IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Indian License Plate Recognition using Deep Learning

[1]Nadagani Ajesh Raj, [2]Durga Ganga Rao Kola
[1]PG Student, [2]Assistant Professor
[1]Department of Electronics and Communication Engineering, UCEK(A), JNTU Kakinada, Andhra Pradesh, India.
[2]Department of Electronics and Communication Engineering, UCEK(A), JNTU Kakinada, Andhra Pradesh, India.

***Abstract*:** The detection and recognition of a license plate is especially an important aspect of Automatic License Plate Recognition (ALPR). In this work, a deep learning model is used to address the problem of license plate digits detection and recognition. The Convolutional Neural Network (CNN) layers are used to detect the license plate. Due to the dust and sand particles on license plates, the recognition of characters is difficult, so for optimal recognition, Optical Character Recognition (OCR) is used. The objective is to detect and recognize the Indian car license plate digits. Different sets of car images are used for training the Convolutional Neural Networks. By using CNN, the system achieves approximately 99.5% mean Average Precision (mAP) with a plate loss of 0.01627 in the detection of the Indian car license plate in real-time.

***Index Terms*— Optical Character Recognition, Convolutional Neural Networks, Automatic License Plate Recognition, License Plate Recognition, Mean Average Precision.**

1. INTRODUCTION

License plate recognition (LPR) is an important research topic with the recent improvement in intelligent transport systems. LPR consists of many applications in the domain of transportation. There are mainly four steps for the LPR: a collection of images, detection of the object, segmentation, and recognition of characters as shown in Fig.1.

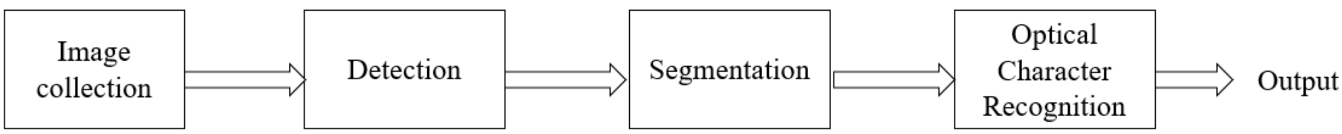

Fig.1: Steps for the Recognition of License Plate.

In the first step of LPR, Images are collected from a video, camera, or an open dataset for image collection. The second step is object detection in which all the relevant objects are detected. This is mainly done by using the Edge detection technique which is a common technique. After the detection phase is completed, segmentation is done for the detection of alphabets and numerals. The final step is to recognition of all the segmented digits and alphabets by using Optical Character Recognition (OCR).

The Convolutional Neural Network (CNN) has become common for object detection because of the development of GPUs that support high computational complexity. Impressive results are produced by using CNN for image classification [3], digit recognition [4], object detection [5]. Region-based CNN can be used for increasing the performance, but the computational complexity is increased. The image is divided into four categories using algorithms that employ morphological categorization approaches: character-based, edge-based, texture-based, and color-based [6]. Among these the most commonly used detection is edge-based detection for detecting the license plate digits.

Since the number plate consists of a white background with black letters, edge-detection can easily detect the features. Although the speed of computation is rapid, it sometimes detects unwanted edges. This is due to the many colors in the image or when there is a difference between white and black that interferes with the detection in edge detection. Differentiating the colors of the car body and license plate is how the color-based approach works. The HSI color model was proposed by Deb and Jo [7] to detect license plate region which is then verified by histogram after the number plate is detected. Mathur et al. [8] recommended the input as a gray-scale image to decrease noise from the license plate. When there is the same color of the car body and the license plate, the color-based ethod struggles to distinguish the license plates. The work takes gray-scale and character-based approaches to learn and detect the license plate digits since it uses CNN which extracts the features directly. The capability of CNN's classification provides guaranteed detection of characters with high performance.

The organization of the paper is as follows: The Pre-processing and Data Collection are discussed in section 2, The Proposed Methodology is discussed in section 3, The Experimental Setup and Results are discussed in section 4, Finally, the Conclusion and Future Scope are given in Section 5.

