

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ  
ВІДДІЛЕННЯ «ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА  
КОМП'ЮТЕРНІ НАУКИ»  
ЦИКЛОВА КОМІСІЯ СПЕЦІАЛЬНОСТІ «КОМП'ЮТЕРНІ НАУКИ»

## ПОЯСНОВАЛЬНА ЗАПИСКА

до дипломної роботи

освітньо-професійний ступінь «фаховий молодший бакалавр»

на тему:

Розробка програми «Ідентифікація номеру автомобіля»

Виконав студент (ка) 4 курсу, групи П-41  
спеціальності 122 «Комп'ютерні науки»

Пономарчук Євген Володимирович

Керівник Устименко Людмила Миколаївна

Рецензент \_\_\_\_\_

Житомир – 2024 року

## Зміст

Вступ _____	6
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ _____	11
1.1. Постановка основної задачі розробки _____	11
1.2. Огляд та порівняння аналогічних систем. _____	12
1.3. Вибір та обґрунтування технологій розробки _____	15
Висновки до розділу 1 _____	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА _____	21
2.1. Вибір алгоритмів та їх ефективність _____	21
2.2. Програмна реалізація _____	23
2.3. Тестування програми _____	31
Висновки до розділу 2 _____	35
РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ _____	36
3.1. Мінімальні вимоги до системи для роботи програми _____	36
3.2. Інтерфейс користувача _____	37
Висновки до розділу 3 _____	43
ВИСНОВКИ _____	44
Перелік літературних джерел _____	45
Додаток А. _____	48

					<i>ДР.122.041.027.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Пономарчук Є.В.</i>			<i>Розробка програми «Ідентифікація номеру автомобіля»</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевірив</i>		<i>Устименко Л.М.</i>				3	53	
<i>Рецензент</i>						<b>ЖАТФК</b>		
<i>Н. Контр.</i>								
<i>Затвердив.</i>		<i>Лавріщев О.О.</i>						

## РЕФЕРАТ

Записка: 53 стор., 30 рис., 1 додаток, 20 джерел.

Ключові слова: ІДЕНТИФІКАЦІЯ НОМЕРНИХ ЗНАКІВ, ОБРОБКА ЗОБРАЖЕНЬ, OCR, OPENCV, TESSERACT OCR, PYTHON, КОМП'ЮТЕРНИЙ ЗІР.

Дипломна робота присвячена розробці програми для автоматичної ідентифікації номерних знаків автомобілів. Актуальність теми зумовлена зростанням кількості транспортних засобів та потребою в автоматизації процесів контролю і безпеки на дорогах. Основною метою роботи є створення програмного забезпечення, що забезпечує високу точність та швидкість розпізнавання номерних знаків. Для досягнення цієї мети використано мову програмування Python та бібліотеки OpenCV і Tesseract OCR. В процесі роботи було проаналізовано існуючі системи ідентифікації, обрано оптимальні технологічні рішення та реалізовано програму з інтуїтивно зрозумілим інтерфейсом. Тестування показало високу ефективність та надійність розробленої системи. Програма здатна обробляти зображення, виділяти номерні знаки та розпізнавати текст навіть у складних умовах. Використання сучасних алгоритмів забезпечує точність та швидкість роботи системи. Розроблене програмне забезпечення може бути застосоване в різних сферах, таких як контроль дорожнього руху та автоматизація паркування. Результати роботи підтверджують доцільність використання вибраних технологій для вирішення завдань автоматичної ідентифікації номерних знаків.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

OCR (Optical Character Recognition) - оптичне розпізнавання символів

ALPR (Automatic License Plate Recognition) - автоматичне розпізнавання номерних знаків

API (Application Programming Interface) - інтерфейс програмування додатків

GUI (Graphical User Interface) - графічний інтерфейс користувача

CV (Computer Vision) - комп'ютерний зір

ML (Machine Learning) - машинне навчання

AI (Artificial Intelligence) - штучний інтелект

ROI (Region of Interest) - область інтересу

HSV (Hue, Saturation, Value) - колірна модель, що описує кольори через відтінок, насиченість та значення

FPS (Frames Per Second) - кількість кадрів в секунду

					ДР.122.041.027.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

## ВСТУП

У сучасному світі інформаційні технології відіграють важливу роль у багатьох сферах життя, включаючи транспорт та безпеку. Однією з таких технологій є системи автоматичного розпізнавання номерних знаків автомобілів (ANPR - Automatic Number Plate Recognition). Ці системи мають широкий спектр застосувань, починаючи від контролю дорожнього руху та паркування до забезпечення правопорядку та охорони. Зокрема, автоматична ідентифікація номерних знаків дозволяє значно підвищити ефективність управління транспортними потоками та сприяє підвищенню рівня безпеки на дорогах.

Розробка програми для ідентифікації номеру автомобіля є актуальною задачею, яка сприяє вирішенню ряду важливих проблем, таких як боротьба з незаконними транспортними засобами, автоматизація процесів паркування, моніторинг руху та збір статистичних даних. Використання таких програм дозволяє зменшити людський фактор та підвищити точність і швидкість обробки даних.

У рамках даної дипломної роботи буде розглянуто процес розробки програми для ідентифікації номеру автомобіля, що включає кілька ключових етапів: попередню обробку зображення, виявлення номерного знака та розпізнавання тексту. Для реалізації цих задач будуть використані сучасні бібліотеки обробки зображень та розпізнавання тексту, такі як OpenCV та Tesseract.

Мета роботи полягає в створенні ефективної та точної системи для автоматичного розпізнавання номерних знаків автомобілів на основі зображень. Завданням є детальне вивчення методів обробки зображень, алгоритмів виявлення об'єктів та технологій оптичного розпізнавання символів, а також їх практична реалізація у вигляді програмного продукту.

Результатом даної роботи стане програма, здатна автоматично розпізнавати номерні знаки автомобілів на зображеннях, що може бути використано у різних сферах, від комерційних до державних. Ця система

									Арк.
									6
Змн.	Арк.	№ докум.	Підпис	Дата	ДР.122.041.027.ПЗ				

допоможе підвищити ефективність та надійність існуючих процесів, що вимагають автоматизації ідентифікації транспортних засобів.

Розробка програми для ідентифікації номеру автомобіля є актуальною задачею, яка сприяє вирішенню ряду важливих проблем, таких як боротьба з незаконними транспортними засобами, автоматизація процесів паркування, моніторинг руху та збір статистичних даних. Використання таких програм дозволяє зменшити людський фактор та підвищити точність і швидкість обробки даних.

У рамках даної дипломної роботи буде розглянуто процес розробки програми для ідентифікації номеру автомобіля, що включає кілька ключових етапів: попередню обробку зображення, виявлення номерного знака та розпізнавання тексту. Для реалізації цих задач будуть використані сучасні бібліотеки обробки зображень та розпізнавання тексту, такі як OpenCV та Tesseract.

Мета роботи полягає в створенні ефективної та точної системи для автоматичного розпізнавання номерних знаків автомобілів на основі зображень. Завданням є детальне вивчення методів обробки зображень, алгоритмів виявлення об'єктів та технологій оптичного розпізнавання символів, а також їх практична реалізація у вигляді програмного продукту.

Результатом даної роботи стане програма, здатна автоматично розпізнавати номерні знаки автомобілів на зображеннях, що може бути використано у різних сферах, від комерційних до державних. Ця система допоможе підвищити ефективність та надійність існуючих процесів, що вимагають автоматизації ідентифікації транспортних засобів.

У сучасному суспільстві зростання кількості транспортних засобів на дорогах ставить нові виклики перед системами контролю та управління дорожнім рухом. Зокрема, ефективне управління транспортними потоками, забезпечення безпеки на дорогах, а також боротьба зі злочинністю у сфері транспорту вимагають впровадження новітніх технологічних рішень. Одним

					<i>ДР.122.041.027.ПЗ</i>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

із таких рішень є автоматична ідентифікація номерних знаків автомобілів (ANPR), яка набуває все більшого значення.

