Recognition of Indian License Plate Number from Live Stream Videos

Sachin Prabhu B , Subramaniam Kalambur, Dinkar Sitaram
Department of Computer Science
PES University
Bengaluru, India
sachin.prabhu.b@gmail.com, {subramaniamkv, dinkars}@pes.edu

***Abstract*— Automatic License Plate Recognition (ALPR) has been employed in many developed countries for traffic management, automatic speed control, tracking stolen cars and also in automatic toll systems for improving the traffic control. ALPR is a surveillance system that extracts the information from the vehicle license plate by capturing the images. Human intervention to recognize the license plates results in delays and cases of missing out certain violations. Automating the process from surveillance cameras is challenging as the system must deal with different angles, distance, resolution and illumination conditions. In this paper, we compare the accuracy of three well known approaches to detect license plates under Indian conditions. We have selected OpenALPR, *k*-NN and Convolutional Neural Networks (CNN) based techniques, adapted them for local conditions, compared their accuracy on both still images and live stream videos. Our analysis demonstrates that the CNN based approach is the most suitable for recognizing number plates in India.**

Keywords—CCTV, OpenALPR, k-NN, CNN, License Plate Recognition, Character recognition.

I. INTRODUCTION

Over past three decades with increase in number of vehicles and road networks the need for monitoring traffic is considered as significant role for efficient utilization of road networks. Closed Circuit Television Camera (CCTV) cameras, traffic enforcement camera (red light camera, speed camera) are used for monitoring traffic which increases efficiency compared to human monitoring. Depending on the camera its mounted beside the road or on instilled in an enforcement vehicle (like police cars) to detect traffic regulation violations which includes going through red light traffic signal, speeding, unauthorized lane change, for recording vehicles within a congestion charge area. These monitoring methods can be linked with automated ticketing system.

Automatic License Plate Recognition (ALPR), Red Light Camera are used in Russia, London, UK, Australia. But considering the Indian scenario it's only restricted to CCTV cameras as of now. The problem with CCTV cameras are low resolution which results in poor footage. Lack of standardization of the number plates in terms of fonts and orientation presents additional challenges.

Localization of license plate area or region from captured images or frames is a challenging task. The reason being, huge variations in size, shape, colour, texture and orientation of license plate region within images (spatial orientation). Objective of any ALPR system is to localize potential license plate area within an image captured through camera mounted at road-side and interpret them with Optical Character Recognition(OCR) system to get license plate number of vehicle.

License plates are strictly maintained in most of developed countries. A particular template for license plate is followed and generalized where it is mandatory to follow the same format for every vehicle. It includes dimension of license plate. The attributes to consider standardization are color, size, of the plate and font. Distance between two characters is also an attribute. In India often License plates with regional fonts are also used which adds to complexity of detecting license plate as system needs to be trained for each region. Figure 1 illustrates the difference between standardized number plates used in Europe against the variations seen in the Indian context.

Fig. 1. License plate images of European and Indian vehicles. European license plates are statndardized while Indian license plates are not.

The character schema is as follows, first two letters represent the state, next two digits is corresponding state code for district within particular range followed by two letters and four digits. Thus, each vehicle has unique number represented by license plate characters. Weather conditions, vehicle movement, damage caused to license plate, movement of vehicle, lighting condition, the location of license plate within an image are some of negative factors which influence on the license plate recognition system.

In this paper, we compare the ability of various techniques to effectively function in presence of such variations. The paper is organized as follows - Section II outlines various previous approaches to solve the problem. In section III, we describe the collection of our dataset. Section IV outlines our methodology and the adaptations that we have performed to the various algorithms. In Section V, we show the comparative results of our study and finally conclude in Section VI.

