Journal of Engineering Sciences
Vol 11, Issue 7,July/2020
ISSN NO:0377-9254

Automated Vehicle Security System Using ALPR and Face Detection

Himanshu Chaudhary[1], Gaurav Singh[2] , Ms. Pratibha Singh [3]
[1]Computer Science & Engineering Department, ABES Engineering College, India
[2]Computer Science & Engineering Department, ABES Engineering College, India
[3]Computer Science & Engineering Department, ABES Engineering College, India

***Abstract*— As the infrastructure in this fast moving world is increasing day by day and with the increase of infrastructure, the concern for security systems such as enhancing the vehicles security at various places like societies, parkings etc. increases. Vehicle theft is one of the major issues regarding the security system. So, security from vehicle theft can be achieved by monitoring the vehicles with Automated Vehicle Security System Using ALPR and Face Detection model. Since the vehicles are increasing in numbers, so it is necessary to develop automation in its recognition and detection procedure.**

The main focus of this work is to build a real time model that acknowledges the registration plate number of each motor vehicle and also the driver's face through surveillance cameras and store this in the database while incoming and then verifying that the car is taken out by the car owner or the right person at the exit of the parking zone. Besides its security application, ALPR system can also be applied towards traffic law enforcement, congestion pricing, and traffic control management for recognizing vehicles that commit traffic violations or for automatic toll tax collection.

***Keywords*— Automatic License Plate Recognition, Detection, Segmentation, Convolution Neural Network, Bi-Directional Long short term memory, Denser Cluster Based Voting, Automated Vehicle Security, CSV(Comma Separated Values), SVM(Support Vector Machine)**

1. INTRODUCTION

In this work our aim is to include a real time model that identifies registration plates of motor vehicles at public places, public parking, etc via security camcorder and also the driver faces. The characters fetched from a license plate can then be used to keep a record of vehicles outgoing and incoming of the societies, parking and public places which solves the security problem to an excellent extent. This system can also be utilized for different aspects by integrating it with differing types of datawares.

The use of vehicles in our current lives is increasing exponentially due to rapid economic development. Managing parking of large crowds is very complicated and insecure. Sometimes it leads to vehicle theft from the parking anonymously. Manual parking systems failed to take care of who parked the vehicle in parking and who had taken it out. This paper helps in dealing with the problems of vehicle theft at public places parking.

2. RELATED PREVIOUS WORK

In this work[1], the author explores the sum of Convolution Neural Networks (CNN) and Recurrent Neural Networks namely, BLSTM for recognition. For the categorization of public and private registration number plate images, the given method by the author explores background and foreground separation and then joins them to make a new solution. Because of high discriminative ability, CNN has been used for feature extraction and BLSTM has the ability to extract context information based on the past information. The author combines the merits of CNN and RNN to handle the issues of poor quality, complex background, blur and noise. Benchmark UCSD dataset and identification experiments on MIMOS dataset shows that the proposed recognition is better than existing methods. The identification result of various other methods including the method proposed in the given paper after and before categorization shows that classification is beneficial for accomplishing a good recognition rate for a posh dataset instead of implementing a replacement method.

In this work[2], the author has taken the problem of multilingual and multi-norm car registration number detection and recognition in real time. First network detect license plates the way it is trained on the complete new image using the newest SOA deep learning based detector mainly YOLO v2. The second part is to use the crop image to identify the registration number from plate photographs[7]. The author compares two recognition engines: a sequence labeling method which recognizes the entire registration number plate without character level segmentation. The proposed system is strong to illumination and weather and is capable of achieving full registration number plate recognition rate of 97.67% int the GAP-LP dataset[9], and of 91.46% in Radar dataset with an inexpensive computational time. So as to scale back the time and price of annotation processing, the author has proposed a replacement semi-automatic annotation procedure of registration number plate images with labeled component bounding boxes.

JES Journal of Engineering Sciences

Vol 11, Issue 7,July/2020

ISSN NO:0377-9254

In the proposed work[3], the author uses YOLO-Darknet Deep Learning Framework[10] which addresses the issue of registration number plate of vehicle. It consists of 7 CNN layers to detect a single class. Author used an AOLP dataset[12] which contains six digit car license plates. The digits in the license plate are detected by sliding window protocol and each window is then detected by a single YOLO framework. The proposed system achieved approx. 98.22% and 78% of license plate detection and recognition respectively. The proposed work consists of a single phase detection and recognition process. Accuracy has been compromised in the proposed work to improve the processing speed of the system.