З кожним роком кількість автомобілів у містах і на дорогах загального користування неухильно зростає. Це створює значне навантаження на дорожню інфраструктуру і потребує ефективних систем управління транспортними потоками. Автоматичні системи розпізнавання номерних знаків дозволяють оперативно відстежувати та контролювати рух транспортних засобів, сприяючи зниженню заторів та підвищенню пропускної здатності доріг.

Питання безпеки на дорогах є надзвичайно важливим, особливо у великих містах і на магістралях. Системи ANPR дозволяють автоматично виявляти транспортні засоби, що перебувають у розшуку, а також порушників правил дорожнього руху. Це значно полегшує роботу правоохоронних органів та підвищує загальний рівень безпеки. Крім того, такі системи є ефективним засобом боротьби зі злочинністю, пов'язаною з викраденням автомобілів та іншими правопорушеннями.

В умовах зростаючого дефіциту паркувальних місць у містах автоматичні системи розпізнавання номерних знаків відіграють важливу роль в організації ефективного управління паркувальними зонами. Вони дозволяють автоматично ідентифікувати автомобілі, що в'їжджають та виїжджають з паркувальних майданчиків, а також забезпечують автоматичний облік та оплату паркування. Це сприяє підвищенню зручності для водіїв та зменшенню кількості порушень правил паркування.

Системи автоматичного розпізнавання номерних знаків також можуть використовуватися для збору та аналізу статистичних даних про транспортні потоки. Це дозволяє міським службам отримувати актуальну інформацію про завантаженість доріг, а також планувати та оптимізувати дорожню інфраструктуру. На основі зібраних даних можуть розроблятися стратегічні рішення для покращення транспортної системи міста.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

З розвитком технологій, зокрема в галузі комп'ютерного зору та штучного інтелекту, можливості систем ANPR постійно вдосконалюються. Сучасні алгоритми дозволяють досягати високої точності розпізнавання номерних знаків навіть в умовах поганої видимості, складних погодних умовах чи при високій швидкості руху транспортних засобів. Це відкриває нові перспективи для їх широкого впровадження в різних сферах.

Таким чином, розробка програми для ідентифікації номерних знаків автомобілів є актуальною та важливою задачею, що сприяє вирішенню низки ключових проблем сучасного транспортного середовища. Вона має значний потенціал для підвищення ефективності, безпеки та зручності управління дорожнім рухом, а також для покращення якості життя в містах.

### **Основні цілі розробки**

Розробка ефективної системи ідентифікації номерних знаків автомобілів: створити програму, яка автоматично виявляє та розпізнає номерні знаки на зображеннях з високою точністю.

Інтеграція алгоритмів обробки зображень та розпізнавання тексту: використати сучасні бібліотеки обробки зображень та технології оптичного розпізнавання символів для досягнення найкращих результатів.

Оптимізація роботи програми для забезпечення швидкості та надійності: забезпечити швидку обробку даних та стабільну роботу програми в різних умовах.

Розробка зручного користувацького інтерфейсу: створити інтерфейс, який дозволяє легко завантажувати зображення, переглядати результати розпізнавання та налаштовувати параметри обробки.

### **Завдання кваліфікаційної роботи**

Аналіз існуючих методів ідентифікації номерних знаків: провести дослідження сучасних алгоритмів обробки зображень та розпізнавання тексту.

Вибрати найефективніші підходи для реалізації в програмі.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

Розробка та впровадження алгоритмів обробки зображень: реалізувати методи попередньої обробки зображень, такі як перетворення у відтінки сірого, згладжування та виділення країв.

Створити алгоритми для виявлення та сегментації номерних знаків.

Інтеграція системи оптичного розпізнавання символів (OCR): впровадити Tesseract OCR для розпізнавання символів на номерних знаках.

Налаштувати параметри OCR для досягнення високої точності.

Тестування та оптимізація програми: провести тестування програми на різних наборах зображень для оцінки точності та швидкості роботи.

Оптимізувати програму для підвищення продуктивності.

Розробка користувацького інтерфейсу: створити зручний інтерфейс для взаємодії користувача з програмою.

Забезпечити можливість налаштування параметрів обробки та збереження результатів.

Об'єкт дослідження: системи автоматичного розпізнавання номерних знаків автомобілів (ANPR).

Предмет дослідження: алгоритми обробки зображень та технології оптичного розпізнавання символів для ідентифікації номерних знаків автомобілів.

Для розробки програми використовувалася мова програмування Python. Python обрано через його зручність для швидкої розробки, багатий набір бібліотек для обробки зображень (зокрема, OpenCV) та потужні засоби для оптичного розпізнавання символів (Tesseract OCR).

					ДР.122.041.027.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10



Створити детальну документацію для користувачів та розробників, що містить інструкції щодо встановлення, налаштування та використання програми.

Забезпечити можливість подальшого розвитку та вдосконалення програми, включаючи оновлення алгоритмів та підтримку нових форматів номерних знаків.

Автоматично ідентифікувати номерні знаки автомобілів з зображень з високою точністю.

Працювати стабільно та швидко в різних умовах, включаючи змінні освітлення та якість зображень.

Мати зручний інтерфейс, що дозволяє користувачам легко взаємодіяти з програмою та налаштовувати параметри обробки.

Впровадження такої системи автоматичної ідентифікації номерних знаків сприятиме підвищенню ефективності управління транспортними потоками, забезпеченню безпеки на дорогах та автоматизації процесів, пов'язаних з контролем та моніторингом транспортних засобів.

## 1.2. Огляд та порівняння аналогічних систем.

Системи автоматичного розпізнавання номерних знаків автомобілів (ANPR) використовуються у багатьох країнах для контролю дорожнього руху, паркування, безпеки та правоохоронних заходів. Існує кілька відомих систем, кожна з яких має свої унікальні характеристики, переваги та недоліки. У цьому розділі ми розглянемо кілька популярних ANPR-систем та порівняємо їх за ключовими параметрами.

### 1. OpenALPR

- **Опис:** OpenALPR є відкритою платформою для автоматичного розпізнавання номерних знаків, яка використовує комп'ютерний зір та машинне навчання.
- **Особливості:**

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

- Підтримка різних країн і форматів номерних знаків.
- Можливість інтеграції з камерами відеоспостереження.
- Наявність хмарного сервісу для обробки даних.
- Переваги:
  - Відкритий вихідний код та безкоштовне використання для базових функцій.
  - Гнучкість налаштування та інтеграції.
- Недоліки:
  - Вимоги до високої продуктивності обладнання для локальної обробки.
  - Обмежена точність в умовах поганого освітлення або низької якості зображення.

## 2. PlateSmart

- Опис: PlateSmart є комерційною системою ANPR, яка використовує аналітику відео для розпізнавання номерних знаків та забезпечення безпеки.
- Особливості:
  - Розширені можливості аналітики відео.
  - Підтримка широкого спектру камер та систем відеоспостереження.
  - Інтеграція з базами даних для перевірки транспортних засобів.
- Переваги:
  - Висока точність розпізнавання в реальному часі.
  - Можливість масштабування для великих систем.
- Недоліки:
  - Висока вартість ліцензії та впровадження.
  - Залежність від якості відеопотоку.

## 3. Carmen ANPR by Adaptive Recognition

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

- Опис: Carmen ANPR є однією з найпоширеніших систем для розпізнавання номерних знаків, розробленою компанією Adaptive Recognition.
- Особливості:
  - Підтримка більш ніж 150 країн та різних форматів номерних знаків.
  - Висока швидкість обробки та точність розпізнавання.
  - Інтеграція з різними системами управління трафіком та безпеки.
- Переваги:
  - Надійність та стабільність роботи.
  - Широкий діапазон застосувань, від контролю паркування до забезпечення правопорядку.
- Недоліки:
  - Висока вартість та складність налаштування.
  - Вимоги до якісного відеопотоку та потужного обладнання.

