Proceedings of CONF-SEML 2026 Symposium: Importance of Machine Learning Methods and Analysis in Engineering
DOI: 10.54254/2755-2721/2026.AS32499

MultiPath Mobile Vision Transformer for Home-Deployed License Plate Recognition

Randy Zhu

Jericho High School, Jericho, USA
randy2020yp@gmail.com

Abstract. License plate recognition (LPR) systems have been widely used for traffic violation monitoring, stolen vehicle detection, and business security purposes; however, they are rarely installed for homeowners due to some significant challenges. Cameras are typically mounted on private property (e.g., garage pillars, lawns, or windows), observe traffic at oblique angles (≈30–40°), need to recognize license plates at long standoff distances (≈100–150 ft), and must operate on low-cost edge hardware. With these constraints, plate crops are often small, motion-blurred, and perspective-distorted. Additionally, with edge hardware usually low on memory and computation power, existing optical character recognition (OCR) solutions either do not support such settings or give low or suboptimal accuracy. This paper presents a robust and efficient OCR model that is tailored for home-deployment settings - the MultiPath Mobile Vision Transformer (MobileViT). The proposed model adopts a MobileViT backbone that combines local feature extraction through convolutional neural network (CNN) layers with global context modeling using lightweight transformer encoders. This structure is well suited to data constrained and compute-constrained settings. A MultiPath, template-aware decoding head is then used to predict each character position independently based on the plate format. In edge deployment experiments, the proposed model achieves 88.59% plate-level accuracy and less than 40 milliseconds per-plate crop inference latency, and costs only 131MB of GPU memory. Evidence also shows that accuracy can exceed 95% as training data increases. Beyond license plate recognition, the proposed architecture illustrates a general approach to structured, data-limited vision tasks operating under strict memory, latency, and power constraints on embedded hardware.

Keywords: License Plate Recognition, Optical Character Recognition, Edge AI, Mobile Vision Transformer, MobileViT, Hybrid CNN–Transformer Models, Real-Time Computer Vision

1. Introduction

Surveillance cameras have become popular security measures in residential communities to deter crimes such as burglaries and break-ins, but they cannot capture clear license plates, critical evidence to identify suspects and perform post-incident analyses.

DOI: 10.54254/2755-2721/2026.AS32499

LPR systems such as license plate readers have been seen as very popular for monitoring traffic violations, detecting stolen vehicles, and tightening security for businesses. They are often installed on street poles and cameras can capture objects through frontal-facing viewpoints with relatively close-range and controlled lighting. Additionally, they are usually bulky, expensive, and need power feed from municipal providers.

On the contrary, LPR systems are rarely seen in residential areas as cameras have to be mounted on-premises such as garages, front lawns, or windows and observe roadways from oblique viewpoints at long distances. As a result, vehicles remain within a camera's field of view for only a brief period, usually less than a second. Therefore, the camera needs to maintain at least 10-15 frames per second (FPS) to capture usable plate images. To improve coverage, homeowner-deployed LPR systems frequently rely on multiple cameras oriented in opposite directions along the roadway, as illustrated in Figure 1. Additionally, home-mounted LPR sets must be low-cost, which further forces it to run on edge hardware that has much lower memory resources and computation powers than those for municipal and business systems.

