




ALPR - An Intelligent Approach Towards Detection and Recognition of License Plates in Uncontrolled Environments

Akshay Bakshi¹, Sudhanshu Gulhane², Tanish Sawant¹, Vijay Sambhe¹,
and Sandeep S. Udmale¹ (✉) 

¹ Department of Computer Engineering and Information Technology,
Veermata Jijabai Technological Institute (VJTI), Mumbai 400019, Maharashtra, India
{apbakshi_b19,tmsawant_b19}@ce.vjti.ac.in,
{vksambhe,ssudmale}@it.vjti.ac.in

² Department of Electronics and Telecommunications Engineering,
Pune Institute of Computer Technology (PICT), Pune 411043, Maharashtra, India
sudhanshugulhane072@gmail.com

Abstract. Most existing Automatic License Plate Recognition (ALPR) approaches focus on images containing approximately frontal views. The considerable variation of LP across complicated environments and perspectives remains a massive challenge for a robust ALPR. This work proposes a comprehensive ALPR paradigm emphasizing unrestricted express screenplays in which the LP may be significantly influenced by diverse shooting angles, illumination circumstances, and complicated surroundings. This system integrates a Spatial Transformer Network, which can catch and repair numerous distorted LPs in an image so that all the plates are consistently aligned. Then, a convolutional neural network is sketched to determine LP characters containing various font styles and sizes. We evaluated the system with a data set containing annotations for a challenging LP image set from multiple areas and acquisition states. The experimental outcomes reveal that our proposed ALPR paradigm attains adequate recognition accuracy compared to existing methods.

Keywords: Convolutional Neural Network (CNN) · License Plate (LP) · Spatial Transformer Network (STN) · YOLO

1 Introduction

Automatic License Plate Recognition (ALPR) routines offer various applications, including identifying stolen vehicles, monitoring traffic, smart toll collection, etc. [9, 23]. The recent advancements in deep learning (DL) and parallel computing have contributed to achieving excellent performance in several digital image/video applications, such as optical character recognition and object detection and recognition, which have tremendously improved ALPR systems. Recently, convolutional neural network (CNN) has achieved exceptional performance and have been the primary machine learning approach for LP detection

and recognition [2, 6, 10–13, 15, 25, 27]. Several ALPR commercial systems have also been employing DL methods. They are usually integrated with web services and large data centers to process millions of vehicle images daily and constantly improve the system. Some of the example systems to be mentioned are: OpenALPR¹, Sighthound², and Amazon Rekognition³.

Moreover, the CNN-based object identification routines have become famous for LPR with the establishment of DL. Typically, faster regions with CNN (R-CNN) [21], Single Shot MultiBox Detector (SSD) [14], and You only look once (YOLO) [18] models are employed. Faster R-CNN [21], a modified version of R-CNN and fast R-CNN that forgoes time-consuming strategy, i.e., selective search, allows the architecture to understand the area manifestos. In this work, a NN has been utilized to forecast the region proposals rather than a particular search procedure to determine the area manifestos on the feature map (FM). Praveen Ravirathinam and Arihant Patawari in [16] demonstrated the effective handling of faster R-CNN in the detection of LP. The proposed model could also detect titled and non-rectangular plates. The mAP of their model went relatively low since it could not catch small-scale images. The study in [11] presented a robust object detection model using Fast Yolo and Yolov2 to detect LP in simple and realistic conditions.

Despite the advances in this field, most approaches focus on recognizing LP in controlled environments, assuming a frontal view of the vehicles and LP. The current challenges in ALPR include image distortion, image quality degradation, weather (snow, rain, etc.), variable illumination conditions, etc. A more permissive picture-gathering setting (e.g., a police car using a camera to track down an unlawful vehicle) could result in slanting vision. In such cases, the LP may be severely distorted and, thus, challenging to recognize, for which even existing standard commercial solutions struggle.

This paper proposes a comprehensive ALPR paradigm capable of performing well over various unrestricted capture screenplays and camera arrangements. We integrate a transformation module to estimate and rectify the distortion and improve the character recognition performance. An additional contribution is the collection of images from natural scenes, which cover various challenging scenarios and contain substantial LP distortions. The proposed system could also discover and identify LPs in independent test data sets using the same configuration. The data sets employed in this assignment are publicly available, and the samples can be obtained from SSIG-SegPlate database [4] and the application-oriented license plate (AOLP) data set [7].

2 Materials

This section provides background information about the various vital components employed in the proposed work (Fig. 1).

¹ <https://www.openalpr.com>.

² <https://www.sighthound.com/products/alpr/>.

