Security Systems built for vehicle thefts using Convolutional Neural Networks

Eunice Olorunshola
Computer Science
Georgia State University
Atlanta, Georgia
Email: eolorunshola1@student.gsu.edu

***Abstract*— Vehicle thefts have been an issue over the years, and it is still an issue today but with technology continuously changing and evolving it paved way for security systems. This could be done by utilizing the strategy of facial recognition and face detection Since the number of vehicles are increasing on the road each day it is important to bring automation in the vehicle theft detection and recognition procedure. Previous research has developed new technology using neural networks such as a Vehicle License Plate Recognition that can be installed in high security surveillance area in parking lots or public places used to detect the vehicle license plate number using low level digital image processing. This research mainly focuses on the blend of methods like license plate recognizer as well as face recognition recognizing the vehicle real**

owner through live images and detection using neural networks.

Keywords—Vehicle thefts, security systems, Neural Network, Convolutional Neural Network, Face Recognition, License plate detection

I. INTRODUCTION

The importance of security systems is still growing, today almost all parking lots, public places are equipped with surveillance systems. This research mainly focuses on the blend of methods like license plate recognizer as well as face recognition recognizing the vehicle real owner through live images and detection using neural networks. Every year vehicle thefts causes an impact to the U.S. according to not just emotionally but also financially therefore, the work on vehicle security is significant. Due to technology growing more secured way to stop vehicle thefts have been created than using static features such as fingerprints, passwords, iris scanning, and so on. The proposed model is implemented in three phases:

- **First Phase**: The first method is the license plate recognition using the Vehicle Number Plate Recognition System, and recognizes the characters printed on the plate number in phases it follows.
- **Second Phase**: The second method is face detection that uses skin like district segmentation, edge density descriptors and skin tone features.
- **Improved Phase**: The third method is the vehicle license plate recognition system. This system capture images of license plate of a vehicle and the vehicle owner face and then use the information to recognize the vehicle and authenticate the vehicle owner to enter and exit from a parking lot or any public place.

II. Literature Review

Vast literature is existing for image processing and applications of neural networks in digital image processing emphasizing on image preprocessing, detection, recognition, and classification. This research finds ways of applying neural network and image processing in vehicle security system. In further discussion one part will focus on detection and recognition of a vehicle license plate and another part will emphasize on the vehicle owner face recognition. Two state of the art methods for detection and recognition of license plates are discussed here. These methods suggest significant applications of neural networks, the first method proposed to collect datasets of vehicles license plates from a public place and afterward process the retrieved information using convolutional neural network models. Its purpose is to deal with the complicated level of the different types of license plates that are in each vehicle. Such as the variety of the font styles and design, since every license plate number has important selective features the detection algorithms were categorized into color and shape depended methods to detect the owner vehicle license plate number. The color dependent methods are used to detect only on the color and the shape method is used to detect the rectangular shape in the vehicle. Convolutional neural network is used to detect the characters in the license plate numbers through the fragmentation of letters. The vehicle license plate number system has uses two phases, the first phase is the low-level image processing, the second phase is the genetic algorithms based neural networks.[1] During the methods of the image processing phase, it segments the license plate number from the vehicle through color to gray scale conversion, dynamic adaptive threshold, and morphological operations. The second phase is when neural networks-based algorithms identify the owner vehicle license plate number by matching the texture features of the optical characters in the rectangular shape area [1]. After reviewing more methods, one method recommended in literature is to use MATLAB software package to detect the vehicle license plate number by creating my own algorithms, graphical user interfaces, plotting data values and using the image processing and neural network toolbox to implement the vehicle license plate number recognition [1]. Face recognition is also one of the mostly used models. The comparisons of various mostly face detectors technologies as VJ face detector, AlexNet , cascade Convolutional Neural Network, multitask cascaded convolutional neural networks, and R-Convolutional Neural Network [2]. The framework phase further explored several approaches for face recognition and three of them are summarized here. Approach 1 [2] data processing the owner of the vehicle face image is resized to different scales. Approach 2 [2] implement the images into the cascaded convolutional network. Approach 3 [2] Show the final face of the vehicle owner as an image pyramid and marthe position of the five features points of the face. The results obtained showed that neural networks without training is not as efficient for vehicle face images, the next plan is to work on the second phase by using the vehicle owner face training set to get a better performance of detection.