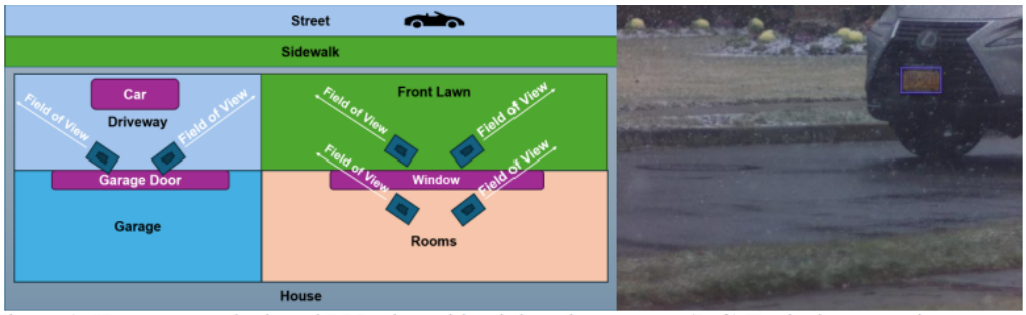

Figure 1. Homeowner-deployed LPRs in residential environments. (Left) Typical on-premises camera placement with oblique roadway views. (Right) Example license plate crop illustrating distortion, blur, lighting challenges, and narrow field of view common in residential capture

Previous studies on license plate recognition can be grouped into several distinct approaches. Transformer-based methods have been applied to both license plate detection and OCR, but detection-oriented models do not address recognition, and OCR-oriented transformer models remain data-hungry and computationally expensive. Vision large language models have recently been explored as general OCR solutions, but their high memory usage and slow inference speed limit their practicality on embedded devices. While hybrid CNN–Transformer architectures such as MobileViT have demonstrated efficient performance for edge-based vision tasks, prior work has not applied this approach to license plate OCR under the challenging conditions found in residential settings. Multi-stage CNN-based automated license plate reader (ALPR) pipelines achieve strong real-time performance, but rely on complex detection, segmentation, and recognition components and are commonly evaluated on high-end hardware.

The structure of this paper is as follows, in section 2 we review related work. In section 3 we introduce the proposed method and explain modules in detail. In section 4 we explain implementation details and training methods and compare the model's performance against previously employed models. Section 5 concludes the paper.

DOI: 10.54254/2755-2721/2026.AS32499

2. Related work

2.1. CNN-based ALPR and OCR systems

Early ALPR systems relied on CNNs arranged in multi-stage pipelines for vehicle and license plate detection and character segmentation and recognition. The spatial locality of CNNs provides a strong local inductive bias that improves robustness in vision tasks with limited training data [1]. Taking advantage of these properties, Laroca et al. [2] propose a real-time ALPR system built on YOLO-based CNN detectors, in which separate networks are trained for detection, segmentation, and recognition. The model employs majority voting across multiple frames, choosing the prediction that occurs most often and yielding strong recognition accuracy (85.45%) at real-time speeds (≈35–47 FPS) on server-class GPUs.

However, such approaches rely on multiple task-specific CNNs and explicit character segmentation, increasing model complexity. Furthermore, evaluation is performed with server-class GPUs rather than memory-constrained embedded edge devices. Similar CNN-based pipelines have been explored by Li and Chen [3] and Bulan et al. [4], where CNN features are combined with recurrent sequence models and connectionist temporal classification (CTC) decoding. Although effective, these methods also require carefully tuned multi-stage pipelines and possess substantial memory costs, hindering their practicality for homeowner-deployed systems.

2.2. Transformer-based detection and OCR approaches

Recently, there has also been consideration of transformer architectures for license plate analysis. Pal et al. [5] utilize a Swin Transformer [6] in tandem with a maximally stable extremal region (MSER)–based regional proposal mechanism [7] to detect license plate text in drone images captured at off angles and various altitudes. MSER first identifies connected regions with stable intensity extrema across thresholds, and then proposals guide the detector toward candidate text regions.

Although the model achieves strong text detection (F1 ≈ 79.8%), it only outputs bounding boxes rather than recognized plate characters, thus making it unsuitable for OCR. Moreover, inference was performed on desktop-class GPUs with a reported throughput of 7.2 FPS, which is less than the 10–15 FPS typically needed for real-time LPRs. The authors explicitly acknowledge these runtime limitations and do not evaluate performance on embedded edge hardware.