³ <https://aws.amazon.com/rekognition/>.

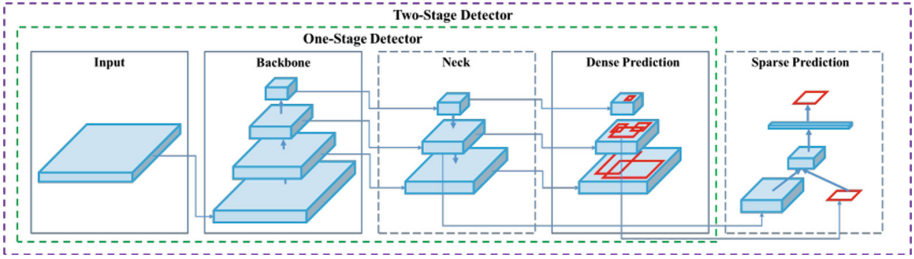


Fig. 1. YOLOv4 architecture [1]

2.1 You Only Look Once (YOLO)

YOLO is one of the one-stage object detector approaches. YOLOv2 [19] model has been built upon YOLO with several incremental enhancements, such as batch normalization, excellent resolution, and anchor boxes. To perform better on smaller objects, YOLOv3 [20] improved upon earlier models by including the bounding box prediction with an objectness score. Also, it attaches links to the backbone network layers and performs predictions at three different degrees of granularity. YOLOv4 [1], a two-stage detector with multiple components, is currently an upgraded version of earlier generations. The higher versions of YOLO are volatile to use as a black box for our proposed methodology. The YOLOv4 model consists of Backbone, Neck, and Head, as shown in Fig. 1.

Backbone: It consists of the CSPDarknet53 model, which detects objects with higher accuracy. Also, it includes the CSPDarknet53 model because it enhances through the MISH and other activation functions [1].

Neck: It consists of a spatial pyramid pooling layer (SPP) and Path Aggregation Network (PAN). SPP plays a crucial role when detecting objects of various scales for adequate context information and, thus, sits between CSPDarknet53 and PAN. It adds a spatial pyramid pooling layer in place of the last pooling layer, which comes after the final convolutional layer. A maximum pool is applied to a sliding kernel of various sizes. The result is then created by concatenating the FMs generated by different kernel sizes [1].

Further, the PAN network's capacity to reliably maintain spatial information, which aids in the proper localization of pixels for mask generation, was chosen, for example, segmentation in YOLOv4. The properties which make PAN so accurate are Bottom-up Path Augmentation, Adaptive Feature Pooling, and Fully-Connected Fusion Network [1].

Head: Bounding box location and categorization has performed using the head (Dense prediction). The procedure is the same as that described for YOLO v3; hence, it detects the score and the bounding box coordinates (x, y, height, and

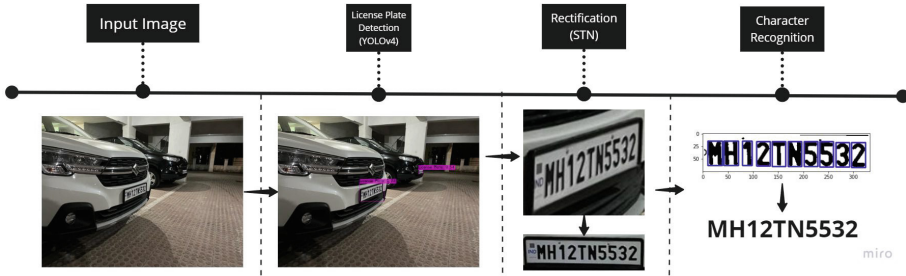


Fig. 2. Proposed system pipeline.

width). The algorithm splits the input image into several grid cells and uses anchor boxes to forecast the likelihood that each cell will contain an object. The result is a vector containing the bounding box coordinates and the class probabilities [1].

2.2 Spatial Transformer Network

Though CNN’s defined as a powerful class of models, they are nonetheless constrained by their inability to be computationally and parameter-efficiently spatially invariant to the input data. The Spatial Transformer Network (STN) [8], a novel teachable module, explicitly permits the spatial modification of information within the architecture. Its differentiable module can be added to current convolutional architectures. It enables the NNs to actively modify FM spatial relationships based on the FM itself without changing the optimization procedure or adding additional training supervision.

3 Proposed Methodology

The proposed structure is demonstrated in Fig. 2 and comprises three main steps: LP Detection, LP Transformation and Rectification, and Character Recognition Network. Given an input image, the custom-trained YOLOv4 model detects LPs in the scene. The detections are cropped and forwarded to an STN to rectify LP images with diverse orientations and surrounding details. The corrected images have a uniform orientation and paltrier surrounding noise. These favorable and repaired detections are presented to a Character Recognition Network.

