TRI-GATE: A Tri-Modal Anti-Spoofing System for Gate Access Using Vehicle, License Plate, and Face Recognition

Muhannad Alsultan, Thamer Alghonaim, Abdulaziz Alorf*,
Bandar Alwazzan, Faisal Alsakakir, Abdullah Alhassan, Yousif Hussain
Department of Electrical Engineering-College of Engineering, Qassim University, Saudi Arabia

***Abstract*—Vehicle gate access, in general, still relies heavily on manual inspection of identification cards and visual verification by security guards, which is slow, tedious, and susceptible to spoofing. Single-modality, computerized systems that utilize license plates, vehicle appearance, and facial recognition can partially alleviate this difficulty. Still, they are prone to spoofing and generally perform poorly in real-world scenarios (e.g., glare, occlusion, and tinted glass). This study presents TRI-GATE, a tri-modal anti-spoofing framework that unifies vehicle, license plate, and face recognition within a single, real-time decision pipeline. The system employs YOLOv4-tiny for vehicle detection and a MobileNetV2-based classifier for make–model recognition, a retrained MTCNN and LPRNet pair for license plate detection and recognition on Saudi-specific datasets (17,000 images for detection and 35,000 for recognition), and RetinaFace with InsightFace embeddings, along with a linear SVM, for driver identification. An IoU-based best-frame selection scheme reduces latency by forwarding only the most informative frame to the recognition modules. Score-level fusion is then performed by a linear SVM that learns the relative importance of each modality for the final access decision. Evaluated on a dedicated tri-modal dataset, TRI-GATE achieves 97% gate-level accuracy with an end-to-end latency of 66 ms per frame (≈ 15.15 FPS), and demonstrates robust performance in a real-world gate-like deployment, substantially improving both security and operational efficiency over existing single- and bi-modal solutions.**

***Keywords*—Tri-modal anti-spoofing; vehicle recognition; license plate recognition; face recognition; real-time gate access control; multimodal biometrics**

I. Introduction

In most conventional vehicle gate access setups, verification still depends largely on manual inspection. Security officers typically visually inspect vehicles and confirm entry permits or identification cards by hand. While simple in principle, this routine often becomes slow, physically demanding, and susceptible to both fatigue and oversight. Delays accumulate during busy hours, creating traffic queues at the gate and inconsistent enforcement of security procedures. What raises greater concern is how easily such systems can be deceived—people with no authorization might convincingly pose as legitimate staff or slip through the gate using falsified documents.

Although automation has made considerable progress, many existing gate systems remain limited to a single mode of recognition—commonly license plate detection or facial identification. Systems built around a single identity marker are easy to compromise: plates can be cloned, and faces may go undetected under tinted glass, dim lighting, or awkward camera angles. Such limitations highlight a crucial shortfall: the absence of a unified, multimodal verification framework capable of confirming both vehicle and driver identity consistently, even amid the unpredictable and imperfect conditions of real-world environments.

This study tackles existing limitations by developing an AI-driven anti-spoofing framework that unifies vehicle detection and classification, license plate recognition, and facial recognition within a single decision pipeline. Using computer vision methods such as YOLO (You Only Look Once) and OpenCV (Open Computer Vision library), along with a Support Vector Machine (SVM) classifier, the system processes and integrates outputs from multiple models to generate intelligent access control decisions. The fusion strategy not only boosts accuracy and processing speed but also makes gate access more robust against spoofing attempts, thereby reducing unauthorized access and improving the operational efficiency and security of facility gate management.

A. Related Work

In recent years, researchers have devoted significant attention to improving the reliability of automated vehicle gate systems. Early systems mainly relied on single data sources—most often license plate recognition or basic vehicle appearance matching. These systems worked reasonably well in controlled environments but struggled when lighting changed, the camera angle shifted, or spoofing was attempted. As listed in Table I, more recent work has shifted toward multimodal approaches that combine license plate data, vehicle signatures, and even driver identity cues. This transition reflects an effort to create systems that not only identify but also verify vehicles under real-world variability. Despite these developments, many solutions still face trade-offs between speed, cost, and recognition stability, especially in real-time gate scenarios where decisions must be made in fractions of a second.

1) License Plate Recognition (LPR): License plate recognition has been studied for years and continues to play a central role in automated gate systems. Early neural models, such as WPOD-NET, demonstrated that end-to-end detection and segmentation were feasible, achieving about 89% accuracy on benchmarks such as SSIG and AOLP [1]. But while that was

*Corresponding author.

TABLE I. Review of Vision-based Recognition Approaches for Automated Gate Entry

Author(s)	Year	Model/System	Fusion Type	Dataset(s)	Best Reported Performance
Silva *et al.* [1]	2018	WPOD-NET	License plate only	Cars Dataset, SSIG, AOLP, OpenALPR (EU and BR), and newly created CD-HARD dataset	89.33% (average accuracy)
Zhu *et al.* [2]	2023	SYOLOv5s + GAM + FEM	License plate only	CCPD	96.6% mAP @ 43.86 FPS, and Parameters = 5.07 M
Sarhan *et al.* [3]	2024	YOLOv8 + Easy-OCR + CNN	License plate only	EALPR (for detection) + Arabic Letters & Numbers OCR (for recognition)	99.42% accuracy (CNN-based recognition), and YOLOv8 detection mAP 94.26%
Tao *et al.* [4]	2024	YOLOv5-PDLPR	License plate only	CCPD	99.4% accuracy @ 159.8 FPS
Lou *et al.* [5]	2019	FDA-Net	Vehicle only	VERI-Wild dataset (main), plus VehicleID and VeRi-776 for comparison	Best (on VeRi-776): mAP = 55.49%, Rank-1 = 84.27%, and Rank-5 = 92.43%
Yu *et al.* [6]	2023	SOFCT	Vehicle only	VeRi-776 and VehicleID	VeRi-776: mAP = 80.7%, and Rank-1 = 96.6%; VehicleID: up to mAP = 89.8%, and Rank-1 = 84.5%
Zhu *et al.* [7]	2023	DSN	Vehicle only	VehicleID, Vehicle-1M, and VeRi-776	Best (on VeRi-776): mAP = 76.3%, and Rank-1 = 94.8%
Liang *et al.* [8]	2023	S-TVReID	Vehicle only	VeRi-776 and VehicleID	Best (on VeRi-776): mAP = 80.8%, Rank-1 = 96.4%, and Rank-5 = 98.5%
Lian *et al.* [9]	2023	MED	Vehicle only	VeRi-776 and VehicleID (+ Market-1501, DukeMTMC-reID, and MSMT17)	VeRi-776: mAP = 83.4%, and Rank-1 = 97.2%; VehicleID: Rank-1 = 87.8/83.1/81.4%; SOTA on pedestrian sets
Alim *et al.* [10]	2023	YOLOv3 + YOLOv5 + LPRNet	License plate + driver information	WiderFace (for face detection), and custom Turkish LP dataset	Accuracy = 97.34% (LPR), AP = 0.926/0.908/0.765 (FD), and 13 FPS combined
Teja *et al.* [11]	2024	ANPR (KNN + CNN) + Face Recognition (LBPH)	License plate + driver information	Custom dataset (vehicle plates & individual faces)	Accuracy = 96.54% (Face), 88.67% (ANPR)
Akbar *et al.* [12]	2024	Face Recognition + EasyOCR + Haar Cascade Classifier	License plate + driver information	8 visitors (faces & number plates, and Universitas Andalas)	Success rate = 62.5%
Iyer *et al.* [13]	2024	YOLOv8 + EasyOCR + dlib (CNN)	License plate + driver information	Combination of 3 open Indian number plate datasets + custom Devanagari dataset	ANPR: mAP@50 = 98.35%, OCR: 88.14%, and Face Recognition: 98.34%
Mustafa *et al.* [14]	2024	MobileNet-V2 + YOLOx + YOLOv4-tiny + PaddleOCR + SVTR-tiny	License plate + Vehicle	COCO, Stanford Car Dataset, and Firat University dataset	97.5% (overall accuracy)
Ramajo-Ballester *et al.* [15]	2024	YOLOv5 + EfficientNetB0 (FastReID)	License plate + Vehicle	UC3M-LP and UC3M-VRI	mAP = 0.893 (detection), 0.764 (OCR), accuracy = 0.979 (re-ID), and ≈ 58.1 ms per image (real-time capable)
Khor *et al.* [16]	2024	Multi-Task YOLOv8 (multi-head YOLOv8 for OCR, LP detection, and VCR)	License plate + Vehicle	In-house multi-labelled dataset (1555 images) + synthetic augmentation using TRDG	mAP$_{50}$ = 0.778 (OCR), 0.963 (LP), 0.881 (VCR); Average = 0.874
AlDahoul *et al.* [17]	2025	VehiclePaliGemma	License plate + Vehicle	258-image Malaysian license plate dataset (Tapway Sdn Bhd) + 600 synthetic images (for fine-tuning)	87.6% (plate accuracy), and 97.66% (character accuracy)
Saadouli *et al.* [18]	2020	Fusion of SIFT + DoG + Viola–Jones modules	License plate + driver information + vehicle	Qatar University car surveillance dataset (225 images and 24 car types)	74.63% accuracy (real dataset)

quite good for its time, accuracy often dropped under glare or distortion.