In this paper[4], the author has introduced a GrabCut algorithm with some modification to localize the number on the license plate. Their approach additionally uses the traditional Grab Cut algorithm with feature extraction method[13], which uses geometric information to provide accurate foreground extraction. It is a robust algorithm and also very efficient for vehicle plate number localization[14]. The proposed algorithm can be used in any country to detect license plates of different shape and size, complex background, trends, different brightness, climatic conditions and variations. The algorithm has been tested with 500 license plate image models from various countries.

In this paper[8], the author describes a ALPR system which uses synthetic images which are required to train the model. This system is developed around the Convolutional neural network. In this paper, the author proposed a method of generating synthetic license plate images. Then, the author described a modular ALPR system which uses two CNNs. One for plate detection and other for character recognition[18] which is experimented over real plate images collected from three different countries. Convolutional Neural Networks (CNNs) have emerged as the cornerstone to efficiently solve a number of computer vision problems [16] such as object detection [17,18].

In [19], Researchers have used the tracking and face tracking app in videos and cameras that can be used for multiple purposes. The purpose of the paper is an in-depth study of face detection using OpenCV. Tabular comparisons are made to understand algorithms in a simple way. It talks about various algorithms like Haar cascades and Adaboost. This paper aims to assist in understanding the first relevant aspect of facial recognition. Local detection can be termed as the removal of facial features using a pattern recognition system. Both MATLAB and OpenCV can be used to build such prototypes and systems. In this paper they have done their research using OpenCV. The reasons for using openCV were Speed, Cost and Portability.

3. PROPOSED METHODOLOGY

The sub-parts that this security model contains are:

Initially the model is trained with the datasets such that it can detect the registration plate of four wheelers directly from vehicles from real-life domains. The second model is used to recognize the characters present on the registration plate forming a License Plate Recognition sub-model. The final system is used to detect the faces and identify them during exit which helps our security system to spot the face of the driver. Then the vehicle number plate gets mapped to the driver face with the help of these models.

3.1. License Plate Detection

1. LICENSE PLATE IMAGES: The images are taken of four wheeler vehicles along with the extended visibility of there license plates and then classify these images as training and testing images.
2. LABEL IMAGES: Since we are using a supervised learning algorithm to train the models, then manually labeled images are needed and thus have a square border on the damage area called a bounding box with a precise image of the characters.
3. LOCALIZE THE LICENSE PLATE: License plate is localized by using sliding window protocol. The area of image where the license plate is localized the window turns green and hence detected.
4. CNN MODEL TRAINING: The model is trained using a supervised learning algorithm. The algorithm used is a 5-layer CNN model which is used to detect the license plate from the license plate image.
5. RESULTS: The result of this part is a cropped image which is further provided as an input image to the character recognition model.

3.2. Character Recognizer

The character recognition will takes place by model trained with datasets and this will be implemented through these steps:

1. DATA USED FOR TRAINING: As the classification of the characters is related to image processing so the computer fonts images used as the registration plates training data. We use UKNumberPlate fonts which are alphanumeric.
2. IMAGE CONVERSION: The synthetic images are transformed into comma separated values format.
3. CLASSIFICATION OF TRAINED AND TEST 4. DATA: For training and testing purposes, we take a significant amount of images.
5. TRAINED MODEL: After providing the necessary accuracy, the model has been trained and we achieved a required model. This model is used to require a fragmented image as input of a registration number plate and recognize the characters written with high accuracy.
6. RESULTS: The result obtained and viewed using OpenCV.

