Measurement: Sensors 27 (2023) 100761

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Implementation of number plate detection system for vehicle registration using IOT and recognition using CNN

M.A. Jawale [a], P. William [a,*], A.B. Pawar [b], Nikhil Marriwala [c]

[a] Department of Information Technology, Sanjivani College of Engineering, Savitribai Phule Pune University, Pune, India
[b] Department of Computer Engineering, Sanjivani College of Engineering, Savitribai Phule Pune University, Pune, India
[c] Electronics and Communication Engineering Department, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, Haryana, India

ARTICLE INFO

Keywords:
Automatic license plate recognition
Image analysis
IOT
Image classification
Segmentation
Character recognition
Convolution neural network

ABSTRACT

In the intelligent transportation system the automatic license plate recognition and detection plays a very important role. This application could be used for traffic control security e-payment systems in the toll pay and parking. Many algorithms have been developed to force license plate detection and recognition and all have many advantages and flaws under different situations. With the advent and rise of deep learning concepts in various fields of artificial intelligence, computer vision has developed in leaps and bounds in terms of innovations and methods. Automatic License Plate Recognition has emerged as an effective method to automate the watch keeping process for traffic systems, parking fee structures, and police surveillance. License plate recognition (LPR) is a quite used and mature technology but much work is needed to be done in order to make it perfect. In recent years, the scientific community has made major advances in methodology and performance. This paper tries to aim at summarizing and analyzing various methodologies and progress in LPR in the deep learning era using IOT sensors. Hence, in this paper, an Automatic License Plate Detection and Recognition (ALPDR) system has been proposed having four steps namely License Plate Extraction, Image Pre-processing, Character Segmentation and Character Recognition. For the first three steps (extraction, pre-processing, and segmentation), unique methods have been proposed. As the character recognition is an important step of license plate recognition and detection, four different methods for character recognition have been experimented on, which include Convolution Neural Network (CNN), MobileNet, Inception V3, ResNet 50.

1. Introduction

In recent times, urban areas are facing increasing problems due to the incremental nature of vehicular traffic. The main reasons are development of urban cities and an increase in cars and vehicles. The congestion of traffic, rules violation, cars stealing are the challenges of today's transportation and management system. There are many solutions to these problems including intelligent traffic surveillance [14,15], autonomous vehicles [13], automatic tracking of vehicles and speed detection [16]. Automatic License Plate Detection and Recognition (ALPDR) is a vital component in intelligent transportation systems. ALPDR system used image processing and Artificial Intelligence to detect the vehicles with their license plates. Generally, ALPDR system is composed of three major steps/phases; the first step is pre-processing, after the image capturing the image is processed for color space to grayscale, resizing and removal of noises. In the second step the license plate localization, the location of the license plate in the vehicle image is found for further processing. The third step is the character recognition, it is an important step to read the license plate and get the character to identify the vehicle.

Despite the fact that much research has been conducted on automatic license plate recognition and detection, many of the techniques consist of limitations and stage satisfactions on some scenarios. Various general restrictions are illumination, presence of noise and blur and distortion, tilt image capturing and so on. In addition to days there are a number of varieties of font shapes, with different sizes and colors of each symbol and it varies from different vehicles types, states and countries.

The proposed architecture for Automatic License Plate Detection & Recognition (ALPDR) involves the processes of acquisition of a vehicle image, license plate (LP) extraction with IOT based sensors, pre-processing of the vehicle image, segmentation of characters from the license plate, and finally character recognition. The extraction

* Corresponding author.
E-mail address: william160891@gmail.com (P. William).

https://doi.org/10.1016/j.measen.2023.100761
Received 3 January 2023; Received in revised form 5 March 2023; Accepted 1 May 2023
Available online 2 May 2023

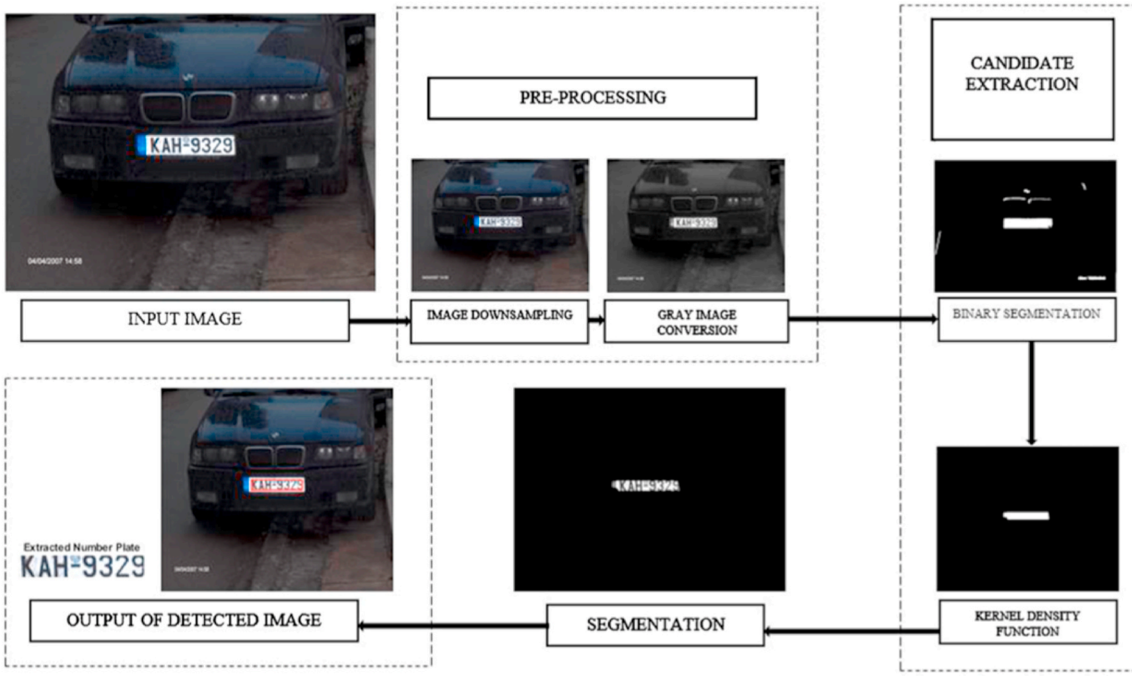


Fig. 1. License plate recognition system.

procedure for the license plate is very complicated in nature and error-prone, because it has a straight effect on the efficiency & accuracy of the successive steps. Hence, it is very important to solve the problems in the presence of illumination circumstances and various other monotonous back-grounds. In this paper, the novel ALPDR system is proposed and implemented, for better understanding the comparison with various methodologies for license plate character recognition has been performed. We have gathered two datasets one for License plate extraction and another for character recognition.

The main objectives of this paper are:

1. To propose a Novel Automatic License Plate Detection and Recognition (ALPDR) system with four proposed steps.
2. To enhance the accuracy of the character recognition using proposed CNN method.
3. To implement the proposed ALPDR and to compare the result with the existing technique under different matrices.

The paper is hence organized as follows: in Section 2 the literature survey has been discussed; in Section 3 the proposed method is described with the equations and diagrams; implementation results and analysis of the proposed algorithm is done in Section 4 followed by Conclusion and References.

2. Literature survey

There are many frameworks that have been proposed for license plate detection and recognition, based on different image processing methods and artificial intelligence techniques. X. Ascar et al. [1] proposed the license plate detection technique which is based on kernel density function and binary techniques used for processing the license plate. The location of the license plate is what we found by multiplying the binary value and the original value of the image. Later on, the filtered binary value of the image has been used.

Ravi Kiran et al. [2] presented an image processing technique for Indian license plate detection and recognition with various conditions such as noisy environment, low light, non-standard license plate and cross angled situation. For the pre-processing they used several techniques such as gaussian smoothing, morphological transform and gaussian thresholding. For the segmentation they used contours and K-nearest neighbor algorithm for character recognition. Another method for Indian license plates has been proposed by Hanit Karwal et al. [3]. They address the problem of recognition of position of character and the scaling problem.

Fei Xie et al. [4] proposed a license plate detection and character recognition method based on a hybrid model composed of an effective feature extraction technique coupled with a Back-Propagation Neural Network (BPNN) classifier. The author claims that their method can be able to overcome the problem of low illumination and complicated backgrounds. The pre-processing step has been taken to strengthen the image of the car and after locating the license plate on the car a feature extraction model is designed and finally, using backpropagation neural network model the character recognition is achieved.

In the framework proposed by Simmani et al. [5], the two-dimensional wavelet transform technique is used for vertical edge-extraction from the input-image, basing on the observation that the large number of vertical edges help to find out the location of the license-plate. The character recognition is done in the license plate using CNN classifier. The author claims that this proposed method is able to overcome different issues of license plate detection and character recognition.

