YAP470 Project
License Plate Recognition System
Final Report

1st Utku Murat ATASOY
211401019 / Artificial Intelligence Engineering
TOBB University of Economics and Technology
u.atasoy@etu.edu.tr

***Abstract*—This study introduces a License Plate Recognition (LPR) system employing YOLO models for real-time object detection and EasyOCR for text recognition. By integrating a hybrid dataset of over 52,000 labeled images and leveraging advanced preprocessing and augmentation techniques. The system addresses diverse real-world challenges such as occlusion, variable lighting, and pose variations. Comparative evaluations of YOLOv8 and YOLOv11 variants demonstrate superior precision (up to 97.38%) and recall (up to 96.93%) across various metrics, achieving a mean Average Precision (mAP) of 0.985 at IoU 50%. EasyOCR enhances text recognition in low-quality imaging conditions and makes the system adaptable to dynamic environments like traffic monitoring and automated tolling. The findings validate the proposed system's efficacy in achieving high accuracy and computational efficiency, positioning it as a viable solution for modern LPR applications.**

***Index Terms*—License Plate Recognition, Machine Learning, Object Detection, OCR, YOLO**

I. INTRODUCTION

License Plate Recognition (LPR) systems play a crucial role in automated traffic management, offering solutions for efficient vehicle monitoring, toll collection, parking management, and law enforcement. The core challenges of these systems include accurate detection of license plates and precise recognition of their alphanumeric characters, especially under real-world conditions like varying lighting, angles, and occlusions. Over the years, advancements in machine learning have made LPR systems increasingly robust, leveraging state-of-the-art object detection and optical character recognition (OCR) models.

This project addresses these challenges by employing YOLO (You Only Look Once) for real-time object detection and EasyOCR for text recognition. A critical aspect of our approach is the creation of a hybrid dataset that combines three distinct datasets: **24,242 images** from the first source, **6,784 images** from a secondary source, and **21,175 images** from a third dataset, **totaling 52,201 labeled images.** These datasets include diverse and challenging scenarios such as pose variations, occlusions, and lighting inconsistencies. Extensive preprocessing and augmentation techniques, such as rotation, cropping, and grayscale adjustments, were applied to simulate real-world conditions and enhance model robustness.

To ensure comprehensive analysis and comparisons, YOLOv8 and YOLOv11 models were trained on the **first dataset** to address which model performs the best. The result of these trainings resulted as YOLOv11 performs better than YOLOv8. Additionally, YOLOv11 alternatives **(n, s, m, l, x)** were evaluated with consistent parameters for ablation studies. Among these, the YOLOv11x model (largest configuration) achieved a **mean Average Precision (mAP) of 0.98466 at IoU 50 % and 0.71605 at IoU 50–95 %, with precision and recall values of 0.97384 and 0.95932**, respectively, after 20 epochs. In contrast, the lightweight YOLOv11n model achieved similar high precision and recall scores (**0.97734 and 0.97025, respectively) after 100 epochs**, demonstrating a balance between performance and computational efficiency.

Unlike many previous studies that focused on simpler datasets or limited configurations, this project emphasizes **both model performance and scalability through rigorous testing on a hybrid, highly diverse dataset.** By combining cutting-edge object detection with advanced OCR capabilities, our system offers an accurate, real-time solution for license plate recognition, suitable for deployment in dynamic traffic environments.

II. RELATED WORK

The YOLO (You Only Look Once) framework has revolutionized real-time object detection by integrating detection and classification tasks into a single, end-to-end neural network. The original YOLO framework, introduced by Redmon et al. **[5]**, proposed a unified approach to object detection by treating it as a regression problem. This method provided an unprecedented speed of up to 45 frames per second (FPS) while maintaining competitive accuracy. However, the framework struggled with the precise localization of small objects, a challenge addressed in subsequent iterations.

YOLOv2 and YOLO9000 introduced by Redmon and Farhadi **[1]** marked significant advancements, such as multi-scale training and the integration of a hierarchical classification system (WordTree). These enhancements allowed the model to predict over 9,000 object categories, bridging the gap between detection and classification tasks. The introduction of multi-scale training also provided a tradeoff between speed and accuracy, positioning YOLOv2 as a state-of-the-art detector in both speed and performance.

The evolution of YOLO models has been comprehensively reviewed by Terven and Cordova-Esparza **[2]**, who highlighted the innovations introduced across versions from YOLOv1 to YOLOv8, including improvements in anchor-free detection, spatial pyramid pooling, and transformer integration. YOLOv8, for example, achieved remarkable accuracy using a decoupled head design for independent optimization of objectness, classification, and regression tasks, making it a robust solution for diverse object detection scenarios.

In their architectural analysis, Khanam and Hussain **[3]** discussed YOLOv11, which introduced the C3k2 block and spatial attention mechanisms like the C2PSA block. These innovations improved feature extraction and detection accuracy, particularly for small and partially occluded objects, while maintaining computational efficiency. YOLOv11 further expanded YOLO's applicability across tasks such as pose estimation and instance segmentation.

