ASHESI UNIVERSITY COLLEGE

AUTOMATIC LICENSE PLATE RECOGNITION SYSTEM FOR TRAFFIC MONITORING

CAPSTONE PROJECT

B.Sc. Electrical/Electronics Engineering

Patrick Nana Asiedu

2021

ASHESI UNIVERSITY COLLEGE

AUTOMATIC LICENSE PLATE RECOGNITION SYSTEM FOR TRAFFIC MONITORING

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Electrical/Electronic Engineering.

Patrick Asiedu

2021

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University College.

Supervisor's Signature:

……

Supervisor's Name:

……

Date:

……

Acknowledgements

To my supervisor, Dr Nathan Amanquah whose encouragement and academic advice helped me undertake this project.

Abstract

In Ghana, road accidents are primarily caused by road traffic law violations. Technologies, such as Automatic License Plate Recognition(ALPR), have been implemented to help reduce road accidents in other countries. However, they are not utilized in less developed countries like Ghana. In this project, an ALPR is implemented using two approaches: computer vision algorithms and YOLOv4. The computer vision approach obtained a 71% ALPR accuracy at 3 meters distance and 60 degrees inclination between the license plate and camera at 7 pm night time. The YOLOv4 obtained a 99% ALPR accuracy at 6 meters distance and 60 degrees inclination between the license plate at 7 pm night time.The results of testing show that the YOLOv4 approach was more accurate than the one with computer vision.

Table of Content

Chapter 1: Introduction

1.1 Background

Not many inventions have had a massive impact on the world as vehicles. Throughout history, vehicles have both positively and negatively than any other means of transportation. Being a mainstream and most utilized way of transport, vehicles have carved the way people live in many aspects of life. As populations increased, vehicles made it easier for people to travel between home and work regularly.

However, there are also adverse downsides to the invention of vehicles. As people gained access to vehicles, it raised an alarming rate of road traffic accidents. World Health Organization reports that approximately 1.2 million people die each year due to road traffic accidents [1]. Road traffic accidents(RTA) cause injuries and death to people, as it also generates economic losses to individuals, families, and nations as a whole. RTA costs most countries 3% of their gross domestic product [1]. Over 90% of road traffic deaths occur in the African region and other low to middle-income countries [1]. Several reasons can be associated with why road accidents occur. RTA could be caused by overspeeding, driving under the influence of psychoactive substances, use of unsafe vehicles, distracted driving, and inadequate law enforcement of traffic laws. However, road traffic regulations have been set to govern traffic and regulate the movement of vehicles.

Although traffic rules have been enacted, drivers still violate them, which is an impending problem in Ghana. About 61.5% of Ghana's road accidents are due to traffic violations [2], and this is most common with local bus drivers (tro-tro). Traffic rules like not picking passengers at prohibited areas, overloading, and the most common practice, "running the light," are customary. It is common to see vehicles crossing at road intersections without paying attention to the traffic light. Many drivers do this to cut time

waiting for a green light and fuel. A red traffic light jumper puts his life in danger and the safety of other drivers and pedestrians, leading to injuries and death in some cases. When drivers keep jumping traffic, it creates chaos, and this is the leading cause of traffic jams making everyone get late to their destinations.

1.2 Proposed Solution

To enforce traffic rules to reduce road accidents in Ghana, drivers who violate them need to be tracked and punished or fined according to their violation. This will make them more responsible and alert, hence reducing road accidents caused by traffic law violations. This project aims at easing the process by making use of a technology called optical character recognition(OCR). In recent years, OCR has allowed text in images and scanned documents to become more than just image files but editable text regardless of it being handwritten or printed. Some OCR applications include processing checks without human involvement at banks, scanning passports to save data when booking a flight at airports, and keeping medical history records in hospitals. Among these applications, the most relevant to this project is automatic license plate recognition (ALPR). ALPR has been used over the years to read the license plates of vehicles without human intervention. In countries such as the USA, UK, and Spain, the police use ALPR to track stolen cars, unlicensed cars and monitor traffic [3].

However, this technology does not exist in Ghana because they are expensive to implement. Also, there are complexities such as viewpoint variation and illumination involved in developing such a system. An estimate for an ALPR project in Arizona reports that camera cost in an ALPR system amounts to $8,320,000 out of $9,984,000, which is 83.3% of the total cost [4]. Because of the expensive nature of cameras relevant to an ALPR system, this project aims to implement an ALPR in Ghana while saving cost by

using cheap cameras. Below are problems and objectives this project aims to solve or accomplish :

- The ALPR should be able to extract license plate details under the constraints of a cheap camera (to save cost)
- The ALPR should be able to detect license plate at oblique angles between the camera and the license plate
- The ALPR should be able to detect license plates at poor lighting conditions and night time
- The system should be able to notify violators of their offense with a fine

The other aspect to enforcing this system after it has been successfully implemented is the police. The Ghanaian police department will be notified of traffic rule violators and those who refuse to pay their fine. These people will be dealt with by the police according to the law.

1.3 Problem Statement

Despite the numerous road safety traffic rules to reduce the deaths caused by road accidents in Ghana, road users are at a high risk of injuries or losing their lives. These road accidents are caused mainly by careless acts of vehicle drivers and motorists who violate traffic regulations. ALPR technology can be used to enforce traffic rules to ensure safety. However, this technology does not exist in Ghana because it is complex and expensive to implement.

1.4 Benefit of Proposed System

Implementing an ALPR for Ghanaian roads will deter vehicle owners to be more vigilant and pay heed to traffic rules. This will ensure road safety and will reduce the

number of road accidents that occur in Ghana. When road accidents reduce, injuries and death will be diminished, eliminating economic loss to individuals, families, and Ghana.

1.5 Complexities/Problems with ALPR

Although many ALPR systems have been deployed, it is still a challenging field affected by conditions in the environment. There is viewpoint variation where an object (license plate) can be oriented in multiple dimensions concerning how it is captured [5].

Another challenge in this field is illumination. This when lightning conditions at an environment setting affect the light intensity and quality of the captured image. This could be undesired and affects the accuracy of an ALPR. The speed of vehicles also has a significant impact on the success of an ALPR. Especially where a low-quality camera(cheap camera) is used, captured pictures could be challenging to work with.

Chapter 2: Related Work

2.1 Review of Relevant Computer Vision and Deep learning methods

Computer vision is a field of artificial intelligence that enables computers and systems to process, analyze and obtain information from digital images, videos, and other visual inputs [6]. The field deploys image processing algorithms such as Thresholding, gradients, and edge detection, morphological operations, and greyscaling.

Binary image processing involves converting an image to binary format using Boolean operators.

Morphological operations are a collection of nonlinear operations related to the shape or morphology of features in an image.

