2023 25th International Multitopic Conference (INMIC) | 979-8-3503-1770-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/INMIC60434.2023.10466061

Enhancing Vehicle Entrance and Parking Management: Deep Learning Solutions for Efficiency and Security

Muhammad Umer Ramzan
Dept. of Computer Science
GIFT University
Gujranwala, Pakistan
umer.ramzan@gift.edu.pk

Usman Ali
Dept. of Computer Science
GIFT University
Gujranwala, Pakistan
usmanali@gift.edu.pk

Syed Haider Abbas Naqvi
Dept. of Computer Science
GIFT University
Gujranwala, Pakistan
191400106@gift.edu.pk

Zeeshan Aslam
Dept. of Computer Science
GIFT University
Gujranwala, Pakistan
191400122@gift.edu.pk

Tehseen, Husnain Ali
Dept. of Computer Science
GIFT University
Gujranwala, Pakistan
{191400138,191400118}@gift.edu.pk

Muhammad Faheem
Dept. of Computer Science
GIFT University
Gujranwala, Pakistan
mfaheem@gift.edu.pk

***Abstract*—The auto-management of vehicle entrance and parking in any organization is a complex challenge encompassing record-keeping, efficiency, and security concerns. Manual methods for tracking vehicles and finding parking spaces are slow and waste of time. To solve the problem of auto-management of vehicle entrance and parking, we have utilized state-of-the-art deep learning models and automated the process of vehicle entrance and parking into any organization. To ensure security, our system integrated vehicle detection, license number plate verification, and face detection and recognition models to ensure that the person and vehicle are registered with the organization. We have trained multiple deep-learning models for vehicle detection, license number plate detection, face detection, and recognition, however, the YOLOv8n model outperformed all the other models. Furthermore, License plate recognition is facilitated by Google's Tesseract-OCR Engine. By integrating these technologies, the system offers efficient vehicle detection, precise identification, streamlined record-keeping, and optimized parking slot allocation in buildings, thereby enhancing convenience, accuracy, and security. Future research opportunities lie in fine-tuning system performance for a wide range of real-world applications.**

***Index Terms*—object detection, deep learning, YOLOv8, Image Processing, OCR, Vehicle Parking**

I. Introduction

The auto-management of vehicle entry and parking in buildings poses challenges in terms of record keeping, time consumption, and security. Manual recording of vehicle information is time-consuming and makes it difficult for any organization to track vehicle movements effectively. Additionally, finding parking slots can be a time-consuming task. In the modern era, Artificial intelligence has multiple applications in different domains [1]. To address the vehicle entrance and parking issues, a comprehensive system is needed to detect and identify vehicles, ensuring authorized entry into the building. For this, we propose an integrated system that utilizes state-of-the-art deep learning techniques to detect the vehicle, the vehicle's licensed number plate, and driver face detection and recognition. Before the integration of the system, we trained multiple deep learning models including Convolutional Neural Network(CNN) on each task of object detection and recognition including vehicle detection and recognition, license number plate detection, and face detection and recognition. The results on each task show that the YOLOv8 (You Only Look Once) model outperforms all the other deep learning models. The YOLOv8 model is an advanced object detection framework that has gained significant attention in computer vision research in recent years. It is a deep learning-based architecture designed for real-time object detection tasks. YOLO operates by dividing an image into a grid and predicting bounding boxes and class probabilities within each grid cell [2].