Transformer-based OCRs have also been applied directly to plate character recognition. Azadbakht et al. [8] propose MultiPath ViT OCR, which replaces recurrent decoding with independent per-position classification heads, eliminating autoregressive sequence modeling. The approach is evaluated on LicenseNet, a large-scale Persian license plate dataset, and achieved 77.25% plate-level accuracy when trained with around one million annotated images. However, experiments show significant deterioration in performance as training data size is reduced. Training and evaluation are conducted on cloud or server-class GPUs, and real-time inference on embedded edge platforms is not demonstrated. Thus, while ViT-based OCRs show promise, it's far too data-intensive and not optimized for edge devices.

2.3. Vision large language models for OCR

Large language models (LLMs) have also been tested as OCR solutions. By combining visual encoders with large autoregressive language decoders, these models are capable of flexible

DOI: 10.54254/2755-2721/2026.AS32499

recognition of text. To assess their practicality, we evaluated MiniCPM-V [9] directly on an NVIDIA Jetson Orin Nano [10]. The model was able to run on-device and achieved approximately 48% plate-level accuracy. However, the model consumed approximately 6.4 GB of GPU memory and 7.2 GB of the available 7.4 GB system memory, meaning other essential components such as detection and alerting could not run simultaneously. In addition, inference time was approximately 2–3 seconds per plate, far slower than real-time requirements. These aspects proved these types of models to be unoptimal for the task.

2.4. Hybrid CNN–transformer backbones for edge vision

MobileViT [11] is a hybrid CNN–Transformer backbone that merges local feature extraction from convolutions with lightweight self-attention applied over feature-space tokens rather than raw image patches. Through these small token sequences and reprojected context from convolutional representations, MobileViT attains a trade-off between global receptive field and computational efficiency. The model achieves great performance across classification, detection, and segmentation with real-time inference on mobile devices.

However, MobileViT is presented as a general-purpose vision backbone and is not specific to OCR. In particular, it does not address sequence decoding, format constraints, or character recognition under severe distortions, key challenges in residential LPR scenarios.

2.5. Summary

Existing approaches to license plate analysis can be broadly categorized into four groups: (i) CNN-based multi-stage ALPR pipelines that achieve real-time performance but rely on complex designs and server-class hardware; (ii) transformer-based detection models that only localize text without performing OCR and fail to meet real-time constraints; (iii) transformer-based OCR models that have the potential for great recognition performances but require massive datasets to achieve; and (iv) vision LLM–based OCR approaches that are far too memory-intensive and slow for embedded deployment.

MobileViT demonstrates that hybrid CNN–Transformer architectures can be well suited for efficient edge vision tasks, but prior work does not apply this specifically to license plate OCR with homeowner constraints. This gap motivates our approach, which combines a MobileViT backbone with a MultiPath, template-aware decoding strategy to achieve high recognition accuracy whilst under strict memory, time, and computation limits.

3. Materials & method

3.1. Materials

The proposed method operates on raw RGB license plate image cropped from vehicle video frames (1920x1080) captured from LPR cameras. The input image is resized to 128x256 pixels, and later normalized before being passed to the OCR module.

The dataset is made up of real license plate crops collected in residential settings, with wide variation in viewing angle, distance, lighting, motion blur, and surface reflections. These conditions mirror the challenges faced by homeowner-deployed LPR systems.

License plates are treated as fixed-format text strings defined by symbolic templates. Each template defines a fixed sequence of character positions, where each position is associated with

DOI: 10.54254/2755-2721/2026.AS32499

either alphabetic or numeric characters. The template design is independent of the underlying OCR architecture, enabling support for additional plate formats without retraining of the model.

The method is designed for running on low-cost, off-the-shelf edge AI hardware, which has limited memory resources especially GPU memory as well as computational capabilities. All models are implemented and trained using the PyTorch deep learning framework.

3.2. Proposed method

In this section, we present the MultiPath MobileViT OCR model. The two major components of the model architecture are the MobileViT backbone and the MultiPath template-aware decoding module. Figure 2 illustrates the end-to-end pipeline of the proposed method. Each of the four heat maps in the figure represents the magnitude of learned feature activations at a particular stage of the backbone.

