-262-19547-X.
9. Блек, Рекс. Critical Testing Processes. Plan, Prepare, Perform, Perfect. [Текст] / Рекс Блек. – М. : Лорі, 2011 – 544 с.
10. Фомін Я. А. Розпізнавання образів: теорія і застосування. - 2-е вид. - М. : фазисами, 2012. - 429 с. - ISBN 978-5-7036-0130-4.
11. Горелик А. Л., Скрипкін В. А. Методи розпізнавання. - 4-е изд. - М. : Вища школа, 1984, 2004. - 262 с.