Sathya et al. [6] designed the framework for a vascular license plate recognition system called capsule network. The author claims that this framework is robust and works fine in any conditions such as various orientations, rotations, shifting and flipping of the license-plate image. The main objective of this framework was identified as the improvement of the processing time using the capsule network framework by segmentation. It involves training and recognition of the system to locate the license-plate.

Swati Jagtap [7] has proposed a framework which checks the license plate number with the authorized user database and only allows authorized vehicles. In this feature extraction has been used by using the technique called histogram of oriented gradient. The number of identified vehicles shows the performance of this framework.

Tejas K et al. [8] proposed an automatic license plate recognition framework in which the license plate region extraction is achieved using

```
        if intWidth > lower_width and intWidth < upper_width and intHeight > lower_height and i
ntHeight < upper_height :
            x_cntr_list.append(intX) #stores the x coordinate of the character's contour, to use
d later for indexing the contours

            char_copy = np.zeros((44,24))
            # extracting each character using the enclosing rectangle's coordinates.
            char = img[intY:intY+intHeight, intX:intX+intWidth]
            char = cv2.resize(char, (20, 40))

            cv2.rectangle(ii, (intX,intY), (intWidth+intX, intY+intHeight), (50,21,200), 2)
            plt.imshow(ii, cmap='gray')

            # Make result formatted for classification: invert colors
            char = cv2.subtract(255, char)

            # Resize the image to 24x44 with black border
            char_copy[2:42, 2:22] = char
            char_copy[0:2, :] = 0
            char_copy[:, 0:2] = 0
            char_copy[42:44, :] = 0
            char_copy[:, 22:24] = 0

            img_res.append(char_copy) # List that stores the character's binary image (unsorted)
```

Fig. 2. Segmentation process.

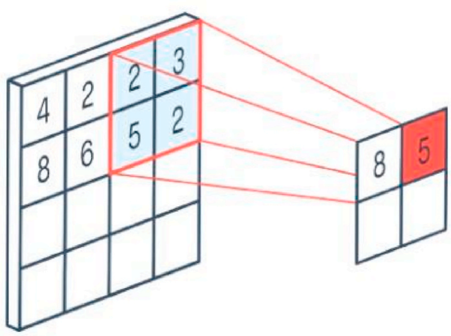


Fig. 3. Max pooling layer.

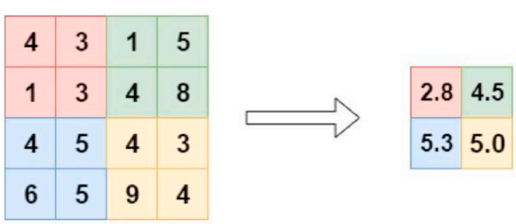


Fig. 4. Average pooling layer.

Sobel edge detection algorithm while morphological operations are used in segmentation. They have used the IoT for the frequent database update of the vehicle.

Priyanka et al. [9] presented the technique for license plate localization segmentation and character recognition despite issues like uneven illumination and tilt. It can process the image from a steel camera or video bye transforming it to grayscale image. At last, every single character and located license plate can be detected with high accuracy.

Young Jung et al. [10] proposed the method for license plate recognition in which the color (RGB) image will be converted into grayscale image and the binarization converts the grayscale into the black and white image then the localization of the license plate along with the character segmentation is achieved and finally character recognition is done. The author claims that this method can remove some preprocessing tasks such as noise filtering histogram equalization contrast enhancement etc.

Anci Manon et al. [11] created an Automating Number Plate Recognition technique which is able to recognize the number-plate of the moving vehicle. The application of this system is to recognize the

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 16)	23248
conv2d_1 (Conv2D)	(None, 28, 28, 32)	131104
conv2d_2 (Conv2D)	(None, 28, 28, 64)	131136
conv2d_3 (Conv2D)	(None, 28, 28, 64)	65600
max_pooling2d (MaxPooling2D)	(None, 7, 7, 64)	0
dropout (Dropout)	(None, 7, 7, 64)	0
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 128)	401536
dense_1 (Dense)	(None, 36)	4644

Fig. 5. Model summary.

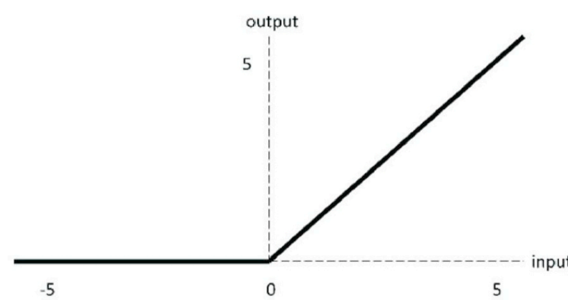


Fig. 6. ReLU function.

vehicle violating the traffic rules such as speed and traffic signal, it will also be helpful for the stolen vehicle detection system.

Shobayo et al. [12] presented a vehicle number plate recognition system which is an IoT based system having the higher sensor for taking the image of a vehicle and the image processing methods to locate, segment and detect the vehicle number from the number plate. It uses OpenCV for implementation along with different IoT related hardware such as higher sensor Raspberry Pi and other components.

In the proposed automatic license plate recognition & detection for the steps has been proposed which includes license plate-extraction image preprocessing, character segmentation and character recognition. For each step you need a novel method and technique that has been applied at last for character recognition. The proposed CNN method has been used and compared the implementation result with different character recognition techniques.

3. Proposed algorithm

The process of number plate recognition consists of four main stages: the first is the License Plate (Region-of-Interest) Extraction; the second, Pre-processing of the extracted regions (ROIs); the third, Character Segmentation; and finally, Recognition of the characters on the license plate. The detailed description about all the steps are shown in Fig. 1. The methodology used in each stage is also discussed in further subsections.

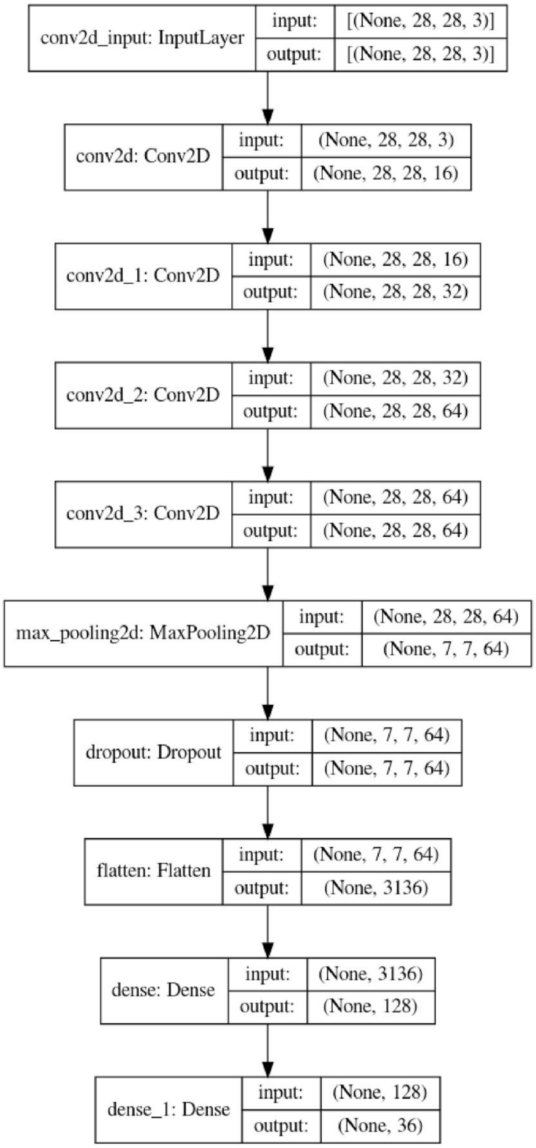


Fig. 7. Flowchart of CNN model.

