MAKERERE UNIVERSITY

COLLEGE OF ENGINEERING, DESIGN ART AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

DESIGN OF A DETECTION AND ALERT SYSTEM FOR ANIMAL ROAD CROSSINGS (ELEPHANTS) IN ELECTRIC VEHICLES USING COMPUTER VISION

By

RONALD SEKASI

18/U/497

Main Supervisor

Mr. Derrick Sebbaale

Co- Supervisor

Ms. Agatha Turyagyenda

A Final Project Report submitted to Makerere University in partial fulfillment of the requirements for the award of BSc in Electrical engineering.

SEPTEMBER 2022

DECLARATION

I declare to the best of my knowledge that this is my original work and has never been presented to any university or academic institution for any academic award.

Signature: ....

Date: 23/09/2022

APROVAL

This Project Report has been submitted with the approval of the following supervisors.

MAIN SUPERVISOR:

Name: MR. DERRICK SEBBAALE

School of Engineering, College of Engineering, Design, Art and Design

Makerere University

Signature.. Date ....23/9/2022....

CO-SUPERVISOR:

Name: Ms. AGATHA TURYAGYENDA

School of Engineering, College of Engineering, Design, Art and Design

Makerere University

Signature .... Date ....23-09-22....

DEDICATION

I dedicate this project report to my beloved parents for the love and support they have provided to me throughout this project period and financial support they rendered to me during the research period.

I also dedicate it to my project supervisors Mr. Derrick Sebbaale and Ms. Agatha Turyagyenda for their unmatched effort and guidance about my project report and the moral support they offered to me during my research period. May the almighty God bless all abundantly.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Mr. Derrick Sebbaale and Ms. Agatha Turyagyenda for their assistance with my project report. I sincerely appreciate the help you provided to me.

Also, I acknowledge all the other department lecturers who have always given me time for consultation regardless of whether they are my supervisors or not, thank you for the helping attitude

My family deserves a special thank you for their unwavering financial and moral support. God willing richly reward them.

Above everything, I thank the Almighty Allah for giving me life, wisdom, and direction throughout this period.

ABSTRACT

Detection and alert systems in electric vehicles inform the driver of impending collisions with other objects along the host vehicle path and similarly electric vehicles as powered by batteries for propulsion thus minimal clean transportation as compared to internal combustion engines. Since Uganda's road network traverses most game parks/reserves, tourism subjects the country's road users to risks associated with animal road crossings. In this project, we developed a low cost and efficient detection and alert system based on a YOLOv4 object detector and a camera for forward vehicle-animal (elephant) collision detection. We trained the object detector on an African elephant dataset and an elephant sign post open dataset. We then implemented a distance, speed and direction estimation algorithm in python based on the triangular similarity using a single reference image with known distances to determine the distances of the new objects detected. Our results for the object detector were based on the @mAP scores of 93.56% and 93.4% for the elephant dataset and elephant signpost dataset respectively evidencing good system performance and accuracy. We had a detection range of 20m from the camera giving accurate distance measurements for first moving frames and also demonstrated implementation of our detection and alert system with a single sensor and processor (Raspberry pi 4 module) making it relatively cheaper and compatible with electric vehicle designs in Uganda.

LIST OF ABBREVIATIONS

ATC	Air Traffic Control
ACC	Adaptive Cruise Control
BEV	Battery Electric Vehicle
CSP	Cross-stage Partial Connection
CA	Collision Avoidance
CAS	Collision Avoidance System
CM	Collision Mitigation
CV	Computer Vision
CNN	Convolution Neural Network
CBN	Cross Batch Normalization
CmBN	Cross-iteration mini-Batch Normalization
EV	Electric vehicle
FPN	Feature Pyramid Network
ICE	Internal Combustion Engine
IOU	Intersection Over Union
LIDar	Light detection and Ranging
mAP	mean Average Precision
PHEV	Plugin Hybrid Electric Vehicle
R-CNN	Region based-CNN
RPN	Region Proposal Network
ResNet	Resolution Network
RoI	Region of Interest
Radar	Radio detection and ranging
SVM	Support Vector Machine
SVD	Singular Value Decomposition
SPP	Spatial Pyramid Pooling
SAT	Self-Adversarial Training
UN	United Nations
WHO	World Health Organization
YOLO	You Look Only Once

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 : INTRODUCTION

1.1 BACKGROUND

An electric vehicle is an automobile fully or partially supplied by electric power, propelled by one or more electric motors. The car battery is purely the energy source ie. Lithium-ion battery, with great longevity and low discharge rate of about 5% per month. (Excellent energy retention). These are basically categorized into battery electric vehicles (BEVs) and plugin hybrid electric vehicles (PHEVs). BEVs use purely and only an electric grid rechargeable battery as the source of energy for propulsion, these vehicles are silent (no moving parts) so low maintenance costs and do not cause noise and greenhouse gas emissions since no fossil fuels exploited. PHEVs use both a rechargeable battery and fossil fuel power (diesel or petrol) making them more costly in terms of maintenance, noisy (moving parts) and pollutive. Nevertheless, PHEVs are preferable over BEVs for long distances since can be operated on fossil fuel backup power before reaching a nearby charging station, thus more reliability[1].

The first electric vehicle was built around 1832 by Robert Anderson, followed by later inventions and modifications in the late 19th and early 20th centuries with electricity majorly for propulsion. These electric cars remained popular until advances in the internal combustion engine (ICE) cars that were preferred due to their cheaper production costs and quicker refueling times. In 1912, introduction of the electric motor starter easy to start reignited the fleet of electric cars. This was realized by the development of the all-electric "Tesla roadster" of 2004 by Tesla Motors in the USA and starting 2008, the resseniance in electric vehicle manufacturing was due to advances in batteries, desire to curb greenhouse gas emissions and improve urban quality[2] Later in 2011, the first electric vehicle (EV) was developed and launched in Africa for the Kiira EV project in Uganda and this sparked off motivation towards electric vehicle development in Uganda and Africa at large.

With the long-term target of reducing air pollution in Kampala, Uganda being one of the most polluted cities in East Africa and the world, electric vehicle development has been greatly dominated by state-owned manufacturers, KIIRA motors which was allocated over 1000 acres of land for the Jinja plant and started production in 2021 expected to produce a total of 5000 electric buses and other EVs per year. This EV development is supported by the Chinese manufacturer CHTC motor but with 90% of the required components locally manufactured such as Ugandan steel, lithium and copper to manufacture electric buses for public and private transport companies[3]

Despite the ongoing electric vehicle development, road safety is still a major challenge with 17443 road accidents reported in 2021, a 42% increase from the 12249 reported in 2020. This also indicated 3757 fatalities, 9070 serious injuries and 4616 minor injuries, which clearly indicates massive loss of lives and destruction of property. This calls for urgency of structuring relevant road safety measures such as Vehicle Collision Avoidance Systems (CAS) to curb the impact of these road accidents[4]

Since most traffic accidents are due to human error resulting from fatigue, in attentiveness and bad poor driving practices, driver support systems must be put in place to curb the impact of these road traffic accidents otherwise massive losses are to be encountered in the long run. Driver support systems such as vehicle detection systems for collision avoidance (CA) are automobile systems that monitor the environment surrounding the host vehicle using sensors to collect data required for processing on processors thus providing information for guided intervention to avoid possible collisions. Some of these systems include;

- Lane departure system: This monitors lane markings and estimates the vehicle's current position within the lane so should the vehicle swerve out of the lane, the system issues a warning to the driver to react accordingly.
- Forward collision warning system: This is installed to avoid rear end collisions with other vehicles since the system issues a warning to the driver should the vehicle in front slow down lower than the set minimum value or when the possibility of collision increases.
- Pedestrian detection system: This detects pedestrians and possibility of collision with them in the surrounding vehicle environment and issues a warning for safe driver maneuvering to avoid possible collisions.