Lavanya and Pande **[4]** emphasized the impact of YOLO's streamlined pipeline in domains requiring real-time detection, such as autonomous vehicles, robotics, and video surveillance. Their study demonstrated how YOLO's high-speed architecture enabled real-time detection without sacrificing accuracy, underscoring its adaptability to various resolutions and input sizes.

These developments showcase YOLO's ability to balance speed, accuracy, and versatility, cementing its position as a cornerstone in real-time object detection. By consistently introducing architectural enhancements, the YOLO series has addressed key challenges in object detection, including scalability, localization precision, and the ability to handle complex, real-world environments.

Optical Character Recognition (OCR) technology plays a crucial role in translating visual text into machine-readable characters and also in our task License Plate Recognition. By enabling automated recognition of text from scanned documents, digital images, or handwritten manuscripts, OCR has been pivotal in numerous applications such as document digitization, text analytics, and accessibility solutions **[6]**. OCR algorithms typically involve a sequence of preprocessing, segmentation, feature extraction, and classification processes to achieve accurate recognition of text from various sources **[8]**. EasyOCR, a modern OCR tool, leverages deep learning techniques to handle a wide variety of text recognition tasks, including printed and handwritten text **[7]**. Its pretrained models and seamless integration with Python frameworks make it a cost-effective and high-performance solution for many applications **[8]**.

License Plate Recognition (LPR) systems have become integral to intelligent transportation systems and security applications, where real-time detection and recognition of license plates are crucial. The field has evolved significantly with advancements in deep learning and image processing techniques. Selmi et al. **[9]** proposed a system employing Convolutional Neural Networks (CNNs) for license plate detection and character recognition. Their approach integrates preprocessing steps such as HSV conversion and morphological filtering to handle complex environments effectively. Zhang et al. **[11]** introduced a cascade classifier combining Haar-like features and AdaBoost learning to detect license plates under diverse conditions with high accuracy. Slimani et al. **[13]** developed a system leveraging wavelet decomposition for license plate localization and CNNs for recognition, achieving high efficiency in both Moroccan and international datasets. Chang et al. **[15]** addressed challenges like non-stationary backgrounds and variable illumination, introducing a fuzzy-based license plate locating module paired with neural networks for robust character recognition. Göde and Doğan **[14]** demonstrated a lightweight, hardware-compatible LPR system utilizing image filtering techniques and artificial intelligence for effective real-world deployment. These studies collectively highlight the advancements in LPR systems, from robust preprocessing methods to deep learning frameworks, catering to various real-world challenges such as low lighting, noise, and complex backgrounds.

III. Dataset

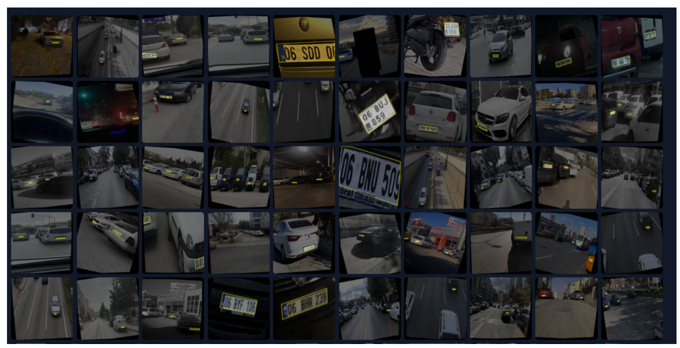


Fig. 1: Example of challenging images in the dataset.

A. Data Source

The dataset for this license plate recognition project combines three distinct datasets to form a **hybrid dataset**. The datasets were obtained from various sources and collectively include **46,278 images for training, 3,957 images for validation, and 1,966 images for testing**, totaling **52,201 labeled images**. These datasets are designed to address real-world challenges such as pose variations, occlusions, and lighting inconsistencies. A detailed breakdown of the datasets is as follows:

- **Dataset 1** (License Plate Recognition Computer Vision Project): 21,175 images, can be found at [16].
- **Dataset 2** (License Plates of Vehicles in Turkey Computer Vision Project): 6,784 images, can be found at [17].
- **Dataset 3** (Vehicle Registration Plates Computer Vision Project): 24,242 images, can be found at [18].

The hybrid dataset, along with its annotations, has been preprocessed and augmented extensively to improve robustness in diverse conditions. For further details, the datasets are accessible at their respective links in the references.

B. Data Division and Usage

The hybrid dataset was split into three parts to maximize model performance:

- **Training Set**: 46,278 images (87%) for model training.
- **Validation Set**: 3,957 images (8%) for model validation.
- **Test Set**: 1,966 images (4%) for model evaluation.

This division ensures that the model is trained, validated, and tested on separate, non-overlapping subsets.