Vol 11, Issue 7,July/2020

ISSN NO:0377-9254

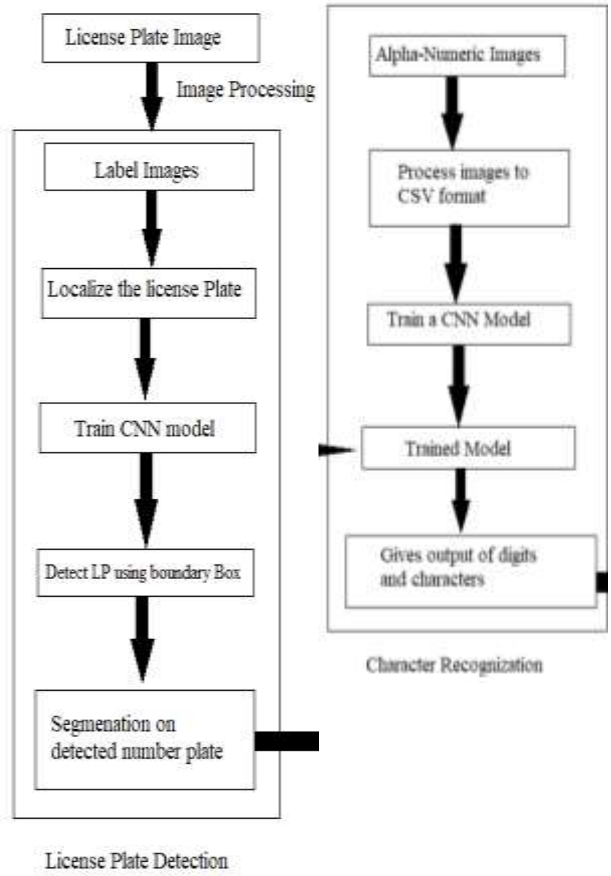

Fig. 1 Proposed Methodology for Car License Plate Recognition

3.3. Face Detection and Recognition

The model that recognize the faces, is built from simple facial recognition technology as described :

1.DATA INPUT: To create the face detection and recognition model, take a proper and good quality image of an individual with an appropriately sized label dataset assigning your name or specific label vs. folder name.

2. PREPROCESSING OF DATA: After successfully accepting the dataset as a training dataset or video, convert the received image into a grayscale.

3. ERADICATION OF OUTLIERS: A more detailed feature is used for background removal, extraction, classification and selection.

4. IDENTIFICATION OF FACES: A boundary box on a given image around the face, which means that the model has the ability to detect faces by tensor of pixels in the image which is given as an input.

5.EXTRACTION OF FEATURES FROM FACES: Get features of face from the images, including areas surrounding the face, eye color, facial structure, or other features used by the LDA. For detailed in-depth training using PCA can make many images for different parts.

6. IMAGE ENHANCEMENTS AND FEATURE SELECTION: This cutout is used to distinguish features such as distance between different components on the bounding box face, which helps its owner to identify the face.

7. MODEL TRAINING: It involves the creation of the right model architecture so that the model can gain the knowledge it needs to apply in the real world.

8. USAGE OF TRAINED MODEL: The model identifies the face of the person or the original owner of the vehicle, which allows the trained system to provide the notification with its certified driver.

9. IRRELEVANCY OR UNCERTAIN HANDLING: Any impropriety for analysis or identification refers to the additional identification need for the vehicle registration plate.

10. TUNING OF MODEL ATTRIBUTES: The model performance can be increased through the optimization of model's hyper attributes and by improving the dataset.

3.4.1. License Plate Detection

In this phase we train a deep CNN model to detect the registration plate. This phase is liable to classify and localize a registration number plate within a given image. For training, we use synthetic images of registration number plate images alongside the network is trained by using license plate annotation. License Plates are detected using sliding window protocol shown in fig 2 and localize the license plate on the image.

Fig. 2 Sliding window protocol[6]

3.4.2. Character Recognition

This phase deals with the recognition of character using a deep CNN model. In this phase we train the model to recognize various characters of different font and size. In order to train the network to recognize the characters of the license plate, we intend to collect the images of the license plates firmly (no context). Like the real world license plate data collection process above, we create our own dataset of synthetic images in different domains, fonts, different backgrounds and under different lights.

Vol 11, Issue 7,July/2020

ISSN NO:0377-9254

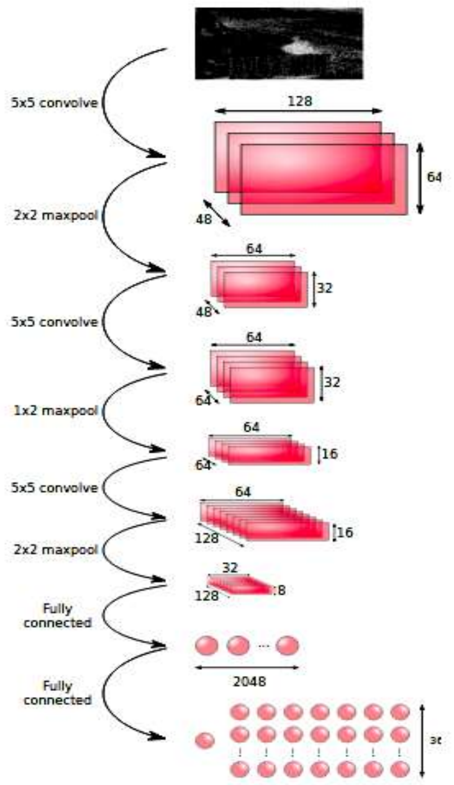

Fig. 3 Five layer network architecture[6]

We use a five layer CNN model shown in fig 4.2 to classify the images and to recognize characters from the license plate. First we create a 5*5 convolve and then find 2*2 maxpool of image till we refine the image to the extent level. Then the extent image is fully connected with the neural network.