Thresholding is the binarization of an image. The process is used to convert a grayscale image to a binary image, where the pixels are either 0 or 255.

Gaussian blurring involves defining a sliding window on top of an image where weighted mean pixels that are nearer to the central pixel of the image contribute to the weight.

Deep learning is a subfield of machine learning concerned with algorithms that learn from data and specialize in pattern recognition, inspired by the structure and function of the brain[5]. Faster Resnet Convolutional Neural Networks(Faster R-CNN), Single-shot detectors(SSD), and You Only Look Once(YOLO) are relevant algorithms used in detecting objects.

Convolutional Neural Network is a deep learning algorithm that can take an input image , assign learnable weights to different objects or aspects in the image and differentiate one from the other.

Faster R-CNN is an object detection network composed of a feature extraction network called a pretrained CNN and a Region Proposal Network(RPN). The RPN is used to prune the number of generated bounding boxes (potential coordinates of an object) to a more manageable size. Faster R-CNNs are very accurate, but they are a little slow, obtaining about five frames per second(FPS) on a GPU when implemented.

Single Shot Detectors take one shot to detect multiple objects present in an image using multi-box [7].

YOLO is an algorithm that also takes a given image and simultaneously learning bounding box coordinates and corresponding class label probabilities, just like SSDs. YOLO and SSDs are incredibly fast. YOLO is capable of real-time object detection obtaining about 45 FPS on a GPU[8]. YOLO has gone through several iterations from YOLOv1 to YOLO9000, up to YOLOv4. YOLOv4 is a combination of up-to-date versions with improved performance [7].

A support vector machine (SVM) is a type of deep learning algorithm that performs supervised learning for classification of data groups.

2.2 Literature Review

Implementing an ANPR is still a challenge since image parameters are highly affected by a complicated environment [9]. Parameters such as viewpoint variation, scale variation, deformation, and illumination, as discussed in chapter 1, significantly impact an ANPR's success. Over the years, there has been a trend in which ALPR is built. Some authors have used computer vision algorithms. Other authors have made use of algorithms in a sub-field in machine learning called deep learning. The key aspect of deep learning is

that human engineers do not design these layers: they are learned from data using a general-purpose learning procedure[10]. Some authors have used an innovative approach of combining both deep learning and computer vision methods. Although these methods differ, the process of building an ALPR can be summarized in two or three steps. The first step is called license plate localization, which detects the possible region where a license plate may be located in a given image. The subsequent, character segmentation involves the separation of each character within the license plate and, lastly, character recognition, which recognizes the characters from the license plate. However, each of these methods has their trade-offs.

An ALPR was made to read Indian license plates using computer vision algorithms [11]. The approach was first to capture the image, run it through computer vision algorithms, and finally read each character present on the license plate. The proposed method for license plate detection in [11] consists of binary image processing, gray-level processing, color processing, and Thresholding. For binary image processing, the authors determined the license plate's bounding box from the image using edge statistics and morphological techniques. This process achieved a 98% recognition rate from 9,745 photos supposing that the number plate frame's edges were perfect [11]. After converting the image into a greyscale image, color processing was done. The motive behind color processing was that due to poor lighting conditions and plate location, the output is not accurate; therefore, to accurately retrieve characters with greater efficiency, color processing was done[11]. Template matching was used for character recognition. It is a procedure of locating the region of a sub-photograph called a template inside a picture. Their approach was tested on several vehicles with entirely different forms and dimensions under different conditions. The authors report that the segmentation method did not produce desired results for plates positioned at certain angles. This was a

significant downside of their approach. Although, a percentage of 75-85% was achieved in recognizing number plates.

There are quite some strengths to the approach of these authors; however, the weaknesses identified are discussed below. The method of extracting characters from the binary image to define the license plate region is time-consuming because it would have to process all the binary objects in the image. Also, there is a high probability of giving incorrect results, especially if there is another text in the image. They should have avoided this when using binary image processing. Also, they should have done a gaussian blur operation right after converting the image to greyscale. It would have made it easier for the edge detector to find the outlines of the license plate. However, the authors used adaptive Thresholding. This method allows handling cases where there could have been dramatic ranges of pixel intensities for different parts of the image.

Another group of authors implemented an ALPR to validate Myanmar vehicle number plates using a similar approach to [11]. What was different about their work is they used a threshold-based system to determine the bounding box of the license plate for different stages of recognition [12]. They tested and had an accuracy of 90%. For future work, the accuracy could be improved by increasing the number of plate samples used. The trend of using computer vision techniques for ALPR is still an excellent option, especially in controlled conditions. However, in uncontrolled conditions, it is better to use advanced deep learning methods such as SSDs, YOLO, and Faster R-CNN at a trade-off of cost since these architectures require a graphics processing unit [5].

Authors in [13] and [14] used deep learning algorithms to build an ALPR. They proposed an efficient license plate recognition system that first detects vehicles and then retrieves license plates from cars to reduce false positives on plate detection. Convolution

neural networks(CNN) were applied to improve the character recognition of blurred and obscure images. A Resnet-Convolution neural network(RCNN) based on a selective search was used to find possible targets of the license plate, and then a CNN was used to determine whether the target is a license plate. After, Support-vector machines(SVM) was adopted to detect the vehicle's license plates. The SVM was trained to classify license plate candidates and non-license plate candidates to efficiently detect the plate. For character recognition, CNN model was used to identify characters that are blurred and skewed. After training with 14,627 samples using a 12gb Nvidia GTx Titan GPU , The CNN model achieved a 99.2% accuracy for the recognition system.

The strengths of this approach were with the use of the CNN model used in character recognition. However, it required the use of a very expensive GPU which is worth about $3000. Also, the use of selective search in RCNN is not the best. Selective Search algorithm takes the place of sliding windows and image pyramids, intelligently examining the input image at various scales and locations, thereby dramatically reducing the total number of proposal ROIs sent to the network for classification[5]. However, for RCNN, they are not as fast and waste a little bit of time. They should have used Faster RCNN to alleviate the need for selective search, which also introduces a Region Proposal Network that runs the CNN classifier once, making it faster[15].

Another group of authors built an ALPR applied for Spanish and Indian license plates, which uses image processing and deep learning methods [16]. In their ALPR, the first phase involved detecting a license plate from a captured image, and then in the second phase, segmented plate is passed to plate recognition to determine the characters and numbers. Also, their work's scope focused on detecting and recognizing multiple cars' license plate from a single frame or picture. Their license plate localization was based on a combination of morphology (where nonlinear neighbourhood operations are used) and

edge statistics. Like [11], these authors did grayscale conversion, Sobel filtering, Thresholding using Otsu's method, followed by morphological operations and contours for their license plate detection. OCR was done with an Artificial Neural Network Algorithm. The model was trained using datasets of Spanish and Indian plates. The paper resulted in the successful detection of multiple license plates.

