A PBL-3 Project Report on

Automatic Number Plate Recognition

Submitted to Manipal University Jaipur

Towards the partial fulfillment for the Award of the Degree of

BACHELORS OF TECHNOLOGY

In Computers Science and Engineering (AIML)

2024-2025

By

Anika Mathur

229310128

MANIPAL UNIVERSITY JAIPUR

INSPIRED BY LIFE

Under the guidance of

Ms. Jyoti Rani

____

Signature of Supervisor (12pt.)

Department of Artificial Intelligence and Machine Learning

School of Computer Science and Engineering

Manipal University Jaipur

Jaipur, Rajasthan

Table of Contents

Introduction

Automatic Number Plate Recognition (ANPR) Systems are slowly becoming a necessity for domains such as transport management, traffic surveillance and law enforcement. The foremost purpose of these systems is to identify and extract vehicle license plate numbers from images or video streams of data, enabling tasks such as toll collection, parking management, crime prevention etc.

What is this Research about?

Now that the adoption of ANPR Systems is on the rise, the question rises, "How do we make the best possible ANPR system?"

We need to research and understand what we can do to make this system as precise and accurate as we possibly can. We need to make sure it can adapt to different environments, different image qualities and varying lighting conditions.

In this project, different approaches to ANPR are examined from a research perspective, aiming at finding and combining modern methodologies to achieve optimal performance. The defining feature of this work relates to the combination of classical computer vision approaches such as Edge and contour Detection and recent deep learning architectures such as YOLO and R-CNNs, and their in-depth examination for the evaluation of strengths and limitations.

Why is this Research Important?

This research is important to understand and overcome the constraints of the existing ANPR systems and make them as efficient and accurate as possible, particularly in a country like India where automated recognition is rendered very difficult due to great differences in the design, fonts, and colour of the number plates as well as environmental factors such as lighting conditions.

Here's why this research is useful:

<u>Integrating the Conventional and AI Models</u>

Lightweight and simple traditional methods, such as edge and contour detection, are effective in structured environments but have difficulty functioning in unstructured and noisy conditions. On the other hand, AI models such as YOLO are very good at the detection of objects in various conditions, but need a lot of computing power and annotated data.

After exploring and understanding traditional techniques such as Edge and Contour Detection and newer methods such as YOLO and R-CNN's we attempt to build new models of our own using Model fusion to integrate the strengths of all models.

<u>Solving Practical Problems</u>

Many Indian number plates are different in dimensions, shape, and placement/orientation. Other factors include blurriness, poor visibility of illumination, and obstruction by dirt, stickers, and so on. This research intends to construct a system that can be operational even in adverse situations which are common in developing countries by merging the edge-based methods with the AI models and fusion of models to strengthen the systems performance.

How will this be achieved?

The project is structured as follows:

Edge Detection Module: Focusing on the license plate region extraction from images, this module employs the Canny Edge Detection and contour generation algorithms through clustering technique.

YOLO Integration: Utilising the YOLO model of object detection,it will create bounding boxes around the number plates and then an OCR system like Tesseract will be used to accurately and rapidly detect plates.

Model Fusion with RNNs: Engaging neural networks in the task of fusing conventional and artificial outputs while, at the same time, providing an overlapping contextual framework to improve accuracy.

Experimental Validation: The models are trained and validated on the Indian Number Plates dataset using precision, recall and inference speed metrics for evaluation.

Objective

- **Comparative Analysis of Different Approaches:**

To analyze the abilities of old computer vision methods such as Canny Edge Detection for plate number detection against modern techniques such as YOLO. This aims at an analysis on how these methods have both pros and cons for applications across different scenarios of image qualities, lighting conditions, and plate designs.

- **Construct a Feasible, Accurate, and Precise ANPR Model:**
  Design and develop an ANPR hybrid system that will integrate traditional techniques such as Edge and Contour Detection with newer deep learning models, such as the YOLO approach and R-CNN's for the development of a model that is efficient in real-time performance with proper adaptation to diverse conditions but is highly accurate and precise enough in both ideal and challenging settings.

