IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 25, 2022, accepted June 8, 2022, date of publication June 14, 2022, date of current version June 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3183101

A Novel Deep Learning Based ANPR Pipeline for Vehicle Access Control

MUHAMMAD GUFRAN KHAN [1], (Senior Member, IEEE), SALMA [1], MAHAM SAEED [1], ADIL ZULFIQAR [1], YAZEED YASIN GHADI [2], (Senior Member, IEEE), AND MUHAMMAD ADNAN [1], (Member, IEEE)

[1]Department of Electrical Engineering, National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan
[2]Department of Software Engineering, Al Ain University, United Arab Emirates

Corresponding author: Muhammad Gufran Khan (m.gufran@nu.edu.pk)

This work was supported in part by the National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, Pakistan.

ABSTRACT Computer Vision and Deep Learning technology are playing a key role in the development of Automatic Number Plate Recognition (ANPR) to achieve the goal of an Intelligent Transportation System (ITS). ANPR systems and pipelines presented in the literature often work on a specific layout of the number plate as every region has a unique plate configuration, font style, size, and layout formation. In this paper, we have developed a smart vehicle access control system considering a wide variety of plate formations and styles for different Asian and European countries and presented novel deep learning based ANPR pipeline that can be used for heterogeneous number plates. The presented improved ANPR pipeline detects vehicle front/rear view and subsequently localizes the number plate area using the YOLOv4 (You Only Look Once) object detection models. Further, an algorithm identifies the unique plate layout, which is either a single or double row layout in different countries, and the last step in the pipeline is to recognize the number plate label using a deep learning architecture (i.e., AlexNet or R-CNNL3). The results show that our trained YOLOv4 model for vehicle front/rear view detection achieves a 98.42% mAP score, and the number plate localization model achieves a 99.71% mAP score on a 0.50 threshold. The overall average plate recognition accuracy of our proposed deep learning-based ANPR pipeline using R-CNNL3 architecture achieved a single character recognition accuracy of 96%, while AlexNet architecture recognized a single character with a 98% accuracy. In contrast, the ANPR pipeline using the OCR method is found to be 90.94%, while latency is computed as 0.99 s/frame on Core i5 CPU and 0.42 s/frame on RTX 2060 GPU. The proposed ANPR system using a deep learning approach is preferred due to better accuracy, but it requires a high-performance GPU for real-time implementation. The presented pipeline is developed and implemented for smart vehicle access control, but it can be deployed for any ANPR application.

INDEX TERMS ANPR, access control, character recognition, deep learning, OCR.

I. INTRODUCTION

The excessive growth in the number of vehicles on the roads is causing significant complexity in managing, identifying, and controlling transportation networks. Consequently, the development of an Intelligent Transportation System (ITS) is necessary to detect, collect, and manage vehicle information for the metropolis of the future. Automatic Number Plate Recognition (ANPR) is an important component of an ITS. The main goal of the ANPR system is to read number plates without any human intervention. The Global Automatic Number Plate Recognition System Market predicts that from 2017 to 2025, the ANPR market will increase with a ratio of 9.63% [1].

A very useful application of ANPR is Access Control Systems as it can assist to identify possible security issues and improve automation. Existing real-time ANPR works in a controlled environment. The CCTV cameras are installed everywhere but they still need human monitoring to keep a record of vehicle entry and exit from the location of the premises. ANPR system can significantly reduce the effort of service personnel, reduce the impact of human factors, and eliminate errors.

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

In a real-time environment, there may be significant differences in vehicle orientation, speed, resolution, and lighting conditions, making number plate recognition more challenging which affects the recognition rate and accuracy. Furthermore, due to the heterogeneous plate formations i.e., double or single-row, and plate style namely Charles's wright, Barcelona, indigo, non-conforming background having stickers, it is difficult to develop a novel ANPR strategy for different regions. There is a gap in the development of a real-time ANPR system due to various plate formations.

The objective of this research is to develop a cost-effective, time-efficient smart ANPR system using computer vision and deep learning (DL) for the Vehicle Access Control application. The proposed ANPR pipeline is an extension and improved version of our previously proposed framework in [2]. We have prepared a multi-national vehicle dataset and applied YOLO object detection models and robust preprocessing techniques on the extracted plate. Further, the OCR Tesseract and Convolutional Neural Network (CNN) architectures are used for number plate recognition. The performance of our system is measured on images and video frames. For real-time implementation, different preprocessing steps are used to denoise the vehicle number plate. This paper forms a contribution to the current body of knowledge as it presents an objective approach to developing a cost-effective, time-efficient smart ANPR system. The YOLOv4 models perform efficiently for Front/ Rear detection and Number Plate Localization with mAP scores of 98.42% and 99.71%, respectively. The average plate recognition accuracy on real-time extracted number plate frames for OCR is 90.94%, for R-CNNL3 is 87.24% and by using AlexNet architecture we achieved 87.56% accuracy.

The remaining paper is organized as follows. In Section II, a detailed review of previously applied techniques is presented. Further, Section III presents the implemented methodology of our proposed ANPR system. In Section IV, the results of the proposed ANPR pipeline are discussed. Moreover, an analysis of previously used methods for the ANPR system is presented in tabular forms. Finally, Section V concludes the paper.

II. LITERATURE REVIEW

The real-time ANPR system has proven to be an important tool for access control and traffic management, with applications ranging from traffic control to vehicle parking and data maintenance for smart surveillance. Several localization techniques, such as edge detection, contouring, and others, were used in the previous ANPR system to detect the vehicle number plate area. Basic image processing techniques can be used by the ANPR system to perform under constraints such as brightness, angle of the plate, and resolution discussed in [3]. Serval localization algorithms are implemented in previous work as they used image preprocessing such as edge detection and contouring for number plate detection.

Furthermore, two different variations of the ANPR system are discussed in [4]; online and offline. The significant deviation among these two terms is that, in an online ANPR system, incoming video frames are interpreted enabling real-time tracking, while offline ANPR systems capture images for supplementary processing using the OpenCV library which is ideal for real-time implementation as it has many predefined methods.

In 2018, Fakhar *et al.* proposed an affordable ANPR system using Raspberry Pi [5], in which the model uses a real-time image captured from a camera. The image is denoised, filtered, and segmented, and the characters on the plate are recognized. The Raspberry Pi handles all calculation complexity, and there is a visible 3-second delay before the result. The resultant label is also saved in the database. The ANPR system for smart check-in and check-out is presented in [6], which reduces waiting time and keeps track of vehicle entry and exit. Images are captured from CCTV cameras so that ANPR systems can instantly detect the plate number through image processing and save the vehicle registration and security information in a database via a web application. The proposed technique would reduce check-in time as well as parking and travel congestion.

Virakwan and Nui Din in [7], suggested an ANPR System in POLIMAS, which verifies that only registered automobiles are permitted to enter the specific place. A webcam is installed using four different orientations: front, back, front top, and rear top. The captured image is converted to grayscale, and the intensity and contrast are adjusted using the histogram equalization approach. The bounding box approach is utilized to identify and crop characters area using a mixture of Sobel edge and Laplacian edge detectors that results in a 95.83 % accuracy rate of plate localization. In OCR, eigenvector and correlation are used to compare each character. The proposed approach resulted in a character recognition accuracy of 87.41 %. The results are collected in a string configuration and compared to the database's reserved entries.

