International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Automated Vehicle Security System using Convolutional Neural Networks and Support Vector Machine

Shubham Agarwal, Kushagra Goel, Anirudh Jain, Pratibha Singh

Abstract*: With the rise in the infrastructure in the global economy, there is a need to impact the growth of security systems such as enhancing the security of vehicles at public places, societies or places with crowd. This could be done by keeping up with the monitoring of vehicles through vehicle License Plate Recognition (LPR). Since the numbers of vehicles are increasing on road day by day, it is essential to bring automation in its detection and recognition procedure. The objective of this presented work is to model a real time application to recognize license plate from a vehicle at parking of any society or public places via surveillance cameras. This paper mainly focuses on implementing the concept of component security which is marked by the presence of a blended system with car license plate recognizer as well as face recognizer recognizing its real owner. In proposed Automated Vehicle Security System (AVSS), the achievable model accuracy for Automated LPR model is 94% marked with the use of Tyserract for character recognition and model accuracy for facial recognition is raised to a mark of 83%. This model provides remarkable results and a need of another system where the owner or permitted drivers for a vehicle are mapped to vehicle license plate which could be made to use as collaboration to make it a real life deployable application.

Keywords*: Convolutional Neural Network, Face Recognition, License Plate detection, MobileNet Architecture, Optical Character Recognition, Support Vector Machine, Tesseract

I. INTRODUCTION

This paper is to integrate two models in a real time application of surveillance system in which first model is built to detect and then recognizes license plates of vehicles entering or exiting from parking mainly of any society via surveillance cameras and second model is to recognize face of person driving vehicle to match it with already mapped (vehicle no, face ids) data. This data is maintained for each vehicle which is almost static for a society but is dynamic for vehicles security in public places. The characters extracted from a vehicle will be used to keep record of a vehicle at "in" and "out" time of the society which will solve problem of security remarkably. Mostly vehicles are recognized by its license plates which can be easily read by persons but it is difficult for machines to recognize it always up to the acceptable level of precision. A license plate is as a dark spot for a machine with specific intensity and luminosity within a particular area of an image. It motivated us to design a vigorous system able to perceive human and the vehicle's registration number from corresponding captured images. Every year many countries witness large number of hit and run cases of road accidents. People lose their life due to denial of responsibilities and mostly due to ignorance. This result in making third party insurance mandatory by the Govt. of India but people generally escapes from it also. So motivation of our work is to take support of technologies including deep learning to impose laws and to make roads safer by recording and reporting details of vehicle and its driver's face to the undersigned. The proposed model capture images of license plate of a vehicle and the driver's face and then use this information to recognize vehicle and authenticate the owner of that vehicle during entry and exit from a parking lot. An external device connected with system will generate an alarm on finding the false identity of the owner and notify the guards about the issue. The proposed model is implemented in three stages:

- **Initial Stage:** The first stage of AVSS focuses on license plate recognition of vehicle and then recognizing characters printed on plate in ideal static phase.
- **Medium Stage:** In second stage system emphasis on face recognition of driver.
- **Enhanced Stage:** The vehicle plate recognition phase is combined with the number plate recognition phases to identify the authorized vehicles and also its authentic driver. Further actions will be initiated in case of any mismatch in any level of information.

II. LITERATURE SURVEY

Extensive literature is existing for Image Processing and also applications of Machine Learning in digital image processing emphasizing on image pre-processing, detection, recognition, classification and also analysis of its different measures due to its various application domains. This paper explores ways of applying machine learning and image processing in vehicle security system and also presents a deep learning based model developed for real time automation providing a security product to its customers.

Manuscript published on 30 August 2019.
***Correspondence Author(s)**
Shubham Agarwal*, Department of Computer Science & Engineering, ABESEC, Ghaziabad, India. Email: shubham.15bec1088@abes.ac.in
Kushagra Goel, Department of Computer Science & Engineering, ABESEC, Ghaziabad, India. Email: kushagra.15bcs1069@abes.ac.in
Anirudh Jain, Department of Computer Science & Engineering, ABESEC, Ghaziabad, India. Email: anirudh.15bcs1065@abes.ac.in
Pratibha Singh*, Department of Computer Science & Engineering, ABESEC, Ghaziabad, India. Email: pratibha.singh@abes.ac.in