Later studies started chasing speed as much as precision. One of the more interesting attempts came from Zhu and colleagues [2], who developed a lightweight version of YOLOv5, called SYOLOv5s. They added a small attention mechanism to help the model focus on fine details on plates rather than get distracted by car reflections. The approach achieved a mean average precision (mAP) of 96.6% on the CCPD dataset at roughly 40–45 FPS, which is impressive for an embedded setup.

In a similar direction, Sarhan *et al.* [3] paired YOLOv8 with Easy-OCR and a compact CNN to recognize Egyptian plates. Their system achieved 94.26% mAP and about 99.4% recognition accuracy. A slightly later work by Tao *et al.* [4] pushed further with YOLOv5-PDLPR, achieving close to 160 FPS while maintaining roughly 99.4% accuracy.

These results look great on clean datasets, but in real life, the story changes. Two identical plates on different vehicles or even a printed photo of a plate can easily pass. *The problem is that plate-based identification only tells what plate it sees—it says nothing about who is driving or whether that plate really* *belongs on that car.*

2) Vehicle recognition and re-identification (Re-ID): As license-plate models reached maturity, attention began shifting to the vehicle's physical appearance. The idea was that the combination of color, make, and shape could help track or verify vehicles across cameras. Lou *et al.* [5] developed FDA-Net, an early method for feature disentanglement that achieved mAP of 55.49% and Rank-1 of 84.27% on the VeRi-776 benchmark. Though not perfect, it was a turning point.

Later, transformer-based models changed the landscape. Yu *et al.* [6] proposed SOFCT, a transformer that couples features across views, boosting accuracy to mAP 80.7% and Rank-1 96.6%. Liang *et al.* [8] followed with a spatial-temporal transformer (S-TVReID) that achieved nearly identical results, while Lian *et al.* [9] introduced a multi-branch model (MED) that squeezed out a bit more, reporting mAP of 83.4% and Rank-1 of 97.2%.

Around the same time, Zhu *et al.* [7] suggested a dual self-attention network (DSN). Instead of relying on a single global attention layer, they used two complementary self-attention stages to capture both coarse and fine details. The model achieved a mAP of 76.3% and a Rank-1 accuracy of 94.8%, indicating that a thoughtfully designed attention mechanism can match the performance of more complex architectures.

Altogether, these advances make vehicle appearance recognition highly reliable, but it still doesn't answer the key security question—who is inside the vehicle? *Appearance cues alone can't guarantee authorized access.*

3) Fusion of license plate and driver information: A natural next step was to include the driver's identity. Several works explored this idea, often combining a license-plate detector with facial recognition. Alim *et al.* [10] proposed an edge-based model using YOLOv3/YOLOv5 and LPRNet, achieving roughly 97% plate accuracy and solid face detection results. Teja *et al.* [11] mixed a KNN/CNN ANPR approach with LBPH facial recognition, scoring 96.54% for the face and 88.67% for the plate.

Other studies followed, but under more constrained setups. Akbar *et al.* [12] tested a prototype using Haar cascades and EasyOCR for visitor entry, but accuracy dropped to around 62%, mostly due to variable lighting and glass reflections. Iyer and Dhavale. [13] later improved on this with YOLOv8, EasyOCR, and dlib-CNN, achieving about 98% accuracy for faces and 88.14% for plates.

Despite encouraging results, most of these projects were small-scale and tested in fairly controlled conditions. *Problems like tinted windshields, poor lighting, or the driver's head turned away still cause big accuracy swings. Moreover, few systems tried to fuse both cues into a single, reliable decision.*

4) Fusion of license plate and vehicle features: Some researchers instead checked whether the vehicle's appearance matched its plate. Mustafa and Karabatak [14] proposed a multi-stage design combining MobileNet-V2, YOLOx, YOLOv4-tiny, PaddleOCR, and SVTR-tiny, with an overall accuracy near 97.5%. Ramajo-Ballester *et al.* [15] merged YOLOv5, EfficientNetB0, and FastReID, reporting mAP of 0.893 for license-plate detection and 97.9% re-identification accuracy, all running at roughly 58 ms per frame.

Khor *et al.* [16] developed a multitask YOLOv8 with shared layers and separate heads for OCR, plate detection, and vehicle recognition, achieving an average mAP of 0.874. Later, AlDahoul *et al.* [17] introduced VehiclePaliGemma, which used a vision-language approach and achieved 87.6% plate accuracy and 97.66% character accuracy on Malaysian datasets.

These multi-modal systems make it easier to spot mismatched or fake plates, but they stop short of confirming driver identity. *That gap still allows a legitimate vehicle to be driven by an unauthorized person.*

5) Tri-modal fusion of plate, vehicle, and driver: A handful of earlier works did try to combine all three components. Saadouli *et al.* [18] developed a system that combined SIFT, Difference-of-Gaussian, and Viola-Jones methods to fuse plate, vehicle, and face information. Tested on a small dataset from Qatar University, the setup achieved about 74.6% accuracy. Although modest, that study demonstrated the concept was feasible and hinted at what future models could achieve with deep learning.

However, since then, the tri-modal direction has received surprisingly little attention. Most newer works still treat plates, vehicles, and faces as separate modules rather than fusing them into a single coherent anti-spoofing decision.

6) Research gap and motivation: From reviewing this body of work, a few consistent gaps appear. Nearly all systems rely on one or two cues rather than combining all three, leaving them vulnerable to spoofing or substitution attacks. Models trained on datasets such as CCPD or VeRi-776 often struggle to generalize to Saudi or GCC license plates, which vary in color, shape, and script. Another issue is that most papers report part-level metrics—such as mAP or recognition rate—without testing full gate-level performance, including false acceptance and false rejection under real attack conditions.

To tackle these problems, the present study develops a deep tri-modal anti-spoofing system that merges three complementary sources: the license plate, the vehicle's visual signature, and the driver's identity. The system uses YOLOv4-tiny [19] for vehicle detection, MobileNetV2 [20] for vehicle classification, MTCNN [21] for license plate detection, LPRNet [22] for license plate recognition, RetinaFace [23] for face detection and alignment, and the InsightFace framework [24] (trained with the ArcFace loss [25] for feature extraction) combined with a linear SVM [26] for face recognition. The recognition results (plate, face, and vehicle) are then combined with a simple linear SVM that learns how much to trust each output based on its confidence. The dataset itself was built around Saudi license plates, so it's tuned to the region. In the end, the goal is to narrow the long-standing gap between strong recognition results in experiments and the reliability needed for real gate security.

B. Contributions

The principal contributions of this work are outlined below:

- A real-time, integrated anti-spoofing system: Unlike traditional gate access mechanisms that rely solely on a single recognition model—be it facial, vehicle, or license plate recognition—our system unifies all three within a single, real-time framework. This integration not only strengthens security but also minimizes unauthorized entries and prevents unnecessary traffic delay at the gate.

- A new labeled dataset for license plate detection: We introduce a fully annotated dataset of 17,000 Saudi Arabian license plate images for detection tasks. To the best of our knowledge, no previous study has provided such a dataset. A visual example of this dataset is shown in Fig. 4.
- A new labeled dataset for license plate recognition: Another dataset, consisting of 35,000 labeled images of Saudi license plates, is presented for recognition purposes. This collection is also newly introduced to the literature, and a representative sample is provided in Fig. 5.
- Public release and implementation details: Both datasets, along with the complete system, are openly available through our GitHub repository [27]. The implementation is clearly documented and designed with scalability and flexibility in mind, allowing it to be adapted or extended to fit a range of deployment environments.

Together, these contributions emphasize the originality and practical value of our study. By merging multiple recognition techniques within a single system and grounding them in newly developed datasets, this work lays a strong foundation for improving automated vehicle access control in real-world conditions.

II. Proposed Model

A. Overview

The primary goal of this research is to make vehicle entry systems faster, smarter, and less dependent on manual checks. In many facilities, access is still verified by guards, which often slows things down and sometimes leads to errors or even security breaches. The problem becomes more obvious during rush hours when the number of vehicles increases and human attention starts to slip. To solve this, we developed an automated model that works in real time and removes the need for constant supervision. The system relies on three main visual features: the driver's face, the vehicle, and its license plate. Our proposed system fuses these features to create a stronger, more reliable decision-making model that speeds up entry validation and makes it safer, without adding more human effort.