3.1 License Plate Detection

Detection of LPs is an essential phase in the ALPR process; hence we adopted a reliable model to carry it out. To select the best algorithm, we defined the criteria as 1) The algorithm must have an acceptable performance and recall rate because even a small amount of missed detection will cause the LP detection process to



Fig. 3. Examples of detected LP from testing data set

perform worse. 2) For real-time detection to be reliable, the method must have a high calculation speed. 3) Additionally, since their use in practical applications won't be hampered, the calculating costs should be reasonable. As a result, we carefully chose YOLOv4 as our network for LP detection. When comparing the cost and speed of calculations, the YOLOv4 algorithm is quite effective. Figure 3 reveals that we have refined the YOLOv4 model configurations according to our requirements to specialize it for LP detection. Since we need only one class, i.e., LP, for object detection, we altered the number of classes from 80 to 1 and, thus, the modified value of maximum batch size according to the below formula,

$$max_batches = \min(training_images, \min(classes * 2000, 6000)); \quad (1)$$

Secondly, we altered the number of filters in the convolutional layers using the formula below.

$$filters = (classes + 5) * 3; \quad (2)$$

Thus, we employ a reconfigured model for the detection of LPs.

3.2 Spatial Transformer Network (STN)

The STN suggested in [8] is a differentiable and self-contained module. Thus, it has been added to current convolutional architectures. It streamlines the

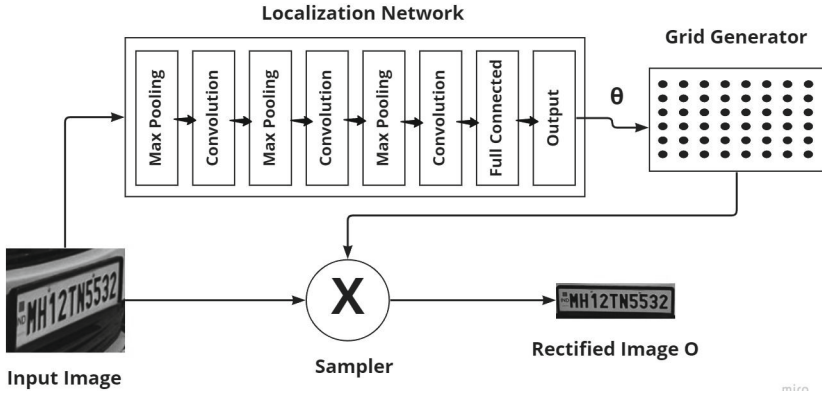


Fig. 4. Structure of STN [8].

subsequent classification work and improves classification results. It strengthens a model’s spatial invariance against non-rigid deformations such as translations, scaling, rotations, and cropping. The suggested model is more resistant to various shooting angles and noises since the input LP photos are first rectified with the trained STN to those with a consistent orientation and reduced noise. Figure 4 shows that it is divided into three divisions. 1) The localization network (LN) derives the affine transformation parameter θ by extracting the key attributes from the input image I . 2) The initial grid is transformed into a new sampling grid by the grid generator based on the input θ . 3) The sampler samples the I by the new grid to create the rectified picture.

Localization Network: The LN accepts the input FM $U \in \mathbb{R}^{H \times W \times C}$ with height (H), width (W), and channels (C) and outputs (θ), the parameters of the transformation \mathbb{T}_θ operated to the FM: $\theta = f_{loc}(U)$. The proportion changes depending on the parameterized kind of transformation; for example, the size of an affine transformation is six dimensions. A final regression layer must be present in the LN function $f_{loc}()$ to obtain the transformation parameters, but it can be fully connected or convolutional.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \theta_{13} \\ \theta_{23} \end{bmatrix} \tag{3}$$

The affine transformation matrix is represented by \mathbb{A}_θ .

$$\mathbb{A}_\theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \tag{4}$$

Table 1. LN Configuration.

Type	Configuration			
Input	Gray-scale distorted LP image			
Layer	Filters	Kernel size	Stride size	Padding
Max_pool.1	–	2×2	2×2	0×0
Conv2D.1	20	5×5	1×1	0×0
Max_pool.2	–	2×2	2×2	0×0
Conv2D.2	20	5×5	1×1	0×0
Max_pool.3	–	2×2	2×2	0×0
Conv2D.3	20	5×5	1×1	0×0
Fully connected	100 hidden units, tanh activation			
Output	6 hidden units, linear output activation			

The LN structure summarized in the Table 1 consists of 3 sets of max-pooling and convolutional layers with a fully connected layer, and finally one output layer.