Vehicle detection and recognition tasks from closed-circuit television (CCTV) footage are important when a vehicle reaches the gate of any organization. For this purpose, a customized dataset is created and labeled manually to train the model. Furthermore, datasets of Pakistan's licensed number plates are also created and labeled for the task of number plate detection to train the model for number plate detection. Once the number plate of the vehicle is detected, the next job is to extract the number from the image of the licensed number plate. For this purpose, we have utilized Google's Tesseract-OCR Engine [3] which extracts the characters from the image. This sub-system ensures that the vehicle belongs to our organization. However, the system should be more secure as someone might have access to the organization who does not belong to that organization anyone can drive a car that belongs to the organization and the system verifies it and gives

access by uplifting the barrier.
To solve this security issue, the vehicle verification system is combined with the driver verification system which will ensure that the vehicle entered into the organization with the registered driver. The driver verification system includes face detection and recognition, for face detection we have utilized the Haar Cascade classifier [4] for the face recognition task as the Haar Cascade classifier is a machine learning-based approach well-suited for detecting objects with distinct visual features, such as faces or eyes [5]. Furthermore, we utilized the DeepFace framework [6] to find the captured driver image from the employee database of the organization. The vehicle will only get access to the organization if the vehicle and the driver belong to the organization.
Once the registered vehicle along with the registered driver gets into the organization the next step is to park the vehicle at the vacant position which is time time-consuming task to find a free parking slot to park the vehicle. For this purpose, we have developed an Android-based mobile application that displays the live view of the parking area along with free and vacant parking slots. For this purpose, Image processing techniques such as edge detection, thresholding, contour detection, and morphological operations are utilized to identify parking slots based on visual cues [7]. By analyzing the characteristics of the parking area image, such as color, texture, and shape, parking slots can be differentiated from the surrounding environment or vehicles. To streamline record-keeping, a NoSQL database called Firebase is employed. This database is flexible, and scalable, and offers offline support, ensuring that data remains available even when the device is not connected to the internet. The manual process of vehicle and driver verification to get access to an organization is a slow and time-consuming task. Moreover, this process lacks the security [8] of the organization which may also lead to some fraud [9]. This paper addresses the issue of manual verification and entrance into an organization by automating the process of vehicle detection and verification and driver verification using deep learning techniques and proposes an integrated system that combines various technologies to efficiently manage vehicle detection, identification, driver verification, record-keeping, and vehicle parking automation in any buildings.

II. Related Work

The auto-verification of vehicle and parking systems consists of different sub-systems including vehicle detection and verification, number plate detection and recognition, driver's face detection, face recognition, and identification of free parking slots in the parking area. There are various methods developed for the implementation of each sub-system. A vehicle detection system is the process of developing advanced techniques like YOLOv5 to accurately detect, classify, and monitor vehicles in real-world traffic scenarios, aiming to enhance safety and contribute to automated driving applications [10]. YOLO framework is very famous for vehicle detection and classification systems. In complex urban environments, real-time object detection is a very difficult task, this problem can be solved by training multiple YOLOv5 models with different image sizes and applying ensemble techniques for faster detection in complex urban environments [2]. The deep learning model, specifically the YOLO v5 model can be used in intelligent transportation systems [11]. Wang et al. developed a lightweight YOLO v5-based network for surveillance video vehicle detection in intelligent transportation systems. By using MobileNetV2 and DSC, they achieved real-time and accurate detection with a 95% reduction in parameters, providing an efficient solution for highway and urban road safety [12]. Kaijie Zhang et al. proposed a YOLOv5-based vehicle detection method for accurate and real-time tracking, recognition, and counting. Their algorithm effectively handled challenging conditions such as heavy traffic, night environments, overlapping vehicles, and partial vehicle loss, achieving excellent performance [13]. Margret Kasper-Euler et al. conducted a study to address rest period planning by detecting heavy goods vehicles at rest areas during winter. They explored the implementation of YOLOv5 with thermal network cameras, focusing on the front and rear features for recognition. The study demonstrated the potential of thermal network imaging for effective vehicle detection in challenging winter conditions, aiming to improve heavy goods vehicle detection in difficult images with overlaps and cut-offs [14].

The Convolutional Neural Networks show promising results for license number plate detection and character segmentation. There are different types of vehicle license number plate datasets are collected for model training. However, the majority of the researchers focused on license plate detection and character recognition using datasets like Caltech [15] along with different techniques such as edge detection, K-means clustering, and CNN for segmentation and recognition [16].