In further discussion one part will focus on detection and recognition of vehicle's license plates and another part will emphasize on its owner's face recognition. Two state-of-the-art methods for detection and recognition of license plates are briefed here among many available in literature. These methods suggest significant applications of deep learning providing better ways for acquisition of dataset, pre-processing of data, feature selection, model selection, training and validation, testing and evaluation of performance of model and tuning of parameters. The first method proposed to collect datasets of car license plates from an open environment and afterward process this retrieved information using Deep Learning models. Its purpose is to deal with the difficulty level of obtaining extremely varying forms of characters of different fonts, sizes, blurred level, distortion, occlusion and highly obscuring backgrounds. The method used in this model emphases on detection of plates along with recognition of its characters through fragmentation of letters written on Car Licence Plates. They used two well-known datasets of Caltech Cars Dataset and AOLP Dataset comprises of license plates of many vehicles for implementation purpose. Convolutional Neural Network (CNN) is used to detect characters and Recurrent Neural Networks (RNN) with LSTM Model is used for recognition of sequential features from images. The proposed method needs to improve its computation speed as it is very slow for real-time application. After surveying more methods to improve it, one method recommended in literature is to use Fast R-CNN. [1]. The second method limits its research and implementation to recognize license plates of motorcycles in Thailand. Their vehicles have a 3-line type number plates and the recognition system needs to take care of assorted settings of taken pictures as color, character sharpness, dust/dirt, light, shadow, language, etc. when training the model. This model detects and recognizes characters on captured via segmentation. The dataset used for training the model consists of high resolution real images with varying sets of angle of positioning. This paper also examines the currently presented implementation strategies as Hidden Markov model (HMM), Support Vector Machine (SVM), Self-Organizing Maps, Multi-layered feed forward Neural Networks (MFNN), probabilistic NNs, etc. They also explored pattern matching methods like Hausdorff distance, normalized cross correlation and Root Mean Squared Error. The implementation part mainly considered efficient and very effective CNN models as Inception-v3, MobileNet and Single Shot multibox Detector (SSD) [2].

Face recognition is another one of the mostly used applications of machine learning models. It is used in extensive ways in applications as unlocking smart phones or house doors, prevent fraud in banking, identifying a missing person, attendance monitoring, security systems and many others. The comparisons of various mostly appropriate algorithms as Adaboost, Camshift and HaarCascades illustrate that detection and recognition of faces using HaarCascades is one of the most convincing and efficient way. Since this paper is based on comparative study of predefined models, a better approach would be to self-train the model using Neural Networks [3].

The implementation phase further explored several approaches for recognition and two of them are summarized here. Approach 1 [4] detects texts present in images of natural scenes which helps to identify number plates from images of car. It filters data to a level that detects and recognizes characters from number plates. Detection of text is processed in two steps:

- Feature extraction using Histogram of Gradients (HoGs)
- Classification using Support Vector Machine with linear kernel

This method performs sequence of steps of preprocessing of images, object detection (number plate), characters detection, characters recognition and reconstruction further for the purpose of validation. The implementation phase detects various objects in an image but sometimes it also detects and recognize characters associated with excessive elements. This results in retrieval of both required and sometimes unessential list of characters. Besides this it also contend with problems of handling various characters.

Approach 2[5] solves the problem stated above and uses scikit-image module available with python firstly to map all the linked regions in to a binary image. Then it labels them to exclude all those regions from image that do not belong to number plate area utilizing some confines characteristics of license plates as :

- These plates are of rectangular shape.
- Its width must be large than height.
- The ratio of width of region of number plate to the full size of image is between 15 to 40%.
- The ratio of height of the region of number plate to the full size of image is between 8 to 20%.

It follows three steps for implementation which include detection of license plate of vehicle, segmentation of characters and its recognition. This approach is not able to perceive region of plate in images if the plate size is very light or very small with respect to the overall size of an image. Results shown lesser accuracy to recognize characters and proposed to improve further by using Neural Networks. We have identified and used another approach depicted in Fig. 1 and also explained in next section. It has better performance and wide area of applications.

III. PROPOSED WORK

Automated Vehicle Security System (AVSS) has various areas of applications as enforcing road safety, in automatic parking places, at toll collection spots, limiting speed, tracking vehicle and identifying owners of vehicles etc. We have used deep learning architecture for implementation of model underlying product. The image processing field is devised to describe the process of altering and considering images to find an exact pattern. It will resolve the detection and recognition parts using MobileNet-v1architecture, Tesseract OCR and Support Vector Machine to solve our proposed problem statement. Implementation phases and analysis of results of AVSS are separated into two sub-modules which can be implemented parallel on several machines.