- **Optimize the model for real-time performance and scalability:**
  We need to optimize ANPR in such a manner that real-time detection is ensured in the most scalable hardware and different environments, especially such applications for which this system should be highly apt in applications of traffic surveillance, law enforcement, and collection of toll.

Literature Survey

Paper Title	Authors	Technology Used	Strengths	Limitations	Application/ Context	Citations
Contour Based Number Plate Recognition System Using EM Segmentation Method	V. R. Viju, Dr. R. Radha	Canny Edge Detection, Clustering, EM Segmentation	Effective in detecting contours for license plate extraction	Sensitive to noise, dependent on high-quality images.	Used in controlled environments where plate contours are clearly visible. The number plate is assumed to be the greatest contour.	Viju, V. R., & Radha, R. (n.d.). *Contour Based Number Plate Recognition System Using EM Segmentation Method*.
Image Binarization using Otsu Thresholding Algorithm	Jamileh Yousefi	Otsu Thresholding for Image Binarization	Simple, computationally efficient	Not ideal for images with uneven lighting or shadows	To understand how binarization and Otsu's Thresholding Algorithm works to implement the Edge Detection Method	Yousefi, J. (2011). *Image Binarization using Otsu Thresholding Algorithm*.

Vehicle Number Plate Detection using Sobel Edge Detection Technique	Dr. P.K. Suri, Dr. Ekta Walia, Er. Amit Verma	Sobel Edge Detection	Effective for detecting edges in clean images	Struggles in low-contrast or noisy environments	Used for basic vehicle plate detection, especially in simpler cases	Suri, P.K., Walia, E., & Verma, A. (n.d.). *Vehicle Number Plate Detection using Sobel Edge Detection Technique*.
Automatic Number Plate Recognition using YOLO for Indian Conditions	Aayush Gattawar, Sandesh Vanwadi, Jayesh Pawar, Pratik Dhore, Prof. Harshada Mhaske	YOLO (You Only Look Once)	High accuracy and real-time detection, effective in cluttered backgrounds	Requires significant computational resources, less effective with small objects	Applied to detect Indian vehicle number plates in diverse real-world conditions	Gattawar, A., Vanwadi, S., Pawar, J., Dhore, P., & Mhaske, H. (n.d.). *Automatic Number Plate Recognition using YOLO for Indian Conditions*.
Leveraging Model Fusion for Improved License Plate Recognition	Rayson Laroca, Luiz A. Zanlorensi, Valter Estevam, Rodrigo Minetto,	Model Fusion (Combining Traditional & Deep Learning Methods)	Increases accuracy and robustness by combining multiple models	Complex implementation and tuning required	Focused on improving ANPR performance through model fusion for better overall recognition	Laroca, R., Zanlorensi, L.A., Estevam, V., Minetto, R., & Menotti, D. (n.d.). *Leveraging Model Fusion for*

	David Menotti					*Improved License Plate Recognition*.
An Overview of the Tesseract OCR Engine	Ray Smith	Optical Character Recognition (OCR)	Open-source, well-documented, widely used	Struggles with noisy backgrounds, needs preprocessing	Used for optical character recognition after detecting plates	Smith, R. (n.d.). *An Overview of the Tesseract OCR Engine*.
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks	Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun	Faster R-CNN with Region Proposal Networks	High detection accuracy, works well with complex objects	High computational cost, slower than YOLO for real-time applications	Applied in complex object detection, including ANPR in controlled environments	Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*.

Planning of Work

1. Requirements Engineering

As discussed, the aim for the Automatic Number Plate Recognition (ANPR) project is to come up with an efficient as well as accurate system that can reliably detect and recognize license plates from images. The requirements for this project will be divided into functional, non-functional, and technical requirements.