A huge number of studies have been completed for the task of face detection and recognition just like face detection using OpenCV, exploring algorithms like Adaboost, Haar cascades and Cam Shift. OpenCV framework includes a lot of functionalities for the different tasks of machine learning. Researchers have used OpenCV as a main module for feature extraction, object detection, and recognition. Apart from OpenCV, there is another more accurate face detection algorithm like Haar Cascade outperformed for the task of face detection [4]. The task of face detection comes with multiple challenges, such as variations in human body attributes, unclear images, and the need for face tracking in video surveillance. The Viola-Jones method is utilized, whereas Adaboost is used to train a cascade technique for face detection [5]. The paper also introduces terms like facial feature detection, face authentication, expression recognition, and face localization. The face detection process involves capturing an image, reading it using the misread function, and detecting the face within the image. Real-time face recognition system using OpenCV utilizes algorithms like Haar Cascade, Fisherman, and LBP for face detection under varying conditions and recognizes faces based on features and compares them with existing records, authors provide an approach for face detection and recognition using OpenCV and Python. The paper guides the reader on creating a system using OpenCV and Python, starting with the creation of datasets. The face recognition system performs tasks using Python queries, such as encoding the faces and generating face embeddings. Furthermore, The system utilizes image processing techniques to identify and recognize faces from the given images in the dataset.

III. Data Collection and Preprocessing

Our comprehensive system comprises two main subsystems: vehicle detection and recognition, followed by license plate detection and recognition. The second subsystem involves driver verification, achieved by identifying the driver's face and subsequently comparing it with the employee database. To train the models for each of these subsystems, we gathered a dataset and meticulously labeled it in the subsequent manner:

A. Vehicle Dataset

Our initial task involved collecting a diverse range of images to construct a dataset tailored for vehicle detection. To achieve this, we systematically acquired 6000 photographs of vehicles from three different categories, including cars, buses, and trucks. This process was carried out in person, ensuring that the dataset encompassed a comprehensive representation of these vehicle types. After that, we performed manual labeling by carefully drawing a bounding box around each vehicle and assigning the class names as cars, buses, and trucks according to the image. After completing the manual labeling process, we downloaded the labeled data in YOLO format. This format provided a text file for each image, containing the coordinates of the bounding box and class labels of the vehicles. To consolidate the dataset, we combined all the labeled image folders together. To ensure proper training and evaluation, we split the dataset into three sets training set, validation set, and test set. The training folder contained 60% of the images, while the validation folder contained 20% images and the remaining 20% images are included in the test set.

B. License Number Plate Dataset

The process of number plate dataset creation is more or less the same as we completed the following steps to prepare the dataset for number detection. Firstly, we manually captured a total of 600 number plate images including the car's and bike's number plate. After that, we began the manual labeling process by drawing rectangles around each vehicle's number plate using a drawing tool. We made sure to assign the class name as a car or bike number plate based on the image. This labeling process was repeated for all the captured images until each one was properly labeled. The dataset was provided in YOLO format, which included a text file accompanying each image. These text files contained the coordinates and class labels for the detected license number plate of the vehicle. To consolidate the dataset, we combined all the labeled images and their corresponding text files. Once the dataset was complete, we divided the dataset into training, validation, and test sets for fair training. The training set contained 80% of the images, while the validation set included 20%, and the test set held 20% of the images. Importantly, all three datasets retained the necessary labels for each image.