Car Licence Plate Recognition Module: This module is for vehicle detection from real life images, plate detection from vehicle and character recognition from Licence plate. The model is trained for vehicle detection from real life background in images. Mobilenet architecture is used for object detection and Tesseract Optical Character Recognizer (OCR) is used to identify characters from recognized number plates of four-wheeler specifically car.

Face Recognition Module: This module is for face detection and recognition. It supports security system to authorize vehicle owners. The corresponding data for matching must be provided initially to the system before operation. The license plate of vehicle is mapped with its driver which is confirmed further to ensure practical usage of our prototype.

A. *Car License Plate Recognition (CLPR) Module*

CLPR model as shown in Fig. 1 can further be categorized into two sub-modules:

i. Car License Plate Detection

- IMAGES: These are input images of front and back side of cars of size 640×480 with enhanced visibility. This dataset of images is divided into two portions with 80% of images are used as training data and remaining 20% are used as test data.
- LABEL_IMAGES: As we used supervised learning to train our model, thus images are labeled manually by drawing a rectangular box around required area of number plate called as bounding box and also characters are correctly represented in that marked region.
- GENERATE TF RECORDS: After preparing the dataset TF record are generated which consists of coordinates of the bounding box for a number plate and is an XML representation of the image.
- MODEL TRAINING: To train a model, TF records are forwarded to MobileNet-v1 [6] architecture and adequate weights are assigned to the model and is trained using CNNs.
- RESULTS: Cropped license plate of a car is extracted from the trained model and is used as an input image for character recognition. To pass an image to character recognition model two approaches are generally used: segmentation and fragmentation. Our proposed work uses fragmentation which requires the resultant license plate to be passed in fragments where each fragment carries a singleton image of a character.

ii. Character Recognizer

- TRAINING DATA: Images of fonts known to system are used as training data for training of character recognizer.
- IMAGE CONVERSION: The features obtained from images are saved in file of .csv format.
- TRAIN: TEST DATA: A substantial number of images are used for training data and a small part of images are used as test data. We tried various ratios and used 80% as training data and 20% as test data.
- MODEL BUILDING: We build a model with 37 class classifier (26 alphabets, 10 numbers, 1 null character) to train these images.
- TRAINED MODEL: A required model is attained after the model has been trained using images of various characters and the desired accuracy is obtained. Fragmented images of the license plate are now provided as an input to this model and characters are recognized efficiently.
- RESULTS: To showcase the obtained result OpenCV is used.

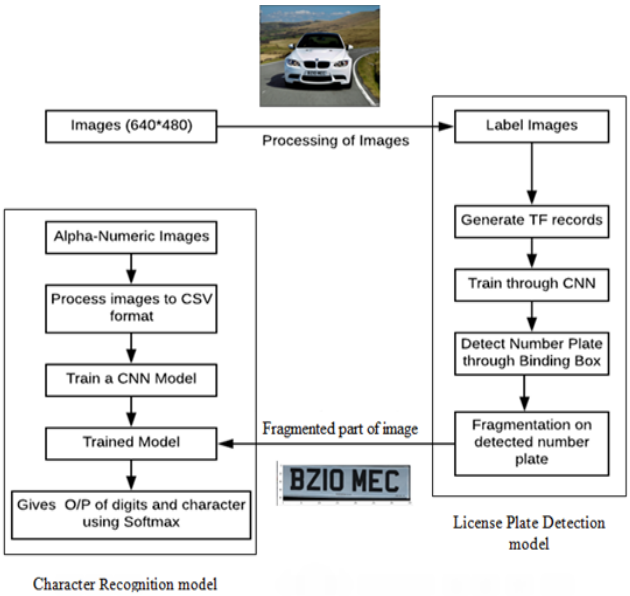

Fig. 1. Proposed Car License Plate Recognition model

B. *Face Recognition Method*

A generic face recognition technique is used to build this model as described below.

