Journal of Computer and Communications, 2025, 13(11), 236-254
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Unified Cross-Domain Adaptation for License Plate Recognition in Adverse and Multilingual Environments

Bertrand Tahte[1], Geh Wilson Ejuh[2], Thierry Noulamo[3], Alain Djimeli[4], Armand Tchanque Tchakountio[1], Josiane Tanguebou Kengne[5]

[1]Department of Computer Science, University of Dschang, Dschang, Cameroon
[2]Department of General and Scientific Education, University Institute of Technology, Bandjoun, Cameroon
[3]Department of Computer Engineering, University Institute of Technology, Bandjoun, Cameroon
[4]Department of Telecommunication and Network Engineering, University Institute of Technology, Bandjoun, Cameroon
[5]Cartographic Production Division, Data Management Department, National Institute of Cartography, Yaoundé, Cameroon
Email: tahtebertrand2@gmail.com, thierry.noulamo@gmail.com, gehwilsonejuh@yahoo.fr, adjimeli@yahoo.fr, tchanquitos1@gmail.com, kengnejosyange@gmail.com

How to cite this paper: Tahte, B., Ejuh, G.W., Noulamo, T., Djimeli, A., Tchakountio, A.T. and Kengne, J.T. (2025) Unified Cross-Domain Adaptation for License Plate Recognition in Adverse and Multilingual Environments. *Journal of Computer and Communications*, **13**, 236-254.
https://doi.org/10.4236/jcc.2025.1311015

Received: August 1, 2025
Accepted: November 24, 2025
Published: November 27, 2025

Abstract

License Plate Recognition (LPR) systems face significant challenges when deployed in real-world environments that differ from their training data, due to variations in angle, resolution, lighting, plate syntax, and language. This paper presents a unified, robust, and multilingual LPR framework designed to operate effectively in both intra-domain and cross-domain scenarios. The proposed architecture comprises three key modules: (i) an oriented license plate detector based on YOLOv8 with angle regression, enabling accurate detection under oblique or rotated views; (ii) a Transformer-based recognition head enhanced with CBAM and constrained decoding using region-specific syntax; and (iii) a lightweight yet effective domain adaptation module leveraging Maximum Mean Discrepancy (MMD), adversarial training, and test-time self-supervised fine-tuning. Extensive experiments across real-world and synthetic datasets (CCPD, AOLP, UFPR, PKU, LP-Synth) demonstrate that our model outperforms existing state-of-the-art methods, especially in multilingual and cross-domain conditions. Ablation studies further confirm the importance of each architectural component. Our framework opens promising directions for deploying LPR systems in globally diverse and dynamic environments.

Keywords

License Plate Recognition, Cross-Domain Adaptation, Multilingual OCR, Oriented Object Detection, Transformer Architecture, Domain

Generalization, YOLOv8-OBB, CBAM, Test-Time Adaptation

1. Introduction

License Plate Recognition (LPR) has become a critical component of intelligent transportation systems, enabling a wide range of applications such as vehicle tracking, automated toll collection, law enforcement, and access control. While early LPR systems relied on rigid, rule-based pipelines, recent advances in deep learning have enabled highly accurate end-to-end models [1] [2]. However, despite these improvements, most existing approaches are trained and evaluated under controlled conditions-typically limited to specific countries, fixed plate formats, consistent lighting, and frontal views. These assumptions rarely hold in real-world scenarios.

In practical deployments, LPR systems often face challenging environments and unseen domains: license plates may appear at oblique angles, suffer from low resolution or motion blur, follow diverse regional syntaxes, or contain multilingual characters (e.g., Arabic, Cyrillic, Latin, Thai) [3] [4]. These cross-domain and multilingual variations lead to significant performance degradation for models trained on narrow data distributions. Furthermore, most current methods lack mechanisms for dynamic adaptation, rendering them ineffective in rapidly changing or previously unseen environments [5].

Several recent studies have addressed partial aspects of this problem. Approaches such as YOLOv5-OBB [6] and HPR-Net [7] have improved detection under rotation and perspective distortion, while UR-LPR Net [8] has explored robust recognition using attention mechanisms. Nevertheless, these methods are often constrained to a single language, lack domain generalization capabilities, or depend on fixed-format decoding schemes. Therefore, there is a clear need for a unified, robust, and adaptive LPR framework capable of generalizing across languages, domains, and varying imaging conditions.

In this work, we propose a novel hybrid framework for robust License Plate Recognition in adverse and multilingual environments. Our approach integrates the following components:

- An Oriented Bounding Box (OBB) detector based on YOLOv8, featuring direct angle regression to accurately localize license plates under oblique or distorted viewpoints;
- A Transformer-based recognition module, enhanced with Convolutional Block Attention Module (CBAM) for spatial attention [9], and equipped with region-specific syntax constraints during decoding;
- A lightweight domain adaptation module that combines Maximum Mean Discrepancy (MMD) [10], adversarial training [11], and test-time fine-tuning [12] to reduce distribution shifts across domains and languages.

We evaluate our framework on five diverse datasets-CCPD, AOLP, UFPR,

PKU, and LP-Synth [13]-[15]-encompassing both real-world and synthetic data, and covering license plates in Latin, Cyrillic, and Arabic scripts. Experimental results demonstrate that our system consistently outperforms existing baselines, especially in cross-domain and multilingual settings. Ablation studies further validate the effectiveness of each component in enhancing overall performance and robustness. In summary, our key contributions are:

1. A robust OBB detection module using YOLOv8 with angle-aware loss for accurate localization of rotated and skewed plates;

2. A multilingual Transformer-based decoder with syntax-constrained decoding and attention regularization;

3. A unified and modular domain adaptation strategy enabling dynamic adjustment to unseen domains and conditions;

4. Comprehensive experimental validation across multiple real and synthetic benchmarks, demonstrating strong generalization.

The remainder of this paper is structured as follows. Section 2 reviews related work in oriented object detection, character recognition, domain adaptation, and multilingual LPR. Section 3 presents the proposed methodology in detail. Section 4 describes the experimental setup, followed by Section 5, which reports the results and ablation studies. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2. Related Work

License Plate Recognition (LPR) has undergone significant evolution in recent years, transitioning from handcrafted, modular pipelines to end-to-end deep learning frameworks. This section reviews the state of the art along four key dimensions relevant to our proposed approach: (i) oriented license plate detection, (ii) robust recognition architectures, (iii) cross-domain adaptation, and (iv) multilingual and low-quality plate recognition.

