|  |  |
| --- | --- |
| **Ain Shams University**  **Faculty of Computer & Information Sciences** |  |

**ANPR**

**Automatic Number Plate Recognition**

**Proposed by:**

**Mohamed Ashraf Elewa** [Information Science]

**Rawan Yahia Sayed** [Information Science]

**Eslam Ragab Mohamed** [Information Science]

**Sara Mohamed Kamel** [Information Science]

**Lamiaa Emad Mohamed** [Information Science]

**Ahmed Nasreldeen Mohamed** [Information Science]

**Supervisors**

*Prof. Dr.* *Nivin Atef*

*Information Sciences Department*

*Faculty of Computer and Information Sciences,*

*Ain Shams University*

*T.A. Nourhan Bahnasy*

*Information Sciences Department*

*Faculty of Computer and Information Sciences,*

*Ain Shams University*

**Cairo, June 2024**

**Acknowledgments**

All praise and thanks to ALLAH, who provided me with the ability to complete this work. I hope to accept this work from me.

I am grateful of *my parents* and *my family* who are always providing help and support throughout the whole years of study. I hope I can give that back to them.

I also offer my sincerest gratitude to my supervisors, *Prof. Dr. Nivin Atef and   
T.A. Nourhan Bahnasy.* who have supported me throughout my thesiswith their patience, knowledge and experience.

Finally, I would like to thank my friends and all people who gave me support and encouragement.

**Abstract**

This project focuses on the development of an Automatic Number Plate Recognition (ANPR) system,

a crucial technology used for vehicle identification and management. The primary problem addressed by this project is the need for an efficient, accurate, and automated method to recognize vehicle license plates, which is essential for applications such as traffic law enforcement, toll collection, and parking management.

Egypt's automated traffic fines, using license plate cameras, aim to catch drivers faster and more precisely. But these automated systems can sometimes make mistakes, wrongly fining drivers. This can be unfair! By fixing these in the system, we can make it fairer and keep our roads safer.

The project employs advanced image processing and machine learning techniques. Key functionalities include vehicle detection, precise license plate localization, character segmentation, and character recognition using Optical Character Recognition (OCR). Environmental noise reduction techniques are applied to improve accuracy under varying conditions. The system also incorporates data verification against a stored database and plans to integrate plate history lookup for violation tracking and security checks. The training model leverages ethically sourced data from public databases like Roboflow, Kaggle, and GitHub. The system demonstrates a high accuracy rate in identifying and recognizing license plates under diverse environmental conditions. Newly observed facts include the significant improvement in recognition accuracy through specific pre-processing techniques such as noise filtering and contrast enhancement. The system also shows robustness in handling different lighting conditions and weather scenarios, ensuring reliable performance. Plate localization Accuracy Epoch50 98.99%. The developed system effectively addresses the problem of unjust fines, providing a fairer and more reliable method for automated traffic enforcement. The argument emphasizes the importance of integrating advanced image processing and machine learning techniques to enhance system accuracy and reliability. By eliminating erroneous fines and improving detection accuracy, the system not only saves time and resources but also contributes to safer roads and better traffic management in Egypt.