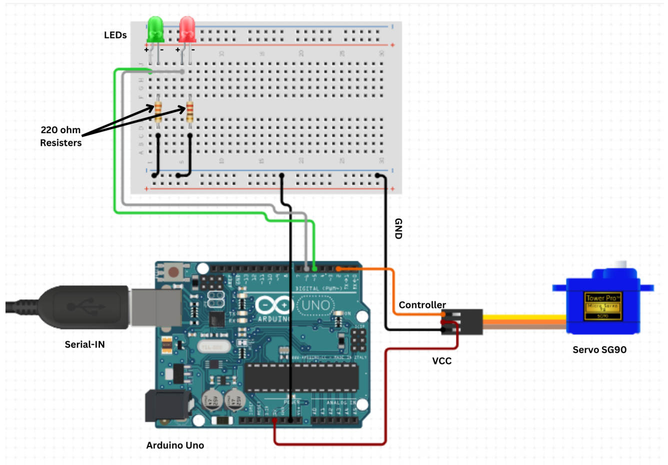

Fig. 1. Circuit diagram of automatic barrier control system

IV. Methodology

In this section, we describe the methodology employed to train a Faster R-CNN model, YOLOv5, and YOLOv8 models for simultaneous vehicle detection and number plate detection tasks. Finally, we have selected YOLOv8 as it produced the best result for vehicle detection and license number plate detection. The detailed flow of our system is shown in

Once we have trained the model for vehicle detection and number plate detection, we need to identify that this vehicle belongs to the organization. For this purpose, we have utilized Google's Tesseract-OCR engine to extract the license number from the detected number plate of the vehicle and match it with the database to authenticate that the vehicle belongs to the organization. To enhance the security of the system we also include the driver's face recognition and verification system. which will ensure that the driver along with the registered vehicle gets access to the organization. For this purpose, we have used Haar cascades which is a machine learning-based approach for face detection. They are widely used for real-time object detection tasks, including face detection. The key idea behind Haar cascades is to use a set of simple features (Haar-like features) to describe regions of an image and train a classifier to distinguish between objects (in this case, faces) and non-object regions. The ”cascade” in Haar cascades refers to a sequence of stages where each stage contains a set of weak classifiers. Each weak classifier is essentially a simple feature that evaluates whether a region of an image matches a specific pattern. To ensure that the detected image of the driver exists in the database of the organization, we employed the DeepFace library to compare the image of the driver with the database. DeepFace is a Python library that provides high-level APIs for deep learning-based face recognition and facial attribute analysis. It uses pre-trained deep learning models to perform tasks like face verification, facial attribute analysis (age, gender, emotion, etc.), and more.