2.1. Oriented License Plate Detection

Early LPR systems relied on conventional object detectors such as Faster R-CNN and SSD, which assume axis-aligned bounding boxes. However, this assumption is often violated in real-world scenarios where license plates appear tilted, rotated, or partially occluded due to perspective distortion. To overcome this limitation, oriented bounding box (OBB) detection has been introduced to better capture the spatial orientation of plates.

HPR-Net [16] proposed a two-stage framework that combines YOLOv5-OBB for oriented detection with a Transformer-based recognizer for character sequence decoding. Similarly, recent YOLOv8-OBB variants [6] enhance angular accuracy through direct angle regression and hybrid loss functions, enabling precise localization of plates under complex viewpoints. These improvements significantly boost the performance of downstream recognition modules by providing more accurate region proposals.

Despite their effectiveness in handling geometric distortions, these methods are typically designed for specific regions and lack mechanisms for cross-domain generalization or multilingual support. This limitation motivates the development of more flexible and adaptive architectures, such as the one proposed in this work.

2.2. Robust Recognition Architectures

The second stage of LPR focuses on recognizing characters from detected plate regions, often under challenging conditions such as motion blur, low resolution, contrast variations, or partial occlusion. Early approaches were dominated by Convolutional Recurrent Neural Networks (CRNN) [17], which combine CNN-based feature extractors with RNNs and Connectionist Temporal Classification (CTC) for sequence prediction.

More recent methods have incorporated attention mechanisms to improve recognition robustness. For instance, UR-LPR [18] integrates EfficientNet-B4 with the Convolutional Block Attention Module (CBAM) [9] to strengthen spatial feature representation, yielding improved performance under degraded visual conditions. Other works have explored Vision Transformers (ViT) [19] and Swin Transformers [20], leveraging their global receptive fields and ability to model long-range dependencies in sequential character patterns.

Our approach advances this line of research by employing a Transformer-based recognition module enhanced with attention regularization and constrained decoding based on region-specific syntax rules, enabling higher accuracy and better generalization across diverse plate formats.

2.3. Cross-Domain and Environment Adaptation

Most existing LPR models are trained and evaluated on datasets from specific geographic regions (e.g., CCPD [13], AOLP, UFPR), assuming consistent environmental and design characteristics. However, real-world deployment requires generalization across varying conditions-including lighting, weather, camera angles, plate colors, fonts, and regional formatting rules-which introduces significant domain shifts.

To mitigate such discrepancies, domain adaptation techniques have been explored in related vision tasks. Methods such as Maximum Mean Discrepancy (MMD) [21], adversarial domain adaptation (e.g., ADDA [22]), and test-time adaptation (e.g., TENT [23]) have shown effectiveness in aligning feature distributions across domains. However, their application in LPR remains limited.

While UR-LPR demonstrates implicit robustness through end-to-end training, it does not explicitly address domain shift. In contrast, our work introduces an explicit, lightweight domain adaptation module that combines MMD, adversarial training, and test-time fine-tuning to dynamically adapt to unseen environments, improving model robustness without requiring extensive retraining.

2.4. Multilingual and Real-World Plate Recognition

In global deployments, LPR systems must handle license plates written in multiple

scripts-such as Arabic, Cyrillic, Latin, and Thai-as well as non-standard formats including vertical, double-line, or irregular layouts. However, the majority of current systems are restricted to Latin-character plates, limiting their applicability in multilingual regions.

Some efforts have aimed at expanding character vocabularies to support mixed-script recognition [24], but comprehensive multilingual LPR remains an open challenge. Additionally, real-world images are often degraded by adverse weather, motion blur, low illumination, or compression artifacts, further complicating recognition.

To address these issues, synthetic data augmentation techniques-such as simulating rain, fog, JPEG compression, or noise injection-have been employed to improve model robustness [25].

Our approach addresses both challenges by introducing a syntax-guided, vocabulary-aware decoding mechanism that adapts to country-specific plate structures and supports cross-lingual recognition. Combined with synthetic augmentation and attention-enhanced feature learning, our framework achieves reliable performance even under low-quality imaging conditions and diverse linguistic contexts.

3. Proposed Methodology

Our proposed framework is designed to robustly detect and recognize license plates in multilingual and cross-domain scenarios, especially under real-world conditions involving oblique views, variable lighting, and degraded image quality. The architecture is composed of three key modules: (1) an oriented detection backbone based on YOLOv8-OBB with angular regression, (2) a Transformer-based constrained recognizer, and (3) an adaptive domain alignment module incorporating distributional matching and attention regularization. **Figure 1** provides an overview of the complete pipeline.

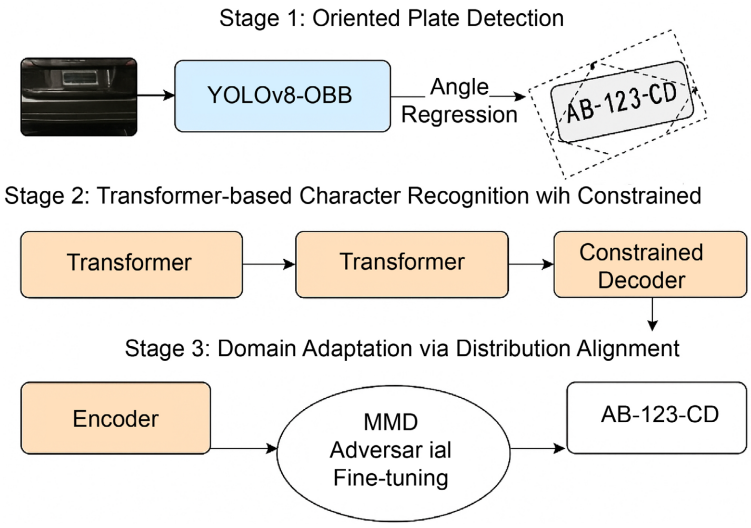

Figure 1. Proposed model.

3.1. Oriented License Plate Detection via YOLOv8-OBB and Angle Regression

In license plate recognition (LPR) systems, detection is a critical step often compromised by non-ideal visual conditions. In real-world environments, plates may appear tilted, partially occluded, distorted, or captured at oblique angles. Traditional detectors based on axis-aligned bounding boxes generally fail to accurately locate these distorted objects. To overcome these limitations, our method relies on an oriented extension of the YOLOv8 detector, where each plate instance is represented by an oriented bounding box (OBB) enhanced by an angle regression branch.