II. RELATED WORK

Previous work in the area of license plate recognition require solving the twin problems of *localization and recognition.* The generic approaches taken to solve these problems is to either use a multimedia based approach or a machine learning based approach–

In this section, we present a brief literature review describing ALPR work. Badr et al. [1] proposed an innovative approach of Automatic License Plate Recognition System (ALPR) using Morphological operations, Histogram manipulation and Edge detection techniques for plate localization and characters segmentation. Artificial Neural Networks are used for character classification and recognition. i.e it makes use of both multimedia based and machine learning technique for localization and recognition of license plate respectively. It uses Hough Transform for de-skewing license plate since the license plate may be skewed because of the angle of the camera while image acquisition process. Benefit is, it can detect license plate for clear image while disadvantage is it has worse recognition rate for blurry and skewed snapshots

License Plate Recognition can be implemented with help of Raspberry Pi. Sundararaman et al. [2] proposed approach of ALPR using Raspberry Pi. It makes use of multimedia based approach for both localization and recognition of license plate. This introduces a Completely Automated License Plate Recognition system using the Optical Character Recognition (OCR) to read the information on the images of license plate of vehicles captured by the camera on real time basis. The image of the Vehicles License plate is captured and is processed by the segmentation of the characters and is verified by the Raspberry Pi Processor for the authentication purpose. Even though it results in 96 % accuracy the processing takes time on Raspberry Pi. It is difficult to implement on large scale because of processing capability of Raspberry Pi.

License Plate Recognition can be implemented with help of Raspberry Pi. MATLAB can be used to implement license plate recognition system. It makes use of multimedia approach for both localization and recognition of license plate. Tomar et. al. [3] proposed a system which captures the image of the license plate of a vehicle using raspberry pi camera and the details are being retrieved using the character segmentation which is done by optical character algorithm. Choosing of an embedded platform leads to the automation in the field of electronics. The objective is to represent a system by using license plate of a vehicle for various application such as inventory control, border checkpoints, highly restricted area (supreme court, military base), etc. The system is executed on Raspberry Pi and imitated in MATLAB because the experimental result shows that the system is fast enough in capturing images, recognition of algorithm and data streaming but disadvantage is it cannot be implemented on large scale because of processing power of Raspberry Pi.

Quadcopters or drones can be used to capture license plate of vehicle at real time. Garcia et.al [4] proposed Autonomous Indoor Navigation of a Quad Copter which is high speed vision based. This paper considers low cost Micro Air Vehicle. The approach relies on analysis and extraction of contours formed due to vision surrounded by environment which successfully enables MAV to steer in hallways and also the intersections. The benefit is the drone can navigate itself based on image while disadvantage is flight time of drone varies from 15 to 30 minutes.

The methods used in processing the image/data plays major role. Sarbjit and Sukhvir [5] proposed an efficient method to detect License plate considering Indian scenario. This makes use of multimedia approach for localization of license plate. This paper presents a simple and efficient method for the recognition of license plate from the vehicle image based on operations like morphological, connected component analysis, thresholding and sobel edge detection. The benefit is it can locate license plate at faster rate but fails to identify for license plate at an angle.

Character Recognition plays significant role in license plate recognition. Wawage and Oza [6] proposed recognition system to detect license plate using classification algorithm. This paper deals with computing techniques from the field of AI, computer vision, and neural networks. It deals with identification and character recognition for system that is automated. It makes use of multimedia approach for localization and recognition of license plate. The benefit is system works for wide variety of illumination condition but has certain restrictions based on clarity of number plate image, skew in image and speed of the vehicle.

Neural Networks can be used for character recognition. Kodabagi and Kanavi [7] proposed a method for LPR. This approach makes use of multimedia and machine learning approach for localization and recognition of license plate respectively. The proposed method for recognizing licence plates, system works in three modules-localization of license plate, character segmentation and recognition of the characters from the license plate. Localization of the license plate is done using horizontal & vertical edge processing which is part of morphological operations. Character segmentation is carried out using connected component labelling. Neural network classifier is used for character recognition. The benefit is it works well for license plate with special orientation but must be improved for illumination and rotation condition.

