Date available online: 09-01-2026
Vol. 3, Issue 1 (July – December 2025)
This is an open-access article.
DOI: https://doi.org/10.24312/ucp-jeit.03.02.708

End-to-End Motorcycle Violation Detection with Region-Specific Automatic License Plate Recognition

Mohamed Rafi Atheek[1], Mohamed Buhary Fathima Anizul Fathool[2], Atif Ishaq Khan[3]
[1]Department of Computer Science, GC University, Lahore, Pakistan (e-mail: mhdatheek136@gmail.com)
[2]Department of Computer Science, GC University, Lahore, Pakistan (e-mail: anizulfathool@gmail.com)
[3]Department of Computer Science, GC University, Lahore, Pakistan (e-mail: atif.ishaq@gcu.edu.pk)

Corresponding author: Atif Ishaq Khan (e-mail: atif.ishaq@gcu.edu.pk).

ABSTRACT
We present a region-aware, end-to-end motorcycle violation detection pipeline tailored to traffic conditions in Punjab, Pakistan, which integrates three YOLOv11-based components into a unified framework: motorcycle violation detection (MCVD) for helmet compliance and multi-rider analysis, license plate detection (LPD), and license plate character detection (LPCD). The system integrates lightweight object detection, BoT-SORT-based tracking, and character-level recognition, supported by a synthetic-toreal adaptation strategy that combines large-scale synthetic data with limited real samples. Two specific datasets are published, a 40,000-sample synthetic Punjab license plate dataset (PS-LPCD) and a 650-sample real-world dataset (PR-LPCD), which are publicly released in order to encourage research development and adaptation to the region. Class consolidation enhanced MCVD performance (weighted average F1 score: 0.77) and the LPD model performed at $\text{mAP}_{50} = 0.99$. Two-stage fine-tuning on synthetic and real samples allowed LPCD to reach a character accuracy of ≈ 98% and a full-plate recognition rate of ≈ 90.7%, both surpassing EasyOCR and PaddleOCR, while also achieving lower per-plate latency. With a single motorcycle per frame, the sequential pipeline maintains a throughput of ≈ 9.5 FPS; the throughput reduces in scenes where there are many motorcycles. These findings indicate that synthetic pretraining, together with a small real fine-tuning, can be used to obtain a powerful, scalable, and region aware automatic license plate recognition (ALPR) system, which provides a reproducible method for detecting traffic violations across a variety of license-plate formats.

INDEX TERMS ALPR, Helmet compliance, License plate recognition, Motorcycle violation detection, Multi-rider counting, Punjab, Synthetic dataset, YOLOv11

I. INTRODUCTION

Motorcycle-related traffic violations are a major contributor to road injuries and fatalities worldwide. The World Health Organization (WHO) reports that motorcyclists account for 21% of all road traffic deaths [1]. In Pakistan, motorcycles are a dominant mode of transport and are disproportionately represented in crash statistics [2], [3], underscoring the need for effective, region-aware monitoring and enforcement systems.

Although the previous literature has already generated precise approaches to individual tasks such as helmet detection [4], multi-rider counting [4], and automatic license plate recognition (ALPR) [5], most systems address these sub-tasks in isolation. Moreover, reliance on generic datasets and off-the-shelf OCR engines (e.g., EasyOCR, Tesseract) limits robustness in regions where license plate formats and scripts vary. This constrains both applicability and reproducibility in real-world deployments.

To fill these gaps, we propose an end-to-end motorcycle violation detection pipeline based on YOLOv11. We emphasize that this work does not introduce new detection architectures or learning algorithms. Instead, the novelty lies in a systems-level integration of existing state-of-the-art components, combined with region-specific dataset design, synthetic-to-real adaptation, and deployment-oriented evaluation. The system integrates three modules: MCVD (Motorcycle Violation Detection) for helmet-use and multi-rider detection, LPD (License Plate Detection) for plate localization, and LPCD (License Plate Character Detection) for character recognition. Together, these components form a practical, reproducible, and region-aware motorcycle violation detection pipeline.

OUR MAIN CONTRIBUTIONS ARE AS FOLLOWS:

- A unified, end-to-end YOLOv11-based pipeline integrating helmet detection, rider counting, and ALPR through system-level design.
- Two new Punjab-specific license plate character detection datasets (synthetic and real) released to support region-aware ALPR research and reproducibility.
- A lightweight character-level detection approach that improves ALPR robustness compared to off-the-shelf OCR engines.
- Models trained and evaluated on augmented public

datasets to ensure both reproducibility and regional applicability.

II. RELATED WORK

Initially, the detection of helmet-use was based on handcrafted features and classical classifiers. As an example, SVMs were used on the histograms of head regions with background subtraction and projection profiling [6], moving blob extraction with K-Nearest Neighbor classification [7], and combined LBP, HOG, and Hough descriptors, achieving an accuracy of 94.23% [8]. These methods were generally sensitive to lighting, occlusion, and crowding.

Deep learning has enhanced its strength and enabled joint tasks. CNN-based classifiers, for example, achieved high accuracy (96.6%) and F_1-score (94.6%) [9]. Various other pipelines, including YOLO-based and alternative approaches, have also been applied for helmet detection and multi-rider identification [4], [10]–[15]. Nevertheless, although the above-mentioned methods work well in their intended applications, they are typically not integrated with the ALPR systems, which restricts their use in end-to-end motorcycle violation detection pipelines.

In the case of Pakistan, the research on the topic has focused on individual tasks and not on comprehensive end-to-end motorcycle violation detection. Deep learning models have been successful in identifying the location of the helmet on a surveillance video with high accuracy [16], [17], while ALPR systems have focused on license plate localization, character segmentation, and OCR [18], [19]. Motorcycle-based end-to-end pipelines involving the detection of helmet violations, multi-rider, and region-specific license plate recognition are still uncommon. This gap is addressed in our work, where a unified framework is proposed, which is specific to Punjab, Pakistan.

Several works outside Pakistan have integrated helmet detection with ALPR in end-to-end pipelines. In some cases, evaluation relied on proprietary datasets or generic OCR systems, and regional plate variations were not always addressed [20]–[24].

Synthetic data has emerged as a viable solution to the scarcity and privacy issues of license plate datasets. Template-based methods [25], rendering pipelines [26], and diffusion models [27] have shown measurable gains in recognition accuracy. Based on such methods, we generate a template-based synthetic dataset of character-level annotated license plates specifically for the Punjab, Pakistan region, complemented with manually labeled real images for evaluation. These findings, along with benchmarking studies [28], demonstrate the viability of synthetic plate generation as a reliable supplement to real data.

III. METHODOLOGY

A. SYSTEM OVERVIEW

The suggested end to end Motorcycle Violation Detection system will be used to monitor helmet compliance, multi riders, and license plate recognition. The pipeline will be composed of motorcycle detection, tracking, license plate detection, character recognition, and violation classification as illustrated in Figure 1.

The detection of motorcycles was conducted with the help of the YOLOv11 model that was trained on the COCO dataset [29]. To optimize efficiency, the system first employs lightweight detection and BoT-SORT tracking [30] to identify candidate violation frames where motorcycles and license plates are both visible and potentially readable. Only these frames are then processed with the heavier MCVD YOLOv11m model, ensuring a balance between accuracy and computational cost. This design prevents violations from being logged on unreadable plates, which is essential for reliable automated enforcement.