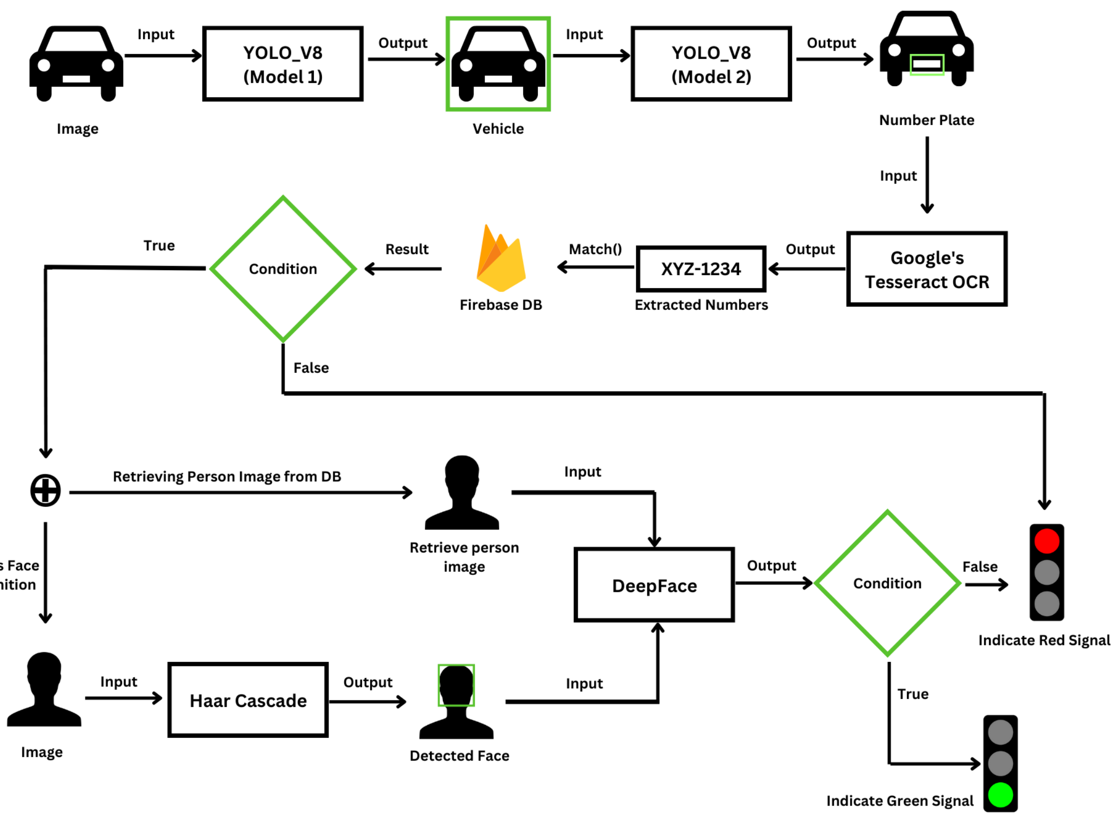

Fig. 2. Smart parking system architecture

We have also introduced a mobile application which helps the driver to find the free slot in the parking area. We have built a prototype of our system using Arduino micro-controller hardware, the circuit diagram of the system is shown in 1.

A. Data Collection and Preparation

A diverse dataset consisting of images containing various vehicle types and their corresponding license plates was collected. The dataset was divided into training, validation, and test sets. Each image was represented as I_i, where i is the index of the image. The detailed process of dataset preparation is explained in section III.

B. Model Architectures and Configuration

1) Faster R-CNN Model: Faster R-CNN (Region-based Convolutional Neural Network) is a deep learning architecture designed for object detection. It's an improvement over earlier object detection models like R-CNN and Fast R-CNN, offering significantly faster processing speeds while maintaining competitive accuracy.

2) YOLOv5 and YOLOv8 Models: The YOLOv5 and YOLOv8 architectures were chosen for their real-time object detection capabilities. They utilized feature extraction layers and detection layers to predict bounding boxes and class labels. The outputs of YOLOv5 and YOLOv8 models for image I_i are denoted as $O_{\text{YOLOv5}}(I_i)$ and $O_{\text{YOLOv8}}(I_i)$ respectively.

3) YOLOv8 Architecture: The YOLOv8 architecture employs several crucial components for performing object detection tasks. The Backbone comprises a sequence of convolutional layers responsible for extracting pertinent features from the input image. The Spatial Pyramid Pooling (SPPF) layer, followed by subsequent convolution layers, handles feature processing at various scales, while the Upsample layers enhance the feature map's resolution. The C2f module combines high-level features with contextual information to enhance detection accuracy. Lastly, the Detection module employs a combination of convolution and linear layers to map high-dimensional features to the output, which includes bounding boxes and object classes. This overall architecture prioritizes speed and efficiency while maintaining a high level of detection accuracy.

In the architecture of YOLOv8 rectangles symbolize different layers, with accompanying labels denoting the layer type (e.g., Conv, Upsample) and any pertinent parameters (such as kernel size or channel count) as shown in 3. The arrows represent the flow of data between these layers, with the direction of the arrow indicating the data's progression from one layer to the next.