Our proposed model, shown in Fig. 1, illustrates how the anti-spoofing gate access system operates as a complete process, from video capture to the final decision. When a vehicle approaches, the system records live video and automatically selects the clearest frame using the intersection-over-union (IoU) method to ensure the vehicle is properly captured. From that frame, the YOLOv4-tiny model [19] detects the car, while MobileNetV2 [20] classifies its make and model. The license plate area is then located using a modified MTCNN [21], and LPRNet [22] reads the characters on it. At the same time, RetinaFace [23] detects and aligns the driver's face, and InsightFace [24], [25]—together with a linear SVM [26] classifier—identifies who the person is. These three recognition results (vehicle, plate, and face) are then combined using a weighted linear SVM, which calculates an overall score to determine whether access should be allowed. In short, the figure illustrates a smooth, AI-based workflow that combines detection, recognition, and decision-making into one unified, real-time security system. The following subsections elaborate on every step in this proposed pipeline.

B. Vehicle Detection

In the proposed anti-spoofing gate access system, identifying vehicles in each video frame is the central operation. To meet real-time requirements without overloading the system, a lightweight detection model was used. In this case, the YOLOv4-tiny network [19] was chosen because it strikes a reasonable balance between detection accuracy and processing speed. Compared to the full YOLOv4 version [28], it runs faster due to its reduced parameter count and simpler convolutional layers. These characteristics make it practical for edge or embedded systems, where memory and processing resources are limited.

Before the detection stage, each input image is processed using OpenCV [29], which handles basic preprocessing tasks such as reading the image, resizing it, and converting it to the correct color space. Once that is done, YOLOv4-tiny divides the image into an $S \times S$ grid. Every cell in this grid predicts several bounding boxes, each one represented by five numeric values and a set of class probabilities. These values describe the position, size, and confidence level associated with each possible vehicle in the frame, defined as:

$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix},$$

where,

- P_c: Confidence score indicating the probability of an object's presence within the bounding box.
- b_x, b_y: Normalized center coordinates of the bounding box.
- b_h, b_w: Normalized height and width of the bounding box.
- C_i: Class probabilities for the object categories (via softmax).

The extracted bounding box parameters are shown in Fig. 2.

After the network finishes predicting several possible bounding boxes, the raw output still needs some cleanup. The goal of this step is to keep the detections that actually matter and discard the ones that don't. Two standard techniques handle this refinement:

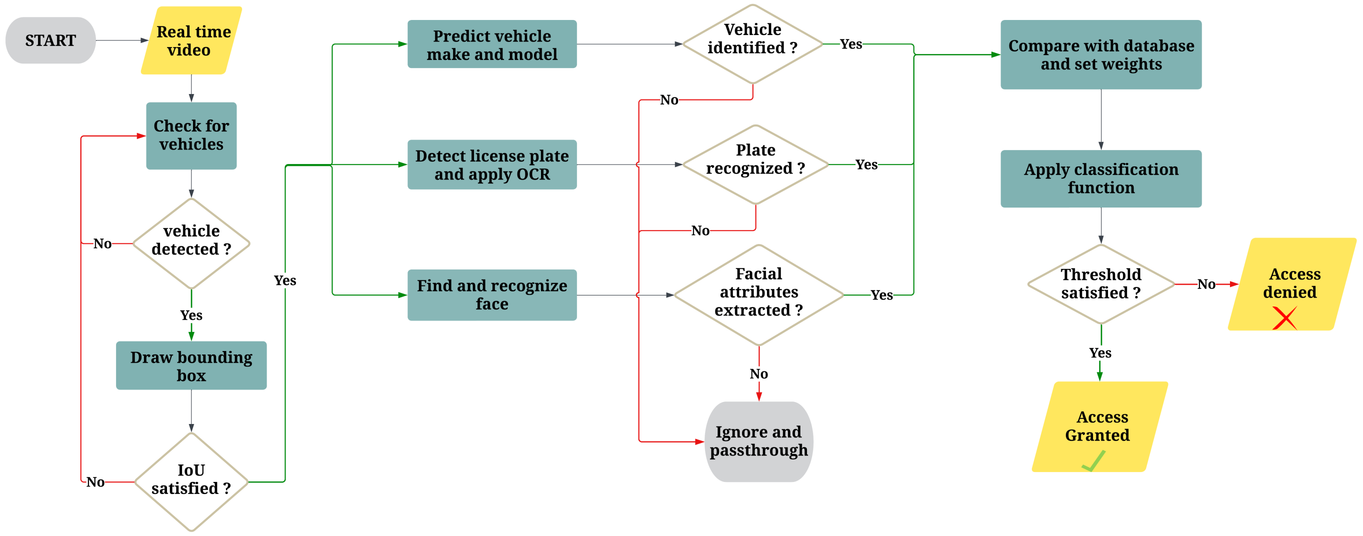

Fig. 1. Workflow of the proposed AI-based anti-spoofing gate access tri-modal system, combining vehicle, license plate, and facial recognition for robust and unified real-time verification.

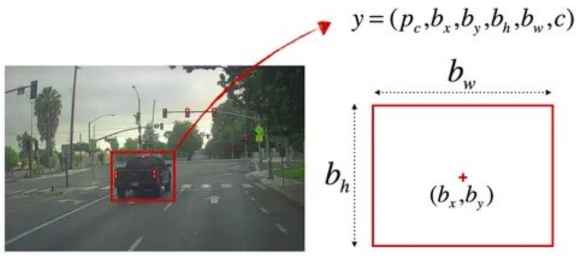

Fig. 2. Bounding box parameters (b_x, b_y, b_h, b_w) predicted by YOLOv4-tiny, defining each detected vehicle's position and size.

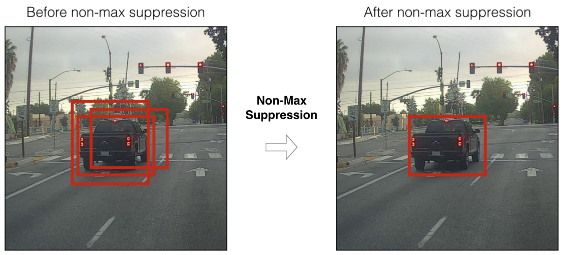

Fig. 3. Illustration of non-maximum suppression (NMS) removing overlapping bounding boxes and keeping only the highest-confidence detection per vehicle.

- Score thresholding: Any bounding box whose confidence value falls below a chosen threshold is simply dropped. This first filter removes weak forecasts, allowing the next phases to focus on stronger ones.
- Non-maximum suppression (NMS): If several boxes overlap on the same object, only the one with the highest confidence is kept. The rest are removed whenever their intersection-over-union (IoU) with the highest-scoring box exceeds a predefined threshold.

Working together, these two steps result in each object being represented by a single, clean bounding box. They also lighten the computational load and help the model reach its decision faster. As illustrated in Fig. 3, NMS trims away overlapping boxes, leaving the final, precise outline.

The use of YOLOv4-tiny comes down to its balance between speed and accuracy. Running approximately 83.33 frames per second on an NVIDIA GeForce RTX 2080 Ti GPU, it can manage real-time monitoring at a gate or checkpoint. Even though it's a smaller variant of YOLOv4, it performs reliably under difficult lighting, partial occlusion, and awkward vehicle angles.

With YOLOv4-tiny handling vehicle detection, later components—vehicle classification, license-plate recognition, and face identification—receive neatly cropped regions of interest. That early precision keeps the entire tri-modal anti-spoofing system steady, fast, and dependable in live operation.

C. Best-Frame Selection for Low-Latency Vehicle Gating

In real-time video, a moving vehicle appears across several consecutive frames at different positions and scales. Processing every frame is wasteful—it drives up latency without significantly improving accuracy. A better strategy is to select a single best frame that preserves the most useful visual detail while keeping computation lean.

As the vehicle approaches the gate, each frame provides a slightly different view. We aim to select the moment when the vehicle is most prominent and least occluded—typically the frame with the largest, well-centered bounding box—so that downstream modules (vehicle classification, license plate recognition, and driver or occupant cues) receive the cleanest signal.

We formalize this with the intersection over union (IoU) between the detected bounding box B_t in frame t and a predefined reference box B^* that encodes the desired on-screen extent:

$$\mathrm{IoU}(B_t, B^*) = \frac{|B_t \cap B^*|}{|B_t \cup B^*|}, \quad t^\dagger = \arg\max_t \mathrm{IoU}(B_t, B^*).$$

This simple rule favors frames where the target is closest and best aligned with the expected view. Crucially, if two different vehicles enter the scene simultaneously, the method still yields a single, unambiguous selection: among all detections across frames, only the vehicle whose bounding box achieves the largest IoU with B^* is chosen, and only its best frame is forwarded downstream. This ensures that the system detects and processes only one vehicle at a time, eliminating multi-vehicle conflicts. Then the procedure per short time window:

- Detect vehicle candidates and obtain B_t for each frame.
- Compute $\mathrm{IoU}(B_t, B^*)$ for all candidates.
- Select $t^\dagger = \arg\max_t \mathrm{IoU}(B_t, B^*)$; export only frame $t^\dagger$ and its associated vehicle.