C. Preprocessing and Augmentation

a) Preprocessing:

- **Auto-Orient**: Ensured consistent orientation of images to avoid alignment issues.
- **Resize**: All images were resized to a uniform size of 640x640 pixels.

b) Data Augmentation: To simulate real-world conditions and enhance the model's robustness, the following transformations were applied:

- **Flip**: Horizontal flips to simulate different viewing angles.
- **Crop**: Random cropping with 0% minimum zoom and 15% maximum zoom.
- **Rotation**: Random rotations between -10° and +10° to mimic slight angle variations.
- **Shear**: Horizontal and vertical shears of ± 2 for distortion.
- **Grayscale**: Converted 10% of images to grayscale to account for different lighting and color conditions.
- **Color Adjustments**:
  - **Hue**: Adjusted by -15° to +15°.
  - **Saturation**: Varied by $\pm 15\%$.
  - **Brightness**: Adjusted by $\pm 15\%$.
  - **Exposure**: Adjusted by $\pm 15\%$.
- **Blur**: Applied up to 0.5px to simulate slight motion blur.
- **Cutout**: Added five random cutout boxes, each covering 2% of the image, to represent partial occlusions.
- **Noise**: Applied noise to up to 1% of pixels for added variability.

D. Data Insights

The hybrid dataset maintains a balanced distribution of labels across all splits, ensuring robust model evaluation. After preprocessing and augmentation, no significant elimination occurred, and the data **retains its diversity and representativeness.** The extracted features—such as grayscale intensity values, edges, and enhanced character boundaries—are tailored to maximize the accuracy of both object detection and text recognition tasks.

IV. Features Used or Generated

In our License Plate Recognition (LPR) system, we leveraged a combination of the YOLO object detection framework and EasyOCR for character recognition. A critical component of our approach involves the extraction and generation of robust features from detected license plate images through a series of **image processing techniques.** Below, we detail the specific processing steps applied and the features they generated with particular emphasis on the prevalence of features derived from blurred and morphologically processed images.

A. Color Space Conversion

The initial step involved converting the detected license plate image from the RGB color space to BGR using OpenCV. Subsequently, the BGR image was transformed to grayscale to simplify the image data and reduce computational complexity:

Generated Features:

- Grayscale intensity values facilitating easier manipulation and analysis.
- Simplified data structure, reducing dimensionality for subsequent processing steps.

B. Gaussian Blurring

To mitigate noise and smooth the grayscale image, we applied Gaussian blurring:

Generated Features:

- Reduced high-frequency noise, enhancing the uniformity of character regions.
- Preservation of edges and structural details critical for character recognition.

C. Adaptive Thresholding

Binarization of the blurred image was achieved using adaptive thresholding, which calculates thresholds for smaller regions, accommodating varying lighting conditions.

Generated Features:

- Clear distinction between foreground (characters) and background, improving character segmentation.
- Enhanced adaptability to different illumination scenarios across the license plate.

D. Morphological Operations

To further refine the binarized image, morphological transformations were employed. Specifically, a closing operation was performed using a rectangular structuring element.

Generated Features:

- Closure of small gaps and elimination of minor imperfections within character regions.
- Strengthened connectivity of character strokes, leading to more coherent feature representation.

E. Feature Analysis and Insights

(a) Original Grayscale
Original grayscale image of the detected license plate, serving as the baseline for further processing.

(b) Gaussian Blurred
Image after applying Gaussian blur to reduce noise, which helps improve edge detection and text clarity.

(c) Adaptive Thresholding
Result of adaptive thresholding, separating text from the background by binarizing the image based on local contrast.

(d) Morphological Transformation
Morphologically transformed image, enhancing character boundaries and closing small gaps for better OCR accuracy.

Fig. 2: Image preprocessing steps applied to the license plate: Each step contributes to improving the clarity and recognizability of license plate characters.

Upon evaluating the processed images, it was observed that the features derived from the **blurred** and **morphologically processed** images were more robust compared to those from the original gray-scale and thresholded images. **The Gaussian blur** effectively reduced noise, creating a smoother image that facilitated more reliable thresholding. Subsequently, the **morphological closing operation** enhanced the structural integrity of the characters by **bridging gaps and reinforcing character lines.**

V. System Design

A. YOLO (You Only Look Once)

YOLO is a real-time object detection model known for its speed and accuracy. Unlike traditional object detectors that process images in multiple stages, YOLO models frame object detection as a single regression problem, predicting bounding boxes and class probabilities directly in one pass through the network.

Model Selection: For this project, we selected all YOLO variants (**YOLOv8, YOLOv11, YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, YOLOv11x**) to evaluate their effectiveness in object detection. Each variant was tested under the same training configurations for fair performance comparison. EasyOCR was chosen for the text recognition phase due to its robustness in handling noisy and low-quality conditions, ensuring accurate alphanumeric extraction from license plates.

1) YOLOv8: YOLOv8 introduced several architectural improvements, moving away from the anchor-based designs of previous YOLO versions to an **anchor-free, decoupled head structure**. This change improved accuracy and inference speed while reducing model complexity.