- INPUT DATA: Minimum one clear image of a person is taken labelled with his name or some other unique label as the folder name. A labeled dataset is prepared with images of different persons same as with any classic image classifier to build the model.
- DATA PREPROCESSING: Obtained images either from video or snapshot are converted to greyscale as an important preprocessing step.
- OUTLIERS ERADICATION: In this process background from the images are removed for obtaining more specific features.
- FACE IDENTIFICATION: If a model is able to identify the face of a person from the tensor of pixels in input test image then a bounding box can be detected around the face of the person in the image.
- FEATURE EXTRACTION: The facial features from the image are obtained which includes variation in face structure or skin tone, eye color, the region around the face or other features using LDA. For detailed in-depth training, various images can be generated as in terms of varied components using PCA.

- FEATURE SELECTION IMAGE ENHANCEMENTS: The bounding box is used to uniquely identify the features from a face such as distance between various parts of the face which will be useful in recognition of faces of its owner.
- MODEL TRAINING: In this process, an appropriate model is built so that it could learn sufficient knowledge which is to be applied to the real world.
- USING TRAINED MODEL: The trained model is subsequently used to recognize a face and turn the signal green if the person is authenticated user of the car.
- UNCERTAINTY OR IRRELEVANT HANDLING: The security system will raise a signal for additional information to identify a car license plate or its driver if there is any missing information in analysis or recognition of the person.
- MODEL PARAMETER TUNING: Hyperparameters of a model can be optimized or dataset can be increased to increase the performance of a model after training and evaluation of the model is ended.

IV. IMPLEMENTATION AND RESULTS

This model is developed on Core i5-3210M CPU@2.500Hz, 64 bit OS, an x64-based processor with 4.00GB RAM. As Python is one of the most popular languages used for deep learning algorithms, architectural model building and for building applications related to machine learning, we used Anaconda3 for Python for implementation of this model. It provides many efficient libraries such as Tensorflow[7], OpenCV[8], Torch, lib, Keras, Scikit-learn, Pandas, etc. We have developed our model by using Tensorflow for detecting license plate of various vehicles and Tesseract engine is used for recognizing characters from the detected number plate. For recognizing the faces efficiently, Caffe and haar cascade models were used for training the model. Mainly Tensorflow, LabelImg, OpenCV, facial_detection, Object_detection, Imutils, lib, pickle packages are used of Python 3.6. We have assumed that images provided as the input are brighter and of good quality for both license plate and faces. The dataset consists of ideal surroundings with Low level of outliers. Apart from the above listed dependencies such as anaconda(Spyder) installed and downloaded during Anaconda installation and python packages there are no other dependencies which are needed for the development phase of this model. All the images which are available in training dataset for model training must be of equal size and good quality and should have strong visibility of characters to pass into LPR model for better efficiency. All the possible matches of a person should be covered in the dataset used for the facial recognition model. The implementation of this paper is completely based on specific version of python with its newly updated packages. All image must have a single face only so that the face can be extracted with high confidence. The face should be rotated in such a way that both eyes are allocated along the same direction.

The implementation starts with the phase of detection and recognition of license plate followed by the face recognition phase. Its steps are described as following:

A. *Dataset Acquisition*

"The mind and soul of a training model is its training dataset" is a noteworthy indicator referring to the urgency of a good dataset selection. To begin with this implementation, we began with the datasets focusing on license plate detection and character recognition on license plate and then face recognition.

• Baza_slika [9]: This dataset is prepared using high end digital cameras under several weather conditions and is available for use as free. The image database contains over 500 images of four wheelers like cars, buses, trucks, etc. under diverse climatic conditions with the focus to make the system robust with respect to illumination conditions and noise like dirty license plate or bad visibility during for or rain. This set is used for making the model capable of detecting the presence of a Licence plate from real environment. Test images are collected from the web satisfying similar conditions.

• Character recognition: Chars74K [10] and LetterRecognition.data.Z [11] datasets were used to recognize the characters once the licence plate has been detected. Both the datasets are a collection of 74,000 images consisting of 62 classes (a-zA-Z0-9) under diverse font family and text type appearances.

• For facial recognition, we have taken into consideration images with front-face of human to mimic the photo of a driver taken while driving a four-wheeler.

B. *Image Labelling*

The training of this model took place under supervised learning paradigm referring the desired output that needs to be known at the time of training. So all the images considered under dataset acquisition requires to undergo a process wherein they are labelled as per the desired output.