3.4.3. Face Detection and Recognition

In this phase we divide the process in two parts i.e. face detection and face recognition. In face detection process we used the OpenCV library that provides face detection algorithm i.e. HaarCascades. In this process we train the model with some positive and some negative images so that the model is able to detect the faces with high accuracy. Positive images refers to dataset images having faces present in it and the negative images refers to dataset images having no face present in it.

For facial recognition we used the LBPH Face Recognizer provided by OpenCV. In this process we train the model with the dataset containing images of various personalities and the two owners of this project. Every individual has their name as the label of their images. The model was trained with the dataset images. After the training of the model, recognition of the individual took place. We recognise the person from the real time images in which the model takes the picture from real time video and recognise it with the trained results. We got better results with a frontal face in real time with LBPH Recognizer.

face_recognizer = cv2.face.createLBPHFaceRecognizer()

4. SYSTEM ARCHITECTURE FLOW

4.1 Car License Plate Detection and Character Recognition

The architecture of our LPR system, which is organized as a cascade of three distinct phases. Using traditional object detection protocol i.e sliding window protocol(fig. 2) and convolutional neural network, this model is implemented..

The full procedure of our number plate detection and character recognition system.The process of ALPR is divided as follow:

1. Dataset Acquisition
2. Deep Convolutional neural network training
3. Result analysis and testing

Before using the trained model for detecting and recognizing the license plate we first purified, described and organized the dataset gathered in a specific format.

5. DATASET ACQUISITION

5.1 DATASET COLLECTION AND PREPARATION

For the license plate detection we create a dataset of various synthetic images. We merge the background images with cropped license plates and create a dataset of 1000 images to train the model. Data set of license plate is shown in fig. 4.

Feature of dataset used :
Dataset type: Indian number plate like data.
Vehicle type: Cars
Images size: 128*64
Number of training images: 1000 images
Images type: Greyscale

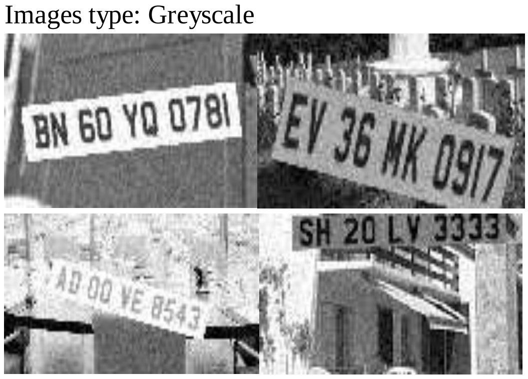

Fig. 4 Training dataset images

Vol 11, Issue 7,July/2020

ISSN NO:0377-9254

5.2 DATASET CLASSIFICATION

In order to test number plate detector CNN, we first take 10,000 background images and generate the various number plate images. After that we merge the background images with the license plate images to create a 1000 images dataset.

We feed batches of 60 samples (30 positive, 30 negative) each time to the network and we compute the gradient deltas minimizing the cost function 1 for each i-th training sample. The Image sample is classified as positive or negative on the following parameters shown in table 1.

A number plate image is also considered as negative if it is not present in the image.

Table I. Dataset Classification

Paramete r	Width of LP	Height of LP
Positive(1)	< 80% of image	<87.5% of image
	>60% of image	>60% of images
Negative(0)	>80% of image	>87.5% of image
	<60% of image	<60% of image

5.3 TRAINING

5.3.1 TRAINING: LICENSE PLATE DETECTION

Similar to [17],License plates are detected using a trained Convolutional Neural Network. This model is liable for finding and detecting a registration number plate in an input image. The network is trained by a dataset of synthetic images (fig. 4.4) alongside vehicle plate annotations. Since the training information we are produced will cover the various real time changes like blur images, high brightness, etc.