As the recognition remained a challenge due to the various formations of plates in different regions, Chou and Liu [8] presented a real-time truck number plate recognition (TNPR) system which reduces the labor force and time spent in identifying number plates. Their system effectively reduces the risks of crime and improves the transparency, automation, and efficiency of frontline human labor. Using YOLO and CNN based DL architectures, the system achieved a single character identification rate of 97.59 %, an overall recognition rate of 93.73 %, and an inference time of 0.3271 seconds per image.

An ITS was introduced by Chen and Hu in [9] where they focused on video-based vehicle identification and classification techniques, that are based on both static and motion features to achieve improved results. The proposed technique localizes vehicle number plate area and classifies vehicle plate characters with 95% confidence accuracy under environmental conditions such as different illuminations.

A vehicle plate recognition system based on OCR and Wireless Sensors Network (WSN) is described in [10]. The

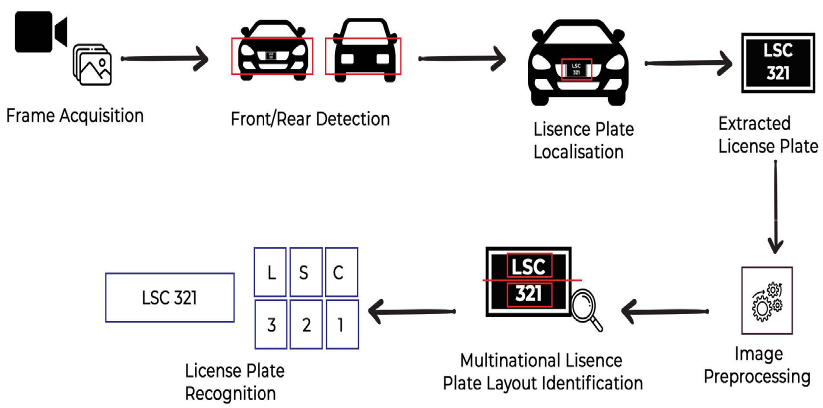


FIGURE 1. **Block diagram of our real-time ANPR framework.**

proposed system uses a Smart Parking Service (SPANS) framework to capture images of parking spaces and recognizes moving or parked vehicle number plates. Furthermore, the system's performance is measured using real-time images. In [11], for plate recognition, R. N. Babu et. al. used state-of-the-art DL techniques. Their image dataset comprised 6500 Indian car number plates that are divided into 90:10 training and testing sets. The images were acquired by three different cameras with different specifications, such as bit rate and focal length. A 37-class CNN model was trained for character recognition. There are 126 filters used in the model and YOLOv3 is used to recognize the vehicle number plates. They achieved a 91% accuracy rate for number plate character recognition.

Moreover, in [12], character recognition is achieved with 98% accuracy using a KNN-based technique. The images were taken with a webcam that has a resolution of 640 $\times$ 480 pixels. Gray-scaling, inversion, thresholding, edge detection, and morphological techniques were all utilized in the process.

Ariff et. al. in [13] used various segmentation techniques for the processing of the noisy license plate images. To remove undesired pixels, 100 Malaysian vehicle plates with a resolution of 1932 $\times$ 2576 pixels were processed using threshold approaches such as Savoula and Niblack. In this case, Savoula segmentation has an average accuracy of 83%. The template matching technique is used to classify characters.

In [14], an accurate vehicle number plate recognition system is proposed based on an OCR engine. Grayscale, global image thresholding using the Otsu method, and noise removal are applied to the input plate image. The processed image is sent to OCR for character recognition. The result is then saved as a string in a text file. The model's overall precision is between 90 and 100%.

In [15], the author introduced a single neural network named ALPRNET, two fully convolutional one-stage object detectors for detecting and classifying license plates and characters simultaneously. A multinational license plate recognition system is introduced in [16] which detects the layout of 17 different countries using Tiny YOLOv3, YOLOv3-SPP (spatial pyramid pooling), and unified character sequence detection. In [17], Shrinivas et. al. proposed a hybrid optimization technique to achieve a high recognition rate with low error rates probability for character recognition.

III. IMPLEMENTED METHODOLOGIES

From the previous studies regarding ANPR, we observed that two different methodologies were frequently used for character recognition in the ANPR systems. The first methodology is OCR Tesseract and the second approach is related to DL. We have also used these two approaches for real-time implementation of our previously proposed ANPR framework [2], in which a novel framework dedicated to Pakistani vehicle number plates was presented. In this research, we have improved our previously proposed pipeline for heterogeneous number plates and considered its application in automatic vehicle access control.

In a typical ANPR system, it is difficult to find an efficient and precise number plate localization and identification method due to diverse practical conditions such as illumination and quality of captured image during the acquisition phase. Traditionally ANPR pipeline comprises the following stages: image acquisition, plate detection, preprocessing of the extracted plate, and number plate recognition. Our proposed real-time ANPR system captures the frames from the installed camera and, using YOLO object detection algorithms, identifies the front or rear view of the vehicle to determine the check-in or check-out state for the access control system. Subsequently, the number plate is localized using the YOLO algorithm. Finally, number plate labels or characters are recognized by efficient state-of-the-art techniques such as CNN-based architectures or OCR-Tesseract. Figure 1 depicts a block diagram of our proposed ANPR system.

A. FRAME ACQUISITION

The frame acquisition is an important stage of any vision system in which an optical image with real-world features transforms into a numerical data array for future manipulation. Pakistani vehicle frames are captured from an Internet Protocol (IP) camera having specifications as given in Table 1. The frames per second (FPS) captured by the camera is 20 FPS with a 1440 × 2156 resolution. We inserted an IP camera at 1.15 m above ground level considering the environmental conditions like illumination, brightness, etc. Moreover, the camera is set at a fixed inclination to capture the vehicle images. However, the training data contained many vehicle images captured from different angles of inclination.

B. DATASET PREPARATION

The training dataset consists of approximately 2200 training samples of cars with Pakistani and international number plates. We created our dataset by capturing 1000 vehicle images of various regions of Pakistan with diverse number plate types in terms of size, fonts, plate color, and orientation. All the images were taken from various angles/inclinations, and they included images with various levels of lighting, dust, and fog. We also used 1000 images from the Stanford vehicle dataset [18], which is an open-source dataset. The vehicle number plates are made up of 36 characters; 26 of which are alphabets, and 10 characters are digits. We used an online web tool provided by Roboflow [19] to transform these images into a Darknet format. Further, annotation of the dataset is done, where a label is assigned to each image. A desktop application named ''LabelImg'' is utilized to generate a.txt file corresponding to every sample, that contains coordinates compatible with the Darknet config.

The model is trained on 1968 number plates (images) for vehicle front/rear detection. Since images contain non-plated vehicles, the number plate localization model is trained using 1660 images. The dataset is split into training and validation sets, with a 90:10 split in each case.

For deep learning-based character recognition (0-9, A-Z, a-z), we used an open-source dataset Chars74K [20], which contains 74,000 images of 64 classes. The collection contains characters extracted from natural images (7,705), hand-drawn characters (3,410), and synthesized characters from computer typefaces (62,992). Since number plates mostly use synthetic characters from computer fonts with digits (0-9) and uppercase letters (A-Z); we have used 36,576 images of this type in the collection that belong to 36 classes. To evaluate the performance of the trained models, we have tested our pipeline on different datasets. Multi-national open-source datasets are also used to evaluate that our proposed pipeline works well on various regions' number plates. The following multi-national open-source datasets are used for evaluation and the results of these datasets are presented in Section IV.