By emitting exactly one, high-quality frame, the system trims redundant computation and preserves recognition accuracy. In practice, this best-frame gate reduces end-to-end latency and strengthens anti-spoofing behavior, making it a robust solution for real-time vehicle access control.

D. Vehicle Classification

Following detection, the vehicle crop is forwarded to the make–model classifier. We employ the off-the-shelf model in [30], which is built on MobileNetV2 [20] and trained using transfer learning similar to [31]. Concretely, MobileNetV2 serves as the backbone feature extractor, and the resulting classifier operates in real-time on gate imagery. Vehicle crops are first resized to 224×224 pixels; in practice, reliable inference is obtained for objects as small as 30×30 pixels after pre-processing. The model's training coverage comprises approximately 400 brands and 7,000 car models, with canonical viewpoints limited to front, rear, and side. Under these conditions, the reported top-1 accuracy reaches about 95%. We selected this module for its real-time operation and strong accuracy, which is backed by training that covers every car make and model in Saudi Arabia. The module integrates with the upstream detector by consuming the refined bounding box for each candidate (see Fig. 2), and its system-level role is consistent with the end-to-end workflow (see Fig. 1).

E. License Plate Detection

After vehicle detection and best-frame selection, the system localizes the license plate using a retrained variant of the multi-task cascaded convolutional network (MTCNN) [21]. Although MTCNN is conventionally used for face detection and alignment, prior work shows that, with suitable retraining and loss-function tuning, it can be adapted to license plate detection under diverse viewpoints and backgrounds [32]. Following this approach, we fine-tuned the network on Saudi Arabian plates to enhance its robustness under challenging capture conditions (e.g., strong illumination, oblique angles, and cluttered scenes).

Because no large, open datasets exist for Saudi plates, we constructed a dedicated detection corpus. Images were captured manually and gathered from public social-media sources (programmatically fetched with Instaloader [33]), yielding approximately 17,000 annotated images that span all Saudi plate types and a broad range of environments (appearance, background, lighting, scale, and viewpoint). Representative examples and detector outputs of our novel dataset are shown in Fig. 4.

Annotations were created using the computer vision annotation tool (CVAT), where plate bounding boxes were produced and exported as XML for training. The novel curated dataset (images and ground truths) is organized with clear documentation to support reproducibility and is publicly available on our GitHub repository cite [27].

After assembling the dataset, the next step is to train the license plate detector. Given MTCNN's cascaded design, we retained the proposal (P-Net) and output (O-Net) stages. We omitted the refinement (R-Net) stage after empirical testing indicated no degradation in accuracy for this task. Training hyperparameters for P-Net and O-Net are summarized in Table II. Our end-to-end system achieved an overall accuracy of 99.2%.

TABLE II. TRAINING HYPERPARAMETERS FOR THE LICENSE-PLATE DETECTION MODEL (RETRAINED MTCNN WITH P-NET AND O-NET STAGES)

Parameter	Value
Number of epochs	50
Learning rate	0.001
Initial weight	0.1
Batch size	64
Dataset split	70% train, 20% validation, 10% test

For classification at each retained stage, we used the standard cross-entropy (log-loss) objective [Eq. (1)]. For bounding-box regression, we minimized mean-squared error [Eq. (2)] against the ground-truth coordinates. This combination yielded stable convergence and accurate plate localization in practice.

$$L(y,p) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right], \quad (1)$$

where,

- $L(y,p)$ represents the log loss function.
- N is the number of samples.
- y_i is the true label of the i-th sample.
- p_i is the predicted probability of the i-th sample being of the positive class.

$$L(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\|y_i - \hat{y}_i\|^2. \quad (2)$$

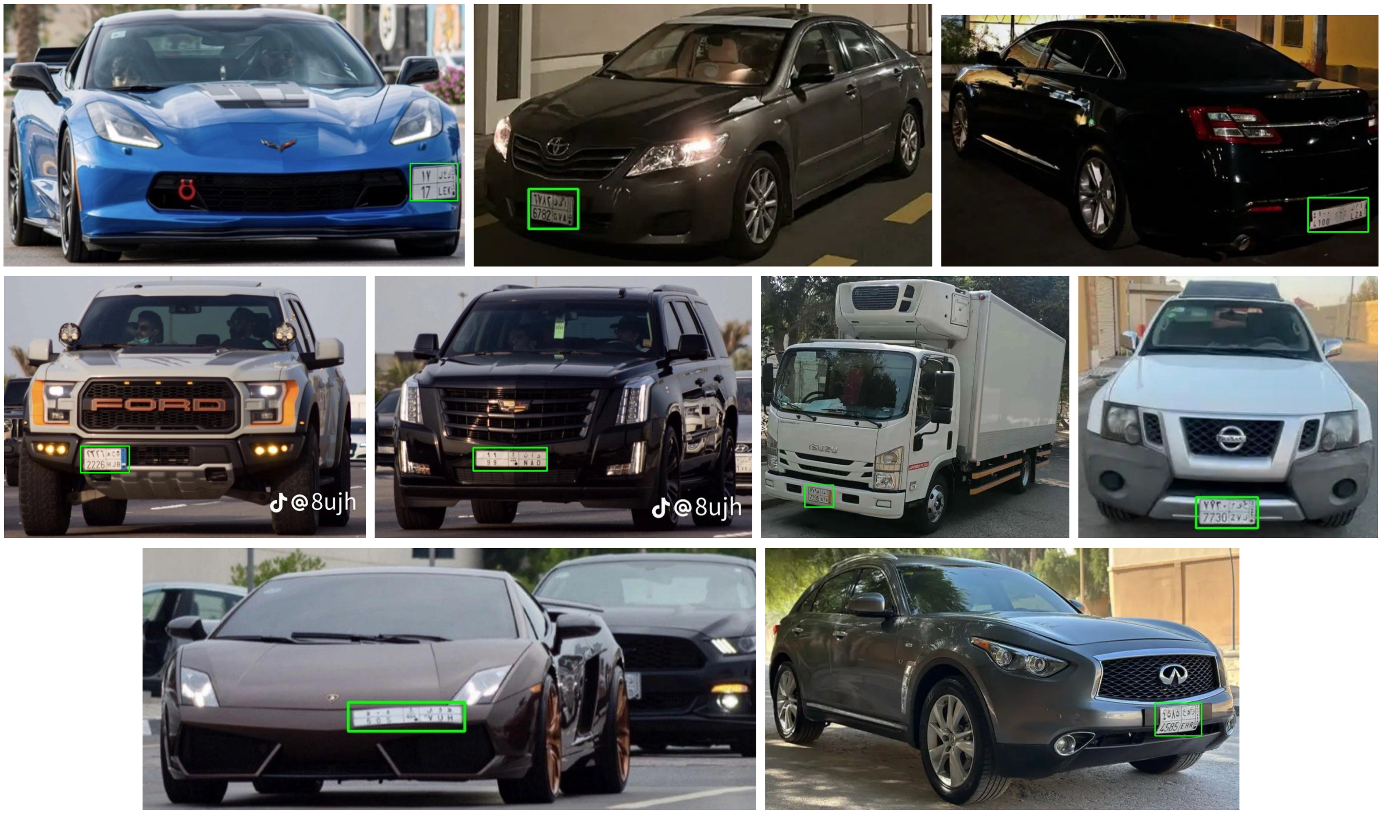

Fig. 4. Examples from our Saudi license-plate detection dataset (17,000 images), covering diverse backgrounds, lighting conditions, scales, viewpoints, and all official plate types. Green boxes indicate detections from the retrained MTCNN detector.

F. License Plate Recognition

Following license-plate detection, character recognition is performed using LPRNet [22], which has been retrained specifically on Saudi Arabian plates. Prior to training, the label space was constrained to reflect national plate conventions: strings were limited to at most seven characters (up to three letters and up to four digits), and characters not present on Saudi plates—C, F, I, M, O, P, Q, W, and Y—were removed. This pruning reduces ambiguity and improves convergence without altering the inference pipeline.

Because a large open dataset for Saudi plates is unavailable, we assembled a dedicated corpus. Data were drawn from three sources: 1) self-captured images, 2) publicly available social-media imagery, and 3) crops automatically harvested by our retrained MTCNN plate detector, which we used to detect and crop Saudi plates to expand further coverage across styles, viewpoints, and illumination conditions. Fig. 5 provides samples of our novel dataset alongside LPRNet predictions.