License plate comprises of multiple digits or alphabets. It is important to recognize each digit or alphabet. Recognizing arbitrary multi-character from street images is complex. Ian J. et.al [8] proposed a method to Recognize Multiple Digits from Street View panoramas. Deep Convolutional Neural Network has been implemented for same. Neural Network directly operates on image pixels involves unified approach that integrates localization, segmentation and recognition steps, while traditional approach separates these steps. For training a framework called DistBelief [22] is employed to implement large scale deep neural networks.

The network architecture implemented in this paper involves convolutional and maxpool operation. It is based on the paper proposed by Stark et.al. [9] to recognize CAPTCHA with Active Deep Learning that makes use of ability to gain training data without human intervention. Also, they demonstrate a mechanism to re-train the network and present results on an autogenerated CAPTCHA dataset. This approach improves the performance of the network if the initial training data is labelled. Similarly, for training purpose in this paper training data of license plate images are labelled.

CNN's can be fine-tuned to work under different conditions i.e. robust under variations in lighting, occlusion etc. Ortiz et.al [10] proposed an approach to detect license plates and recognition using deeply learned convolutional neural network for localization and recognition of license plate. In this approach, the CNN's are trained and fine-tuned. It can work across different variety of license plate templates (e.g. Background, font, size. The system is compared with OpenALPR and gives better results considerably with accuracy of 90%.

III. COLLECTION OF DATASET

The dataset involves two aspects. Still images of Indian license plate vehicles and video of vehicles captured at traffic junction. Dataset consists of 200 still images and 250 frames are considered in case of video. Images of only cars are considered since license plate dimensions of car is considered for training. Video of resolution 1920 X 1080 i.e. Full HD is considered with 30 frames per second (fps) In case of video each second leads to 30 frames out of which frames in which vehicles are present are considered. In case of still images 3120 x 4160 are considered captured from 13MP mobile camera. It is rescaled to 576 x 768 to reduce computation time while quality of image remains medium measured in terms of DPI depth per inch (changes from 72 to 120). A video of 1 minute 37 seconds at 30 fps is considered. It results in 3249 frames out of which 250 images are selected which contains vehicles are considered for calculating accuracy. The following assumptions are made considering dataset.

- For capturing license plate at traffic junction video is captured with mobile (1920 x1088) with frame rate of 30fps
- Camera is mounted at 5ft height.
- Only License plate of car is concentrated.
- The video is in MP4 or H.264 format

IV. METHODOLOGY

In this section, we shall look at components of the system and then discuss alternative algorithms. The high-level block diagram is as illustrated in Figure 2 followed by constraints and dependencies.

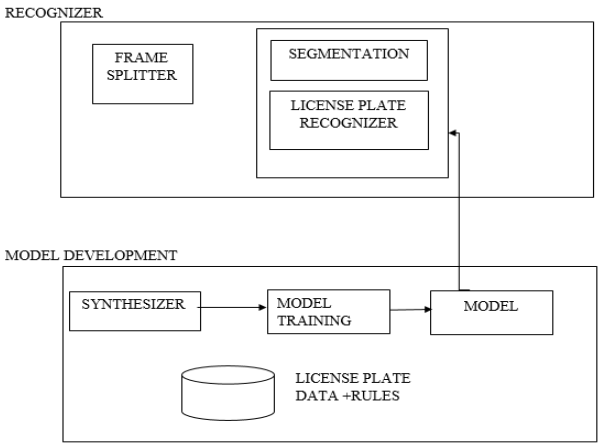

Fig.2. High-level block diagram

As shown in Figure 2, it has two main components, Recognizer and Model Development. The given video is spilt into frames for recognition. License plates are synthesized and subjected to training. The License plate within given frame is recognized with help of trained model. It involves character recognition within License plate.

A. Constraints and Dependency

- The license plate must be clearly visible in video (resolution considered 1920*1080)
- Dependent on height, distance, illumination condition resolution and angle from which video is captured.
- Dependent on weights generated due to neural network by synthesized images used for training along with various fonts.