Parameterised Sampling Grid: Every pixel of the input LP image has a corresponding vector of coordinate, i.e., $K_i = (x_i, y_i)^T$ with the pixel index i . A multiplication operation is performed on θ , and K_i to obtain the affine converted vector of coordinate, i.e., $K'_i = (x'_i, y'_i)^T$. It is expressed as

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = A_\theta \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad (5)$$

$K' = (K'_1, K'_2, \dots, K'_i, \dots, K'_{W \times H})$ are set up to obtain the grid generator's final output, where W and H in our experiments are 270 and 70, respectively.

Differentiable Image Sampler: In order to generate the rectified image O , the sampler samples the original image using the sampling grid K' . Bilinear interpolation, a differentiable module, is used in this sampling process. The STN, which may be trained end-to-end alongside other sections of the model, comprises the LN, the parameterized grid generator, and the image sampler. The STN is created by combining the LN, parameterized grid generator, and image sampler. It can be trained end-to-end with other model components. Please refer to [8] for further information.

3.3 Character Recognition

The recognition process consists of three parts: (1) Preprocessing the rectified image output of STN; (2) Character Segmentation; (3) Recognition of segmented characters.

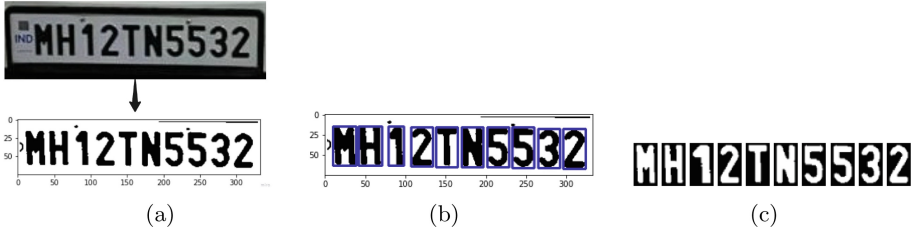


Fig. 5. (a) Binary conversion of the detected plate. (b) Bounding rectangles containing contours. (c) Binary images of segmented characters.

Preprocessing Stage: The rectified LP image is processed to make the character extraction easier. With a single 8-bit channel and values ranging from 0–255, where 0 and 255 indicate black and white, the input image is transformed into a grayscale image. This image is then further altered to become a binary image, where each pixel has a value of either 0 or 1, as shown in Fig. 5(a). Black is represented by the value 0, and white by the value 1. A threshold with a value between 0 and 255 is used to achieve it. We set the threshold value at 200 value. A pixel over 200 value in the grayscale image will be given a value of 1; otherwise, the value is 0.

The binary image is further processed for erosion. Erosion [5] is a technique applied to eliminate unwanted pixels from the object’s boundary, i.e., pixels that have a value of 1 but should contain a value of 0. First, it considers each pixel in the image, then its neighbors (kernel size determines the number of neighbors). The pixel only receives a value of 1 if all of its neighbors also have values of 1, otherwise, it receives a value of 0.

The noise-free image is further processed for dilation. Dilation [5] fills up the absent pixels, i.e., pixels that should have a value of 1 but have a value of 0. Every pixel in the image is first taken into account, followed by its neighbors (kernel size determines the number of neighbors); a pixel is given a value of 1 if at least one of its neighbors is also a 1.

Discovering every contour in the input image is essential for extracting the individual characters from the LP. Curves with the same hue or intensity that connect all the continuous points (along the boundary) are called contours. After locating each contour, we examine it individually and determine the size of each bounding rectangle, as shown in Fig. 5(b). Once we have the dimensions of the bounding rectangles, we adjust the parameters and filter the necessary rectangles that contain the required text.

$$W = range\{0, \frac{input_length}{character_count}\} \tag{6}$$

$$L = range\{\frac{W}{2}, 4 * (\frac{W}{5})\} \tag{7}$$

Using the above equations, we perform a dimension comparison. The rectangles accepted have width and length in the range specified. To achieve this, we

Table 2. The layout of the designed CNN.