B. DATASETS AND PREPROCESSING

1) Motorcycle Violation Dataset (HELMET)

The HELMET dataset [4] is a widely used benchmark for helmet-use and multi-rider detection and was adopted for training the MCVD model. It comprises 91,000 annotated frames with 283,377 labeled object instances spanning 36 fine-grained classes. As is typical of real-world traffic data, this detailed class structure introduces a substantial class imbalance, with several safety-critical violations occurring far less frequently than compliant riding behaviors.

To address this imbalance, two complementary strategies were employed. First, class consolidation was performed to simplify the label space and better reflect traffic enforcement practices. The front-child passenger (P0) class was removed, and all cases involving three or more riders were grouped into a single *MoreThanTwoRider* category, since any rider count exceeding two constitutes a violation regardless of helmet usage. Second, a targeted sampling approach was applied during training, where horizontal flipping augmentation was restricted to underrepresented classes (those with fewer than 20,000 samples). This selectively increases the representation of minority classes without distorting the natural distribution of dominant categories.

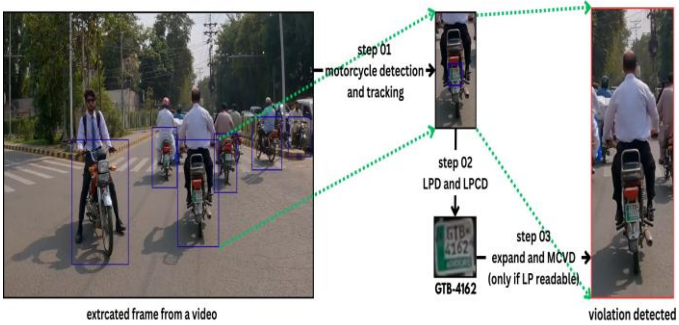

FIGURE 1. Pipeline of the proposed End-to-End Motorcycle Violation Detection system, including motorcycle detection, tracking, license plate recognition, and violation classification.

Following consolidation and augmentation, the dataset contained 318,131 annotated instances across seven violation-relevant classes. The class distributions before and after consolidation are provided in Appendix A.

The dataset was divided into 70% for training, 10% for validation, and 20% for testing. Since the original dataset consists primarily of CCTV footage in which motorcycles often appear at a distance, near-camera close-up views of riders and helmets are underrepresented. To address this limitation and improve robustness to real-world scale variations, we applied a two-step augmentation strategy:

1) **Close-up cropping:** Every 10th frame containing a motorcycle was cropped with a padding of 0.1 (as a fraction of the bounding-box size) to synthetically generate near-camera views while preserving annotation coordinates.

2) **Super-resolution enhancement:** The resulting low-resolution close-up crops were enhanced using RealESRGAN [31] to recover fine-grained details and improve object detectability.

In addition, mosaic blending was applied to simulate dense traffic conditions and improve robustness to occlusion and scale variation.

2) License Plate Detection Dataset (UFPR-ALPR)

In the case of license plate detection, we resorted to the UFPR-ALPR dataset that consists of 4,500 annotated images that represent various types of vehicles [5]. Each annotation included a license-plate bounding box and metadata (vehicle type, camera type, lighting conditions). We focused on plates that are visible on motorcycles and on vehicle types that are relevant for LPD.

To improve regional relevance for Punjab, Pakistan, we generated synthetic Punjab-style license plates and replaced the original plates in the images while preserving plate aspect ratios (single-line vs. double-line), as shown in Figure 3. This augmentation doubled the dataset to 9,000 images. Following the original dataset recommendations, the split was 40% training (3,600 images), 40% validation (3,600 images), and 20% test (1,800 images) [5].

Training augmentations for the LPD model included mosaic blending, shear, perspective deformation, and limited horizontal flipping. This set of augmentations simulates viewpoint variation and minor geometric distortions while preserving plate legibility.

3) License Plate Character Dataset (PS-LPCD and PR-LPCD)

To train a robust character-level detector, we created the Punjab Synthetic License Plate Character Dataset (PS-LPCD) and a complementary real-world dataset, the Punjab Real License Plate Character Dataset (PR-LPCD). PS-LPCD contains 40,000 synthetic images generated across four Punjab plate templates, while PR-LPCD comprises 650 annotated crops extracted from the PK-Number-Plates-V3 collection [32]. After filtering for Punjab templates, 500 samples were reserved for fine-tuning and 150 for final testing. PSLPCD was split into 80% for training and 20% for validation. Sample synthetic examples are shown in Figure 4.

Both datasets are freely available for research purposes as part of the Punjab Pakistan Synthetic and Real License Plate Character Datasets (P-LPCD), available at Zenodo (https://doi.org/10.5281/zenodo.17182320) [33].

PS-LPCD contains 40,000 synthetic images equally divided among four Punjab plate templates (front/back × old/new; 10,000 images per template). We annotated 37 classes: digits 0–9, uppercase letters A–Z, and a special class "PUNJAB" used to detect decorative or regional markers and to filter irrelevant glyphs. Synthetic images were randomized with the following augmentations to emulate real capture artifacts:

- Spatial transforms applied with probability 0.7: translation ±10 pixels, shear ±15°, rotation ±15°.
- Perspective warp (small magnitude) to simulate viewpoint changes.
- Photometric and environmental noise: dirt, dust, Gaussian noise, and blur.
- Motion blur applied with probability 0.5; kernel size n chosen randomly from odd integers in [1, 29]. The discrete motion-blur kernel $K_{i,j}$ is defined as

$$K_{i,j} = \begin{cases} \dfrac{1}{n}, & \text{if } i = \lceil n/2 \rceil \quad \text{(horizontal blur row)}, \\ \dfrac{1}{n}, & \text{if } j = \lceil n/2 \rceil \quad \text{(vertical blur column)}, \\ 0, & \text{otherwise}. \end{cases} \tag{1}$$

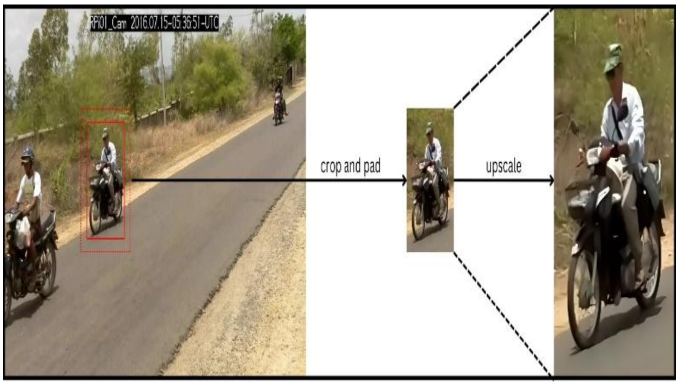

FIGURE 2. Two-stage super-resolution augmentation process. The first stage generates synthetic close-up views through cropping, and the second stage applies Real-ESRGAN to enhance visual details for improved detection performance.