III. RELATED WORK

In previous papers, there were many proposed techniques for license plate number recognition , many paper used the ANPR(Automatic Number Plate Recognition) system but faced many challenges when it came to detecting the license plate number such as selective features: font styles, design, colors, shape, and fast-moving vehicles. Also faced Lighting conditions: reflections, shadows and blurring. Since these papers include many advantages and disadvantages, I tried to use the similar methods used in the papers to build my system that will overcome those challenges. There are also many previous papers on face recognition and verification , but only a few focus on detecting the object before recognizing the driver. One of the earliest proposed works related to detecting a driver face is where a technique is used to detect stable face and eye regions both during day and nighttime. Infrared rays were used to illuminate the face of the driver by capturing the movement of the eye and how stable the face using the IR-LED. This system was used for detecting when a drive becomes drowsy, but no object detection was used to make sure that it was a human face that was being detected. Several other approaches proposed a system that monitors driver facial features by using a pretrained face model. This was the best method for me to use and include a technique that detects the object first before detecting the facial regions and recognizing the driver face .

IV. PROPOSED WORK

A. VehicleNumberPlateRecognitionSystem (VNPR)

The proposed work for my vehicle number plate recognition is presented in this section, input to the system is a vehicle image that has been acquired through a device that captures images and the output is the vehicle number printed for recognition.

The flowchart of my proposed system is shown in figure 1 which consists of the following main steps:

1. input image
2. Image Pre-processing stage:

- Color to Gray Scale Conversion
-Noise Reduction

- Dynamic Adaptive Threshold
- Morphological Operations
3. Genetic Neural Network stage:
- OCR method
4. Character Segmentation
5. Vehicle Number Extraction
6. Vehicle Number Recognition

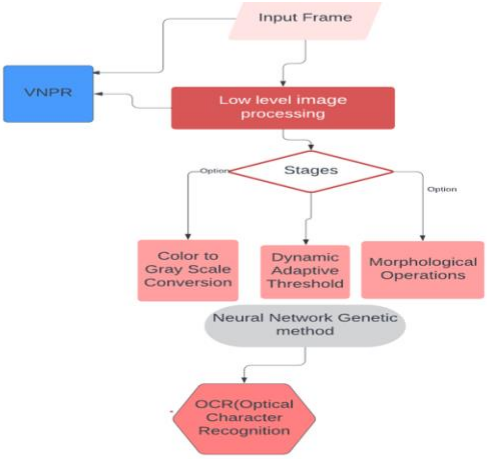

Input Frame
VNPR
Low level image processing
Stages
Option
Option
Color to Gray Scale Conversion
Dynamic Adaptive Threshold
Morphological Operations
Neural Network Genetic method
OCR(Optical Character Recognition

Fig. 1. Flow Chart of the Vehicle Number Plate Recognition (VNPR) System

Vehicle Number Plate Recognition (VNPR) System is used to overcome the challenges when it comes to detecting the license plate number through the selective features of the font styles, design, colors, shapes, fast moving vehicles, obstructions, and lighting conditions such as reflections , shadows, and blurring. For VNPR system to work correctly it is separated in two phases which can be implemented parallel on several machines.

2.1 Input Image

This is the first step that obtains the image of the vehicle with number plate which in my case is taking from existing set of images.

2.2 Image Preprocessing

This is the first phase and an important step in any image analyzing system. The image processing field is used to describe the process of changing the images to find an exact pattern . It is processed using three methods: (1) Color to Grayscale, (2) Noise reduction, (3) Dynamic Adaptive Threshold , and (4) Morphological Operation.