Annotation was streamlined with a lightweight Python tool that renames each image to its alphanumeric ground truth. Given the fixed mapping between Arabic and English characters on Saudi plates, labels were stored in English to simplify recognition. In total, the recognition dataset comprises approximately 35,000 labeled plate images. The curated dataset (images and ground-truth labels) is documented for reproducibility and is publicly available via our GitHub repository cite [27]. After collecting and pre-processing the dataset, we retained the LPRNet model, with key hyperparameters summarized in Table III. We achieved an overall accuracy of 93.1%, with a loss of 0.0059 at epoch 25, as shown in Fig. 6.

TABLE III. TRAINING HYPERPARAMETERS FOR THE LICENSE-PLATE RECOGNITION MODEL (RETRAINED LPRNET)

Parameter	Value
Number of epochs	50
Learning rate	0.001
Learning rate schedule	at epochs 4, 8, 12, 16, 32, and 45
Batch size	128
Dataset split	70% train, 20% validation, 10% test

G. Face Detection and Alignment

Within the vehicle region of interest, faces are localized using RetinaFace [23]. This single-shot detector jointly predicts bounding boxes and sparse facial landmarks (e.g., eye centers, nose tip, mouth corners). This joint formulation enhances robustness to scale changes, partial occlusions, and challenging illumination conditions, which are common in in-vehicle imagery. As illustrated in Fig. 7, the detected landmarks are subsequently used to geometrically normalize the crop, producing an upright, aligned face that is forwarded to the recognition backend. In practice, this alignment step reduces pose variability and stabilizes the downstream feature extraction stage.

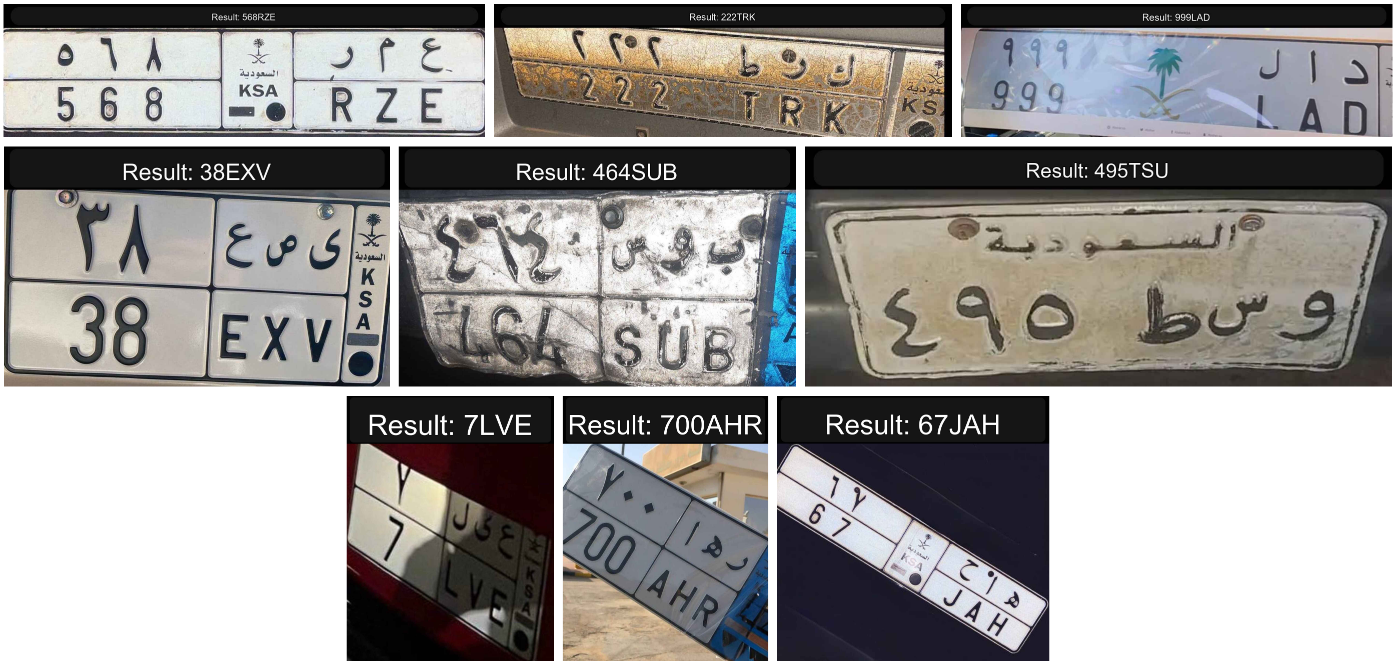

Fig. 5. Samples from the Saudi license-plate recognition dataset (35,000 images) encompass a diverse range of backgrounds, lighting conditions, scales, viewpoints, and official plate types. LPRNet predictions are shown above each plate.

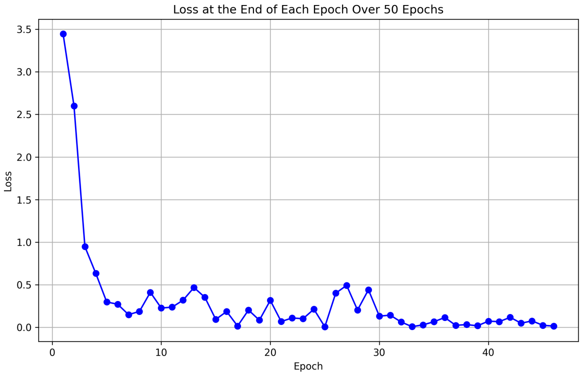

Fig. 6. Training loss of the retrained LPRNet across 50 epochs; the minimum loss (0.0059) occurs at epoch 25, coinciding with a 93.1% test accuracy on Saudi license plates.

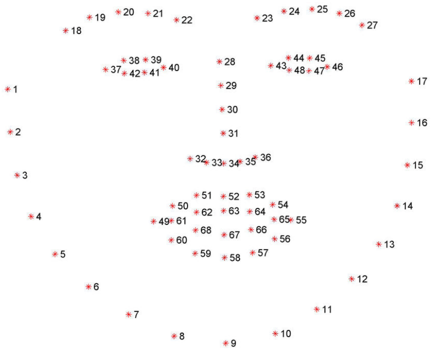

Fig. 7. RetinaFace-detected facial landmarks (eye centers, nose tip, mouth corners) and the resulting geometric alignment prior to face recognition.

H. Face Recognition

Following detection and geometric alignment (Fig. 7), identity is inferred using the InsightFace pipeline [24], trained with the ArcFace additive angular-margin loss [25], to obtain a compact and discriminative embedding for each detected face. To reduce computational cost and mitigate redundancy in the embedding space, dimensionality reduction methods such as principal component analysis (PCA) can be applied prior to classification [34]. The resulting feature vectors are classified using a one-vs-rest linear SVM [26], an approach chosen for its low latency, stable generalization in small-sample regimes, and straightforward score calibration in the tri-modal setting (see also related ensemble/OvR usage in [35]). Within our dataset of 30 subjects with 20 images each (600 total; 70/20/10 train/validation/test split), this module achieved $\approx 99.5\%$ identification accuracy on the closed set. The classifier's confidence score is exported to the fusion stage as the face term in the linear decision function, consistent with the end-to-end workflow illustrated in Fig. 1 and the global formulation described earlier.

I. Tri-Modal Score Fusion for End-to-End Access Classification

To convert per-modality predictions into a single access decision, the system fuses confidence scores from the vehicle make and model classifier, the license plate recognizer, and the face recognizer. Let the score vector be:

$$\boldsymbol{x} \triangleq \begin{bmatrix} x_{\text{vehicle}} \\ x_{\text{plate}} \\ x_{\text{face}} \end{bmatrix},$$

where, each component denotes the corresponding model's confidence. A linear support vector machine (SVM) [26] is employed to learn a separating hyperplane over $\boldsymbol{x}$, chosen for its computational simplicity in real-time settings and its stable behavior with limited training data.

The decision function is:

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\top}\boldsymbol{x} + b = w_1\, x_{\text{vehicle}} + w_2\, x_{\text{plate}} + w_3\, x_{\text{face}} + b, \quad (3)$$

with $\boldsymbol{w} = [\, w_1,\, w_2,\, w_3 \,]^{\top}$ and b learned from labeled training samples. Access is granted when

$$f(\boldsymbol{x}) \geq 0,$$

and denied otherwise. This linear fusion enables the classifier to weight each modality according to its discriminative value at the operating point, yielding a single, consistent pass/fail verdict suitable for real-time gate control.

III. Results and Discussion

In this section, we present the training of our tri-modal system and report its overall performance (quantitative results). We also provide a real-world demonstration of the proposed model (qualitative results).