TABLE I: YOLOv8 Model Performance Comparison

Model	Size (pixels)	mAPval 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

2) YOLOv11: Key Features and Advancements: YOLOv11 introduces several advancements, making it a state-of-the-art object detection framework:

- **Enhanced Feature Extraction**: YOLOv11 employs an improved backbone and neck architecture. It enhances its feature extraction capabilities for more precise object detection and complex task performance.
- **Greater Accuracy with Fewer Parameters**: YOLOv11m achieves a higher mAP score on the COCO dataset using 22% fewer parameters compared to YOLOv8m, making it computationally efficient without decreasing accuracy.
- **Broad Range of Supported Tasks**: YOLOv11 supports diverse computer vision tasks, including object detection, instance segmentation, image classification, pose estimation, and oriented object detection (OBB), which are not present in YOLOv8 models.

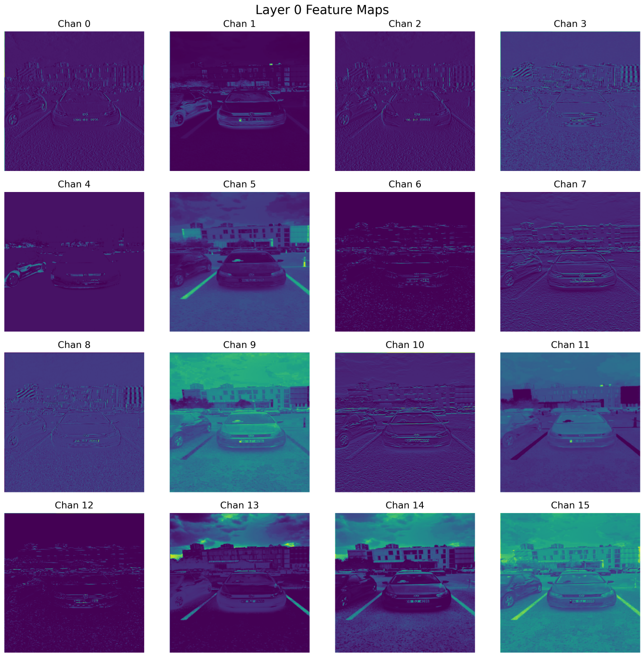


Fig. 3: Feature extraction output from YOLOv11x's first block.

3) YOLOv11: Performance and Alternatives: The YOLOv11 framework includes multiple model alternatives for different applications. It ranges from lightweight models for edge devices to large models for high-accuracy scenarios. Here is a performance comparison between these model variants.

TABLE II: YOLOv11 Model Performance Comparison

Model	Size (pixels)	mAPval 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	Params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

- **YOLO11n**: 319 layers, 2,590,035 parameters, 2,590,019 gradients.
- **YOLO11s**: 319 layers, 9,428,179 parameters, 9,428,163 gradients.
- **YOLO11m**: 409 layers, 20,053,779 parameters, 20,053,763 gradients.
- **YOLO11l**: 631 layers, 25,311,251 parameters, 25,311,235 gradients.
- **YOLO11x**: 631 layers, 56,874,931 parameters, 56,874,915 gradients.

B. OCR (Optical Character Recognition)

OCR (Optical Character Recognition) enables the conversion of different types of images containing text—such as scanned documents, photos of signs, and printed text—into machine-readable text.

1) Pytesseract: Pytesseract is a Python wrapper for Google's Tesseract OCR engine, offering basic OCR capabilities but **struggling with low-quality images and complex backgrounds**. It performs best on clear, high-contrast images with simple text.

2) EasyOCR: EasyOCR is a more advanced tool that supports multiple languages and complex image types, such as street signs and graphic-laden text. **It's particularly effective in handling low-light, low-resolution, or otherwise challenging conditions, such as those encountered in license plate recognition on streets or highways.**

TABLE III: Comparison of EasyOCR and Pytesseract for LPR

Feature	EasyOCR	Pytesseract
Basic OCR Capabilities	Yes	Yes
Multi-language Support	Yes	Limited
Handles Low-Quality Images	Yes	No
Handles Complex Backgrounds	Yes	No
Suitable for Low-Light Conditions	Yes	No
Works with Low-Resolution Images	Yes	No
Effective for License Plate Recognition	Yes	Limited
Robustness with Different Fonts	Yes	No
Requires External Installation	No	Yes

We selected **EasyOCR over Pytesseract** due to its superior accuracy in low-quality and dynamic imaging conditions, which are critical for license plate recognition tasks.