- **Color to Grayscale Conversion**: is when the camera captures the number plate images as an input in RGB format. The captured images could have effects on some of the concepts of a number plate image as an input such as noise, low light, shadows, and illuminations. The color to gray scale conversion is used to remove these concepts that affects the vehicle number plate image. Once the image is converted to gray it cancels out the noise and illuminations.
- **Noise Reduction**: Image noises are distorted caused by faulty in camera or result of poor visibility from changing weather conditions There are various types of noise that can cause distortion in images such as Gaussian noise , salt, and pepper noise. In this proposed work, I will use Gaussian blur to reduce the noise and bilateral filter. Bilateral filter does not only reduce noise but also keeps the edges strong. This method is more effective than median filter that has been used in previous proposed systems.
- **Dynamic Adaptive Threshold**: Is when the vehicle number plate is segmented from the input image. During the segmentation, the number plate is localized from the input image using the dynamic adaptive threshold algorithm. The gray input image is also important in this stage because it is converted into binary image. The binary image correlates with each pixel, then compensate with the threshold value in the background and foreground pixels of the input image.
- **Morphological Operation**: This method correlates with the adaptive threshold to get the region of interest in the input image of the vehicle number plate through erosion and dilation. After using the process of erosion and dilation in the region of interest, then the vehicle number plate is segmented from the input image without showing any illumination noises.

2.3 Genetic Method

Optical Character Recognition: During this phrase OCR is used as the process of recognizing any characters either printed or handwritten images, in this case it prints the characters or numbers from the vehicle plate and display them on the image. The genetic method follows a genetic algorithm to reduce the layers of the convolutional network. The convolutional layers are used to train the vehicle number plate , multiple training process must occur to reduce the possibility of errors in characters identification. After the training process, testing the model is important to measure the accuracy of the trained system.

2.4 . Vehicle Number Plate Recognizer Model Process

- *Image*s :- The input images of the vehicle plate number should be the size 740 x 680 with better visibility due to the input image converting to gray and elimination of the noises and illuminations . The datasets of the images are divided into two portions with 70% of images are used as the training data and 30% as the testing data.
- *Classify_Images* :- Since convolutional neural network is being used , the model of the input images must be trained using supervised learning , previous ideas had classified the images by drawing a rectangular box around the number plate corresponding to the location of the license plate in the input image. But this is not enough to make the image look efficient because the region of the license plate can be distorted and skewed which can affect identifying the characters correctly , because of this I proposed a new idea to apply a transform to turn the input image region as a 90-degree top-down angle for a better view.
- *Collect trained features* :- After preparing the datasets , the trained features are collected which is from the coordinates that marks the region surrounding the number plate and is an xml representation of the image.
- *Train the model* :- The records of the dataset are assigned to the system and the weights are transferred to the model to train it using convolutional neural network.
- *Results* :- The vehicle plate number is shown cropped to make it standout and is extracted from the trained model and is used as the input image for recognizing the characters of the plate number. Previous ideas used fragmentation to get an image to the character recognition model , but for my proposed work will use segmentation which requires segmenting the characters from the vehicle plate itself using low level image processing techniques including adaptive thresholding, component analysis and contour properties.

2.5. Recognizing characters in vehicle plate number Model Process

- *Training Data* :- The fonts of the characters shown in the system of number plate input image
- *Features Collected* :- The region of the plate number and extraction of the foreground plate number characters from the background of the plate number are obtained from dataset of the images. And the input images are saved as a csv file to contain large data.
- *Train & Test Data* :- 70% of the images are used for the training data and 30 % as the test data.
- *Build Model* :- The model of the system contains characters including 26 alphabets, 10 numbers to train the input images of the plate number.
- Train Model :- The various characters contained in the input images of the vehicle plate number is trained and after it is trained. A required model is obtained , during the segmentation using the low-level image processing techniques including adaptive threshold, component analysis, and contour properties to recognize the characters efficiently.
- *Results* :- To get the correct results of the characters in the input image , the low-level image processing techniques are used in the model to obtain accuracy. Adaptive thresholding is used to extract the characters from the background to eliminate extra images that are shown on the plate number such as the branding logos, decoration on the frames. This allows for the viewing focus to just be on

the characters in the plate number and allow for spacing between each character. Contour properties technique is used to segment the foreground vehicle plate characters from the background. The last technique connected component analysis locates the region on the plate and finds the characters that are similar in each section of the image. And the results of the data collected are saved as a csv fi shown in OpenCV.