A. Tri-Modal System Training and Performance

The tri-modal fusion classifier maps modality confidences—vehicle make/model, license plate recognition, and face recognition—-into a single access decision using a linear SVM operating on the score vector with decision function, as shown in Eq. (3). The training corpus comprises 1,000 images spanning a diverse range of vehicles, license plates, and driver appearances. The dataset was partitioned into 70%/15%/15% for training/validation/testing, respectively. Under this protocol, the fused classifier attained an overall accuracy of 97% with an end-to-end latency of 66 ms per frame (approximately 15.15 FPS). All experiments were conducted on an NVIDIA GeForce RTX 2080 Ti GPU. The learned parameters are $w_1 = 2.103$ (vehicle), $w_2 = 6.270$ (plate), $w_3 = 3.392$ (face), and $b = -7.588$; the separating hyperplane and score geometry are illustrated in Fig. 8.

The weight magnitudes follow an intuitive ordering. The license-plate stream receives the largest weight, reflecting the high distinctiveness of plate identities; the face stream is second—useful but occasionally ambiguous (e.g., similar appearance)—and the vehicle make/model stream is weighted lowest, consistent with the possibility of visually similar trims across brands. This ranking provides a straightforward interpretation of how the fusion balances complementary cues at inference time.

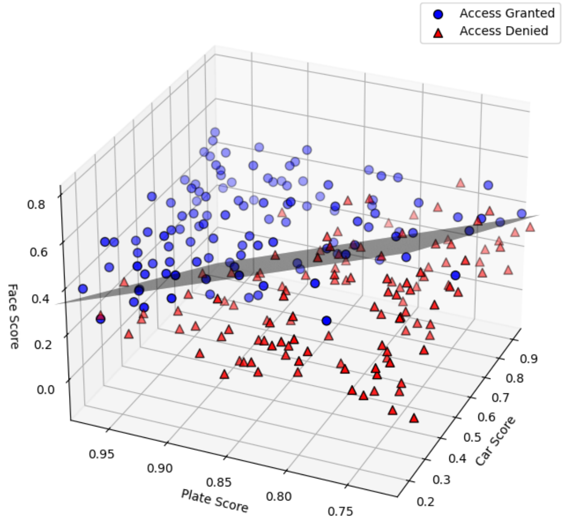

Fig. 8. Linear SVM decision boundary in the tri-modal confidence space.

At the system level, 66 ms/frame is compatible with near-stop gate operation, where vehicles either halt or approach slowly (< 15 km/h). Numerically, 15 km/h $\approx$ 4.17 m/s, so a 15.15 FPS stream observes frames every $\approx$ 66 ms; the vehicle advances only $\approx 4.17 \times \frac{1}{15.15} \approx 0.28$ meters between frames. This inter-frame motion is sufficiently small to maintain detector stability and OCR legibility as the vehicle settles at the barrier, while preserving temporal continuity for the face pipeline inside the cabin. The operating point, therefore, satisfies the real-time constraint for access control. The end-to-end frame time is directly determined by the per-module (subsystem) runtimes that execute in sequence. The speeds of these subsystems are summarized in Table IV.

TABLE IV. Per-Module Runtime and throughput (MS and FPS) for the Sequential Tri-Modal Pipeline—Vehicle Detection/Recognition, Plate Detection/Recognition, and Face Detection/Recognition—Along with the Overall End-to-End Latency

Model	Speed
Vehicle Detector	12ms ($\approx$ 83.33 FPS)
Vehicle Recognizer	9ms ($\approx$ 111.11 FPS)
License Plate Detector	11ms ($\approx$ 90.91 FPS)
License Plate Recognizer	9ms ($\approx$ 111.11 FPS)
Face Detector	15ms ($\approx$ 66.67 FPS)
Face Recognizer	10ms (100 FPS)
End-to-end	66ms ($\approx$ 15.15 FPS)

B. A Real-World Demonstration

To examine how the proposed tri-modal pipeline behaves outside controlled datasets, we recorded a 23-second video sequence at a gate-like entrance, as illustrated in Fig. 9. The scenario involves a Toyota Camry approaching the barrier at low speed, bearing the Saudi license plate 1812SGD and driven by an enrolled subject (Muhannad). The raw video stream is fed directly to the system described in Section II, with no manual frame selection or offline post-processing.

As the vehicle moves through the field of view, the YOLOv4-tiny detector produces a set of candidate bounding boxes B_t over successive frames. The best-frame selection module (Subsection II-C) evaluates $\mathrm{IoU}(B_t, B^*)$ with respect to a fixed reference box B^* centred in the image, and selects the frame $t^{\dagger}$ that maximizes this overlap. In the recorded sequence, the chosen frame corresponds to the instant when the Camry is nearly frontal and occupies most of the reference region, as shown in Fig. 10. The resulting bounding box tightly

Fig. 9. Input video sequence recorded at a gate-like entrance; click the image (or the link) to view the full clip.

encloses the vehicle, suppressing background clutter and providing a clean region of interest (ROI) for the downstream modules. This step is consistent with the low-latency strategy adopted in the proposed model, where only a single, high-quality frame is propagated to the rest of the pipeline.

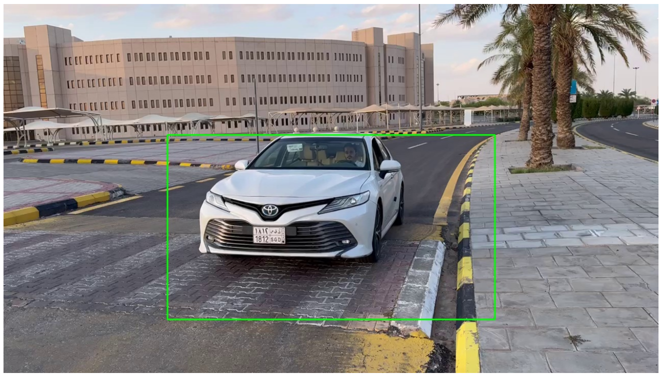

Fig. 10. Best frame selected by the IoU-based gating module, showing the detected vehicle together with the reference bounding box.

Once the best frame is fixed, the corresponding vehicle crop is forwarded to the MobileNetV2-based make–model classifier described in Subsection II-D. In this experiment, the classifier correctly identifies the vehicle as a Toyota Camry with a confidence score of 0.9417, as depicted in Fig. 11. This result aligns with the high accuracy reported for the vehicle-recognition component under real-time operating conditions.

The same ROI is then passed to the license plate branch. The retrained MTCNN detector localizes the Saudi plate with a precise bounding box, and LPRNet decodes the alphanumeric content. The predicted string matches the ground-truth plate number 1812SGD with a confidence of 0.97 (Fig. 12), demonstrating that the dataset and retraining strategy introduced in Subsection II-E and Subsection II-F generalize well to real capture conditions.

In parallel, the face-analysis branch operates inside the vehicle ROI. RetinaFace searches for a driver, estimates facial landmarks, and produces an aligned face crop, as outlined in Subsection II-G. InsightFace then maps this crop into a feature embedding, which is classified by the one-vs-rest linear SVM trained on enrolled personnel (Subsection II-H). In the recorded sequence, the system correctly recognizes the driver as Muhannad, but with a relatively modest confidence score of 0.35 (Fig. 13). This lower margin is consistent with the challenging imaging conditions: the face is partially occluded by the windshield, affected by sunlight reflections, and drawn from a comparatively small in-cabin dataset that does not yet fully capture these variations.

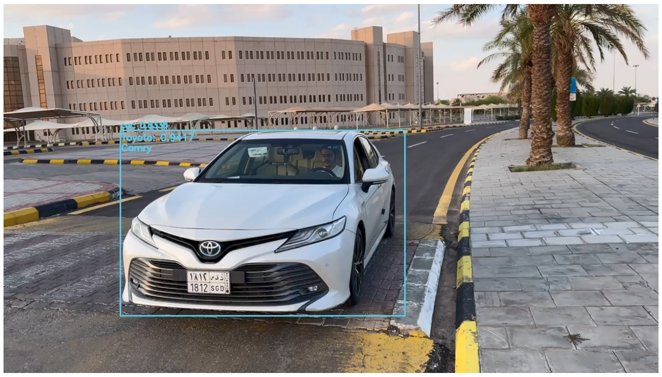

Fig. 11. Output of the vehicle make–model classifier, correctly identifying the car as a Toyota Camry with a confidence score of 0.9417.

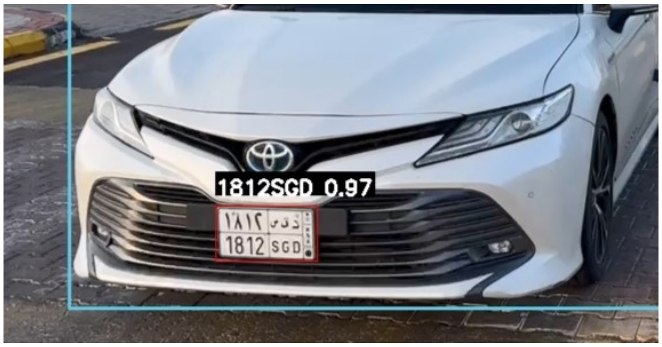

Fig. 12. License-plate recognition result for the selected frame, correctly decoding the plate number 1812SGD with a confidence score of 0.97.

