Comprehensive Multi-Modal Analysis for Enhanced Road Safety and Traffic Law Enforcement

Kesanam Ashinee, Simhadri Sevitha, Renukasakshi V Patil, and Dr. Ram Mohana Reddy Guddeti

National Institute of Technology Karnataka, Surathkal
ashineekesanam@gmail.com,renukasakshivpatil@gmail.com,munnysimhadri5544@gmail.com
NH 66, Srinivasnagar, Surathkal, Mangalore, Karnataka - 575 025

Abstract. This research presents a comprehensive approach to enhancing road safety and Traffic Law enforcement through advanced multi-modal image analysis. The methodology encompasses three key components: Vehicle Traffic Signal Violation Detection, Pedestrian Counting, and Driver Identification, with an extension for Vehicle License Plate Recognition. When a vehicle violates traffic signal laws, the system intelligently assesses the situation. Initially, it checks for the visibility of the driver's face, and if obscured, it proceeds to detect and recognize the vehicle's license plate. Furthermore, pedestrian counting near zebra crossings is employed to ensure the safety of road users. The methodology leverages state-of-the-art object detection and recognition models, including Easy OCR, and YOLOv3, for robust and accurate results. The integration of these components creates a powerful system for traffic signal violation detection, pedestrian safety assessment, and driver identification.

Keywords: Traffic Signal Violation, Pedestrian Counting, License Plate Recognition, Object Detection, Image Analysis, Easy OCR, YOLOv3

1 Introduction

The enforcement of traffic regulations has long been a significant concern for public safety and law enforcement agencies. With the increasing volume of vehicles on the roads, monitoring and controlling traffic violations have become paramount in ensuring road safety and adherence to legal mandates. In light of this pressing need, our research endeavors to introduce an advanced system that combines the realms of vehicle traffic signal violation detection, driver identification, and pedestrian safety assurance.

Our primary motivation for this research is to address a critical gap in law enforcement. The proposed system seeks to revolutionize traffic regulation enforcement by harnessing the power of cutting-edge technologies. Through this endeavor, we aim to enhance the ability to monitor and act upon traffic signal violations, thereby contributing to a safer and more orderly traffic environment.

The methodology employed is multifaceted, comprising a sequence of technologies, including YOLOv3 and Easy OCR. When a vehicle infringes upon traffic signal regulations, our system engages in a layered analysis. First, it checks for the visibility of the driver's face, ensuring compliance with regulations regarding the driver's presence and attention. If the driver's face is obscured, the system seamlessly transitions to license plate detection and recognition, allowing for efficient vehicle and driver identification. Furthermore, the system also counts pedestrians in the vicinity during signal violations, aiming to prioritize their safety.

Key components of this proposed system include:

1. **Traffic Signal Violation Detection:** Utilizing computer vision technology, the system identifies vehicles that breach traffic signal regulations. This detection is a crucial step towards improving road safety by discouraging reckless behavior and promoting adherence to traffic rules. [2] [10]

2. **Driver Face Identification:** When a vehicle violating a traffic signal is captured with its driver's face visible, the system employs facial recognition techniques[9] to match the driver with a pre-existing database of driver information. This allows for swift identification and retrieval of pertinent driver details, such as license information and traffic violation history. [1] [5]

3. **Vehicle License Plate Recognition:** In cases where the driver's face is not visible, the system switches to license plate recognition technology. By recognizing the vehicle's license plate, it accesses information associated with the vehicle and its owner, aiding law enforcement. [7] [8]

4. **Pedestrian Count:** To enhance overall road safety, the system also incorporates pedestrian counting capabilities, especially around zebra crossings near traffic signals. Accurate counting of pedestrians helps in optimizing traffic signal timing and ensures safe passage for walkers, reducing the risk of accidents.

The integration of these functions into a unified system offers law enforcement agencies a powerful tool to curb traffic violations effectively, enhance road safety, and promote responsible driving behavior. An extension for Vehicle License Plate Recognition further strengthens the capabilities of the system. When a vehicle is detected violating traffic signal regulations, the system employs a smart approach. It first verifies the visibility of the driver's face; if detected then recognizes the driver but if the face is not detected, it proceeds to detect and recognize the vehicle's license plate, facilitating the identification process. Moreover, pedestrian counting is utilized near zebra crossings to ensure the safety of pedestrians. This paper details our methodology and presents the results of its application, underscoring the potential impact on road safety and traffic regulation enforcement.