Type	Configuration			
Input	220 × 70 × 1 rectified image			
Layer	Filter size	Kernel size	Stride size	Padding
Conv2D_1	64	3 × 3	1 × 1	1 × 1
Batch_norm_1	–	–	–	–
ReLU_1	–	–	–	–
Max_pool_1	–	2 × 2	2 × 2	0 × 0
Conv2D_2	128	3 × 3	1 × 1	1 × 1
Batch_norm_2	–	–	–	–
ReLU_2	–	–	–	–
Max_pool_2	–	2 × 2	2 × 2	0 × 0
Conv2D_3	256	3 × 3	1 × 1	1 × 1
ReLU_3	–	–	–	–
Conv2D_4	256	3 × 3	1 × 1	1 × 1
Batch_norm_4	–	–	–	–
ReLU_4	–	–	–	–
Max_pooling_3	–	2 × 2	2 × 2	0 × 0
Conv2D_5	512	3 × 3	1 × 1	1 × 1
ReLU_5	–	–	–	–
Conv2D_6	512	3 × 3	1 × 1	1 × 1
Batch_norm_5	–	–	–	–
ReLU_6	–	–	–	–
Max_pool_4	–	2 × 2	2 × 2	0 × 0
Dropout	Rate: 0.4			
Flatten	–	–	–	–
Dense	Units: 128, Activation: ReLU			
Dense	Units: 36, Activation: Softmax			

perform dimension comparison by accepting only rectangles that have width in a range of 0, (length of input)/(number of characters) and length in the range of (width of the input)/2, 4* (width of the input)/5. This process results in segmenting all the characters as binary images, as shown in Fig. 5(c).

Recognition of Segmented Characters: CNN, a trainable feature extractor, has recently achieved significant success in computer vision problems. The success of CNNs results from advancements in two technical areas: developing methods to prevent overfitting and creating more robust models [3, 17]. CNNs are formed of artificial neurons with self-optimizing properties, making them capable

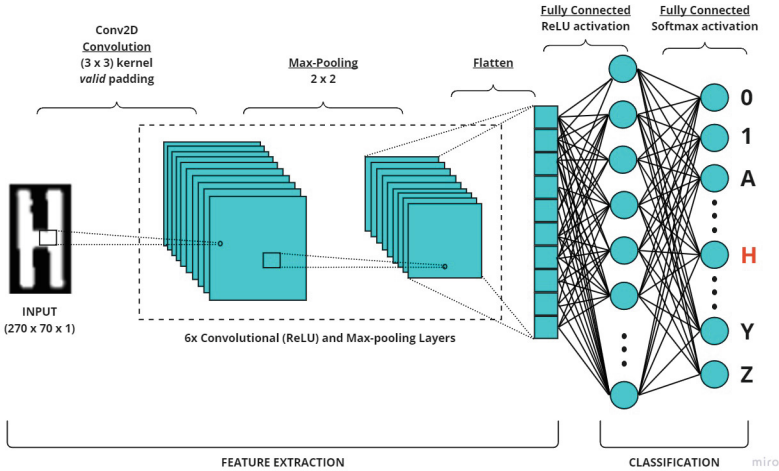


Fig. 6. Architecture of the proposed CNN

of extracting and classifying features from images more precisely than any other algorithm. Since the LP text consists of various font styles and sizes, we trained a more powerful deep network for this task. We want to give the model a more instinctive comprehension of the text or character. Among these fundamental characteristics lower-level text features like character labels and explicitly placed text pixels. We propose a Deep LPR CNN to accomplish this by training it on highly supervised text information at multiple levels, including segmentation of character regions, character labels, and text/non-text binary information. The additional supervised information provides the model with more specific textual features, enabling it to do tasks of high-level classification and low-level region segmentation. It allows our model to systematically recognize where and what the character is, which is crucial to make a reliable decision.

Table 2 and Fig. 6 display the detailed configuration and structure of the proposed CNN. The number of channels, stride, padding, and kernel sizes are similar to the VGGNET [26]. Other LP Recognition Tasks [12, 13] have successfully applied these configurations.

4 Results and Discussion

The proposed ALPR paradigm is verified for effectiveness; thus, Tensorflow and Keras frameworks have been utilized to implement the model. Our system configuration for evaluation is as follows: Intel core 9th Gen i7 CPU, NVIDIA GeForce GTX 1650Ti with 4 GB memory, and RAM of 16 GB.

4.1 Data Sets Description

As per our work, a general data set for distorted LP images is unavailable. The use of robust DL algorithms in the smart recognition of distorted LP is hampered



Fig. 7. Data set samples of distorted LPs.



Fig. 8. Data set samples of characters of various fonts.

by the absence of enough images. To effectively train our custom YOLOv4, we created a data set of vehicles with deformed LP in different shooting angles and complex backgrounds, as shown in Fig. 7. The images were collected from google images and natural scenes. We collected 3000 images of vehicles with various LP styles and annotated them to train the model.