Although the confidence of the face recognizer is low, it only contributes partially to the final access decision, since the tri-modal SVM assigns it a smaller weight whenever the vehicle and license-plate cues are more reliable. This demonstrates the robustness of our model, as it does not rely on a single feature; instead, it fuses three heterogeneous features to make a coherent vehicle-access decision, thereby reducing the impact of occasional failures in any individual modality. Additionally, the face recognizer can be improved by automatically collecting more in-the-wild data each time a person passes through the gate, particularly under varying lighting, weather, and viewing conditions. A growing set of face crops—captured at different times of day, with different windshield states and head poses—can then be used to periodically retrain the face-recognition branch, making it more tolerant to glare, occlusion, and motion blur. Over time, this continuous update loop helps close the gap between facial and non-facial cues and stabilizes the fused decision in real-world gate-control deployments.

The three scalar confidences from the previous stages:

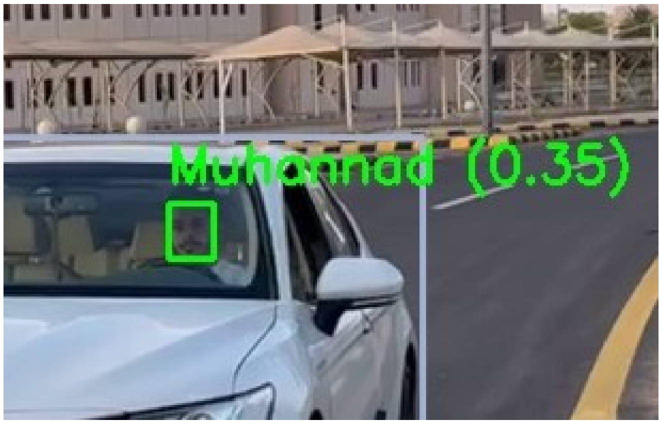

Fig. 13. In-vehicle face detection and recognition result, identifying the enrolled driver (Muhannad) with a confidence score of 0.35.

$$x_{\text{vehicle}} = 0.9417, \quad x_{\text{plate}} = 0.97, \quad x_{\text{face}} = 0.35,$$

are then combined by the tri-modal fusion SVM introduced in Subsection II-I and trained, as detailed in Subsection III-A. Using the learned parameters:

$$(w_1, w_2, w_3) = (2.103,\ 6.270,\ 3.392), \quad b = -7.588,$$

the decision function in Eq. (3) yields a positive fused score $f(x)$, and the system grants access. This outcome is consistent with the ground-truth label for the scenario (authorized vehicle and authorized driver). A representative frame from the processed video shows the intermediate decisions (vehicle class, plate text, and identity label) overlaid alongside the final "access granted" indicator, as shown in Fig. 14. You can click on the figure to watch the processed video or on the link provided in the figure caption.

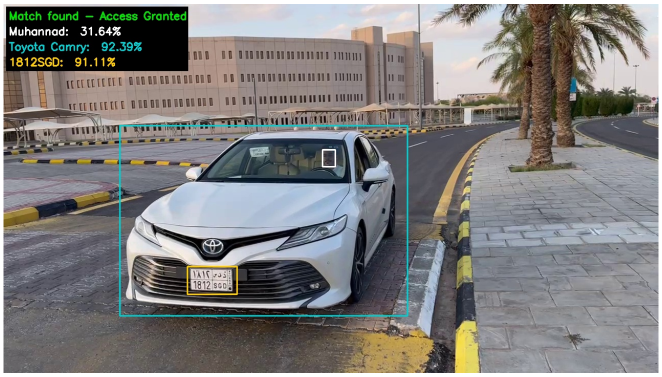

Fig. 14. Combined output of the proposed tri-modal system for the test sequence, including vehicle class, license-plate text, driver identity, and the final "access granted" decision; click the image (or the link) to view the full processed video.

Finally, this qualitative demonstration highlights several aspects of the proposed model working together in practice. First, the best-frame selection mechanism reduces redundant computation while still capturing a view in which all three modalities—vehicle appearance, license plate, and face—are informative. Second, the fusion stage naturally compensates for the lower face-recognition confidence by assigning higher weight to the more distinctive license-plate and make–model scores, as reflected in the relative magnitudes of w_2 and w_1. Third, when combined with the measured end-to-end latency of 66 ms per frame (approximately 15.15 FPS) reported in Subsection III-A, the experiment confirms that the full tri-modal pipeline can operate in real time at a physical gate, where vehicles approach slowly and can be stopped or released based on a single, coherent anti-spoofing decision.

C. Comparison with the Tri-Modal Benchmark

As summarized in Table I, vision-based gate systems typically rely on one or two cues—most often the license plate, sometimes combined with vehicle appearance or driver information. Only a small number of works attempt a full tri-modal design. Among these, the study by Saadouli *et al.* [18] is particularly relevant, as it also fuses vehicle, license plate, and face information to control an electronic gate.

In their framework, car make and model are recognized from handcrafted features based on Difference-of-Gaussians and SIFT descriptors, while license plates are processed with connected-components analysis and OCR; driver verification relies on a Viola–Jones face detector. The system is evaluated on a small surveillance dataset collected at Qatar University (225 images, 24 vehicle types), where the make–model subsystem attains an accuracy of approximately 74.6%. *Although this work demonstrates that combining the three modalities is feasible, it remains constrained by classical computer vision methods, limited data diversity, and a processing time of approximately two seconds per vehicle in the prototype.*

The tri-modal system proposed in this paper revisits the same high-level objective but with a modern deep-learning pipeline and a significantly larger, region-specific data foundation. Vehicle detection is handled by YOLOv4-tiny, vehicle make–model recognition by a MobileNetV2-based classifier trained on roughly 400 brands and 7 000 models, and Saudi license plates are processed by retrained MTCNN and LPR-Net modules using newly curated detection and recognition datasets (17 000 and 35 000 images, respectively). Faces are localized with RetinaFace and recognized via InsightFace embeddings with a linear SVM backend. All three confidence scores are then fused through a learned linear SVM, which automatically assigns a higher weight to the more distinctive license-plate and face streams while still exploiting the vehicle signature.

Quantitatively, the proposed gate-level classifier reaches 97% overall accuracy with an end-to-end latency of 66 ms per frame (approximately 15.15 FPS), which is compatible with real-time deployment at a physical barrier. By contrast, Saadouli *et al.* report an accuracy of approximately 74.6% on their make–model benchmark and a per-vehicle decision time of about two seconds in the prototype. *In practical terms, our system delivers both higher recognition performance and substantially lower latency under more challenging and diverse operating conditions.*

Another important distinction lies in reproducibility. The datasets used in [18] are institution-specific and not publicly distributed, which makes direct comparison and follow-on studies difficult. In this work, both Saudi license-plate datasets, together with the implementation of all modules and the tri-modal fusion logic, are made openly available through a public repository, enabling independent validation and adaptation to other gate environments. *Overall, the proposed system can therefore be viewed as a deep-learning-based, real-time extension of the earlier tri-modal concept, closing much of the gap between proof-of-concept prototypes and deployable anti-spoofing vehicle gate solutions.*

IV. Conclusion

This work introduced TRI-GATE, a tri-modal, AI-based anti-spoofing system that integrates vehicle, license plate, and facial information into a unified framework for secure, real-time gate access. Unlike traditional gate setups that depend on human guards or on a single recognition cue, TRI-GATE leverages complementary visual signals and fuses them using a linear SVM at the score level, thereby reducing the likelihood that a single point of failure—such as a cloned plate or a partially occluded face—can be exploited to gain unauthorized entry.

At the subsystem level, each component is tailored for practical deployment in Saudi and GCC environments. YOLOv4-tiny and a MobileNetV2-based make–model classifier deliver accurate vehicle detection and recognition in real time. A retrained MTCNN paired with LPRNet operates on two newly curated Saudi plate datasets (17,000 images for detection and 35,000 for recognition), achieving 99.2% detection accuracy and 93.1% recognition accuracy, respectively. For driver identity, RetinaFace and InsightFace embeddings combined with a linear SVM reach approximately 99.5% accuracy on the collected subject set. These modules are tied together by an IoU-based best-frame selection strategy that forwards only a single, high-quality frame, enabling the overall system to maintain an end-to-end latency of 66 ms per frame, or approximately 15.15 FPS, under realistic gate conditions.