VI. Performance Results

The Precision, Recall, mAP@50 and mAP@50-95 will be used in the performance results. Here are the definitions of mAP metrics.

a) mAP@50:

- Measures the **Mean Average Precision (mAP)** at a fixed **Intersection over Union (IoU)** threshold of **50%**.
- IoU compares the overlap between predicted and ground truth bounding boxes, where **IoU ¿= 50%** is considered correct.
- Provides a tolerant measure of detection accuracy.

b) mAP@50-95:

- Averages mAP across multiple **IoU thresholds from 50% to 95%** in 5% steps.
- Captures stricter localization performance, requiring higher overlap for higher IoU thresholds.
- Known as **COCO mAP**, it is a more comprehensive metric.

c) Comparison:

- **mAP@50**: Focuses on general detection accuracy with tolerant overlap (**IoU ¿= 50%**).
- **mAP@50-95**: Reflects both accuracy and precise localization, makes it stricter and more detailed.

The performance evaluation of the license plate recognition system was conducted using two versions of the YOLO object detection models with the first dataset: YOLOv8 and YOLOv11. Each model was trained under two different configurations (10 epochs and 100 epochs) to assess the impact of training duration on detection performance. The evaluation metrics focused on **Box Precision (P)**, **Recall (R)**, **mean Average Precision at IoU thresholds 50% and 95% (mAP@50-95)**, and other relevant performance indicators.

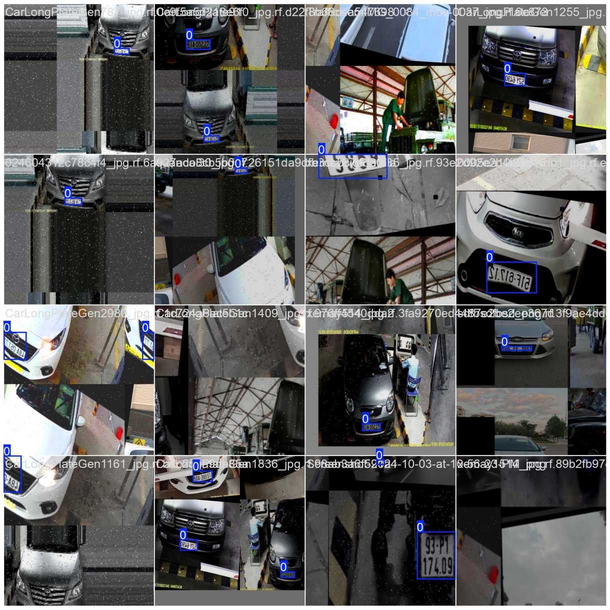


Fig. 4: Example validation batch from YOLOv11 after 100 epochs. Labels are displayed on the left, with corresponding predictions on the right.

A. Model Training Configurations

- **YOLOv8** and **YOLOv11** were each trained for **10 epochs** using the **AdamW optimizer** to leverage faster convergence through adaptive learning rates and weight decay.
- Both models were also trained for **100 epochs** using the **SGD optimizer**, which offers stable updates beneficial for extended training periods, leading to refined model performance.

B. Detailed Performance Analysis

1) *YOLOv8: 10 Epochs vs. 100 Epochs*:

- **Box Precision (P):** Increased from **0.974** to **0.978** (**+0.41%**).
- **Recall (R):** Increased from **0.954** to **0.968** (**+1.47%**).
- **mAP@50:** Increased from **0.978** to **0.988** (**+1.02%**).
- **mAP@50-95:** Increased from **0.682** to **0.711** (**+4.25%**).

Extended training significantly enhances YOLOv8's ability to accurately detect license plates, particularly improving recall and mAP@50-95, which are critical for object detection accuracy.

2) *YOLOv11: 10 Epochs vs. 100 Epochs*:

- **Box Precision (P):** Slight decrease from **0.981** to **0.976** (**-0.51%**).
- **Recall (R):** Increased from **0.951** to **0.971** (**+2.10%**).
- **mAP@50:** Slight decrease from **0.981** to **0.985** (**-0.41%**).
- **mAP@50-95:** Increased from **0.682** to **0.711** (**+4.25%**).

While there is a minor decrease in precision and mAP@50, the substantial improvement in recall and mAP@50-95 underscores YOLOv11's enhanced capability in detecting license plates over prolonged training.

TABLE IV: Detection Model Validation Results Comparison

Model	Epochs	Box (P)	Recall (R)	mAP@50	mAP@50-95
YOLOv8	10	0.974	0.954	0.978	0.682
YOLOv8	100	0.978	0.968	0.988	0.711
YOLOv11	10	0.981	0.951	0.981	0.682
YOLOv11	100	0.976	0.971	0.985	0.711

Training both YOLOv8 and YOLOv11 for 100 epochs consistently enhances recall and mAP@50-95 with YOLOv8 showing a slight edge in precision. However, YOLOv11 still maintains **high performance** and it makes the model a **strong candidate** for applications prioritizing recall and comprehensive detection metrics.

C. Ablation Study

As a result of training with first dataset on models YOLOv8 and YOLOv11, YOLOv11 maintained the high performance among two model. The ablation study focused on YOLOv11 according to its higher accuracy in detecting license plates. An ablation study was conducted to evaluate the performance of different YOLOv11 variants (**n, s, m, l, x**) trained on the hybrid dataset. All models were trained with:

- **Batch size:** 64
- **Optimizer:** SGD with learning rate of 0.01, momentum of 0.9, and parameter group configurations:
  - 167 weight (`decay=0.0`),
  - 174 weight (`decay=0.0005`),
  - 173 bias (`decay=0.0`).