2. PRE-PROCESSING AND DATA COLLECTION

For the detection and recognition of characters in license plates using CNN, the considered dataset should be collected first. The dataset consists of 935 images of cars. The images are taken, and they are labeled for the license plate. After labeling all the images, the images are made into a single class. The dataset is split into three parts for training, testing, and validating in the partitions of 70%,20%,10% then the images are then made into a dataset. The flow of the preparation of the dataset can be seen in Fig.2.

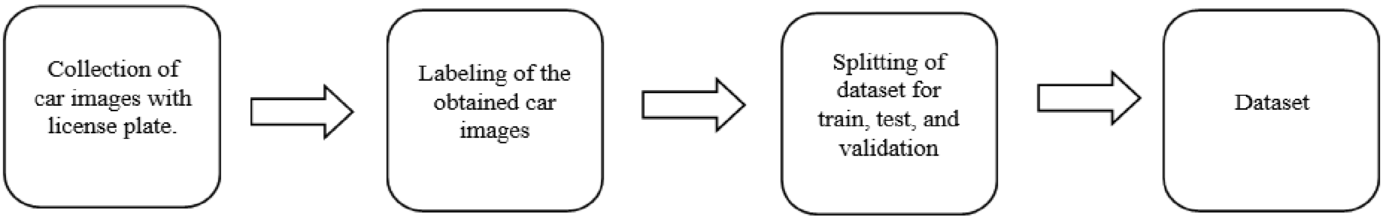

Fig.2: Dataset preparation flow diagram.

3. PROPOSED APPROACH FOR DETECTION AND RECOGNITION

The main objective of license plate detection is to detect the license plate precisely. To increase the precision of the detection, CNN is used. CNN has the capability to differentiate classes from one another. It consists of different types of layers in its architecture. The architecture of the CNN consists of 13 convolutional layers and the considered dataset is processed through these layers. By using the modified architecture of You Only Look Once (YOLO) [9] this CNN architecture is created. First, the dataset is given to the CNN, and then the weights and data are labeled according to the classes obtained. All the weights of the dataset are saved for the training of different CNN models. These labels and weights are used for detecting the digits and characters on the license plate. YOLO is used for the detection of objects, and the recognition of the digits is done by using the OCR [10] as shown in Fig.3.

The dataset is split into three parts for the training, testing, and validation of the data. The dataset is trained for 50 to 500 epochs depending on the model of CNN.