Beyond component-level metrics, TRI-GATE was evaluated as a complete gate-control solution. On a tri-modal dataset of 1,000 images, the fused classifier achieves 97% accuracy, utilizing learned fusion weights that naturally emphasize license plate and face cues while still leveraging the discriminative power of vehicle appearance. A real-world demonstration involving an approaching vehicle confirmed that the system can reliably select the best frame, recognize the vehicle and its license plate, identify the enrolled driver, and issue a coherent "access granted" decision in real-time. Compared with prior tri-modal work built on handcrafted features and slower processing pipelines, TRI-GATE offers both higher accuracy and substantially reduced decision time, moving the tri-modal concept from prototype-level feasibility to deployable practice.

Future work will focus on turning TRI-GATE into a more general and deployment-ready platform. A first step is to expand the training datasets with additional subjects, wider geographic coverage, night-time recordings, and more varied weather and windshield conditions. This should help reduce overfitting to specific sites and make the face and vehicle modules more stable in day-to-day operation. A second step is to incorporate open-set recognition and anomaly detection, allowing the system to explicitly flag vehicles or drivers that do not belong to the enrolled population, rather than forcing a hard decision.

In parallel, we plan to investigate privacy-aware data handling, secure storage, and auditable logging, so that TRI-GATE can better align with regulatory and organizational requirements in sensitive infrastructures. Finally, extending the framework to multi-lane, high-throughput checkpoints and optimizing it for edge or embedded hardware would further increase its practical usefulness and scalability. Together, these directions outline a natural path from the current prototype toward a robust, multimodal gate access solution suitable for long-term field deployment.

Acknowledgment

The authors gratefully acknowledge Qassim University, represented by the Deanship of Graduate Studies and Scientific Research, on the financial support for this research under the number (QU-J-PG-2-2025-56804) during the academic year 1446 AH / 2024 AD.

References

[1] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proceedings of the European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11216. Munich, Germany: Springer, Cham, 2018, pp. 593–609.

[2] S. Zhu, Y. Wang, and Z. Wang, "A lightweight license plate detection algorithm based on deep learning," *IET Image Processing*, vol. 18, no. 2, pp. 403–411, 2024.

[3] A. Sarhan, R. Abdel-Rahem, B. Darwish, A. Abou-Attia, A. Sneed, S. Hatem, A. Badran, and M. Ramadan, "Egyptian car plate recognition based on YOLOv8, Easy-OCR, and CNN," *Journal of Electrical Systems and Information Technology*, vol. 11, 2024.

[4] L. Tao, S. Hong, Y. Lin, Y. Chen, P. He, and Z. Tie, "A real-time license plate detection and recognition model in unconstrained scenarios," *Sensors*, vol. 24, no. 9, p. 2791, 2024.

[5] Y. Lou, Y. Bai, J. Liu, S. Wang, and L.-Y. Duan, "VERI-Wild: A large dataset and a new method for vehicle re-identification in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 3235–3243.

[6] Z. Yu, Z. Huang, J. Pei, L. Tahsin, and D. Sun, "Semantic-oriented feature coupling transformer for vehicle re-identification in intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 3, pp. 2803–2813, 2024.

[7] W. Zhu, Z. Wang, X. Wang, R. Hu, H. Liu, C. Liu, C. Wang, and D. Li, "A dual self-attention mechanism for vehicle re-identification," *Pattern Recognition*, vol. 137, p. 109258, 2023.

[8] Y. Liang, Y. Gao, and Z. Y. Shen, "Transformer vehicle re-identification of intelligent transportation system under carbon neutral target," *Computers & Industrial Engineering*, vol. 185, p. 109619, 2023.

[9] J. Lian, D.-H. Wang, Y. Wu, and S. Zhu, "Multi-branch enhanced discriminative network for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 2, pp. 1263–1274, 2024.

[10] F. Alim, E. Kavakli, S. B. Okcu, E. Dogan, and C. Cigla, "Simultaneous license plate recognition and face detection at the edge," in *Proceedings of SPIE 12542, Real-Time Image Processing and Deep Learning*. San Francisco, CA, USA: SPIE, Jan. 2023, p. 1254204.

[11] B. Teja, S. S. Raju, R. Bhuvaneswari, and G. U. Stella, "Automatic vehicle access control into the residential areas using face detection and ANPR technologies," in *Proceedings of the IEEE Technology & Engineering Management Conference - Asia Pacific (TEMSCON-ASPAC)*. IEEE, 2023, pp. 1–6.

[12] F. Akbar, R. P. Santi, and A. F. Ramadhan, "Two-factor authentication using face and number-plate recognition for a secure campus entry system," in *Proceedings of the 2nd International Symposium on Information Technology and Digital Innovation (ISITDI)*. IEEE, 2024, pp. 191–195.

[13] H. U. Iyer and S. Dhavale, "Automatic number plate and face recognition system for secure gate entry into military establishments," in *Proceedings of the International Conference on Smart Systems for Applications in Electrical Sciences (ICSSES)*. IEEE, 2024, pp. 1–8.

[14] T. Mustafa and M. Karabatak, "Real time car model and plate detection system by using deep learning architectures," *IEEE Access*, vol. 12, pp. 107 616–107 628, 2024.

[15] Á. Ramajo-Ballester, J. M. A. Moreno, and A. de la Escalera Hueso, "Dual license plate recognition and visual features encoding for vehicle identification," *Robotics and Autonomous Systems*, vol. 172, p. 104608, 2024.

[16] Y. L. Khor, Y. C. Chang, Y. J. Wong, B.-H. Kwan, M.-L. Tham, and K.-C. Khor, "Multi-task YOLO for vehicle colour recognition and automatic license plate recognition," in *Proceedings of the IEEE Conference on Emerging Applications of Information Systems (EAIS)*. IEEE, May 2024, pp. 1–8.

[17] N. AlDahoul, M. J. T. Tan, R. R. Tera, H. A. Karim, C. H. Lim, M. K. Mishra, and Y. Zaki, "Multitasking vision language models for vehicle plate recognition with VehiclePaliGemma," *Scientific Reports*, vol. 15, 2025.

[18] G. Saadouli, M. I. Elburdani, R. M. Al-Qatouni, S. Kunhoth, and S. Al-Maadeed, "Automatic and secure electronic gate system using fusion of license plate, car make recognition and face detection," in *Proceedings of the International Conference on Internet of Things (ICIoT)*. IEEE, Feb. 2020, pp. 1–7.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[21] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[22] S. Zherzdev and A. Gruzdev, "LPRNet: License plate recognition via deep neural networks," *arXiv preprint arXiv:1806.10447*, 2018.

[23] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 5203–5212.

[24] J. Guo and J. Deng, "InsightFace: 2D and 3D face analysis project," GitHub repository, apr 2023, version v0.7. [Online]. Available: https://github.com/deepinsight/insightface

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[27] M. Alsultan, T. Alghonaim, A. Alorf, B. Alwazzan, F. Alsakakir, A. Alhassan, and Y. Hussain, "TRI-GATE: A tri-modal anti-spoofing system for gate access using vehicle, license plate, and face recognition," GitHub repository, Nov. 2025, accessed: Nov. 8, 2025. [Online]. Available: https://github.com/SIGNALinLab/TRI-GATE-A-Tri-Modal-Anti-Spoofing-System-for-Gate-Access.git

[28] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[29] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[30] J. Ribeiro, "Car make and model classification example with YOLOv3 object detector," GitHub repository, 2019, accessed: 2025-11-02. [Online]. Available: https://github.com/josesaribeiro/car-make-model-classifier-yolo3-python

[31] O. Bourja, A. Maach, Z. Zannouti, H. Derrouz, H. Mekhzoum, H. A. Abdelali, R. O. H. Thami, and F. Bourzeix, "End-to-end car make and model classification using compound scaling and transfer learning," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 5, pp. 994–1001, 2022.

[32] W. Wang, J. Yang, M. Chen, and P. Wang, "A light CNN for end-to-end car license plates detection and recognition," *IEEE Access*, vol. 7, pp. 173 875–173 883, 2019.

[33] A. Graf, A. Koch-Kramer, L. Lindqvist, and E. M. Kalinowski. (2025) Instaloader — download instagram photos and metadata. Instaloader Documentation, version 4.14.2. Accessed: 2025-11-01. [Online]. Available: https://instaloader.github.io/

[34] A. Alorf, "Performance evaluation of the PCA versus improved PCA (IPCA) in image compression, and in face detection and recognition," in *Proceedings of the Future Technologies Conference (FTC)*, 2016, pp. 537–546.

[35] B. Kanawade, J. Surve, S. R. Khonde, S. P. Khedkar, J. R. Pansare, B. Patil, S. Pisal, and A. Deshpande, "Automated human recognition in surveillance systems: An ensemble learning approach for enhanced face recognition," *Ingénierie des Systèmes d'Information*, vol. 28, no. 4, pp. 877–885, 2023.