2 Related Work

1. **"Intelligent Traffic Violation Detection"**

Paper by Roopa Ravish, Shanta Rangaswamy, and S. N. Omkar proposes an algorithm called YOLOv3 to detect traffic violations using Artificial Intelligence and deep learning concepts. The authors propose the YOLOv3 algorithm to detect traffic violations such as jumping red signals, riding vehicles without helmets, driving without seat belts, and vehicles stepping over the stop line during red signals. The YOLOv3 algorithm uses Convolutional Neural Networks (CNN) to detect an object and Darknet-53 as a feature extractor. The paper discusses the implementation details, results, and analysis of the proposed algorithm. The authors claim that the algorithm offers a precision of 97.67% for vehicle identification and an exactness of 89.24% to identify the speed of a moving vehicle. [6]

2. **"A novel Squeeze YOLO-based real-time people counting approach"**
Paper by Peiming Ren, Lin Wang, Wei Fang* and Shulin Song addresses the critical task of real-time people counting in the context of smart cities and intelligent building systems. The paper introduces the YOLO (You Only Look Once) framework as an effective real-time object detection [3]approach, emphasizing its advantages in accuracy and speed. To tackle the challenge of large model sizes, the authors propose Squeeze YOLO-based people counting (S-YOLO-PC), which optimizes the network structure using the fire layer of SqueezeNet. This optimization significantly reduces model parameters while maintaining high accuracy. S-YOLO-PC is presented as a promising solution for practical applications in intelligent systems.

3. **"License Plate Recognition System Based on Improved YOLOv5 and GRU"**
Paper by Hengliang Shi and Dongnan Zhao presents an end-to-end deep learning model for license plate location and recognition in natural scenarios. The proposed model is based on the YOLOv5-LSE architecture and includes an improved channel attention mechanism to enhance accuracy and speed. The model is trained on a large-scale dataset of license plate images and tested on the CCPD dataset. The experimental results show that the proposed method outperforms other state-of-the-art license plate recognition algorithms in terms of accuracy and efficiency. The paper also includes a detailed comparative analysis with other cutting-edge recognition algorithms.

4. **"Face Transformer for Recognition"**
In paper by ,Yaoyao Zhong and Weihong [11] Deng explore the application of Transformers in face recognition, extending beyond their prevalent use in NLP and delving into computer vision. They address the question of whether Transformers can outperform CNNs. The authors modify the patch generation process to enhance inter-patch information, utilizing sliding patches with overlaps. Training on CASIA-WebFace and MS-Celeb-1M databases and evaluating on prominent benchmarks, their Face Transformer models exhibit comparable performance to CNNs with similar parameters and MACs. The paper contributes to the field by providing openly accessible Face Transformer models and codes, for further research at the intersection of Transformers and face recognition.

Table 1. State-of-the-art comparison table

Paper Name	Technique	Limitations	Acurracy
Squeeze YOLO-based real-time people counting	S-YOLO-PC	Comparison made with only three YOLO Techniques	72%
Intelligent Traffic Violation Detection	Yolo V3, CNNs	Traffic Violation Detection in late night	99.18%
"License Plate Recognition System	YOLOV5-LSE	Not applicable to embedded systems	96.94%
Face Transformer for Recognition	Transformers, CNNs	Employed on a dataset rather than Real-Time	95.96%

3 Problem Statement

The current state of traffic management is marked by widespread signal violations, limited means of swiftly identifying violators, and inadequate pedestrian safety monitoring. This research addresses these issues by developing an integrated traffic management system to detect signal violations, identify drivers, and count pedestrians for enhanced road safety and responsible driving behavior.

Objectives

– Develop a robust Traffic Signal Violation Detection System using computer vision and AI.
– Implement Facial Recognition for swift driver identification when the driver's face is visible.
– Integrate License Plate Recognition technology for identification when facial recognition is not possible.
– Create pedestrian counting mechanisms near zebra crossings to enhance pedestrian safety.

4 Proposed Work

4.1 Vehicle Traffic Signal Violation Detection

– **Data Collection:** We collected a diverse set of images featuring vehicles within a real-world traffic environment. These images served as the foundation of our database.
– **Object Detection:** Utilizing the YOLO (You Only Look Once) version 3 algorithm, we conducted object detection to identify and extract vehicles within the collected images. The extracted vehicle images were then organized and stored in a designated folder, constituting our database of vehicle images.