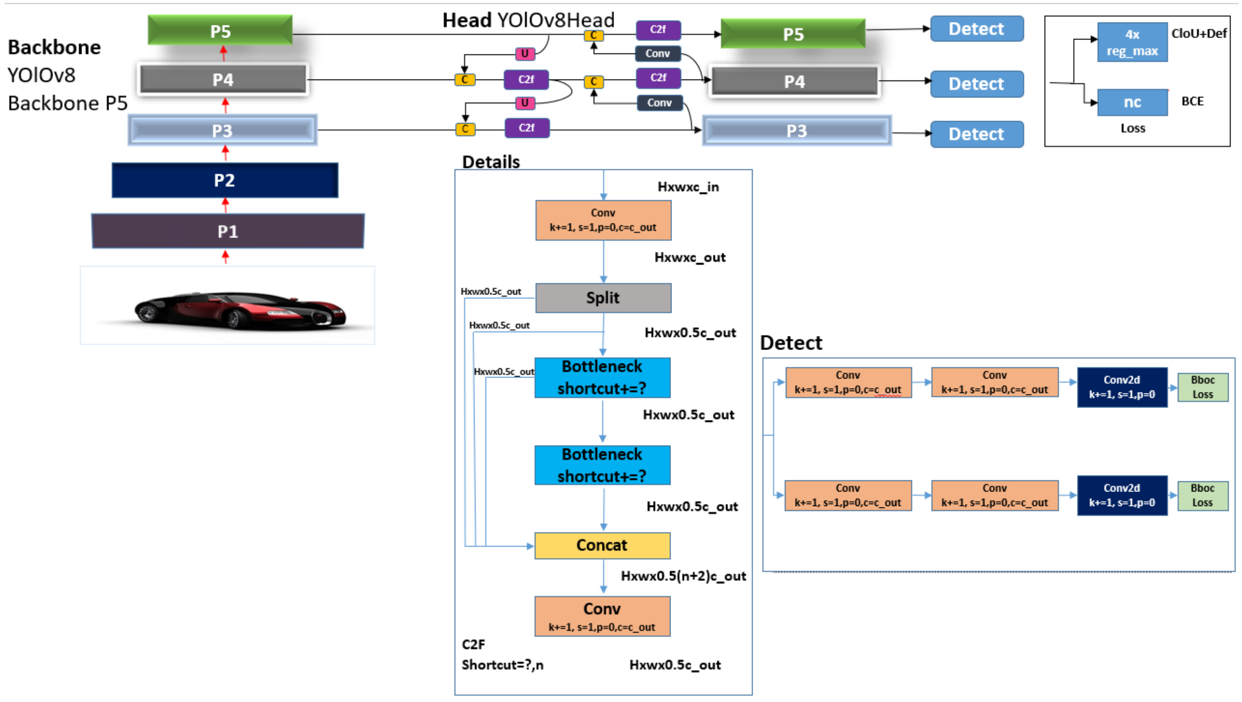

Fig. 3. The architectural configuration of YOLOv8 object detection model

C. Loss Functions

For each model, the training process involved minimizing a combination of localization loss (L_{loc}), objectness loss (L_{obj}), and class prediction loss (L_{cls}). The overall loss function for image I_i is calculated as:

$$L(I_i) = \lambda_{\text{loc}} \cdot L_{\text{loc}}(I_i) + \lambda_{\text{obj}} \cdot L_{\text{obj}}(I_i) + \lambda_{\text{cls}} \cdot L_{\text{cls}}(I_i) \quad (1)$$

Where:

$$L_{\text{loc}}(I_i) = \sum_{j} \sum_{\text{boxes}} \mathbb{1}_{ij}^{\text{obj}} \left((x_j - \hat{x}_j)^2 + (y_j - \hat{y}_j)^2 \right) \quad (2)$$

$$E_1 = \sum_{j} \sum_{\text{boxes}} \mathbb{1}_{ij}^{\text{obj}} \left(C_j - \hat{C}_j \right)^2 \quad (3)$$

$$E_2 = \mathbb{1}_{ij}^{\text{noobj}} \left(C_j - \hat{C}_j \right)^2 \quad (4)$$

$$L_{\text{obj}}(I_i) = E_1 + E_2 \quad (5)$$

$$L_{\text{cls}}(I_i) = \sum_{j} \sum_{\text{boxes}} \mathbb{1}_{ij}^{\text{obj}} \sum_{c \in \text{classes}} (p_j(c) - \hat{p}_j(c))^2 \quad (6)$$

λ_{loc} is the weight coefficient for the localization loss

λ_{obj} is the weight coefficient for the objectness loss

λ_{cls} is the weight coefficient for the class prediction loss

The localization loss (L_{loc}) measures the accuracy of predicted bounding box coordinates compared to ground truth coordinates.