The work in [16] achieved an ALPR for multiple license plates in a single frame. One strength identified is with the order and combination of image processing effects used. Performing otsu's threshold method after Sobel filter achieves remarkable results.

Object detection is one of the problems in computer vision. It is a challenge to know specifically what objects are inside a given image. However, Deep learning has revolutionized computer vision. With Faster RCNN, YOLO, and SSDs, object detection has been made less challenging with the added speed in executing the object detection.

Chapter 3 : Requirements & Design

3.1 System Overview

To implement an ALPR system to track traffic rule violators while saving cost, a cheap camera integrated into a traffic light system is used to take an image. The image capture is triggered immediately after the red light on the traffic system comes on. This is because it is at this point where people will try to run the traffic light. Hence the image is captured after the red light comes on. The captured image is sent for image processing, where the actual ALPR algorithm is. The ALPR executes and extracts the license plate from the captured image. If the ALPR successfully finds the plate's characters, it sends the plate details to a database containing details of registered number plates. If not, the ALPR retries and exits. After the plate characters have been sent to the database,a comparison is done to identify the vehicle owner's details. If the details are found, a notification is sent to the violator. If not, it exits the system loop. Below is a flow chart representation of the explained system overview.

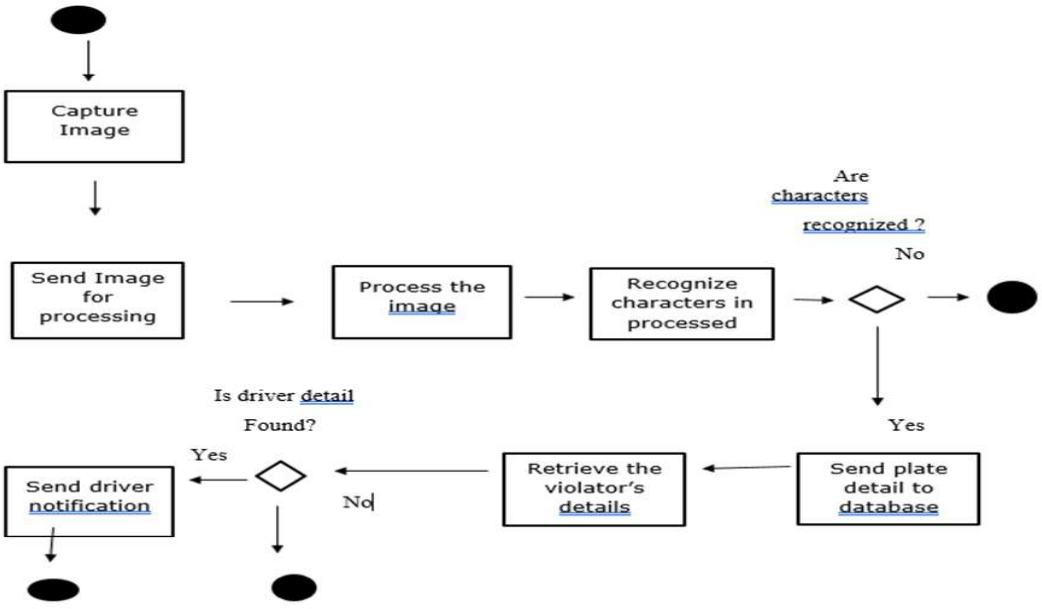

Figure 1 System Flow Chart

3.2 System Requirements

To have a successful system, as described earlier, the following system requirements need to be met.

1. The System should be able to capture images and send them for pre-processing
2. The ALPR algorithm should be able to detect license plates at low-quality camera settings
3. The System should be able to detect the license plate at oblique angles from the camera
4. The System should be able to detect plate under poor lighting conditions
5. It should be able to find driver details from the registered plates database
6. It should be able to notify the violator with a fine

3.3 Scope

The ALPR consists of two main aspects, software and hardware. The software Component of this System also comprises three subsystems. One is the server application that is being run on captured images to extract the license plate. The other is a database that collects the extracted license plate, and compares it to a registered plate list to find details of the violator. Lastly is a Web-based application that sends an email to the violator with a fine. An administrator is given access to this Web-base application to view the violators. The below figure has the same functionality as the explained system overview except for the inclusion of a Web server that allows an administrator to view a list of violators.

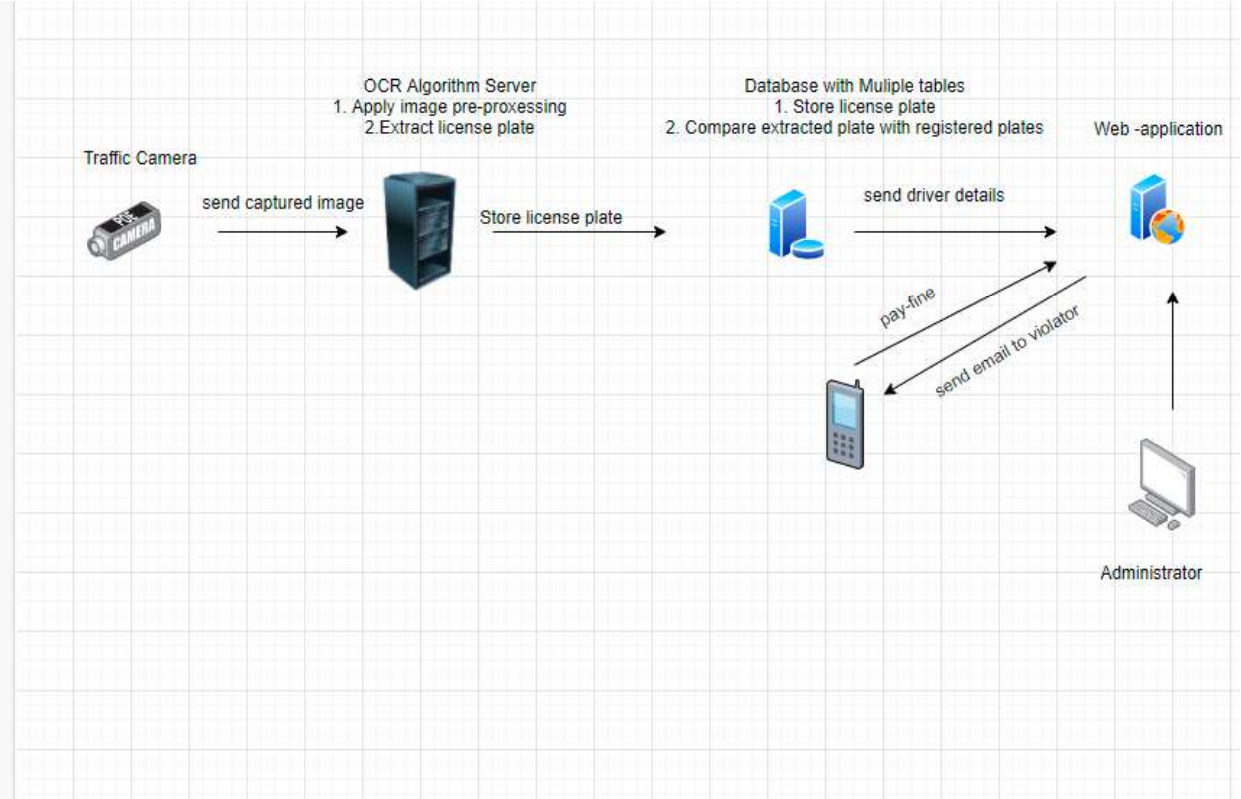

Figure 2 Detailed design of system

The hardware component consists of a camera.