We have a data set of 37,623 images to train our CNN model. The data set includes letters (A–Z) and numbers (0–9) with 50+ unique fonts that are commonly found on various LP, as shown in Fig. 8. To make the model resistant to various oblique views, data augmentation methods, including random rotation and perspective transformations, were used. Therefore, each class of alphabet or digit contains 1045 images of size 28×28 . We randomly select 33,861 character images for training and the remaining 3762 images for testing. Besides, Table 3 provides the comparative analysis of various data sets.

4.2 Result Analysis

The objective is to create a method that works well in several uncontrolled situations but simultaneously functions adequately in controlled ones (such as primarily frontal views). We have selected four online data sets: AOLP (RP), SSIG, and OpenALPR (EU and BR), which, as shown in Table 3, cover a wide range of scenarios. We have considered two variables: LP angles (frontal and oblique), as well as the separation between the vehicle and the camera (close

Table 3. Comparative analysis of various data sets.

Data sets	LP angle	Images	Vehicle Dist
AOLP (Road Patrol)	Frontal + oblique	611	Close view
SSIG (test set)	Frontal	804	Medium, distant
OpenALPR (BR)	Frontal	108	Close view
OpenALPR (EU)	Frontal	104	Close view
Proposed data set	Oblique	100	All views

Table 4. Performance analysis and comparison for multiple data sets.

Methods	AOLP (RP)	SSIG test	OpenALPR		Proposed data set
			EU	BR	
Proposed method (with no STN)	83.11%	82.01%	92.88%	89.71%	70.67%
Proposed method (with STN)	96.56%	89.55%	91.35%	92.69%	85.00%
OpenALPR (See footnote 1)	69.72%	87.44%	96.30%	85.96%	75.32%
Sighthound (See footnote 2)	83.47%	81.46%	83.33%	94.73%	50.98%
Severo et al. [11]	–	85.45%	–	–	–
Wang et al. [13]	88.38%	–	–	–	–
Shen et al. [12]	83.63%	–	–	–	–
G.S. Hsu et al. [6]	85.70%	–	–	–	–

view, intermediate view, distant view). Although these data sets cover various scenarios, a more general-purpose data set for challenging scenes is still a limitation. Thus as an additional contribution from our collected images, we have selected and manually annotated a set of 104 images that cover various challenging scenarios. The images contain substantial LP distortions but are still viewable to humans. A few images are shown in Fig. 7.

Experimental Results: This section expresses the experimental outcome analysis of the proposed ALPR mechanism and the comparison with other implemented methods. To testify to the overall performance of the presented model, we take the percentage of accurately identified LPs (CL) from the total number of testing LP images (TL). The recognition accuracy is given by

$$A = CL/TL \quad (8)$$

A point to note is that all the test data sets have been tested on the same network. No additional fine-tuning was performed to the network for a specific data set.

Table 5. mAP comparison of proposed YOLOv4.

Models	mAP
Proposed YoloV4	90%
YOLOv3 [22]	89%
YOLOv2 [24]	76.8%

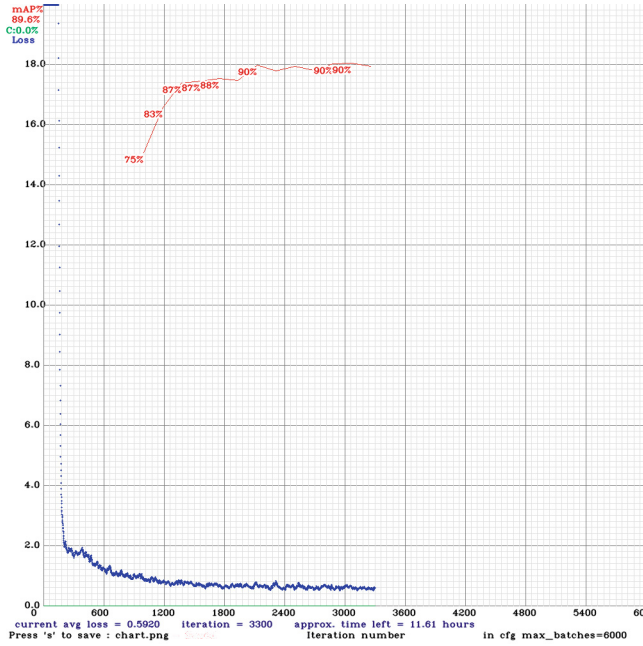
**Fig. 9.** Training performance of custom YOLOv4.