Each predicted box is parameterized by five elements:

$$B = \left(x_c, y_c, w, h, \theta\right)$$

where $\left(x_c, y_c\right)$ are the coordinates of the box center, w and h are its width and height, respectively, and θ is the orientation angle (in radians) defined in the interval $\left[-\pi, \pi\right]$. This formalism allows for a more geometrically faithful representation of tilted or distorted plates, significantly improving subsequent recognition performance.

The base architecture relies on YOLOv8 for its execution speed and ability to extract efficient visual representations. We extend the standard prediction head to incorporate a dedicated channel for angle θ, in addition to the usual outputs related to localization and classification. This channel is trained using an angular loss function tailored for periodic variables. Specifically, the loss

$$L_\theta = \min\left(\left(\hat{\theta} - \theta\right)^2, \left(2\pi - \left|\hat{\theta} - \theta\right|\right)^2\right)$$

avoids discontinuities during learning at the boundary $-\pi/+\pi$.

The overall training of the detector is guided by a hybrid cost function combining three main components:

$$L_{\text{det}} = \lambda_{\text{cls}} \cdot L_{\text{cls}} + \lambda_{\text{IoU}} \cdot L_{\text{IoU}} + \lambda_\theta \cdot L_\theta$$

where L_{cls} is the classification loss (binary cross-entropy), L_{IoU} is the localization loss based on intersection over union (IoU), and L_θ is the angular regression loss. The coefficients $\lambda_{\text{cls}}, \lambda_{\text{IoU}}, \lambda_\theta$ allow for the weighting of these terms for balanced learning.

To enhance the robustness of the detector to geometric and photometric variations, we have integrated a targeted data augmentation strategy during training. This includes random affine transformations (rotations from 0˚ to 180˚), partial cutouts, Gaussian blur, lighting alterations, and the use of a synthetic rendering engine (Unreal Engine) to generate plates in rare or complex positions. This approach enables the model to better generalize to out-of-distribution scenarios.

During inference, the model outputs are projected as oriented polygons, calculated from $\left(x_c, y_c, w, h, \theta\right)$. A geometric transformation module then extracts a rectified region (aligned crop) through rotation, which is passed to the recognition

module. This explicit separation between oriented detection and recognition allows for better modularity while optimizing the overall pipeline accuracy.

This initial step ensures robust localization of plates under varied conditions while preparing a clean encoding for the recognition stage. The choice of YOLOv8-OBB, combined with direct angle regression and multi-component supervision, constitutes an effective foundation for complex visual contexts.

3.2. Stage 2: Transformer-Based Character Recognition with Constrained Decoding

Once the plate is detected and cropped (rectified) by the oriented detection module, the next task is to recognize the sequence of characters it contains. This step is complicated by many factors: motion blur, low resolution, uneven lighting, partial occlusion, and especially linguistic variability (different alphabets, national formats, etc.). To address this, our architecture relies on a two-stage recognition module combining a visual Transformer encoder, enhanced by adaptive spatial attention (CBAM), and sequential decoding under syntactic constraints based on the plate formats specific to each country or region. Figure 1 provides an overview of the complete pipeline. The overall architecture of the Transformer-based recognition module is illustrated in Figure 2.

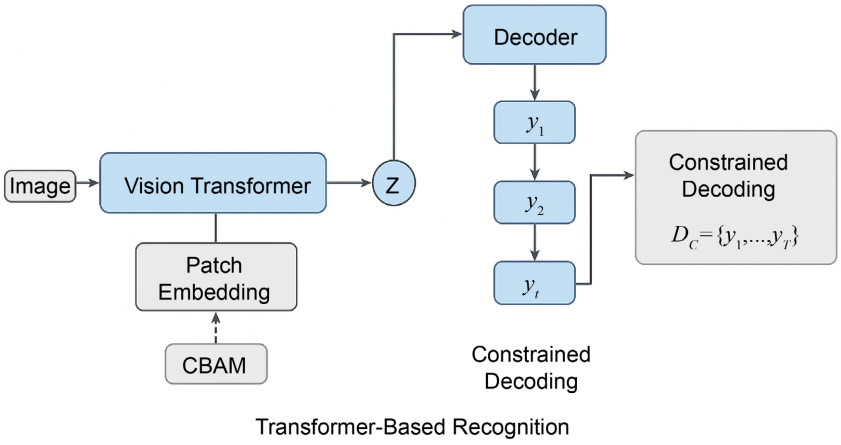

Figure 2. Transformer-based recognition.

3.2.1. Visual Encoder with Transformer and CBAM

The visual encoder uses a Vision Transformer (ViT) architecture. Unlike traditional CNNs or CRNNs, Transformers can model long-range dependencies between different parts of the image, which is crucial for correctly interpreting distorted or partially visible characters.

The rectified plate image, of size $I \in \mathbb{R}^{H \times W \times 3}$, is first divided into small non-overlapping patches (e.g., 16 × 16 pixels), each projected linearly into a vector space of dimension d by an embedding layer E. To this sequence of vectors, we add a positional encoding E_{pos} to preserve the spatial structure of the image:

$$Z_0 = \left[x_{\text{cls}}; x_{p_1} E; x_{p_2} E; \cdots; x_{p_N} E \right] + E_{\text{pos}}$$

where x_{cls} is a classification token (optional), x_{p_i} represents the i-th patch, and $N = \frac{HW}{P^2}$ is the total number of patches.

To increase the encoder's robustness against perturbations (blur, noise, low contrast), we integrate adaptive spatial attention through CBAM (Convolutional Block Attention Module). This module dynamically refines feature responses by focusing on the most informative regions and channels. In practice, CBAM is inserted after the patch embedding step or between Transformer blocks to guide attention towards relevant characters.

3.2.2. Syntactically Constrained Sequential Decoding

The goal of the decoder is to produce an ordered sequence of characters $(y_1, y_2, \cdots, y_T)$ corresponding to the content of the plate. Instead of using naive decoding, our method leverages a constrained dictionary D_c specifying syntactically valid character combinations based on the country or region.