• LPR model: In order to determine the presence of a number plate in real environment, it is essential to instruct the model by defining the correct coordinates around the number plate boundary which is achieved by an image annotation tool called Labellmg. It is written in python and uses Qt5 for graphical interface. These annotations are saved as xml files in a format used by ImageNets supporting YOLO. These xml files are then gathered, saved as .csv file format and converted to tf understandable format.

• LP character recognizer: Characters printed on license plate is a multiple class classification where about the folder name represents the character of license plate as detected from the trained pattern.

• Recognizing the driver's face: To detect and recognize the driver's face, images of authorized personnel is collected with various face orientations and these are mapped to the license plate and the name of the person creating the folders for the same. Fig. 2 represents the user interface of labellmg in order to understand the proper working of labelling tool. The image which needs to be labelled is simply selected from the File in menu bar and then information such as license plate number and coordinates of license plate boundary is added to it.

The generated xml file is then grouped and then converted to csv format once all the images are labelled.

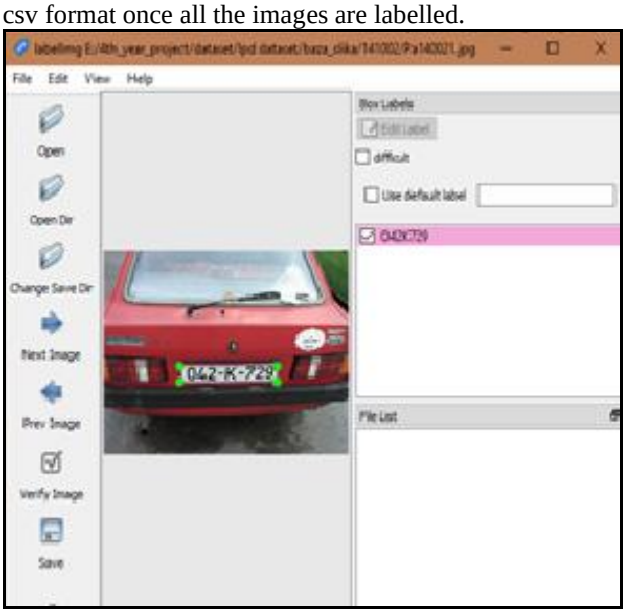

Fig. 2. Labellmg User Interface

C. *Model training*

The labelled images are then proceeded to a convolutional neural network with known architecture called mobilenet. The parameters like image size, learning rate, momentum, regularization weight, image batch_size, resolution, color scheme and other evaluation or tuning factors are set to maximize the obtained results as shown in Table I. The training is achieved using normal tensorflow processing due to the lack of GPUs on training machine.

Table- I: Tally describing the architecture parameters

Parameter	Field	Value
Ssd	Num_classes	90
Image reshaper	Height	300
	Width	300
Convolutional box	Dropout	0
	Kernel size	1
	Activation	RELU
	Regularization_weight	0.000004
	Batch_size	16
	Momentum_optimizer	0.9

The advantage of using the pretrained model with learned configuration of parameters and its architecture is that it will be fast in computation. It uses bounded box representation for object detection and has a separate .config file for the developer to try different configurations of its architecture. The detected license plate of a car from test dataset is as shown in Fig. 3.

D. *Trained model usage and results*

The detected image is cropped using the rectangular bounding box dimensions. Tesseract [12] is a tool which is used to recognize the characters present on the detected number plate. It has a support for Unicode system capable of recognizing hundreds of languages out of which only English has been chosen for current implementation. The cropped image is transformed to grayscale and is supplied to OCR to recognize characters impressed on its license plate. The tool provide the recognized characters along with the input image as shown in Fig.4.

Fig. 3. Output of an image from the test dataset

Fig. 4. Clipped Images along with the recognized characters for the two input number plates (100% success rate)

E. *Face Recognition*

Following steps are executed for car driver's face detection after detection of number plate in previous phase:

- *Extraction of embedding from training samples:*

Face embedding is used to record all essential points for recognizing faces in an input image. It generates an array of regions. Each region represents a face detected in image in terms of coordinates of bounding box, data object with a vector and its dimension. This data can easily be represented and analyzed in .csv, .txt or .xls format to be processed further. Thus the main implementation step for face recognition is to generate this file for face embedding. We used deep learning model for feature extraction generating a 128-dimensional real valued feature vector for detecting a face. We marked highest confidence value as threshold which must be satisfied for detection to avoid detections with weak confidence level.