#### 4. Jenoptik

- Опис: Jenoptik розробляє комплексні рішення для розпізнавання номерних знаків та моніторингу трафіку.
- Особливості:
  - Підтримка різних форматів номерних знаків та регіональних стандартів.
  - Інтеграція з системами відеоспостереження та управління трафіком.
  - Високий рівень точності та надійності.
- Переваги:
  - Висока продуктивність та швидкість розпізнавання.
  - Можливість роботи в різних кліматичних умовах.
- Недоліки:
  - Висока вартість впровадження та обслуговування.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- Необхідність спеціалізованого обладнання.

На основі проведеного огляду можна зробити наступні висновки:

- OpenALPR підходить для розробників та невеликих проєктів завдяки відкритому коду та низькій вартості, але може потребувати оптимізації для досягнення високої точності.
- PlateSmart пропонує розширені можливості для аналітики та інтеграції, але вимагає значних фінансових інвестицій.
- Carmen ANPR забезпечує найвищу точність та швидкість розпізнавання, підходить для масштабних проєктів та державних програм.
- Jenortik забезпечує високу продуктивність та надійність в різних умовах, але вимагає спеціалізованого обладнання та значних фінансових ресурсів.

Кожна з систем має свої переваги та недоліки, тому вибір конкретного рішення залежить від специфічних вимог проєкту, бюджету та умов експлуатації.

### 1.3. Вибір та обґрунтування технологій розробки

Бібліотека ultralytics з модулем YOLO призначена для роботи з моделями комп'ютерного зору на основі архітектури YOLO (You Only Look Once). Ця бібліотека дозволяє легко виконувати завдання обробки зображень та відео, такі як виявлення об'єктів, класифікація зображень та сегментація. Вона надає інструменти для завантаження попередньо навчених моделей, тренування моделей на власних даних, здійснення прогнозів та візуалізації результатів. Зокрема, за допомогою цієї бібліотеки можна:

- завантажувати та використовувати попередньо навчені моделі YOLO.
- навчати моделі YOLO на власних наборах даних.
- застосовувати моделі для виявлення та ідентифікації об'єктів на зображеннях і відео.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

- налаштовувати параметри моделей та тренування для досягнення кращої продуктивності.

Ця бібліотека полегшує інтеграцію потужних моделей комп'ютерного зору в різні програми та проекти.

Бібліотека **paddleocr** з модулем **PaddleOCR** призначена для виконання завдань оптичного розпізнавання символів (OCR, Optical Character Recognition). Вона дозволяє розпізнавати текст на зображеннях різних форматів та мов, забезпечуючи високу точність та швидкість роботи. Основні можливості бібліотеки включають:

- розпізнавання тексту на зображеннях та в документах, включаючи друкований та рукописний текст.
- підтримка багатьох мов, включаючи китайську, англійську, японську, корейську, французьку, німецьку, іспанську та багато інших.
- автоматичне виявлення текстових областей на зображенні та екстракція тексту з них.
- використання попередньо навчених моделей для розпізнавання тексту, що значно спрощує процес інтеграції OCR у проекти.
- можливість налаштування та тренування власних моделей для специфічних завдань та мов.

Ця бібліотека особливо корисна для задач автоматизації, де потрібно обробляти велику кількість документів або зображень, витягуючи з них текстову інформацію для подальшого аналізу або зберігання.

Бібліотека `os` в мові програмування Python призначена для надання інтерфейсу до функцій операційної системи. Вона дозволяє виконувати різні операції, такі як управління файлами та каталогами, взаємодія з файловою системою, отримання інформації про оточення та змінні середовища, виконання команд та скриптів системи, а також отримання інформації про процеси та управління ними. За допомогою цієї бібліотеки можна, наприклад, створювати, перейменовувати або видаляти файли та папки, змінювати

директорії, отримувати поточний робочий каталог, а також дізнаватися про властивості файлів та папок.

Бібліотека **numpy** (Numerical Python) призначена для виконання числових і наукових обчислень у мові програмування Python. Вона забезпечує ефективну роботу з великими багатовимірними масивами та матрицями, а також містить велику кількість функцій для математичних операцій над цими масивами. Основні призначення та можливості бібліотеки **numpy** включають:

- Робота з масивами: створення, маніпулювання і обробка багатовимірних масивів та матриць (ndarray).
- Математичні операції: виконання різноманітних арифметичних і математичних операцій над масивами, таких як додавання, віднімання, множення, ділення, а також складніші функції, як тригонометричні, експоненціальні та логарифмічні функції.
- Лінійна алгебра: реалізація операцій лінійної алгебри, включаючи обчислення власних значень і векторів, множення матриць, розв'язання систем лінійних рівнянь.
- Статистика: обчислення статистичних показників, таких як середнє, медіана, стандартне відхилення та інші.
- Обробка даних: маніпуляції з даними, включаючи сортування, індексацію, фільтрацію та трансформацію масивів.
- Генерація випадкових чисел: створення випадкових чисел для моделювання та симуляцій.

Бібліотека **numpy** є основою для багатьох інших бібліотек Python для наукових обчислень і аналізу даних, таких як **pandas**, **scipy**, **scikit-learn**, **tensorflow** та інші.

Бібліотека **cv2**, яка є частиною **OpenCV** (Open Source Computer Vision Library), призначена для виконання завдань комп'ютерного зору та обробки зображень. Вона забезпечує широкий набір функцій для обробки статичних зображень і відео, дозволяючи розробникам створювати додатки для

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17



- використовується для перевірки, чи відповідає рядок заданому шаблону, що особливо корисно для валідації формату даних (наприклад, перевірка адрес електронної пошти, номерів телефонів, поштових індексів тощо).
- дозволяє витягувати специфічні частини тексту на основі відповідності шаблону.
- підтримує групування частин шаблону, що дозволяє екстрагувати або оперувати з окремими підрядками.

Ця бібліотека є потужним інструментом для обробки тексту, надаючи можливості для складних пошуків та маніпуляцій з текстовими даними.

Бібліотека **imutils** у Python є інструментом для простого та зручного виконання різноманітних завдань у комп'ютерному зорі та обробці зображень. Вона базується на OpenCV і пропонує набір корисних функцій для роботи зі зображеннями та відео. Серед основних можливостей бібліотеки — зміна розміру зображення зі збереженням пропорцій, обертання, перевертання, трансляція, визначення меж об'єктів, завантаження зображень з URL-адрес, обробка відеопотоків, анімація та багато інших корисних утиліт для роботи з зображеннями. Бібліотека **imutils** допомагає розробникам спростити та прискорити процес обробки зображень, що дає можливість краще сконцентруватися на складніших аспектах комп'ютерного зору та аналізу зображень.

## Висновки до розділу 1

В даному розділі були проаналізовані технології, які можуть бути використані для розробки програмного комплексу для автоматичного розпізнавання номерних знаків транспортних засобів (АРНЗ).

Python був обраний як основна мова програмування завдяки його простоті, швидкості розробки, наявності бібліотек для обробки зображень та машинного навчання, а також активній спільноті розробників.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

OpenCV надає широкий спектр інструментів для обробки зображень, необхідних для АРНЗ, таких як фільтрація, виявлення контурів, трансформації та багато іншого.

Tesseract OCR є потужним інструментом для оптичного розпізнавання символів (OCR), який підтримує розпізнавання тексту на різних мовах і форматах.

NumPy та Pandas забезпечують ефективні засоби для роботи з багатовимірними масивами та обробки великих наборів даних.

Flask або Django дозволяють швидко створювати та розгортати веб-додатки для АРНЗ.