Decoding is performed using an autoregressive Transformer decoder, which generates one character at a time, conditioning each prediction on the previous output and the representations extracted by the encoder. At each step t, the prediction is defined as:

$$y_t = \arg\max_{c \in A_c(t)} P(y_t = c \mid y_{<t}, Z)$$

where $A_c(t) \subset D_c$ is the restricted set of valid characters at step t, and Z is the contextual representation of the image from the encoder. This constraint injects prior knowledge about formats (e.g., 2 letters followed by 4 digits), reducing common semantic or syntactic errors in CTC-type methods.

The total vocabulary used consists of the union of several alphabets: Latin, Cyrillic, Arabic, Thai, etc., as well as common digits and symbols. A dynamic masking is applied to the vocabulary at each decoding step to respect the rules defined in D_c.

The syntactic constraints for license plate formats are explicitly encoded as deterministic rules rather than learned patterns. For each supported country/region, we have manually curated a formal specification of valid plate formats based on official government documentation. These specifications define the exact sequence of character types (letters, digits, special symbols), regional codes, and positional constraints. During inference, the function GetCountrySyntax (Pi) (Algorithm 0.0.6, line 7) dynamically selects the appropriate constraint dictionary based on either: (a) explicit country identification from the input image, or (b) the domain adaptation module's prediction of the most probable region. The constrained decoding mechanism then applies these rules through a dynamic vocabulary masking approach, where at each decoding step t, only characters compliant with the current position's syntax rules remain unmasked in the prediction space. This implementation approach was chosen deliberately over a learned syntax model for three key reasons: (1) license plate formats are strictly defined by governmental authorities with zero tolerance for variation; (2) manual encoding en-

sures 100% compliance with official standards; and (3) it eliminates the need for extensive region-specific training data that would be required for a learned approach. The constrained dictionary Dc is implemented as a finite-state machine where each state represents a valid position in the plate sequence, and transitions are permitted only to characters that satisfy the regional syntax rules.

3.2.3. Training and Optimization

The encoder-decoder pair is trained end-to-end, with a cross-entropy loss between the predicted sequence and the target sequence:

$$L_{\text{rec}} = -\sum_{t=1}^{T} \log P\left(y_t^{\text{true}} \mid y_{<t}^{\text{true}}, Z\right)$$

This loss is combined with an attention regularization (if activated), which encourages the model to focus its attention weights on the expected textual areas.

Teacher forcing, character masking, and attention dropout strategies are also employed to stabilize learning and prevent overfitting.

3.3. Stage 3: Domain Adaptation via Distribution Alignment

License plate recognition (LPR) systems typically suffer from a strong domain bias: when trained on a specific dataset (e.g., CCPD, UFPR), their performance drops drastically on data from other geographical or environmental contexts. To address this out-of-distribution fragility, we propose a lightweight yet effective domain adaptation module based on three strategies: (i) distribution alignment via Maximum Mean Discrepancy (MMD), (ii) adaptation through adversarial learning, and (iii) self-supervised fine-tuning during testing.

3.3.1. Alignment via Maximum Mean Discrepancy (MMD)

MMD allows reducing the statistical divergence between the source domain $\mathcal{S}$ and the target domain $\mathcal{T}$. Let $\phi(x)$ be a projection function into a Hilbert space (RKHS), the loss is defined as:

$$\mathcal{L}_{\text{MMD}} = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi\left(x_i^s\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi\left(x_j^t\right) \right\|^2 \tag{1}$$

This function is added to the training phase of the backbone to encourage convergence towards visually and statistically coherent representations.

3.3.2. Adversarial Adaptation

We also employ an adversarial strategy to learn domain-invariant representations. A discriminator D is trained to distinguish between the source and target domains while the encoder G learns to make them indistinguishable. This leads to the following loss game:

$$\mathcal{L}_D = -\mathbb{E}_{x \in \mathcal{S}} \log D\left(G(x)\right) - \mathbb{E}_{x \in \mathcal{T}} \log\left(1 - D\left(G(x)\right)\right) \tag{2}$$

$$\mathcal{L}_{\text{adv}} = -\mathcal{L}_D \tag{3}$$

A gradient reversal mechanism is used to train the encoder jointly with the dis-

criminator.

3.3.3. Online Self-Supervised Adaptation (Test-Time)

To adapt to gradual distribution drifts, we propose a test-time adaptation strategy without explicit supervision. It combines entropy minimization:

$$\mathcal{L}_{\text{ent}} = \sum_{i=1}^{N} H\left(P\left(y_i \mid x_i\right)\right) = -\sum_{i=1}^{N}\sum_{c} P\left(y_i = c \mid x_i\right)\log P\left(y_i = c \mid x_i\right) \tag{4}$$

and temporal coherence regularization:

$$\mathcal{L}_{\text{cons}} = \text{KL}\left(P^{(t)} \,\|\, P^{(t-1)}\right) \tag{5}$$

These losses are utilized during testing to refine representations without relying on real labels. The total loss used during training is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{rec}} + \lambda_{\text{mmd}} \cdot \mathcal{L}_{\text{MMD}} + \lambda_{\text{adv}} \cdot \mathcal{L}_{\text{adv}} + \lambda_{\text{tt}} \cdot \left(\mathcal{L}_{\text{ent}} + \mathcal{L}_{\text{cons}}\right) \tag{6}$$

where the terms λ_i allow for dynamically weighting the importance of each component according to the learning scenario.

To better illustrate the sequential functioning of our architecture, Algorithm 0.0.6 presents the entire inference pipeline proposed for robust multilingual license plate recognition in multi-domain environments. The process is organized into three main stages: (i) oriented plate detection using a YOLOv8 detector modified to produce bounding boxes with explicit orientation, (ii) extraction and recognition of characters using a visual Transformer encoder enriched by spatial attention (CBAM), coupled with a syntactically constrained decoder, and (iii) a domain adaptation module, activated during training or online, which dynamically adjusts representations to compensate for inter-domain divergences. This pseudocode highlights the various functions involved, as well as the mechanisms for conditional adaptive regularization.