The objectness loss (L_{obj}) quantifies the difference between predicted objectness scores and true objectness values, accounting for object and non-object cells.

The class prediction loss (L_{cls}) computes the difference between predicted class probabilities and true class labels.

The λ coefficients balance the contributions of each loss component to the overall loss function.

D. Training and Optimization

The models were trained using stochastic gradient descent (SGD) to minimize the loss function over the training dataset. The weights and biases of the models were updated iteratively using back-propagation. The optimization process aimed to find the optimal model parameters θ that minimized the loss function:

$$\theta^* = \arg\min_{\theta} \sum_{i} L(I_i) \quad (7)$$

E. Performance Evaluation

The trained models were evaluated using metrics such as Intersection over Union (IOU), mean Average Precision (mAP), and Precision, Recall. These metrics provided insights into the models' accuracy in detecting vehicles and number plates.

1) Intersection Over Union IOU: IOU stands for Intersection over Union. It is a common evaluation metric used in computer vision and object detection tasks to measure the accuracy of object localization. IOU is calculated as the ratio of the area of intersection between the predicted bounding box and the ground truth bounding box to the area of their union.

$$IOU = \frac{Area\_of\_Intersection}{Area\_of\_Union}$$

TABLE I
VEHICLE DETECTION RESULTS OF FASTER R-CNN, YOLOV5 AND YOLOV8.

		Precision	Recall	mAP@50	IOU
Faster R-CNN	Training Data	98.1%	97.8%	98.8%	96.7%
	Validation Data	96.2%	95.7%	98.1%	96.1%
	Test Data	96.4%	96.1%	98.2%	96.1%
YOLOv5	Training Data	98.3%	98%	99.4%	96.9%
	Validation Data	96.6%	94.8%	99%	96.6%
	Test Data	97.8%	98.8%	99.1%	95%
YOLOv8	Training Data	**99.8%**	**99.4%**	**99.5%**	**99.2%**
	Validation Data	**99.7%**	**99.1%**	**99.4%**	**99.1%**
	Test Data	**99.2%**	**99%**	**99.4%**	**99.1%**

TABLE II
NUMBER DETECTION RESULTS OF FASTER R-CNN, YOLOV5 AND YOLOV8.

		Precision	Recall	mAP@50	IOU
Faster R-CNN	Training Data	98%	97.1%	98.5%	95.1%
	Validation Data	97.9%	96.2%	98.3%	94.2%
	Test Data	97.6%	96%	98.1%	94.2%
YOLOv5	Training Data	99%	98.9%	99.2%	98%
	Validation Data	98.8%	98.9%	99.1%	97.9%
	Test Data	98.6%	98.6%	99.1%	97.6%
YOLOv8	Training Data	**99.5%**	**99.3%**	**99.5%**	**98.9%**
	Validation Data	**99.2%**	**99.1%**	**99.3%**	**98.5%**
	Test Data	**99.1%**	**99.1%**	**99.2%**	**98.2%**

In the context of object detection, a higher IOU score indicates that the predicted bounding box closely aligns with the actual object's location. A perfect overlap results in an IOU score of 1, while no overlap or completely disjoint bounding boxes yield an IOU score of 0.

2) Mean Average Precision (mAP): To get an overall performance measure for the entire object detection model, you compute the mean of the AP values across all the object classes. This gives you the mAP, which is often reported as a single number representing the model's performance.

V. RESULTS AND ANALYSIS

Quantitative evaluations were conducted on the validation and test sets to assess the performance of each model. Through a comparative analysis, we aimed to discern the strengths and weaknesses of the Faster R-CNN, YOLOv5, and YOLOv8 models, focusing on detection accuracy and computational efficiency. The results, displayed in the table, underscore that the YOLOv8, a state-of-the-art approach, exhibited outstanding performance across all evaluation metrics in vehicle detection as shown in Table I.