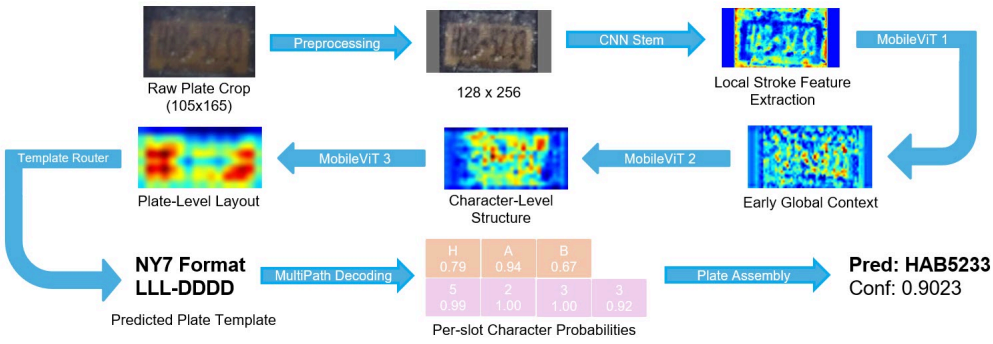

Figure 2. End-to-end pipeline of the proposed MultiPath MobileViT OCR for residential license plate recognition. A license plate crop is processed by a CNN stem and successive MobileViT blocks to extract local and global features, followed by template selection and template-aware MultiPath decoding to produce the final plate string and confidence

A raw license plate crop is first preprocessed to a fixed spatial resolution and normalized before being passed through the CNN stem. As the start of the MobileViT backbone, the CNN stem is responsible for extracting local stroke-level features that are critical for character recognition. These features then go through successive compact MobileViT blocks, which integrate global contextual information while preserving strong convolutional inductive bias. Global contextual information allows the representation to evolve to character-level structure and eventually to plate-level layout. At the same time the model still preserves local character features from the convolutional layers, as visualized in Figure 2. The final plate-level representation is aggregated via global pooling (not shown in Figure 2), which marks the end of the MobileViT backbone.

The representation is then provided to a lightweight template router, entering the MulitPath decoding module. The template router predicts the most likely license plate format (e.g. NY7: LLL-DDDD). Based on the selected template, a template-aware MultiPath decoder independently predicts each character slot under the predefined template configurations. The predicted characters and their associated probabilities are then assembled into the final license plate string along with a confidence score.

DOI: 10.54254/2755-2721/2026.AS32499

3.2.1. MobileViT backbone

MobileViT backbone f_θ performs data-efficient local–global feature extraction. It begins with a CNN stem, followed by a series of MobileViT blocks, and ends with a global average pooling. Given an input of license plate crop $x \in R^{3\times H\times W}$, the backbone produces a spatial feature map F as shown in the equation below, which encodes both local visual information and global context information.

$$F = f_\theta(x) \tag{1}$$

The CNN stem extracts visual features such as edges and character strokes. Since CNN provides strong local inductive bias at the earliest stage of the processing, subsequent transformer blocks can be compact enough to efficiently run under limited data and compute budgets.

Following the stem is a series of compact MobileViT blocks, each of which combines convolutional layers with embedded transformer encoder modules. Each MobileViT block unfolds intermediate feature maps into small patches, which are then converted into tokens. The transformer encoders operate on these patches of tokens to model long-range relationships. The transformed features are then folded back into the convolutional layers, allowing local and global information to be processed together.

At the end of the MobileViT backbone is a global average pooling (GAP) operation. It is responsible for converting the transformed feature map F into a descriptor g to be shared with downstream modules to later predict template format and to decode the individual character.

$$g = GAP(F) \tag{2}$$

As we can see, MobileViT backbone adopts a hybrid CNN-Transformer architecture. Convolutions extract local features to learn how plates look. Compact transformer blocks later efficiently model spatial relationships across the plate, enabling robust global layout representation.