2.5. Model Architecture

2.5.1 Convolutional Neural Network

My vehicle number plate recognition model is developed on Convolutional Neural Network(CNN). CNN has four main components: (1) Convolution Layer, (2) Activation function, (3) Pooling layer , and (4) Classification layer(this are the fully connected network) . CNN can be seen correlated with my model architecture in figure 2.

- **Convolution layer:** The main purpose of convolution is to extract features from a input image. It also learns the image features by using small squares of input data.
- **Activation function :** is a node that is put in between neural networks , there are many activation functions but for my proposed work, I focused on using ReLU function , which is a function that does not activate all neurons at the same time.
- **Pooling layer:** reduces the dimensionality of each feature map but keeps the important information . In my proposed work , I used max pooling which is a 2 x 2 window layer and take the largest element from the feaute map in the window.
- **Classification layer (this is the fully connected network) :** The main purpose of fully connected layer is to use the features for classifying the input image by predicting the highest probability for the correct class.

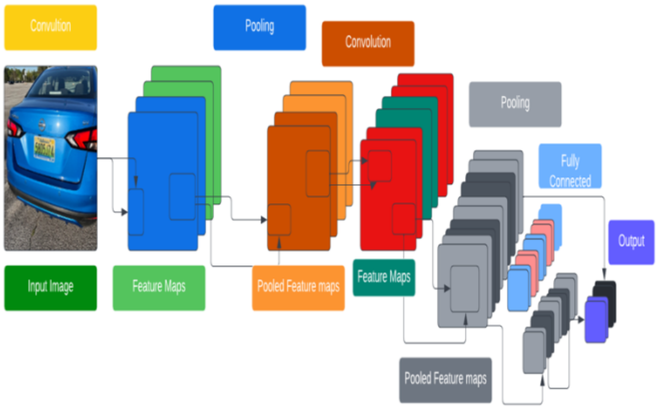


Fig. 2 CNN Architecture of (VNPR) System model

B. Face Recognition System

Face Recognition system method is used to recognize and detect the owners of the vehicle by collecting facial images of drivers and using convolutional neural network to extract the features . The system also supports security systems to authorize the vehicle owner. The vehicle owner facial images correlate with the vehicle owner while driving to correctly detect the owner. The approach of the system is to follow my proposed framework of capturing the image of the vehicle owner in the vehicle, face detection, face extraction, feature extractor, face quality and the results that contains three stages : database constructing, input the constructed images, and output the final facial images and mark the position of the five feature points that correlates with the five feature points of the face, eyes, nose, and every corner of the mouth.

The proposed work follows two main stages :

1. Input the constructed images
- Input the image
- Image Preprocessing: image resizing, mean subtraction, and scaling.
2. Output the final facial images and mark the position of the feature points of the face, eyes, nose, and every corner of the mouth.

3.1 Input the constructed images

3.1.1 Input image

This is the first step that obtains the image of faces from existing set of images of faces in vehicles

3.2 Image Preprocessing

This is the first phase and an important step in any image analyzing system. It is processed using three methods: (1) Image resizing, (2) Mean Subtraction, (3) Scaling.

- **Image resizing:** The input image is resized to a fixed resolution for faster processing
- **Mean subtraction:** The input image is normalized to find the closet pixel value.
- **Scaling:** The input image size is scaled according to the fixed resolution to ensure it can be processed faster.

3.3 Face Detection and Extraction:

Previous work has used the popular Open CV that is known for face detection , but for my proposed work Dlib CNN face detector will be used since it presents more accuracy than Open CV. Once the face is detected , a rectangular box will be shown to eliminate the background and focus only on the region of the face to be used correctly for the feature extraction.

3.4 Face Extraction

This is the step that to get accuracy of the facial region , it is important to use pre trained convolutional neural network to extract the features. By using wider face dataset and randomly selecting as the training set for classifying the face and removing the last layer which contains variations in scale, lighting, occlusion and pose. This also helps with the classification of the original training set. After the pre training the dataset the pre trained model parameters are updated.

3.5 Face Quality Observation:

The challenges that come with after extracting the images are from the camera angle and the way the vehicle driver head is positioned while capturing their face. Therefore, it is important to analyze the quality of the face before moving on to further steps. During this step we can eliminate low quality images that are needed for classification to make training the model accurate. Not having accurate detections will cause the system to fail.