These steps are repeated for all images in training data and generated embedding data with its label are added to a dictionary and saved in pickle file to use in advanced steps.

- *Model Training:*

The model is trained after the extraction of embeddings from training samples to recognize faces and map it with label. Some most eminent classification algorithm in machine learning for supervised classification problems are Decision Trees, Naïve Bayes, Support Vector Machine (SVM), and Random Forests etc. We used Support Vector Machine with linear kernel function for classification of images and initialized scikit-learn LabelEncoder to encode and normalize images with labels. The output of the model and LabelEncoder is recorded as pickle files after training.

- *Face recognition using OpenCV:*

We need subsequent files for recognizing face using build trained model:

- Detector: It is a pre-trained Caffe model to detect any face present in images
- Embedding: It is a pre-trained Torch Deep Learning model to generate 128-D vector of face embedding.
- Recognizer: It is a trained model used for recognition of test data

Following steps are essential to be applied on a captured image for detection and recognition of faces after training and validating the face recognition model:

- Detect face from input image
- Generate 128-d vectors of face embedding
- Use trained SVM model for recognition

The prototype of Automated Vehicle Security System is shown in Fig. 5. The object detection model [13] has obtained 84% accuracy and the face recognition model has attained accuracy of 94% as specified in Table II. Tesseract Engine for character recognition is an OCR model with an acceptable accuracy of 98.2% also depicted in Table II for this particular application. The important point observed as part of implementation is that high resolution cameras must be installed. It will generate high quality input images to efficiently recognize the car and its owner. The performance of entire system can further be improved by increasing training samples and can be applied to identify different types of License plates by extending the training data.

Table- II: Performance evaluation of LPD

MEASURE	VALUE
No. of images tested	483
No. of images detected	436
Accuracy of LPD	90.26 %
Accuracy of Tesseract OCR	98.02 %
Accuracy of Facial Recognition model	94 %

V. CONCLUSION

With the rise in the infrastructure in the global economy, there is a need to focus on enhancing the security of vehicles at public places and societies. In this paper, we proposed to implement an Automated Vehicle Security System which will detect and recognize license plates of vehicles during entry and exit from parking of societies, organizations and other public places through installed cameras. It is enhanced further by combining face recognition method with number plate recognition to authorize vehicle and owner of the vehicle also. It is required to provide mapping of images of owners and drivers with their cars for regular parking areas of vehicle. For

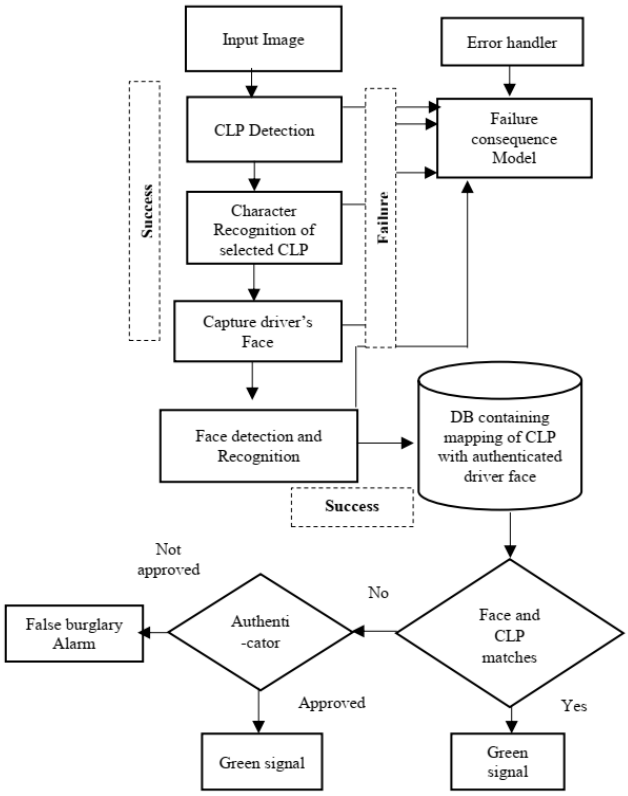

Fig. 5. Automated Vehicle Security System

visitors this mapping may be prepared during entry and verified at the time of exit. This integrated application using pre-trained models restrict unauthorized persons to take away vehicles and will also avoid incidents if installed on different spots on roadway. Thus it is a blended system with car license plate recognizer as well as face recognizer. We identified best architecture of mobilenet and its parameters and used Convolution Neural Networks for license plate detection, tesseract for character recognition and pretrained Caffe model with linear SVM for face detection and recognition. The whole idea is to build an automated system which must remain light weight using pre-trained NNs and must be capable to ensure security and safety of vehicles when parked on regular parking or visitors' parking. Methods implemented have provided good results and therefore suggests to advance security system which records and verify only authorized vehicles but do not whether driver is authorized or not. It will also reduce the possibilities of entry of illegitimate persons who can otherwise take entry driving a car in colleges/University etc.