1.2 PROBLEM

According to the World Health Organization (WHO), road traffic accidents are prospected to be the fifth major leading cause of mortalities by 2030, with about 2% of the total road traffic accidents being animal related as per the traffic police report of 2020 in Uganda though less attention has been paid to the matter. Similarly, tourism as an economic activity subjects the country's road users to risks associated with animal road crossings since the country's road network traverses most of its game parks and reserves, so as electric vehicles are further being developed in Uganda, the impact of animal road crossings (elephants) on road safety should be greatly put into consideration.

1.3 JUSTIFICATION

As per SDG 15: Life on land which aims at protecting sustainable use of terrestrial ecosystems and halting biodiversity loss, degradation in the population size of these animal species (elephants) due to vehicle-animal collisions in the long run is undesirable since they underlie the ecosystem.

Similarly, tourism provides a large foreign exchange base for Uganda's economy so with poaching already being a major cause of reduction in elephant species, further reduction in numbers due to vehicle-animal collisions is greatly forbidden.

With Uganda being an active member of the United Nations (UN), the UN General Assembly adopted a resolution ARES/74/299 for (2021- 2030) of preventing at least 50% the number of road traffic deaths and injuries. Therefore, Uganda is obliged to contribute towards this cause by improvising solutions to curbing on the number of road traffic accidents.

1.4OBJECTIVES

Main Objective

To develop a system that can detect and alert the driver on possible collisions with elephants on roads.

Specific Objectives

- To develop an elephant and elephant signposts detection model to aid and assist in generation of alerts to the driver.
- To integrate a distance, speed and direction estimation algorithm with the object detection model.
- To deploy the system on a processor and evaluate its performance.

1.5 PROJECT SCOPE

This report comprises of five chapters namely introduction, literature review, methodology, results and conclusions with recommendations.

Chapter 1: This is the introduction which comprises of the project background, problem, justification, project objectives and project scope.

Chapter 2: This is the literature review that comprises of theory about collision avoidance systems, model architecture and processors.

Chapter 3: This is methodology the comprises of the project objectives and techniques necessary in achieving them.

Chapter 4: This is the results section which comprises of the achieved experimental results and the system performance evaluation.

Chapter 5: This are conclusions and recommendations which comprise of analysis and proof of the expected results, project limitations and what should be done to achieve an optimal system design

CHAPTER 2 : LITERATURE REVIEW

2.1 COLLISION AVOIDANCE

Collision avoidance is a very vital issue in most transportation systems and many more several applications such as detection and avoiding possible collision in Air traffic control (ATC), automotive collision avoidance and robot manipulator control. The major role of collision avoidance systems is to avoid two or more objects from colliding with each other and this achieved by issuing a warning to an operator (driver) or performance of an autonomous avoidance maneuver, an activity known as an intervention. For any collision avoidance system (CAS), the decision-making algorithm and the metric for measuring the collision threat might vary significantly depending on the type and application of the intervention.

With CA systems being used widely in various areas of application and under very many circumstances for this chapter, we shall mainly focus on the automotive collision avoidance.

2.1.1 Automotive Collision avoidance

Since traffic accidents are the one of the major and leading causes of deaths and injuries, automotive manufacturers need to put more emphasis on developing driver support systems so as to curb these accidents. Under CA systems, adaptive cruise control (ACC) available for most car models is a key starting step where there is adoption of the speed in any in-path vehicle should its speed reduce lower than the set speed of the host vehicle. These ACC systems exert a limited acceleration (-3m/square seconds) and some issue a warning to the driver for acceleration not sufficient to avoid collision so can be switched on and off by the driver and disengaged at low speeds of about 40km/hr.(comfort systems). The next step in automotive collision is sending a warning to the driver or performance of braking in case of a possible collision but the greatest challenge is accurately predicting the driving situation even for normal driving since several obstacles are detected and classified, making automotive collision avoidance complex.

2.1.2 Automotive safety

Despite progress of the existent crash worthiness of new and latest cars of a 90% reduction in the risk of a fatal injury as compared to older cars of the 80's, many people are still killed and injured in traffic accidents therefore more effort has been made to curb these injuries. This has been achieved to a certain degree by implementation of active safety systems such as the yaw control which is claimed to reduce the risk of fatal injury by up to 50% depending on the road condition.

Similarly, collision mitigation systems (CM) have also been very essential since they use sensors to observe the environment directly underlying the front of the host vehicle thus issuing information necessary for decision making in deploying countermeasures to avoid possible frontal collisions

Collision mitigation (CM) systems are referred to as collision avoidance systems (CAS) that are given some perception of the surrounding environment and take steps to avoid imminent collisions. Some of the actions performed to mitigate imminent collisions are issuing a warning to the driver, applying brakes and changing the course of the vehicle by applying a torque to the steering wheel and some of these CM systems include;

- Lane departure systems: These monitor lane markings and with these observations an estimate of the vehicle's position in the lane is obtained so should the vehicle swerve out of the lane; a warning is issued or a steering intervention is performed.

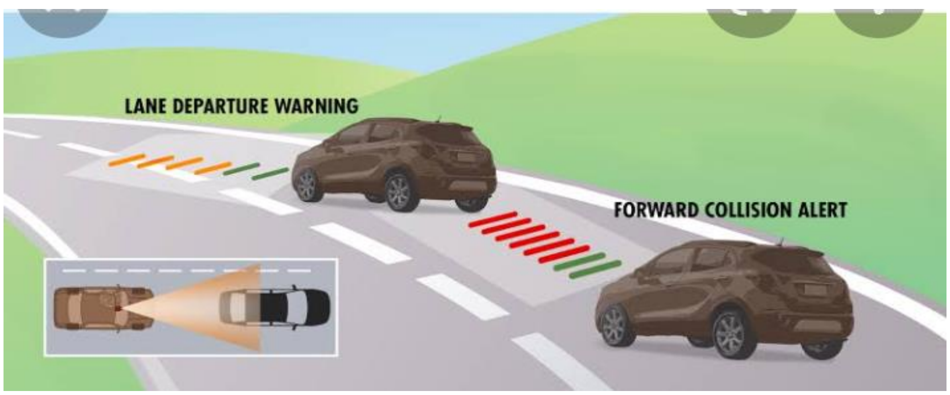

Figure 2-1: Lane departure system and forward collision system operations [5]

- Forward collision mitigation systems: These monitor what is in front of the host vehicle and intervene to prevent frontal collisions with obstacles.
- Lane change aid systems: These monitor the blind spot and some distance behind the car then issue a warning or intervene by applying a torque to the steering wheel when changing lane to avoid collision.
- Adaptive cruise control systems (ACC) earlier discussed under automotive collision avoidance

Essentially collision mitigation systems (CMS) utilize sensors for monitoring the environment surrounding the host vehicle so as to intervene in the event of preventing possible collisions. These sensors hold for different functional applications and these include;

2.1.2.1 Vision systems for obstacle recognition

Figure 2-2: Vision camera monitoring

These are one or more several cameras together with a microprocessor to perform image processing. They have similarities to the human eye capabilities since they operate in the visible light region. The performance of such systems depends on the size of the sensor, number of pixels, the optics and the dynamic range because of 3D information obstacles though the image processing may be very computationally demanding. The update frequency of most vision sensors is 25Hz and these sensors are categorized mainly into two types namely;

- Single camera systems: These use either a monochrome or color camera. They are used to monitor the lane markings in lane-keeping aid systems.
- Stereo camera systems - These provide a 3D image by combining the images from two (or more) cameras. In such a system, range can be measured through triangulation.

Advantages:

- High resolution: Commonly a pixel array with 640 × 480 pixels is used making it possible to measure spatial properties of the obstacle with good accuracy. Advanced target classification is also possible from the detailed images.