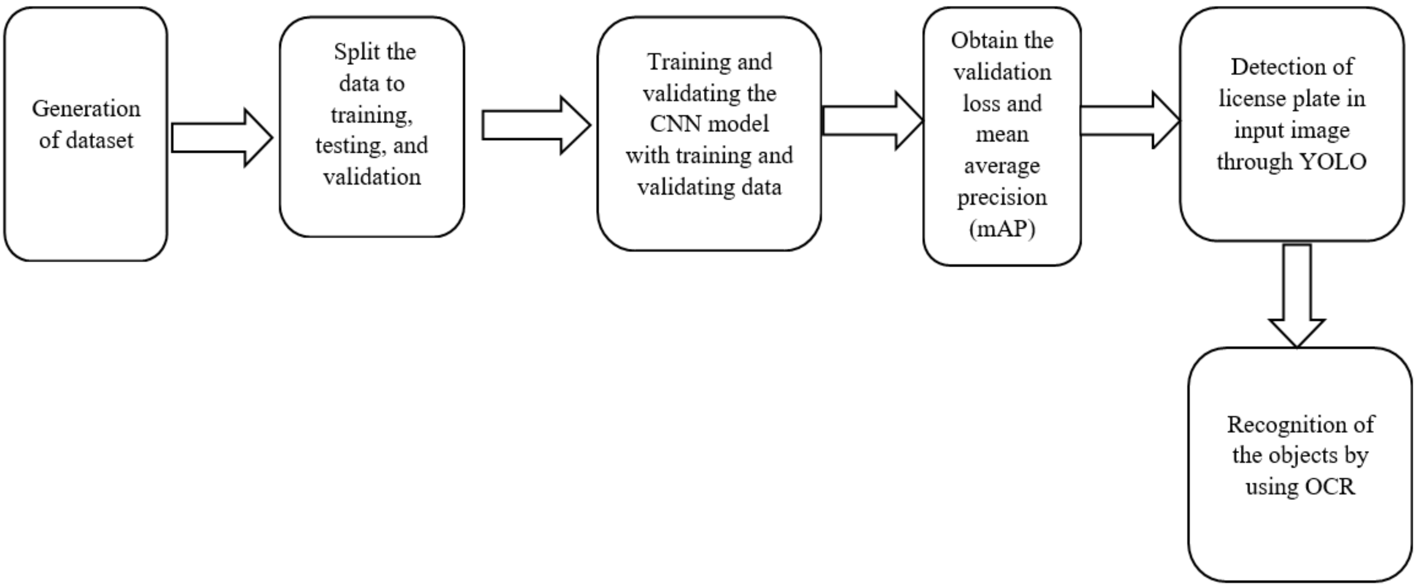

Fig.3: Proposed architecture flow diagram.

3.1 Generation of dataset:

The generation of the dataset is done by collecting various images of Indian car license plates and labeling the images at the license plate region. The number of images that are taken for this work is 935. By generating the bounding boxes at the License plate region, the labeling is done. After labeling the images, these images can be used for the training of the CNN model.

3.2 Splitting the data:

The data is split into three parts for training, testing and validation. In this method, the data is split into 70%, 20%, 10% which are partitions for training, testing, and validation respectively. This dataset is created by taking many images of Indian car License Plates and generating labels for the dataset.

3.3 Training and validating the CNN model:

After the generation of the dataset, the CNN model is trained and validated by using the dataset. The training is done for 50 epochs for this work. After the training, the model is validated with the validating data. Then the testing of the model is done with the testing images that are present in the dataset.

3.4 mAP and validation loss:

The mAP [9] curves are obtained after training the model. The validation loss [9] curves can be obtained after training. The mAP is obtained along with Precision and Recall (PR) curves [9]. These curves are used for the generation of mAP curves. These curves vary with different models.

3.5 Detection with CNN:

In this phase, the weight file of the model after the training and validation is generated. This file is obtained, and it is given to the CNN for object detection. The CNN takes the weights of the trained model and maps the weights to the input image for the detection of the License Plate.

A bounding box is generated for the detected object. The object in this work is the license plate. After the generation of the bounding box to the relevant object, the CNN detects the object. In this way, the object is said to be detected by CNN.

3.6 Recognition by using OCR:

After the detection of the objects by using CNN, the recognition of the objects is done by the OCR [10]. OCR recognizes the characters on the License plate. So, the detected object is the license plate in this work which is sent to the OCR for recognition. In this phase, the characters and the numerals are recognized by the OCR. The recognition and detection precision are based on the iterations in which the model is trained [10].

4. EXPERIMENTAL SETUP AND RESULTS

The experimental procedure, dataset and results are discussed in this session :

The dataset consists of 935 images of car license plates, and they are labeled at the license plate for the training of the CNN model. The normalization of images is made to the dataset. For training, testing, and validation, the dataset is split into three parts with ratios of 70/20/10. The architecture of the network is trained with the considered dataset. The batch size is 8 and the max epochs are 50. After the training of the model, the weight file is taken. The CNN is trained with this weight file for the detection of the license plate. After detecting the license plate, the OCR is used for the recognition of the characters and numbers on the license plate. The mAP and PR curves along with Validation loss curves are obtained after the process.

The considered dataset is trained in the CNN architecture and the validation is also done in it. The Validation Loss of the CNN models are shown in Fig.4, Fig.5, and Fig.6. The mAP is 99.5% and the validation loss is 0.01652 for the CNN model. For the purpose of comparison, ResNet-50, and EfficientNet_b0 models are used to get the precision and validations loss. The experimental results are shown in Table-1.The Validation Loss for 50 epochs of the CNN model is 0.01645 whereas the loss of ResNet-50 and EfficientNet_b0 are 0.01645 and 0.01652, respectively. The mAP of the CNN models is shown in Fig.7, Fig.8, and Fig.9. The mAP for 50 epochs of CNN, ResNet-50, and EfficientNet_b0 is 99.5. The comparison of performance metrics of all CNN models can be seen from Fig.10 to Fig.13. The Precision and Recall (PR) curves for 50 epochs of CNN are 99.9% and 99.9%. The Precision and Recall (PR) curves for 50 epochs of ResNet-50 are 99.66% and 99.9%. The Precision and Recall (PR) curves for 50 epochs of EfficientNet_b0 are 99.46% and 99.46%. The real-time detected image is shown in Fig.14.