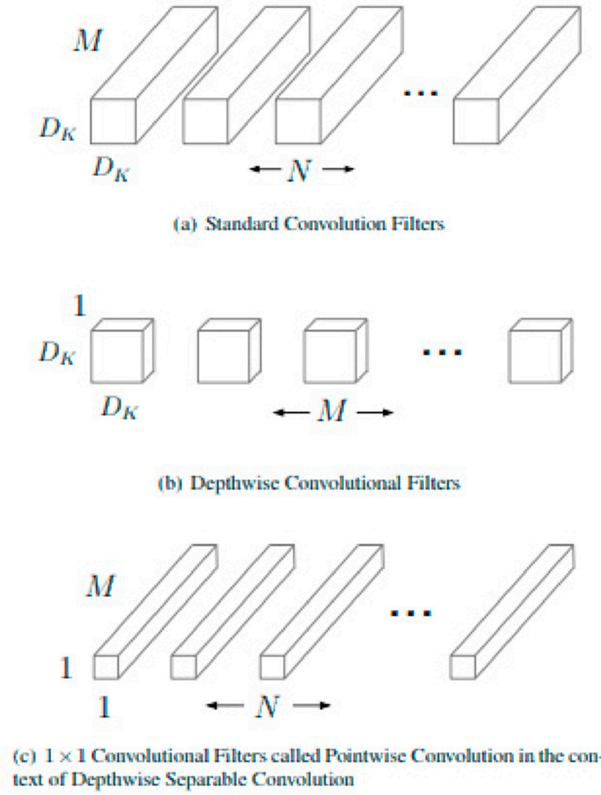


Fig. 8. Depth-wise separable convolution in MobileNets.

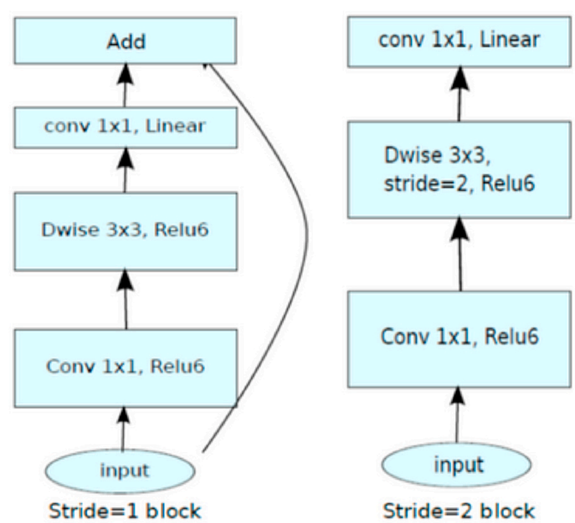


Fig. 9. MobileNet architecture.

3.1. *License plate extraction*

For License Plate Extraction, we have used pre-trained Haar Cascade Classifier.

Haar Cascade Classifier: Haar Cascade is an object recognition algorithm that can be used to detect objects and faces in images and videos. It is focused on the idea of features. It serves as the foundation for all object detection using the Haar-like features algorithm. Haar cascade is a technique that uses a huge number of positive and negative images to train the cascade classifier. We have used Cascade-Trainer-GUI to train Cascade-Classifier from our Indian license plate dataset [13–15]. We also have taken 3000 negative images dataset to improve the classifier prediction. We ran the training process for 10 stages. Finally, it generated an XML weighted file. XML file serves as the basis for the algorithm which contains the features of the images [16,17]. In the license plate extraction process our cascade classifier performs very well to extract the license plate.

- Positive images – Positive images are those images which are supposed to be recognized by our classifier. Our dataset contained 1100 images of positive images
- Negative Images – Negative images are all the random images which are of no use i.e., all those images which do not contain the object we want our classifier to detect. Our dataset contains 3000 negative images.

3.2. *Image Pre-Processing*

After extraction of the number plate, further image processing operations are to be performed on the extracted license plate in order to eliminate noisy data [18].

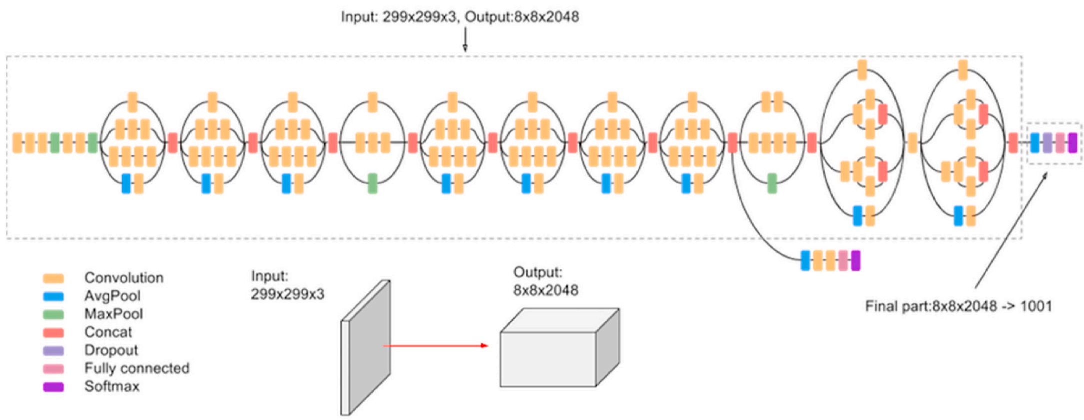


Fig. 10. High Level Diagram of Inception V3 model.

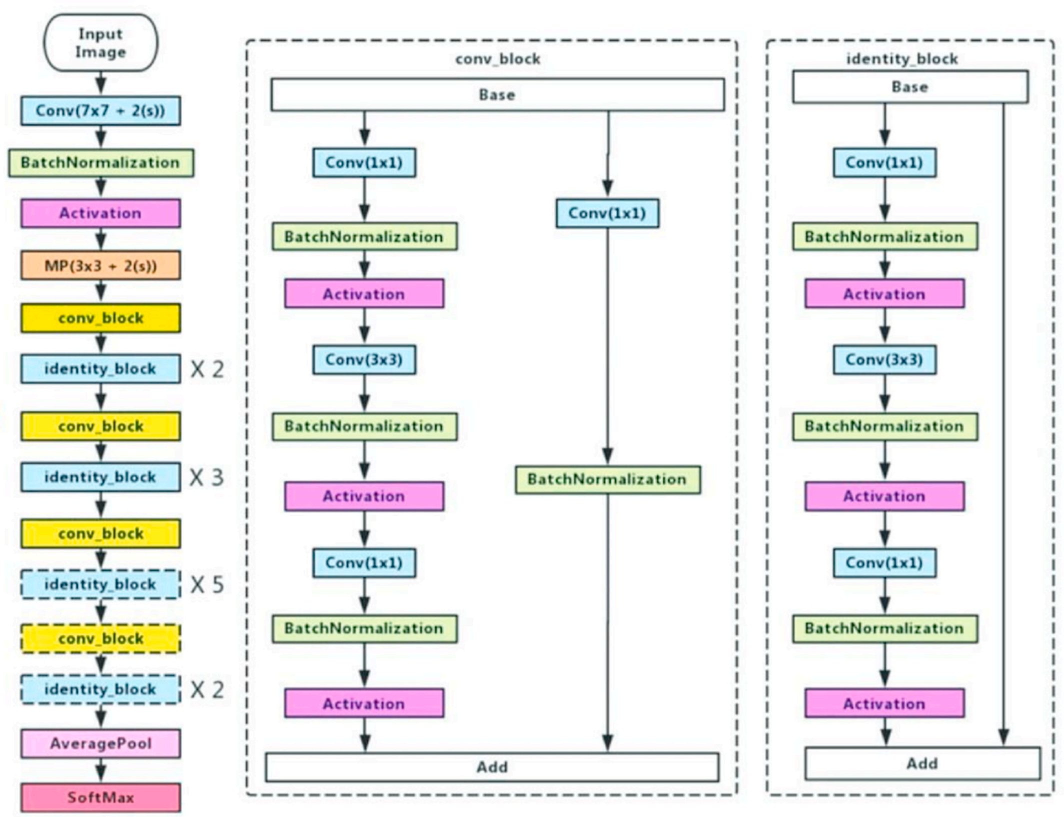


Fig. 11. High Level Diagram of ResNet50 model.

3.2.1. *Image transformation*

Image transformation is a process of converting an image from one domain to another domain, it means mapping the one set to another set after performing some operations. Here, the image is transformed into a grey image. There are various ways to do it, the method used here is an average method [19]. It takes into consideration all the three-color values r, g, b and takes average of it as shown below in Equation (1).

$$Grayscale = \frac{R+G+B}{3} \tag{1}$$

3.2.2. *Image thresholding*

Thresholding is the process of converting a color or grayscale image into a binary image. We have used two different types of thresholding methods. Threshold function transforms the grey-scaled image to binary image, which means each pixel value will be either 0 or 1. 0 value corresponds to black and 1 corresponds to white. This is achieved by applying the threshold that has a value between 0 and 255. Here, the threshold value is set to 200 that means pixels having value greater than 200 will be assigned value 1 in the new binary image and value less than 200 will be assigned 0 [20].