3.4 System Design Interfaces

Due to the system requirements that need to be met, the following interfaces would be needed.

3.4.1 User Interface

A web application based on HTML and CSS provides a simple user interface for the administrator. It is secured with an administrator username and password. After login, a dashboard provides a list of violators with their details based on the date and time.

3.4.2 Hardware Interface

The camera is connected to the ALPR algorithm utilizing streaming its IP address.

3.4.3 Software Interfaces

This consists of the server that will run the license plate recognition algorithm and the database and web app.

To run the license plate detection, below are essential libraries required to be installed . The ALPR is written in python language.

- Open Cv
- Numpy
- Scikit- Learn
- Tesseract OCR

The database used is a MySQL framework within SQLAlchemy, an SQLL tool in Python. SQLAlchemy provides the interface for creating a database that is based on SQL without writing SQL statements. This enables Python to map data from the database to application-defined objects. The database management system used is also MySQL, which has high scalability and standard security requirements.

The database will have two tables. One table will contain the license plate characters that the OCR Algorithm has extracted. The other table contains the details of all vehicles and their owners registered under DVLA. This will include the owner's contact details such as phone number and email. The data in the two tables are matched respectively to the vehicle owners who have violated a traffic law.

3.4.4 Communication Interfaces

- Email protocol: to notify drivers of traffic violation and the amount of fine to be paid. This feature will be developed will Python
- FTP: for sending images from the camera to the ALPR program
- HTTP protocol for uploading license plate details to the database

3.5 Definitions of Libraries

OpenCV is a library of programming functions mainly aimed at real-time computer vision. The library has been around since 1999, written in C/C++, but python bindings have been provided when running the installer [17]. The open cv library works best with Python 3 and is used for most of the pre-processing performed on the image.

NumPy is a library for python language that provides support for large multi-dimensional arrays [17]. It makes representing images as NumPy arrays resource-efficient.

Scikit-Learn is a machine learning library with a set of functions for image feature extraction. It is used for clustering, vector quantization, and classification models.

Tesseract OCR is an Ocr library that utilizes a Long Short-Term Memory (LSTM) network, a kind of Recurrent Neural Network (RNN). Tesseract became recognized in 1995, as it became one of the best OCR engines. It was initially developed in C++ but with bindings for other languages such as Python. Nonetheless, it requires a lot of pre-processing of an image to achieve high accuracy.

Chapter 4: Implementation, testing & Results

4.1 Hardware Setup

This is the central part of interaction within the whole system, and it is the main point for feeding data into the system to be processed. The setup consists of a Samsung galaxy note 3 is set to a minimum resolution of 2mp to achieve a low-quality Camera. This was to imitate the properties of a cheap camera, as cheap cameras do not have higher resolutions. The camera is used to capture images at different distances and angles, elaborated in detail in the testing stage. Another hardware component used is a computer that hosts the OCR algorithm. This component does all the pre-processing to the image and transfers the extracted license plate to the database where it is stored.

4.2 Software Setup

4.2.1 The ALPR

Two approaches were used to build the ALPR algorithm. The first approach was using a set of computer vision algorithms, and the second, YOLOv4. The results were then compared.

4.2.1.1 The Computer Vision Approach

With this approach, the ALPR algorithm is written in Python and uses mainly open cv and tesseract Ocr library. The following steps were used in the process of making the ALPR program:

- License plate detection and localization from an image
- Extraction of characters from the license plate
- Applying Tesseract OCR library to recognize the segmented characters

4.2.1.1.1 License plate detection And Extraction of Characters

First of all, scikit-learn was used to clear the borders of the image. A minimum and maximum aspect ratio are set as parameters. The minimum and maximum aspect ratios are were used to detect and filter out a rectangular number plate. The rectangular dimensions of the license plate correspond to these ratios.

Ghanaian license plates have a lighter background with a lighter foreground (where characters can be found) as yellow and white have a higher pixel value of (255,255,0) and (255,255,255) respectively compare to black (0,0,0), making them lighter [17]. However, a white Ghanaian plate was used. To reveal the characters from the light background , a blackckhat morphological operation was done.

Figure 3 Blackhat morpholoical operation brings out license plate numbers

```
rectKern = cv2.getStructuringElement(cv2.MORPH_RECT, (13, 5))
blackhat = cv2.morphologyEx(gray, cv2.MORPH_BLACKHAT, rectKern)
```

Figure 4 code for blackhat

As seen in the above figure 3, background noise is cleared, making the plate numbers visible against the darker background. Next, portions of the image which are light and could contain characters are searched and found. A square kernel was used to identify larger structures present in the image by filling up tiny holes. Using Otsu's method, a binary threshold was performed to disclose the lighter portions of the image, a character. Thresholding is used to convert grayscale images to binary where the pixels are either black or white (0 or 255). A simple thresholding example would be selecting a pixel value T and then setting all pixel intensities less than T to zero and all pixel values greater than T to 255[17]. This provided a binary representation of the image. What makes Otsu's method of Thresholding different is it attempts to find an optimal value of T to separate the two-pixel intensities.

Figure 5 Results for Otsu's thresholding Method

After using Otsu's method, the license plate regions become a large white surface.

```
squareKern = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
light = cv2.morphologyEx(gray, cv2.MORPH_CLOSE, squareKern)
light = cv2.threshold(light, 0, 255,
    cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
```

Figure 6 code for figure 5

Using Scharr gradient, the edges in the image are detected while the characters' boundaries are highlighted. Computation of the Scharr gradient magnitude in a direction x, of the image, run through blackhat . After, the scale of the resulting intensities are returned to a range of (0,255) .

```
gradX = cv2.Sobel(blackhat, ddepth=cv2.CV_32F,
    dx=1, dy=0, ksize=-1)
gradX = np.absolute(gradX)
(minVal, maxVal) = (np.min(gradX), np.max(gradX))
gradX = 255 * ((gradX - minVal) / (maxVal - minVal))
gradX = gradX.astype("uint8")
```

Figure 7 code for scharr's gradient

Figure 8 After applying Scharr's gradient

The edges in our blackhat operation image become more emphasized after applying Scharr's gradient. The license plate characters also appear observably variant from the rest of the image.