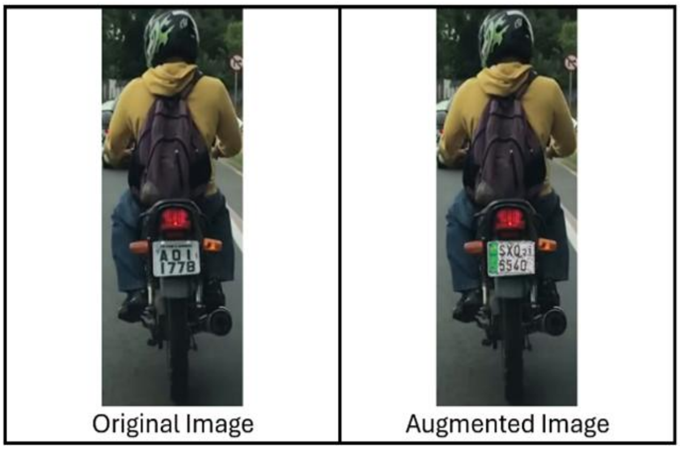

FIGURE 3. Synthetic replacement of license plates to adapt the dataset to regional characteristics.

This kernel produces a uniform linear blur across the central row (horizontal) or column (vertical), modeling motion along the principal axes.

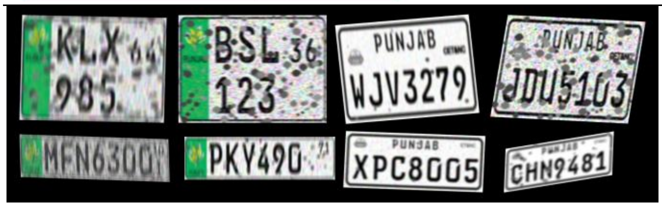

FIGURE 4. Sample synthetic license plates from PS-LPCD.

C. MODEL ARCHITECTURE AND TRAINING

The suggested system utilizes the models based on the YOLOv11 to perform three main tasks, namely, Motorcycle Violation Detection (MCVD), License Plate Detection (LPD), and License Plate Character Detection (LPCD). Two model versions were used: YOLOv11n, a small model with real time inference, and YOLOv11m, a large model that is designed to achieve the highest accuracy possible at the cost of computational performance [34].

All models were initialized with weights pretrained on the COCO dataset [35], a large-scale benchmark dataset. Pretraining provides transferable and generalized feature representations that improve performance across various vision tasks [36]. Task-specific data preprocessing and augmentation strategies were then applied to enhance robustness under challenging traffic conditions, including occlusion, motion blur, and varied viewpoints. For example, MCVD training incorporated moderate mosaic blending to improve detection in dense scenes, while LPCD training disabled mosaic augmentation and horizontal flips to preserve character orientation. The LPD training used a mixed augmentation, which consisted of mosaic blending, shear, perspective deformation, and restricted flips to preserve the geometry of the license plates. In addition to these custom settings, all models utilized the default augmentation pipeline provided in the YOLOv11 documentation [29].

The training was done on a 64-bit system with an NVIDIA RTX 2060 GPU (VRAM: 6GB) and an AMD Ryzen 7 4800H CPU (with 8 cores and 16 threads), along with 40 GB of RAM, and operated under the Windows operating system. Hyperparameters were tuned to balance accuracy and inference efficiency. In particular, the MCVD model was trained using the SGD optimizer with momentum, as the HELMET dataset is large and SGD is known to offer better generalization on large datasets. In contrast, the LPD and LPCD models were trained using the Adam optimizer, since their datasets are medium or small in size, where adaptive methods such as Adam converge faster [37]. The overall training configuration is summarized in Table 1.

TABLE 1. Training configurations for MCVD, LPD, and LPCD models

Model	Epochs	Batch Size	Optimizer
MCVD (YOLOv11n)	20	16	SGD
MCVD (YOLOv11m)	15	8	SGD
LPD (YOLOv11n)	50	16	Adam
LPD (YOLOv11m)	20	8	Adam
LPCD (YOLOv11n)	20	16	Adam

All models were trained with an image size of 640 × 640.

Complete training and validation curves, including loss (box, cls, dfl), mAP, precision, and recall for all YOLOv11 models, are provided in Appendix B for reference and reproducibility.

In the case of LPCD, domain adaptation was performed through a two-step fine-tuning process on the Punjab Real License Plate Character Dataset (PR-LPCD) following the initial training, as shown in Table 1. In Stage One, the first eight layers were frozen, and the model was trained for 10 epochs with a learning rate of 1×10^{-5} and a batch size of 8. In Stage Two, only the first four layers remained frozen, and training continued for another 10 epochs with a reduced learning rate of 1×10^{-6}. This gradual unfreezing approach allowed the model to fit well to real-world data and alleviate overfitting while retaining the generalizable characteristics acquired in synthetic training.

D. LICENSE PLATE CLASSIFICATION

The LPCD pipeline includes a license plate layout classification step in order to allow the correct sequencing of the detected characters, separating single-line and double-line plates. This difference is essential, as the position of the characters varies dramatically across layouts.

The classification is based on a normalized vertical variation measure that is calculated using the bounding boxes of the identified characters. Let y_i denote the vertical center of the i^{th} character, and h_i its bounding box height. The metric is defined as:

$$\text{Normalized Variation} = \frac{\sigma_y}{\mu_h} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2}}{\frac{1}{N}\sum_{i=1}^{N} h_i} \quad (2)$$

where σ_y represents the standard deviation of the vertical centers, μ_h the mean character height, and N the total number of detected characters.

Plates where the normalized variation was more than 0.45 were defined as double-line because the characters were more vertically spread. Values below this threshold indicated single-line plates. The threshold was chosen empirically in steps of 0.05 by comparing the accuracy of classification on the PR-LPCD data.

After classification, single-line plates were read sequentially, whereas double-line plates were parsed line by line. For double-line plates, a vertical midline, computed as the average of all vertical points, separates the two lines, which are read independently. Additional heuristics were incorporated to improve robustness, such as identifying the smaller-sized final two digits of the registration year that commonly appear on Punjab single-line plates.

Figure 5 illustrates the process, showing bounding box distributions for singleand double-line plates. A vertical yellow midline separates the two lines in double-line plates, while red dots mark each character's center, and dotted red lines indicate the spread from the midline.

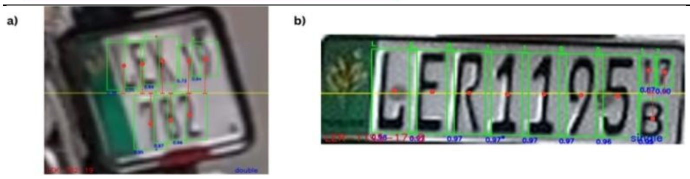

FIGURE 5. Normalized Vertical Variations for Single-Line and Double-Line License Plates. Red dots indicate character centers, dotted red lines show the vertical spread, and the yellow midline separates the two lines in double-line plates.