Disadvantages:

- Sensitivity to light conditions: Vision systems can suffer severe performance degradation in certain light conditions such as heavy rain, thick fog or wet roadway combined with backlight.
- Computational demands: The image processing algorithms are computationally intensive
- Sensitivity to dirt: Dirt deposited on/in front of the lens can distort operation of a vision sensor.

2.1.2.2 Infrared (IR) vision sensors

Figure 2-3: Night vision system operation in a car using an infrared sensor [7]

These are also known as thermal sensors that provide the same information as any normal camera so IR vision system properties are no different from the normal vision systems but differ due to their sensitive to wavelengths typical to heat radiation (regions for lidar operation). Recently, IR cameras have been applied to night vision systems that enhance the driver's perception capabilities. These sensors can detect large obstacles such as animals and human beings since they are most sensitive to wavelengths corresponding to the normal body temperature of human beings and animals. IR sensors could potentially with be useful in accurate angle measurements and object classification.

Despite the essence of these IR sensors, their relatively higher cost above normal vision sensors for visible light are a major disadvantage hindering their usage in automotive collision mitigation (CM) systems.

2.1.2.3 Radar sensors

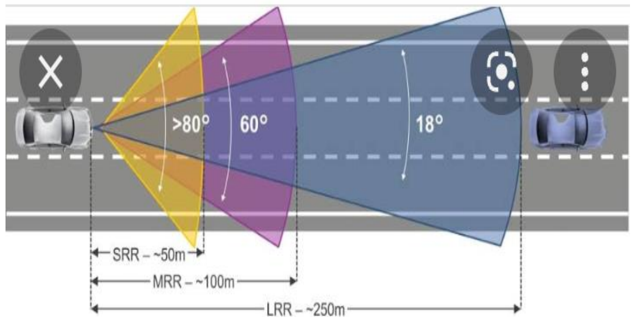

Figure2-4: Radar sensor operation in automotive collision avoidance vehicle systems [8]

Radar (Radio detection and ranging) sensors are the most common traditionally used type of tracking sensors in military applications over time and have been introduced in automotive adaptive cruise control (ACC) systems.

Radar sensors provide key measurements of range, range rate, azimuth angle and elevation angle. These sensors are active sensors since they emit electromagnetic radiations to illuminate their targets using the same antenna for sending and receiving signals thus constant switching between the sending and receiving mode with a particular waveform for the emitted energy. So, information on the environment is gathered by examining the echo of the transmitted signal. Millimeter radars are the longest automotive radars that use frequencies between 76-77 GHz whereas others use frequencies of 5GHz and 24GHz region and most available radars have an update frequency of 10Hz.

The radiation pattern shape depends on the antenna design whose main lobe has a conical shape for automotive radars so to provide azimuth angle information, the sensor either mechanically sweeps the antenna over a range or electronically switches between different emission angles. Similarly, techniques used to provide azimuth resolution could be the same for providing angle elevation information but however, the most existing automotive radars do not provide elevation measurements because scanning is only one-dimensional.

Advantages:

- Bad-weather performance: Radar sensors have the ability to detect objects in darkness, haze, snow and rain for the short distances that are required for automotive applications in the range normally less than 200m and automotive radar sensors are also insensitive to dirt deposits that is dirt from the road.

- Range and range rate: Unlike passive sensors, automotive radars provide accurate range measurements and some radars also measure the range rate using the Doppler effect.

Disadvantages:

- Resolution: The spatial resolution of the radar is poor because of the relatively wide lobe thus limited ability to measure spatial properties of the observed object.
- Elevation: It might be hard to discriminate obstacles from low overhead objects (road signs above the bridges, road tunnels) since automotive radars have no resolution in elevation and a wide beam.
- Clutter and spurious reflections: A number of objects in the environment reflect emitted radar signals. Unwanted reflections (often called clutter) from the road surface for instance, might give “ghost” obstacles (i.e., obstacles that do not exist). Multipath propagation might also cause such phenomena.

2.1.2.4 Lidar sensors

Figure2-5: Lidar sensor operation in vehicles for automotive collision avoidance [9]

Laser radar (lidar) sensors function in a similar way to the millimeter automotive radars and also provide measurements for the range, range rate, azimuth and elevation angles. Lidar sensors are active sensors that illuminate obstacles using laser diodes and operate in the infrared frequency region with wavelengths of 850nm slightly above visible light holding similar wave propagation properties as of visible light. These sensors are made of a distance (one dimensional) measurement device with a mechanical beam deflection system (rotating mirror) for spatial measurements and the distance is measured by observing the time of flight between transmitted and received signals. Lidar sensors like any other bare advantages and disadvantages.

Advantages:

- Clutter: Lidar sensors do not experience clutter and spurious reflections due to narrow beam and detection except a few with a wide beam and detection.
- Resolution: Since some lidar sensors measure at several hundreds of azimuth angles and several elevation angles in one sweep (100ms), provide a relatively large pixel array that could potentially give more detailed information of obstacles than millimeter radars
- Light sensitivity: Lidar sensors are insensitive to light conditions thus reliability.
- Gray scale: The photo detector can provide a gray scale picture that can be used to monitor lane markings since reflected intensity can be detected.
- Lidars are cheaper as compared to millimeter radars.

Disadvantages:

- Bad weather: Lidars experience performance degradation in hard rain, fog or snow.
- Dirt deposits: Altered reflectivity and problems in detection are possible to occur due to dirt deposits on the tracked vehicles since lidars are sensitive to dirt deposits on the lenses[5]

2.2 OBJECT DETECTION

Object detection is a computer technology related to computer vision (CV) and image processing that is associated with detecting the instances of a certain class of schematic objects such as humans, buildings and cars in digital images and videos. Object detection has been realized mostly in well researched domains such as pedestrian detection, scene text detection, face detection, edge detection, salient edge detection and multi categories detection thus object detection has been widely used in modern fields in the recent years such as military field, transportation field, medical field and security field for real world applications say monitoring security, transportation surveillance, autonomous driving, robotic vision, drone scene analysis and so on.

There are basically two categories of domain specific image object detectors which include the two stage object detectors such as fast R-CNN and one stage object detectors such as YOLO and SSD. Two stage object detectors have a high object recognition accuracy and high localization. The two stages of the two stage detectors can be divided by Region of Interest (RoI) pooling for example in fast R-CNN, the first stage is Regional Proposal Network (RPN) that proposes candidate object bounding boxes and for the second stage, features are extracted from each candidate bounding box for following classification and bounding box regression tasks by the Role Pooling (Rol Pool) operation. The basic architecture of two stage detectors is as illustrated in the figure below.