1) IRANI VEHICLE DATASET

The Irani vehicle dataset [21] comprises 313 images with a resolution of 224 × 224 pixels. These vehicle images for the front/rear detection are used and an evaluation of our YOLO models is done.

2) CROATIA VEHICLE DATASET

This dataset comprises 636 vehicle images containing Croatian license plates [22] gathered by the University of Zagreb, Croatia. The characters '0' and 'O' have the same appearance on Croatian license plates. These images are used for testing front/rear detection, license plate localization, and multinational layout identification.

TABLE 1. Frame acquisition parameters.

Parameter	Value
Width	1440
Height	2156
Frame Rate	20 FPS
Resolution	1440 x 2156
Camera Height (above ground)	1.15 m

3) BRAZILIAN VEHICLE DATASET

The dataset [23] contains images of real scenarios divided into five categories in which 2925 images were acquired using a license plate detection model in a public video monitoring system whereas 620 images of cars in a parking slot were captured using the camera. We have used 620 images of the parked cars for testing and evaluation.

4) INDIAN NUMBER PLATE DATASET

The Indian number plate dataset [24] consists of 10,000 images. A subset of 100 images is used for testing and evaluation.

5) PAKISTANI NUMBER PLATE DATASET

For object detection models, we have used the Pakistani number plate dataset. Due to heterogeneous plate formation followed in various regions of Pakistan, we have prepared our dataset considering various factors namely illumination, font style, number plate formation, size, and plate orientation. For evaluating the system performance, we have used 823 samples of Pakistani license plates. Moreover, 200 data samples were used for evaluating the character recognition model of the proposed pipeline.

C. OBJECT DETECTION USING YOLO FRAMEWORK

We used the YOLO framework for object detection as it's fast and yields reliable results in a real-time environment. The literature review demonstrating number plate localization using YOLO models has a higher precision rate provides the idea to prefer the YOLO model. To locate an object, the YOLO model's basic working principle is to apply to an image at multiple scales and locations, assign some scores to different regions, and then choose only the regions with the highest priority as detection regions. Object detection for vehicle front/rear and the number plate is done with YOLOv4 and Tiny YOLOv4.

1) FRONT/REAR DETECTION AND LOCALIZATION

We used a deep learning approach to determine and monitor the vehicle for the access control mechanism. YOLO is opted for because of its real-time performance and high precision. The YOLOv4 and Tiny YOLOv4 are used with the hyperparameters listed in Table 2. In comparison to the

TABLE 2. Hyperparameters of YOLO models for front/rear detection.

Model Hyperparameter	YOLOv4 Value	Tiny YOLOv4 Value
Batch	64	64
Subdivions	64	16
Width	416	416
Height	416	416
Channels	3	3
Momentum	0.949	0.9
Decay	0.0005	0.0005
Angle	0	0
Saturation	1.5	1.5
Exposure	1.5	1.5
Hue	0.1	0.1
learning rate	0.001	0.00261
burn_in	1000	1000
maximum batches	4000	4000
Policy	Steps	Steps
Steps	3200, 3600	3200, 3600

TABLE 3. Hyperparameters of YOLO models for number plate detection and localization.

Model Hyperparameter	YOLOv4 Value	Tiny YOLOv4 Value
Batch	64	64
Subdivions	64	16
Width	416	416
Height	416	416
Channels	3	3
Momentum	0.949	0.9
Decay	0.0005	0.0005
Angle	0	0
Saturation	1.5	1.5
Exposure	1.5	1.5
Hue	0.1	0.1
learning rate	0.001	0.00261
maximum batches	2000	2000
Policy	Steps	Steps
Steps	1600, 1800	1600, 1800

Tiny YOLOv4, the YOLOv4 is preferred for our suggested system as it has a higher mean average precision (mAP). The identification of front and rear objects in vehicles is used to develop the Access controlled system for the ITS. We evaluate the model's performance by maintaining a record of vehicle check-in and check-out with a timestamp in a file using real-time video frames.

2) NUMBER PLATE DETECTION AND LOCALIZATION

The main purpose of this step is to detect and localize the number plate. The YOLO model applies to an image at multiple scales and gives higher accuracy precision on the base of the score around the detection region or boundary box. YOLOv4 and Tiny YOLOv4 are used to localize the region of the vehicle plate. Table 3 represents the Hyperparameters of YOLO models in which the maximum batches are around 2000.

D. IMAGE PREPROCESSING

During this stage, such images are obtained which are passed to algorithms for further training and predictions. Image preprocessing is an important part of any computer vision-related system hence the main objective of image preprocessing is to extract the information (features) of the image for further process. Image Pre-processing [25] is substituted for functionalities performed on nonfigurative images. The goal is to enhance image information so that undesired deformations are reduced or features are amplified for further restoration. The image pre-processing steps for an OCR-based pipeline are shown in Figure 2. We applied different image processing techniques, including grey scaling, binarization, thresholding, and histogram equalization. To begin, we used grey scaling, which removes all color information from each pixel and only leaves the brightness. The image is then denoised using bilateral filtering. Using histogram equalization, the intensities are modified to increase the image's overall contrast. It also improves the margins of each image's object region. Lighting conditions vary widely over a plate, making thresholding necessary. Binarization is a technique for converting images into black-and-white pixels (0 bit for black and 1 bit for white). The pixels with a value less than a threshold are changed to 0 and the image with a value more than the threshold is transformed to 1. Small pixels were removed from the image using morphology. The morphological opening function not only removes the small pixels from the image but also keeps the shape and scale of the larger object.

E. LAYOUT IDENTIFICATION FOR MULTINATIONAL NUMBER PLATE

Every region has a unique number plate configuration. The number plate has a distinct number of rows commonly one or two followed by a unique sequence of characters composed of numeric, alphabetical characters, or the country's national language characters. The extracted number plate from YOLOv4 undergoes a series of image processing. The image is converted to grayscale, later a Gaussian blur filter is applied to remove noise and smoothen the image. After binarization and morphological operation implementation on the smoothed image, a decision is made about the layout precisely the number of rows a license plate holds. Figure 3 represent the flow diagram of handling multinational license plate in which different preprocessing steps are done to reduce noise on the number plate.