5.3.2 TRAINING: LICENSE PLATE RECOGNITION

The system is strong enough to filter the non-character pieces from the license plate. For this we use a variety of fonts and good quality samples of hard non-positive synthetic image samples. train_step is used to train the model with given learn rate and loss.
t_s = tf.train.AdamOptimizer(lr).minimize(l) (where lr = learn rate and l = loss)
If characters exist in LP, then it recognizes the character. On an equivalent line as [15], to train a deep CNN model for the task of character identification. Our network consists of 35 different classes which consist of 10 classes of numeric characters i.e 0-9 and and 25 classes of alphabetic characters i.e A-Z excluding O (because 0 and O has some form of similarity). We ignored usage of lower case letters because the license plate does not consist of lower case characters.

5.4 TESTING

In testing the model we take an image and give it as an input to the model. The trained model performs license plate detection and then character recognition.

1. The first step is to localize the license plate in image using sliding window protocol and then crop the image which is further forwarded to character recognizing phase to recognize the character.

2. After receiving the license plate image the trained model is used to recognize the character from the image and produce output as shown in fig. 8.

Fig. 5 Output of image with 50% success rate

Fig. 6 Output of ANPR with 100% success rate

5.5 Face Detection and Recognition

There are 4 basic and easy steps to computer coding for facial recognition, which are the same as steps that a human brain uses for recognizing and identifying human faces. These basic steps are:
1. Detection of Face : Find the face in the picture.
2. Gathering of Face Data : Gather the data related to face(i.e. facial images for this case) of the person to whom you want to identify.
3. Training Recognizer : Feed the extracted face data and names against their respective data of every face to the recognizer so that it can save or learn the face data with the name of each individual as given in the label.

4. Recognition of Faces : Now Feed new faces of the same or different individual to find if the trained model of face recognizer recognizes them correctly.

For facial recognition we use OpenCV's LBPH (Local Binary Pattern Histogram) Face Recognizer because both Fisher's faces and Eigenfaces are not able to provide good results in low lighting conditions and we can’t be sure about the exact light state every time. To overcome this error we used the improved version of it i.e. LBPH Face Recognizer. The main idea of this identifier is rather than looking at the whole image, try to get its local structure by surrounding and neighbouring pixels comparison.

1. Preparing Training Data : The more images will be used in the training, the better. This principle is important because it is the only way to train the base identifier, so that different faces of the same person can be taught to the recognizer as shown in fig 8.
So, two people in our training data will have to be at least 12 images. All the training data is within the data folder, the training-data.
This folder contains a subfolder for every person, named as s1, s2. For example, the subfolder called s1 means that it contains images for person 1.
We can divide this step into the following steps:
Read sub-folder names provided in folder training-data. We have the s1 and s2 folders.
Extract the label number assigned to each person. We assign the integers collected at this stage to each face that appears next.
After reading all the pictures, apply Face Detection to each one.
Data is added to each face with the corresponding person label.
2. Train Face Recognizer : OpenCV provides three face identifiers i.e.
Fisher Faces, cv2.face.createFisherFaceRecognizer()
LBPH (Local Binary Patterns Histogram): cv2.face.LBPHFisherFaceRecognizer()
Eigen Faces, cv2.face.createEigenFaceRecognizer()
In this work we used the LBPH identifier for the training phase identifier.
face_recognizer = cv2.face.createLBPHFaceRecognizer()
3. Prediction of Faces : This is the stage in which our algorithm is able to detect our face. We took a test image of each person and used the face detection algorithm and sent those faces to our trained LBPH Face Identifier, where we found that our facial recognition was successful with high accuracy as shown in fig 9.