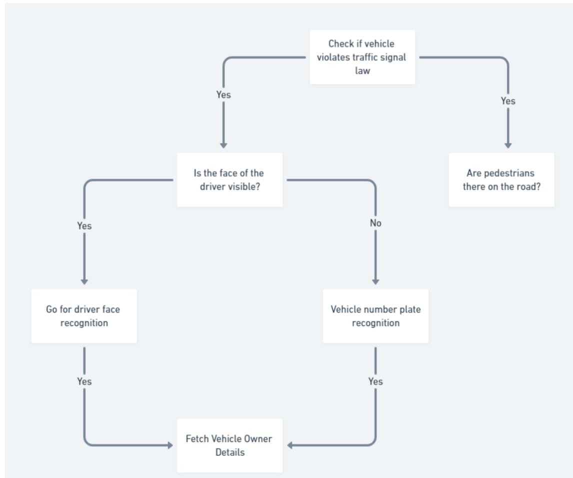

Fig. 1. Methodology

- **Defining Violation Boundaries:** In order to differentiate between vehicles following traffic signal regulations and those violating them, we established a boundary within each image. If a vehicle crosses this predefined boundary, it is considered to have committed a traffic signal violation.
- **Collection of Violation Images:** We also gathered a separate set of images that exemplify instances of vehicles violating traffic signal laws. These images were used to train and evaluate our detection system.
- **Hopcroft-Karp Algorithm:** The Hopcroft-Karp algorithm, a powerful method for bipartite graph matching, was utilized to efficiently search for matches between the features of the vehicles in violation images and those in the database. This algorithm helped us identify the vehicles involved in traffic signal violations. [Fig. 2]

4.2 Pedestrian Count

- **People Detection and Annotation with YOLOv3:** Utilize the YOLOv3 model to perform real-time object detection[4] on image frames from a specified source.Annotate the detected objects with bounding boxes and labels using the Supervision library.Refer [Fig3] This process helps identify and mark the pedestrians near the zebra crossing, classifying them as needed.Implement any additional steps or filtering criteria, such as focusing on a specific class (e.g., classid 0, representing pedestrians), to refine the annotations as required.Use polygon zones to define regions of interest (e.g., the zebra crossing area) for tracking pedestrians. Annotate frames accordingly using the defined polygon zones to highlight pedestrian movements in the specified area.Repeat these steps for multiple frames to ensure consistent and accurate annotation. Refer [Fig. 3]

```
FOR EACH violating_img_name IN violating_image_names:
    img2 = READ_IMAGE(violating_img_name)
    Initialize empty lists key1, key2, features
    max_val = 0
    max_img = None
    matchings = []

    FOR EACH img_name IN image_names:
        img1 = READ_IMAGE(img_name)
        img1, img2 = RESIZE_IMAGES(img1, img2, desired_width)
        key_pnt1, descrip1, key_pnt2, descrip2 = DETECT_AND_COMPUTE_FEATURES(img1, img2)
        matches = FEATURE_MATCHING(descrip1, descrip2)
        matchings.append(LENGTH(matches))
        IF LENGTH(matches) > max_val:
            max_val = LENGTH(matches)
            max_img = COPY_IMAGE(img1)
            features.append(matches)
            key1.append(key_pnt1)
            key2.append(key_pnt2)

    im1_gray, im2_gray = CONVERT_TO_GRAYSCALE(img1, img2)

    IF max_img is not None:
        DISPLAY_OUTPUT(max_img, key1[-1], img2, key2[-1], features[-1])
```