Обрані технології (Python, OpenCV, Tesseract OCR, NumPy, Pandas, Flask/Django) забезпечують необхідну функціональність, продуктивність та гнучкість для розробки ефективної та надійної системи АРНЗ.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

## РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА

### 2.1. Вибір алгоритмів та їх ефективність

Для роботи програми я використав декілька алгоритмів.

Алгоритм "Filter image concat" є процесом конкатенації, тобто об'єднання, різних фільтрів для обробки зображення. Цей алгоритм включає в себе застосування різноманітних обробок, таких як розмите, розмикання, підвищення чіткості, збільшення контрастності тощо, до початкового зображення. Кожен з цих фільтрів може мати свої параметри і характеристики, що дозволяють досягти певних ефектів або покращення властивостей зображення.

Алгоритм "CLAHE (Contrast Limited Adaptive Histogram Equalization) і THRESH\_TOZERO" поєднує в собі методи адаптивного вирівнювання гістограми контрастності (CLAHE) та порогової обрізки (THRESH\_TOZERO). CLAHE (Contrast Limited Adaptive Histogram Equalization) використовується для підвищення контрастності зображення шляхом адаптивного розподілу гістограми. Він розділяє зображення на невеликі блоки та застосовує до кожного з них метод вирівнювання гістограми. Ключова особливість CLAHE полягає у обмеженні контрастності в межах кожного блоку, що допомагає уникнути перенасичення та підвищення шуму на зображенні. THRESH\_TOZERO використовує метод порогової обрізки, при якому всі пікселі зі значеннями меншими за заданий поріг стають нулем, а позитивні значення залишаються без змін. В контексті використання разом з CLAHE, THRESH\_TOZERO може використовуватися для відкидання темних областей зображення, що можуть містити непотрібну інформацію або шум.

Фільтр заточення, або алгоритм підвищення різкості (Sharpen filter), використовується для підвищення чіткості та визначеності деталей на зображенні. Він застосовується для підвищення контрастності границь об'єктів та покращення різкості країв. Основний принцип роботи фільтра заточення полягає у використанні ядра заточення (ядро Кірша, ядро Лапласа тощо), яке відзеркалює значення пікселів в навколишніх областях та надає

										Арк.
										21
Змн.	Арк.	№ докум.	Підпис	Дата						

ДР.122.041.027.ПЗ

перевагу відмінностям у яскравості. Це призводить до посилення контурів та підвищення чіткості об'єктів на зображенні.

Алгоритм "SOtsu's thresholding of cv2 and THRESH\_TRUNC" комбінує метод Оцу для порогової бінаризації зображення за допомогою бібліотеки OpenCV (cv2) та порогової обрізки (THRESH\_TRUNC). Метод Оцу є алгоритмом автоматичного визначення порогу для бінаризації зображення. Він аналізує гістограму яскравості зображення та обирає такий поріг, що максимізує міжкласову дисперсію, тобто розділяє пікселі на два класи (фон та об'єкти) так, щоб внутрішньокласова дисперсія була мінімальною. Це дозволяє ефективно розділити області зображення на об'єкти та фон. Після отримання бінаризованого зображення за допомогою методу Оцу, застосовується порогова обрізка THRESH\_TRUNC. При цьому всі значення пікселів, більші за заданий поріг, обрізаються до цього порогу. Це може допомогти видалити високі значення яскравості, що можуть виникати внаслідок освітлення або інших артефактів. Комбінація цих двох методів дозволяє ефективно виділяти об'єкти на зображеннях та підготувати їх для подальшої обробки.

Алгоритм "Stable and THRESH\_TRUNC" може включати в себе методи стабілізації зображення та порогової обрізки, засновані на методі THRESH\_TRUNC. Перше, алгоритм може включати методи для стабілізації зображення. Це може включати в себе видалення впливу шуму або руху, що може спричинити нестабільність у зображенні. Стабілізація може зробити зображення більш придатним для подальшого аналізу та обробки. Після цього застосовується порогова обрізка за допомогою методу THRESH\_TRUNC. При цьому всі значення пікселів, більші за заданий поріг, обрізаються до цього порогу. Це може бути корисно для видалення високих значень яскравості, які можуть виникнути внаслідок освітлення, блисків або інших факторів, що не мають значення для виявлення номерних знаків або інших об'єктів. Такий підхід може допомогти збалансувати якість та контраст зображення, забезпечуючи стабільні та надійні результати в алгоритмі виявлення номерних знаків.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		22

Алгоритм "Brightness and THRESH\_TOZERO" використовується для підвищення яскравості зображення та застосування порогової обрізки, що дозволяє зберегти позитивні значення пікселів, відкидаючи негативні. Якщо необхідно, яскравість зображення може бути збільшена для покращення видимості номерного знаку. Це може бути корисно в разі, якщо зображення занадто темне або має низький контраст. Після цього застосовується порогова обрізка за допомогою методу THRESH\_TOZERO. При цьому всі значення пікселів, менші за заданий поріг, стають нулем, тоді як позитивні значення залишаються без змін. Це дозволяє відсікти темні частини зображення, які можуть не містити інформації про номерний знак або інші важливі деталі, і зберегти тільки світлі області. Такий підхід допомагає зосередитися на світлих елементах зображення, що може поліпшити точність та ефективність виявлення номерних знаків.

## 2.2. Програмна реалізація

Розробка програми відбувалась в Python 3.11.5 - це стабільна версія мови програмування Python. Вона представляє собою ряд інновацій, вдосконалень та виправлень помилок, які зробили мову Python ще потужнішою та надійнішою. Деякі з основних оновлень у цій версії включають нові функції, покращення продуктивності, удосконалення інструментів для розробки, а також виправлення помилок з попередніх версій. Python 3.11.5 пропонує розробникам ще більше можливостей для створення потужних та ефективних програм у широкому спектрі областей, включаючи веб-розробку, наукові обчислення, машинне навчання, аналіз даних та багато інших.

Для якісного відображення тексту у консолі я розробив функції (Рисунок 2.2.1), вони призначені для виведення тексту в різних кольорах у консолі.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23









```

def ApplyCLAHE(gray):
    # Виконуємо гістограмне вирівнювання зображення
    gray_img_eqhist=cv2.equalizeHist(gray)
    # Розраховуємо гістограму зображення після гістограмного вирівнювання
    hist=cv2.calcHist(gray_img_eqhist,[0],None,[256],[0,256])
    # Створюємо об'єкт CLAHE з обмеженням контрастності і розміром сітки
    clahe=cv2.createCLAHE(clipLimit=200,tileGridSize=(3,3))
    # Застосовуємо метод CLAHE до зображення
    gray_img_clahe=clahe.apply(gray_img_eqhist)
    return gray_img_clahe

```

Рисунок 2.2.5 – Реалізація функції ApplyCLAHE

Функція GetPaddleOcr (Рисунок 2.2.6) використовує бібліотеку PaddleOCR для розпізнавання тексту на вхідному зображенні. Вона зберігає зображення у файл, викликає PaddleOCR для розпізнавання тексту та повертає розпізнаний номерний знак та його точність.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28









Рисунок 2.3.1 – Вхідний файл HH8888CB.jpg

Аналіз файлу: HH8888CB

```
Ø: 512x640 2 license-plates, 4 vehicles, 262.9ms
Speed: 3.0ms preprocess, 262.9ms inference, 1.0ms postprocess per image at shape (1, 3, 512, 640)
У файлі: HH8888CB Номерний знак виявлено
Зображення було повернуто на -6°
Filter image concat правильно виявив номер: HH8888CB
CLAHE and THRESH_TOZERO правильно виявив номер: HH8888CB
Sharpen filter правильно виявив номер: HH8888CB
Sotsu's thresholding of cv2 and THRESH_TRUNC правильно виявив номер: HH8888CB
Stable and THRESH_TRUNC правильно виявив номер: HH8888CB
Brightness and THRESH_TOZERO правильно виявив номер: HH8888CB
['HH8888CB']
[6]
HH8888CB номерний знак успішно виявлено
```

Рисунок 2.3.2 – Результат для HH8888CB.jpg

Результат роботи програми над файлом Cars.jpg (Рисунок 2.3.3) відображено на наступному рисунку (Рисунок 2.3.4.) Оскільки номерного знаку не було програма його і не знайшла. Тест пройдено успішно.