Retrieval Number: J98180881019/19©BEIESP
DOI: 10.35940/ijitee.J9818.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Capturing and recognizing images of all persons in a car has a good scope for future to increase security level. The model suggested here does not need lot of computations after deployment due to use of pre-trained models and legitimate mappings (vehicle_no, driver_id) are also not large for verification.

REFERENCES

1. Hui Li, Peng Wang, Mingyu You, Chunhua Shen, "Reading Car License Plates Using Deep Neural Networks", *Image and Vision Computing, 2018*
2. Wichai Puarungroj, Narong Boonsirisumpun "Thai License Plate Recognition Based on Deep Learning", Computer Science Department, Faculty of Science and Technology, Loei Rajabhat University, Loei, 42000, Thailand, 3rd International Conference on Computer Science and Computational Intelligence 2018
3. Kruti Goyal, Kartikey Agarwal, Rishi Kumar, "Face Detection and Tracking using OpenCV", International Conference on Electronics, Communication and Aerospace Technology 2017
4. "Image Text Recognition in Python", http://francescopochetti.com/ text-recognition-natural-scenes/
5. "Developing a License Plate Recognition System with Machine Learning in Python", https://blog.devcenter.co/developing-a-license-plate-recognition-system-with-machine-learning-in-python-787833569ccd
6. Tensorflow, https://www.tensorflow.org
7. OpenCV, https://opencv.org
8. Baza Slika,www.zemris.fer.hr/papers/LicensePlates/english/ baza_slika.
9. Chars74K dataset, http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/
10. LetterRecognition.data.Z, 26 capital letters in 20 fonts in csv format, https://archive.ics.uci.edu/ ml/datasets/letter+recognition/
11. A basic approach and hands-on tutorial for object detection and COCO model, https://www.oreilly.com/ideas/object-detection-with-tensorflow/
12. Tesseract-OCR: github library and documentation, https://github.com/ tesseract-ocr/
13. Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

AUTHORS PROFILE

Shubham Agarwal has done his matriculation and higher secondary school from Tata Chem D.A.V. Public School, Babrala followed by graduation in B.Tech specialization in Computer Science and Engineering from ABES Engineering college, Ghaziabad. He has worked on many real time projects using bootcamp on design principles and web development fundamentals using ruby, rails, javascript and React.js as learning framework. Currently he is working as a software developer at Mavenhive Technologies Pvt Ltd, Bengaluru.

Kushagra Goel is a software developer at a product based organisation in India. He has been graduated from ABES Engineering College, Ghaziabad in Computer Science and Engineering major in 2019. He is passionate to apply his skills and knowledge in solving global problems. His works and projects are mainly confined in AI domain including data analysis and Neural Networks. The idea for this research came up as a solution for vehicle surveillance in communities. He loves to read and write articles and learn new things.

Anirudh Jain has done his matriculation and higher secondary school from St. Mary's Academy, Saharanpur followed by graduation in B. Tech in Computer Science and Engineering from ABES Engineering College, Ghaziabad. He has been a part of the Computer Society of India(CSI) for 2years(2016-2018) and many other groups including an NGO and NSS. He has worked as an intern in IDEMIA in the field of Machine Learning, and then a Data Analyst Intern for Forensic Investigation and Dispute Services(FIDS) department in Ernst & Young. Currently, he is working as a Software Test Engineer at InfoEdge India Ltd. He likes to learn new things every day and everywhere which keeps him rejuvenated.

Pratibha Singh is an Associate Professor in Department of Computer Science & Engineering in ABESEC Ghaziabad. She is B.Tech (HBTI K) and M.Tech (IITD). She is co-author of books on Operating System, Distributed System and Pattern Recognition and has published papers in International conferences and journals. She has worked on many projects solving real world problems using Machine Learning.