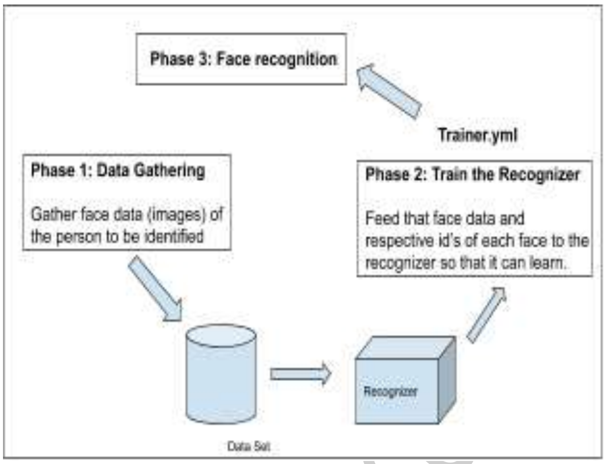

Fig. 7 Face recognition process

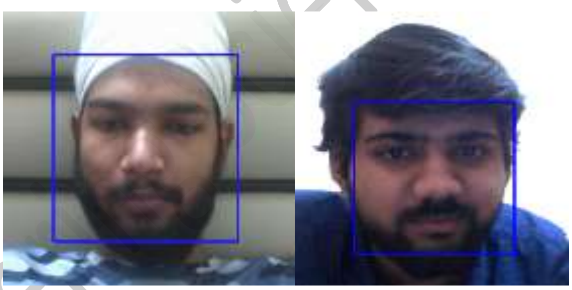

Fig. 8 Detection results of image including two person

Fig. 9 Face Recognition results of image

6. Result and Analysis

6.1 License Plate Detection and Recognition

1. Variable assessment

We test our system by varying some of the Synthetic images. Initial tests confirmed that some variables such as pan rotations which are associated with plate perspective leads to a dropdown of the system performance even if the value is fixed.

Table II shows the accuracy of our license plate detection and recognition on synthetic Indian number plate dataset. If we off the pixelated noise then it leads to a negligible loss in performance while if we off the variance leads to high loss in accuracy of the system. By disabling color

Vol 11, Issue 7,July/2020

ISSN NO:0377-9254

shift and blur done by motion leads to improvement in performance which leads to accurate license plate detection and recognition.

Table II. LPDR system accuracy with different features excluded and included

LPDR Variation	ON	OFF	Δ (ON-OFF)
Pixelated noise	85.2	87.1	-1.9
Illumination variance	90.9	96.9	-6.0
Motion blur	85.3	87.6	-2.3
Reference	87.5	87.5	-
HSL color shift	87.4	86.0	1.40
Overall System accuracy	87.42	89.26	-

Our ALPR system is able to detect the license plate in the image with 95.5% precision and 90% accuracy. The system is very much able to detect the image in any weather condition. Since table 2 shows the performance of LPDR in different conditions. Our ALPR system will be able to detect the license plate which gives the precision and accuracy of 87.5% and 86.8% respectively.

2. Computational complexity

As a final experiment, we measure the GPU computational of our ALPR architecture. Computational complexity of NVIDIA GTX 1080 GPU with high resolution camera images is shown in table 3.

Table III. GPU processing time(ms)

Platform	NVIDIA GTX 1080	
Resolution	640 * 480	1280 * 960
Plate classified (base layer)	12.5	47.5
Plate classified (other layers)	5.5	18.0
Char classified	2.2	2.2
Total proposed	25.5	72.7

Drawbacks of our ALPR System :

1. Our system works only in a specific format. The network makes assumptions of 10 chars exactly visible and gives output with 10 chars.
2. It only works for indian number plate fonts only.
3. The system is slow because it takes several seconds to run on the medium size image.

6.2 Facial Detection and Recognition

The images to be identified are send or upload into the program. The experimental results are shown in the following images. Experiments have shown that images, including single facial objects, are well recognized. The detector is effective with single frontal face detection. There are false-positives and false-negatives for identifying images with more than one face.

Table IV Face detection system accuracy

Face Orientations	Rate of Detection	Rate of Recognition
0°(Front Face)	97.6 %	96%
18°	81.1 %	79%
54°	59.3%	59%
72°	0.0 %	0.0%
90°(Profile Face)	0.0%	0.0%

Through test results, the use of Haar classifiers is an effective method for facial detection. The optimal rate of detection of identified faces is greater than 70%. Classifiers run faster, better in real time as shown in table 4.

Drawbacks of our Face Detection Model :

1. It works perfect for frontal face detection and predicts best at 0° orientation.
2. Proper lighting conditions should be there.
3. It is slow. The System takes several seconds to predict or recognise the individual.

7. CONCLUSIONS AND FUTURE WORK

In this work, we have proposed the combination of CNN and sliding window for license plate detection, extraction and recognition in different phases and SVM or KNN for face detection. Face data and license plate number will be stored at incoming and will be checked at outgoing. This will provide a layer of security at parking and public places.