F. NUMBER PLATE LABEL RECOGNITION

Plate label recognition is defined as identifying the characters on the number plate and configuring them into the string format. In our proposed system we employed OCR Tesseract and DL approaches/architecture namely R-CNNL3 [8], AlexNet architecture [26], etc. The OCR Tesseract is used to read all types of images and return the recognized text; it is

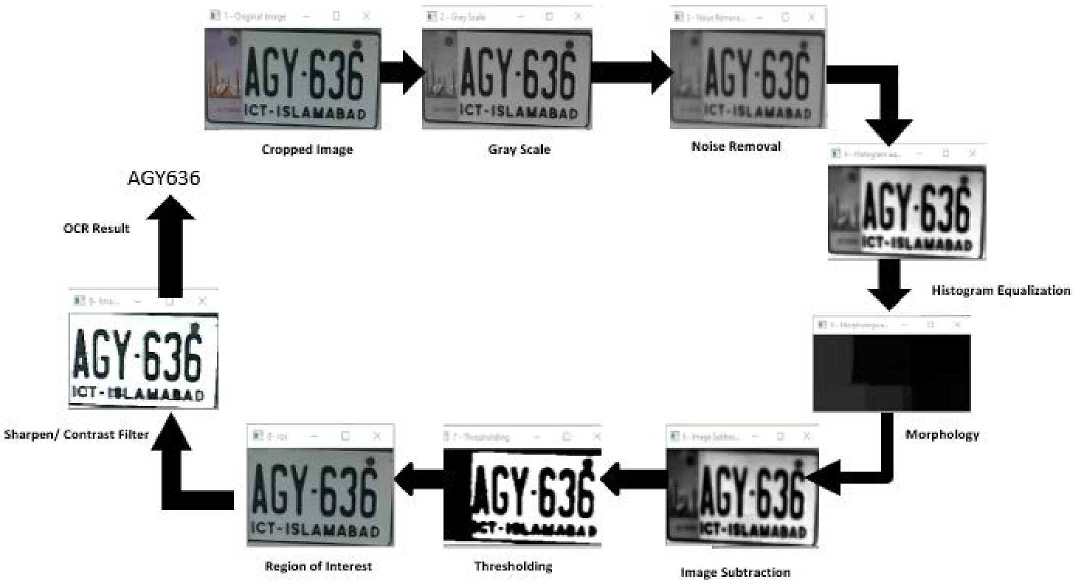


FIGURE 2. OCR based ANPR preprocessing pipeline.

an open-source library that uses the OCR Engine to recognize characters with good accuracy.

1) R-CNNL3 ARCHITECTURE

R-CNNL3 Architecture was introduced by Jui-Sheng and Chia-Hsuan in which 5419 images of dump cars were employed for training models to locate truck license plates, determine the number of characters, and recognize those characters. We also used the R-CNNL3 in our proposed pipeline by fine-tuning the model on our collected dataset and applying the regularization technique to handle the overfitting problem. A dense layer with 128 activation functions and ReLU is added. Finally, there's a dense layer with 36 outputs and a Softmax (probabilistic final decision) activation function in the last layer. While predicting the character dataset we are dealing with mutually exclusive output, Softmax classifier works best when we are dealing with mutually exclusive output. The R-CNNL3 architectural parameters are shown in Table 4, and the results are reviewed in Section IV. On a test dataset of characters with 20 epochs and 128 batch sizes, we achieved 97% training accuracy using this architecture.

2) ALEXNET ARCHITECTURE

To recognize the characters, an AlexNet architecture is used having eight layers among which five are convolutional layers and three are fully connected layers. The first convolutional layer has 96 kernels of size 11×11 with a stride of 4. The second convolutional layer has 256 kernels of size 5×5. The third and fourth convolutional layers have 384 kernels of size 3×3. And the fifth convolutional layer has 256 kernels of size 3×3. The fully connected layers have 4096 neurons. Each layer is followed by the ReLU activation function. And max pooling is applied in the first, second, and fifth layers with size 3×3 and stride 2×2. The input image is

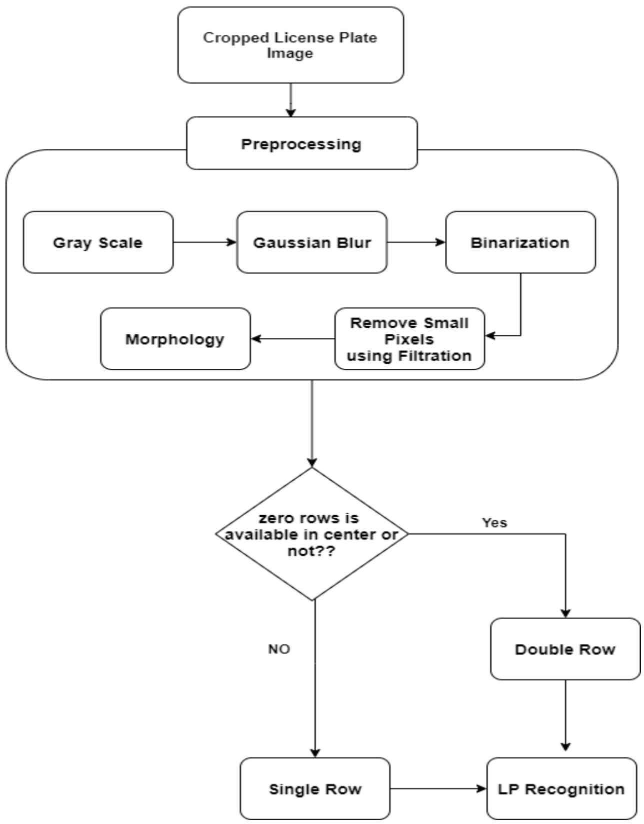


FIGURE 3. Flow diagram for layout identification.

128×128. Table 5 depicts the Alex net architecture. By using this architecture, we have achieved 98% training accuracy with 20 epochs and 128 batch sizes.

3) OCR-TESSERACT

To implement OCR, we have used the Python-tesseract library which is an open-source library often used to

TABLE 4. R-CNNL3 model parameters.

Layer	Type	Network
1	Input	45 × 85 × 3
2	Convolutional	48@5 × 5
3	Max-pooling	2 × 2
4	Convolutional	64@5 × 5
5	Max-pooling	2 × 2
6	Convolutional	128@5 × 5
7	Max-pooling	2 × 2
8	Dropout	0.5
9	Flatten	-
10	Dense Layer	Fully connected

recognize characters embedded in the optical images. It is the wrapper for Google's Tesseract-OCR Engine [27]. It is efficient in time and has built-in modes namely Page Segmentation Mode (PSM) and OCR Engine Mode (OEM). These modes help recognize the characters and we have set the value of PSM and OEM according to our characters on license plates.

G. NUMBER PLATE LABEL MANAGEMENT

At this stage, the number plate labels are stored in an entity for example a file, database, or cloud storage to keep a record. We have stored recognized characters in a file which is further used to process the plate number and its check-in and check-out time

IV. RESULTS AND ANALYSIS

In this Section, we have discussed the results and comparative analysis of the above approaches. We have used mAP, loss value, precision, recall, and F1 score to measure the performance of object detection methods implemented in our pipeline.

A. FRONT/WWWWW/REAR DETECTION AND LOCALIZATION

We used YOLO models for vehicle front/rear detection to identify whether a vehicle is checking-in or checking-out. YOLO models are considered to be a fast and effective way to detect objects in real-time. We already discussed the parameters that are used to train these models in Section III. The average loss and mAP score of YOLOv4 of front/rear detection is shown in Figure 4. The average loss rate over all iterations and batches is represented by the dotted line. During training, the graph indicates an average loss of less than 2.0, which shows that the model is efficient. The average precision score is shown as a solid line. On the 0.5 thresholds, we achieved 95.3% mAP with tiny YOLOv4, 96.7 % mAP with YOLOv4, and 91.0 % mAP with YOLOv3. In terms of performance metrics, Table 6 shows different evaluation measures of front/rear detection.

B. NUMBER PLATE DETECTION AND LOCALIZATION

The localization of the number plate is also done using the YOLO model. YOLO darknet model applies a single neural network to the full image; the network divides the image into regions and predicts bounding boxes and probabilities for each region. All the regions are given weights according to the probability and the regions with the highest weighted scores in the image are considered detections. Our YOLO model is capable to localize every type of vehicle plate in a real-time system including cars, busses, trucks, and bikes.