3.2.2. Template-aware MultiPath decoding

License plate templates are designated as strings of characters, each one within some symbolic alphabet (L for letters and D for digits). For example, NY7 has three letters followed by four digits (LLL-DDDD) and NY5D2L has five digits followed by two letters (DDDDD-LL). For any template t, there are a fixed number of slots and each one is associated with a character set. Special characters such as hyphens are ignored by the model and are inserted during formatting. Moreover, to prevent retraining whenever a new format is added, templates are stored as lightweight configuration artifacts.

Character recognition is performed using per-slot classification heads, where each slot is predicted independently.

For each template t and the character position s (slot), we define a small classifier:

$$l_{t,s} = h_{t,s}(g) \in R^{|A_{t,s}|} \tag{3}$$

where $A_{t,s}$ is the allowed character set at position s: letters L (A-Z), digits D (0-9), or special symbols. The pooled feature representation g is produced by applying global average pooling to the

DOI: 10.54254/2755-2721/2026.AS32499

backbone feature map. The function $h_{t,s}(g)$ represents the classification head associated with slot s under template t, and its output is a vector of logits over the valid character set for that slot.

The per-slot character probabilities are:

$$p(x,t) = softmax(l_{t,s})_c \tag{4}$$

Here, x denotes the input license plate image crop, t is the predicted license plate template, and s is the character slot defined by that template. The Softmax function outputs a probability distribution over all possible characters, and $p(x,t)$ gives the probability that slot s corresponds to character c. The full plate string $y = (y_1, \cdots, y_{S_t})$ for a template t is the concatenation of these predicted slots.

All slots are evaluated in parallel, eliminating the need for autoregressive decoding. With this fixed-length decoding, the model avoids insertion and deletion errors and always produces a valid plate string consistent with the selected template. A lightweight template-aware decoder formats the predicted characters and computes an overall confidence score for the output plate.

A lightweight router head predicts template logits from g. During inference, the top-K templates (typically K = 3) are retained to account for failures caused by incorrect format selection. During training, the model minimizes a loss that combines template classification and per-slot recognition:

$$L_{total} = \lambda_{fmt} L_{fmt} + L_{slots} \tag{5}$$

where L_{fmt} is the template cross-entropy loss and L_{slots} is the sum of the cross-entropy losses across all slots for the ground-truth template. A small weighting factor (λ_{fmt} $\approx$ 0.2) prevents the router from dominating optimization once it becomes accurate. Templates with differing slot counts are padded to a maximum length, with padded positions ignored during loss computation.

Overall, the MultiPath MobileViT OCR architecture combines efficient feature extraction with structural priors and parallel decoding to produce accurate, real-time license plate recognition on resource-constrained edge devices. The design targets the requirements of home-mounted systems and accomplishes transformer-based OCR accuracy and practical edge deployment.

3.3. Datasets

3.3.1. OCR performance

The dataset consists of 35,232 real license plate crops with 5,148 unique license plate IDs, as shown in Table 1. Each plate ID may have multiple crops in the dataset, but they vary in collection dates, lighting and weather conditions, viewing angles, as well as vehicle motion states. All the plate crops were collected from three residential locations, with each location equipped with two cameras. Cameras capture traffic at approximately 100–150 feet and 30–40° viewing angles, mirroring the actual home-based installations.

The OCR model was trained exclusively on NY7 plate format. All training samples were collected from a single residential location, with no data from the other two sites. The training and validation sets are fully plate-ID disjoint. The model was evaluated on a fully held-out test set consisting of plate crops collected from two additional residential locations. These test locations differ in camera mounting height, viewing angle, lighting conditions, background, and traffic

DOI: 10.54254/2755-2721/2026.AS32499

patterns, and are fully disjoint from the training data in terms of location, time, and vehicles. Table 1 summarizes the dataset splits used for OCR performance evaluation.