3.2.2.1. Simple thresholding. In simple thresholding the same threshold value is used for each pixel. If the pixel intensity value is less than the defined threshold, it is set to 0, otherwise a maximum value is used as shown in Equation (2).

$$dst(x,y) = \begin{cases} maxval \ \ if \ src(x,y) \geq tresh \\ 0 \qquad otherwise \end{cases} \tag{2}$$

Fig. 12. Extraction & Pre-Processing result of different Indian Vehicle Images.

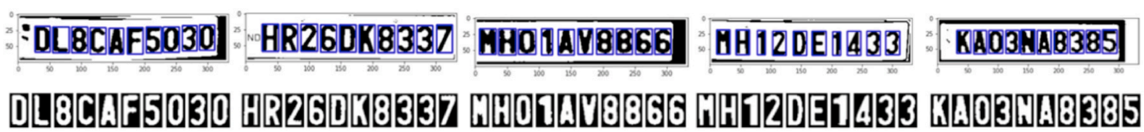


Fig. 13. Segmentation result of different Indian Vehicle License Plate.

3.2.2.2. Otsu's thresholding. It iterates from the entire threshold estimate and calculates the expansion for the level of the pixels at either edge of the threshold estimate, that is the pixels which belong to either in the background or in the foreground. The objective is to acquire the threshold estimation in the scenario where summation of background and foreground and expansion is lowest. The algorithm iteratively calculates a threshold value 't' that serves to minimize the intra-class (within the class) variance as illustrated in Equation (3).

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{3}$$

Here,

$$\begin{gathered} q_1(t) = \sum\nolimits_{i=1}^{t} P(i) \ \& \ q_1 \ (t) = \sum\nolimits_{i=t+1}^{I} P(i) \\ \mu_1(t) = \sum\nolimits_{i=1}^{t} \frac{iP(i)}{q_1(t)} \ \& \ \mu_2 \ (t) = \sum\nolimits_{i=t+1}^{I} \frac{iP(i)}{q_2(t)} \\ \sigma_1^2(t) = \sum\nolimits_{i=1}^{t} [i - \mu_1 \ (t)]^2 \frac{P(i)}{q_1(t)} \ \& \ \sigma_2^2(t) = \sum\nolimits_{i=t+1}^{I} [i - \mu_2 \ (t)]^2 \frac{P(i)}{q_2(t)} \end{gathered} \tag{4}$$

3.2.3. Image filling

For the purpose of image filling, morphological operations are used. Morphological Operations are a group of image processing operations that manipulate digital images based on their shapes. In these operations each pixel value corresponds to other pixels in its neighborhood. On selecting the neighborhood pixel's shape and size, a morphological function could be applied that would be effective for a particular input image shape. The two basic morphological functions when applied on binary images produce contrasting results [21–23].

3.2.3.1. Erosion. Erosion is the process of removing the edge pixels. It shrinks/decreases/contracts the boundaries of the image, meaning that it decreases the white region in the image.

3.2.3.2. Dilation. It is the reverse process of erosion. This process increases/expands the boundaries or the white portion of the image. It is used to join the broken parts of the object.

3.3. Character segmentation

After preprocessing operations are performed on the extracted license plate, further segmentation is done on the number plate image to segment the characters for the next stage as shown in Fig. 2.

As shown above segmentation is achieved by comparing the height width ratio of the bounding box of characters with that of number plate dimensions. If the dimensions of the bounding box of characters lies within the dimensions of the license plate, then it is accepted, else it is rejected. All the accepted dimension values are stored in an array. After iterating through every pixel of the license plate we get all the characters which are stored in the array. Then the characters are inverted and given as the output [24–26].

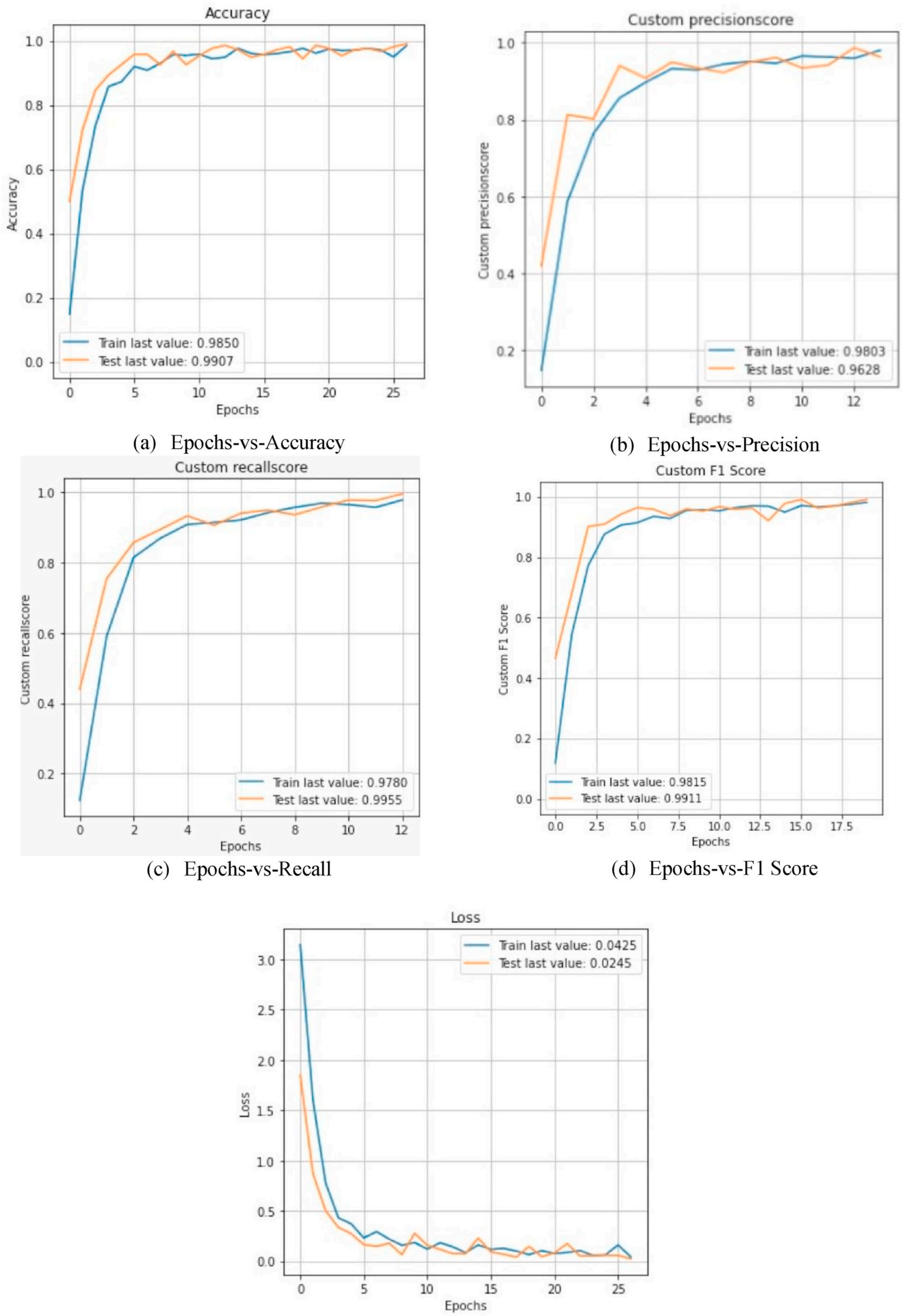
3.4. Character recognition

In this section we emphasize on character recognition after the character segmentation process. Character recognition is the process where we extract the characters/figures from the detected license-plate region and it is a very crucial stage for the Automatic License Plate Recognition and Detection system. Hence for the character recognition step we proposed four different models such as CNN, MobileNet, Inception V3 and ResNet50. In the subsequent subsections we describe all three models for character recognition in detail.

3.4.1. CNN model

Convolutional Neural Networks are deep learning architectures that take input images and assign biases & weights to the objects in the image in order to distinguish them from the other images in the input dataset. In CNN models, less preprocessing is required in comparison with other classification algorithms. Three types of layers are required for creating a generic deep learning model [27,28].