**Table of Contents**

**Acknowledgments.…….……..……………..……………………………….i**

**Abstract………….………………...………………….………………………ii**

**List of Figures….……………………………….............…………….………iii**

**List of Tables…………...……………………………….……….……….…...iv**

**List of Abbreviations….……………………..…………………….…………v**

**List of Symbols…… ….……………………..…………………….…………vi**

**Chapter 1: Introduction………………………………………………...……1**

**1.1 Problem Definition…………………………………………1**

**1.2 Motivation…………………………………………………2**

**1.3 Objectives………………………………………………….2**

**1.4 Methodology** **………………………………………………3**

**1.5 Time plan………………………………………………….3**

**1.6 Thesis Outline ……………… ……………………… ……4**

**Chapter 2: Literature Review …………………….…..…………….5**

**Chapter 3: System Architecture and Methods ...…………………..**

**3.1 System Architecture ………………………………………**

**3.2 Description of methods and procedures used …………**

**Chapter 4: System Implementation and Results …..………………..**

**4.1 Dataset …..………………..………………..……………**

**4.2** **Description of Software Tools Used ..………………..………**

**4.3 Stepup Configuration (hardware)**

**4.4 Experimental and Results …..…………………..……**

**Chapter 5: Run the Application....…………………………..…..**

**Chapter 6: Conclusion and Future Work………….…………………..**

**6.1 Conclusion…………………………………………………………**

**6.2 Future Work………………………………………………………**

**References……………………………………………………………….…….**

**List of Figures**

**[State all figures in your documentation with page number]**

**Format**

Chapter-number.Figure-number: Figure title…………..…………….page number

**For example**

Figure 1.1: Time Plane …..………………………………………………………11

Figure 1.2: System Architecture….………………………………………………20

**List of Tables**

**[State all tables in your documentation with page number]**

Table 1.1: Results of (….).……………...……………………………………21

Table 1.2: System Advantages and Disadvantages…….……………………39

**List of Abbreviations**

**[State all abbreviations in your documentation]**

SVM: Support Vector Machine

NNs: Neural Networks

**Chapter One:**

**Introduction**

1. [Introduction: 13](https://docs.google.com/document/d/1QXsvFDZSWKm3FPuEzFANYijyNhfdUSR3/edit#heading=h.3rdcrjn)
   1. [Overview 13](https://docs.google.com/document/d/1QXsvFDZSWKm3FPuEzFANYijyNhfdUSR3/edit#heading=h.26in1rg)
   2. [History: 13](https://docs.google.com/document/d/1QXsvFDZSWKm3FPuEzFANYijyNhfdUSR3/edit#heading=h.lnxbz9)
   3. [Problem Definition: 14](https://docs.google.com/document/d/1QXsvFDZSWKm3FPuEzFANYijyNhfdUSR3/edit#heading=h.35nkun2)
   4. [Motivation: 15](https://docs.google.com/document/d/1QXsvFDZSWKm3FPuEzFANYijyNhfdUSR3/edit#heading=h.1ksv4uv)
   5. [Objectives 15](https://docs.google.com/document/d/1QXsvFDZSWKm3FPuEzFANYijyNhfdUSR3/edit#heading=h.44sinio)
   6. [Applications: 16](https://docs.google.com/document/d/1QXsvFDZSWKm3FPuEzFANYijyNhfdUSR3/edit#heading=h.2jxsxqh)
   7. [Document Outline: 18](https://docs.google.com/document/d/1QXsvFDZSWKm3FPuEzFANYijyNhfdUSR3/edit#heading=h.3j2qqm3)

**Chapter One:**

**Introduction**

* 1. **Problem Definition:**

In recent years, the increasing number of registered vehicles in Egypt has posed significant challenges for traffic management and law enforcement. Traditional manual monitoring methods have proven to be inefficient and error-prone, often leading to incorrect fines and missed violations. This situation has underscored the urgent need for more advanced and reliable systems to handle the growing traffic volumes and ensure fair enforcement of traffic laws.

Unjustified fines create financial issue, while false accusations can harm one's legal standing , impacting employment and personal life, The potential for abuse in traffic enforcement systems raises concerns about integrity. Inaccurate car plate detection that cause misleading information about vehicles in crimes:​ Unjustified fines, False Accusations and Potential for Abuse.

* 1. **Motivation:**

Developing a system that automates the recognition of license plates is a critical advancement in modern traffic management and law enforcement. The motivation for this project is driven by several key factors that highlight the necessity and benefits of such technology.

First and foremost, the efficiency gains from automating license plate recognition are substantial. With the growing number of vehicles on the road, manual monitoring and enforcement are increasingly impractical and resource-intensive. An automated system can process vast amounts of data in real-time, significantly reducing the workload on human operators and allowing for more effective allocation of resources.

Accuracy is another crucial aspect. Manual methods are prone to human error, leading to incorrect fines and missed violations. By leveraging advanced image processing and machine learning techniques, an automated system can achieve high levels of accuracy in identifying and reading license plates, ensuring that only legitimate violations are recorded and penalized. This reduces the occurrence of unjust fines and enhances the overall fairness of the traffic enforcement process.

Security is also greatly enhanced with automated license plate recognition. Accurate and reliable identification of vehicles plays a vital role in various security applications, including tracking stolen vehicles, monitoring for unauthorized use, and managing access control in sensitive areas. By upgrading automatic identification and monitoring capabilities, the system can provide real-time alerts and detailed records, contributing to a safer and more secure environment.

Moreover, the application of such a system extends beyond traffic enforcement. It can be integrated into toll collection, parking management, and border control, among other uses. This versatility makes it a valuable tool for improving the efficiency and effectiveness of numerous public and private sector operations.

Finally, the overarching goal of this project is to enhance public safety. By ensuring accurate and fair enforcement of traffic laws, the system helps to deter violations, reduce accidents, and promote responsible driving behavior. The improved management of traffic flows and the ability to quickly respond to incidents further contribute to safer roads and communities.

In summary, the motivation for developing an automated license plate recognition system lies in its potential to provide significant improvements in efficiency, accuracy, and security across various applications, ultimately contributing to enhanced public safety and better resource management.

* 1. **Objectives:**

**Eliminate Unjust Fines:** ensuring that individuals are not burdened by financial issues that they did not do. ​

**Save Valuable Time:** free up both individual and administrative time for more productive thing.​

**Enhancement road safety and traffic management** : help to catch correctly drivers breaking the rule and  making roads safer.​

**Handling External Factor :** Under any condition can detect more accurately.​

* 1. **Methodology:**

ANPR (Automatic Number Plate Recognition) System project is structured around a comprehensive implementation strategy and the application of advanced methodologies to ensure high accuracy and efficiency. The project incorporates multiple phases, including data acquisition, preprocessing, model training, and performance evaluation.