IV. RESULTS AND DISCUSSION

A. *EVALUATION METRICS*

The performance of the proposed MCVD system was evaluated using metrics that reflect both detection accuracy and computational efficiency. For object detection, the primary metric is mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), which quantifies the model's ability to correctly localize and classify objects. Formally, for *C* classes and AP_c representing the average precision for class *c*:

$$\mathrm{mAP@0.5} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{AP}_c \qquad (3)$$

To provide a stricter assessment of detection robustness, mAP averaged over IoU thresholds from 0.5 to 0.95 in 0.05 increments (mAP@[0.5:0.95]) is computed as:

$$\mathrm{mAP@[0.5{:}0.95]} = \frac{1}{10}\sum_{t=0.5}^{0.95}\mathrm{mAP@}t \qquad (4)$$

Given the residual imbalance across violation categories, the weighted F1-score was used as the primary evaluation metric for MCVD to ensure fair performance assessment across both frequent and rare classes. This is defined as:

$$F1_{\text{weighted}} = \sum_{c=1}^{C} w_c \cdot F1_c, \quad w_c = \frac{n_c}{\sum_{i=1}^{C} n_i} \qquad (5)$$

where n_c denotes the number of true samples in class *c* and $F1_c$ is the F1-score for that class.

In addition, standard evaluation metrics such as accuracy, precision, recall, and F1-score are computed to provide additional insights into the system's performance. These are defined as:

$$\mathrm{Precision} = \frac{TP}{TP+FP} \qquad (6)$$

$$\mathrm{Recall} = \frac{TP}{TP+FN} \qquad (7)$$

$$F1 = \frac{2\cdot\mathrm{Precision}\cdot\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \qquad (8)$$

where *TP* , *FP* , *FN* , and *TN* represent true positives, false positives, false negatives, and true negatives, respectively.

To assess computational efficiency, inference speed was measured in frames per second (FPS). Higher FPS values indicate faster processing, which is crucial for real-time traffic monitoring applications.

B. *MOTORCYCLE VIOLATION DETECTION (MCVD) MODELS*

Three YOLO-based MCVD models were trained to investigate the effects of model size and architecture and to identify the best-performing model. Their performance was also compared with the CNN-MTL baseline (CNN-based Multi-Task Learning for helmet detection) proposed by Lin *et al.* [4], as shown in Table 2.

Consolidating the original 36 classes into 7 core violation classes significantly improved detection performance. YOLOv11n trained on 7 classes achieved an mAP@50 of 0.6584, nearly doubling YOLOv8n's 0.3514 trained on all 36 classes. This reduction in classes helped reduce label noise and class imbalance, thereby boosting accuracy and model focus. Notably, even with the same 36 classes, YOLOv8n outperformed the CNN-MTL baseline (F1 score 0.70 vs. 0.673), likely due to the more efficient and advanced YOLO architecture. YOLOv11m further improved performance, reaching an mAP@50 of 0.7127 and a weighted F1 score of 0.77, demonstrating the combined benefits of class consolidation and targeted minority-class augmentation in mitigating class imbalance and improving detection robustness.

For further analysis, the consolidated classes were grouped into *Non-Violation* (DHelmet, DHelmetP1Helmet) and *Violation* (DNoHelmet, DNoHelmetP1NoHelmet, DHelmetP1NoHelmet, DNoHelmetP1-Helmet, MoreThanTwoRider) categories, yielding the weighted average results shown in Table 3.

These results further illustrate the effectiveness of the proposed imbalance mitigation strategies, as minority violation classes benefit from improved recall without sacrificing precision on dominant non-violation categories.

The models show higher precision, recall, and F1-scores for compliant rider classes, effectively reducing false positives and improving the classification of non-violators. Inference speed tests on 1,000 random test images indicate that YOLOv11m achieves near real-time performance at approximately 25 frames per second (FPS), comparable to YOLOv11n's ~27 FPS, maintaining efficiency despite increased complexity.

Despite the strong overall performance, several failure

TABLE 2. Performance comparison of MCVD models, including YOLO variants and CNN-MTL baseline

MCVD Model	Precision	Recall	mAP@50	mAP@[50–95]	Classes	Weighted Average F1 Score
CNN-MTL [4]	–	–	–	–	36	0.673
YOLOv8n	0.3803	0.3889	0.3514	0.3104	36	0.70
YOLOv11n	0.6505	0.6157	0.6584	0.5915	7	0.72
YOLOv11m	0.7096	0.6664	0.7127	0.6517	7	0.77

cases were observed in challenging real-world scenarios. In highly crowded scenes, riders positioned very close to each other can be confused, leading to false predictions due to inter-instance occlusion. When occlusion occurs with nonmotorcycle objects, the model generally remains robust and is able to correctly distinguish riders. However, for distant motorcycles, headwear such as caps or hats is occasionally misclassified as helmets due to limited spatial resolution and visual similarity. Representative failure cases are illustrated in Figure 6.

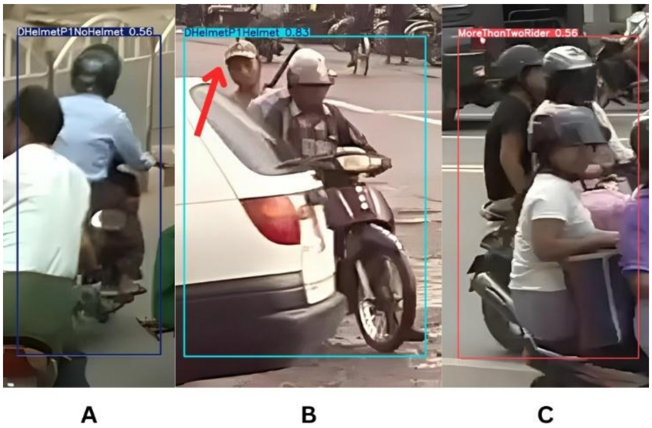

FIGURE 6. Sample failure cases of the MCVD model. Annotations A and C illustrate misclassifications in highly crowded scenes involving closely spaced motorcycles, while B shows confusion between a cap and a helmet for distant rider instances.

In summary, label consolidation, enhanced training methods, and tailored augmentations enable YOLOv11-based MCVD models to deliver superior accuracy, precision, and practical deployment readiness, outperforming earlier state-of-the-art approaches.

C. LICENSE PLATE DETECTION (LPD) MODELS

The proposed LPD YOLOv11 models were evaluated on the UFPR-ALPR dataset [5], with results summarized in Table 4.

The lightweight YOLOv11n, trained for 50 epochs, slightly outperformed YOLOv11m, which was trained for 20 epochs, likely due to the longer training duration. Both models achieved high precision, recall, F1 score, and mAP, demonstrating accurate and well-localized license plate detection. While Laroca *et al.* [5] achieved a marginally higher recall (98.33% vs. 97.89%), our models provide complete metric coverage and near-perfect localization (mAP@0.5 = 0.9910). The mAP averaged over IoU thresholds 0.5 to 0.95 (mAP@[0.5–0.95]) reached 0.8485 for YOLOv11n and 0.8295 for YOLOv11m, highlighting robust detection across varying localization criteria and benefiting subsequent LPCD tasks.

These results were obtained on an augmented UFPRALPR dataset, including synthetic regional plates (see Section III-B2), which enhanced dataset diversity and emphasized the robustness of our models.