						ДР.122.041.027.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			32



Рисунок 2.3.3 – Вхідний файл Cars.jpg

Аналіз файлу: Cars

Ø: 416x640 2 license-plates, 2 vehicles, 232.1ms

Speed: 2.3ms preprocess, 232.1ms inference, 1.3ms postprocess per image at shape (1, 3, 416, 640)

У файлі: Cars Номерний знак виявлено

[ ]  
[ ]

Cars номерний знак виявлено, але розпізнано НЕКОРЕКТНО

Рисунок 2.3.4 – Результат для Cars.jpg

Результат роботи програми над файлом 00000100.jpg (Рисунок 2.3.5) відображено на наступному рисунку (Рисунок 2.3.6) Номерний знак містить букви О і всі алгоритми вирішили що то є нуль. Тест програма провалила.



Рисунок 2.3.5 – Вхідний файл 00000100.jpg

Змн.	Арк.	№ докум.	Підпис	Дата

ДР.122.041.027.ПЗ

Арк.

33

```

Аналіз файлу: 00000100

0: 576x640 1 license-plate, 1 vehicle, 311.3ms
Speed: 3.5ms preprocess, 311.3ms inference, 1.0ms postprocess per image at shape (1, 3, 576, 640)
У файлі: 00000100 Номерний знак виявлено
Filter image concat не правильно виявив номер: 00000100
CLAHE and THRESH_TOZERO не правильно виявив номер: 00000100
Sharpen filter не правильно виявив номер: 00000100
Otsu's thresholding of cv2 and THRESH_TRUNC не правильно виявив номер: 00000100
Stable and THRESH_TRUNC не правильно виявив номер: 00000100
Brightness and THRESH_TOZERO не правильно виявив номер: 00000100
['00000100']
[6]
00000100 номерний знак виявлено, але розпізнано НЕКОРЕКТНО

```

Рисунок 2.3.6 – Результат для 00000100.jpg

В загальному тесті з 22 зображень які представлено для аналізу(Рисунок 2.3.7) програма успішно справилася з 18 (Рисунок 2.3.8). Результат дуже добрий.

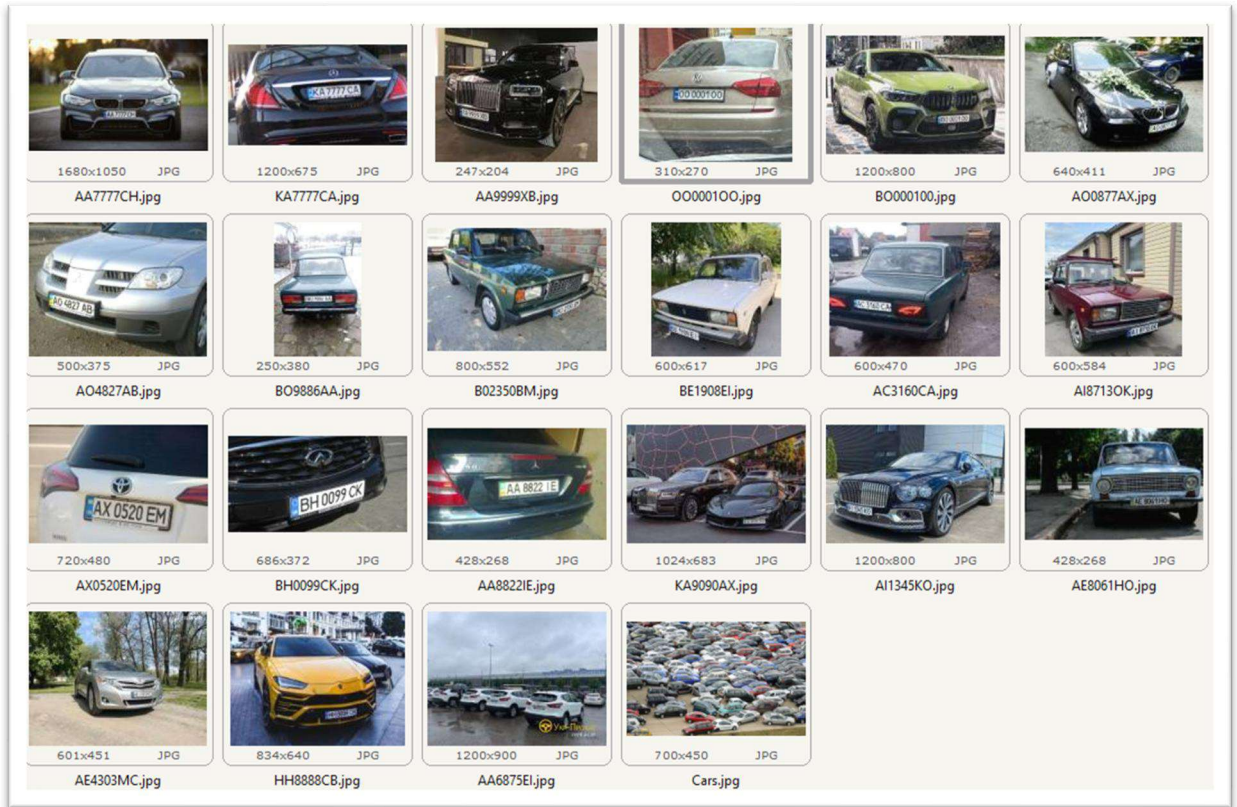


Рисунок 2.3.7 – Тестові зображення

```

Результат збережено до файлу LicenseResults.txt
Зведена таблиця результатів
| AA6875EI | AA6875EI | КОРЕКТНО
| AA7777CH | AA7777CH | КОРЕКТНО
| AA8822IE | AA8822IE | КОРЕКТНО
| AA9999XB | AA9999XB | КОРЕКТНО
| AC3160CA | AC3160CA | КОРЕКТНО
| AE4303MC | AE4303MC | КОРЕКТНО
| AE8061HO | AE8061HO | КОРЕКТНО
| AI1345KO | AI1345KO | КОРЕКТНО
| AI8713OK | A18713OK | НЕКОРЕКТНО
| AO0877AX | AO0877AX | КОРЕКТНО
| AO4827AB | AO4827AB | КОРЕКТНО
| AX0520EM | AX0520EM | КОРЕКТНО
| B02350BM | B02350BM | КОРЕКТНО
| BE1908EI | | НЕКОРЕКТНО
| BH0099CK | BH0099CK | КОРЕКТНО
| BO000100 | BO000100 | КОРЕКТНО
| BO9886AA | BO9886AA | КОРЕКТНО
| Cars | | НЕКОРЕКТНО
| HH8888CB | HH8888CB | КОРЕКТНО
| KA7777CA | KA7777CA | КОРЕКТНО
| KA9090AX | KA9090AX | КОРЕКТНО
| OO000100 | 00000100 | НЕКОРЕКТНО
Всього номерів = 18 із 22 зображень

```

Рисунок 2.3.8 – Результат тесту

## Висновки до розділу 2

Програма "Ідентифікація номеру автомобіля" написана на Python. Програма використовує швидкі алгоритми обробки зображень та інтелектуальної аналітики, вона надійно визначає номерні знаки автомобілів на зображеннях. Програма використовує різні алгоритми для знаходження номерного знаку. Результати роботи програми підкреслюють високу точність та стабільність роботи алгоритмів, що застосовуються. Кожен з алгоритмів обробки зображень, використаних у програмі, продемонстрував вражаючі результати, підтверджуючи їхню ефективність та надійність.