The system architecture is as shown in Figure 3 and we shall compare the methods based on approach used for localization and recognition of license plate i.e. either multimedia or machine learning approach as shown Table I.

TABLE I. DIFFERENT APPROACHES USED

Method name	**Localization of license plate**	**Recognition of license plate**
k-NN (LPRI)	Multimedia	Machine learning
OpenALPRI	Multimedia	Multimedia
CNN	Machine learning	Machine learning

We made configuration changes pertaining to Indian conditions i.e consider Indian license plates each of the three methods. The implementation details carried out for Indian license plates are summarized in Table II.

TABLE II. IMPLEMENTATION DETAILS IN BRIEF

Method name	Additional Implementation
k-NN (LPRI)	Trained with different font styles and tested for Indian license plate images.
OpenALPRI	Configured patterns to recognize Indian license plate
CNN	Synthesized Indian license plate images for training purpouse, Trained the neural network to recognize characters

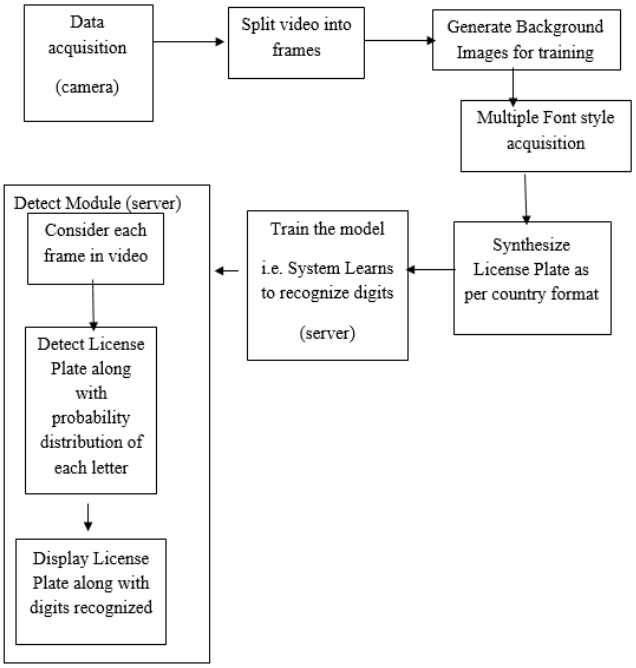

Fig. 3 Architecture diagram

B. License Plate recognition with k-NN (LPRI)

In this approach, the Character Recognition is performed using the *k*-Nearest Neighbor Algorithm, training and testing model proposed by Dahms [17]. We trained the model with different fonts to match Indian license plate font. Video acquired was split into frames and stored in file system considered as test data. For the remainder of this paper, we will refer to this method as LPRI (License Plate Recognition - India).

The training model involves following steps. First step is to classify the various font images and map them according to character. These are training samples i.e. classification and images, where classification indicates set of numbers and image is image corresponding to particular digit is to be used for k-NN classification. Training samples are from 5 different fonts. These files are subjected to training by passing on to *KNearest.train()* function provided by OpenCV. The k nearest samples are found using *KNearest.find_nearest()* function provided by OpenCV used for recognizing characters. *k* is chosen as odd to avoid tie. *k* is set as the minimum number of samples available for any of the numbers. So, training data is generated as mentioned earlier, next step is to generate test data. Consider a video which is sequence of frames. Each frame is read and stored in file system subjected for license plate recognition. It has 2 main steps.