Table-1: Training and test results

Network models	Epochs	Train/Test			
		mAP	Validation loss	Precision	Recall
ResNet-50	50	99.5%	0.01645	99.66%	99.99%
EfficientNet_b0	50	99.5%	0.01652	99.46%	99.46%
CNN model	50	99.5%	0.01627	99.99%	99.99%

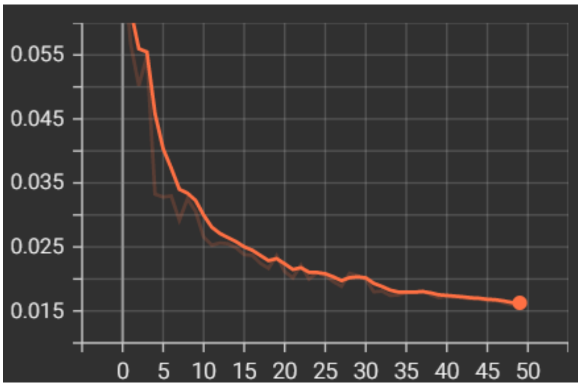

Fig.4: Proposed CNN validation loss

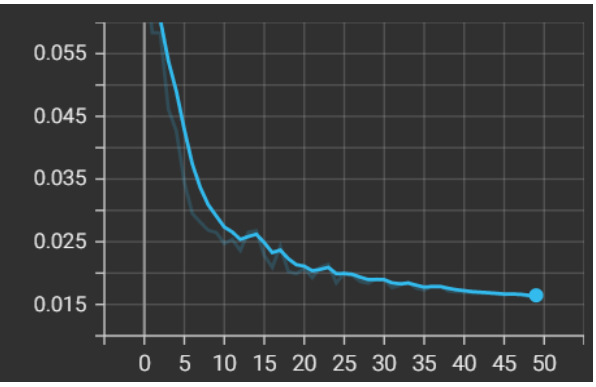

Fig.5: EfficientNet_b0 validation loss

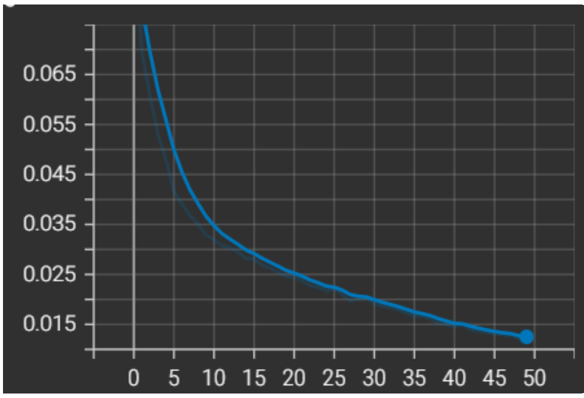

Fig.6: Validation loss of ResNet-50