TABLE 3. Weighted average performance for consolidated violation categories

Category	Precision	Recall	F1	Support
Non-violating	0.73	0.86	0.79	42,267
Violating	0.59	0.66	0.63	28,174

TABLE 4. Performance of License Plate Detection (LPD) models on UFPR-ALPR dataset

LPD Model	Precision	Recall	F1 Score	mAP@0.5
Laroca *et al.* [5]	-	0.9833	-	-
YOLOv11n	0.9729	0.9789	0.9759	0.9910
YOLOv11m	0.9601	0.9719	0.9660	0.9846

D. LICENSE PLATE CHARACTER DETECTION (LPCD) MODELS

The performance of the LPCD model was evaluated on both synthetic and real datasets under different training regimes, including training solely on synthetic data, training solely on real data, and the proposed two-stage fine-tuning approach. Table 5 summarizes the quantitative results.

Training exclusively on synthetic data (Experiment 1) yielded excellent performance on synthetic validation images (mAP@0.5 = 0.9948), demonstrating the effectiveness of large-scale synthetic samples for learning character features. However, evaluation on real data (Experiment 2) revealed a notable performance drop, with precision decreasing by 7.90% and recall by 14.15%, highlighting the limitations of domain shift.

Using only real samples for training (Experiment 3) improved precision on real test data (+2.98% compared to Experiment 2) but slightly decreased recall (-3.35%), indicating that limited real data captures fewer variations. The two-stage approach (Experiment 4), where a synthetically trained model was fine-tuned on just 500 real samples, achieved the best results. Precision increased by 4.19% and recall by 9.22% over Experiment 2, while mAP@0.5 improved by 4.46% and mAP@[0.5–0.95] by 14.20% (0.8813 vs. 0.7717), confirming that synthetic pretraining provides transferable features, and modest real fine-tuning effectively bridges the domain gap.

The normalized vertical variation method reliably distinguished single-line and double-line plates, supporting accurate character sequencing. Overall, the LPCD pipeline achieved 98.46% reading accuracy on

TABLE 5. LPCD Model Performance Metrics under different training experiments

Experiment	Training Data	Validation / Test Data	Precision	Recall	mAP@0.5	mAP@[0.5–0.95]
1	Synthetic Only (PS-LPCD 32k)	Synthetic Validation (8k)	0.9948	0.9921	0.9948	0.9546
2	Synthetic Only (PS-LPCD 32k)	Real Test (PR-LPCD 150)	0.9162	0.8517	0.9375	0.7717
3	Real Only (PR-LPCD 500)	Real Test (PR-LPCD 150)	0.9435	0.8232	0.8803	0.7515
4	Synthetic + Fine-tune on Real (PSLPCD 32k + PR-LPCD 500)	Real Test (PR-LPCD 150)	0.9546	0.9302	0.9793	0.8813

the 650-image PRLPCD dataset. Failures occurred mainly under extreme rotations or shearing (Figure 7), where, for example, plate C missed the last character “7” as it fell below the midline, whereas plates A and B with milder distortions were read correctly as LEC-967-17.

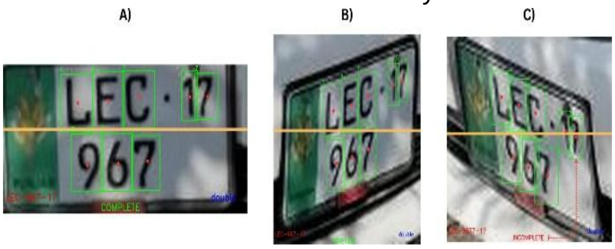

FIGURE 7. Examples of license plates demonstrating correct reading and failure cases.

Further analysis of the PR-LPCD real test set (150 plates) revealed that approximately 78% of recognition errors originated from the small-character regions inherent to Punjab license plate designs. These regions are particularly sensitive to adverse imaging conditions.

Typical failure cases include environmental degradation (LP_0019), where dust caused confusions such as ‘5’→‘S’ and ‘0’→‘U’; geometric distortion (LP_0332 and LP_0341) due to extreme viewing angles combined with low-resolution

small characters; physical wear (LP_0345), where faded printing reduced character contrast; and low-light conditions (LP_0401 and LP_0589), leading to misclassification among visually similar digits. These representative failures are illustrated in Figure 8.

The predominance of small-character-related errors suggests that robustness could be further improved through targeted synthetic augmentations (dust, blur, perspective warping) and enhanced multi-scale feature extraction.

Comparative evaluation on the 150-image PR-LPCD real test set further demonstrates the effectiveness of the proposed approach compared with EasyOCR [38] and PaddleOCR [39], the latter being based on a PP-OCR framework with a CRNN-style text recognition architecture, as shown in Table 6.

The proposed LPCD model consistently outperforms both general-purpose OCR baselines across all evaluation metrics. Compared to EasyOCR, character-level accuracy improves by 12.60% (from 87.67% to 98.72%), while fullplate match accuracy increases substantially from 33.33% to 90.67%. When compared with PaddleOCR, which demonstrates stronger character recognition performance (95.84%), the proposed method still achieves higher character-level accuracy and nearly doubles the full-plate match accuracy (49.33% to 90.67%).

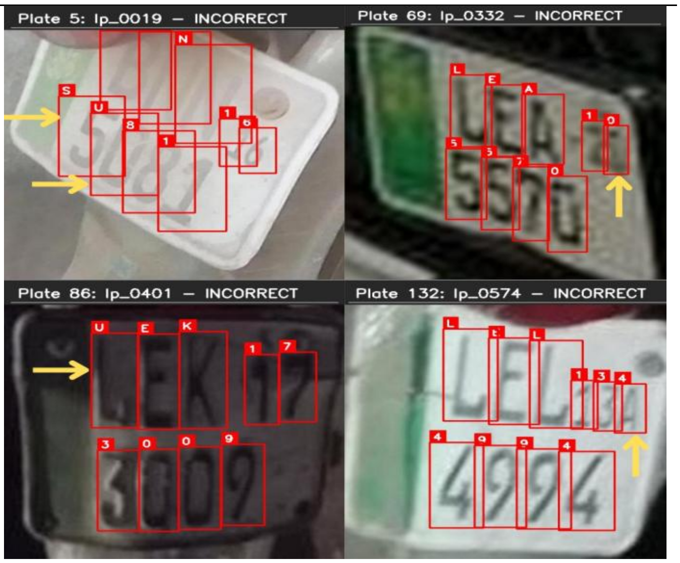
Plate 5: lp_0019 — INCORRECT
Plate 69: lp_0332 — INCORRECT
Plate 86: lp_0401 — INCORRECT
Plate 132: lp_0574 — INCORRECT

FIGURE 8. Representative failure cases from the PR-LPCD test set, highlighting typical character recognition errors. Yellow arrows indicate incorrectly predicted characters.

TABLE 6. LPCD vs. OCR Baselines Performance Comparison

Method	Character-level Accuracy	Full-Plate Match Accuracy
EasyOCR	0.8767	0.3333
PaddleOCR	0.9584	0.4933
LPCD Model	0.9872	0.9067

In addition to accuracy gains, the LPCD model demonstrates superior computational efficiency relative to EasyOCR. Total inference time on the test set is reduced by approximately 2.63× (from 53.04 s to 20.13 s), with average per-plate latency decreasing from 0.0816 s to 0.0310 s. These results demonstrate that task-specific pretraining on synthetic license plate data, combined with the proposed double-line classification heuristic, yields substantial improvements in recognition accuracy while maintaining real-time applicability in practical deployment scenarios.