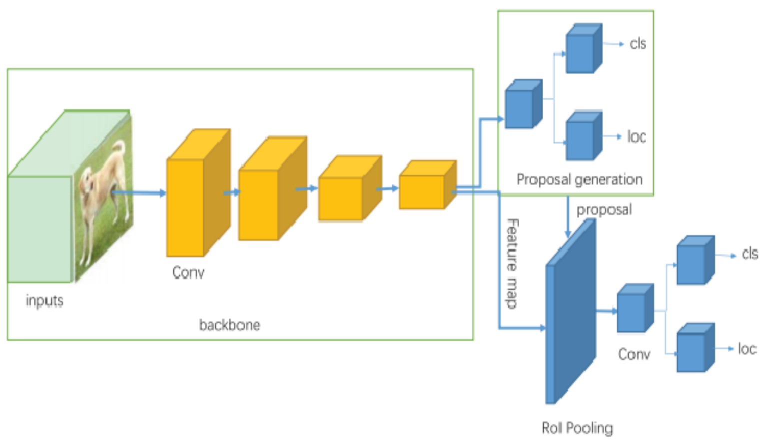

Figure2-6: The basic architecture design of two-stage object detectors [10]

The one stage object detectors propose predicted boxes directly from the input images without the regional proposal step. This makes such detectors time efficient and can be used for real time devices as illustrated by the basic architecture in the figure below;

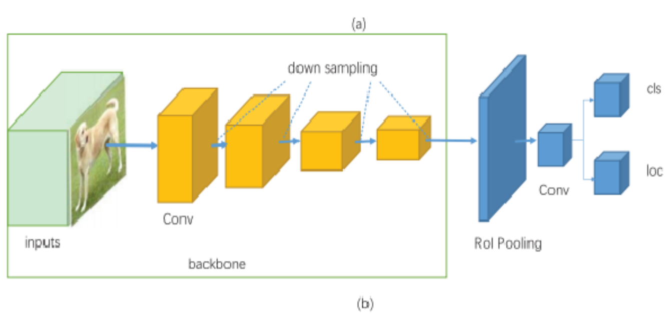

Figure2-7: The basic architecture design of one-stage detectors [10]

2.2.1 Two stage object detectors

2.2.1.1 R-CNN

This is a region-based CNN (Convolution Neural Network) and consists of four modules that is the first module generates category-independent regional proposals, the second module extracts a fixed length feature from each region proposal, the third module is a set of class-linear SVMs to classify the objects in one image and the last module is the bounding box regressor for precisely bounding box prediction.

According to authors, the selective search method is adopted for generating the regional proposals, a 4096-dimensional feature in each regional proposal is extracted by a CNN for region proposal features of the same size since the connected layer needs input vectors of fixed length. The authors adopt a fixed 227*227 pixel as the CNN input size, all wrapped around in a tight bounding-box despite the fact that objects in various images have different sizes or aspect ratio implying different sizes for the region proposals extracted by the first module. The feature extraction network consists of five convolution layers and two fully connected layers with all CNN parameters shared between all categories and each category trains category-independent SVM that does not share parameters among other different SVMs

NB: Training on a large dataset and fine tuning on a specific dataset is a good training practice.

For an Intersection over Union (IOU) overlap threshold of 0.5 in the process of fine tuning, region proposals are defined as positives above 0.5 and defined as negatives below 0.5 and also object proposals who's maximum IoU overlap with ground-truth class are assigned to the ground-truth box. In contrast, only ground-truth boxes are taken as positive examples in their respective classes and the object proposals have a 0.3 IOU overlap with all ground-truth instances of one class as a negative proposal for that class.

For proposals with a maximum overlap between 0.5 and 1 but are not ground-truth, expand the positive examples by approximately 30x, thus avoiding over fitting during the fine tuning effectively.

2.2.1.2 Fast R-CNN

This is a faster version of R-CNN which is a one stage end to end training process using multi task on each labeled RoI (Region of Interest) to jointly train the network. Since R-CNN takes a long time on SVMs classifications, fast R-CNN extracts features from an entire input image and then passes the region of interest (RoI) pooling layer to get the fixed size features as the input of the following classifications and bounding box regression fully connected layers. The features are extracted from the entire image at once and are sent for classification to CNN thus a large amount of time can be saved for CNN processing and large disk storage to store more features can also be saved as compared to R-CNN. Similarly, fast R-CNN uses RoI pooling layer to extract a fixed size feature map from different sized region proposals with no need of warping regions and reserves spatial information of features of region proposals so for fast detection, truncated SVD is used to accelerate the forward pass of computing fully connected layers.

2.2.1.3 Faster R-CNN

This is a faster version of fast R-CNN which improves the region-based CNN baseline. Faster R-CNN replaces the fast R-CNN with a novel RPN (region proposal Network) which is a fully convolutional network to predict region proposals effectively with a wide range of scales and aspect ratio. RPN accelerates the generating speed of region proposals because it shares a common set and fully-image convolutional layers with the detection network. Similarly, multi scale anchors are used as a reference in another method where the process of generating various sized region proposals with no need of multiple scales of input images or features.

On the feature maps (outputs) of the last shared convolutional layer sliding over a fixed size window(3x3), the center point of each feature window is relative to a point of the original input image that is center point to k (3x3) anchor boxes that have 3 different scales and 3 aspect ratios. The region proposal is given parameters relative to a reference anchor box and then the distance between the predicted box and its corresponding ground-truth box is measured to optimize the location of the predicted box therefore faster R-CNN has greatly improved both detection efficiency and precision.

2.2.1.4 Mask R-CNN

This is an extension of faster R-CNN and can be seen as a more accurate object detector mainly for segmentation task for instance.

This is achieved using a Faster R-CNN with Resnet-FPN (Feature pyramid network), a backbone which extracts RoI features from different levels of feature pyramid according to their scale and this achieves excellent processing speed and accuracy. FPN contains a top-down pathway which produces higher resolution features important for detecting small objects by up sampling coarser with semantically stronger feature maps from higher pyramid levels and bottom-up pathway is a backbone ConvNet which computes feature hierarchy comprising of feature maps at various scales at a scaling step of 2.

At the beginning, the top pyramid feature maps are captured by the output of the last convolutional layer of the bottom-up pathway and each lateral connection merges feature maps of the same spatial size of the bottom-up pathway and top-down pathway forming a new pyramid level and predictions are made independently at each level while the 1x1 convolutional layer can change the dimension when dimensions of the feature maps are different.

2.2.2 One stage object detectors

2.2.2.1 YOLO (you only look once)

This is a one stage object detector whose main contribution is real time detection of full images and webcam which predicts less than 100 bounding boxes per image. A unified architecture can extract features straightly from input images to predict bounding boxes and class probabilities since YOLO frames detection as a regression problem. YOLO network runs at 45 frames per second with no batch processing on a Titan X GPU. The YOLO pipeline first divides the input image into an S X S grid where a grid cell detects the object whose center falls in the grid and by multiplying twice parts the confidence score is obtained where People (object) denotes the probability of the bounding box in which contains an object and the Intersection over Union (IOU) which shows how accurate is the bounding box containing that object.

Each grid cell predicts B bounding-boxes and confidence scores for them and C-dimension conditional class probabilities for C categories. The feature extraction layer consists of 24 convolutional layers followed by 2 fully connected layers. The whole network is used for better performance for detection and doubling the input resolution of 224 x 224 in the pre training stage in order to get fine grained visual information to improve detection precision. Therefore, experiments showed that YOLO not good at localization due to large localization error as compared to fast R-CNN and faster R-CNN

2.2.2.2 YOLO V2

This is a second version of YOLO which adopts a series of design decisions from past works with novel concepts to improve YOLO's speed and precision. This is because of YOLO v2 improved performance characteristics such as;

- High resolution classifier: YOLO v2 adds a fine-tuning process to the classification network at 448 x 448 for 10 epochs on Image Net dataset to increase the mAP at 4%.
- Convolutional with anchor boxes: YOLO v2 adopts prediction boxes and firstly removes connected layers then predicts class and objectness of each anchor box thus a 7% increase in recall and a 0.3% drop in mAP
- Batch normalization: YOLO v2 adds ahead of each convolutional layer, a BN (Batch Normalization) layer which accelerates the network to gain convergence and regularize the model thus batch normalization gets 2% improvement in mAP.
- Predicting the size and aspect ratio of the anchor boxes using dimension clusters: YOLOv2 uses K means clustering on the training set bounding-boxes to achieve good priors automatically. Therefore, YOLO is improved by almost 5% over the above version with anchor boxes when dimension clusters are used directly along with predicting the bounding-box center location.
- Multi scale training: At high resolution detection, YOLO v2 achieves a higher mAP value at 40 frames per second as compared to YOLO at 45 frames per second for the training on the same backbone dataset. Similarly, YOLO v2 proposes a new classification backbone known as darknet 19 with 19 convolutional layers and 5 max pooling layers requiring fewer operations to process an image but maintaining a high accuracy and detecting precision due to the backbone and 7 main improvements.
- Fine grained features: YOLO v2 concatenates high resolution features with low resolution features by stacking adjacent features into different channels thus giving a 1% modest performance increase.