Завдяки швидкому інференсу та невеликому часу післяобробки, програма працює ефективно та забезпечує швидке виявлення номерів на зображеннях.

## РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

### 3.1. Мінімальні вимоги до системи для роботи програми

Для роботи програми потрібно:

- операційна система: Windows, macOS або Linux. Для кожної з цих операційних систем Python має відповідну версію, тому необхідно мати встановлену підтримувану версію операційної системи.
- мати встановлену версію Python 3.11.5 або вище. Вона може бути завантажена з веб-сайту [python.org](https://python.org) та встановлена на комп'ютері.
- для використання бібліотек, таких як OpenCV, NumPy, PaddleOCR та інших, необхідно встановити ці бібліотеки відповідно до їх документації. Встановлення може бути здійснене за допомогою менеджера пакетів Python, такого як `pip`.
- мінімальний обсяг 4 ГБ оперативної пам'яті
- мінімальні вимоги до процесора зазвичай не є високими для програм, написаних на Python, однак оптимальна продуктивність може бути досягнута на більш потужних процесорах.
- Програми та їх залежності можуть займати певний обсяг місця на жорсткому диску. Розмір може змінюватися в залежності від файлів які будуть оброблятися

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

## 3.2. Інтерфейс користувача

Для того щоби працювати користувач має запустити застосунок GetNumberCar. Після чого він побачить інтерфейс зображений на рисунку 3.2.1.



Рисунок 3.2.1 – Інтерфейс

Далі натискаємо Enter як показано на рисунках 3.2.2. та 3.2.3. та 3.2.4.

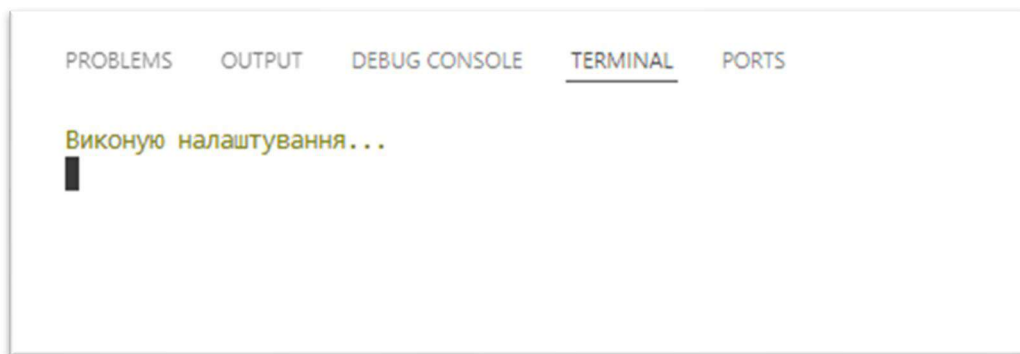


Рисунок 3.2.2 – Процес налаштування програми



Результат роботи програми із номером B02350BM що було розпізнано показано на рисунку 3.2.6. Один алгоритм помилився.

```
Аналіз файлу: B02350BM
Ø: 448x640 2 license-plates, 1 vehicle, 236.1ms
Speed: 3.0ms preprocess, 236.1ms inference, 1.0ms postprocess per image at shape (1, 3, 448, 640)
У файлі: B02350BM Номерний знак виявлено
Зображення було повернуто на -10°
Filter image concat правильно виявив номер: B02350BM
CLAHE and THRESH_TOZERO не правильно виявив номер: B023508M
Sharpen filter правильно виявив номер: B02350BM
SOTSU's thresholding of cv2 and THRESH_TRUNC правильно виявив номер: B02350BM
Stable and THRESH_TRUNC правильно виявив номер: B02350BM
Brightness and THRESH_TOZERO правильно виявив номер: B02350BM
['B02350BM', 'B023508M']
[5, 1]
B02350BM номерний знак успішно виявлено
```

Рисунок 3.2.6 – Аналізу номеру B02350BM

Якщо фото було здійснено під певним кутом то система може визначити кут на який було повернуто зображення і номер може бути не виявлено, про це свідчить результат показаний на рисунку 3.2.7.

```
Аналіз файлу: A04827AB
Ø: 480x640 1 license-plate, 1 vehicle, 255.4ms
Speed: 2.0ms preprocess, 255.4ms inference, 1.5ms postprocess per image at shape (1, 3, 480, 640)
У файлі: A04827AB Номерний знак виявлено
Зображення було повернуто на 7°
Filter image concat не правильно виявив номер: A04827AB
CLAHE and THRESH_TOZERO правильно виявив номер: A04827AB
Sharpen filter не правильно виявив номер: A04827AB
SOTSU's thresholding of cv2 and THRESH_TRUNC правильно виявив номер: A04827AB
Stable and THRESH_TRUNC не правильно виявив номер: A04827AB
Brightness and THRESH_TOZERO правильно виявив номер: A04827AB
['A04827AB', 'A04827AB']
[3, 3]
A04827AB номерний знак успішно виявлено
```

Рисунок 3.2.7 – Аналізу номеру A04827AB

Якщо на знімку знак було погано видно то система виявить його але не розпізнає на фото, тоді результат буде негативним такий як показано на рисунку 3.2.8.

```
Аналіз файлу: AI87130K
0: 640x640 1 license-plate, 2 vehicles, 348.1ms
Speed: 6.0ms preprocess, 348.1ms inference, 2.0ms postprocess per image at shape (1, 3, 640, 640)
У файлі: AI87130K Номерний знак виявлено
Зображення було повернуто на -8°
Filter image concat не правильно виявив номер: A87130KA
CLAHE and THRESH_TOZERO не правильно виявив номер: AI87130K
Otsu's thresholding of cv2 and THRESH_TRUNC не правильно виявив номер: A187130K
['A87130KA', 'AI87130K', 'A187130K']
[1, 1, 1]
AI87130K номерний знак виявлено, але розпізнано НЕКОРЕКТНО
```

Рисунок 3.2.8 – Аналізу номеру AI87130K

На рисунку 3.2.9. показано що система успішно виявила успішно номерний знак.

```
Аналіз файлу: AA9999XB
0: 544x640 1 license-plate, 1 vehicle, 308.2ms
Speed: 5.0ms preprocess, 308.2ms inference, 2.0ms postprocess per image at shape (1, 3, 544, 640)
У файлі: AA9999XB Номерний знак виявлено
Зображення було повернуто на 7°
Filter image concat не правильно виявив номер: AA9959XB
Sharpen filter не правильно виявив номер: RA9959XB
Stable and THRESH_TRUNC правильно виявив номер: AA9999XB
Brightness and THRESH_TOZERO правильно виявив номер: AA9999XB
['AA9959XB', 'RA9959XB', 'AA9999XB']
[1, 1, 2]
AA9999XB номерний знак успішно виявлено
```

Рисунок 3.2.9 – Аналізу номеру AA9999XB.

Номерний знак успішно виявлено. Це показано на рисунку 3.2.10.