1. Detect License plates in given Image
2. Detect Characters in License plate

1. Detect License Plate in an Image

Given an image first step is to find license plates within an image. It involves following steps

- Consider given image. The image is a frame read which is part of video captured by camera.
- Subject it to preprocessing – Gaussian Blur, Adaptive Threshold results in 2 images with gray scale and threshold image. It results in edge detection and Noise removal.
- Find Possible Characters in the above resulted image. This function first finds all contours, then only includes contours that could be characters (without comparison to other characters yet)
- Find List of Matching Characters. Given a list of all possible characters, find groups of matching characters in the next steps each group of matching characters will attempt to be recognized as a plate. For each group of matching characters attempt to extract plate, if plate was found add to list of possible plates.
- Sort characters from left to right based on x-axis position. Calculate the center point of the plate. calculate plate width and height. calculate correction angle of plate region. Pack plate region center point, width and height, and correction angle into rotated rectangle member variable of plate. perform the actual rotation. Perform the actual rotation. unpack original image width and height. rotate the entire image. Copy the cropped plate image into the applicable member variable of the possible plate Extract Plate from the given Image which results in List of Possible Plates. Display with contours i.e. red rectangle around the plates

Thus, it results in possible number of license plates found within an image or frame.

2. Detect Characters within License Plate

Once license plate is extracted, next step is to find the characters in the plate. It follows similar approach as shown in following steps.

- List of Possible Plates got above is subjected to Preprocessing I.e. Convert it to Grayscale and Threshold Image.
- Check for possible characters in the given plate. Does a rough check on contour to see if it could be a character.

- List Possible Characters in Plate. This function first finds all contours, then only includes contours that could be characters (without comparison to other characters yet)
- List Matching Characters. Remove Overlapping Characters E.g.: for the letter 'O' both the inner ring and the outer ring may be found as contours, but character should be only included once
- Check for Longest List of Matching Characters. The purpose of this function is to re-arrange the one big list of characters into a list of lists of matching characters.
- Load k-NN data i.e. text files generated earlier
- Recognize characters in plate. The data must be loaded to K-NN training before the characters are recognize method is called or else it won't be able to recognize.
- List Possible Plates with most characters found.
- Display License plate found in the image with maximum characters recognized.

Thus, it results in recognition of license plate along with characters in an image.

C. License Plate recognition with OpenALPR (OpenALPRI)

OpenALPR [21] is an open source *Automatic License Plate Recognition* library. It works on principle of Local Binary Pattern. It has been trained to locate license plate of US, Europe etc. OpenALPR is trained to recognize vehicle depending on country using the pattern of license plate. Our custom implementation of OpenALPR is referred to as OpenALPRI in the remainder of this paper. We trained the model for India by configuring patterns of license plate. For Indian License plates, it's of 10 characters. i.e. for example Karnataka it would be KA 02 MG 3890. So, the pattern file present in *runtime data/postprocess/* folder of alpr is configured for India as KA ## @@ #### where # refers to digits, @ refers to alphabets. Next step is to load various fonts and provide width and height of license plate. Indian and European License plates almost have same font considering. It can be trained for various fonts as referred in OpenALPR documentation. Thus, for recognizing Indian License Plates, pattern has been configured as mentioned and European license plate pre-trained OCR results has been referred.

D. License Plate recognition with CNN

This approach has training and testing model as proposed by Earl [12]. We trained the model to match Indian license plate fonts [18,19]. The video acquired is split into frames is stored in file system is used as test data. Input Image resolution is chosen as 128*64 for training images. To locate the license plate with image of large size a sliding window approach is harnessed. Considering the window, the neural network should produce following output

- Locate License plate (the probability of plate within input image)
- Recognize characters (the probability of the digit in each position). Considering Indian License Plates for each 10-possible position it should return probability distribution across 36 possible characters.

The license plate within image is considered if it follows following rules.

- The license plate is within image bounds.
- The plate's height is less than 87.5% of the image's height and the plate's width is less than 80% of the image's width.
- The plate's height is greater than 60% of the image's height and the width is greater than 60% of the image's width.

The sliding window moves 8 pixels at a time. i.e. Stride = 8.

To train the neural net a training data must be provided along with correct outputs. In this case there will be set of images of size 128*64 along with expected output. The SUN database [16] is considered for background images over which plates with fonts are overlaid. The database contains about 100,000 images. The network must identify bound of plates.