E. END-TO-END PIPELINE INFERENCE

Two pipeline configurations integrating MCVD, LPD, and LPCD were evaluated on an NVIDIA RTX 2060 GPU:

- **NNN**: Nano variants for all models
- **MMN**: Medium variants for MCVD and LPD, Nano for LPCD

Each configuration was executed 5,000 times across 20 test images to estimate baseline inference speed without additional preprocessing. The NNN configuration achieved ~19 FPS, demonstrating real-time capability, while the MMN configuration achieved ~15 FPS, reflecting a trade-off between speed and accuracy.

To evaluate real-world performance, we applied the proposed sequential pipeline (see Section III-A) to five traffic surveillance videos from Lahore, Pakistan. These videos are available in our GitHub repository (https://github.com/mhdatheek136/P-LPCD). The results are summarized in Table 7.

TABLE 7. End-to-end pipeline inference results on traffic surveillance videos

Video	Pipeline FPS	Motorcycles Detected	Crowdedness
hd1	8.69	1596	1.93
hd2	8.11	3147	1.78
hd3	6.97	4181	2.78
hd4	9.17	1633	1.22
hd5	6.89	4725	2.67

The results show a strong dependency between crowdedness (average motorcycles per frame) and inference speed. With approximately one motorcycle per frame (e.g., Video hd4, crowdedness = 1.22), the pipeline maintained ~9 FPS. When crowdedness exceeded two motorcycles per frame (e.g., Videos hd3 and hd5), throughput dropped to ~7 FPS, revealing a near-linear decline in performance as object density increased.

Extrapolation suggests that under conditions of exactly one motorcycle per frame, the pipeline would sustain approximately 9.5 FPS. In contrast, at ~3 motorcycles per frame, throughput decreases by nearly 25–30%. This performance– density trade-off highlights the importance of optimizing inference strategies for deployment in dense urban environments.

V. CONCLUSION AND FUTURE WORK

The paper introduces a full YOLOv11-based motorcycle traffic violation detection pipeline, in particular, dealing with multi-rider detection, helmet-usage detection, and automatic license plate recognition (ALPR). The system is designed to suit Punjab, Pakistan, where motorcycles are a major cause of road deaths, and the levels of helmet usage among motorcycles are very low.

Key contributions of this study include the consolidation of classes for the MCVD model, achieving *mAP@50* scores of 0.66 with YOLOv11n and 0.71 with YOLOv11m, as well as strong license plate detection performance with *mAP@50* values approaching 0.99. The proposed PS-LPCD synthetic dataset, fine-tuned with the PR-LPCD real-world dataset, achieved 0.98 accuracy on real test sets. Both datasets (PSLPCD and PR-LPCD) are publicly available to support future research and to provide a reproducible procedure for creating region-specific license plate datasets, which is particularly useful in scenarios where real-world data is limited. Moreover, the normalized vertical variation technique that was proposed to differentiate between single-line and double-line plates is a lightweight alternative to the traditional deep learning methods. To provide a powerful video-based analysis, the BoT-SORT tracker was added, and the presented sequential pipeline could maintain around 9.5 FPS when it was applied to the scenes with one motorcycle per frame.

Several challenges were mitigated using task-specific augmentations, particularly the lack of near-camera close-up views in CCTV footage, which results in small and partially occluded license plates, along with varied camera perspectives and limited training data for multi-passenger cases. However, some limitations remain, such as lower efficiency when processing full-resolution images, potential identity mismatches in congested traffic, reliance on continuously updated synthetic datasets to accommodate evolving license plate formats, and difficulty in detecting small character regions on Punjab license plates under extreme conditions.

The following directions are suggested for future research:

(i) the use of regions of interest (ROIs) and rider-passenger relations modeling to enhance efficiency; (ii) the separation of nearby riders by the use of segmentation or pose estimation; (iii) the use of lightweight language models to refine the predictions of character sequences in the license plates;(iv) Controlled ablation studies to assess the impact of synthetic close-up augmentation on MCVD performance and the influence of synthetic data volume on LPCD accuracy; (v) parallelization of the ALPR process to enhance inference speed in a real-world application; and (vi) Developing specialized detection strategies for small character regions, particularly on Punjab license plates, to improve robustness under challenging conditions. These enhancements are meant to improve the strength, accuracy, and scalability of the proposed system to be used in large-scale traffic monitoring and enforcement applications.

ACKNOWLEDGMENT

We thank the contributors of the HELMET dataset [4], PK Number Plates V3 [32], and UFPR-ALPR dataset [5] for making their data publicly available for research purposes.

REFERENCES

[1] World Health Organization, *Global status report on road safety 2023: Summary*. Geneva, Switzerland: WHO, 2023. ISBN: 978-92-4-008645-6. [Online]. Available: https://www.who.int/publications/i/item/9789240086456.Accessed on: Sept. 18, 2025.

[2] A. Pervez, J. Lee, and H. Huang, “Identifying factors contributing to the motorcycle crash severity in Pakistan,” *Journal of Advanced Transportation*, vol. 2021, Art. no. 6636130, 2021, DOI: 10.1155/2021/6636130.

[3] E. A. Malhi, A. Sharif, and R. H. Javaid, “Pattern and frequency of motorcycle accident injuries reported to CMH Pannu Aqil, Sind,” *Pakistan Armed Forces Medical Journal*, vol. 67, no. Suppl-1, pp. S58–S63, 2017.

[4] H. Lin, J. D. Deng, D. Albers, and F. W. Siebert, “Helmet Use Detection of Tracked Motorcycles Using CNN-Based Multi-Task Learning,” *IEEE Access*, vol. 8, pp. 162073–162084, 2020, DOI: 10.1109/ACCESS.2020.3021357.

[5] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W.R. Schwartz, and D. Menotti, “A robust real-time automatic license plate recognition based on the YOLO detector,” in *Proc. 2018 Int. Joint Conf. Neural Netw. (IJCNN)*, 2018, pp. 1–10.

[6] J. Chiverton, “Helmet presence classification with motorcycle detection and tracking,” *IET Intelligent Transport Syst.*, vol. 6, no. 3, pp. 259–269, 2012, DOI: 10.1049/iet-its.2011.0138.

[7] R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, “Machine vision techniques for motorcycle safety helmet detection,” in *Proc. 2013 28th Int. Conf. Image and Vision Comput. New Zealand (IVCNZ)*, 2013, pp. 35–40, DOI: 10.1109/IVCNZ.2013.6726989.

[8] R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras, and A. Soares, “Automatic detection of motorcyclists without helmet,” in *Proc. 2013 XXXIX Latin American Comput. Conf. (CLEI)*,

2013, pp. 1–7, DOI: 10.1109/CLEI.2013.6670613.

[9] A. Hirota, N. H. Tiep, L. V. Khanh, and N. Oka, "Classifying Helmeted and Non-helmeted Motorcyclists," in F. Cong, A. Leung, and Q. Wei, Eds., *Advances in Neural Networks ISNN 2017*. Cham: Springer International Publishing, 2017, pp. 81–86, DOI: 10.1007/978-3-319-59072-1_10.