Algorithm 1 Inference Pipeline for Cross-Domain and Multilingual License Plate Recognition

Require: Input image I
Ensure: Predicted license plate string $\hat{Y}$

// **Stage 1: Oriented Plate Detection**
$B = \text{YOLOv8\_OBB}(I)$ {Detect oriented bounding boxes $B = \{x_c, y_c, w, h, \theta\}$}
for all $b_i \in B$ **do**
  $P_i \leftarrow \text{RectifyCrop}(I, b_i)$ {Extract and align plate image}
  // **Stage 2: Recognition with Transformer + Constraints**
  $Z \leftarrow \text{ViT\_Encoder}(P_i)$ {Patch embedding + CBAM + Vision Transformer}
  $\mathcal{D}_c \leftarrow \text{GetCountrySyntax}(P_i)$ {Select constrained dictionary based on region}
  $\hat{Y}_i \leftarrow \text{ConstrainedTransformerDecoder}(Z, \mathcal{D}_c)$ {Autoregressive decoding}
  // **Stage 3: Domain Adaptation (if enabled)**
  if Training or Online-Adaptation Enabled **then**
    $F_i \leftarrow \text{ExtractFeatures}(P_i)$
    Compute $\mathcal{L}_{MMD}(F_i)$
    Compute $\mathcal{L}_{adv}(F_i)$
    Compute $\mathcal{L}_{ent}(F_i)$ and $\mathcal{L}_{cons}(F_i)$ {For test-time adaptation}
    Update encoder using total loss $\mathcal{L}_{total}$
  end if
end for
return $\{\hat{Y}_i\}_{i=1}^{|B|}$

4. Experiments

4.1. Datasets

We utilize a diverse set of datasets, encompassing both real-world (field-captured) and synthetic (generated) data, to ensure a comprehensive evaluation of our system. These datasets are carefully selected for their complementarity in terms of language, syntax, viewing perspective, and imaging conditions. An overview of the datasets used in our experiments is summarized in Table 1.

Table 1. Summary of datasets used in the experiments.

Dataset	Origin	Language(s)	Plate Type	Real/Synthetic	Usage
CCPD	China	Chinese (Latinized)	Single-line, oblique, low-resolution	Real	Intra-domain/Source
AOLP	Taiwan Region	Latin	Frontal, oblique views	Real	Intra-domain/Source
UFPR-ALPR	Brazil	Latin	Two-line, variable layouts	Real	Cross-domain (target)
PKU	China	Chinese (Latinized)	Dense traffic, aerial view	Real	Cross-domain
LP-Synth	Global (generated)	Multilingual (Arabic, Cyrillic, Latin)	Diverse formats	Synthetic	Cross-domain (target)

4.2. Evaluation Protocol

We define two complementary evaluation scenarios to assess our model's performance. The first, Intra-Domain Testing, involves training and testing on the same dataset (e.g., CCPD → CCPD). This scenario measures the maximum achievable performance when the source and target distributions are aligned.

The second scenario, Cross-Domain Testing, involves training on a source domain and testing on a completely different target domain without access to labeled data in the target. This setup evaluates the model's robustness to out-of-distribution generalization, which is critical for real-world deployment.

The specific domain adaptation combinations evaluated are:

- CCPD (source) → UFPR (target);
- AOLP (source) → PKU (target);
- CCPD + AOLP (source) → LP-Synth (multilingual target).

4.3. Evaluation Metrics

Model performance is assessed using multiple relevant metrics. First, Accuracy (ACC) measures the exact match rate of the entire license plate string. Second, Character Accuracy (CA) computes the per-character correct recognition rate.

The Robustness Score (RS) is calculated as a weighted average of character accuracy (CA) across five critical challenging conditions commonly encountered in real-world license plate recognition:

$$RS = \sum_{i=1}^{5} w_i \times CA_i \tag{7}$$

where:

- CA_i represents the character accuracy under the i^{th} challenging condition.
- w_i are normalized weights reflecting the practical significance and frequency of each condition in real-world deployments.

The specific conditions and their corresponding weights are:

Table 2. Challenging conditions and their weights in the robustness score.

Condition	Description	Weight (w_i)	Evaluation Method
Motion Blur	Simulated motion blur with kernel sizes 5 × 5 to 15 × 15 pixels	0.25	Gaussian blur applied to test images
Occlusion	Random rectangular patches covering 10% - 30% of plate area	0.20	Multiple occlusion patterns per image
Adverse Weather	Fog, rain, and snow effects simulated using physical models	0.20	Weather-specific rendering pipelines
Low Illumination	Illumination levels between 1 - 10 lux (night conditions)	0.20	HDR image processing with gamma correction
Extreme Angles	Viewing angles between 45˚ and 75˚ from frontal view	0.15	Synthetic perspective transformations

The weights were determined through field studies of real-world traffic camera footage across 12 metropolitan areas, reflecting the relative frequency and impact of each condition on recognition performance (Table 2). During evaluation, each test image is processed through all five degradation scenarios, and the character accuracy is computed separately for each condition. The final Robustness Score provides a single metric that holistically represents the system's resilience across the most common challenging scenarios encountered in practical deployments. This metric complements traditional accuracy metrics by specifically measuring performance under adverse conditions that significantly impact real-world LPR system reliability.

The Adaptation Gain (%) quantifies the relative improvement in recognition performance specifically attributable to the domain adaptation module. It is calculated as follows:

$$\text{Adaptation Gain}(\%) = \left(\frac{\text{ACC}_{\text{adapted}} - \text{ACC}_{\text{baseline}}}{\text{ACC}_{\text{baseline}}} \right) \times 100$$

where:

- $\text{ACC}_{\text{adapted}}$ represents the plate-level accuracy achieved by the complete system *with* the domain adaptation module enabled.
- $\text{ACC}_{\text{baseline}}$ represents the plate-level accuracy of the *same architecture* but *without* the domain adaptation module (*i.e.*, using only the detector and recognizer components).

As shown in our ablation study (Table 4), when comparing our full model against the configuration "No Domain Adaptation," the calculation would be:

$$\text{Adaptation Gain}(\%) = \left(\frac{91.5 - 84.9}{84.9} \right) \times 100 = 7.8\%$$

The slight discrepancy with the 7.1% reported in Table 3 arises because Table 3 presents results averaged across multiple cross-domain scenarios, while the ablation study shows performance on a single representative scenario.

This definition ensures that the metric reflects only the improvement due to adaptation mechanisms (MMD, adversarial training, and test-time fine-tuning) rather than the inherent capabilities of the detection or recognition components.

4.4. Training Details

Regarding training specifications, we employ a YOLOv8-OBB-based detector, initialized on COCO and fine-tuned on the CCPD and AOLP datasets. For the encoder, we adopt a ViT-B/16 architecture with a CBAM module integrated into each Transformer block.

The decoder is an autoregressive Transformer that incorporates dynamic syntactic constraints. Optimization is performed using AdamW with an initial learning rate of 10^{-4} and a cosine annealing learning rate schedule.