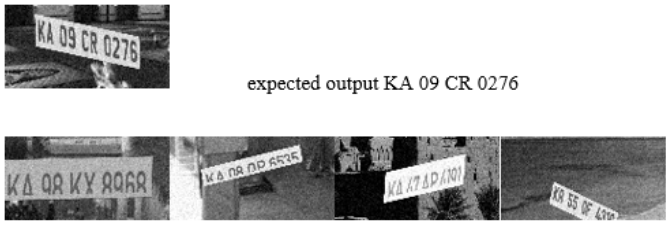

Fig. 4 Synthesized Indian License Plates

Figure 4 illustrates the synthesized images generated for Indian License Plates with 10 characters. Each expected output contains two parts. First part is number of the plate or neural net should output. The second part indicates the license plate to be present. It is either 1 or 0. If it's considered it is said to be 1 or else 0. The synthesized images don't follow the rules according to real license plate like state code, it randomly selects one particular font among many fonts given for training and generates license plate.

The process of generating training images is as shown in Figure 5

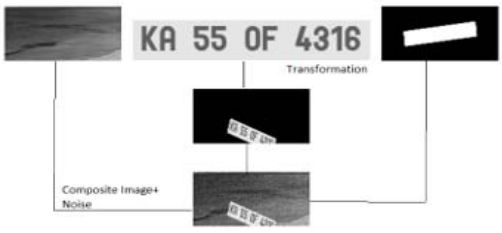

Fig .5 Process of generation of synthesized Indian License Plates

To simulate real world lighting variation and license plate data the text must be darker than the license plate. The color of the license plate is chosen randomly. Noise is added as in real world the camera will produce sensor noise or some images will be out of focus.

To generate synthetic images of Indian License Plates the background images have been referred to SUN database and license plate fonts. The generation of synthetic license plates follows the rules of an Indian License Plate. i.e., 10 characters format. The steps for generation of synthetic images are as follows.

- Load random background image of size 64 x 128 from SUN database
- Generate License plate image in Indian format by loading random fonts for each plate.
- Make affine transformation of license plate considering background image so that license plate is at an angle
- License plate on composition with background image is generated.
- Noise is added to resemble original picture from camera.

License plate images of required number can be synthesized for training purpose. For training purpose one thousand Indian license plates are generated. The training model harness convolutional neural networks.

The convolutional neural network has 5 layers. It has convolutional and maxpool layers alternatively resulting in fully connected layer. The model is subjected for training. The trained model generates weights. These weights along with image is passed to detect model which is similar to detect model with 5 layers while last 2 layers are convolutional instead of fully connected. The image of various size can be given for detection and adjacent windows result in many convolutional features as it yields presence of license plate along with probability of each character at each pixel. The bounding boxes are drawn for each character which later on subjected to non-maximum suppression resulting in license plate with character recognition. We trained CNN for Indian License plate and tested successfully on Indian vehicles using frames from video acquired.

V. RESULTS

In this section, we shall analyze result for each approach considering still images of 200 and 250 frames in case of video for Indian license plates. Still images of parked Indian vehicles are considered while frames belong to video acquired at traffic signal junction. The results involve two aspects i.e localization of license plate and character recognition. It can be inferred that CNN achieves result of 90% compared to OpenALPRI and CNN which is around 40% and 60% respectively. The reason for this is the dimensions of license plate is assigned prior during training in case of OpenALPRI and CNN and it checks for same in the image while in case of LPRI it is calculated based on the image after preprocessing. In case of CNN size of sliding window matters. If CNN is fine-tuned it can achieve better results. Fine tuning is to enhance size of sliding window and hidden layer number of neural network. Even the angle at which license plate is aligned matters. The lighting conditions also affect the detection of license plate. In case of OpenALPR, it must be trained with different dimensions of license plate. But problem again is the dimensions are not standardized in case of India. Figure 6 illustrates the localization of license plate in percentage