* 1. **Time Plan:**

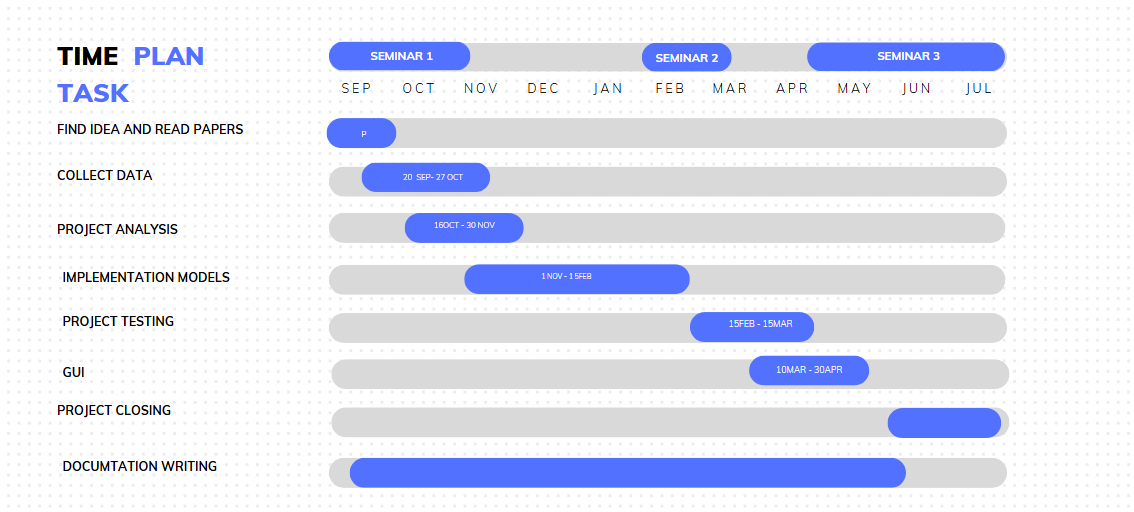


Figure 1

**Project Phases:**

1. **Find idea and read papers**

Resource: Google Scholar

Start Date:1-Sep-2023 End Date: 20-Sep-2023

1. **collect data**

Start Date: 20-Sep-2023 End Date: 27-Oct-2023

1. **project analysis**

Start Date: 16-Oct-2023 End Date: 30-Nov-2023

1. **implementation models**

Resource: Google Colab

Start Date: 1-Nov-2023 End Date: 15-Feb-2024

1. **project testing**

Start Date: 15-Feb-2024 End Date: 15-Mar-2024

1. **GUI**

Start Date: 10-Mar-2024 End Date: 30-Apr-2024

1. **Project closing**

Start Date: 30-Apr-2024

1. **Documentation writing**

Resource: Microsoft Word

Start Date: 16-Oct-2023 End Date: 15-Jun-2024

* 1. **Thesis Outline:**
* **Chapter 1: Introduction**

This chapter provides an overview of the research project or system development, setting the context for the entire document. It outlines the problem statement, objectives, and research questions, highlighting the significance and relevance of the study. The chapter introduces the scope and limitations of the project and presents an outline of the subsequent chapters.

* **Chapter 2: Literature Review**

In this chapter, a comprehensive review of existing literature and related works is presented. It involves an in-depth analysis of scholarly articles, research papers, and relevant sources to identify gaps, trends, and insights related to the topic. The literature review provides a theoretical framework and supports the research methodology adopted in the study.

* **Chapter 3: System Architecture**

This chapter delves into the technical aspects of the system architecture. It outlines the overall design, components, and infrastructure of the developed application. The chapter discusses the selection of technologies, frameworks, and tools used in the development process.

* **Chapter 4: System Implementation & Results**

This chapter focuses on the practical implementation of the system. It details the development process, including coding practices, algorithms, and data structures utilized. The chapter presents the results of the system implementation, such as performance metrics, accuracy rates, and validation tests.

* **Chapter 5: Run the Application**

This chapter provides instructions on how to run and utilize the developed application. It guides users through the installation process, system setup, and configuration steps. The chapter explains the application's user interface, features, and functionalities.

* **Chapter 6: Conclusion & Future Work**

The final chapter summarizes the key findings, contributions, and implications of the research or system development. It reflects on the objectives stated in the introduction and addresses the research questions. The chapter discusses the significance of the results and their potential impact on the field. Additionally, it suggests areas for future research, enhancements, or improvements to extend the system's capabilities and address any limitations identified during the project.

**These chapter descriptions provide a general understanding of the content covered in each chapter.**

**Chapter Two:**

**Literature Review**

* 1. **Introduction:**

The goal of this project is to build an automated number Plate Recognition (ANPR) system. They are cutting-edge technologies designed to automatically identify and read vehicle license plates. These systems utilize advanced optical character recognition (OCR) technology, coupled with high-resolution cameras and powerful processing algorithms, to capture, analyze, and store license plate data in real-time.

* 1. **Technology Background:**

Wrong plate recognition is a significant problem in Egypt, with a high rate of false positives and negatives. This can lead to several issues as Inaccurate traffic enforcement, Increased crime and Reduced public safety . The aim goal is to provide a reliable, accurate, and efficient means of automatically identifying vehicle license plates. This involves detecting and recognizing plates in real-time, ensuring high accuracy, and integrating seamlessly with other systems and infrastructure.

Computer vision techniques are crucial for the initial stages of image processing, detection, and segmentation in ANPR systems. They help in capturing and preparing the image for further analysis. Machine learning, particularly deep learning, significantly enhances the accuracy and efficiency of license plate detection and character recognition. By integrating these technologies, ANPR systems can achieve high accuracy, real-time processing, and adaptability to different environments and conditions, making them robust and reliable for various applications.

* 1. **Related Works:**

In 2022, Ahmed Ramadan Youssef, Abdelmagid Ameen Ali, Fawziya Ramadan Sayed [[1]](https://thesai.org/Downloads/Volume13No7/Paper_99-Real_time_Egyptian_License_Plate_Detection_and_Recognition.pdf)  Developed A new real-time system for Egyptian license plate detection and recognition was developed using the Tiny-YOLOv3 object detection deep learning model. The system operates in two stages: the first stage detects the license plate location from the input vehicle image, and the second stage recognizes the characters and digits from the detected license plate. The EALPR dataset was utilized, comprising 2,450 vehicle images and over 12,000 Arabic digits and characters. The system's performance metrics are as follows: Plate detection accuracy: 78%, Character segmentation accuracy: 85%, Character recognition accuracy: 74%Overall, the recognition accuracy of the system is 93.53%

In 2021, Nguyen Huu and Cuong Vu Quoc  [[2]](https://pdfs.semanticscholar.org/514a/494ccd27bc059ecae8c47abc250be2f0c6cd.pdf)  utilized a deep learning model. The model was (WPOD-NET) is a pre-trained model are typically trained on datasets that contain many images with annotated planar objects, particularly license plates, from various angles, lighting conditions, and environments. And using a modified support vector machine (SVM) system is proposed. The proposal improves the range of detection angle and the accuracy of detecting in shady conditions. The results show that the accuracy of proposal model is up to 95.1% with 1000 testing images in various scenarios.