3.6 Feature selection Image Efficiency:

The bounding box around the facial region is used for efficiency to identify the features extracted from the face such as the position between the various five features of the face such as the eyes, nose, and every corner of the mouth which will be useful in making the vehicle owner face recognition more accurate and easier to detect.

.

V. EXPERIMENTAL IMPLEMENTATION AND RESULTS

A. Vehicle Plate Number Character Recognition Experiment

This system was developed on a MAC OS I used Anaconda3 and implemented in python , it provides many libraries such as TensorFlow, dlib, OpenCV, keras, matplotlib, easyocr,numpy and tyssteract. My proposed system was developed using tensorflow for detecting the vehicle plate , tyssteract and easy ocr was used for recognizing characters from the detected vehicle plate. The packages of python that are used are imutils and OpenCV. The existing image obtained as the input was bright and good quality for the vehicle plate. There were no other dependencies installed for this system, using the two dependencies stated above was enough to have efficient development for the system . The different functions used are cv2,imread , GuassianBlur,bilateralFilter, Sobel, Otsu threshold function , finding the threshold value, morphology Exit, and getsStructuringElement. The implementation starts with the phase of detection and recognition of vehicle number plate.

B. Vehicle Plate Number Character Recognition Experiment Results

1. Input the Image

Figure 3 shown below is the input image from an existing image shown in color

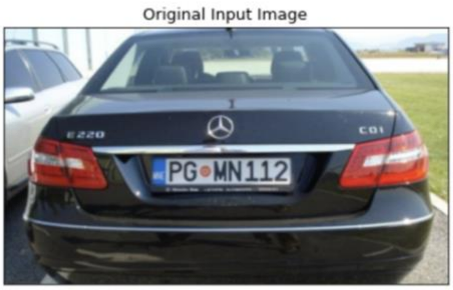


Fig. 3 Original image Inputted

2. Color to Grayscale Conversion

Figure 4 shown below is the input image converted from color to gray , this is needed to reduce the number of colors and removes illuminations and any interference such as low light and shadows that can cause the image to be distorted.

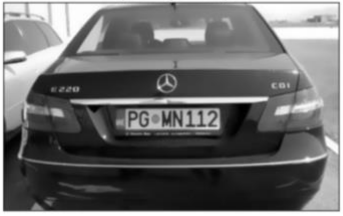


Fig. 4 Original image Inputted converted from color to gray

3. Remove the noise by using Gaussian Blur

Figure 5 below show the gray image blurred to reduce the noise caused by image distortion

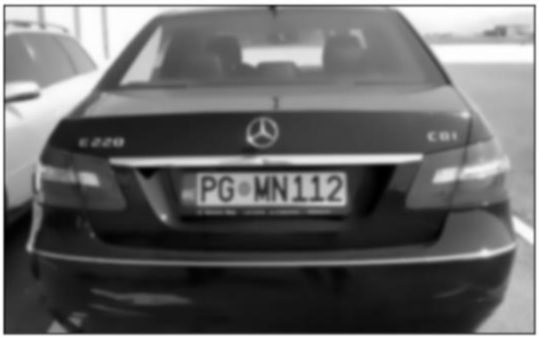
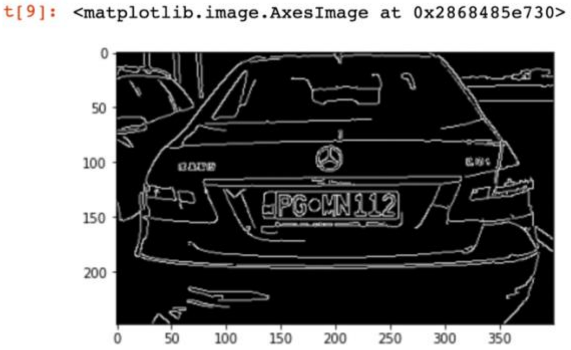


Fig. 5 gray image using Gaussian blur to reduce noise

4. Remove the noise while keeping the edges sharp by using bilateral Filter

Figure 6 below show the gray image using bilateral Filter a more effective method to remove the noise while keeping the edges sharp, this method also used canny edge detector method to detect the edges.