Figure 5 depicts the average loss and mAP score of YOLOv4 for vehicle number plate localization. The dotted line shows the average loss rate throughout all iterations and batches. While training, the graph shows the average loss under 2.0 which makes the model efficient. The solid line shows the mean average precision score. We achieved 94.90% mAP using Tiny YOLOv4, 98.6% mAP using YOLOv4, and 94.1 % mAP using YOLOv3 on the 0.5 thresholds. Table 8 represents the license plate training accuracy rate in terms of performance metrics. Table 9 depicts the model accuracy rate based on different datasets.

TABLE 5. Alex net model parameters.

Layer	Type	Network
1	Input	128 × 128 × 3
2	Convolutional	96@11 × 11@4
3	Max-pooling	3 × 3@2
4	Convolutional	256@5 × 5@1
5	Max-pooling	3 × 3@1
6	Convolutional	384@3 × 3@1
7	Convolutional	384@3 x 3@1
8	Convolutional	256@3 x 3@1
9	Max-pooling	3 × 3@2
10	Flatten	-
11	Fully Connected	4096
12	Dropout	0.5
13	Fully Connected	4096
14	Dropout	0.5
15	Fully Connected	36

TABLE 6. Performance measure of front/rear detection.

Model	mAP@0.5 (%)	Loss Value	Precision	Recall	F1-score
YOLOv3	91.00	0.0820	0.96	0.86	0.91
YOLOv4	96.72	0.7340	0.95	0.93	0.94
Tiny YOLOv4	95.32	0.1347	0.92	0.93	0.93

TABLE 7. Localization accuracy comparison of front/rear detection.

Dataset	No. of sample	Correctly detected YOLOv4	Correctly detected Tiny YOLOv4	Accuracy (YOLO v4)	Accuracy (Tiny YOLOv4)
Pakistani	823	810	805	98.42%	97.81%
Irani	313	305	300	97.44%	95.84%
Croatia	636	620	598	97.48%	94.02%
Europe	100	98	96	98.00%	96.00%
Brazilian	620	612	599	98.70%	96.61%

C. REAL-TIME NUMBER PLATE DETECTION

The object detection model is implemented in a real-time scenario. Figure 6 represents some resulting frames from the video stream by using the YOLOv4 model. The first

TABLE 8. Performance measure of number plate localization.

Model	mAP@ 0.5 (%)	Loss Value	Precision	Recall	F1-score
YOLOv3	94.09	0.1017	0.96	0.93	0.94
YOLOv4	98.60	0.3474	0.94	0.98	0.96
Tiny YOLOv4	94.90	0.1002	0.88	0.97	0.92

TABLE 9. Accuracy comparison of number plate localization.

Dataset	No. of sample	Correctly detected YOLOv4	Correctly detected Tiny YOLOv4	Accuracy (YOLO v4)	Accuracy (Tiny YOLOv4)
Pakistani	810	808	805	99.75%	98.13%
Irani	305	302	300	99.01%	96.33%
Croatia	620	618	598	99.68%	97.49%
Europe	98	97	96	97.00%	96.00%
Brazilian	612	610	599	99.67%	98.83%

TABLE 10. Accuracy of muti-layout identification on a different dataset.

Dataset	Total Number of Samples	No. of Correct Samples	Accuracy
Pakistani	200	193	96.5%
Europe	95	91	95.78%
Irani	50	46	92%
Indian	100	90	90%
Croatia	250	242	96.8%
North American	60	57	95%
South American	150	142	94.6%
Algerian	100	92	92%
Total	910	862	94.73%

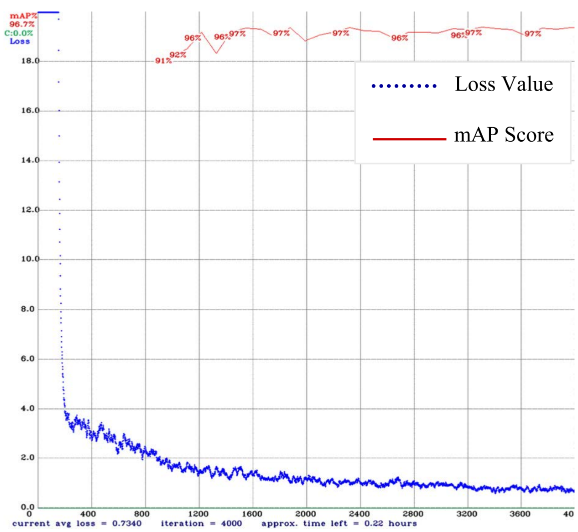


FIGURE 4. Training graph for front/rear detection using YOLOv4.

TABLE 11. OCR overall character recognition performance.

Dataset	Total Sample size	Correctly recognize sample of OCR	Accuracy of OCR
Pakistan	193	180	93.26%
Europe	91	88	96.70%
India	90	78	86.67%
Croatia	242	213	88.01%
North American	57	52	91.22%
South American	142	127	89.43%
Algeria	92	84	91.30%

TABLE 12. R-CNNL3 and AlexNet overall character recognition performance.

Dataset	Total Sample size	Correct recognize for R-CNNL3	Correct recognize for AlexNet	Accuracy of R-CNNL3	Accuracy of AlexNet
Pakistan	193	169	174	87.56%	90.15%
Europe	91	82	81	90.10%	89.01%
India	90	79	80	87.78%	88.89%
Croatia	242	199	194	82.23%	80.16%
North American	57	50	51	87.71%	89.47%
South American	142	127	130	89.43%	91.54%
Algeria	92	79	77	85.86%	83.69%

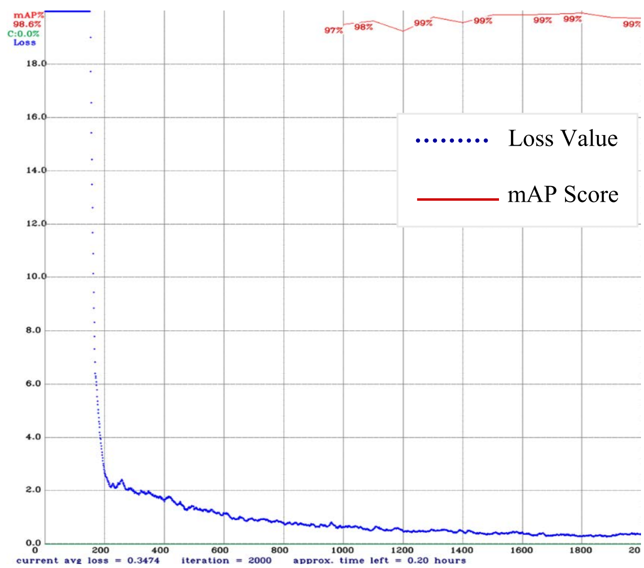


FIGURE 5. Training graph for number plate detection and localization using YOLOv4.

boundary box represents the vehicle's front/rear view and the second boundary box shows the number plate area. After the extraction of the number plate area, the layout identification process is done on the plate.

D. LAYOUT IDENTIFICATION FOR MULTINATIONAL NUMBER PLATE

For multinational layout-based number plate detection, we have experimented with different datasets using our pipeline in which we found a 94% average accuracy rate. Table 10 represents the accuracy rate of multi-layout identification.

E. RECOGNITION OF LICENSE PLATE LABEL

1) RECOGNITION USING OCR

For character recognition, we have used two methods. The first is OCR Tesseract and the second is DL based models. In Table 11, the recognition results of OCR-based

TABLE 13. Character recognition confidence in implemented methods.