2.2.2.3 YOLO v3

This is an improved version of YOLO V2 that uses multi label classifications (independent logistic classifiers) to adapt to more complex datasets comprised of many overlapping labels. YOLO V3 uses three different feature maps to predict the bounding-box and the last convolutional layer predicts a 3D tensor encoding class predictions, objectness and bounding-box. This model proposes a deeper and robust feature extractor inspired by ResNet known as Darknet -53 and due to multi-label predictions advantage, YOLO V3 can better predict small sized objects even more as compared to medium and large sized objects with a worse performance [6]

2.2.2.4 YOLO V4

This is a one-stage object detector, an improved version of YOLO V3 that provides optimal speed and accuracy for object detection. YOLO V4 has a backbone of CSP darknet 53, SSP+PANet as the neck and YOLO V3 as the head. The model requires a higher input network size for detecting multiple small-sized objects and more parameters for greater model capacity to detect multiple objects of different sizes in the single image.

The YOLO V4 backbone utilizes mosaic data augmentation which allows detection of objects outside their normal context and reduces the need for large mini-batch size since batch normalization calculates activation statistics from four different images. Similarly, it also integrates CSP (Cross-Stage Partial Connection) which involves skip connections (like DesNet) such that the gradient can back propagate to the initial layers with relative ease because as the number of layers increases, the last layers hold lesser context of the features learned in the initial layers. The YOLO V4 detector also utilizes mosaic data augmentation and Self-Adversarial Training (SAT)data augmentation which helps the model become more generalized since it introduces the set amount of perturbation to the training data until the predicted label remains the same as the original class. Similarly, the detector also uses the CmBN (Cross-iteration mini–Batch Normalization) which introduces the idea of CBN (Cross batch Normalization) with mini-batch normalization in as single batch where the data is normalized with 4 batches in training by taking into account the mean and standard deviation and the batch (Training data) has no correlation with the previous batch and the batch coming after it.

The YOLO V4 neck that follows the backbone is used to increase the receptive field while separating out the most important features from the backbone, where an input image is taken and its feature map is extracted using convolutional layers after which a feature set is generated using a max pool of window size_1 and again using a max pool of window size_2. Therefore, by repeating the procedure n times, feature maps of different height and width dimension forming a pyramid will be obtained. YOLO V4 divides the feature along depth dimension applying SPP on each part and then combines it again to generate an output feature map. PANet is added to the SSP (Spatial Pyramid Pooling) to achieve the FPN (Feature Pyramid Network) [7]

2.3 RECENT WORK

J Chen et al proposed a LiDAR technology that can give a 360-degree surrounding of the car with high accuracy in turn providing real-time deer detection. The LiDAR sensors can work day and night without the influence of light conditions. J Chen et al proposed a new method for detecting deer from the data collected by the LiDAR sensor. The new deer detection algorithm contains three main parts which are background filtering, object clustering, and object classification where the background filtering stage eliminates all other objects from the image except the deer, based on 3D density static filtering (3D-DSF). Object clustering was intended for detecting the exact location of the deer by collecting all points belonging to the deer in one group which can accurately represent the deer and thus can be continually tracked to detect behavioral characteristics basing a DBSCAN clustering algorithm with Min Pts. The final stage involved the use of a classifier for three different targets i.e. deer, vehicles, and pedestrians. He also suggested there was room for improvement in the deer detection algorithm since LiDAR sensor has been adopted by some designers but they are limited by cost.

Schneider et al. used the modern deep learning object detection techniques, namely faster R-CNN and YOLO for detecting animals where he used camera-trap images from Gold Standard Snapshot Serengeti (GSSS) dataset and for feature extraction, ResNet101 architecture was used containing 101 convolutional layers. For the training weights of model trained on Common Object in Context 2017 dataset was used. The faster R-CNN and YOLO model were able to achieve an accuracy of 76.7% and 43.3%, respectively [5] [6].

CHAPTER 3 : METHODOLOGY

In this chapter, we clearly present how we implemented the project of designing a detection and alert system for elephant road crossings in electric vehicles during driving. This as a main objective was sequentially achieved by three specific objectives as stated below;

3.1 Main Objective

- To develop a system that can detect and alert the electric vehicle driver on possible collisions with elephants on roads.

3.2 Specific Objective

- To develop an elephant and road sign post detection model to aid and assist with generation of alerts to the driver
- To integrate a distance, speed and direction estimation algorithm with the object detection model
- To deploy the system on a processor and evaluate its performance.

We developed an elephant and elephant sign posts object detection model in a series of steps such as dataset creation, training the YOLOV 4 tiny model on the custom datasets.

3.3 System design and architecture

The system we designed was comprised of basically two stages namely;

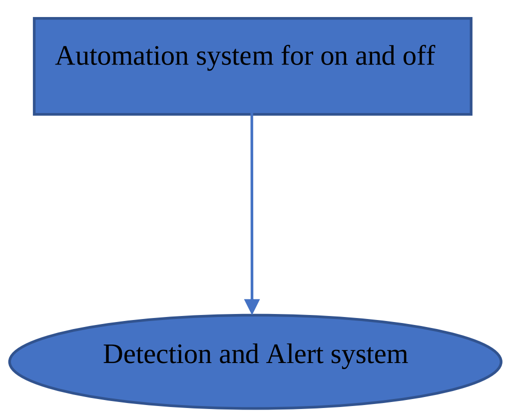

Figure3-1: A representation of our 2-stage system

3.3.1 Detection and alert system

The detection and alert system is to collect information from the surrounding environment during driving situations and detect any elephants are within the video frames.

For any animal detected in the frame, an elephant is tracked through the new frames for position location thereafter a distance and speed estimator algorithm use the bounding box placed on the elephant by the object detector to estimate its relative distance from the camera. With a threshold distance of 10m, an alert is generated to warn the driver of the elephant ahead of the host vehicle if the elephant exceeds the set distance.

3.3.2 Automation for system switching

With detailed review, we found to it that one of the major constraints to the adoption of collision avoidance systems is that with time their repetitive alerts become annoying and destructive with system surveillance added to keep switching them on and off when not needed. Therefore, in our system, we included the ability to automatically switch itself on and off without the intervention of the driver.

The next phase is therefore based on an elephant sign post detector which on detecting a sign post from the video captured by the camera, the system automatically turns on and if then the driver wishes, switching off of the system can be done manually.

3.4 Developing an elephant and an elephant sign post detection model.

This was done to generate a model that can accurately detect an elephant and elephant signpost through a series of steps including image data collection, image processing, model architecture specifications, training YOLO V4-tiny object detection model on our generated custom datasets and performance evaluation as described in this section.

3.4.1 DATASET CREATION (image data collection)

Like for any other machine learning model, the model performance increases with training on sufficiently large (usually thousands) of image data (datasets) which is usually very intensive since more requirements are needed in gathering and preparing the necessary image data for training. Some of these datasets are already created open source and local whereas some need to be created since they are not currently available.

Due to the number of requirements required to obtain local datasets from the relevant authorities we opted to using open data sets since African elephant data sets stored by Uganda wildlife authority (UWA) we're not readily available and sufficient for our intended purpose and similarly, there was no response for the emails addressed to the relevant parties in our attempts to access the datasets collected by UWA and therefore chose to do open data set collection.

Therefore, we collected our own elephants and elephant sign posts datasets from several online Google images of different sizes and resolution for either of the datasets in addition to the Kaggle elephant dataset, which were downloaded as zip files and saved in separate folders.