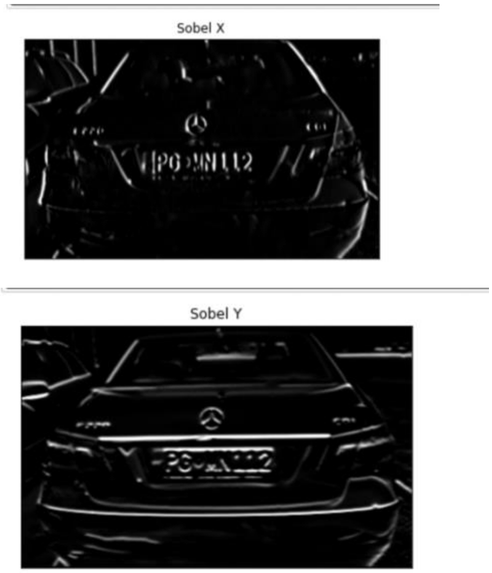


```
t[9]: <matplotlib.image.AxesImage at 0x2868485e730>
```

Fig. 6 gray image using bilateral Filter to reduce more noise and show the edges.

5. Sobel operator used to output the blur image

Figure 7 below show the gray image blurred using Sobel operator with the detected edges bright on a darker region for both x and y direction.



Sobel X

Sobel Y

Figure 7 sobel operator used to blur image for both x and y direction

6. Otsu threshold method used to reveal the light regions in the image

Figure 8 below show the gray image using Otsu threshold method to reveal the light regions to make the vehicle plate numbers more visible for both x and y direction.

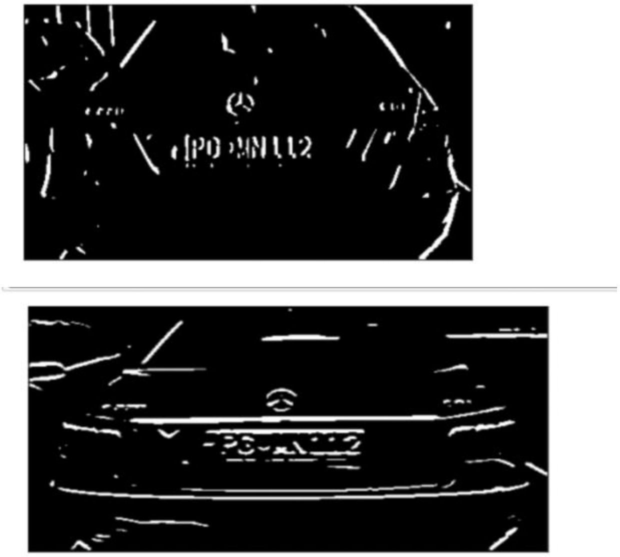
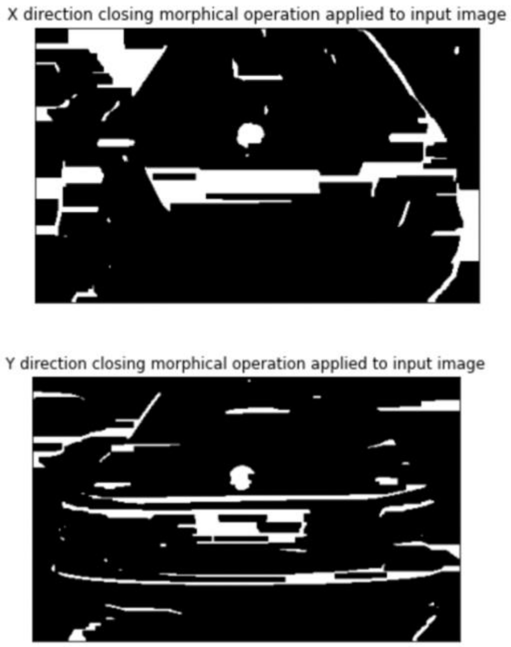


Fig. 8 gray image using otsu threshold method

7. Gray image shown using morphocial operation

Figure 9 below show the gray image using close and exit morphology function to remove the blank spaces between each vertical edge line, to identify possible regions that can contain the vehicle plate for both x and y direction