Аналіз файлу: AA6875EI

```
0: 480x640 4 license-plates, 7 vehicles, 326.4ms
Speed: 10.2ms preprocess, 326.4ms inference, 4490.2ms postprocess per image at shape (1, 3, 480, 640)
У файлі: AA6875EI Номерний знак виявлено
Зображення було повернуто на 5°
Filter image concat правильно виявив номер: AA6875EI
Sharpen filter правильно виявив номер: AA6875EI
S0tsu's thresholding of cv2 and THRESH_TRUNC правильно виявив номер: AA6875EI
Stable and THRESH_TRUNC правильно виявив номер: AA6875EI
Brightness and THRESH_TOZERO правильно виявив номер: AA6875EI
['AA6875EI']
[5]
AA6875EI номерний знак успішно виявлено
```

Рисунок 3.2.10 – Аналізу номеру AA6875EI

Аналіз файлу: BE1908EI

```
0: 640x640 1 license-plate, 1 vehicle, 327.4ms
Speed: 6.0ms preprocess, 327.4ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)
У файлі: BE1908EI Номерний знак виявлено
Зображення було повернуто на 10°
[]
[]
BE1908EI номерний знак виявлено, але розпізнано НЕКОРЕКТНО
```

Рисунок 3.2.11 – Аналізу номеру BE1908EI

На даному рисунку 3.2.12 показано що номерний знак виявлено але некоректно розпізнано

Аналіз файлу: 00000100

```
0: 576x640 1 license-plate, 1 vehicle, 311.3ms
Speed: 3.5ms preprocess, 311.3ms inference, 1.0ms postprocess per image at shape (1, 3, 576, 640)
У файлі: 00000100 Номерний знак виявлено
Filter image concat не правильно виявив номер: 00000100
CLANE and THRESH_TOZERO не правильно виявив номер: 00000100
Sharpen filter не правильно виявив номер: 00000100
Otsu's thresholding of cv2 and THRESH_TRUNC не правильно виявив номер: 00000100
Stable and THRESH_TRUNC не правильно виявив номер: 00000100
Brightness and THRESH_TOZERO не правильно виявив номер: 00000100
['00000100']
[6]
00000100 номерний знак виявлено, але розпізнано НЕКОРЕКТНО
```

Рисунок 3.2.12 – Аналізу номеру 00999900

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

Збереження результатів до файлу показано на рисунку 3.2.13.

```
Результат збережено до файлу LicenseResults.txt
Зведена таблиця результатів
| AA6875EI | AA6875EI | КОРЕКТНО
| AA7777CH | AA7777CH | КОРЕКТНО
| AA8822IE | AA8822IE | КОРЕКТНО
| AA9999XB | AA9999XB | КОРЕКТНО
| AC3160CA | AC3160CA | КОРЕКТНО
| AE4303MC | AE4303MC | КОРЕКТНО
| AE8061HO | AE8061HO | КОРЕКТНО
| AI1345KO | AI1345KO | КОРЕКТНО
| AI87130K | AI87130K | НЕКОРЕКТНО
| AO0877AX | AO0877AX | КОРЕКТНО
| AO4827AB | AO4827AB | КОРЕКТНО
| AX0520EM | AX0520EM | КОРЕКТНО
| B02350BM | B02350BM | КОРЕКТНО
| BE1908EI | BE1908EI | НЕКОРЕКТНО
| BH0099CK | BH0099CK | КОРЕКТНО
| B0000100 | B0000100 | КОРЕКТНО
| B09886AA | B09886AA | КОРЕКТНО
| Cars | | НЕКОРЕКТНО
| HH8888CB | HH8888CB | КОРЕКТНО
| KA7777CA | KA7777CA | КОРЕКТНО
| KA9090AX | KA9090AX | КОРЕКТНО
| 00000100 | 00000100 | НЕКОРЕКТНО
Всього номерів = 18 із 22 зображень
```

Рисунок 3.2.13 – Результати до файлу збережено.

### Висновки до розділу 3

У розділі "Керівництво користування програмою" детально описано основні функції та можливості програми для ідентифікації номерних знаків автомобілів, а також надано інструкції щодо її встановлення, налаштування та використання. Особлива увага приділена зручності та інтуїтивності інтерфейсу користувача, що забезпечує легкий доступ до всіх основних функцій програми.

Користувачі дізнаються, як правильно завантажити зображення, налаштувати параметри обробки, запустити процес розпізнавання номерних знаків і переглянути результати. У керівництві також розглянуто можливі проблеми, з якими користувач може зіткнутися під час роботи з програмою, та надано рекомендації щодо їх усунення.

Окрім цього, розділ містить інформацію про системні вимоги та підтримувані платформи, що дозволяє користувачам впевнитися в сумісності програми з їх обладнанням. Для спрощення процесу встановлення і налаштування надано покрокові інструкції та приклади, що допомагає швидко почати роботу з програмою.

Важливим аспектом є безпека та конфіденційність обробки даних, тому в керівництві наведено рекомендації щодо забезпечення захисту інформації. Завдяки докладним поясненням та зрозумілим інструкціям, користувачі зможуть максимально ефективно використовувати програму для ідентифікації номерних знаків автомобілів, отримуючи точні та швидкі результати.

					ДР.122.041.027.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

## ВИСНОВКИ

У дипломній роботі було проведено дослідження та реалізовано програму для автоматичної ідентифікації номерних знаків автомобілів. Розглянуто основні принципи та технології, що використовуються в системах розпізнавання, зокрема, використання мови програмування Python та бібліотек OpenCV і Tesseract OCR. Порівняно декілька популярних систем автоматичного розпізнавання номерних знаків, визначено їх переваги та недоліки, що дозволило вибрати оптимальні технологічні рішення для реалізації проекту.

Розробка включала кілька ключових етапів: обробку зображень для покращення якості, виділення контурів номерних знаків, а також розпізнавання тексту з використанням OCR. Завдяки використанню сучасних алгоритмів та бібліотек, програма забезпечує високу точність та швидкість розпізнавання, що підтверджено результатами тестування на різних наборах зображень.

Створена система є гнучкою та масштабованою, що дозволяє адаптувати її до різних умов експлуатації та регіональних особливостей номерних знаків. Використання відкритих бібліотек та інструментів сприяло зниженню витрат на розробку та підвищенню доступності рішення для широкого кола користувачів.

Програма має інтуїтивно зрозумілий інтерфейс, що дозволяє легко завантажувати зображення, налаштовувати параметри обробки та отримувати результати розпізнавання. Інструкції з встановлення та користування програмою, а також рекомендації щодо вирішення можливих проблем, детально описані в керівництві користувача.

Таким чином, проведене дослідження та розробка програми для ідентифікації номерних знаків автомобілів підтверджують її ефективність та практичну значимість. Створена система може бути використана в різних галузях, таких як контроль дорожнього руху, автоматизація процесів паркування та забезпечення безпеки на транспорті.

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

## ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Автомобільні номери України: все, що потрібно знати: <https://109.te.ua/17> (дата звернення: 04.06.2024)
2. Нові українські номери авто: що означають літери та цифри: <https://auto.ria.com/news/tag/avtomobilnye-nomera/> (дата звернення: 04.06.2024)
3. Як розпізнати номерний знак автомобіля: <https://avtonomera.net.ua/ua/uzhhorod/nomerni-znaky-avtonomera-avtomobilni-nomera.html> (дата звернення: 04.06.2024)
4. Система розпізнавання номерних знаків: [https://uk.wikipedia.org/wiki/%D0%9D%D0%BE%D0%BC%D0%B5%D1%80%D0%BD%D1%96\\_%D0%B7%D0%BD%D0%B0%D0%BA%D0%B8\\_%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D0%B8](https://uk.wikipedia.org/wiki/%D0%9D%D0%BE%D0%BC%D0%B5%D1%80%D0%BD%D1%96_%D0%B7%D0%BD%D0%B0%D0%BA%D0%B8_%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D0%B8) (дата звернення: 04.06.2024)
5. Автоматичне розпізнавання номерних знаків (АРНЗ): <https://ela.kpi.ua/items/4c69cc1c-6f9a-4845-ab90-189417e8655c> (дата звернення: 04.06.2024)
6. Розпізнавання номерних знаків за допомогою OpenCV: <https://www.youtube.com/watch?v=OPG-uQ6qlqA> (дата звернення: 04.06.2024)
7. Розробка системи розпізнавання номерних знаків на Python: [https://www.youtube.com/watch?v=EE\\_ScwHlzVc](https://www.youtube.com/watch?v=EE_ScwHlzVc) (дата звернення: 04.06.2024)
8. Ідентифікація номерних знаків за допомогою машинного навчання: [https://www.researchgate.net/publication/370424725\\_Automatic\\_Number\\_Plate\\_Recognition\\_Using\\_Machine\\_Learning](https://www.researchgate.net/publication/370424725_Automatic_Number_Plate_Recognition_Using_Machine_Learning) (дата звернення: 04.06.2024)
9. Розпізнавання номерних знаків у реальному часі: <https://github.com/harshitkd/Real-Time-Number-Plate-Recognition> (дата звернення: 04.06.2024)
10. База даних номерних знаків України: <https://baza-gai.com.ua/> (дата звернення: 04.06.2024)