Fig. 2. Vehicle Traffic Signal Violation Detection Model Algorithm

4.3 Driver Face Identification

- **Data Collection and Preprocessing:** This experiment involves obtaining and using a diverse dataset of driver face images, capturing them under varying lighting conditions, angles, and levels of blurriness. The method initializes video capture and a face detection classifier using OpenCV. Users input a label (e.g., driver's name) for the collected facial data. The system continuously captures frames, detects faces, and stores cropped and resized facial images in the faces data list. The code checks for existing data storage files; if absent, it creates and saves files with the user's label and facial data. For existing files, the code appends new data, maintaining a comprehensive, labeled facial data database. Simultaneously, the Vision Transformer (ViT) model training uses a comprehensive dataset of labeled facial images from the web
- **Model training :** This experiment provides a methodology for face recognition and driver face tracking. It initiates by setting up the system to capture and detect the image using the OpenCV . Pre-trained facial data and associated labels are loaded from dataset and a K-Nearest Neighbors classifier is trained for face recognition. The code continuously processes the webcam feed, detecting faces and predicting identities. Recognized names and timestamps are used to mark the time , which is then logged in a file for record-keeping. To adapt this code for driver face recognition, driver names and their corresponding facial data can be stored in files, respectively. The code would then enable driver identification and tracking, making it a versatile tool for applications such as driver monitoring, access control, or security. [Fig. 4] Coming to the ViT model,it initiates the identification process by capturing the facial features of the driver. This step is crucial for establishing a foun-

```
FUNCTION process_frame(frame, _):
    results = DETECT_OBJECTS(frame)
    detections = FILTER_DETECTIONS(results)
    TRIGGER_ZONE(detections)
    frame = ANNOTATE_BOXES(frame, detections)
    frame = ANNOTATE_ZONE(frame)
    RETURN frame

FUNCTION transform_image(array):
    normalized_image = TRANSFORM_IMAGE(array)
    RETURN normalized_image
norm_image, unnorm_image = transform_image(test_output)

FUNCTION detect(network, data):
    pred = network(data)
    class_ids, scores, bounding_boxes = EXTRACT_PREDICTIONS(pred)
    RETURN class_ids, scores, bounding_boxes

FUNCTION count_object(network, class_ids, scores, bounding_boxes, object_label,
threshold=0.75):
    idx = GET_CLASS_INDEX(network, object_label)
    scores, class_ids = FILTER_DETECTIONS(scores, class_ids, idx, threshold)
    num_people = COUNT_PEOPLE(scores)
    RETURN num_people

FOR EACH object_label IN ["person", "sports ball"]:
    count = count_object(network, class_ids, scores, bounding_boxes, object_label)
    PRINT_COUNT(count, object_label)

FOR EACH threshold IN thresholds:
    num_people = count_object(network, class_ids, scores, bounding_boxes, "person", threshold)
    PRINT_COUNT(num_people, threshold)
```

Fig. 3. Pedestrain Count Model Algorithm

dation for subsequent comparisons and verifications. Once the facial data is obtained, it undergoes a meticulous recognition process using the processing model based on the Vision Transformer (ViT) architecture, specifically the ViT-B-16-plus-240 variant. The processing model is optimized for efficiency and speed, leveraging GPU acceleration through PyTorch. Then the comparison of facial features between the captured driver's face and the registered dataset, producing cosine similarity scores is done. We also observe that this model performs better.

4.4 Vehicle License plate Recogniton

- **License Plate Detection:** To extract license plate information from violating vehicles, the YOLOv5 model was implemented. This model effectively identified and localized license plates within the images.
- **License Plate Recognition:** For the recognition of license plate characters, we employed the Easy OCR system. It provided accurate text extraction from the detected license plates.
- **Driver Database Matching:** The extracted license plate numbers were then compared to the entries in the driver database, where driver information such as name, contact details, and license details were stored.
- **Driver Details Retrieval:** In cases of a successful match between the detected license plate and the database, driver details were retrieved, providing essential information about the vehicle's owner. [Fig. 5]

```
WHILE True:
   ret, frame = READ_FRAME(video)
   gray = CONVERT_TO_GRAY(frame)
   faces = DETECT_FACES(facedetect, gray, scaleFactor=1.3, minNeighbors=5)

   FOR EACH (x, y, w, h) IN faces:
      crop_img = CROP_IMAGE(frame, (x, y, x + w, y + h))
      resized_img = RESIZE_IMAGE(crop_img, (50, 50))

      IF LENGTH(faces_data) <= 100 AND i % 10 == 0:
         APPEND_TO_FACES_DATA(faces_data, resized_img)

      i = i + 1
      DRAW_TEXT(frame, str(LENGTH(faces_data)), (50, 50))
      DRAW_RECTANGLE(frame, (x, y), (x + w, y + h), color=(50, 50, 255))

   SHOW_FRAME("Frame", frame)
   k = WAIT_KEY(1)

   IF k == ord('q') OR LENGTH(faces_data) == 100:
      BREAK
```