Sorting out regions in the image that could be a boundary for characters in the license plate was the next step required. To achieve this, Gaussian blur was done. While blurring an image may seem undesired, it is a valuable tool to smoothen an image. Regular blurring involves defining a sliding window on top of the image with sides k * k, where k is an odd number. The window slides from horizontally from left to right and vertically from top to bottom. At the center of the k matrix, a pixel value is set to be the average of all other pixels[18] Gaussian blur differs here by making use of a weighted mean, where pixels that are nearer to the central pixel contribute to the "weight" of the average [19].

Figure 9 Results after applying Gaussian blur

```
gradX = cv2.GaussianBlur(gradX, (5, 5), 0)
gradX = cv2.morphologyEx(gradX, cv2.MORPH_CLOSE, rectKern)
thresh = cv2.threshold(gradX, 0, 255,
    cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
```

Figure 10 code for Gaussian blur

Initially, it seemed like the image has been cluttered but the same region of the license plate bounding box is clearly defined. To clean up some of the noise in figure 9, an erosion operation was performed.

```
thresh = cv2.bitwise_and(thresh, thresh, mask=light)
thresh = cv2.dilate(thresh, None, iterations=2)
thresh = cv2.erode(thresh, None, iterations=1)
```

Figure 11 code for erosion

Figure 12 After Applying erosion operation to figure 6

The contours belonging to the license plate is not the largest white region. It is probably the second largest or the third largest. Also, it is not touching the outward edges of the image. With this, some sorting was done based on the size of contours in the image, maintaining only the large ones. Next, the mostly likely contour to contain the number plate was found by looping over the potential contours that could contain the license plate

against the input image. The region of interest containing the license plate is extracted while determining the license plate's rectangular contours. In order to the region extracted is the fine rectangular shape, the aspect ratios of the rectangle was computed.

4.2.1.1.2 Character Recognition with Tesseract

To improve results obtained from Tesseract OCR, any pixels of the foreground(where the charatcers are) that touch the edges or bounding box of the license plate is cleared. There are several Tesseract options, but what was used is the "treat image as a single text line" option (page segmentation method, a mode in Tessearct). From the Tesseract list of characters , we compile into a string, the OCR'd text from the image.

```
alphanumeric = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
options = "-c tessedit_char_whitelist={}".format(alphanumeric)

# set the PSM mode
options += " --psm {}".format(psm)

return options
```

Figure 13 configuration for Tesseract

The result is displayed to a screen where the OpenCV draw function is used to draw a rectangle around the license plate once contours are sorted. Also, a text of the extracted characters is displayed on top of the bounding box.

```
box = cv2.boxPoints(cv2.minAreaRect(lpCnt))
box = box.astype("int")
cv2.drawContours(image, [box], -1, (0, 255, 0), 2)
```

Figure 14 draw rectangle code

```
(x, y, w, h) = cv2.boundingRect(lpCnt)
cv2.putText(image, cleanup_text(lpText), (x, y - 15),
    cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 255, 0), 2)
```

Figure 15 write text code

Figure 16 Result after running the ALPR algorithm

4.2.1.2 The YOLOv4 approach

4.2.1.2.1 Dataset Gathering

The dataset used to train the YOLOv4 model was obtained from Google's open images data. A collection of 1500 images were obtained. These images contained different types of vehicles with license plates at different resolutions ranging from 640×480 pixels to a higher resolution of about 1024×522 pixels. Also, some of the vehicle plates in the images were tilted at several angles. A custom dataset based on Ghanaian plates was not

created because there was a problem of leaking people's license plates without their permission, which is a breach of rights. After, validation of the dataset was done to create labels(bounding boxes on license plates) which are converted into YOLOv4 format for training.

4.2.1.2.2 Splitting Dataset

The dataset was split into two parts, a training set, and a testing set. The YOLOv4 detector used the training set to learn what each plate looks like by making predictions on the input data and then correcting itself when predictions are wrong. The testing dataset was used to evaluate the performance of YOLOv4 after training.

4.2.1.2.3 Training the YOLOv4 model

The custom dataset with the YOLOv4 annotations were uploaded to a google collab notebook. The notebook provides a free environment with GPU for training deep learning algorithms such as YOLO.

```
!./darknet detector train data/obj.data cfg/yolov4-obj.cfg /mydrive/yolov4/backup/yolov4-obj_last.weights -dont_
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 139 Avg (IOU: 0.276707, GIOU: 0.223270), Class: 0.395642
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 150 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 161 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000
 total_bbox = 2832, rewritten_bbox = 0.035311 %

 Tensor Cores are disabled until the first 3000 iterations are reached.

 209: 2.029309, 2.256250 avg loss, 0.000002 rate, 6.098144 seconds, 13376 images, 9.544629 hours left
Loaded: 0.000036 seconds
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 139 Avg (IOU: 0.356682, GIOU: 0.241047), Class: 0.520622
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 150 Avg (IOU: 0.296364, GIOU: 0.178781), Class: 0.502531
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 161 Avg (IOU: 0.257917, GIOU: 0.183860), Class: 0.447824
 total_bbox = 2863, rewritten_bbox = 0.034928 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 139 Avg (IOU: 0.347835, GIOU: 0.222600), Class: 0.511115
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 150 Avg (IOU: 0.349900, GIOU: 0.276677), Class: 0.569027
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 161 Avg (IOU: 0.457595, GIOU: 0.418805), Class: 0.479964
 total_bbox = 2892, rewritten_bbox = 0.034578 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 139 Avg (IOU: 0.213389, GIOU: 0.035141), Class: 0.444126
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 150 Avg (IOU: 0.427256, GIOU: 0.373092), Class: 0.596059
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 161 Avg (IOU: 0.327854, GIOU: 0.195858), Class: 0.449307
 total_bbox = 2910, rewritten_bbox = 0.034364 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 139 Avg (IOU: 0.318174, GIOU: 0.234684), Class: 0.478360
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 150 Avg (IOU: 0.308447, GIOU: 0.123223), Class: 0.575418
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 161 Avg (IOU: 0.370564, GIOU: 0.309438), Class: 0.526972
 total_bbox = 2941, rewritten_bbox = 0.034002 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 139 Avg (IOU: 0.383823, GIOU: 0.342873), Class: 0.494629
```

Figure 17 Training the object detector

The image above shows how the objector detector is being trained on the set of 1500 images. It also shows the average loss in training per time after 3000 iterations.