In 2020, Sriashika Addala [[3]](https://www.researchgate.net/publication/344668186_Research_paper_on_vehicle_detection_and_recognition)developed a vehicle detection and recognition application using smart sensors like cameras to capture live data, processed in the cloud for analysis. Their optimized model achieved 97% accuracy in vehicle detection. Integrated Traffic Management Systems (ITMS) further synchronized data to enhance traffic monitoring and emergency response. The application also focused on License Plate Recognition (LPR), leveraging contrasting backgrounds for improved plate region identification, followed by segmentation and character recognition using binarization and pixel-by-pixel comparison for precise identification.  The calculated average accuracy of number plate recognition using their model is 94.34% which, their model is considered the best for ANPR or LPR which is Automatic Number Plate Recognition

Amany M. Ahmed , Sherin F. Aly [[4]](https://www.researchgate.net/profile/Sherin-Aly/publication/338645726_Egyptian_License_Plates_Recognition_System_Using_Morphologial_Operations_and_Multi_Layered_Perceptron/links/5e21805aa6fdcc10156fed3d/Egyptian-License-Plates-Recognition-System-Using-Morphologial-Operations-and-Multi-Layered-Perceptron.pdf) applied an automatic license plate recognition (ALPR) system for the new Egyptian license plates. The system presents an algorithm for the extraction, segmentation, and recognition of the car license plate. Segmentation step is performed using image processing techniques such as edge detection and morphological operation. Recognition is done using multi layered perceptron (MLP). The performance of the system has been investigated on real images of about 100 vehicles captured under different angles, distances, scales, and illumination conditions. The system achieves a recognition rate of about 78%.

[Prem Chandra Roy,](https://www.researchgate.net/profile/Prem-Roy?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19) [Arjun Bhandari Thapa,](https://www.researchgate.net/scientific-contributions/Arjun-Bhandari-Thapa-2159980557?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19) [Kumar Shrestha](https://www.researchgate.net/profile/Kumar-Shrestha?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19) [[5]](https://www.researchgate.net/publication/341282748_Vehicle_Number_Plate_Recognition_and_Parking_System) A Vehicle Number Plate Recognition and Parking system has been developed for use in traffic systems, parking areas, and border crossings in Nepal. This integrated automatic system combines hardware and software to efficiently manage vehicle entry. Utilizing digital image processing, the system captures a photo of the car's license plate at the entrance using a Pi Camera or Webcam. The image is stored in JPEG format and converted to grayscale. During pre-processing, noise is removed using grayscale conversion and median filtering. The system then extracts the number plate from the image through segmentation methods, typically involving binarization. In the character segmentation stage, the system separates and identifies each character on the plate, matches them against a database, and converts the recognized characters into a string. The resulting license plate number determines if the vehicle can enter the parking lot.

M.A. Massoud \*, M. Sabee, M. Gergais, R. Bakhit [[6]](https://www.sciencedirect.com/science/article/pii/S1110016813000276#s0010) the Egyptian Vehicle License Plate Recognition (LPR) system comprises three main components: Image Capture, which involves isolating features; Pre-processing, including Gray Scale Conversion and Noise Removal. Egyptian license plates (17 cm (about 6.69 in) × 32 cm) are segmented into sections: top for country names and vehicle type, sides for characters and digits. Grayscale and original images are split 1:2; the first part identifies vehicle type using a color filter, and the second undergoes median filtering and noise removal for clarity and object removal, ensuring accurate extraction for character recognition and vehicle classification. A database using Microsoft Access includes car type, city detection, and fault cost. Characters and numbers are segmented into fixed-size blocks and matched with database entries. UDP facilitates data transmission across servers, supporting real-time video stream recognition with 91% accuracy under varied conditions, noting challenges with motion blur and plate obstruction. This system, utilizing GUI and Matlab, aligns with traffic administration rules for practical application.

* 1. **Conclusion:**

The Automated Number Plate Recognition (ANPR) system is a significant improvement for traffic management and law enforcement in Egypt. By automating the identification and reading of vehicle license plates, the system reduces human error, ensures accurate fines, and correctly identifies violations. This leads to more efficient and fair traffic law enforcement, improved public safety, and better resource management.

ANPR systems also provide real-time alerts and detailed records, helping track stolen vehicles, monitor unauthorized use, and manage access control. Their versatility extends to toll collection, parking management, and border control, making them valuable for various applications.

In summary, ANPR systems enhance efficiency, accuracy, and security in traffic management, promoting safer roads and responsible driving behavior.