Table 3-1: A table showing the online image sets gathered

Image type	Overall collected images	Images after sorting
African elephants	428	328
Elephant signposts	310	238
Kaggle elephant dataset	373	373

Using roboflow (online site), a user friendly easy to use program for annotating and assigning specific classes to image data collected, we performed annotation (bounding box over the image region of interest) and labeling of different classes on either of the datasets with the elephants' dataset labeled as an 'elephant' class and the elephant sign posts as an 'ele-sign' class. The numbers of images collected from Google online were more than the annotated images used in creation of the actual datasets because of desire to minimize image repeat ability for better training as shown in the figures below

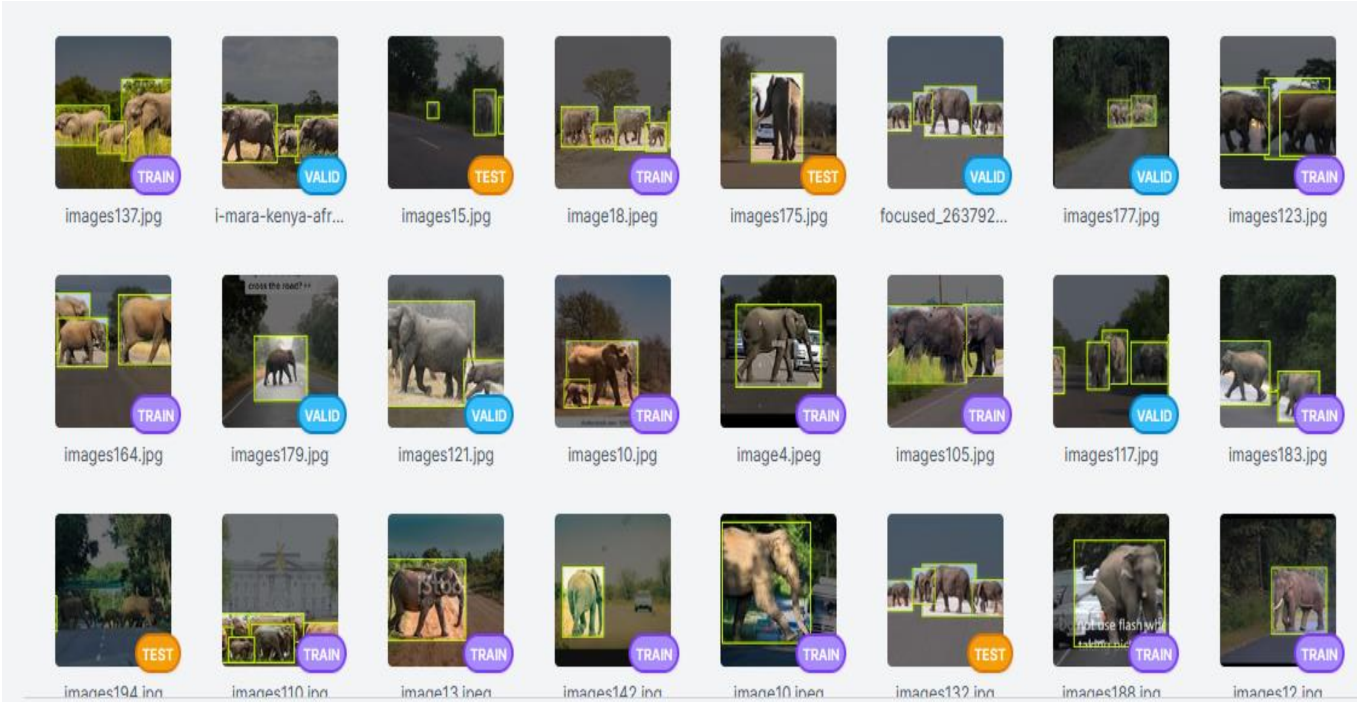

Figure 3-2: The African elephants' dataset

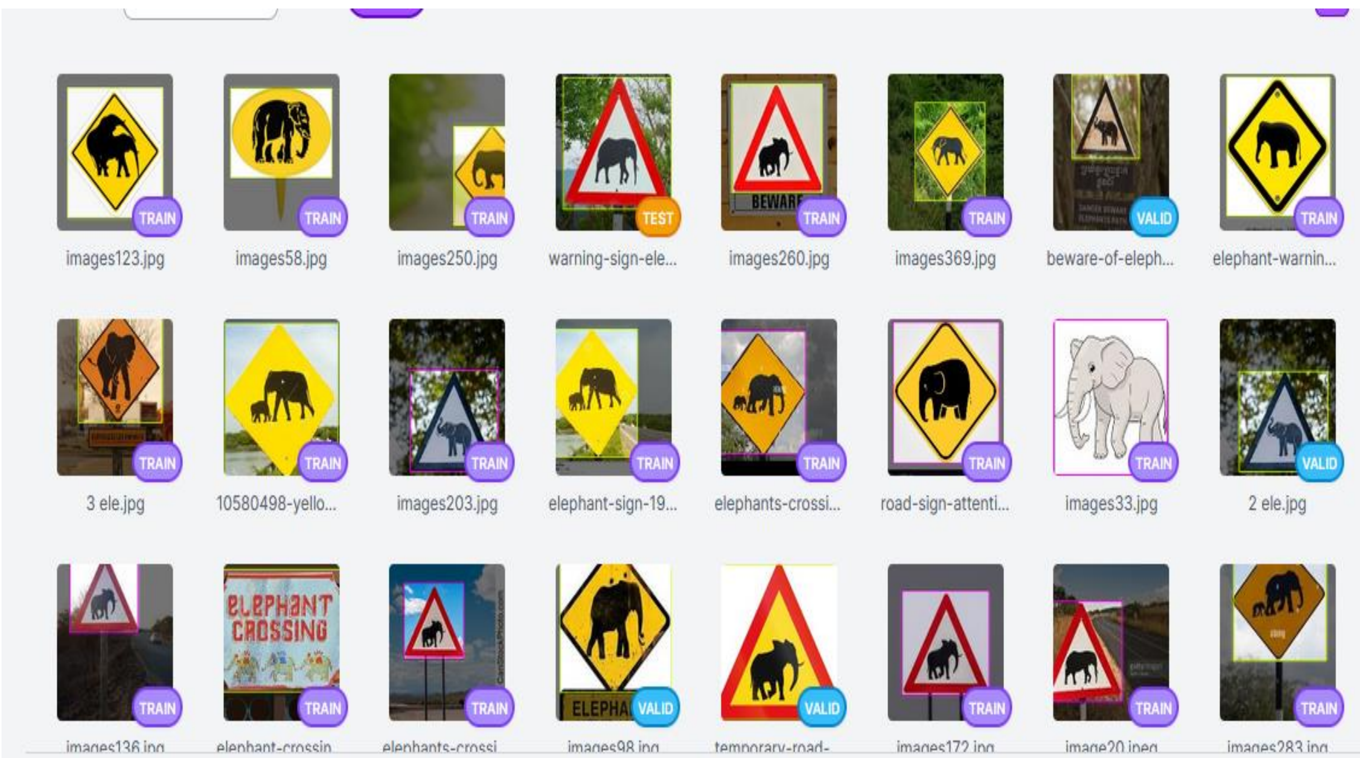

Figure3-3: The Elephant signposts dataset

3.4.2 Data preprocessing and augmentation

Initially the elephant sign posts dataset had a total of 237 images and the elephant's dataset with 328 images of different sizes and shapes after sorting which wasn't sufficient enough to generate better accuracy and performance of our model due to the fewer numbers lower than desirable, prompting us to perform preprocessing and augmentation steps for model performance improvement.