Fig. 4. Driver Face Identification Model Algorithm

```
FOR EACH image_path in images
   yolo = LOAD_YOLO_MODEL(yolo_model_path)
   input_image = READ_IMAGE(image_path)
   results = yolo(input_image)

   ocr_reader = INIT_OCR_READER(ocr_reader_languages)
   output_image = PROCESS_RESULTS(results, input_image, ocr_reader, confidence_threshold)
   SAVE_OR_DISPLAY_OUTPUT(output_image, output_image_path)
```

Fig. 5. Vehicle License plate Recognitonn Model Algorithm

5 Experimental Results

5.1 Vehicle Traffic Signal Violation Detection

- The Hopcroft-Karp searches for matches between the features of the vehicles in violation images and those in the database
- The image with the highest count of matched features was considered the best match. The lines in the images shows the features matched by the feature matching Hopkroft Karp Algorithm Refer [Fig. 6]
- A visual representation of the matching was generated using Matplotlib, displaying the number of features matched in each case. Refer [Fig. 7]
- The overall accuracy is 74.5%, Comparison results are given in [Table 2]

5.2 Driver Face Identification

- By Utilizing the Vision Transformer (ViT) model (ViT-B-16-plus-240) with GPU acceleration via PyTorch, our experiment achieved a good performance in facial feature recognition. Comparison results are given in [Table 3]

Fig. 6. Vehicles matched by lines using feature matching Hopcroft Karp Algorithm.

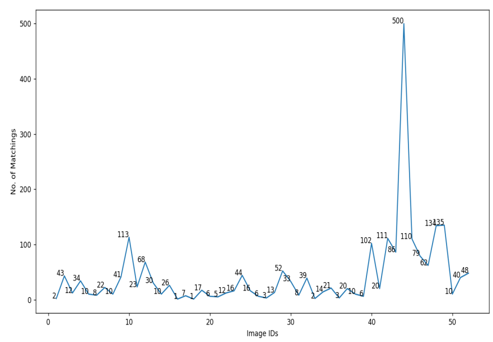

Fig. 7. Line plot to show the number of features matched

Table 2. Comparision between YoloV3 and Hopcroft-Karp integrated Yolo V3 models

Model Name	Accuracy	Precision
Yolo V3 model	0.701	0.81
Yolo V3 with Hopcroft Algorithm	0.745	0.89

– The ViT model showcased efficiency and speed, making our system a reliable tool for various applications.

– The model which used OpenCV and detected images in various diverse conditions enabling face recognition using the K-Nearest Neighbors classifier. This contributed to enhanced road safety and traffic law enforcement.

– It allows for versatile Driver Face Recognition. The system gives an approach for storage of driver names and facial data in necessary situations , facilitating applications such as driver monitoring, access control, and security.

Table 3. Comparision between Vision Transformer and KNN Classifier

Model Name	Accuracy	Precision
Vision Transformer	94.8	0.927
KNN Classifier	81.02	0.735

5.3 Pedestrian Counting

- Dataset was compiled from a wide array of internet images, including those captured in urban settings like busy streets, parks, and transit stations. It deliberately includes scenes with overlapping pedestrians and a range of crowd densities, mirroring the complexities of real-world urban environments.

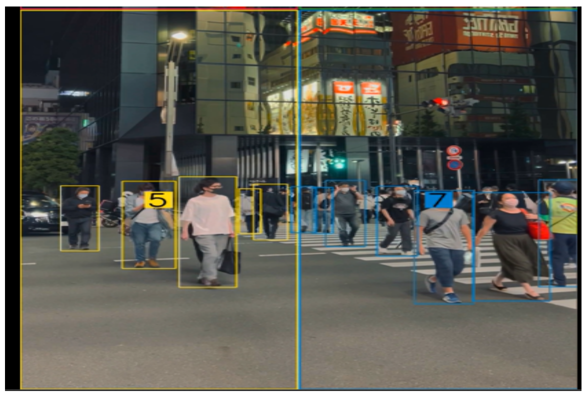

Fig. 8. Zebra Cross

- Yolo finds the person in the background with an amazing 86.7 confidence, despite the fact that they are fuzzy, as you can see above.This makes YOLO extremely popular despite its issues with overlapping objects.
- Some of the individuals are absent from Fig8.This is due to Yolo's subpar performance with overlapping objects.This is YOLO models' primary drawback.But because this algorithm is quick and memory-efficient, it's highly recommended.