1. **Input Layer:** Inputs are provided to the model using this layer. In input layer, number of neurons equals the number of features (image features here) per sample (image, in the present case) in the input dataset.
2. **Hidden Layers:** Input layer's output servers as the input to the hidden layer. The number of hidden layers along with the number of



(a) Epochs-vs-Accuracy

(b) Epochs-vs-Precision

(c) Epochs-vs-Recall

(d) Epochs-vs-F1 Score

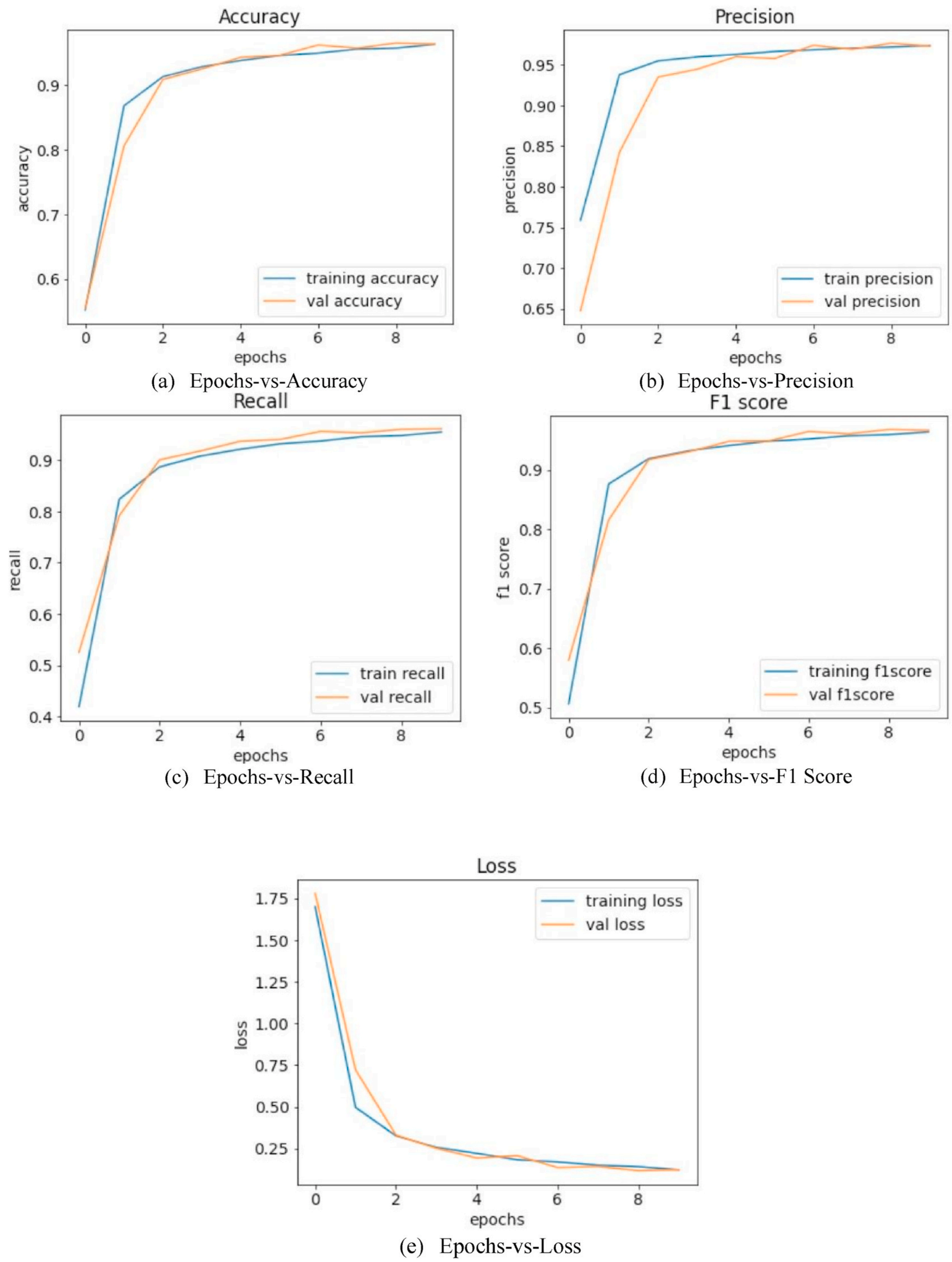
(e) Epochs-vs-Loss

Fig. 14. Character Recognition Results using CNN.

hidden neurons per layer depends on the type, size, and depth of the network. Output of the current layer is computed by matrix multiplication of the output of the previous layer and learning weights and after it by addition of biases then activation function is applied which makes the network non-linear.

3. **Output Layer:** Logistic function like SoftMax is put into the output of the hidden layer. It transforms the output of every class into a probability score.

The following layers constitute the architecture of a general Convolutional Neural Network:



(a) Epochs-vs-Accuracy (b) Epochs-vs-Precision (c) Epochs-vs-Recall (d) Epochs-vs-F1 Score (e) Epochs-vs-Loss

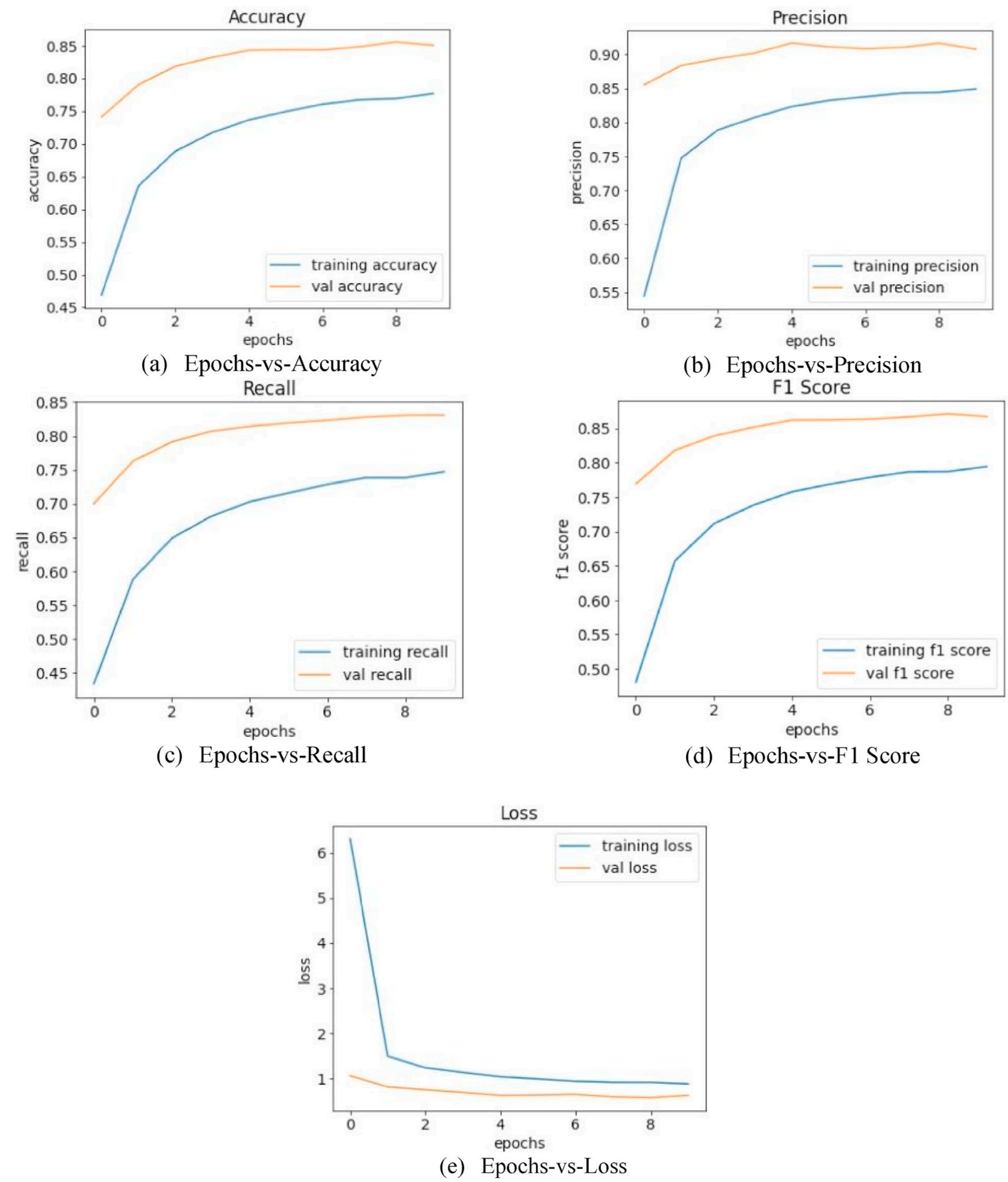
Fig. 15. Character Recognition Results using MobileNet.

1. **Convolutional Layer:** This layer is the foundation layer of any CNN model. It contains the main part of the network. Dot product is performed with the two matrices, one is the learning matrix and the other matrix is the confined part of the receptive field [31–33].
2. **Pooling Layer:** The pooling layer is used to minimize the spatial dimension of the convoluted features. It decreases the power required for processing the data with the help of dimensionality reduction. Max Pooling and Average Pooling are the two most common types of pooling operations used in Convolutional models.