Similarly, in the context of license plate detection, a range of deep learning models including Fast R-CNN, YOLOv5, and YOLOv8 were trained. The Table II depicting the outcomes verifies that YOLOv8, once again, excelled in terms of all evaluation metrics, establishing its superior performance.

VI. CONCLUSION

To address the challenge of vehicle entrance and parking management within organizations, we leveraged cutting-edge deep learning models to automate these processes. The existing manual methods, characterized by inefficiency and time wastage, were surpassed by our approach.

Our system's core components, including vehicle detection, license plate verification, and face detection and recognition models, ensure both the security and registration of individuals and vehicles within the organization. Through rigorous training and evaluation, YOLOv8n emerged as the superior model across various tasks.

Furthermore, the integration of Google's Tesseract-OCR Engine for license plate recognition enhances the system's precision and efficiency. As a result, our system provides efficient vehicle detection, accurate identification, streamlined record-keeping, and optimized parking slot allocation within buildings, significantly enhancing convenience, accuracy, and security.

REFERENCES

[1] U. Ali, R. Hafiz, T. Tauqeer, U. Younis, W. Ali, and A. Ahmad, "Towards machine learning based real-time fault identification and classification in high power induction motors," in *2020 5th International Conference on Robotics and Automation Engineering (ICRAE)*, pp. 46–53, IEEE, 2020.

[2] R. Rahman, Z. Bin Azad, and M. Bakhtiar Hasan, "Densely-populated traffic detection using yolov5 and non-maximum suppression ensembling," in *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021*, pp. 567–578, Springer, 2022.

[3] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2, pp. 629–633, IEEE, 2007.

[4] H. Kaur and A. Mirza, "Face detection using haar cascades classifier," *ICIDSSD 2020*, p. 158, 2021.

[5] D. Meena and R. Sharan, "An approach to face detection and recognition," in *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1–6, IEEE, 2016.

[6] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

[7] C. I. C. Crisostomo, R. V. C. Malalis, R. S. Saysay, and R. G. Baldovino, "A multi-storey garage smart parking system based on image processing," in *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA)*, pp. 52–55, IEEE, 2019.

[8] F. Pervaiz, R. S. Nawaz, M. U. Ramzan, M. Z. Usmani, S. Mare, K. Heimerl, F. Kamiran, R. Anderson, and L. Razaq, "An assessment of sms fraud in pakistan," in *Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies*, pp. 195–205, 2019.

[9] L. Razaq, T. Ahmad, S. Ibtasam, U. Ramzan, and S. Mare, "" we even borrowed money from our neighbor" understanding mobile-based frauds through victims' experiences," *Proceedings of the ACM on human-computer interaction*, vol. 5, no. CSCW1, pp. 1–30, 2021.

[10] A. A. M. Qazzaz, "Car detection and features identification based on yolov5,"

[11] A. Farid, F. Hussain, K. Khan, M. Shahzad, U. Khan, and Z. Mahmood, "A fast and accurate real-time vehicle detection method using deep learning for unconstrained environments," *Applied Sciences*, vol. 13, no. 5, p. 3059, 2023.

[12] Y. Wang, G. Yang, and J. Guo, "Vehicle detection in surveillance videos based on yolov5 lightweight network," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, pp. e143644–e143644, 2022.

[13] K. Zhang, C. Wang, X. Yu, A. Zheng, M. Gao, Z. Pan, G. Chen, and Z. Shen, "Research on mine vehicle tracking and detection technology based on yolov5," *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 347–366, 2022.

[14] M. Kasper-Eulaers, N. Hahn, S. Berger, T. Sebulonsen, Ø. Myrland, and P. E. Kummervold, "Detecting heavy goods vehicles in rest areas in winter conditions using yolov5," *Algorithms*, vol. 14, no. 4, p. 114, 2021.

[15] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[16] B. Su and S. Lu, "Accurate recognition of words in scenes without character segmentation using recurrent neural network," *Pattern Recognition*, vol. 63, pp. 397–405, 2017.