**Chapter Three:**

**System Architecture**

* 1. **System Architecture:**

A screenshot of a diagram

Description automatically generated

Figure 2

This diagram represents a multi-layered system architecture designed for vehicle and license plate recognition. The architecture is divided into three main layers: Presentation Layer, Application Layer, and Dataset Layer. Here is a detailed explanation for documentation purposes:

**Layers :**

1. **Presentation Layer**

* **Purpose:** Acts as the interface between the external systems and the application. This layer handles interactions and inputs from external systems.
* **Components:**
  + External System: Represents any system or user interface that interacts with the application, such as a user interface for uploading images or a third-party system sending vehicle images for processing.

1. **Application Layer**

* **Purpose**: Contains the core logic and processing capabilities of the system. This layer is responsible for the entire workflow from image pre-processing to data validation.
* **Components**:
  + **Image Pre-Processing**: The initial stage where raw images are enhanced and prepared for further analysis. This could involve noise reduction, resizing, and normalization.
  + **Vehicles Detection**: Identifies and locates vehicles within the processed images. This step isolates the region of interest (ROI) that contains the vehicle.
  + **Plate Localization**: Detects and extracts the license plate area from the identified vehicle regions. This involves locating the exact position of the license plate within the vehicle's image.
  + **Character Segmentation**: Segments the characters on the license plate for individual recognition. This step breaks down the license plate image into separate characters.
  + **Character Recognition**: Recognizes and converts the segmented characters into readable text. This often involves optical character recognition (OCR) techniques.
  + **Data Validation**: Ensures that the recognized data is accurate and valid. This step may involve cross-referencing with known data or applying certain rules to verify the information.

1. **Dataset Layer**

* **Purpose**: Manages the storage and retrieval of data. This layer handles both the input images and the associated vehicle owner information.
* **Components**:
  + **Dataset of Image Vehicles**: A database or storage system containing the collected images of vehicles. This dataset is used for processing and reference.
  + **Data of Vehicles' Owner**: A database that stores the recognized and validated data related to vehicle owners. This might include information such as owner details, vehicle registration information, and more.

1. **Interaction Flow**
2. An external system (user interface or third-party system) provides an image to the Presentation Layer.
3. The image is passed to the Application Layer where it undergoes a series of processing steps:
   * Image Pre-Processing
   * Vehicles Detection
   * Plate Localization
   * Character Segmentation
   * Character Recognition
   * Data Validation
4. The processed data and recognized information are then stored and managed in the Dataset Layer.
5. The final validated data can be sent back to the external system as required.

This architecture ensures a structured and systematic approach to vehicle and license plate recognition, facilitating clear separation of concerns and scalable system design.

* 1. **Description of methods and procedures used**

### 3.2.1 Image Pre-Processing Steps for Vehicle and License Plate Recognition

#### **1. Data Rotation**

* **Description**: Rotation of images within a specified range to simulate different angles and orientations of the input images.
* **Range**: Between -15° and +15°
* **Purpose**: Enhances the model’s robustness to variations in the angle of the vehicle or license plate, ensuring that the system can accurately detect and recognize plates regardless of their orientation.

#### **2. Brightness Adjustment**

* **Description**: Adjusting the brightness of images within a specified range to simulate different lighting conditions.
* **Range**: Between -22% and +22%
* **Purpose**: Improves the model's ability to handle variations in lighting conditions, such as overexposed or underexposed images, making the recognition process more reliable in diverse environments.

#### **3. Blur Application**

* **Description**: Adding a slight blur to the images to mimic the effect of motion blur or out-of-focus images.
* **Amount**: Up to 1.25 pixels
* **Purpose**: Helps the model learn to handle and correctly interpret images that may be blurred due to movement or camera focus issues.

#### **4. Noise Addition**

* **Description**: Introducing random noise to the images, affecting a certain percentage of the pixels.
* **Amount**: Up to 5% of pixels
* **Purpose**: Increases the model's resilience to noisy images, ensuring accurate recognition even when images are degraded by noise.

#### **5. Conversion to Array**

* **Description**: Converting each image into a numerical array representation.
* **Purpose**: Transforms the image into a format suitable for processing by machine learning models, where each pixel value is represented numerically.

#### **6. Resizing**

* **Description**: Resizing the image to a standard size.
* **Target Size**: 224 x 224 pixels
* **Purpose**: Ensures uniformity in input image dimensions, making them suitable for input into transfer learning models that require a fixed input size.

#### **7. Normalization**

* **Description**: Scaling pixel values to a standard range, typically between 0 and 1 or -1 and 1.
* **Purpose**: Helps in faster convergence during model training by normalizing the input data. This process adjusts the pixel values to a consistent scale, improving the model's performance and stability.

### 3.2.2 Car Detection for Vehicle and License Plate Recognition

In the car detection stage, we employed two distinct models to ensure high accuracy and robustness in identifying vehicles within the input images. Here is a detailed explanation of the models used and their performance:

#### **Models Used for Car Detection:**

Reference

[(PDF) Research paper on vehicle detection and recognition (researchgate.net)](https://www.researchgate.net/publication/344668186_Research_paper_on_vehicle_detection_and_recognition?enrichId=rgreq-b231eb981863d928d58d0b89a92ccfb8-XXX&enrichSource=Y292ZXJQYWdlOzM0NDY2ODE4NjtBUzo5NDY4NDY3MDA2OTE0NThAMTYwMjc1NzI1MzI4MQ%3D%3D&el=1_x_3&_esc=publicationCoverPdf)

#### **1. YOLOv8 (You Only Look Once version 8)**

* **Description**: YOLOv8 is a state-of-the-art object detection model known for its speed and accuracy. It processes images in a single pass (hence "You Only Look Once"), making it highly efficient for real-time applications.
* **Pre-trained Model**: We utilized a pre-trained YOLOv8 model, which means it was already trained on a large dataset of images to recognize various objects, including vehicles. Using a pre-trained model allows us to leverage existing knowledge and adapt it to our specific needs with minimal additional training.