Table 1. Disjoint plate-ID datasets used for OCR performance evaluation. Training and validation sets were collected from the same location while test set was collected from two other locations. All three sets do not share plate IDs

Split	Split Name	Total Images	Unique Plate IDs	Locations
Training	Train-22.8K	22,871	3,208	Residential Location A
Validation	Val-2.5K	2,574	356	Residential Location A
Test	Test-9.7K	9,787	1,584	Residential Location B & C

3.3.2. Dataset scale vs OCR accuracy

Earlier studies suggest that license plate OCR accuracy depends on the amount of training data. In MultiPath ViT-OCR [8], accuracy decreases from 77.25% with one million samples to 63.54% when the training set is reduced to 100K samples.

Although these results were obtained on Persian license plates and are not directly comparable to U.S. plates, they point to a broader pattern. Vision Transformer–based OCR models typically require large training datasets to reach competitive accuracy.

To examine whether this behavior also appears in residential, home-mounted LPR scenarios, we conducted a proof-of-concept data-scaling experiment using MultiPath MobileViT OCR. Because no additional plate images were available beyond the existing 35,232 real license plate crops, larger training sets were created by repartitioning the same dataset rather than introducing new data. The dataset splits used in this study are summarized in Table 2.

Table 2. Mixed-site plate-ID datasets used for OCR data-scaling experiment. Datasets were collected from all three locations. No datasets share the same plate IDs

Split	Split Name	Total Images	Unique Plate IDs	Locations
Training	Train-28K	28,218	4070	Mixed residential locations
Validation	Val-3.6K	3,608	539	Mixed residential locations
Test	Test-3.4K	3,406	539	Mixed residential locations

4. Experimental results

We conducted two separate experiments, OCR performance test and dataset scaling test.

In the OCR performance test, we compared the proposed method against a few OCR baselines over the same held-out test-set. Performance metrics were reported for test-set plate-level accuracy (Test plate_acc), test-set character-level accuracy (Test char_acc), GPU memory cost, and per-plate inference latency.

In the dataset scaling test, we re-split the dataset to increase the training sample size as described in section 3.3.2 and tested the plate-level accuracy on a new held-out test-set that is different from the one used in OCR performance test. This scaling test represents a proof-of-concept to observe if performance is able to scale up with the increase of training sample size.

Besides the two experiments, we also studied the impact of training data size on the plate recognition accuracy across several OCR models. Training set volumes and test-set plate accuracy

DOI: 10.54254/2755-2721/2026.AS32499

were used for the comparison study, and the metrics data were from the experimental results reported from existing work as well as our proposed method.

4.1. Implementation details

PyTorch framework [12] is applied to implement the proposed method. Model training was performed on a desktop-class GPU (NVIDIA RTX 3090), while all inference and deployment experiments were conducted on an NVIDIA Jetson Orin Nano, an edge AI computer with 7.4 GB of memory and approximately 67 tera-operations per second (TOPS) of AI compute capability. Jeston Orin is also the edge hardware used for the real home-based LPR deployments.

The OCR model was trained for 300 epochs using the AdamW optimizer [13] with a learning rate of 5×10^{-4}. Input plate crops were resized using a letterbox operation to 128 × 256 (RGB).

The MobileViT backbone uses a patch size of 2 and MobileNetV2-style inverted residual blocks (expansion ratio 4). The backbone includes three MobileViT stages with channel widths of 48, 64, and 80.

In each MobileViT block, a 3×3 convolution is used to extract local features. These features are embedded and processed by several transformer encoder layers with four-head self-attention. The transformer output is then combined with the original convolutional features.

For multi-template plates, the OCR head is divided into two parts. A small MLP (hidden size 256) predicts the plate template. Separate template-specific MultiPath slot heads are then used for character recognition. Each slot head is an MLP with a hidden size of 512 and dropout 0.1 and produces per-position character logits.

Although the model supports multiple plate templates, all experiments reported in this paper are conducted on the NY7 format (LLL-DDDD) as we lack plate samples from other formats. As a result, the accuracy of predicting plate format in our experiments is always 100%, and the total loss is the sum of cross-entropy losses across all slot heads. Experiments on additional plate formats are deferred to future work when we collect sufficient samples.