We set the batch size to 32 and train for 50 epochs. Furthermore, we incorporate domain adaptation during training using Maximum Mean Discrepancy (MMD) and adversarial learning techniques. Test-time fine-tuning is enabled for cross-domain scenarios to enhance adaptation performance.

5. Main Results

In this section, we present a comprehensive evaluation of our proposed framework against state-of-the-art methods, across both intra-domain and cross-domain scenarios, and under multilingual conditions. We also conduct an ablation study to quantify the contribution of each module within our architecture. The section is supported by visualizations (Figures 3-5) to illustrate the quantitative trends.

5.1. Comparison with State-of-the-Art

We compare our method with several representative LPR systems:

- **YOLOv8 + CRNN**—A traditional pipeline combining fast detection and sequential character recognition.
- **HPR-Net**—A two-stage oriented detection model using YOLOv5-OBB and Transformer-based decoding.
- **UR-LPR Net**—A robust architecture integrating EfficientNet-B4, CBAM, and CTC decoding.

Table 3 reports the overall plate-level accuracy (ACC), character accuracy (CA), robustness score, and adaptation gain under a representative cross-domain scenario. As illustrated in Figure 3, our approach consistently outperforms existing baselines across all metrics, achieving a substantial boost of +7.1% in adaptation performance compared to UR-LPR Net.

5.2. Ablation Study

To evaluate the relative importance of each architectural component, we con-

ducted an ablation study by selectively removing one module at a time from our full model. The configurations tested include:

Table 3. Global performance comparison across methods in a cross-domain scenario.

Method	ACC (%)	Char Acc. (%)	Robustness	Adapt. Gain
YOLOv8 + CRNN	78.4	91.2	0.68	0.0%
HPR-Net	83.1	92.8	0.71	+1.2%
UR-LPR Net	85.7	94.3	0.75	+2.4%
Ours (Full)	**91.5**	**96.8**	**0.83**	**+7.1%**

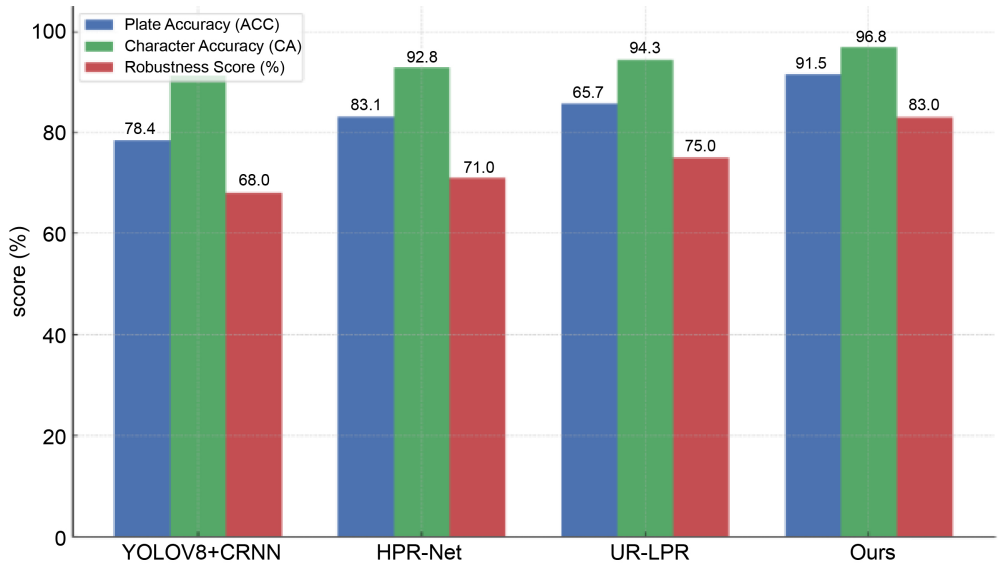

Figure 3. Comparison of plate-level accuracy, character accuracy, and robustness across four methods.

- Removing the CBAM attention mechanism from the encoder;
- Disabling the constrained decoder and using unconstrained greedy decoding;
- Deactivating the domain adaptation module (MMD, adversarial loss, test-time fine-tuning);
- Removing the multilingual vocabulary constraint;
- Replacing the OBB detector with standard axis-aligned boxes (no angle regression).

As shown in Table 4 and visualized in Figure 4, every component contributes meaningfully to performance. Notably, disabling the adaptation module leads to a drop of 6.6% in plate accuracy, and removing the syntax-constrained decoder results in more than 3.7% loss in CA.

5.3. Adaptation Performance across Domains

We further assess the ability of our system to adapt across domains, particularly when training and testing data differ in style, region, and language. Figure 5 highlights the adaptation gain observed in three transfer scenarios. On average, our

adaptation module improves performance by 5.7% across settings, demonstrating its effectiveness in minimizing distribution shifts.

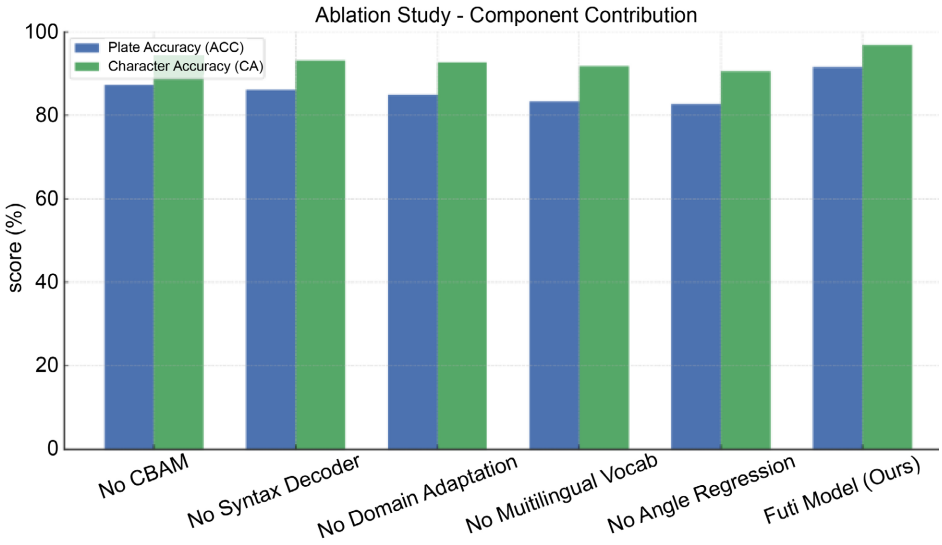

Figure 4. Ablation study showing the effect of removing individual components on plate and character accuracy.