4.2.1.2.4 Checking the Mean Average Precision(mAP)

A mAP is a machine learning method used to compute accuracy per class and across all data classes. To calculate the mean average precision, the average IoU(a ratio of intersection area to union area)for all N classes is computed. The average of these N averages is taken; hence the term mean average precision. This method is used to evaluate the accuracy of object detectors such as SSDs, Faster RCNN, and YOLO. Figure 10 shows the mAP obtained after training the YOLO model.

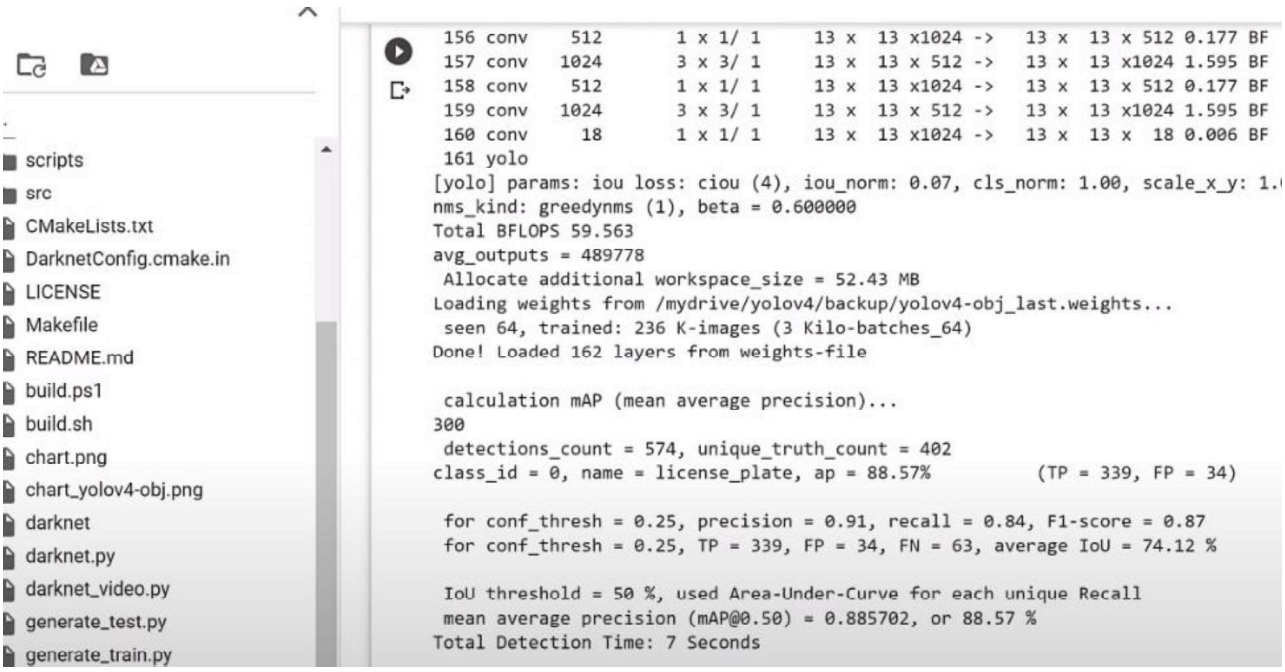

Figure 18 YOLOv4 obtained 88.57% mAP

According to [5] mAP > 0.5 is good.

Figure 18 shows the average training loss against the number of iterations on the 1500 images during training. Training loss decreases as training is started, and it projects towards an overage of 0.25% as the number of YOLOv4 weights or iterations increases (can be seen as the blue curve in the graph). The brown curve also represents the mAP, and at about 4000 iterations, the accuracy does not increase any further from 91%. This is due to over-fitting. Over-fitting is when objects on images can be detected from training-dataset but can't detect any other images.

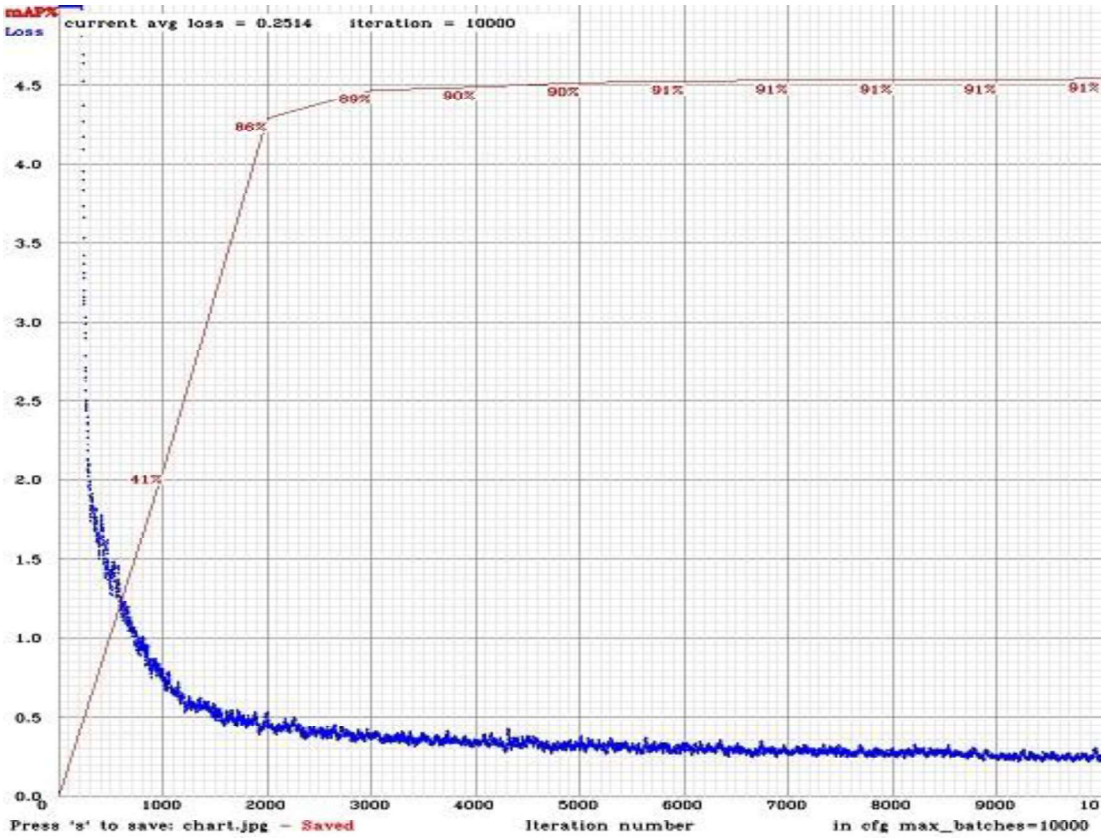

Figure 19 A graph of test loss against number of iterations

After successfully training the license plate detector, Tesseract was used to perform an OCR to retrieve the characters in the plate. The YOLOv4 model became a success with the help of [20].