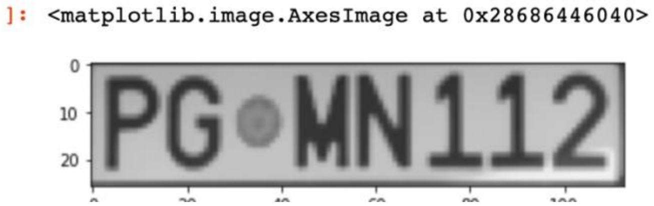
X direction closing morphical operation applied to input image

Y direction closing morphical operation applied to input image

Fig. 9 gray image using morophocial operation

8. Vehicle plate cropped from background of input image in gray

Figure 10 below show the vehicle plate number cropped in gray to show that it was recognized



```
]: <matplotlib.image.AxesImage at 0x28686446040>
```

Fig. 10 cropped vehicle plate number

9. Vehicle number plate recognized and detected from original input image

Figure 11 below show that the vehicle plate was correctly detected and extracted by printing the numbers on the image using easy ocr method

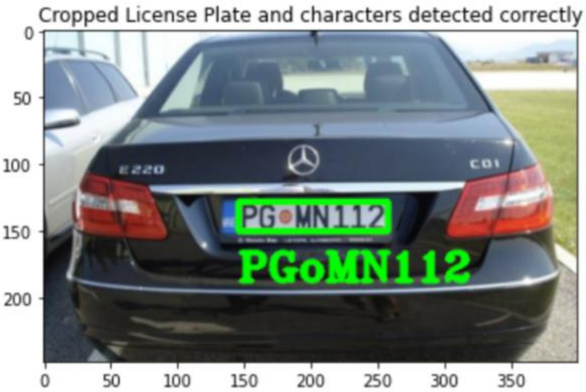


Fig. 10 Original image showing vehicle plate correctly detected and recognized

C. Face Recognition System Experiment

This system was developed on a MAC OS I used Anaconda3 and implemented in python , it provides many libraris such as tensorflow, dlib, opencv, keras, matplotlib, easyocr,numpy and tyssteract. My proposed system was developed using tensorflow for detecting the facw, opencv and MTCNN for recognizing the facial features was used for recognizing characters from the detected vehicle plate. There were no other dependencies installed for this system, using the two dependencies stated above was enough to have efficient development for the system . implementation starts with the phase of detection and recognition of vehicle number plate.

.

D. Face Recognition System Experiment Results

1. input image

Figure 1 shown below is the input image from an existing image of real time driving snapshot.