Model that developed provides 87% LPDR accuracy and the model of facial identification provides 80% accuracy approximately. The LPDR System accuracy in different variable assessment which is done by synthetic images as shown in table II. The accuracy of face identification systems with different aspects as shown in table IV. Other considerations taken after the implementation of the related project suggest that the cameras are of high resolution as part of the hardware installing, which can be used for creating good quality photographs that recognize the vehicle LP and its owner faces as discussed. Overall System performance can also be improved by giving more training dataset and time to train the model.

Using a convolution neural network (deep learning technology) to develop automated systems able to ensure the safety and security of automobiles in parkings or passing through the target community was the main goal of this project to implement. This completes with better results and hence creates a better opportunity for future forecasts at the security level.

We can extend this project for further domains like tax collection, car parking system, traffic violation systems etc.

REFERENCES

[1]. Palaiahnakote Shiva kumara, Dongqi Tang, Maryam Asadzadehkaljahi, Tong Lu, Umapada Pal and Mohammad Hossein Anisi, "CNN-RNN based method for license plate recognition ", The 4th Asian Conference on Pattern Recognition (2017).

[2]. Yousri Kessentini, Mohamed Dhia Besbes, Sourour Ammar, Achraf Chabbouha, "A two-Stage deep neural network for multi-norm license plate detection and recognition", Experts system with Application (2019).

[3]. Hendry and Rung-Ching Chena, "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning", Image and vision computing (2019).

[4]. Ayodeji Olalekan Salau, Thomas Kokumo Yesufu and Babatunde Sunday Ogundare, "Vehicle plate number localization using a modified GrabCut algorithm", Journal of King Saud University – Computer and Information Sciences (2019)

[5]. Bai, X., Cong, Y., Wenyu, L.: "Strokelets: a learned multi-scale mid-level representation for scene text recognition", IEEE Trans. Image Process., 2016, 25, pp. 2789-2802

[6]."ANPR using deep learning algorithm ", https://github.com/matthewearl/deep-anpr, Accessed 20 october 2019

[7]. Montazzoli, S. Jung, C.: "Real time Brazilian license plate detection and recognition with deep convolutional neural networks". Proc. SIBGRAPI on Conf. on graphics, Pattern and Images, Brazil, 2017, pp. 286-294

[8]. Bjorklund T., Fiandrotti A., Annarumma M., Francini, G., & Magli, E.(2019). "Robust License plate recognition using neural networks trained on synthetic images". Pattern Recognition, 93, 134-146

[9]. Naimi, A., Kessentini, Y., & Hammami, M. (2016) "Multinational and multi-norm license plates detection in real traffic surveillance environment using deep learning". In Proceedings of the 23rd international conference on neural information processing - volume 9948 (pp. 462-469). New York, NY, USA: Springer-Verlag New York, Inc.

[10]. T.Kumar, S.Gupta, D.S. Kushwaha, "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, ACM, New York, NY, USA, 2016, pp. 53-57. https://doi.org/10.1145/3033288.3033332.

[11]. H. Li, P. Wang, M. You, C. Shen, "Reading car license plates using deep neural networks, Image and Vision Computing 72 (2018) 14-23. http://www.sciencedirect.com/science/article/pii/S0262885618300155. https://doi.org/10.1016/j.imavis.2018.02.002.

[12]. Cheang, T. K., Chong, Y. S., & Tay, Y. H. (2017). "Segmentation-free vehicle license plate recognition". IEEE Transactions on Intelligent Transportation Systems, 5(1),42-53.

[13]. Boykov, Y., Jolly, M.P., 2001. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. International Conference on Computer Vision. 1, 105-112.

[14]. Comelli, P., Ferragina, P., Granieri, M.N., Stabile, F., 1995. "Optical recognition of motor vehicle license plates", IEEE Trans. Veh. Technol. 44(4), 790-799.

[15]. B.-H. Chen, S.-C. Huang, "An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks", IEEE Trans. Multimedia 16(3) (2014) 837-847.

[16]. J.Schmidhuber, "Deep learning in neural networks: an overview", NeuralNetwork 61 (2015) 85-117.

[17]. M. Mathieu, Y. LeCun, R. Fergus, D.Eigen, P.Sermanet, X.Zhang, "Overfeat: Integrated recognition, localization and detection using convolutional networks", 2013.

[18]. Y. Netzer, T.Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, "Reading digits in natural images with unsupervised feature learning", in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011, 2011, p.5.

[19]. Kruti Goyal, Kartikey Agarwal, Rishi Kumar, "Face Detection and Tracking Using OpenCV", in International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017.