[10] M. Dasgupta, O. Bandyopadhyay, and S. Chatterji, "Automated Helmet Detection for Multiple Motorcycle Riders using CNN," in *Proc. 2019 IEEE Conf. Inf. Commun. Technol.*, 2019, pp. 1–4, DOI: 10.1109/CICT48419.2019.9066191.

[11] C. A. Rohith, S. A. Nair, P. S. Nair, S. Alphonsa, and N. P. John, "An Efficient Helmet Detection for MVD using Deep learning," in *Proc. 2019 3rd Int. Conf. Trends Electron. Informat. (ICOEI)*, 2019, pp. 282–286, DOI: 10.1109/ICOEI.2019.8862543.

[12] F. W. Siebert and H. Lin, "Detecting motorcycle helmet use with deep learning," *Accident Anal. Prevent.*, vol. 134, Art. no. 105319, 2020, DOI: 10.1016/j.aap.2019.105319. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S0001457519308401. Accessed on: Sept. 18, 2025.

[13] A. Chairat, M. Dailey, S. Limsoonthrakul, M. Ekpanyapong, and D. R. KC, "Low cost, high performance automatic motorcycle helmet violation detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, 2020, pp. 3560–3568.

[14] N. Kumar, G. K. Sahu, M. Ravi, S. Kumar, V. Sukruth, and A. N. M. Rao, "Triple Riding and No-Helmet Detection," in *Proc. 2023 4th IEEE Global Conf. for Advancement in Technology (GCAT)*, 2023, pp. 1–6, DOI: 10.1109/GCAT59970.2023.10353440.

[15] A. Goyal, D. Agarwal, A. Subramanian, C. V. Jawahar, R. K. Sarvadevabhatla, and R. Saluja, "Detecting, tracking and counting motorcycle rider traffic violations on unconstrained roads," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 4303–4312.

[16] T. Waris, M. Asif, M. B. Ahmad, T. Mahmood, S. Zafar, M. Shah, and A. Ayaz, "CNN-Based Automatic Helmet Violation Detection of Motorcyclists for an Intelligent Transportation System," *Mathematical Problems in Engineering*, vol. 2022, Art. no. 8246776, 2022, DOI: 10.1155/2022/8246776.

[17] A. Afzal, H. U. Draz, M. Z. Khan, and M. U. G. Khan, "Automatic helmet violation detection of motorcyclists from surveillance videos using deep learning approaches of computer vision," in *Proc. 2021 Int. Conf. Artif. Intell. (ICAI)*, 2021, pp. 252–257, DOI: 10.1109/ICAI52203.2021.9445206.

[18] M. Usama, H. Anwar, and S. Anwar, "Vehicle and license plate recognition with novel dataset for toll collection," *Pattern Anal. Appl.*, vol. 28, pp. 57, 2025, DOI: 10.1007/s10044-025-01443-8.

[19] S. U. Rehman, M. Ahmad, A. Nawaz, and T. Ali, "An efficient approach for vehicle number plate recognition in Pakistan," *The Open Artificial Intelligence Journal*, vol. 6, no. 1, pp. 12–21, 2020.

[20] J. Mistry, A. K. Misraa, M. Agarwal, A. Vyas, V. M. Chudasama, and K. P. Upla, "An automatic detection of helmeted and non-helmeted motorcyclist with license plate extraction using convolutional neural network," in *Proc. 2017 Seventh Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, 2017, pp. 1–6, DOI: 10.1109/IPTA.2017.8310092.

[21] L. Allamki, M. Panchakshari, A. Sateesha, and K. S. Pratheek, "Helmet detection using machine learning and automatic License Plate Recognition," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 6, no. 12, pp. 4475–4480, 2019.

[22] R. G. Nandhakumar, K. N. Devi, N. Krishnamoorthy, S. Shanthi, V. R. Pranesh, and S. Nikhalyaa, "Yolo Based License Plate Detection Of Triple Riders And Violators," in *Proc. 2023 Int. Conf. Comput. Commun. Informat. (ICCCI)*, 2023, pp. 1–6, DOI: 10.1109/ICCCI56745.2023.10128537.

[23] D. Anusha, D. S. Matta, D. A. Kumar, and D. Prabhakar, "Identifying Bikers Without Helmets and Triple Riding Automatically," in *Cognitive Computing and Cyber Physical Systems*. Cham: Springer Nature Switzerland, 2025, pp. 464–473, DOI: 10.1007/978-3-031-77081-4_36.

[24] R. R. Muley, G. K. Kethura, S. Lalitha, Y. Battula, and T. K. Sri, "Intelligent Traffic Monitoring System Using Deep Learning: Triple Riding, Automatic License Plate Recognition, and Helmet Detection," in *Proc. Fifth Doctoral Symp. Comput. Intell.*, Singapore: Springer Nature Singapore, 2024, pp. 391–402, DOI: 10.1007/978-981-97-6726-7_31.

[25] G. Silvano, V. Ribeiro, V. Greati, et al., "Synthetic image generation for training deep learning-based automated license plate recognition systems on the Brazilian Mercosur standard," *Designs, Automation & Embedded Systems*, vol. 25, pp. 113–133, 2021, DOI: 10.1007/s10617-020-09241-7.

[26] A. Spruck, M. Gruber, A. Maier, D. Moussa, J. Seiler, C. Riess, and A. Kaup, "Synthesizing Annotated Image and Video Data Using a Rendering-Based Pipeline for Improved License Plate Recognition," *arXiv preprint arXiv:2209.14448*, 2022. [Online]. Available: https://arxiv.org/abs/2209.14448. Accessed on: Sept. 18, 2025.

[27] M. Shpir, N. Shvai, and A. Nakib, "License Plate Images Generation with Diffusion Models," *arXiv preprint arXiv:2501.03374*, 2025. [Online]. Available: https://arxiv.org/abs/2501.03374. Accessed on: Sept. 18, 2025.

[28] M. Del Castillo Velarde and G. Velarde, "Benchmarking Algorithms for Automatic License Plate Recognition," *arXiv preprint arXiv:2203.14298*, 2022. [Online]. Available: https://arxiv.org/abs/2203.14298. Accessed on: Sept. 18, 2025.

[29] Ultralytics, "Yolo11 new," 2025. [Online]. Available: https://docs.ultralytics.com/models/yolo11/. Accessed on: Jan. 26, 2025.

[30] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," *arXiv preprint arXiv:2206.14651*, 2022. [Online]. Available: https://arxiv.org/abs/2206.14651. Accessed on: Sept. 18, 2025.

[31] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-ESRGAN: Training Real-World Blind Super-Resolution With Pure Synthetic Data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV) Workshops*, Oct. 2021, pp. 1905–1914.

[32] B. Khan, "PK-Number-Plates-V3," Roboflow Universe, 2023. [Online]. Available: https://universe.roboflow.com/burhan-khan/pk-number-plates. License: CC BY 4.0. Accessed on: Sept. 18, 2025.

[33] A. Mohamed Rafi, F. A. F. Mohamed Buhary, and A. Ishaq Khan, "Punjab Pakistan Synthetic and Real License Plate Character Datasets (P-LPCD)," *Zenodo*, ver. 1.0.0, Sept. 23, 2025, DOI: 10.5281/zenodo.17182320.