We preformed the preprocessing step by adjusting the image resolution size to 448x448 from 416x416 originally on either of the datasets and also applied a variety of augmentation approaches such as gray scale (applied to 25% of the images in each dataset), horizontal and vertical flips, brightness (between -25% and +25%), shear (+15% and -15% horizontal and vertical) and rotation between -15° and +15°. This increased the number of images to 742 for the elephant sign posts dataset, 491 images for the elephants 'dataset and 1756 images for the Kaggle elephant dataset. For each new dataset after augmentation, 87% was for training, 8% for validation and 4% for testing which facilitated better and improved model training.

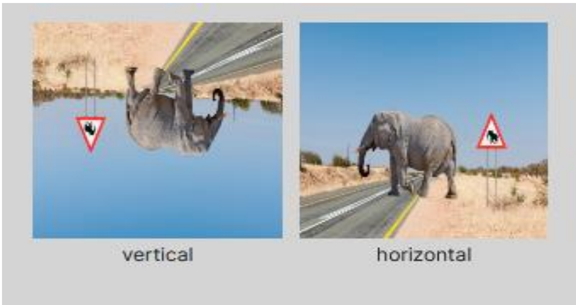

Figure3-4: An elephant image after vertical and horizontal augmentation step

After augmentation we generated over 2000 images that we use used for training our object detectors and the number was appreciable since our model only mainly performs detection and classification steps for the pre trained model can be the same for both classes.

3.4.3 Model architecture specification

We chose YOLOv4 because it is a fast object detector compared to other detectors like EfficientDet which makes its real time application very feasible and can analyze a large number of frames per second. This makes it desirable for our project that requires analyzing a large number of frames in a short amount of time as shown in the figure below.

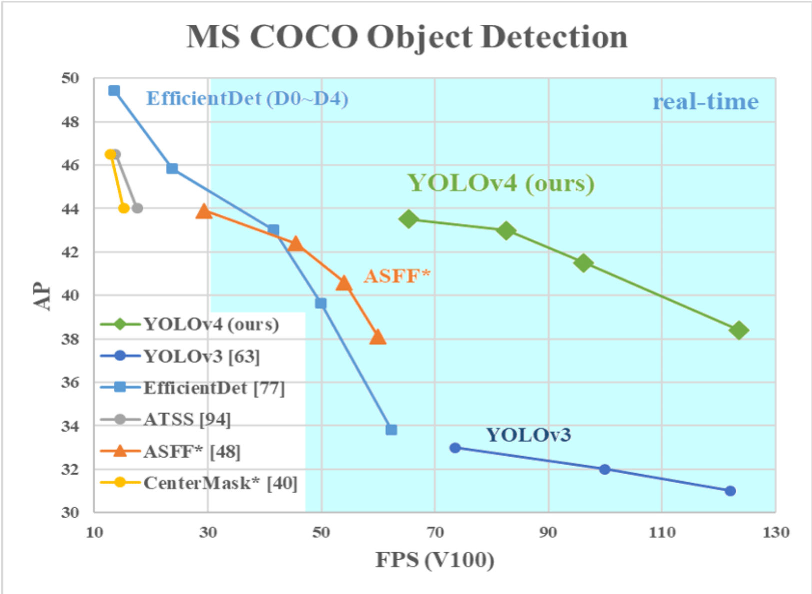

Figure3-5: A graph showing YOLOv4 superior performance on AP and frames per second

NB: Average precision (AP) and frames per second are high compared to other object detectors as clearly observed from the graph that there was a 12% increase in Frames per second and 10% increase in average precision compared to YOLOv3

3.4.3.1 Google Co laboratory

Google Co laboratory being a free source jupyter notebook environment provided by google wherein we can write and execute code, providing RAM of 12 GB with a maximum extension of 25 GB and a disk space of 358.27 GB. This assisted greatly in overcoming the limitation we had of actual implementation of the object detector on our personal computers which were limited to RAM of 4GB. Therefore, Google Co laboratory was an easy and cheap means to implement the training and inference.

Model architecture

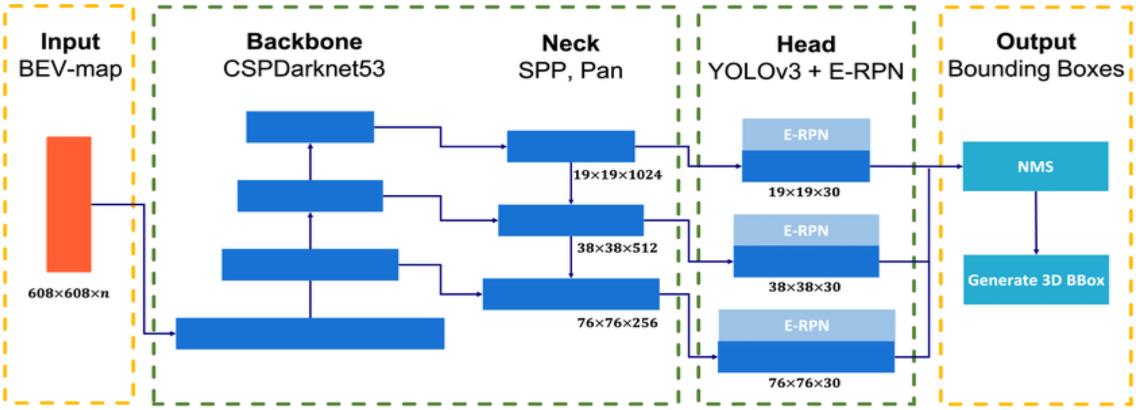

Figure3-6: YOLO v4 model architecture.

3.4.3.2 Model specifications

Our model was developed with implementation of specific parameters so as to achieve our required performance as shown in the table below.

Table 3-2: Table showing the parameters used for model training to obtain an optimal performance

YOLO v4-tiny parameter	Size
Batch	32
Width	448
Height	448
Subdivisions	24
Channels	3
Hue	0.1
Decay	0.0005
Angle	0
Momentum	0.9
Saturation	1.5
Exposure	1.5

NB: These are the optimal parameters that were achieved by analyzing the required accuracy and precision.

3.4.4 Mode Training

With the generated custom datasets, we carried out training of the YOLO V4 tiny model on these datasets in the darknet framework as follows;

- Installed darknet dependencies and framework of YOLO V4 tiny.
- Downloaded custom dataset for YOLO V4-tiny.
- Wrote custom YOLO V4-tiny training configuration.
- Trained custom YOLO V4-tiny detector.

After specifying the model architecture and running the model to train for the first time on our datasets, we realized a very large deviation between what average values we obtained and what we expected so we performed the following steps to improve on the model performance;

- We first adjusted the image resolution sizes of the images from 512 by 512 to lower sizes; 448 by 448
- We then reduced the batch numbers for model evaluation to achieve shorter computational time.
- We also collected more images in addition to the ones gathered earlier from Google to improve accuracy.

During training, we observed mainly two key parameters; @mAP and loss

3.4.4.1 Mean average Precision (@mAP)

Mean Average Precision (mAP) is the current metric used by the computer vision community to evaluate the robustness of object detection models and is based on four parameters which are the confusion matrix, intersection over union (IOU), recall and precision which are the basis of generating the average precision (AP) of the object detector whereas the mean of the average precision over the number of classes gives the mean average precision.

The mean average precision ranges from 0 to 1 where mAP=1 being the optimal predicted value (good performance) and 0 being the lowest predicted value (poor performance).

NB: When we trained our models, from our first training we obtained mAP@0.5 of 0.69 and with model architecture adjustments as earlier discussed we obtained mAP@0.5 of 0.93 which clearly indicates improved the performance of our models

3.4.4.2 Loss

The loss function composes of the localization loss (errors between the predicted boundary box and the ground truth). the confidence loss (the abjectness of the box) and the classification loss so a very low loss value implies a faster and more accurate model training because the loss represents the error made in predictions by the object detector and values of loss below 0.2 are acceptable for most object detectors

NB: For our first training, we obtained very large loss values for our object detectors but after considerable adjustments earlier discussed, the loss values reduced to values below 0.2 which evidenced faster and more accurate model training of our object detectors.