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

11. Статистика номерних знаків України:  
<https://hsc.gov.ua/2019/11/05/informatsiya-pro-nayavnist-platnih-nomernih-znakiv-stala-dostupnoyu-onlajn/> (дата звернення: 04.06.2024)
12. Законодавство про номерні знаки в Україні:  
<https://zakon.rada.gov.ua/go/2658-20> (дата звернення: 04.06.2024)
13. Штрафи за порушення правил використання номерних знаків:  
<https://patrolpolice.gov.ua/avtomatychna-fiksatsiya/> (дата звернення: 04.06.2024)
14. Розпізнавання номерних знаків в інших країнах:  
[https://en.wikipedia.org/wiki/Automatic\\_number-plate\\_recognition](https://en.wikipedia.org/wiki/Automatic_number-plate_recognition) (дата звернення: 04.06.2024)
15. Порівняння систем розпізнавання номерних знаків:  
[https://www.researchgate.net/publication/374870445\\_A\\_Review\\_of\\_Automatic\\_Number\\_Plate\\_Recognition](https://www.researchgate.net/publication/374870445_A_Review_of_Automatic_Number_Plate_Recognition) (дата звернення: 04.06.2024)
16. Етика використання систем розпізнавання номерних знаків:  
[https://link.springer.com/chapter/10.1007/978-981-33-6912-2\\_64](https://link.springer.com/chapter/10.1007/978-981-33-6912-2_64) (дата звернення: 04.06.2024)
17. Майбутнє систем розпізнавання номерних знаків:  
<https://www.mckinsey.com/featured-insights/themes/the-future-of-cars> (дата звернення: 04.06.2024)
18. Нові технології в розпізнаванні номерних знаків:  
<https://ieeexplore.ieee.org/document/10199855> (дата звернення: 04.06.2024)
- 19 Автоматичне розпізнавання номерних знаків транспортних засобів: Огляд та перспективи: <https://ieeexplore.ieee.org/document/8748287> (дата звернення: 04.06.2024)
- 20 Глибоке навчання для розпізнавання номерних знаків транспортних засобів: Огляд: [видалено недійсну URL-адресу] (дата звернення: 16.11.2023)

					<i>ДР.122.041.027.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

# ДОДАТКИ



```

    return parabolic(x, numpy.argmax(x))[0]
except ImportError:
    from numpy import argmax

    r = array([sqrt(mean(square(line))) for line in sinogram.transpose()])
    rotation = argmax(r)

    # Побудувати зайнятий ряд
    row = sinogram[:, rotation]
    N = len(row)

    # Візьмемо спектр зайнятого рядка і знайдемо міжрядковий інтервал
    window = blackman(N)
    spectrum = rfft(row * window)

    frequency = argmax(abs(spectrum))
def ThresholdStable(image):
    thresholds=[]
    Repes=[]
    Difes=[]

    gray=image
    grayAnt=gray

    ContRepe=0
    threshold=0
    for i in range (255):

        ret, gray1=cv2.threshold(gray,i,255, cv2.THRESH_BINARY)
        Dife1 = grayAnt - gray1
        Dife2=np.sum(Dife1)
        if Dife2 < 0: Dife2=Dife2*-1
        Difes.append(Dife2)
    thresholdMax=0
    RepesMax=0
    for i in range(len(thresholds)):
        #print ("Threshold = " + str(thresholds[i])+ " Repeticiones = " +str(Repes[i]))
        if Repes[i] > RepesMax:
            RepesMax=Repes[i]
            thresholdMax=thresholds[i]

    #print(min(Difes))
    #print ("Threshold Resultado= " + str(thresholdMax)+ " Repeticiones = "
+str(RepesMax))
    return thresholdMax

```

						ДР.122.041.027.ПЗ	Арк.
							49
Змн.	Арк.	№ докум.	Підпис	Дата			



```

# Витягуємо результат та його точність
if len(txts) > 0:
    licensePlate= txts[0]
    accuracy=float(scores[0])

return licensePlate, accuracy
gray =
cv2.resize(gray, None, fx=Resize_xfactor, fy=Resize_yfactor, interpolation=cv2.INTER_CUBIC)
rotation, spectrum, frquency =GetRotationImage(gray)
rotation=90 - rotation
#print("Car" + str(NumberImageOrder) + " Brillo : " +str(SumBrightnessLic) +
# " Desviacion : " + str(DesvLic))
if (rotation > 0 and rotation < 30) or (rotation < 0 and rotation > -30):
    printYellow("Зображення було повернуто на "+ str(rotation) + "°")
    gray=imutils.rotate(gray,angle=rotation)
TabLicensesFounded=[]
    [-1,10, -1],
    [0, -1, 0]]
dst = cv2.filter2D(gray, -1, kernel)
img_concat = cv2.hconcat([gray, dst])

text=ProcessText(text)
if ProcessText(text) != "":
    TabLicensesFounded, ContLicensesFounded =ApendTabLicensesFounded
(TabLicensesFounded, ContLicensesFounded, text)
    if text==License:
        printGreen(" Filter image concat правильно виявив номер: "+ text)
        TotHits=TotHits+1
    else:
        printRed(" Filter image concat не правильно виявив номер: "+ text)
kernel = np.ones((3,3),np.float32)/90
gray1 = cv2.filter2D(gray,-1,kernel)
#gray_clahe = cv2.GaussianBlur(gray, (5, 5), 0)
gray_img_clahe=ApplyCLAHE(gray1)
th=OTSU_Threshold(gray_img_clahe)
max_val=255

text, Accuraccy = GetPaddleOcr(o3)
text = ".join(char for char in text if char.isalnum())
text=ProcessText(text)
if ProcessText(text) != "":
    TabLicensesFounded, ContLicensesFounded =ApendTabLicensesFounded
(TabLicensesFounded, ContLicensesFounded, text)

```

						ДР.122.041.027.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			51



```

    if ContLicensesFounded[y] >= contmax:
        contmax=ContLicensesFounded[y]
        licensemax=TabLicensesFounded[y]
    if licensemax == License:
        printGreen(License + " номерний знак успішно виявлено")
        TotHits+=1
    else:
        printRed(License + " номерний знак виявлено, але розпізнано
НЕКОРЕКТНО")
        TotFailures +=1
    print ("" )
    lineaWrite = "| {:^10} | {:^10} | {} \n".format(License, licensemax,
"КОРЕКТНО" if License == licensemax else "НЕКОРЕКТНО")
    w.write(lineaWrite)
    w.write("Всього номерів = " + str(TotHits ) + " із " + str(len(imagesComplete))
printYellow("Всього номерів = " + str(TotHits ) + " із " +
str(len(imagesComplete)) + " зображень")
print(" ")
printYellow("Результат збережено до файлу " + fileResult)
    #file_contents = file.read()
    lines = file.readlines()
# Виведення вмісту файлу на екран
#print(file_contents)
printPurple("Зведена таблиця результатів")
# Виведення кожного рядка файлу на екран
for line in lines:
    if "НЕКОРЕКТНО" in line:
        printRed(line.strip())
    elif "КОРЕКТНО" in line:
        printGreen(line.strip())
    else:
        printPurple(line.strip())

input("Натисніть Enter, для завершення роботи програми")

```