[34] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024. [Online]. Available: https://arxiv.org/abs/2410.17725. Accessed on: Sept. 18, 2025.

[35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, pp. 740–755, DOI: 10.1007/978-3-319-10602-1_48.

[36] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.

[37] N. S. Keskar and R. Socher, "Improving generalization performance by switching from adam to sgd," *arXiv preprint arXiv:1712.07628*, 2017. [Online]. Available: https://arxiv.org/abs/1712.07628. Accessed on: Sept. 18, 2025.

[38] JaidedAI, *EasyOCR: Ready-to-Use Optical Character Recognition*, GitHub repository, 2021. [Online]. Available: https://github.com/JaidedAI/EasyOCR.

[39] Y. Du, C. Li, R. Guo, S. Xu, Y. Liu, J. Wang, B. Zhou, and X. Bai, "PaddleOCR: A Lightweight and Efficient OCR System," *arXiv preprint arXiv:2009.09941*, 2020. [Online]. Available: https://arxiv.org/abs/2009.09941. Accessed on: Sept. 18, 2025.

APPENDIX A CLASS DISTRIBUTION BEFORE AND AFTER CONSOLIDATION OF THE HELMET DATASET

This appendix shows the HELMET dataset's class distribution before and after consolidation. Initially comprising 36 fine-grained classes (Figure 9), the dataset was merged and augmented into 7 broader categories (Figure 10) to reduce class imbalance.

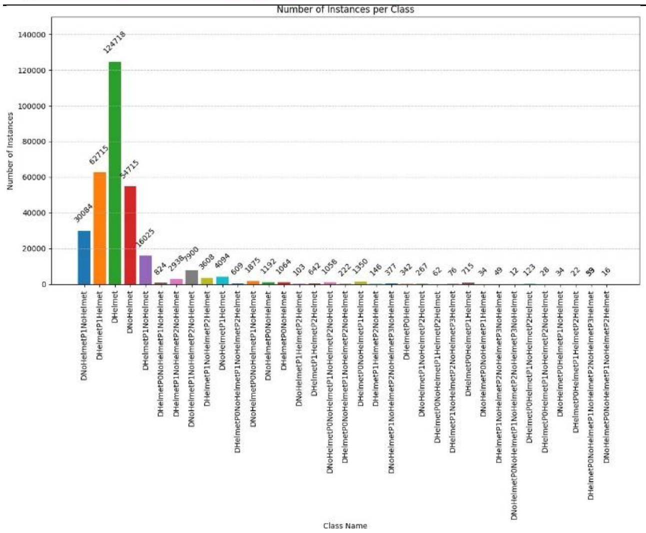

FIGURE 9. Class distribution of the HELMET dataset before consolidation (36 classes).

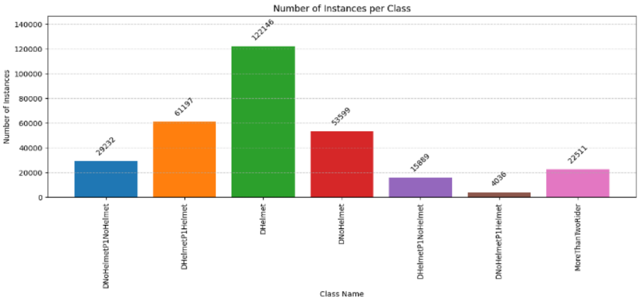

FIGURE 10. Class distribution of the HELMET dataset after consolidation and augmentation (7 classes).

APPENDIX B TRAINING AND VALIDATION CURVES

This appendix provides the full training dynamics for all YOLOv11 models used in MCVD, LPD, and LPCD experiments. Each figure shows training and validation loss curves (box, cls, dfl) along with validation mAP, precision, and recall over epochs.

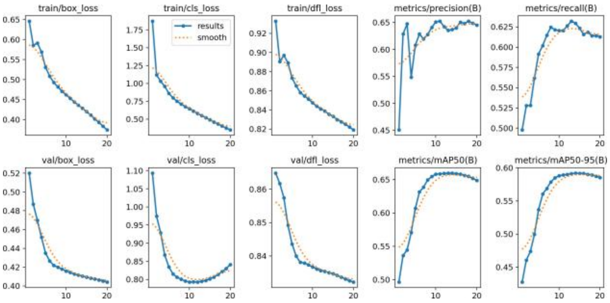

FIGURE 11. Training and validation curves for MCVD (YOLOv11n).

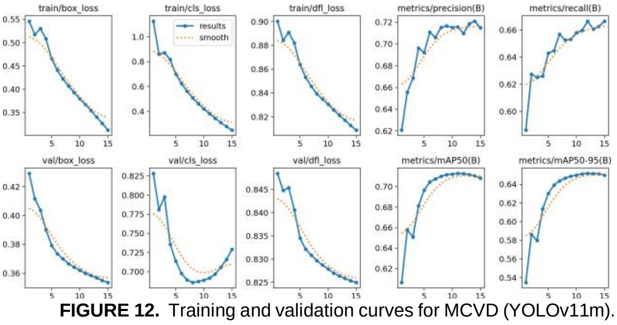

FIGURE 12. Training and validation curves for MCVD (YOLOv11m).

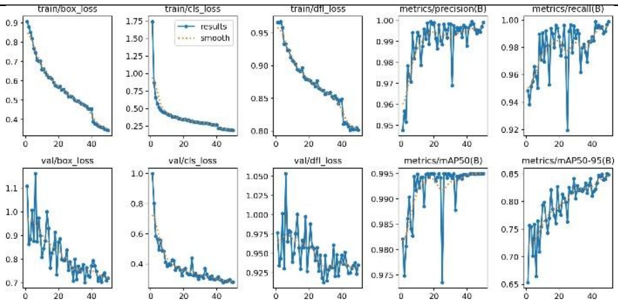

FIGURE 13. Training and validation curves for LPD (YOLOv11n).

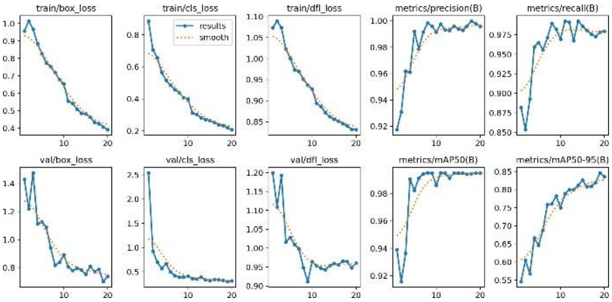

FIGURE 14. Training and validation curves for LPD (YOLOv11m).

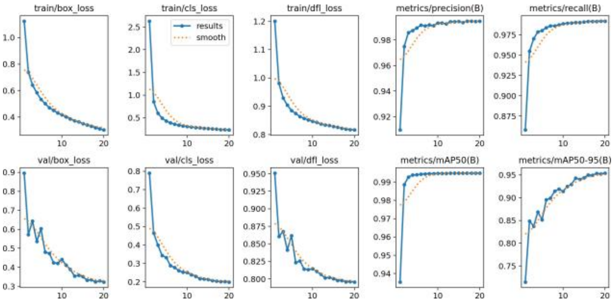

FIGURE 15. Training and validation curves for LPCD (YOLOv11n).