Functional Requirements

1. License Plate Detection: Detect license plates in images under varied conditions using an object detection model (YOLO) to generate bounding boxes or finding regions of interest by generating contours.
2. ROI Extraction: Extract the license plate as a Region of Interest (ROI) for further processing.
3. Image Preprocessing: Convert the extracted ROI to grayscale, apply Gaussian blur or Median filtering for noise reduction, binarize using Otsu's Thresholding, and lastly, dilate the image for OCR.
4. OCR: Use Tesseract OCR to extract alphanumeric characters, as license plates are in multiple formats.
5. Model Fusion: Integrate traditional (Canny edge detection) and AI-based methods (YOLO + OCR) to build an efficient and highly accurate system for ANPR so as to fulfil one of our objectives for this Project.
6. Performance Metrics: Measure precision, recall, F1-score, and inference speed to evaluate detection and recognition effectiveness.

Non-Functional Requirements

1. Accuracy: The system should have a high accuracy in detecting and recognizing license plates with few false positives and negatives.
2. Speed: The system should be efficient with a low latency. Ideally, the detection and OCR should be done in real time or with minimal delay, depending on the deployment scenario. However, in this scenario, we are focusing more on the static images rather than the real time video streams.
3. Scalability: Eventually, it shall be scalable enough to accept multiple input images in parallel or large video streams.
4. Maintainability: The system must have clear documentation and modular design such that it can be easy to maintain so that it can be improved with evolving technology and algorithms.
5. Security: The system should ensure data security and privacy, especially whenever license plate information is stored or transmitted.

Technical Requirements

1. Hardware Requirements:
    a. CPU: High-performance CPU; e.g., Intel i5/i7 or Apple Silicone
    b. Graphics Card: A GPU like NVIDIA GTX/RTX to accelerate deep learning model inference, specifically for YOLO.
    c. RAM: At least 8 GB of RAM to ensure smooth processing.
    d. Storage: Storage for the images and the model weights with sufficient space. An SSD is recommended for faster access

2. Software Requirements:
    a. Programming Language: Python (for model development and data processing).
    b. Deep Learning Frameworks: PyTorch or TensorFlow (for YOLO and model training).
    c. OCR Engine: Tesseract OCR for text recognition.
    d. Image Processing Libraries: OpenCV for image manipulation and preprocessing.
    e. Model Training Framework: PyTorch or TensorFlow for training custom models (YOLO, Canny, etc.).
    f. Operating System: Mac/Linux/Windows

3. Model Requirements:
    a. *Edge Detection Module*
        i. Input Requirements: Grayscale image input with preprocessing capabilities for noise reduction and contrast enhancement.
        ii. Algorithm: Canny Edge Detection with Otsu's Thresholding for adaptive threshold setting.
        iii. Expected Output: Binary edge map with accurately detected license plate contours.
    b. *YOLO Object Detection Module*
        i. Input Requirements: Pre-trained YOLO model weights fine-tuned for license plate detection.
        ii. Model Parameters: Confidence threshold, IoU threshold for Non-Max Suppression.
        iii. Expected Output: Bounding boxes with high confidence for detected license plates.
        iv.

c. OCR with Tesseract

i. Input Requirements: Preprocessed and segmented license plate ROI.

ii. Configuration: Language settings for alphanumeric character recognition.

iii. Expected Output: Recognized license plate characters in text format.

2. <u>Design</u>

The design of the Automatic Number Plate Recognition (ANPR) project is structured towards modularity, efficiency, and integration of traditional approaches with deep learning-based techniques. This section includes designs for system architecture, data flow, and individual modules.

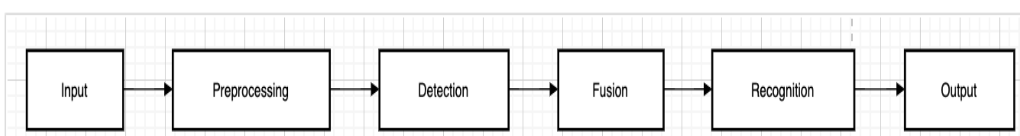
System Architecture

The ANPR system has the following layers:

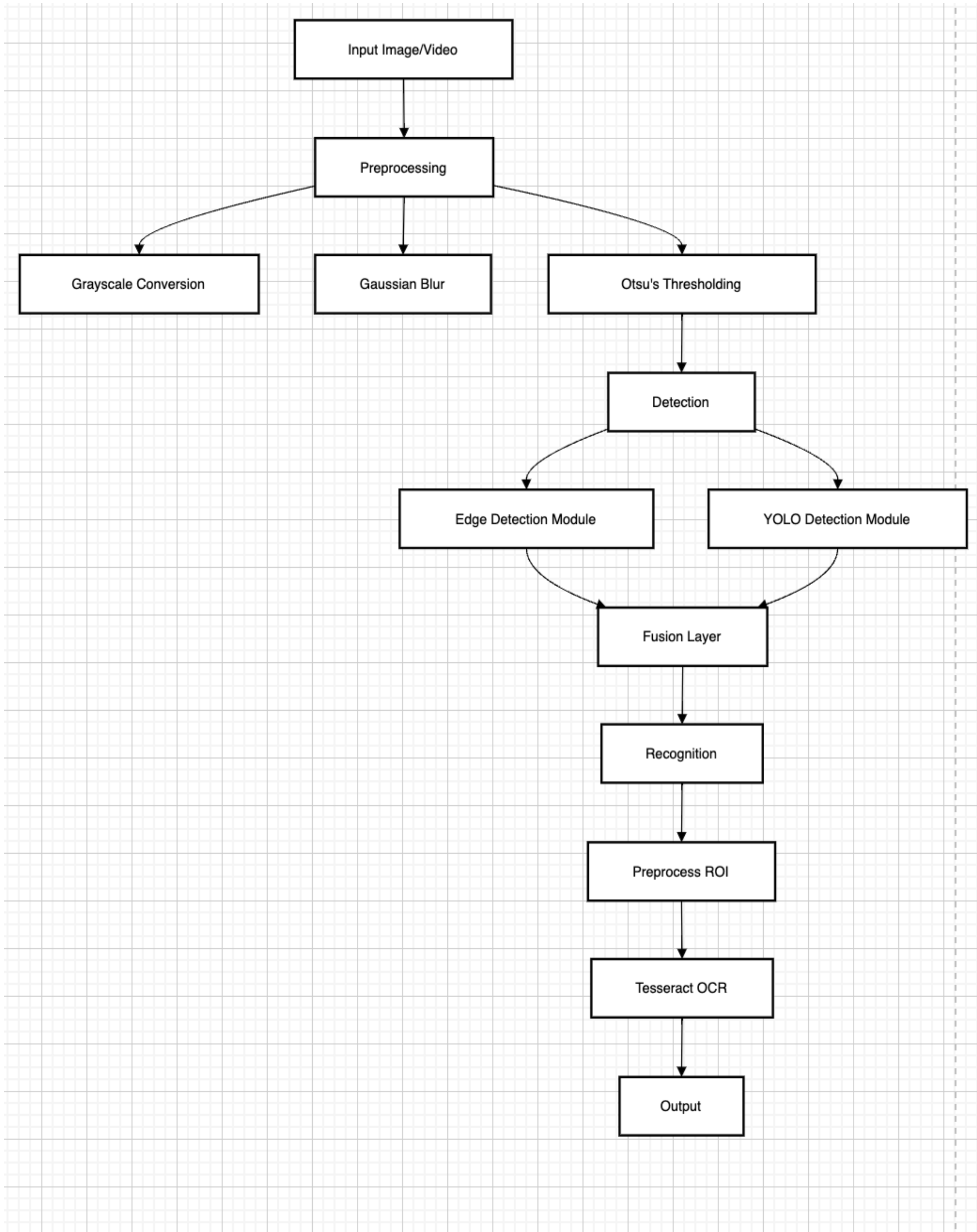
1. Input Layer: accepts image or video frames for processing.
2. Preprocessing Layer: enhances the input data for downstream tasks using grayscale conversion, blurring, and thresholding.
3. Detection Layer: combines Edge Detection and YOLO models to locate license plates.
4. Fusion Layer: uses a Recurrent Neural Network (RNN) for model fusion, refining outputs from the Detection Layer.
5. Recognition Layer: Uses Tesseract OCR to recognize characters from the preprocessed license plate regions.
6. Output Layer: Outputs the extracted license plate text and saves it to a database or file.