● **Max Pooling:** The maximum value of the image window convolved by the kernel/filter is considered, as shown in Fig. 3.

● **Average Pooling:** The average of all the values of the image window convolved by the kernel/filter is calculated, as shown in Fig. 4.

4. **Dropout Layer:** It is added to a model with some dropout rate to prevent it from over-fitting. It randomly selects the neuron to be ignored during the training. When training is set to true, then it is applied.
5. **Flatten Layer:** Flatten layer converts data to 1 dimensional array, which serves as input to the following layer. An exclusive extended feature vector is generated to level the result of the convolutional layer.



(a) Epochs-vs-Accuracy (b) Epochs-vs-Precision (c) Epochs-vs-Recall (d) Epochs-vs-F1 Score (e) Epochs-vs-Loss

Fig. 16. Character Recognition Results using Inception V3.

6. **Dense Layer:** The dense layer is a layer of neurons where each neuron is connected to all output neurons in the next layer.

Model Architecture: Architecture is shown below in Fig. 5 and the flow chart is shown in Fig. 7.

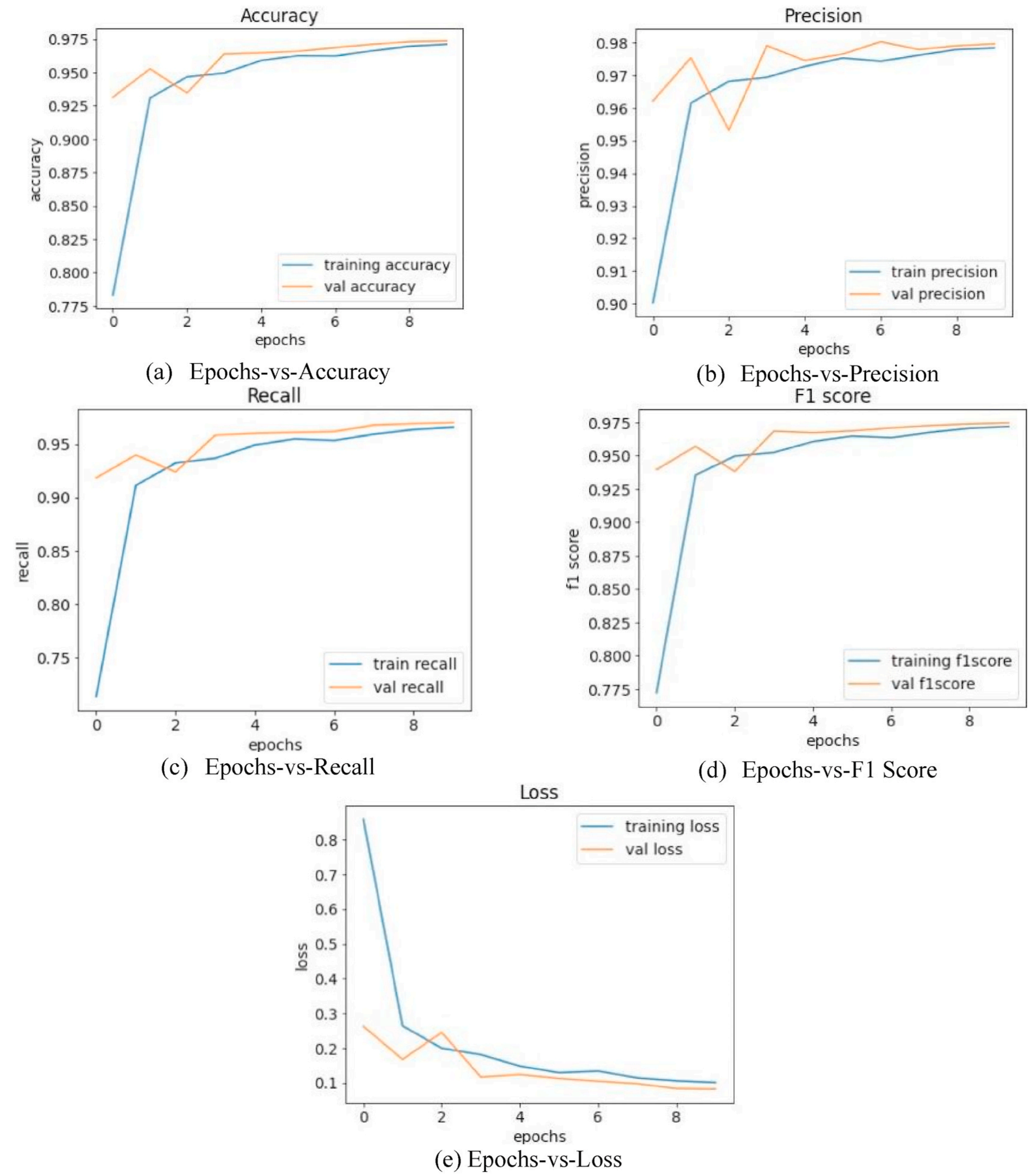
1. First a sequential object is created.
2. Add first input convolutional layer with input tensor size (28,28) with 16 output filters. **ReLU Activation Function:** It is the most common activation function used in deep learning models. It can be written as shown in Equation (5) and also shown in graph form in Fig. 6.

$$f(x) = max(0, x) \quad (5)$$

3. After the first conv2d layer add another 3 conv2d layers as mentioned above.
4. A max pooling layer follows the final conv2d layer, with size of window (4,4).
5. Next a dropout layer is added to the model with the dropout rate = 0.4.
6. Flatten layer is added next to the model to represent the layers in 1 dimension.
7. In the last two dense layers are added to the model. Last dense layer has 36 outputs for 26 alphabets and 10 digits and SoftMax as the activation function.

3.4.2. *MobileNet*

MobileNet V2 is a 53 layers CNN model. It is pretrained on the ImageNet dataset. It was designed for image classification and mobile vision. It takes less computation power for transfer learning. The prime purpose of this model is the reduction of the size of the deep learning model and efficient usage in mobile versions using technique called Depth-wise Separable Convolution [23] as shown in Fig. 8.



(a) Epochs-vs-Accuracy (b) Epochs-vs-Precision (c) Epochs-vs-Recall (d) Epochs-vs-F1 Score (e) Epochs-vs-Loss

Fig. 17. Character Recognition Results using ResNet50.

The block diagram of the MobileNetV2 consists of the first fully convolution layer which consists of 32 filters, it is further added by 19 remaining bottleneck layers as illustrated in the architecture in Fig. 9 below.

We have used MobileNets architecture as shown in Fig. 9, with pre-trained weights on an ImageNet dataset. The model is imported directly from Keras Application package. There are several crucial elements that re-quire to be noticed here:

1. Discard the last output layers of default MobileNets architecture and replace it with our desired output layer. Output layer would contain 36 nodes associating to 36 characters. The shape of our input images is (80,80,3), we need to configure our input layer as the same dimension.
2. If training = True, then define all layers in base model as trainable layers (weight can be updated during training)
3. After it initializes the learning rate and decay value and compiles the model with losses and metrics as categorical cross entropy and accuracy respectively.

3.4.3. Inception V3

Inception V3 is 48 layers deep, convolutional neural network model. It is mostly used as an image recognition model. The ImageNet dataset has millions of images. These images have bounding boxes mentioning the exact location of the labelled objects [8].

We have used inception v3 architecture as shown in Fig. 10, with pre-trained weights on an ImageNet dataset. The model is imported directly from Keras Application package. Some changes are made in the last layers according to the classes.

1. Discard the last output layers of default inception v3 architecture and replace it with our desired output layer. Output layer would contain 36 nodes associating to 36 characters. The shape of our input images

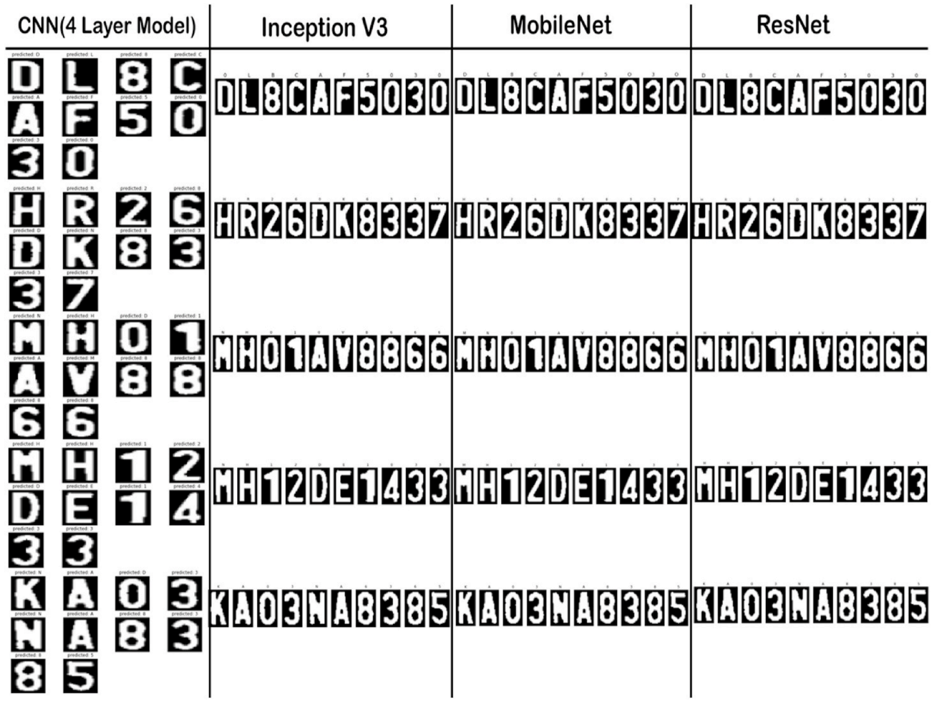


Fig. 18. Model character prediction result.