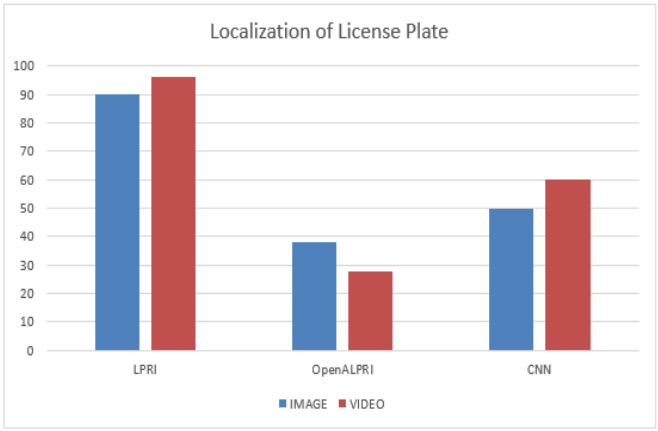

Fig 6. Localization of License Plate

The accuracy of each approach is measured in terms of characters recognized. LPRI has minimum recognition rate of around 10% while OpenALPR has about 30% and CNN has maximum of 50%. The reason for this is neural network has better learning rate which is trained with different fonts along with different angles of license plate with sliding window approach since it has been trained with huge quantity of training dataset i.e of 1000 images while in case of LPRI and OpenALPR the model fails to recognize the font of license plate. In case of CNN license plates are synthesized to match with original license plate while it's not in case of others. The accuracy of recognition of characters is as shown in Figure 7.

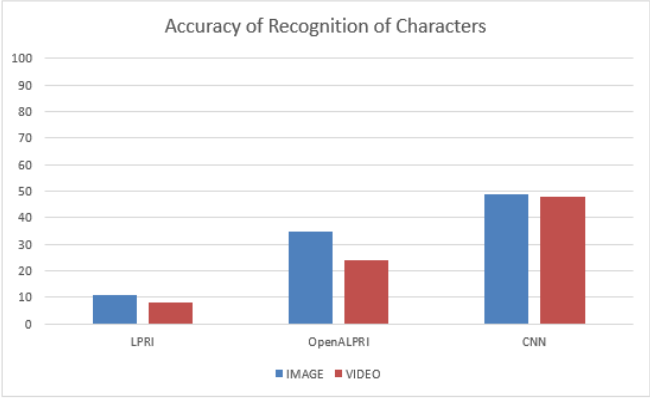

Fig 7. Accuracy of Recognition of characters

TABLE III. TIME FOR LICENSE PLATE RECOGNITION

Method name	Time take for recognition per frame
k-NN (LPRI)	600 milliseconds
OpenALPRI	1 second
CNN	30 – 70 seconds

The time taken for recognition of license plate for each method is as shown in Table III. CNN takes more time because of sliding window approach but gives accurate results compared to LPRI and OpenALPRI.

VI. CONCLUSIONS

The CNN approach is scalable and can be implemented efficiently compared to other two approaches i.e *k*-NN (LPRI) and OpenALPR for Indian License Plates i.e OpenALPRI. LPRI is able to recognize most of license plate within an image considering still image or the frame in case of video compared to other OpenALPRI and CNN. But CNN and OpenALPRI has better recognition of characters compared to LPRI. *k*-NN approach needs to be improved by training with different sample character sets. Indian License plates are not standardized and this is a drawback considering present scenario as the system needs to be trained for multiple fonts. CNN approach recognizes particular fonts and is slow considering the sliding window approach for image with high resolution. GPU power can be harnessed to overcome this as processing is done on server. CNN approach has maximum accuracy and is more adaptable, also it recognizes license plates from CCTV footage with Full HD quality. Thus, license plates can be recognized from still image or live streaming video.

REFERENCES

[1] Badr, Amr, Mohamed Mahmoud Abdelwahab, Ahmed Mohmad Thabet, and Ahmed Mohamed Abdelsadek. "Automatic license plate recognition system." Annals of the University of Craiova Mathematics and Computer Science Series 38, no.1 (2011):62-71.