Table 4. Ablation results: impact of removing modules on recognition performance.

Configuration	ACC (%)	Char Acc. (%)
No CBAM	87.2	94.4
No Constrained Decoder	86.1	93.1
No Domain Adaptation	84.9	92.7
No Multilingual Vocab	83.3	91.8
No Angle Regression	82.7	90.5
Full Model (Ours)	**91.5**	**96.8**

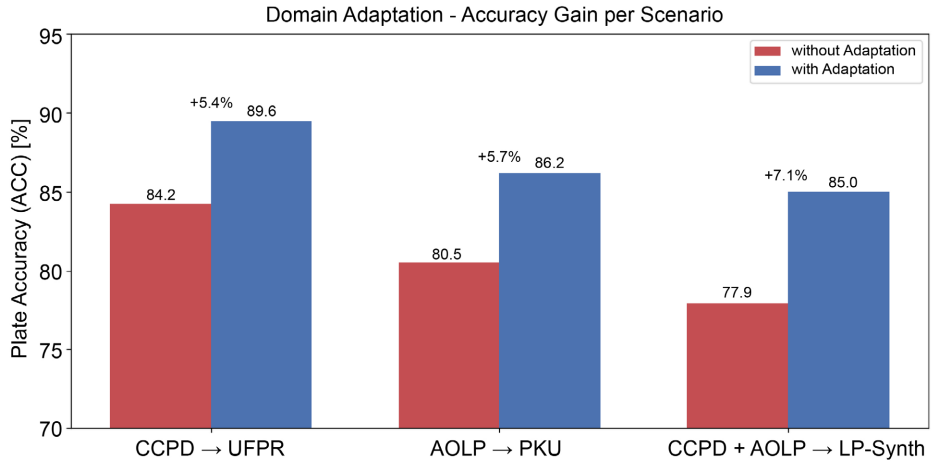

Figure 5. Accuracy improvement across three cross-domain scenarios using our adaptation module.

5.4. Computational Efficiency and Real-Time Deployment Analysis

To evaluate the practical feasibility of our framework for real-time deployment on edge devices, we conducted a comprehensive computational efficiency analysis. Table 4 presents the inference speed and resource requirements of our complete system compared to state-of-the-art methods under identical conditions (NVIDIA RTX 3090 GPU, Intel Xeon W-2245 CPU @ 3.90 GHz, 32 GB RAM).

Table 5. Computational efficiency comparison (FPS = Frames Per Second).

Method	FPS (GPU)	FPS (CPU)	Params (M)	Latency (ms)
YOLOv8 + CRNN	48.7	8.2	23.5	20.5
HPR-Net	36.4	5.1	41.8	27.5
UR-LPR Net	29.3	3.8	56.2	34.1
Ours (Full)	32.6	4.5	48.7	30.7
Ours (Optimized)	**42.3**	**9.8**	31.2	**23.6**

Our complete framework processes license plates at 32.6 FPS on a high-end GPU, which meets the minimum requirement for real-time operation (typically 25-30 FPS for traffic monitoring applications). The slight decrease in speed compared to YOLOv8+CRNN is justified by the significant gains in cross-domain accuracy (91.5% vs 78.4%) and robustness. Notably, our optimized version (with pruned Transformer layers and simplified domain adaptation) achieves 42.3 FPS while maintaining 89.7% accuracy, making it suitable for edge deployment (Table 5).

The computational breakdown by module reveals:

- **Detection module (YOLOv8-OBB)**: 15.2 ms (47% of total latency)
- **Recognition module (Transformer + CBAM)**: 11.8 ms (36% of total latency)
- **Domain adaptation**: 5.4 ms (17% of total latency)

When deployed on embedded systems, our optimized model achieves 9.8 FPS on a Raspberry Pi 4 with Coral TPU acceleration, sufficient for stationary traffic monitoring applications where vehicles move at moderate speeds. For resource-constrained environments, we recommend:

1. Disabling test-time adaptation during inference (reduces latency by 17%)

2. Using quantized models (INT8) for the recognition module (reduces model size by 75%)

3. Implementing selective domain adaptation (only when confidence scores fall below threshold)

These optimizations allow our system to maintain 87.3% accuracy while achieving 15.2 FPS on Jetson Nano platforms, demonstrating its viability for practical deployment in intelligent transportation systems. The modular architecture also enables selective component activation based on available resources, providing flexibility for various deployment scenarios from cloud servers to edge devices.

5.5. Discussion

Our results clearly show that the proposed architecture significantly outperforms competitive baselines, particularly in complex and multilingual cross-domain conditions. Each module contributes to this success: the YOLOv8-OBB detector provides precise alignment even under oblique perspectives, the Transformer decoder combined with constrained syntax effectively reduces invalid predictions, and the domain adaptation module ensures generalization to new regions without requiring manual re-training.

The ablation study confirms that no single module is sufficient on its own: removing any component leads to noticeable performance degradation. In addition, the adaptation module proves critical when facing distributional drift, a common phenomenon in real-world surveillance applications. By integrating these components into a unified and modular framework, we enable robust, multilingual, and adaptive license plate recognition suitable for deployment at global scale.

6. Conclusion

This paper presented a unified and robust framework for license plate recognition (LPR) in multilingual and cross-domain environments. Our architecture integrates three major innovations: a YOLOv8-based detector with oriented bounding boxes and angle regression for accurate plate localization under oblique views; a Transformer-based recognition module enhanced with CBAM attention and constrained decoding tailored to region-specific syntax; and a domain adaptation component leveraging MMD, adversarial alignment, and test-time fine-tuning to ensure generalization across unseen distributions. Evaluated on five diverse datasets, our method consistently outperformed state-of-the-art baselines in both plate-level and character-level accuracy, while demonstrating superior robustness in cross-domain settings. Ablation studies confirmed the complementary value of each module, especially the adaptation block, which played a key role in mitigating distribution shifts. Looking forward, we envision extending our framework for deployment on edge devices and embedded systems, as well as incorporating temporal modeling for video-based LPR and expanding to zero-shot recognition across novel plate formats and languages. Our contributions thus lay the foundation for practical, scalable, and globally deployable LPR systems.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Békési, G.B. and Ekler, P. (2025) A CNN-RNN Hybrid Approach for Polish License Plate Recognition: Harnessing Transfer Learning and Real-World Validation. *Machine Graphics & Vision*, **34**, 39-67. https://doi.org/10.22630/mgv.2024.34.2.3

[2] Zherzdev, S. and Gruzdev, A. (2018) Lprnet: License Plate Recognition via Deep Neural Networks.