	Parameters	CNN	MobileNet	Inception V3	ResNet
0	Accuracy	0.9850	0.9639	0.850385	0.9737
1	Precision	0.9803	0.9729	0.907655	0.9796
2	Recall	0.9780	0.9611	0.830481	0.9699
3	F1 Score	0.9791	0.9670	0.866688	0.9747
4	Loss	0.0425	0.1224	0.213159	0.0816

Fig. 19. Performance results under various metrics: Accuracy, Precision, Recall, F1 Score & Loss.

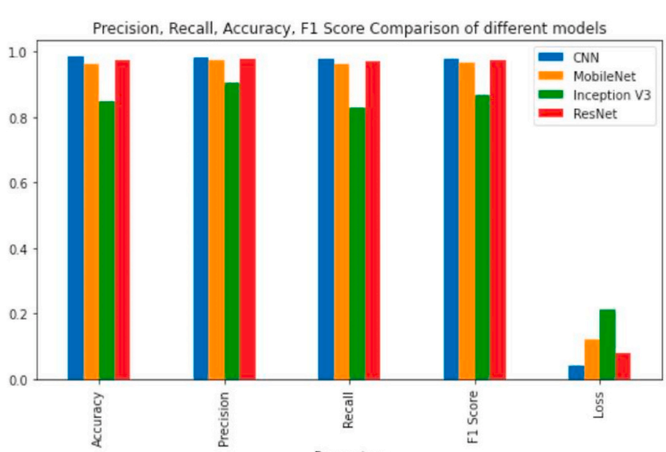


Fig. 20. Comparison of evaluation scores: Accuracy, Precision, Recall, F1 Score & Loss.

is (80,80,3), we need to configure our input layer as the same dimension.

2. If training = True, then define all layers in base model as train-able layers (weight can be updated during training)
3. 3. After it initializes the learning rate and decay value and compiles the model with losses and metrics as categorical cross entropy and accuracy respectively.

3.4.4. ResNet50

ResNet50 is a deep Residual Neural Network that is 50 layers deep and is pre-trained using the gigantic ImageNet dataset as its training data. The model is imported directly from Keras Application package. Some changes are made in the last layers according to the classes as the changes are made in MobileNet and inception v3 [24].

There are five stages in the ResNet50 model. Where, each stage contains a convolution and an identity block. There are three layers in each convolution block and also in each identity block. Fig. 11 depicts the complete structure of the ResNet50 deep learning framework.

4. Implementation results

As mentioned in the previous section, the proposed ALPDR framework consists of four major phases – License Plate Extraction (region acquisition), Image Pre-Processing (enhancement of details), Character Segmentation (extracting the characters from the region), and Character Recognition. For the first three steps, single techniques have been utilized, while for the final step of recognition, four different models have been used – CNN, MobileNet, Inception V3 and ResNet50. The implementation of the proposed ALPDR system is done using OpenCV, with the image processing tool and a PC of 1.2 GHz of Processing, 8 GB of RAM.

4.1. Datasets

We have used two different datasets for our image classification. Indian License Plate dataset consists of 1100 images of Indian license plates of different vehicles. We used this dataset to train our Haar Cascade classifier to detect license plates from the original image of vehicle. Next dataset is for character recognition for License plates. This dataset contains 32,000 images for 36 different classes. Classes are namely of alphanumeric characters(0–9 and A to Z).

4.2. License plate extraction & image pre-processing results

As described in Section 3.1 and Section 3.2 the License Plate Extraction and Image Pre-processing has been achieved and implemented, the result is illustrated in Fig. 12.

4.3. Character segmentation results

After the License Plate Extraction and Pre-processing the Character Segmentation is done as described in Section 3.3. After the Character Segmentation the result is shown in Fig. 13.

4.4. Character recognition

In the character recognition step, we proposed four different models such as CNN, MobileNet, Inception V3 and ResNet50. The implementation results of all these Character Recognition methods are described in subsequent Sections. The performance of these methods is recorded via five evaluation measures – Precision, Accuracy, Recall, F1Score, and Loss.

- **Accuracy:**It is the parameter which can be calculated as the ratio of total number of correct predictions (True Positives + True Negatives) to the total number of productions (True Positives + True Negatives + False Positives + False Negatives). In our proposal, accuracy tells us how accurately and precisely the license plate has been detected and recognized by the proposed models. Mathematically it can be represented as shown in Equation (6).

$$Accuracy = \frac{True\ Positive(TP) + True\ Negative(TN)}{(True\ Positive(TP) + False\ Positive(FP) + True\ Negative(TN) + False\ Negative(FN))} \tag{6}$$

- **Precision:** It a parameter can be calculated as the ratio between the true positives and all the positives including false positive and true positive. In our proposal it would be the measure of the license plates that has been correctly identified rates character out of all the license plates Mathematically it can be represented as shown in Equation (7).

$$Precision = \frac{True\ Positive(TP)}{(True\ Positive(TP) + False\ Positive(FP))} \tag{7}$$

- **Recall:** It is the parameter which measures that the proposed model is correctly identifying the True Positives. In our proposal it tells us that how many correct license-plates have been recognized. Mathematically it can be represented as shown in Equation (8).

$$Recall = \frac{True\ Positive(TP)}{(True\ Positive(TP) + False\ Negative(FN))} \tag{8}$$

- **F1 Score:** This parameter gives the result in the basis of Precision and Recall. In some applications, Precision is an important parameter and, in some applications, Recall has an important role for decision making. F1 score is calculated as the harmonic mean of the Precision and Recall values, emphasizing the importance of both the parameters. Mathematically it can be represented as shown in Equation (9).

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

- **Loss:** Loss or Log-Loss is a parameter which is able to indicate that how close the prediction probability is to the corresponding true value or actual value. If the prediction probability diverges from the true value or actual value then the loss will be more.

Character recognition results using CNN in various forms is shown in Fig. 14, Figs. 15 and 16 respectively.

4.4.1. Character recognition using CNN

4.4.2. Character recognition using MobileNet

4.4.3. Character recognition using inception V3

4.4.4. Character recognition using ResNet50

Character Recognition Results using ResNet50 results with graph is shown in Fig. 17:

4.4.5. Character recognition comparison result

Model character result is shown in Fig. 18:

As the comparison result shown in Figs. 19 and 20 the scene and outperform the other three character-recognition models MobileNet, Inception V3, ResNet 50. The accuracy with the CNN proposed model is around 98.5% and having the minimum loss of 4.25% signifies that if in the proposed model the CNN will be used in the character recognition step the system performs well.

5. Conclusion

In this paper an Automatic License Plate Detection and Recognition (ALPDR) system has been proposed having four steps namely License Plate Extraction, Image Pre-processing, Character Segmentation and Character Recognition. For the License Plate Extraction, Image Pre-Processing and Character Segmentation, unique methods have been proposed. As the character recognition is a very vital phase of license plate recognition and detection, we proposed four different methods for character recognition which includes Convolution Neural Network (CNN), MobileNet, Inception V3, ResNet 50. The model has been implemented and the results of each stage are shown and discussed in this paper. For the Character Recognition as four different methods are used, we compared the performance of all four methods on the basis of evaluation metrics namely Precision, Accuracy, Recall, F1-Score and Loss. Based on the results we can conclude that the Character Recognition with Convolutional Neural Network (CNN) performs better with accuracy of 98.5% and loss of 4.25%. This proposed system performs well in different conditions such as low illumination, blur or hazy image and tilt license plate.