4.2. Results

4.2.1. OCR performance evaluation and comparison

We evaluate OCR performance on an NVIDIA Jetson Orin Nano edge computer across three models. Accuracy is measured as top-1 percentage value on the same held-out test-set. Inference latency is measured as the median per-plate crop duration time over multiple runs, and memory usage corresponds to peak GPU resident memory during inference.

We compare the proposed method against a vision LLM model and a CNN model. For the vision LLM model, we use a pre-trained model, MiniCPM-V [9], as described in section 2. For the CNN model, we implemented one using a MobileNet-style convolutional backbone. Table 3 summarizes OCR accuracy, inference latency, and memory usage for all models under identical test conditions.

DOI: 10.54254/2755-2721/2026.AS32499

Table 3. Results of plate-level and character-level recognition accuracy, computational time required for per-plate crop, and GPU memory cost across three OCR models. Vision LLM OCR [9] is a pre-trained model and it doesn't publish character-level accuracy and dataset volume

Model	Test plate_acc (%)	Test char_acc (%)	Dataset (images)	GPU Memory (GB)	Inference Speed (ms)
MP MobileViT OCR	88.59	98.38	35,232	0.13	39.0
CNN OCR Baseline	73.08	95.76	35,232	0.82	57.6
Vision LLM OCR	47.9	-	-	6.4	2619.5

As shown in Table 3, the proposed method achieves higher plate-level accuracy compared to both the vision LLM and CNN models. It also costs significantly less GPU memory and delivers much faster inference speed than the vision LLM model. Its efficiency on resource consumption allows the system to allocate sufficient memory and computation powers to run other critical programs such as live video feed, license plate detection, as well as notifications, along with recognition models, enabling end-to-end real time performance for LPR systems running in edge hardware.

The CNN baseline underperforms primarily because purely convolutional features lack an explicit mechanism for modeling global character relationships and plate-level structure, making them brittle under perspective distortion, long-range capture, and variable spacing common in home-mounted LPR scenarios. In contrast, the MobileViT backbone integrates lightweight global context through transformer blocks, enabling more robust character disambiguation while maintaining efficiency suitable for edge deployment.

4.2.2. Dataset scale vs OCR accuracy

By repartitioning existing dataset as described in Section 3.3.2, we increased the number of training samples from 22.8K to 28K and re-trained the model. On the held-out test-set (Test-3.4K) the re-trained model achieved a plate-level accuracy of 95.36%, a significant gain over the 88.59% accuracy from the model evaluated in the cross-site OCR performance test (section 3.3.1).

This experiment shows that performance is limited by data availability rather than model architecture. Increasing sample data, even a relatively modest size, can lead to clear gains in plate-level accuracy, without changing the model architecture or adding computational cost.

To further validate the concept, we plan a controlled follow-up evaluation as part of the future work. Similar to what we did for the cross-site performance evaluation, additional training samples will be collected from a single residential location to expand the training set beyond 30K images. The validation set comes from the same location as the training set and test-set are collected from the other two sites. Training, validation, and test sets are maintained plate-ID disjoint as usual. This setup enables a direct measurement of accuracy gains attributable solely to increased training samples.

4.2.3. MobileViT vs standard vision transformer

While the data scaling experiment discussed in the previous section was not designed as a direct benchmark comparison against the existing work [8], we observe that the proposed method achieves substantially higher plate-level accuracy (~95% vs ~77%) by using only tens of thousands of training samples than the standard vision transformer–based OCR approaches, which was trained on

DOI: 10.54254/2755-2721/2026.AS32499

hundreds of thousands to millions of images. It is worth noting that the plate datasets in both experiments are different in language, plate design, and evaluation protocol; nevertheless, the magnitude of this contrast is striking. Table 4 contextualizes this observation between MobileViT and standard ViT.