Data Flow Diagrams

Level 0:



Level 1



Input Image/Video
Preprocessing
Grayscale Conversion
Gaussian Blur
Otsu's Thresholding
Detection
Edge Detection Module
YOLO Detection Module
Fusion Layer
Recognition
Preprocess ROI
Tesseract OCR
Output

Module Design

Preprocessing Module

1. Purpose: Improve the image quality and isolate key features.
2. Steps:Grayscale conversion to remove colour information and simplify image data.
3. Gaussian Blur: Reduced noise for better edge detection and YOLO predictions.
4. Otsu's Thresholding: Separates foreground (license plate) from the background to segmentation.

Detection Module

1. Edge Detection: Applies Canny Edge Detection to detect edges. Generates contours that may contain potential license plate regions. Filters the contours based on size and aspect ratio constraints.
2. YOLO: Processes the image with a trained YOLO model to detect bounding boxes. Performs Non-Max Suppression (NMS) to refine multiple detections. Extracts high-confidence regions as ROIs for license plates.

Recognition Module

1. Preprocessing the ROI for OCR with:
    a. Grayscale conversion and binarization.
    b. Dilation and Segmentation-to separate characters.
2. It feeds preprocessed data to the Tesseract OCR engine.
3. Extracts alphanumeric text with high accuracy.

3. <u>Proposed Methodology</u>

As discussed above, one of the goals of this project is to design a strong ANPR system by combining the best features of traditional image processing techniques and modern deep learning methods by conducting a comparative analysis to identify the strengths and weaknesses of various methodologies and synthesize a hybrid approach that leverages their individual merits.

In this approach, we emphasize on integrating the simplicity of resource efficiency edge detection techniques together with object detection models from YOLO to enjoy speed and accuracy. This system will be further enhanced using model fusion techniques by the application of other models such as R-CNNs in order to combine outputs, reduce errors, and incorporate contextual learning.

The following sections detail the steps to be taken during the project's development.

Edge and Contour Detection

The Edge Detection Module is one of the core methodologies of this project, using traditional image processing methods to locate and isolate the region of interest, the license plate, in an image. Below are the research methodology steps for this module:

1. Finding a Suitable Dataset

- Collect high-quality images from publicly available datasets, ensure diverse scenarios including varying lighting conditions, angles, and backgrounds.
- Selected Dataset: https://www.kaggle.com/datasets/andrewmvd/car-plate-detection

2. Preprocessing

- <u>Grayscale Conversion:</u> Convert the image to grayscale to simplify processing by reducing it to a single intensity channel.
- <u>Noise Reduction:</u> Apply Gaussian Blur or Median Filtering technique, to reduce noise and smoothen the image, critical for effective edge detection.

3. Canny Algorithm

- Use the thresholds obtained by the Otsu algorithm to separate edges and non-edges:
  - i. Lower Threshold Automatically computed as a fraction of the Otsu-calculated value for edge connectivity.
  - ii. Upper Threshold: Is set using the Otsu value to detect strong edges.

4. Identifying and Generating Contours

- Identify all contours in the edge-detected image using clustering algorithms.
- Pick the largest contour to locate the Region of Interest assuming that the number plate is the largest contour.

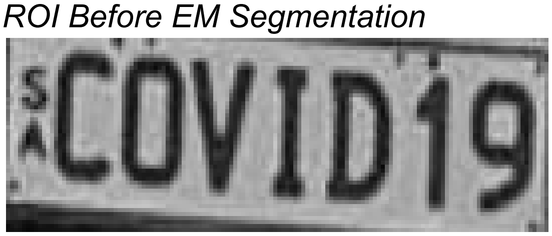
Original Image

Image after Locating Region of Interest

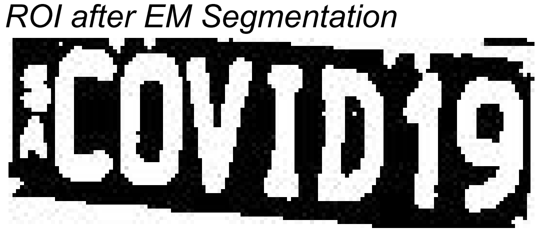
5. EM Segmentation

- Apply the Expectation-Maximization (EM) Algorithm to group pixels and refine the contours.