Table 4 indicates that the proposed method performs well with various data sets. Compared to other alternatives, it is superior on AOLP (RP) and SSIG Test data sets. The AOLP (RP) and SSIG Test data sets manifest the performance of 96.56% and 89.55% on the proposed method. The variation in performance on AOLP (RP) data set is approximately 27.0% for different approaches. Similarly, it is nearly 8.0% for the SSIG Test data set. Also, the error rate reduction due to the proposed method is 88.63% and 79.19%, respectively, compared to OpenALPR and Sighthound. Table 4 shows the comparison with other implemented systems. Our system has achieved recognition rates comparable to commercially available systems representing controlled scenes, where the LPs have frontal views and less complicated environments. Our system has achieved the best performance in AOLP RP and the proposed oblique LPs data sets.

Furthermore, the proposed ALPR method performance on the OpenALPR data set is inferior compared to other alternatives. The proposed ALPR app-

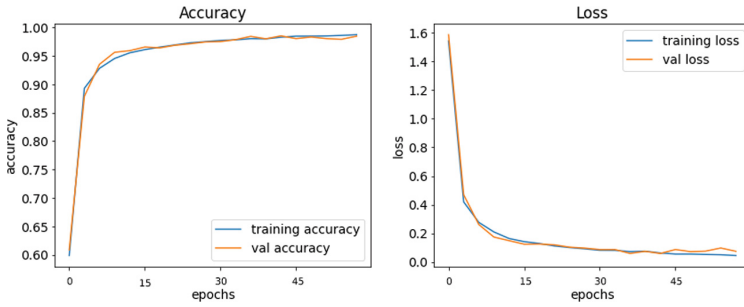


Fig. 10. Training accuracy and loss analysis of proposed model.

roach attains more than 90.0% performance but less than 4.95% and 2.04%, respectively, compared to OpenALPR and Sighthound methods. In addition, the proposed method, OpenALPR, and Sighthound approaches vary by 7.01%, 26.58%, and 13.27%, respectively, on AOLP (RP), SSIG Test, and OpenALPR data sets. It indicates the stability of the proposed mechanism in comparison to other alternatives.

Moreover, the proposed system presents superior outcomes than other mechanisms on proposed data sets. The performance of 85.0% is attained for the proposed data set, and it is better than 10.0% and 35.0%, respectively, compared to OpenALPR and Sighthound. Besides, it is essential to note that STN has a beneficial impact on identification outcomes. We remove the STN module from the proposed mechanism to demonstrate the effect. The recognition performance in oblique scenes of AOLP and the presented data sets have a significant gap, as seen in Table 4. This performance difference demonstrates how STN contributes to improved performance in identifying distorted LP.

Table 5 and Figure 9 illustrates the training performance of the custom Yolov4 model. The model achieved 90.0% mAP with 2800 iterations which outperformed the Yolov2 and Yolov3 used in [22, 24]. Also, Fig. 10 indicates that the model is not overfitted on given input data. The continuous decrease in error trend is observed for the proposed model. Besides, character recognition performance is analyzed by a confusion matrix, and it is illustrated in Fig. 10. It is observed that the presented APLR method misclassified the ‘O’ and ‘0’.

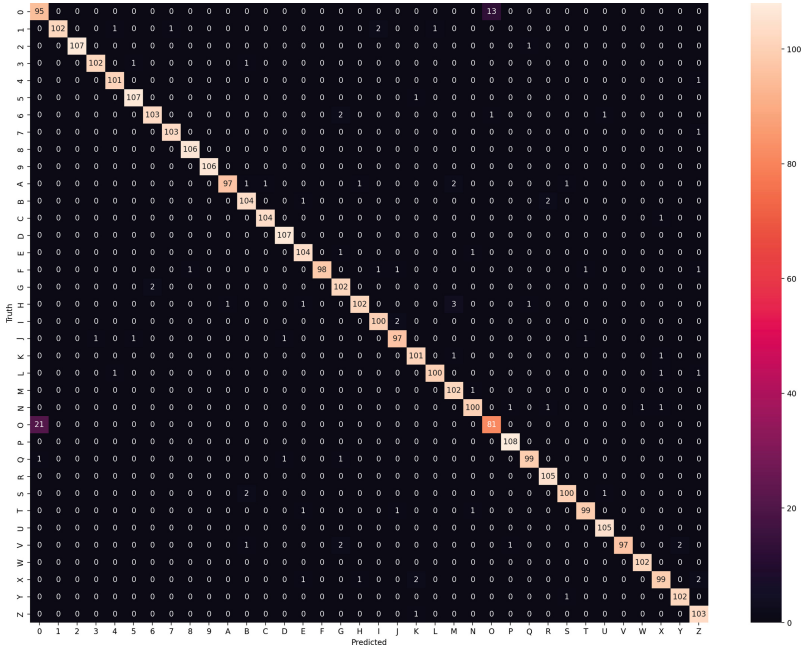


Fig. 11. Confusion matrix of the character recognition model per class.