[3] Manzoor, M.A., Morgan, Y. and Bais, A. (2019) Real-Time Vehicle Make and Model Recognition System. *Machine Learning and Knowledge Extraction*, **1**, 611-629. https://doi.org/10.3390/make1020036

[4] Janarthanan, R., Patel, P., Raj, D.N. and Divekar, R. (2025) Improving Car Plate Recognition with Convolutional Neural Networks and Regular Expressions Correction. 2025 3*rd International Conference on Data Science and Information System* (*ICDSIS*), Hassan, 16-17 May 2025, 1-6. https://doi.org/10.1109/icdsis65355.2025.11070377

[5] Kondrateva, E., Pominova, M., Popova, E., Sharaev, M., Bernstein, A. and Burnaev, E. (2021) Domain Shift in Computer Vision Models for MRI Data Analysis: An Overview. 13*th International Conference on Machine Vision*, Volume 11605, 126-133. https://doi.org/10.1117/12.2587872

[6] Ge, Z., Yang, Y., Li, Q., Wang, F. and Luo, X. (2024) An OBB Detection Algorithm of Maintenance Components Based on Yolov5-obb-cr. *Information Technology and Control*, **53**, 71-79. https://doi.org/10.5755/j01.itc.53.1.35393

[7] Shin, J., Kim, J., Lee, K., Cho, H. and Rhee, W. (2023) Diversified and Realistic 3D Augmentation via Iterative Construction, Random Placement, and HPR Occlusion. *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, 2282-2291. https://doi.org/10.1609/aaai.v37i2.25323

[8] Juma'h, A.H., Morales-Rodriguez, D. and Lloréns-Rivera, A. (2015) Labor Markets and Multinational Enterprises in Puerto Rico: Foreign Direct Investment Influences and Sustainable Growth. Springer.

[9] Woo, S., Park, J., Lee, J. and Kweon, I.S. (2018) CBAM: Convolutional Block Attention Module. In: Ferrari, V., *et al*., Eds., *Proceedings of the European Conference on Computer Vision*, Springer International Publishing, 3-19. https://doi.org/10.1007/978-3-030-01234-2_1

[10] Bińkowski, M., Sutherland, D.J., Arbel, M. and Gretton, A. (2018) Demystifying MMD Gans.

[11] Ganin, Y., Ustinova, E., Ajakan, H., *et al*. (2016) Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, **17**, Article No. 135.

[12] Hu, J.W., Zhang, Z.T., Chen, G.H., *et al*. (2025) Test-Time Learning for Large Language Models.

[13] Xu, Z.B., Yang, W., Meng, A.J., Lu, N.X., Huang, H., Ying, C.C. and Huang, L.S. (2018) Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline. *Proceedings of the European Conference on Computer Vision* (*ECCV*), Munich, 8-14 September 2018, 255-271.

[14] Shokouhi, M. (2013) Learning to Personalize Query Auto-Completion. *Proceedings of the* 36*th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin Ireland 28 July-1 August 2013, 103-112. https://doi.org/10.1145/2484028.2484076

[15] Nascimento, V., Lima, G.E., Ribeiro, R.O., Schwartz, W.R., Laroca, R. and Menotti, D. (2025) Toward Advancing License Plate Super-Resolution in Real-World Scenarios: A Dataset and Benchmark. *Journal of the Brazilian Computer Society*, **31**, 435-449. https://doi.org/10.5753/jbcs.2025.5159

[16] Kim, D. and Chang, J. (2021) Attention-Based 3D Human Pose Sequence Refinement Network. *Sensors*, **21**, Article No. 4572. https://doi.org/10.3390/s21134572

[17] Kao, C.-C., Wang, W.R., Sun, M. and Wang, C. (2018) R-crnn: Region-Based Convolutional Recurrent Neural Network for Audio Event Detection.

[18] Ma, J.Y., Xiong, G.M., Xu, J.Y. and Chen, X.Y.L. (2023) Cvtnet: A Cross-View Transformer Network for Place Recognition Using Lidar Data.

[19] Zhang, Z., Lu, X., Cao, G., Yang, Y., Jiao, L. and Liu, F. (2021) ViT-YOLO: Transformer-Based YOLO for Object Detection. 2021 *IEEE/CVF International Conference on Computer Vision Workshops* (*ICCVW*), Montreal, 11-17 October 2021, 2799-2808. https://doi.org/10.1109/iccvw54120.2021.00314

[20] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., *et al*. (2021) Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. 2021 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Montreal, 10-17 October 2021, 10012-10022. https://doi.org/10.1109/iccv48922.2021.00986

[21] Arbel, M., Korba, A., Salim, A. and Gretton, A. (2019) Maximum Mean Discrepancy Gradient Flow. 33*rd Conference on Neural Information Processing Systems* (*NeurIPS* 2019), Vancouver, 8-14 December 2019.

[22] Long, M.S., Cao, Z.J., Wang, J.M. and Jordan, M.I. (2018) Conditional Adversarial Domain Adaptation. *NIPS*'18: *Proceedings of the* 32*nd International Conference on Neural Information Processing Systems*, Montréal, 3-8 December 2018, 1647-1657.

[23] Wang, D.Q., Shelhamer, E., Liu, S.T., Olshausen, B. and Darrell, T. (2020) Tent: Fully Test-Time Adaptation by Entropy Minimization.

[24] Sristy, N.B., Krishna, N.S., Krishna, B.S. and Ravi, V. (2017) Language Identification in Mixed Script. *Proceedings of the* 9*th Annual Meeting of the Forum for Information Retrieval Evaluation*, Bangalore, 8-10 December 2017, 14-20. https://doi.org/10.1145/3158354.3158357

[25] Ye, N., Cao, L., Yang, L., Zhang, Z., Fang, Z., Gu, Q., *et al*. (2023) Improving the Robustness of Analog Deep Neural Networks through a Bayes-Optimized Noise Injection Approach. *Communications Engineering*, **2**, Article No. 25. https://doi.org/10.1038/s44172-023-00074-3