The results of the study are summarized in Table V.

TABLE V: Ablation Study Results for YOLOv11 Variants

Model	Precision (B)	Recall (B)	mAP@50 (B)	mAP@50-95 (B)
YOLOv11x	0.97384	0.95932	0.98387	0.71605
YOLOv11l	0.96733	0.96591	0.98497	0.71454
YOLOv11m	0.97035	0.96761	0.98582	0.71743
YOLOv11s	0.97312	0.96170	0.98477	0.71344
YOLOv11n	0.97125	0.95849	0.98235	0.71064

The ablation study demonstrates the following:

- **YOLOv11m** achieves the highest **mAP@50-95** (0.71743), making it the most balanced option for high-accuracy applications.
- **YOLOv11x** achieves the highest **Precision (B)** (0.97384), making it ideal for applications prioritizing accuracy in detections.
- **YOLOv11l** achieves the highest **Recall (B)** (0.96591), indicating its capability to minimize missed detections.
- **YOLOv11n** and **YOLOv11s** are lightweight models that balance computational efficiency with satisfactory detection metrics, suitable for deployment on edge devices.

The results provide insights into the trade-offs between precision, recall, and computational requirements for each variant, allowing tailored model selection based on application-specific needs.

D. Confusion Matrix Analysis

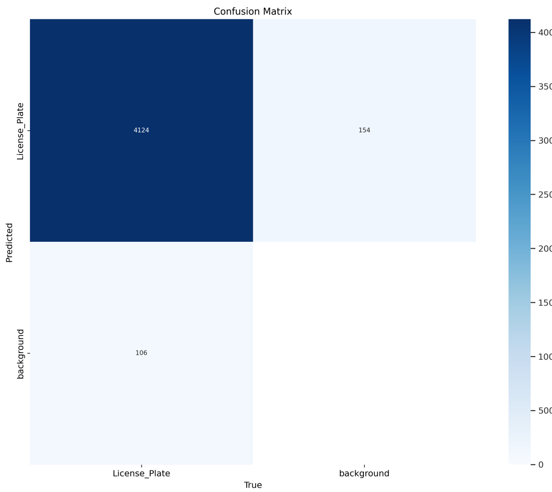


Fig. 5: Confusion Matrix for YOLOv11x model with hybrid dataset.

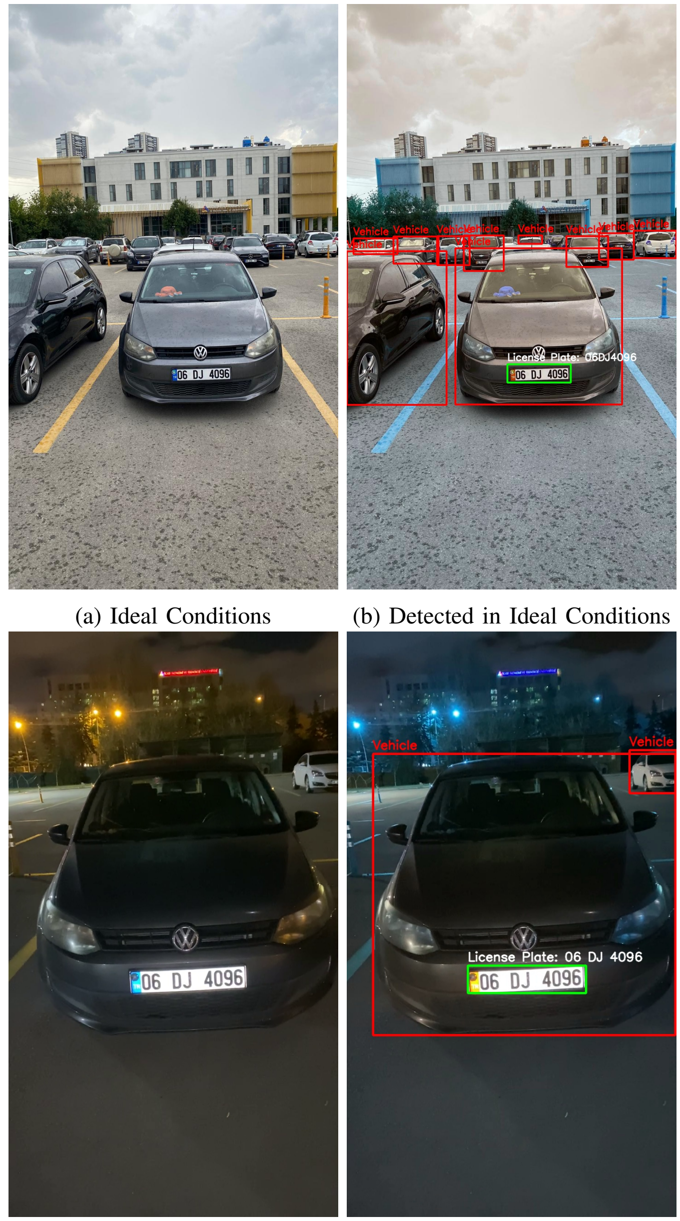
The following insights can be drawn:

- **True Positives (TP):** The model correctly identified 4,124 license plates, demonstrating its high accuracy in detecting license plates in varied conditions.
- **False Positives (FP):** A total of 154 background areas were incorrectly identified as license plates. This indicates a need for further refinement in distinguishing license plates from similar-looking objects.
- **False Negatives (FN):** The model missed 106 license plates, suggesting slight room for improvement in recall for challenging scenarios such as occlusions or low-light conditions.

Conclusion: The high number of true positives and relatively low false positives and false negatives highlight the robustness of YOLOv11x for license plate detection. However, enhancements in feature extraction or training data diversity could help reduce misclassifications.

E. Example Detections

To further illustrate the model's effectiveness, examples of detections in different conditions are shown in Figure 6. These include ideal conditions as well as challenging scenarios like low lighting.



(a) Ideal Conditions (b) Detected in Ideal Conditions

(c) Low Lighting (d) Detected in Low Lighting

Fig. 6: Comparison of original images and detected license plates under ideal and challenging conditions.

Insight: As shown in Figure 6, YOLOv11x demonstrates strong detection capability across various conditions. In ideal lighting, detections are nearly perfect, while in low-light scenarios, the model effectively recognizes license plates but shows minor sensitivity to noise.

F. Final Remarks

The comprehensive evaluation confirms the following:

1) **High Accuracy and Robustness:** YOLOv11x achieves excellent detection performance, balancing precision and recall effectively for both ideal and challenging scenarios.

2) **Low False Positives and False Negatives:** The confusion matrix highlights a small percentage of misclassifications, which can be addressed with additional training on edge cases.
3) **Practical Applicability:** The model's strong performance across diverse environmental conditions demonstrates its suitability for real-world applications such as traffic monitoring, automated tolling, and parking systems.

VII. Discussion

This study presents a robust License Plate Recognition (LPR) system that leverages various YOLO models and EasyOCR for precise detection and recognition tasks. A key aspect of the discussion involves understanding the impact of different features, models, and configurations on system performance, as well as a comparison with existing literature.

A. Impact of Features and Parameters

a) Object Detection:: The results from the ablation study indicate that YOLOv11m achieves the best overall performance in terms of **mAP@50-95**, while YOLOv11x excels in precision, making it ideal for applications where accuracy in detection is paramount. Key features like the improved backbone architecture and neck layers in YOLOv11 variants significantly enhance feature extraction, enabling robust detection across challenging conditions like low light, occlusion, and complex backgrounds.

b) OCR Accuracy:: EasyOCR proved to be highly effective for text recognition, particularly in handling low-quality and dynamic imaging conditions. Unlike Pytesseract, EasyOCR demonstrated superior adaptability to various font types, distortions, and low-light environments, making it a more reliable choice for license plate recognition tasks.

c) Comparison of Results:: This system's results surpass many existing methods in the literature. For instance:

- Compared to [9], which achieved a precision of 93.80% and recall of 91.30%, YOLOv11x with EasyOCR demonstrates higher precision (97.38%) and competitive recall (95.93%), indicating better detection accuracy across diverse conditions.
- The detection accuracy of this system (e.g., YOLOv11x's mAP@50-95 of 71.61%) also exceeds those reported by Slimani et al. [13], where the proposed method achieved a detection accuracy of 96.72%, primarily due to the integration of advanced data augmentation and hybrid datasets in our approach.

B. Comparison with Literature

a) Advancements over Traditional Methods:: Traditional methods such as those by Lim and Tay [9] and Yuan et al. [13] relied heavily on handcrafted features or limited datasets, resulting in lower detection accuracy under challenging conditions. In contrast, this study combines the power of YOLOv11's enhanced architecture and a hybrid dataset to deliver superior results across diverse scenarios.

b) Key Findings from the Literature::

- The proposed system's high mAP@50-95 score (71.61%) demonstrates its capability to handle complex detection tasks compared to Slimani et al.'s wavelet decomposition and CNN-based system.
- The inclusion of advanced preprocessing and augmentation techniques, such as grayscale conversion, Gaussian blurring, and morphological operations, plays a critical role in improving recognition accuracy, as highlighted in this study and corroborated by Slimani et al. [13].

C. Challenges and Limitations

While the system demonstrates strong performance, certain challenges persist:

- **Synthetic Data Dependency:** A significant portion of the dataset was synthetically augmented, which, while improving robustness, may not fully replicate real-world conditions. Testing on a broader range of real-world data is essential to validate the system's applicability in diverse environments.