#### **2. SSD (Single Shot MultiBox Detector)**

* **Description**: SSD is another powerful object detection model that detects objects in images in a single shot, similar to YOLO. It divides the image into a grid and predicts bounding boxes and class scores for each grid cell.
* **Training**: The SSD model was trained and evaluated over multiple epochs to enhance its detection accuracy. Below is the performance analysis over different epochs:
* Performance of SSD Model

A collage of cars

Description automatically generated

Figure 3

|  |  |
| --- | --- |
| * Epoch | * Accuracy |
| * 10 | * 87.31% |
| * 20 | * 87.81% |
| * 30 | * 87.91% |
| * 40 | * 88.31% |
| * 50 | * 88.99% |
| * 60 | * 88.99% |
| * 70 | * 90.2% |
| * 80 | * 90.15% |

### 3.2.3 Plate Localization for Vehicle and License Plate Recognition

In the plate localization stage, two different models were utilized to achieve high accuracy in detecting and isolating license plates from the detected vehicles. Below is a detailed explanation of the models used and their performance metrics:

#### **Models Used for Plate Localization:**

#### **1. YOLOv8 (You Only Look Once version 8)**

* **Description**: YOLOv8 is a highly efficient object detection model known for its speed and accuracy. It processes images in a single pass, making it ideal for real-time applications.
* **Training and Performance**: The YOLOv8 model was trained and evaluated over multiple epochs, showing remarkable accuracy. Below is the performance analysis over different epochs:

|  |  |
| --- | --- |
| * Epoch | * Accuracy |
| * 10 | * 98.68% |
| * 20 | * 98.98% |
| * 30 | * 98.85% |
| * 40 | * 98.93% |
| * 50 | * 98.99% |
| * 60 | * 98.92% |
| * 70 | * 99.0% |
| * 80 | * 99.01% |

#### **2. Inception ResNet**

* **Description**: Inception ResNet combines the architecture of Inception networks with residual connections, which helps in achieving high accuracy with efficient computation.
* **Training and Performance**: The Inception ResNet model was also trained and evaluated over multiple epochs. Below is the performance analysis over different epochs:

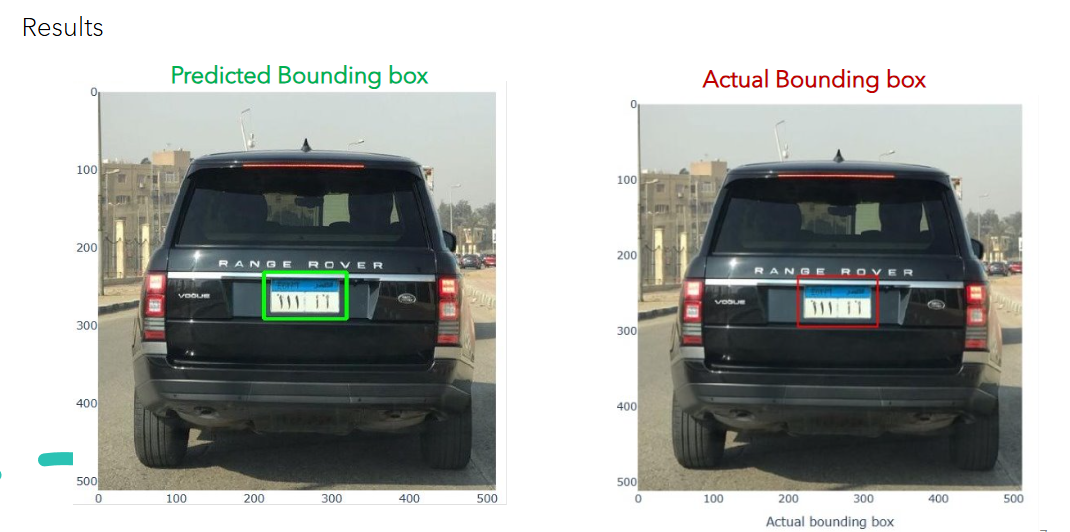


Figure 4

|  |  |
| --- | --- |
| * Epoch | * Accuracy |
| * 10 | * 79.12% |
| * 20 | * 85.16% |
| * 30 | * 95.05% |
| * 40 | * 96.7% |
| * 50 | * 96.7% |

A screenshot of a screenshot of a truck

Description automatically generated

Figure 5

* For plate localization, the YOLOv8 model is highly recommended due to its superior accuracy and efficiency. Its ability to achieve over 99% accuracy makes it the optimal choice for accurately identifying and isolating license plates from vehicle images in real-time applications.
* Reference
* Egyptian License Plates Recognition System Using Morphologial Operations and Multi-Layered Perceptron
* [Egyptian-License-Plates-Recognition-System-Using-Morphologial-Operations-and-Multi-Layered-Perceptron.pdf (researchgate.net)](https://www.researchgate.net/profile/Sherin-Aly/publication/338645726_Egyptian_License_Plates_Recognition_System_Using_Morphologial_Operations_and_Multi_Layered_Perceptron/links/5e21805aa6fdcc10156fed3d/Egyptian-License-Plates-Recognition-System-Using-Morphologial-Operations-and-Multi-Layered-Perceptron.pdf)​
* Vehicle Number Plate Recognition and Parking System December 2018
* [(PDF) Vehicle Number Plate Recognition and Parking System (researchgate.net)](https://www.researchgate.net/publication/341282748_Vehicle_Number_Plate_Recognition_and_Parking_System)​

**Chapter Four:**

**System Implementation & Results**

* 1. **Description of Materials (Datasets):**

We use 2 different datasets one of them is images and the second is row data:

4.1.1  Images

In developing our Automatic Number-Plate Recognition (ANPR) system, the quality and diversity of our datasets are crucial for training robust models capable of accurate vehicle detection, license plate localization, and character recognition.