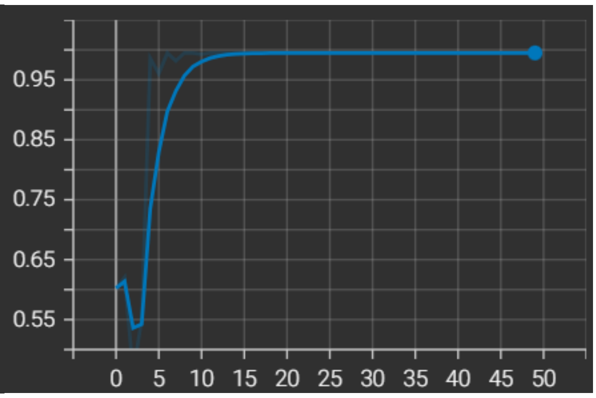

Fig.7: mAP of ResNet-50

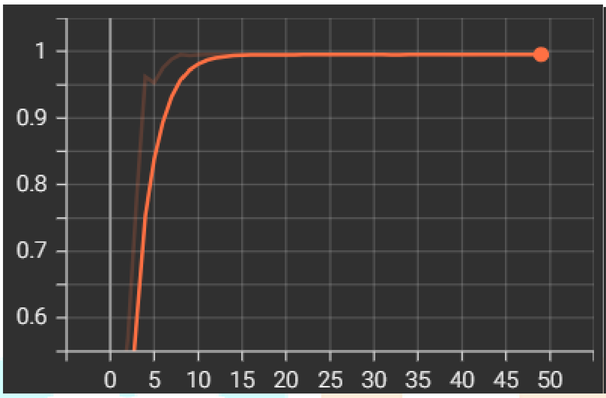

Fig.8: mAP of proposed CNN

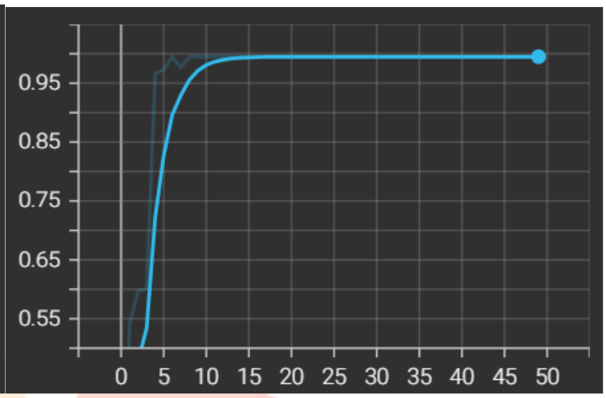

Fig.9: mAP of EfficientNet_b0

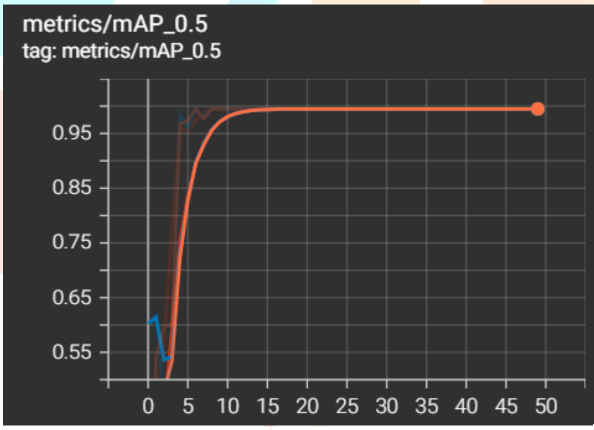

Fig.10: mAP of all models

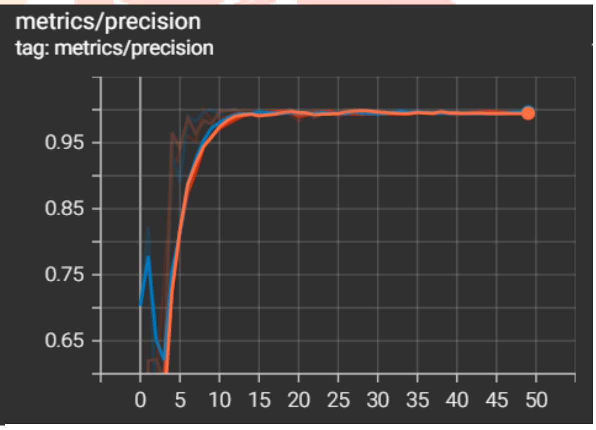

Fig.11: Precision of all models

CNN model
ResNet50
EfficientNet_b0

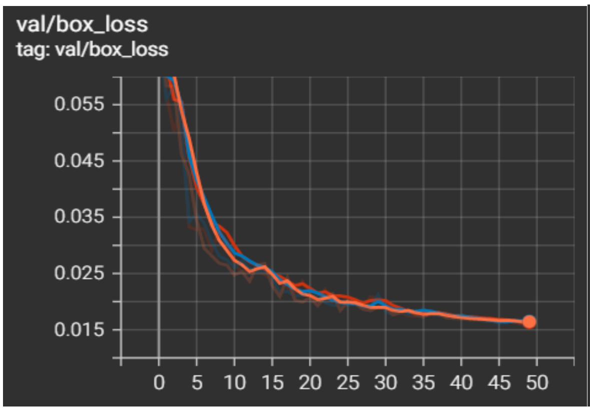

Fig.12: Validation loss of all models.