[2] Sundararaman, Vijayaraghavan, T. G. Vijayalakshmi, G. V. Swathi, and Sambit Mohapatra. "Automatic License Plate Recognition System Using Raspberry Pi." In Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing, pp. 217-222. Springer India, 2016.

[3] Khushboo Chhikara, Dr.Pankaj Tomar – " A Smart Technique for Accurate Identification of Vehicle License plate Using MATLAB and Raspberry Pi 2", www.ijetmas.com May 2016, Volume 4, Issue 5, ISSN 2349-4476

[4] Garcia, Adriano, Edward Mattison, and Kanad Ghose. "High-speed vision-based autonomous indoor navigation of a quadcopter." In Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, pp. 338-347. IEEE, 2015.

[5] Sarbjit Kaur Sukhvir Kaur – "An Efficient Method of License plate Extraction from Indian Vehicles Image" International Journal of Computer Applications (0975 – 8887) Volume 88 – No.4, February 2014

[6] Wawage, Pawan, and Shraddha Oza. "Design Approach for Vehicle License Plate Automatic Detection and Character Recognition System Using Classification Algorithm." International Journal of Emerging Technology and Advanced Engineering 2, no. 10 (2011): 132-135.

[7] Kodabagi, M. M., and Vijayamahantesh S. Kanavi. "License Plate Recognition System For Indian Vehicles." International Journal of advance research in Engineering an technology (IJARET) 2 (2013).

[8] Goodfellow, Ian J., Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. "Multi-digit number recognition from street view imagery using deep convolutional neural networks." arXiv preprint arXiv:1312.6082 (2013).

[9] Fabian Stark, Caner Haz_rba_s, Rudolph Triebel, and Daniel Cremers "CAPTCHA Recognition with Active Deep Learning."

[10] Masood, Syed Zain, Guang Shu, Afshin Dehghan, and Enrique G. Ortiz. "License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks." arXiv preprint arXiv:1703.07330 (2017).

[11] OpenCV website http://opencv.org/

[12] Recognition of License plate with Tensorflow https://matthewearl.github.io/2016/05/06/cnn-anpr/

[13] OpenALPR documentation http://doc.openalpr.com/opensource.html

[14] OCR hand written Data using K-NN (OpenCV documentation)

[15] OpenALPR documentation for adding license plate http://doc.openalpr.com/opensource.html#adding-a-new-country

[16] SUN database http://groups.csail.mit.edu/vision/SUN/

[17] Chris Dahms, Recognition of License plate,2016, Github repository, https://github.com/MicrocontrollersAndMore/OpenCV_3_License_Plate_Recognition_Python https://github.com/MicrocontrollersAndMore/OpenCV_3_KNN_Character_Recognition_Python

[18] License plate fonts http://www.fontspace.com/category/license%20plate

[19] License plate fonts http://www.dafont.com/uk-number-plate.font http://www.1001fonts.com/license-plates-fonts.html

[20] Brazil vehicle dataset – CCTV footage - DAINF - Departamento Acadêmico de Informática http://www.dainf.ct.utfpr.edu.br/~rminetto/projects/vehicle-speed/dataset/?C=S;O=A http://www.dainf.ct.utfpr.edu.br/~rminetto/projects/license-plate/

[21] OpenALPR documentation – Camera calibration http://doc.openalpr.com/accuracy_improvements.html#training-ocr

[22] Dean, Jeffrey, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior et al. "Large scale distributed deep networks." In Advances in neural information processing systems, pp. 1223-1231. 2012.

[23] C.Wolf and J. Jolion. "Extraction and recognition of artificial text in multimedia documents." Pattern Anal. Applic., 6:309–326, 2003.

[24] Sauvola, Jaakko, and Matti Pietikäinen. "Adaptive document image binarization." Pattern recognition 33, no. 2 (2000): 225-236.

[25] Guo, Zhenhua, Lei Zhang, and David Zhang. "A completed modeling of local binary pattern operator for texture classification." IEEE Transactions on Image Processing 19.6 (2010): 1657-1663.