4.2.2 The Database

The database consists of two tables. One is the registered plates with the owner's details, and the other, plates extracted by the ALPR algorithm. The number plates were compared in database to unveal the details of the violator where the violator's email is taken, and a notification of fine was sent. Only one administrator has access to the database.

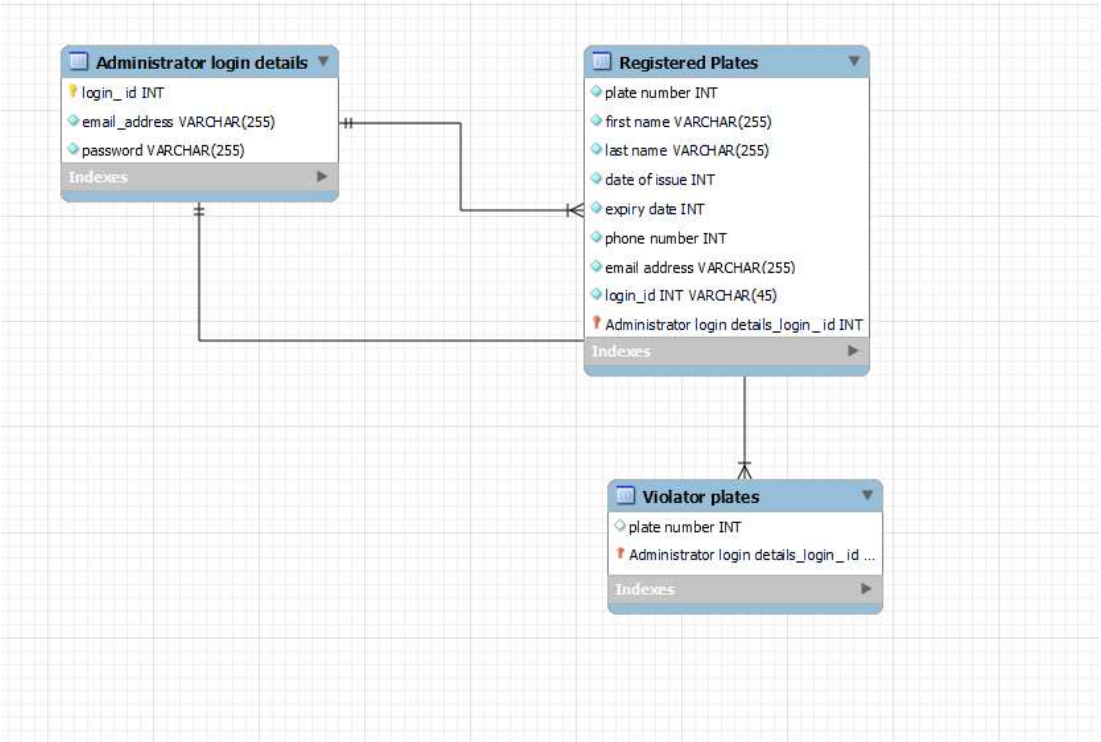

Figure 20 entity relation diagram of database

4.3 Testing The ALPR Algorithm

Testing the ALPR algorithm for two approaches was done by taking images of a vehicle to increase steps of 1 meter from zero to oblique angles from the license plate. After every one meter, an image is captured and processed on the ALPR algorithm. This was done under two settings : at 4 pm with Partly cloudy weather conditions and 7 pm Passing clouds conditions. The testing was done on only the white license plates as the yellow ones were out of reach.

4.4 Results

The objectives of this project stated in chapter 1 have been achieved. These objectives are

- The ALPR should be able to extract license plate details under the constraints of a cheap camera (to save cost)
- The ALPR should be able to detect license plate at oblique angles between the camera and the license plate
- The ALPR should be able to detect license plates at poor lighting conditions and night time
- The System should be able to notify violators of their offense with a fine

Using both the computer vision and YOLOv4 approach, the above objectives have been satisfied. The results of these can be found in the tables below. The computer vision approach was able to obtain about a 71% ALPR accuracy at 3 meters and 60 degrees inclination between the plate and camera at 7pm night time. The YOLOv4 on the other hand did significantly well with a 99% ALPR accuracy at 6 meters and 60 degrees inclination between the plate at 7pm night time. In the tables below, 99% was used to represent success in detecting all characters correctly, where percentages below 99% had some of the characters recognized. 0% represents inability to detect the license plate region at all.

Table 1 : 4pm daytime with Partly cloudy weather condition at zero degrees from white plate

Computer Vision Approah		YOLOv4	
Distance from Vehicle/meters	Number Plate Recognition success(%)	Distance from Vehicle/meters	Number Plate Recognition success(%)
1	99	1	99

2	99	2	99
3	99	3	99
4	99	4	99
5	99	5	99
6	99	6	99
7	0	7	99
8	0	8	99

Table 2 : 7pm nighttime with Passing clouds weather condition at zero degrees from white plate

Computer Vision Approah		YOLOv4	
Distance from Vehicle/meters	Number Plate Recognition success(%)	Distance from Vehicle/meters	Number Plate Recognition success(%)
1	99	1	99
2	99	2	99
3	99	3	99
4	99	4	99
5	85	5	99
6	71	6	99
7	0	7	99
8	0	8	99

Table 3 : 4pm daytime with Passing clouds weather condition at 30 degrees from white plate

Computer Vision Approah		YOLOv4	
Distance from Vehicle/meters	Number Plate Recognition success(%)	Distance from Vehicle/meters	Number Plate Recognition success(%)
1	99	1	99
2	99	2	99
3	99	3	99
4	71	4	99
5	71	5	99
6	0	6	99
7	0	7	0
8	0	8	0

Table 4 : 7pm nightime with Passing clouds weather condition at 30 degrees from white plate

Computer Vision Approah		YOLOv4	
Distance from Vehicle/meters	Number Plate Recognition success(%)	Distance from Vehicle/meters	Number Plate Recognition success(%)
1	99	1	99
2	99	2	99
3	99	3	99
4	71	4	99
5	0	5	99
6	0	6	99
7	0	7	0

8	0	8	0

Table 5 : 4pm daytime with Partly cloudy weather condition at 60 degrees from white plate