5.4 Vehicle License PLate Recognition

- The model trained in 15 training epochs
- the model achieved a box loss of less than 10%, signifying its proficiency in localizing objects accurately within the images.
- the model attained an image loss of less than 2%, demonstrating its ability to minimize errors in image recognition and classification.
- The final output images featured license plate numbers annotated above the detected license plates. Refer [Fig. 9]

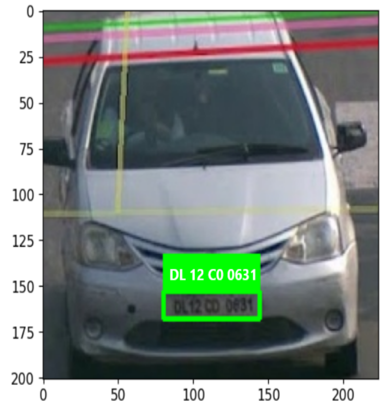

Fig. 9. License plate detected

-

6 Contributions

- A novel approach is being discussed in the paper that ensures law enforcement
- Enhanced YOLO v3 with Hopcroft Karp algorithm for traffic violation detection and it performs better compared to the base YOLO v3 model.
- The Vision Transformer (ViT) model demonstrated a better performance in facial recognition compared to the K-Nearest Neighbors classifier, achieving higher accuracy and precision in our Driver face recognition in diverse environments experiment.
- Additional to people counting in general crowded places here pedestrian counting is performed.To improvise the safety of the pedestrians.

7 Conclusion and Future Work

This paper introduces an integrated system for improving road safety and traffic law enforcement. It incorporates components such as Traffic Signal Violation Detection, Driver Identification, Vehicle License Plate Recognition, and Pedestrian Counting. This system combines advanced computer vision and AI technologies to detect violations and, when possible, identify drivers for more effective law enforcement.

In Future we aim to:

- **Video and Real-Time Implementation:** We plan to extend our models to process videos and implement real-time object recognition and tracking. This advancement will enable us to detect and track objects, such as vehicles and pedestrians,

– **Integration with CCTV Cameras:** We intend to integrate our models with real-time camera, we can actively monitor and enforce traffic regulations, identify violations, and contribute to improved traffic safety and law enforcement.

Acknowledgment

Our sincere thanks to Springer for publishing our paper. We extend our heartfelt gratitude to NIT Karnataka, Prof. Ram Mohan Reddy Guddeti for their invaluable guidance and support. Additionally, we appreciate the contributions of past researchers and the essential tools utilized in this research.

References

1. Lahiru Dinalankara. Face detection & face recognition using open computer vision classifies. *ResearchGate*, 2017.
2. DMBWDK Dissanayake, WBWMRUPU Aluvihare, KD Rajapaksha, et al. "smartgo"–intelligent traffic controlling system with violators detection. In *2022 4th International Conference on Advancements in Computing (ICAC)*, pages 304–309. IEEE, 2022.
3. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
4. Honglian Ma, Huchuan Lu, and Mingxiu Zhang. A real-time effective system for tracking passing people using a single camera. In *2008 7th World Congress on Intelligent Control and Automation*, pages 6173–6177. IEEE, 2008.
5. Farzana Arefin Nazira, Md Imriaz Uddin, Md Hafizuddin Raju, Shajnin Hossain, Md Naimur Rahman, and MF Mridha. Face recognition based driver detection system. In *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, pages 572–577. IEEE, 2021.
6. Roopa Ravish, Shanta Rangaswamy, and Kausthub Char. Intelligent traffic violation detection. In *2021 2nd Global Conference for Advancement in Technology (GCAT)*, pages 1–7. IEEE, 2021.
7. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
8. Hengliang Shi and Dongnan Zhao. License plate recognition system based on improved yolov5 and gru. *IEEE Access*, 11:10429–10439, 2023.
9. Gurpreet Singh, Ishika Gupta, Jaspreet Singh, and Navneet Kaur. Face recognition using open source computer vision library (opencv) with python. In *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pages 1–6. IEEE, 2022.
10. Aniruddha Tonge, Shashank Chandak, Renuka Khiste, Usman Khan, and LA Bewoor. Traffic rules violation detection using deep learning. In *2020 4th international conference on electronics, communication and aerospace technology (ICECA)*, pages 1250–1257. IEEE, 2020.
11. Yaoyao Zhong and Weihong Deng. Face transformer for recognition. *arXiv preprint arXiv:2103.14803*, 2021.