Table 4. Observations of the differences in dataset volume and plate-level accuracy between MultiPath MobileViT OCR and standard ViT OCR [8]

Model	Training Set Volume	Test-set Plate Acc (%)
MP MobileViT OCR	28K	95.36
ViT OCR	1 million	77.25

The big gaps of data efficiency and accuracy between the two models, as seen in Table 4, indicates that the proposed method benefits from the hybrid CNN–Transformer architecture. The MobileViT OCR method blends CNN and Transformer together. This design allows the model to extract local features (through convolutional layers) and capture global visual context (through lightweight transformers) at the same time, enabling the model to learn effectively from much smaller datasets.

5. Conclusion

In this paper, we demonstrated that with a hybrid CNN-Transformer architecture, combined with a template-aware MultiPath decoding strategy, we could achieve a very competitive plate-level recognition accuracy for edge hardware-based LPR systems to be deployed in residential communities, where an LPR device has to operate under oblique viewing angles, long standoff distances, and strict memory and latency constraints.

It is also worth mentioning that the proposed method relies on much smaller datasets than existing approaches and could further improve performance with an increase in the number of more training samples, even at modest scale. This data-efficiency and edge-friendly approach could be expanded to other vision tasks beyond just license plate recognitions.

For future work, we plan to (i) evaluate performance gains in relationships with more training samples introduced and retrain models through cross-site generalization test; (ii) expand plate template supports to more formats including out-of-state, conduct performance test on new formats without retaining backbone model, hence evaluate the effectiveness of model's modularity and deployment flexibility.

References

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.https: //www.deeplearningbook.org/

[2] Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018). A Robust Real-Time Automatic License Plate Recognition based on the YOLO Detector. arXiv preprint arXiv: 1802.09567v5.

[3] Li, H., & Chen, C. (2016). Reading car license plates using deep convolutional neural networks and LSTMs. arXiv preprint arXiv: 1601.05610.

[4] Bulan, O., Kozitsky, V., Ramesh, P., & Shreve, M. (2017). Segmentation- and annotation-free license plate recognition with deep localization and failure identification. IEEE Transactions on Intelligent Transportation Systems, 18(9), 2351–2363. DOI: 10.1109/TITS.2016.2639020.

[5] Pal, S., Roy, A., Shivakumara, P., & Pal, U. (2023). Adapting a Swin transformer for license plate number and text detection in drone images. Artificial Intelligence and Applications, 1(3), 129–138. DOI: 10.47852/bonviewAIA3202549.

DOI: 10.54254/2755-2721/2026.AS32499

[6] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. 2021 IEEE/CVF International Conference on Computer Vision, 9992-10002. DOI: 10.1109/ICCV48922.2021.00986.
[7] Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. Image and Vision Computing, 22(10), 761–767. DOI: 10.1016/j.imavis.2004.02.006.
[8] Azadbakht, A., Kheradpisheh, S. R., & Farahani, H. (2022). MultiPath ViT OCR: A Lightweight Visual Transformer-based License Plate Optical Character Recognition. 12th International Conference on Computer and Knowledge Engineering (ICCKE), 92–95. DOI: 10.1109/ICCKE57176.2022.9960026.
[9] OpenBMB. (n.d.). MiniCPM-V. GitHub. https: //github.com/OpenBMB/MiniCPM-V/
[10] NVIDIA. (n.d.). Jetson Orin Nano Super Developer Kit. Nvidia Marketplace. https: //marketplace.nvidia.com/en-us/enterprise/robotics-edge/jetson-orin-nano-super-developer-kit/
[11] Mehta, S., & Rastegari, M. (2022). MOBILEVIT: LIGHT-WEIGHT, GENERAL-PURPOSE, AND MOBILE-FRIENDLY VISION TRANSFORMER. Proceedings of the International Conference on Learning Representations (ICLR). arXiv: 2110.02178v2.
[12] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv: 1912.01703v1.
[13] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv. DOI: 1711.05101v3