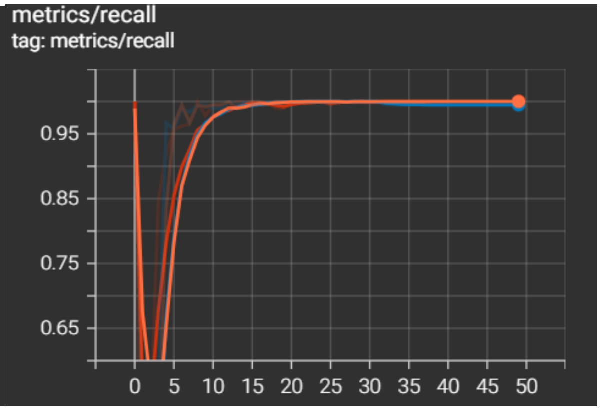

Fig.13: Recall of all models.

Fig.14: Indian car license plate detected and recognized.

5. CONCLUSION

In this paper, a CNN architecture is proposed to obtain the highest Precision and Recall (PR) of 99.99% for the detection and recognition of an Indian license plate with the least digit loss. The CNN model provides 99.5% mAP is obtained and with a Validation Loss of 0.01627 for 70% training, 20% testing, and 10% validation data with 50 epochs of iteration. The detection is done by the CNN model and the character recognition is done by the OCR for real-time applications. In the future, a faster R-CNN can be used to reduce the time taken to train the model with a little sacrifice of accuracy.

6. REFERENCES

[1] A. Safaei1 et al., “Real-time search-free multiple license plate recognition via likelihood estimation of saliency,” ELSEVIER, vol.56, 2016, pp. 15–29.

[2] H. Li1 et al.,” Toward end-to-end car license plate detection and recognition with deep neural networks,” IEEE, Trans. Intell. Transp. Syst, vol.20, no.3, 2019, pp. 1126–1136.

[3] T. Kumar1 et al, “An efficient approach for automatic number plate recognition for low resolution images,” Proceedings of the Fifth International Conference on Network, Communication and Computing, ICNCC ’16, vol.5, 2016, pp. 53–57.

[4] H. Li1 et al., “Reading car license plates using deep neural networks, Image and Vision Computing,” ELSEVIER, vol. 72, 2018, pp. 14–23.

[5] G. Hsu et al., “Application-oriented license plate recognition,” IEEE, vol.62, no.2, 2013, pp.552–561.

[6] C.E. Anagnostopoulos1 et al., “License plate recognition from still images and video sequences: a survey,” IEEE, vol.9, no.3, 2008, pp. 377–391.

[7] K. Deb1 and K. Jo2, “HSI color-based vehicle license plate detection,” 2008, International Conference on Control, Automation and Systems, 2008, pp. 687–691.

[8] N. Mathur1 et al., “A novel approach to improve Sobel edge detector,” ELSVIER, vol.93, 2016, pp.431–438.

[9] Hendry1 and Rung-Ching Chen2, “Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning,” ELSEVIER, vol.87, 2019, pp. 47-56.

[10] N.Sahu1 and M.Sonkusare2, “A Study on Optical Characters Recognition Techniques,” IJCSITCE, vol.4, no.1, 2017, pp. 1-13

[11] Y. Seok1 and Shik Shin2, “Hierarchical convolutional neural networks for fashion image classification,” ELSEVIER, vol.116 2019, pp. 328–339.

[12] M. Paoletti1 et al., “A new deep convolutional neural network for fast hyperspectral image classification,” ELSEVIER, vol.145, 2018, pp. 120–147.

[13] S. Wan1 et al., “Deep convolutional neural networks for diabetic retinopathy detection by image classification,” ELSEVIER, vol.72, 2018, pp. 274–282.

[14] N. Sharma1 et al., "An analysis of convolutional neural networks for image classification," ELSEVIER, vol.132, 2018, pp. 377–384.

[15] X. Wang1 et al., "Semi-supervised adaptive feature analysis and its application for multimedia understanding," vol.77, 2017, pp. 3083–3104.

[16] A. Baldominos et al., "Evolutionary convolutional neural networks: an application to handwriting recognition," ELSEVIER, vol.283, 2018, pp. 38– 52