Extracted Frames	OCR Recognition Result	OCR Confidence Score (%)	AlexNet Based CNN Recognition Result	AlexNet Based CNN Confidence Score (%)
	FDS4014	100	FD54014	86
	FDS4014	100	F0S404	77
	FD5401	86	FD404	83
	LR35D3	89	LR3503	100
	LR3503	100	LR3SD3	67
	LR3503	100	LR303	91
	MN6	67	M696	80
	MNI696	100	MNI696	100

implemented methods are represented. The average results of character recognition on all datasets are 90.94%.

2) RECOGNITION USING R-CNNL3

For character recognition, we have used DL based R-CNNL3 model architecture. R-CNNL3 architectures, 97% training accuracy is achieved. In Table 12, the recognition results of R-CNNL3 architecture-based implemented methods are represented in which the segmented image is passed to the architecture then finally the resulted plate label is generated and further stored in a file. On the other hand, single character recognition accuracy is 96%.

3) RECOGNITION USING ALEXNET

For character recognition, we have used deep learning-based AlexNet model architecture. In Table 12, the recognition results of AlexNet architecture-based implemented methods are represented by employing it on various datasets after the segmentation process. 98% training accuracy is achieved by using AlexNet. On the other hand, the single character recognition accuracy is 98%.

It is observed from Table 12 that the two different CNN architectures were used to explore the state-of-the-art DL approaches namely AlexNet and RCNN-L3 for character recognition; both models performed well but the training and single character recognition accuracy of AlexNet architecture is better than R-CNNL3. So, for comparative analysis considering the performance between these models; we further explore AlexNet for real-time implementation as described in Table 13 in which the real-time frames were extracted from the video stream and after some preprocessing steps, the preprocessed frame passed to recognition models and finally, after the plate character recognition process, some confidence score is generated by using fuzzywuzzy library;

TABLE 14. Comparison with previously used implemented methods for ANPR.

Ref.	Localization	Segmentation	Recognition	Image Condition	Overall Recognition Rate on datasets (Average)	Country	Constraint and Limitation
[28]	Already cropped	Not used	HOG feature and Extreme Learning Machine	Low resolution portion of the image, 15–18 px Height	90%	South Thailand	Day time only, no license localization process is applied
[29]	Object detection, CNN—(YOLO Detector)	Character Segmentation CNN, Bounding box	Data augmentation, Distant CNN for letter and Digits	1920 × 1080 Pixels	SSIG: 93.53% UFPR-ALPR: 78.33%	Brazil	Adjustments have to be made for other than Brazilian formats. Dependent on license plate layout.
[30]	LBP, Character and edge information	Vertical Histogram	Tesseract OCR with preprocessing techniques	250 pixels wide, Various conditions and colors	90%	Myanmar	High processing time.
[31]	Cascade Classifier with LBP features	-	Tesseract OCR	640×480 pixels with 50×11 pixels aspect ratio	92.12%	Indian	Dependent on Standardized Number plates, Overall accuracy is for the front side number plate only at a fixed angle, High processing time.
Our proposed	**YOLOv4, Tiny YOLOv4, and YOLOv3**	**Segmentation for CNN architecture-based model**	**OCR Tesseract, Two CNN based architectures (R-CNNL3 and AlexNet)**	**For training resolution of images are 416x416 while for testing we used an IP camera having a resolution of 1440x2156**	**OCR: 90.94% R-CNNL3: 87.24% AlexNet: 87.56%**	**Pakistan Europe India Croatia North American South American Algeria**	**Having good results in a real-time environment by taking less than 1 second inference time for each recognition model**

over these frameworks namely OCR Tesseract and AlexNet Architecture. The real-time ANPR system for access control is used to keep the record of vehicle check-in and check-out. The number plate recognition is done using various methods in which the OCR Tesseract performs well on the video frames and real-time images.

Table 14 shows the performance of our ANPR system. As for the diversity of the environment, we carried out two existing systems, one that used CNN and the other is YOLO. It is observed that Previously proposed ANPR approaches work well on specific format or layout of the plate. Due to the various plate formation among various regions, we proposed a universal Multinational real-time ANPR system that generates reasonably accurate and fast results as compared to the previous one.

TABLE 15. Inference time per frame of localization models.

Model	Inference Time using CPU (seconds)	Inference Time using GPU (seconds)
Front/Rear YOLOv4	0.48	0.08
Front/Rear Tiny YOLOv4	0.28	0.05
Number Plate YOLOv4	0.45	0.07
Number Plate Tiny YOLOv4	0.25	0.06

FIGURE 6. Real-time implementation of YOLOv4 model-based detection and localization pipeline.

TABLE 16. Inference time per frame of recognition models.

Method	Inference Time using CPU (seconds)	Inference time using GPU (seconds)
OCR	0.45	0.018
R-CNNL3	0.67	0.024
AlexNet	0.84	0.035

Tables 15, 16, and 17 describe the computation time taken by recognition models on CPU with Intel Core i5 CPU and

TABLE 17. Overall inference time per frame of implemented pipeline.

Method	Approach	Inference time using CPU (seconds)	Inference time using GPU (seconds)
OCR	With Front/Rear Detection	1.50	0.42
	Without Front/Rear Detection	0.99	0.17
AlexNet	With Front/Rear Detection	1.90	0.27
	Without Front/Rear Detection	1.48	0.19
R-CNNL3	With Front/Rear Detection	1.70	0.26
	Without Front/Rear Detection	1.21	0.18

GPU RTX 2060 It concludes that running on GPU reduces the response time without compromising the accuracy of the model.

V. CONCLUSION

This paper presents a novel deep learning-based ANPR pipeline that is implemented and tested for automatic vehicle access control applications. By using object detection and DL models, we counter the heterogeneity and assortment