CRediT authorship contribution statement

M.A. Jawale: Conceptualization, and design of study, Revising the manuscript critically for important intellectual content. **P. William:** Conceptualization, and design of study, Data curation, Writing – original draft, Revising the manuscript critically for important intellectual content. **A.B. Pawar:** Data curation, Formal analysis, Writing – original draft. **Nikhil Marriwala:** Formal analysis, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] X. Ascar Davix, C. Seldev Christopher, D. Judson, Detection of the vehicle License Plate using a kernel density with default search radius algorithm filter, in: International Journal of Light and Electron Optics, Optik, Elsevier, 2018, pp. 1–8, 2020.
[2] P. Ravi Kiran Varma, Srikanth Ganta, B. Hari Krishna, S.V.S.R.K. Praveen, A novel method for Indian vehicle registration Number Plate Detection and recognition using image processing techniques detection and recognition using image processing techniques, in: International Conference on Computational Intelligence and Data Science (ICCIDS 2019) vol. 167, Elsevier, Procedia Computer Science, 2020, pp. 2623–2633.
[3] Hanit Karwal, Akshay Girdhar, Vehicle Number Plate Detection System for Indian Vehicles", IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 8–12.
[4] P. William, A. Shrivastava, N. Shunmuga Karpagam, T.A. Mohanaprakash, K. Tongkachok, K. Kumar, Crime analysis using computer vision approach with machine learning, in: N. Marriwala, C. Tripathi, S. Jain, D. Kumar (Eds.), Mobile Radio Communications and 5G Networks, Lecture Notes in Networks and Systems, vol. 588, Springer, Singapore, 2023, https://doi.org/10.1007/978-981-19-7982-8_25.
[5] Fei Xie, Ming Zhang, Jing Zhao, Jiquan Yang, Yijian Liu, Xinyue Yuan, A robust License Plate Detection and character recognition algorithm based on a combined feature extraction model and BPNN, J. Adv. Transport. Hindawi 2018 (2018) 1–14.
[6] Ibtissam Slimani, Abdelmoghit Zaarane, Wahban Al Okaishi, Issam Atouf, Abdellatif Hamdoun, An Automated License Plate Detection and Recognition System Based on Wavelet Decomposition and CNN" Array vol. 8, Elsevier, 2020, pp. 1–7.
[7] K.B. Sathya, S. Vasuhi, V. Vaidehi, Perspective vehicle License Plate transformation using deep neural network on genesis of CPNet, in: Third International Conference on Computing and Network Communications vol. 171, Elsevier, Procedia Computer Science, 2020, pp. 1858–1867.
[8] Swati Jagtap, Analysis of feature extraction techniques for vehicle Number Plate Detection, Int. J. Comput. Sci. Inf. Technol. 6 (6) (2015) 5342–5346.
[9] K. Tejas, K. Ashok Reddy, D. Pradeep Reddy, M. Rajesh Kumar, Efficient License Plate Detection by Unique Edge Detection Algorithm and Smarter Interpretation through IoT", 7th International Conference on Soft Computing and Problem Solving, 2020, pp. 1–11.
[10] Priyanka Prabhakar, P. Anupama, S.R. Resmi, Automatic vehicle Number Plate Detection and recognition, in: International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014, pp. 185–190.
[11] Choong Young Jung, Keong Lee Kim, Tan Chye Cheah, License Plate number detection and recognition using simplified linear-model, J. Crit. Rev. 7 (3) (2020) 55–60.
[12] Olamilekan Shobayo, Ayobami Olajube, Ohere Nathan, Modupe Odusami, Obinna Okoyeigbo, Development of smart plate number recognition system for fast cars with web application, Appl. Computat. Intellig. Soft Comput. Hindawi 2020 (2020) 1–7.
[13] Weihong Wang, Jiaoyang Tu, Research on License Plate Recognition algorithms based on deep learning in complex environment, IEEE Access 8 (2020) 91661–91675.
[14] Palaiahnakote Shivakumara, CNN-RNN based method for license plate recognition, CAAI Trans. Intellig. Technol. 3 (3) (2018) 169–175.
[15] P. William, A. Shrivastava, P.S. Chauhan, M. Raja, S.B. Ojha, K. Kumar, Natural Language processing implementation for sentiment analysis on tweets, in: N. Marriwala, C. Tripathi, S. Jain, D. Kumar (Eds.), Mobile Radio Communications and 5G Networks, Lecture Notes in Networks and Systems, vol. 588, Springer, Singapore, 2023, https://doi.org/10.1007/978-981-19-7982-8_26.
[16] S. Mishra, S. Choubey, A. Choubey, N. Yogeesh, J. Durga Prasad Rao, P. William, Data extraction approach using natural language processing for sentiment analysis, in: International Conference on Automation vol. 2022, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2022, pp. 970–972, https://doi.org/10.1109/ICACRS55517.2022.10029216.
[17] Deepak Narayan Paithankar, Abhijeet Rajendra Pabale, P. William Rushikesh Vilas Kolhe, Prashant Madhukar Yawalkar, Framework for implementing air quality monitoring system using LPWA-based IoT technique, Measurement: Sensors 100709 (2023), https://doi.org/10.1016/j.measen.2023.100709. ISSN 2665-9174.
[18] K. Gupta, S. Choubey, Y. N, P. William, V. T.N, C.P. Kale, Implementation of motorist weariness detection system using a conventional object recognition technique, in: International Conference on Intelligent Data Communication Technologies and Internet of Things vol. 2023, IDCIoT), Bengaluru, India, 2023, pp. 640–646, https://doi.org/10.1109/IDCIoT56793.2023.10052783.
[19] P. William, Y. N, V.M. Tidake, S. Sumit Gondkar, C. R, K. Vengatesan, Framework for implementation of personality inventory model on natural language processing with personality traits analysis, in: International Conference on Intelligent Data Communication Technologies and Internet of Things vol. 2023, IDCIoT), Bengaluru, India, 2023, pp. 625–628, https://doi.org/10.1109/IDCIoT56793.2023.10053501.
[20] Mark Sandler, Mobilenetv2, Inverted residuals and linear bottle- necks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[21] Kaiming He, Deep residual learning for image recognition", in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[22] P. William, A. Shrivastava, H. Chauhan, P. Nagpal, V.K. T. N, P. Singh, Framework for intelligent smart city deployment via artificial intelligence software networking, in: 3rd International Conference on Intelligent Engineering and Management vol. 2022, ICIEM), 2022, pp. 455–460, https://doi.org/10.1109/ICIEM54221.2022.9853119.
[23] P. William, Y. N, S. Vimala, P. Gite, S.K. S, Blockchain technology for data privacy using contract mechanism for 5G networks, in: 3rd International Conference on Intelligent Engineering and Management vol. 2022, ICIEM), 2022, pp. 461–465, https://doi.org/10.1109/ICIEM54221.2022.9853118.
[24] S. Yuvaraj, A. Badholia, P. William, K. Vengatesan, R. Bibave, Speech recognition based robotic arm writing, in: V. Goyal, M. Gupta, S. Mirjalili, A. Trivedi (Eds.), Proceedings of International Conference on Communication and Artificial Intelligence, Lecture Notes in Networks and Systems, vol. 435, Springer, Singapore, 2022, https://doi.org/10.1007/978-981-19-0976-4_3.
[25] S.S. Gondkar, D.B. Pardeshi, P. William, Innovative system for water level management using IoT to prevent water wastage, in: International Conference on Applied Artificial Intelligence and Computing vol. 2022, ICAAIC), 2022, pp. 1555–1558, https://doi.org/10.1109/ICAAIC53929.2022.9792746.
[26] P. William, et al., Systematic approach for detection and assessment of dark web threat evolution, in: Romil Rawat, et al. (Eds.), Using Computational Intelligence for the Dark Web and Illicit Behavior Detection, IGI Global, 2022, pp. 230–256, https://doi.org/10.4018/978-1-6684-6444-1.ch013.
[27] R. Jadhav, A. Shaikh, M.A. Jawale, A.B. Pawar, P. William, System for identifying fake product using blockchain technology, in: 7th International Conference on Communication and Electronics Systems vol. 2022, ICCES, 2022, pp. 851–854, https://doi.org/10.1109/ICCES54183.2022.9835866.
[28] P. William, N.K. Darwante, A.B. Pawar, M.A. Jawale, A. Verma, Framework for implementation of smart driver assistance system using augmented reality, in: N. Marriwala, C. Tripathi, S. Jain, D. Kumar (Eds.), Mobile Radio Communications and 5G Networks, Lecture Notes in Networks and Systems, vol. 588, Springer, Singapore, 2023, https://doi.org/10.1007/978-981-19-7982-8_30.