ROI Before EM Segmentation



ROI after EM Segmentation



6. Optical Character Recognition (OCR)

- Use Tesseract for OCR to extract the text on the number plate.

Extracted text from image2.jpg: 'COVIDI9

7. Challenges

- Low-Contrast Plates: Tuning thresholds dynamically using adaptive methods.
- Complex Backgrounds: Employing morphological operations (e.g., dilation and erosion) to eliminate irrelevant edges.
- The largest contour isn't always the number plate as we have assumed.

YOLO Method

We will employ the YOLO model in this project as a model that detects objects with a real-time setting with its ability to handle very complex backgrounds for plate localization. The dataset will be annotated as bounding boxes is that of the selected dataset in YOLO format.

1. Dataset

- Use the same one as used in the previous one. Make sure the Dataset has annotations to draw bounding boxes around the number plates and convert the images to YOLO format.
- Ensure the dataset is inclusive of challenging conditions like oblique angles, poor lighting, and cluttered backgrounds.

2. YOLO Based Detection

- Use a pre-trained YOLO model version that is fine-tuned for detecting number plates.
- Bounding Box Generation: Input image into the YOLO model, which interprets the image as a single convolutional network to draw bounding boxes surrounding the detected license plates.Bounding boxes are defined by their class, license plate, and confidence score.
- ROI Extraction: Filter bounding boxes based on confidence thresholds to remove false positives and then extract the ROI

Original Image

Image with Bounding Box

3. **Preprocessing ROI**
    - We preprocess the images to enhance clarity for OCR
    - Grayscale Conversion: We convert the image to Grayscale in order to make it easier to process.
    - Gaussian Blur: Apply Gaussian smoothing to reduce noise and ensure clean edges.
    - Binarization: Find an optimal threshold value using Otsu's method to binarize the ROI. Separate the foreground (characters) from the background for clearer segmentation.
    - Dilation: Enhance the structure of characters for higher OCR accuracy by making them bold.
    - Segmentation: Isolate characters using connected components or projection-based segmentation methods.
4. **OCR with Tesseract**
    - Pass the identified ROI to Tesseract for text recognition.
    - Configure Tesseract to focus on alphanumeric characters to match the different license plate formats.
5. **Challenges**
    - False Positives: Fine-tune the thresholds of YOLO on confidence and suppression settings.
    - Plates may be partially obscured by dirt, reflections, or other vehicles, especially in traffic monitoring or surveillance settings.
    - Bounding box collisions happen when there are multiple objects that are detected in the same area, leading to overlap between the boxes. NMS can be applied to remove redundant detections and retain the best one.

Bibliography

1. Viju, V. R., & Radha, R. (n.d.). **Contour-Based Number Plate Recognition System Using EM Segmentation Method**.

2. Yousefi, J. (2011). **Image Binarization Using Otsu Thresholding Algorithm**. *University of Guelph, Ontario, Canada*

3. Suri, P. K., Walia, E., & Verma, A. (n.d.). **Vehicle Number Plate Detection Using Sobel Edge Detection Technique**. *Kurukshetra University, Haryana, India*

4. Gattawar, A., Vanwadi, S., Pawar, J., Dhore, P., & Mhaske, H. (n.d.). **Automatic Number Plate Recognition Using YOLO for Indian Conditions**. *Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India*

5. Laroca, R., Zanlorensi, L. A., Estevam, V., Minetto, R., & Menotti, D. (n.d.). **Leveraging Model Fusion for Improved License Plate Recognition**. *Federal University of Paraná, Brazil*.

6. Smith, R. (n.d.). **An Overview of the Tesseract OCR Engine**.

7. Ren, S., He, K., Girshick, R., & Sun, J. (2015). **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.