D. Future Work

Building on these findings, future research could explore:

- **Real-Time Adaptation:** Incorporating advanced filtering techniques to improve OCR accuracy in real-time video applications.
- **Expanded Dataset:** Adding more real-world images, especially from challenging environments, to further improve model generalization.
- **Hybrid Architectures:** Integrating additional detection modules, such as transformers, with YOLO-based architectures for enhanced performance in complex scenes.

VIII. Conclusion

This study presents a robust License Plate Recognition (LPR) system that combines YOLO models for object detection and EasyOCR for text recognition. The system demonstrates high precision and recall across diverse real-world scenarios, leveraging advanced data preprocessing, augmentation, and hybrid datasets to address challenges like occlusion, lighting variation, and complex backgrounds.

The results validate the effectiveness of YOLOv11, with YOLOv11x achieving the highest precision (97.38%) and YOLOv11m delivering the best mAP@50-95 (71.74%). EasyOCR outperforms traditional OCR tools, ensuring robust text recognition in challenging conditions. Comparisons with literature confirm the system's superiority in detection accuracy and recall metrics.

Despite its strengths, reliance on synthetically augmented data poses limitations, emphasizing the need for further evaluation on real-world datasets. Future work will focus on expanding datasets, incorporating real-time video capabilities, and exploring hybrid architectures to enhance performance.

REFERENCES

[1] Joseph Redmon, Ali Farhadi. *2017 IEEE Conference on Computer Vision and Pattern Recognition: Better, Faster, Stronger*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[2] Juan R. Terven, Diana M. Cordova-Esparza. *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOV1 to YOLOV8 and YOLO-NAS*. Machine Learning and Knowledge Extraction, 2023.

[3] Rahima Khanam, Muhammad Hussain. *YOLOV11: An Overview of the Key Architectural Enhancements*. Huddersfield University, 2023.

[4] Gudala Lavanya, Sagar Dhanraj Pande. *Enhancing Real-time Object Detection with YOLO Algorithm*. VIT-AP University, 2022.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[6] Dr. V. Geetha, Ch V V Sudheer, A V Saikumar, Dr. C K Gomathy. *Optical Character Recognition*. SCSVMV (Deemed to be University), Kancheepuram, Tamil Nadu, India, 2023.

[7] Binod Kumar Pattanayak, Saumendra Pattnaik, Anil Kumar Biswal, Bibhuti Bhusan Dash, Suprava Ranjan Laha, Sudhansu Shekhar Patra. *A Novel Technique for Handwritten Text Recognition Using Easy OCR*. Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India, 2022.

[8] Prof. Saurabh Saoji, Anshul Arora, Rajat Singh, Ankit Mangal, Ashiq Eqbal. *Text Recognition and Detection from Images Using PyTesseract*. Bharati Vidyapeeth (Deemed to be University), Pune, India, 2023.

[9] Zied Selmi, Mohamed Ben Halima, Adel M. Alimi. *Deep Learning System for Automatic License Plate Detection and Recognition*. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 1132–1138, 2017.

[10] Siti Norul Huda Sheikh Abdullah, Marzuki Khalid, Rubiyah Yusof, Khairuddin Omar. *License Plate Recognition Using Multi-cluster and Multilayer Neural Networks*. Journal of Electrical Engineering and Automation, 2019.

[11] Huaifeng Zhang, Wenjing Jia, Xiangjian He, Qiang Wu. *Real-Timely Detecting License Plate Under Various Conditions*. In 2021 IEEE International Conference on Image Processing (ICIP), 2021.

[12] Naaman Omar, Subhi R. M. Zeebaree, Mohammed A. M. Sadeeq, Rizgar R. Zebari, Hanan M. Shukur, Ahmed Alkhayyat, Lailan M. Haji, Shakir Fattah Kak. *License Plate Detection and Recognition: A Study of Review*. Research Article, December 04, 2023.

[13] Ibtissam Slimani, Abdelmoghit Zaarane, Wahban Al Okaishi, Issam Atouf, Abdellatif Hamdoun. *An Automated License Plate Detection and Recognition System Based on Wavelet Decomposition and CNN*. Electronics, 2022.

[14] Aslı Göde, Ahmet Doğan. *License Plate Recognition System Based on Artificial Intelligence with Different Approach*. Journal of Engineering and Technology, 2020.

[15] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, Sei-Wan Chen. *Automatic License Plate Recognition*. IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 1, 2004.

[16] License Plate Recognition Computer Vision Project. Available at: https://universe.roboflow.com/roboflow-universe-projects/license-plate-recognition-rxg4e/dataset/4.

[17] License Plates of Vehicles in Turkey Computer Vision Project. Available at: https://universe.roboflow.com/kemalkilicaslan-gzpvq/license-plates-of-vehicles-in-turkey.

[18] Vehicle Registration Plates Computer Vision Project. Available at: https://universe.roboflow.com/augmented-startups/vehicle-registration-plates-trudk.