A collage of cars

Description automatically generated

Figure 6

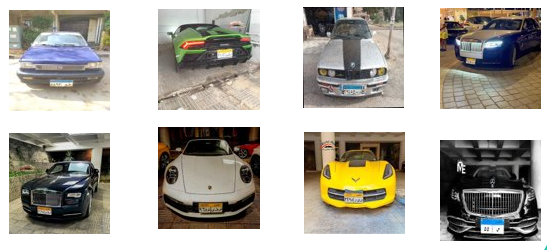


Figure 7

**Characteristics:**

* Vehicle Diversity: Our datasets include images of various vehicle types, such as cars and trucks, ensuring our model can recognize different vehicle shapes and sizes.
* Environmental Conditions: Images cover diverse environmental conditions, including different times of day, weather conditions (e.g., sunny, rainy), and lighting scenarios (e.g., day and night), enabling our model to perform well under real-world settings.
* License Plate Variability: We incorporate images with a wide range of license plate designs, fonts, and colors to train the model to accurately detect and read plates from Egypt.

1. Row Data

**Consistency Check**

* Verify that attributes such as MAKE, PROD\_YEAR, and CCM\_TON are uniform for each DRIVER\_ID across all insurance records.
* Ensure PROD\_YEAR logically precedes or matches the vehicle's insurance inception date (INSR\_BEGIN).
* Address any discrepancies found to maintain the integrity of historical vehicle data.

**Driver-Specific Vehicle Validation:**

* Validate that attributes such as PROD\_YEAR, and CCM\_TON consistently correspond to the specific DRIVER\_ID across all insurance records.
* Confirm that the vehicle details (MAKE, PROD\_YEAR) match consistently over time for each DRIVER\_ID, ensuring continuity in ownership and vehicle characteristics.

A screenshot of a computer

Description automatically generated

Figure 8

1. **Preprocessing**

**Dataset 1: Vehicle Images**

The image dataset consists of visual representations of vehicles, such as photographs or scans. Preprocessing steps include:

* **Data Cleaning:** Removing corrupted or irrelevant images to ensure data quality.
* **Image Processing:** Resizing images to a standard resolution, adjusting color channels and enhancing clarity if necessary.
* **Normalization:** Scaling pixel values to a standard range (0, 1) for consistency in analysis.

**Dataset 2: Tabular Vehicle Data**

The tabular dataset provides structured information about vehicles, including attributes such as driver id, type of vehicle ,usages , and insurance details. Preprocessing steps include:

* **Data Cleaning:** Handling missing values, correcting data formats ,and addressing outliers in numerical columns (insured value, premium).
* **Feature Engineering:** Deriving additional features such as calculating derived metrics like average insurance premiums per vehicle model.

1. **Care Detection**

**4.3.1** Vehicle Detection using YOLO

**Input:**

* **Input Image:** The image captured by the camera, containing one or more vehicles within the field of view.

**Processing:**

* **YOLO Framework:** YOLO takes the input image and performs convolutional neural network (CNN) operations to detect and localize vehicles.
* **Object Detection:** YOLO processes the entire image in one pass to output bounding boxes around detected vehicles along with confidence scores.

**Output:**

* **Detected Objects:** Output includes bounding box coordinates (x, y, width, height) and confidence scores for each detected vehicle in the image.

**4.3.2** Character Segmentation using YOLO

**Input:**

* **License Plate Region:** After vehicle detection, the region of interest (ROI) containing the license plate is extracted from the original image based on the bounding box coordinates provided by YOLO.

**Processing:**

* **Adapted YOLO Model:** YOLO is adapted or fine-tuned to focus on the license plate region for character segmentation.
* **Segmentation Techniques:** Within the localized license plate region, segmentation techniques such as contour detection or modified YOLO output are applied to identify individual characters.

**Output:**

* **Segmented Characters:** Output consists of bounding boxes or segmented regions around individual characters within the license plate image.

**4.3.3** Character Recognition using OCR

**Input:**

* **Segmented Characters:** The extracted characters or segmented regions from the license plate image serve as input to the OCR system.

**Processing:**

* **OCR Model:** Optical Character Recognition (OCR) models, often based on deep learning architectures like CNNs , process the segmented character images.
* **Character Classification:** OCR algorithms classify each character image to recognize the alphanumeric symbol it represents.

**Output:**

* **Recognized License Plate Number:** The final output is the recognized sequence of characters (license plate number) extracted from the segmented regions, verified against a database if necessary.

**4.3.4** Environmental Noise Reduction

**Input:**

* **Noisy Image:** Raw image captured by the camera, possibly affected by external factors such as rain, dust, shadows, or poor lighting conditions.

**Processing:**

* **Noise Filtering:** Apply techniques such as median filtering  to reduce noise and smooth the image.
* **Contrast Enhancement:** Techniques like histogram equalization to improve contrast and enhance visibility of details.
* **Image Normalization:** Adjust the overall brightness and color balance to standardize the image quality.