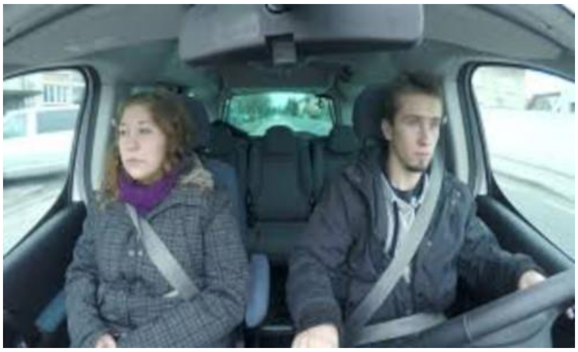


Fig. 1 Original image Inputted

2. Input image resized

Figure 2 below show the input image resized for faster processing

Fig. 2 Original image Inputted resized

3. Face detected

Figure 3 below show the faces detected correctly

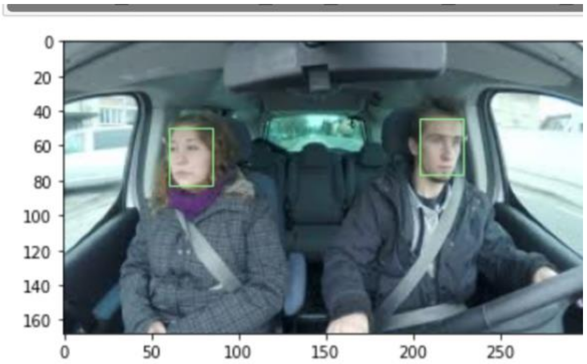


Fig. 3 Face detected

4. Face Extracted

Figure 4 below show the face extracted by marking the five facial region points

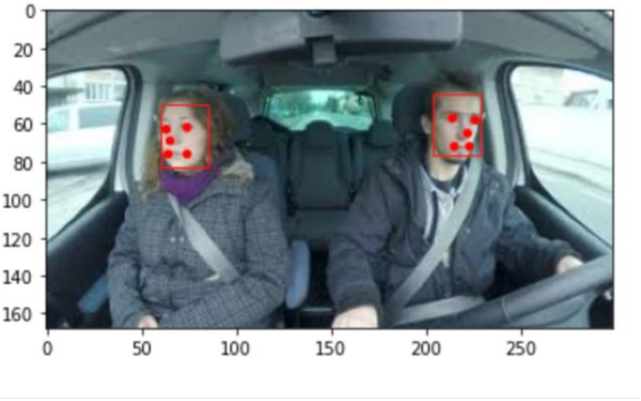


Fig. 4 Face extracted

5. Face detected and recognized

Figure 5 below show that the face was correctly detected and recognized by using object detection to detect that it is a human face for driver authentication**.**

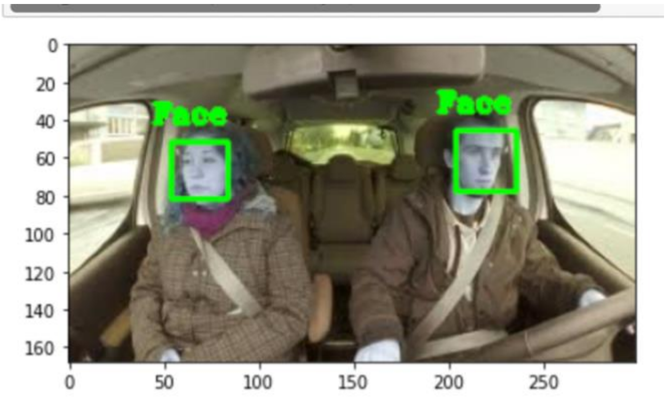


Fig. 5 Face detected and recognized correctly

VI. Conclusion

With the increase in vehicle thefts, there is a need to focus on enhancing the security of vehicles at public places, and parking lots. In this paper, I proposed to implement an VNPR and Face Recognition System which will capture and recognize images of a vehicle and has a good scope for future to increase security level.

VNPR is implemented using many different algorithms and techniques

My proposed methodology initially does the pre-preprocessing steps which includes RGB to gray scale conversion, noise reduction , and binarization.

The vehicle plate is extracted using Sobel's edge detection algorithms. Then the characters are segmented which is given as input to the CNN to recognize the character correctly.

Face detection can be applied in a lot of the different places such as security systems to enhance the security and to track certain people or objects.

The facial detection algorithm tries to localize the faces and then extract each of the faces present in the input.

REFERENCES

[1] R. Rajathilagam, K. Sivamani, R. Seetharaman and D. Nedumaran,"Neural Network based Vehicle Number Plate Recognition System," *2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC),* 2019, pp. 102-104

[2] S.Jing, C. Hu, C. Wang, G. Zhou and J. Yu, "Vehicle Face Detection Based on Cascaded Convlutational Neural Netoworks," *2019 Chinese Automation Congress (CAC),* 2019, pp. 5149-5152

[3] K. Chandra prabha, P. Selavaraj, V.K. Burugari and P. Kanmani, "Image extraction for vehicle theft detection using Nueral Network," *2021 International Conference on Computer Communication and Informatics(ICCCI),*2021, pp. 1-5

[4] T. Nukano, M. Fukumi and M.Khalid, "Vehicle license plate character recognition by neural networks," *Proceedings of 2004 International Symposium on Intelligent Signal Processing and Communication Systems, 2004.* ISPACS 2004., 2004, pp. 771-775

[5] A. Jain, K.Goel, P. Singh and S. Agarwal,"Automated Vehicle Secruity System using Convolutional Neural Networks and Support Vector Machine," *2019 International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019,* pp.4057-4063