5 Conclusion

This work demonstrated a comprehensive approach for ALPR in uncontrolled environments. Results indicate that the presented ALPR paradigm performs significantly better than the existing methods in challenging data sets with License Plates captured at severely oblique viewpoints. The use of the spatial transformer network, which aids in rectifying the distorted license plates, is the primary contribution of this work. This step helps the Recognition Network (Convolutional Neural Network) to understand the character patterns in a simplified way because it has to deal with far minimal distortion. Besides, we generated a complex data set by augmenting the images to detect license plates in skewed views. Currently, the system proposed can recognize the license plate number in English. For future work, we intend to enhance the current paradigm to recognize multilingual license plates written in the Devanagari language.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv preprint [arXiv:2004.10934](https://arxiv.org/abs/2004.10934) (2020)
2. Bulan, O., Kozitsky, V., Ramesh, P., Shreve, M.: Segmentation-and annotation-free license plate recognition with deep localization and failure identification. *IEEE Trans. Intell. Transp. Syst.* **18**(9), 2351–2363 (2017)

3. Dhillon, A., Verma, G.K.: Convolutional neural network: a review of models, methodologies and applications to object detection. *Progr. Artif. Intell.* **9**(2), 85–112 (2020)
4. Gonçalves, G.R., da Silva, S.P.G., Menotti, D., Schwartz, W.R.: Benchmark for license plate character segmentation. *J. Electron. Imaging* **25**(5), 053034 (2016)
5. Gonzalez, R.C.: *Digital Image Processing*. Pearson Education India (2009)
6. Hsu, G.S., Ambikapathi, A., Chung, S.L., Su, C.P.: Robust license plate detection in the wild. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. IEEE (2017)
7. Hsu, G.S., Chen, J.C., Chung, Y.Z.: Application-oriented license plate recognition. *IEEE Trans. Veh. Technol.* **62**(2), 552–561 (2012)
8. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. *Adv. Neural Inf. Process. Syst.* **28** (2015)
9. Kaur, P., Kumar, Y., Gupta, S.: Artificial intelligence techniques for the recognition of multi-plate multi-vehicle tracking systems: a systematic review. *Arch. Comput. Methods Eng.* **29**, 4897–4914 (2022)
10. Kurpiel, F.D., Minetto, R., Nassu, B.T.: Convolutional neural networks for license plate detection in images. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 3395–3399. IEEE (2017)
11. Laroca, R., et al.: A robust real-time automatic license plate recognition based on the yolo detector. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–10. IEEE (2018)
12. Li, H., Wang, P., Shen, C.: Towards end-to-end car license plates detection and recognition with deep neural networks. *CoRR* abs/1709.08828 (2017)
13. Li, H., Shen, C.: Reading car license plates using deep convolutional neural networks and lstms. *arXiv preprint [arXiv:1601.05610](https://arxiv.org/abs/1601.05610)* (2016)
14. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
15. Montazzoli, S., Jung, C.: Real-time Brazilian license plate detection and recognition using deep convolutional neural networks. In: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 55–62. IEEE (2017)
16. Ravirathinam, P., Patawari, A.: Automatic license plate recognition for Indian roads using faster-RCNN. In: 2019 11th International Conference on Advanced Computing (ICoAC), pp. 275–281. IEEE (2019)
17. Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: a comprehensive review. *Neural Comput.* **29**(9), 2352–2449 (2017)
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788 (2016)
19. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271 (2017)
20. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. *arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)* (2018)
21. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **28** (2015)
22. Sahu, C.K., Pattnayak, S.B., Behera, S., Mohanty, M.R.: A comparative analysis of deep learning approach for automatic number plate recognition. In: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 932–937. IEEE (2020)

23. Saidani, T., Touati, Y.E.: A vehicle plate recognition system based on deep learning algorithms. *Multimed. Tools Appl.* **80**(30), 36237–36248 (2021). <https://doi.org/10.1007/s11042-021-11233-z>
24. Silva, S.M., Jung, C.R.: License plate detection and recognition in unconstrained scenarios. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 580–596 (2018)
25. Silva, S.M., Jung, C.R.: Real-time license plate detection and recognition using deep convolutional neural networks. *J. Vis. Commun. Image Represent.* **71**, 102773 (2020)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
27. Xie, L., Ahmad, T., Jin, L., Liu, Y., Zhang, S.: A new CNN-based method for multi-directional car license plate detection. *IEEE Trans. Intell. Transp. Syst.* **19**(2), 507–517 (2018)