**Output:**

* **Enhanced Image:** Processed image with reduced noise and enhanced contrast, ready for character segmentation and OCR.

**4.3.5** Data Verification

**Input:**

* **Recognized License Plate Number:** Output from the OCR system, which is the recognized sequence of characters from the license plate.

**Processing:**

* **Database Integration:** Compare the recognized plate number against a stored database of registered vehicles.
* **Validation Process:** Verify the accuracy and validity of the recognized plate number by checking against database entries.
* **Error Handling:** Implement algorithms to handle discrepancies or errors in recognition to ensure robust verification.

**Output:**

* **Validation Result:** Confirmation of whether the recognized plate number matches an entry in the database, with additional metadata (such as : vehicle owner information, insurance date).

**4.3.6** Plate History Lookup

**Input:**

* **Recognized License Plate Number:** Same as in Data Verification, the output from the OCR system.

**Processing:**

* **Database Query:** Retrieve historical records associated with the recognized license plate number from relevant databases.
* **Data Integration:** Integrate retrieved information such as vehicle ownership id and insurance history.

**Output:**

* **Historical Information:** Provide details retrieved from the database related to the recognized license plate number, enabling comprehensive vehicle history lookup.

* 1. **Description of Software Tools used \**

1. **Google Collaboratory:**

Google Colab is like jupyter notebooks that allows you to write and execute Python code without any setup and gives us free GPU.

We tried to use Google Colab to execute our model, but it keeps crashing all the time due to our large dataset and types of layers used in model’s architectures like Conv3d and LSTMs which needs some additional resources to train such models.

1. **Kaggle**

Kaggle is an online platform that hosts machine learning competitions, provides datasets for practice, and offers a collaborative environment for data scientists and machine learning practitioners. It is widely used by the data science community to develop and showcase their skills, learn from others, and participate in competitions.

Kernels/Notebooks: Kaggle offers a feature called "Kernels" (previously known as notebooks) which allows users to create and share code, analysis, and visualizations. This is a convenient way to develop and run deep learning models using popular frameworks such as TensorFlow, PyTorch, or Keras. Users can also explore and learn from the kernels shared by others.

GPU and TPU Support: Kaggle provides access to powerful hardware accelerators like GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units). Deep learning models can greatly benefit from the parallel computing capabilities of these devices, enabling faster training and inference times.

Overall, Kaggle provides a comprehensive platform for running deep learning models, leveraging its datasets, kernels, GPU/TPU support, competitions, and a vibrant community.

1. **Python**

Python is a versatile programming language widely used in machine learning, web development, and scientific computing. It's ideal for implementing image processing algorithms (using OpenCV), training OCR models.

1. **GitHub:**

For maintaining source code and working together on software development projects, many people utilize the distributed version control system known as Git. Depending on the version being utilized, the particular version number and the year of manufacture may change. What Motivated Us to Use It:

* **Code Management:** Git provides detailed change tracking, recording all modifications, authors, and timestamps. This feature facilitates code review, accountability, and transparency. Git's integration with collaboration platforms enhances code visibility and enables efficient issue tracking and project management. Additionally, Git simplifies deployment and release management by tagging specific commits, making it easier to track and deploy sentiment analysis applications in different environments.
* **Version Control:** Effective version control is made possible by Git by keeping track of changes made to the codebase over time and promoting teamwork. It enables concurrent work on the project and easy collaboration by making branching, merging, and dispute resolution possible. Git's distributed architecture and built-in backup features guarantee code preservation and simple recovery, reducing the likelihood of data loss.

 Chapter 5:

References

* [[1]Automated new license plate recognition in Egypt – ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1110016813000276)
* [[2](PDF) Research paper on vehicle detection and recognition (researchgate.net)](https://www.researchgate.net/publication/344668186_Research_paper_on_vehicle_detection_and_recognition?enrichId=rgreq-b231eb981863d928d58d0b89a92ccfb8-XXX&enrichSource=Y292ZXJQYWdlOzM0NDY2ODE4NjtBUzo5NDY4NDY3MDA2OTE0NThAMTYwMjc1NzI1MzI4MQ%3D%3D&el=1_x_3&_esc=publicationCoverPdf)
* [[3]494ccd27bc059ecae8c47abc250be2f0c6cd.pdf (semanticscholar.org)](https://pdfs.semanticscholar.org/514a/494ccd27bc059ecae8c47abc250be2f0c6cd.pdf)
* [[4]thesai.org/Downloads/Volume13No7/Paper\_99-Real\_time\_Egyptian\_License\_Plate\_Detection\_and\_Recognition.pdf](https://thesai.org/Downloads/Volume13No7/Paper_99-Real_time_Egyptian_License_Plate_Detection_and_Recognition.pdf)
* [[5]Egyptian-License-Plates-Recognition-System-Using-Morphologial-Operations-and-Multi-Layered-Perceptron.pdf (researchgate.net)](https://www.researchgate.net/profile/Sherin-Aly/publication/338645726_Egyptian_License_Plates_Recognition_System_Using_Morphologial_Operations_and_Multi_Layered_Perceptron/links/5e21805aa6fdcc10156fed3d/Egyptian-License-Plates-Recognition-System-Using-Morphologial-Operations-and-Multi-Layered-Perceptron.pdf)
* [[6](PDF) Vehicle Number Plate Recognition and Parking System (researchgate.net)](https://www.researchgate.net/publication/341282748_Vehicle_Number_Plate_Recognition_and_Parking_System)