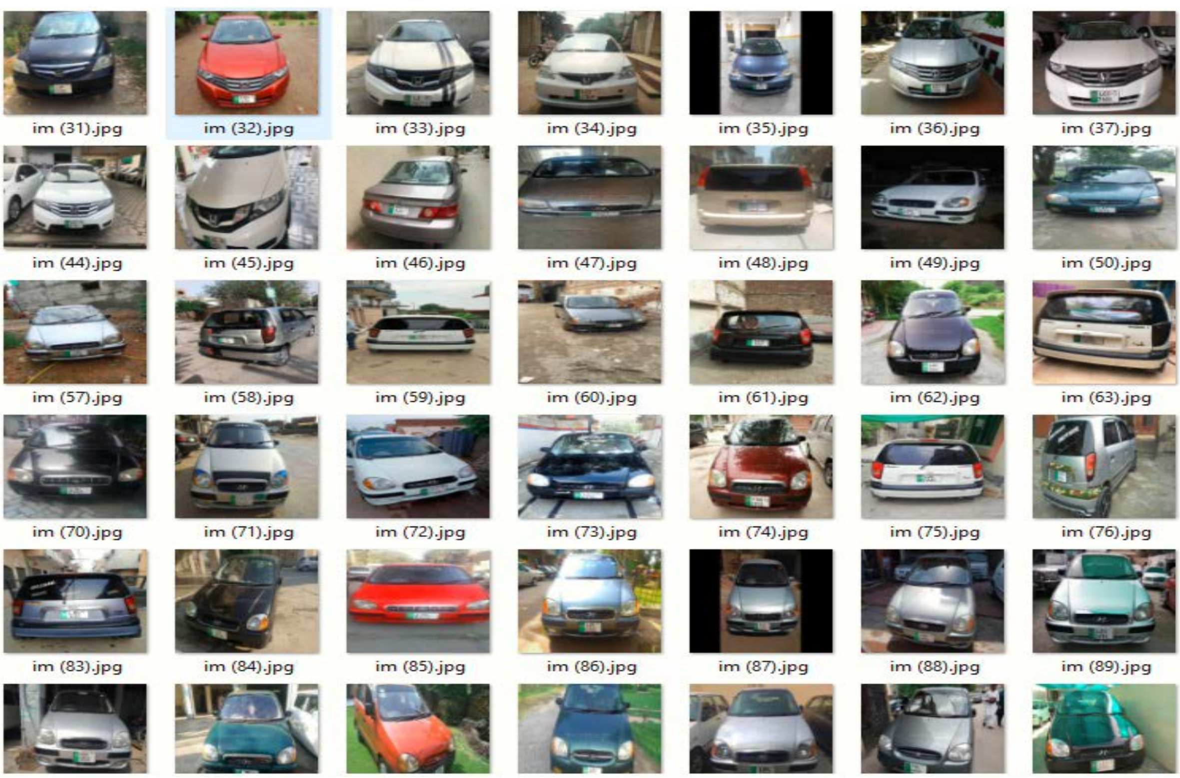


FIGURE 7. Example test images of vehicle number plates.

problem of number plates across various Asian and European region number plates. The proposed real-time ANPR pipeline is tested using an IP camera video frames collected by considering the variations in the environment illumination and frame orientation. The obtained mAP score for plate extraction using YOLOv4 is 99.71% on a 0.50 threshold. In addition, for vehicle front/rear-view, we used another YOLOv4 which gives a 98.42% mAP score. The preprocessing techniques are applied to the localized plate, and after the identification of the plate layout, the last step is to pass the frames to the DL character recognition model namely AlexNet or R-CNNL3. Our ANPR pipeline with the YOLOv4 model and OCR gives an overall average plate recognition accuracy score of 90.94%. Furthermore, DL-based architectures namely R-CNNL3 gives 96% on single characters recognition and 87.24% average accuracy on overall character recognition on different datasets. Meanwhile, the AlexNet architecture gives 98%-single character recognition accuracy and 87.56% as overall character recognition accuracy on different datasets. The inference time of the OCR-based pipeline is 0.99 s/frame with CPU Core i5 and 0.42 s/frame with GPU RTX 2060. AlexNet architecture-based pipeline takes 1.48 s/frame computation time using CPU and 0.19 s/frame using GPU. Similarly, the R-CNNL3-architecture based pipeline takes 1.21 s/frame using CPU and 0.18 s/frame using GPU. By considering its computation time per frame, we conclude that our proposed novel ANPR pipeline performs well with reasonable accuracy in a real-time scenario, while the inference time can be decreased further using Tiny YOLOv4. It should be mentioned that the proposed ANPR pipeline is not limited to vehicle access control, it is applicable to different traffic scenarios and ANPR applications.

In the future, we will work on making the proposed solution generic for all ANPR applications by removing the limitation of fixed camera orientation for real-time ANPR systems. Another possible future work is to use the YOLOv5 model with the proposed pipeline.

APPENDIX

See Figure 7.

REFERENCES

[1] *Global Automatic Number Plate Recognition (ANPR) System Market Size*. Accessed: Sep. 24, 2021. [Online]. Available: https://www.bccresearch.com/partners/verified-market-research/global-automatic-number-plate-recognition-anpr-system-market.html

[2] Salma, M. Saeed, R. U. Rahim, M. G. Khan, A. Zulfiqar, and M. T. Bhatti, "Development of ANPR framework for Pakistani vehicle number plates using object detection and OCR," *Complexity*, vol. 2021, pp. 1–14, Oct. 2021, doi: 10.1155/2021/5597337.

[3] S. Saha, S. Basu, and M. Nasipuri, "Automatic localization and recognition of license plate characters for Indian vehicles," *Int. J. Comput. Sci. Emerg. Technol.*, vol. 2, pp. 520–533, Aug. 2011.

[4] N. Abirami, "Accurate vehicle number plate recognition and real-time identification using Raspberry Pi," *Int. Res. J. Eng. Technol.*, vol. 5, no. 4, p. 7, 2018.

[5] S. Fakhar, M. Saad, A. Fauzan, R. Affendi, and M. Aidil, "Development of portable automatic number plate recognition (ANPR) system on Raspberry Pi," *Int. J. Electr. Comput. Eng. IJECE*, vol. 9, no. 3, pp. 1805–1813, Jun. 2019, doi: 10.11591/ijece.v9i3.pp1765-1772.

[6] S. Tenzin, P. Dorji, B. Subba, and T. Tobgay, "Smart check-in check-out system for vehicles using automatic number plate recognition," in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2020, pp. 1–6, doi: 10.1109/ICCCNT49239.2020.9225555.

[7] N. D. Keraf, P. Ehkan, V. H. Kelian, N. M. Noor, H. Darus, and E. Loke, "Automatic vehicle identification system using number plate recognition in POLIMAS," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 767, Mar. 2020, Art. no. 012056, doi: 10.1088/1757-899X/767/1/012056.

[8] J.-S. Chou and C.-H. Liu, "Automated sensing system for real-time recognition of trucks in river dredging areas using computer vision and convolutional deep learning," *Sensors*, vol. 21, no. 2, p. 555, Jan. 2021, doi: 10.3390/s21020555.

[9] Y. Chen and W. Hu, "A video-based method with strong-robustness for vehicle detection and classification based on static appearance features and motion features," *IEEE Access*, vol. 9, pp. 13083–13098, 2021, doi: 10.1109/ACCESS.2021.3051659.

[10] N. D. V. Dalarmelina, M. A. Teixeira, and R. I. Meneguette, "A real-time automatic plate recognition system based on optical character recognition and wireless sensor networks for ITS," *Sensors*, vol. 20, no. 1, p. 55, Dec. 2019, doi: 10.3390/s20010055.

[11] R. N. Babu, V. Sowmya, and K. P. Soman, "Indian car number plate recognition using deep learning," in *Proc. 2nd Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT)*, Jul. 2019, pp. 1269–1272, doi: 10.1109/ICICICT46008.2019.8993238.

[12] S. Sugeng and E. Y. Syamsuddin, "Designing automatic number plate recognition (ANPR) systems based on K-NN machine learning on the Raspberry Pi embedded system," *Jurnal Teknik Elektro dan Vokasional*, vol. 5, no. 1, Sep. 2019, Art. no. 1, doi: 10.24036/jtev.v5i1.1.106135.

[13] F. N. M. Ariff, A. S. A. Nasir, H. Jaafar, and A. Zulkifli, "Sauvola and Niblack techniques analysis for segmentation of vehicle license plate," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 864, no. 1, May 2020, Art. no. 012136, doi: 10.1088/1757-899X/864/1/012136.

[14] V. Nayak, "Automatic number plate recognition," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 3783–3787, Jun. 2020, doi: 10.30534/ijatcse/2020/195932020.