Computer Vision Approah		YOLOv4	
Distance from Vehicle/meters	Number Plate Recognition success(%)	Distance from Vehicle/meters	Number Plate Recognition success(%)
1	99	1	99
2	99	2	99
3	99	3	99
4	0	4	99
5	0	5	99
6	0	6	99
7	0	7	0
8	0	8	0

Table 5 : 7pm nighttime with Passing clouds weather condition at 60 degrees from white plate

Computer Vision Approah		YOLOv4	
Distance from Vehicle/meters	Number Plate Recognition success(%)	Distance from Vehicle/meters	Number Plate Recognition success(%)
1	99	1	99
2	99	2	99

3	99	3	99
4	0	4	99
5	0	5	99
6	0	6	0
7	0	7	0
8	0	8	0

Chapter 5 : Discussion & Conclusion

5.1 Limitations

Although some promising results have been achieved there is still more room for accuracy improvement as illumination and viewpoint variation still affect the success of an ALPR. Also, the ALPR is limited to vehicle speeds as testing for various vehicle speeds was not done. The ALPR is also limited to detecting only one plate at a time. Yellow license plates were not tested because of not having access to datasets for Ghanaian license plates.

5.2 Future work

Based on the discussed limitations, more training of the YOLOv4 can improve the accuracy of the ALPR. Also, YOLOv4 can be trained to detect multiple license plates in a single frame. The ALPR using YOLOv4 can be made into a real-time algorithm that can detect any plate if it is adjusted for multiple detections.

5.3 Conclusion

Using computer vision algorithms to implement an ALPR, good results were still obtained, saving from buying a GPU for advanced deep learning algorithms. However, with the help of the google collab notebook, which gave a free GPU for training, a more accurate ALPR was obtained by using YOLOv4. Hence, even though the cost of a camera for an ALPR system is expensive, the cost has been saved by building an ALPR that is still accurate with a cheap camera with lower resolutions. Even though vehicles have changed the lives of people, it causes road accidents. Several measures have been taken to reduce road accidents in Ghana, yet failed. Technologies such as ALPR can help eliminate

road accidents; however they are expensive due to the cost of cameras. This project has implemented an ALPR that is accurate under the constraints of a cheap, low-quality camera using YOLOv4 and computer vision algorithms such as guassian blur, Otsu binary thresholding, morphological operations, and Tesseract OCR.

References

[1] W. H. Organization, *Global Status Report on Road Safety 2015*. World Health Organization, 2015.

[2] F. Afukaar, P. Antwi, and S. Ofosu-Amaah, "Pattern of Road Traffic Injuries in Ghana: Implications for Control," *Inj. Control Saf. Promot.*, vol. 10, pp. 69–76, Apr. 2003, doi: 10.1076/icsp.10.1.69.14107.

[3] N. Farajian and M. Rahimi, "Algorithms for licenseplate detection: A survey," in *2014 International Congress on Technology, Communication and Knowledge (ICTCK)*, Nov. 2014, pp. 1–8, doi: 10.1109/ICTCK.2014.7033529.

[4] "In Arizona, an automatic license plate recognition (ALPR) system has an estimated benefit-to-cost ratio of 9.6 due to improved vehicle registration and insurance compliance. | U.S. Department of Transportation." https://www.itskrs.its.dot.gov/its/benecost.nsf/ID/20746487b947b3358525797c006528cf (accessed Apr. 21, 2021).

[5] A. Rosebrock, *Deep Learning for Computer Vision with Python*, 1st edition. PyImageSearch.com, 2017.

[6] M. Bajammal, A. Stocco, D. Mazinanian, and A. Mesbah, "A Survey on the Use of Computer Vision to Improve Software Engineering Tasks," *IEEE Trans. Softw. Eng.*, pp. 1–1, 2020, doi: 10.1109/TSE.2020.3032986.

[7] J.-Y. Sung, S.-B. Yu, and S. P. Korea, "Real-time Automatic License Plate Recognition System using YOLOv4," in *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, Nov. 2020, pp. 1–3, doi: 10.1109/ICCE-Asia49877.2020.9277050.

[8] Y. Lu, L. Zhang, and W. Xie, "YOLO-compact: An Efficient YOLO Network for Single Category Real-time Object Detection," in *2020 Chinese Control And Decision Conference (CCDC)*, Aug. 2020, pp. 1931–1936, doi: 10.1109/CCDC49329.2020.9164580.

[9] J. Wang, X. Liu, A. Liu, and J. Xiao, "A deep learning-based method for vehicle licenseplate recognition in natural scene," *APSIPA Trans. Signal Inf. Process.*, vol. 8, ed 2019, doi: 10.1017/ATSIP.2019.8.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," pp. 436–444, 2015.

[11] A. Kashyap, B. Suresh, A. Patil, S. Sharma, and A. Jaiswal, "Automatic Number Plate Recognition," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Oct. 2018, pp. 838–843, doi: 10.1109/ICACCCN.2018.8748287.

[12] O. Khin, M. Phothisonothai, and S. Choomchuay, "Myanmar character extraction from vehicle images using aspect ratio and bounding box," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Chiang Mai, Jan. 2018, pp. 1–4, doi: 10.1109/IWAIT.2018.8369714.

[13] C.-H. Lin, Y.-S. Lin, and W.-C. Liu, "An efficient license plate recognition system using convolution neural networks," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, Apr. 2018, pp. 224–227, doi: 10.1109/ICASI.2018.8394573.

[14] C.-H. Lin and Y. Li, "A License Plate Recognition System for Severe Tilt Angles Using Mask R-CNN," in *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*, Aug. 2019, pp. 229–234, doi: 10.1109/ICAMechS.2019.8861691.

[15] S. Choyal and A. K. Singh, “An Acoustic based Roadside Symbols Detection and Identification using Faster RCNN and SSD,” in *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, Feb. 2020, pp. 1–4, doi: 10.1109/ICONC345789.2020.9117222.
[16] A. Menon and B. Omman, “Detection and Recognition of Multiple License Plate From Still Images,” in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, Dec. 2018, pp. 1–5, doi: 10.1109/ICCSDET.2018.8821138.
[17] A. Rosebrock, *Practical Python and OpenCV*. PyImageSearch.com, 2016.
[18] A. Das, A. Medhi, R. K. Karsh, and R. H. Laskar, “Image splicing detection using Gaussian or defocus blur,” in *2016 International Conference on Communication and Signal Processing (ICCSP)*, Apr. 2016, pp. 1237–1241, doi: 10.1109/ICCSP.2016.7754350.
[19] E. S. Gedraite and M. Hadad, “Investigation on the effect of a Gaussian Blur in image filtering and segmentation,” in *Proceedings ELMAR-2011*, Sep. 2011, pp. 393–396.
[20] T. A. Guy, *theAIGuysCode/yolov4-custom-functions*. 2021.