[15] Q. Huang, Z. Cai, and T. Lan, "A single neural network for mixed style license plate detection and recognition," *IEEE Access*, vol. 9, pp. 21777–21785, 2021, doi: 10.1109/ACCESS.2021.3055243.

[16] C. Henry, S. Y. Ahn, and S. Lee, "Multinational license plate recognition using generalized character sequence detection," *IEEE Access*, vol. 8, pp. 35185–35199, 2020, doi: 10.1109/ACCESS.2020.2974973.

[17] S. R. Zanwar, U. B. Shinde, A. S. Narote, and S. P. Narote, "Hybrid optimization and effectual classification for high recognitions in OCR systems," *J. Inst. Eng. India B*, vol. 102, no. 5, pp. 969–977, Oct. 2021, doi: 10.1007/s40031-021-00604-7.

[18] *Cars Dataset*. Accessed: Sep. 25, 2021. [Online]. Available: http://ai.stanford.edu/~jkrause/cars/car_dataset.html

[19] *Roboflow: Give Your Software the Power to See Objects in Images and Video*. Accessed: Sep. 24, 2021. [Online]. Available: https://roboflow.com/

[20] *The Chars74K Image Dataset—Character Recognition in Natural Images*. Accessed: Sep. 24, 2021. [Online]. Available: http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

[21] *Iran-Vehicle-Plate-Dataset*. Accessed: Sep. 24, 2021. [Online]. Available: https://kaggle.com/samyarr/iranvehicleplatedataset

[22] *Projekt License Plates*. Accessed: Sep. 24, 2021. [Online]. Available: http://www.zemris.fer.hr/projects/LicensePlates/english/results.shtml

[23] G. Silvano, P. T. Endo, V. C. T. Ribeiro, V. Greati, I. Silva, T. Lynn, and A. Bezerra, "Artificial Mercosur license plates," *Mendeley Data*, vol. 2, 2020, doi: 10.17632/nx9xbs4rgx.2.

[24] *Indian License Plates | Kaggle*. Accessed: Sep. 24, 2021. [Online]. Available: https://www.kaggle.com/thamizhsterio/indian-license-plates

[25] S. Inamoto, K. Sato, S. Nagao, and D. Ogata, "Image processing device, method and program," U.S. Patent, 2010 0 177 976 A1, Jul. 15, 2010. Accessed: Sep. 24, 2021. [Online]. Available: https://patents.google.com/patent/US20100177976/en

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9. Accessed: Sep. 24, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[27] Tessdoc. *Tesseract User Manual*. Accessed: Sep. 24, 2021. [Online]. Available: https://tesseract-ocr.github.io/tessdoc/

[28] S. Kraisin and N. Kaothanthong, "Accuracy improvement of a province name recognition on Thai license plate," in *Proc. Int. Joint Symp. Artif. Intell. Natural Lang. Process. (iSAI-NLP)*, Nov. 2018, pp. 1–6, doi: 10.1109/iSAI-NLP.2018.8692983.

[29] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, ''A robust real-time automatic license plate recognition based on the YOLO detector,'' 2018, *arXiv:1802.09567*.

[30] N. H. Lin, Y. L. Aung, and W. K. Khaing, ''Automatic vehicle license plate recognition system for smart transportation,'' in *Proc. IEEE Int. Conf. Internet Things Intell. Syst. (IOTAIS)*, Nov. 2018, pp. 97–103, doi: 10.1109/IOTAIS.2018.8600829.

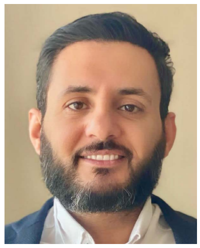
[31] G. G. Desai and P. P. Bartakke, ''Real-time implementation of Indian license plate recognition system,'' in *Proc. IEEE Punecon*, Nov. 2018, pp. 1–5, doi: 10.1109/PUNECON.2018.8745419.

MUHAMMAD GUFRAN KHAN (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2003, and the M.Sc. degree in electrical engineering (specialization in signal processing) and the Ph.D. degree in electrical engineering (specialization in wireless communication) from the Blekinge Institute of Technology, Sweden, in 2005 and 2011, respectively. He is currently an Associate Professor at the Department of Electrical Engineering, FAST NUCES Chiniot-Faisalabad Campus. Before joining FAST, he worked as an Analysis Engineer with Volvo Car Corporation, Sweden. He has conducted research in the areas of signal processing, computer vision, and machine learning. He worked on different funded research projects in the area of embedded systems and robotics. He is actively involved in the applications of AI and the IoT technology to solve real-world problems.

SALMA received the bachelor's degree in computer science from the University of Central Punjab, Faisalabad, Pakistan, in 2020. She is currently pursuing the M.S. degree in computer science with the FAST National University of Computer and Emerging Sciences (NUCES), Chiniot-Faisalabad Campus. She is also working as a Research Assistant at the FAST NUCES, Chiniot-Faisalabad Campus. Her research interests include intelligent machine, artificial intelligence, machine learning, data science, and computer vision.

MAHAM SAEED received the degree in computer and information science from the Pakistan Institute of Engineering and Applied Sciences, Pakistan, in 2020. Since October 2020, she has been a Research Assistant at the FAST National University of Computer and Emerging Sciences (NUCES), Chiniot-Faisalabad Campus. Her research interests include intelligent machine, artificial intelligence, machine learning, edge computing, and robotics. She is the Winner of the National Robotics Pakistan conducted by Starfall Foundation and participated in the International Championship in 2015.

ADIL ZULFIQAR received the M.S. degree in electronic engineering from the GIK Institute of Engineering Sciences and Technology, in 2014, and the Ph.D. degree from the City University of Hong Kong, in 2018. He is currently the Head of the Electrical Engineering Department and an Assistant Professor at the FAST NUCES. His research interests include time-delay systems, robust control, networked control systems, feedback, graph theory, multi-agent systems, uncertain systems, approximation theory, continuous-time systems, control system synthesis, discrete-time systems, distributed control, eigenvalues, eigenfunctions, interpolation, linear systems, network theory (graphs), poles and zeros, stability, three-term control, delay systems, open-loop systems, and polynomials.



YAZEED YASIN GHADI (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Queensland University. His dissertation on developing novel hybrid plasmonicphotonic on-chip biochemical sensors received the Sigma Xi Best Ph.D. Thesis Award. He was a Postdoctoral Researcher at Queensland University before joining Al Ain University. He is currently an Assistant Professor of software engineering at Al Ain University. He has published more than 25 peer-reviewed journals and conference papers and he holds three pending patents. His current research interests include developing novel electro-acousto-optic neural interfaces for large-scale high-resolution electrophysiology and distributed optogenetic stimulation. He was a recipient of a number of awards.

MUHAMMAD ADNAN (Member, IEEE) received the B.S. degree in electrical engineering from the National University of Computer and Emerging Sciences, Peshawar, Pakistan, in 2013, the M.S. degree in electrical engineering from the COMSATS Institute of Information and Technology, Islamabad, Pakistan, in 2015, and the Ph.D. degree in electrical engineering from the National University of Computer and Emerging Sciences. From January 2017 to December 2019, he worked as a Research Fellow with the Department of Electrical Power Engineering. He is currently working as an Assistant Professor with the Department of Electrical Engineering, FAST NUCES, CFD Campus. His research interests include energy management systems, load flow balancing, load forecasting, power systems dynamic analysis, protection, stability, and intelligent control in renewable energy resources using a fuzzy controller and unified power flow controller.