3.4.4.3 Comparison with other models

We trained other object detectors like the YOLOS coco and Detectron 2 and compared their performance with the YOLOv4 tiny model to see how it faired based on mAP and loss against its counter parts

3.5 Integrating a distance, speed and direction estimation algorithm with our object detection model

3.5.1 Triangular similarity for distance estimation in python.

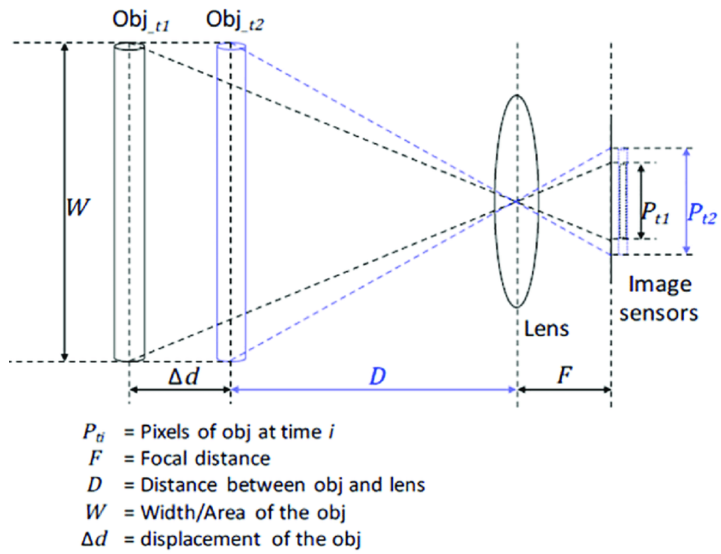

Figure 3-7: The triangular similarity

This principle is based on the property of similar triangles and by using a reference image of known width and known distance from the camera, we can determine the apparent width of the reference image which assists in determining the focal length of the camera as shown below.

$$F = \frac{D.Pt1}{W}$$

where F = focal length, Pt1 = Pixels of reference image, D = distance between object and lens and W is the measures width of the object.

On moving the object through a distance away from the camera, we then use the concept of similar triangles to get the new distance.

$$d = \frac{F.W}{Pt2}$$

Where'd is the new distance of the object from the camera and Pt2 is the pixels for the new object.

3.5.2 The implementation in python

This was done in series of steps sequentially as described below;

- Capturing reference images with known distance from the camera and this was achieved using our smart phones while also measuring the distance from the camera using a ruler.
- We defined a function for our object detector using the trained weights from the YOLOv4 tiny repository. With a bounding box around the reference image by the object detector, we use the size of the box to determine the distances of the subsequent objects detected from the camera. This is shown in the code below;

```
def object_detector(image):
    classes, scores, boxes = model.detect(image, CONFIDENCE_THRESHOLD, NMS_THRESHOLD)
    # creating empty list to add objects data
    data_list =[]
    for (classid, score, box) in zip(classes, scores, boxes):
        # define color of each, object based on its class id
        color= COLORS[int(classid) % len(COLORS)]

        label = "%s : %f" % (class_names[classid[0]],score)

        # draw rectangle on and label on object
        cv.rectangle(image, box, color, 2)
        cv.putText(image, label, (box[0], box[1]-14), FONTS, 0.5, color, 2)

        # getting the data
        # 1: class name  2: object width in pixels, 3: position where have to draw text(distance)
        if classid ==0: # person class id
            data_list.append([class_names[classid[0]], box[2], (box[0], box[1]-2)])
        elif classid ==67:
            data_list.append([class_names[classid[0]], box[2], (box[0], box[1]-2)])
        elif classid ==20:#elephant
            data_list.append([class_names[classid[0]], box[2], (box[0], box[1]-2)])
        # if you want inclulde more classes then you have to simply add more [elif] statements here
        # returning list containing the object data.
    return data_list
```

Figure 3-8: Code snippet of the object detector

- We then defined functions for computing the focal length and distance as shown below;

```
def focal_length_finder (measured_distance, real_width, width_in_rf):
    focal_length = (width_in_rf * measured_distance) / real_width

    return focal_length

# distance finder function
def distance_finder(focal_length, real_object_width, width_in_frmae):
    distance = (real_object_width * focal_length) / width_in_frmae
    return distance
```

Figure 3-9: Code snippet of the distance and focal length

- We then averaged the distance of the object detected in the frames over 10 frames so as to get the average distance from the camera.

- We also included the time library to measure the time between 10 frames and by using distance and time relation, we obtained the speed of the object detected.
- We then averaged the speed over 10 frames to obtain the average speed so as to improve accuracy as shown in the code below;

```python
def speedfinder ( covereddistance , timetaken ):
    speed = covereddistance/timetaken
    return speed

# average finder
def averagefinder ( completelist , averageofitems):

    lengthoflist = len ( completelist)
    selecteditems = lengthoflist - averageofitems
    selecteditemslist = completelist[selecteditems :]
    average = sum ( selecteditemslist)/len (selecteditemslist)
    return average
```

Figure 3-10: Code snippet of average and speed finder

NB: The position detection is based on obtaining a positive distance or a negative distance where a positive distance implies motion of the detected object towards the camera and a negative distance implies motion away from the camera.

3.6 Deploying the system on a processor and evaluating its performance.

We used the raspberry Pi4 as our choice for the processor because it has a fast 1.5GHz processor that could run the object detector because object detectors require a fast CPU and 1.5 GHz processor can handle a large number of frames per second without declining in performance which allows for real time performance of the system.

Figure 3-11: Raspberry Pi4 Module we used

Despite the fast processor speed, the major limitation to implementing object detectors is the computation requirements for such models

NB:

i. The raspberry Pi 4 has a camera interface as the input source to the object detector necessary to collect video feed from the environment under monitoring. Therefore, from the video feed, the object detector performs computation to detect and generate alerts.

Figure 3-12: A raspberry Pi 4 camera

ii. The processor system is compatible with the python programming language meaning that the implemented code does not require converting to any other programming language thus could be implemented directly.

3.6.1 Implementation of system deployment on the processor

We acquired the raspberry Pi 4 together with a keyboard, mouse and monitor necessary for interacting with the raspberry pi 4 module.

We installed the Raspbian OS and module libraries which are required for the operation of the module and included Jupyter a python IDE and transferred all the code we had implemented to a SD- card that was inserted into the module

We then added the camera module on its prescribed slot of the raspberry Pi 4 module.

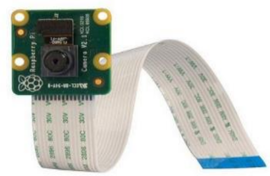

Figure 3-13: An installed Raspberry pi 4 camera module

We then set the camera feed to be the input to the module and started running the object detector and achieved promising results.

CHAPTER 4 : RESULTS

4.1 Experimental Results

4.1.1 Object detection model training results

Model training was done using Google Co laboratory an online accessed free GPU environment with sufficient RAM and storage to store the values of the object detector. After adjustments made in the model specifications as earlier discussed and after multiple training sessions, we obtained the best desired results as tabulated below.

A table showing the results obtained for the key parameters after 4000 iterations model training on the different datasets.

Table 4-1: shows the results of elephant and elephant signposts detection.

DATASET	mAP@0.5	Average loss	Average IoU
African elephant and Kaggle elephant dataset	0.93	0.187	70.14%
Elephant signposts	0.93	0.098	71.32%

NB: It can also be seen that the model can better detect objects of different sizes under varying constraints as depicted by the graph below of mAP@0.5 and loss against iterations. These graphs were obtained over 1000 iterations in which we plot the final value of @mAP_0.5 for the last image of each iteration.

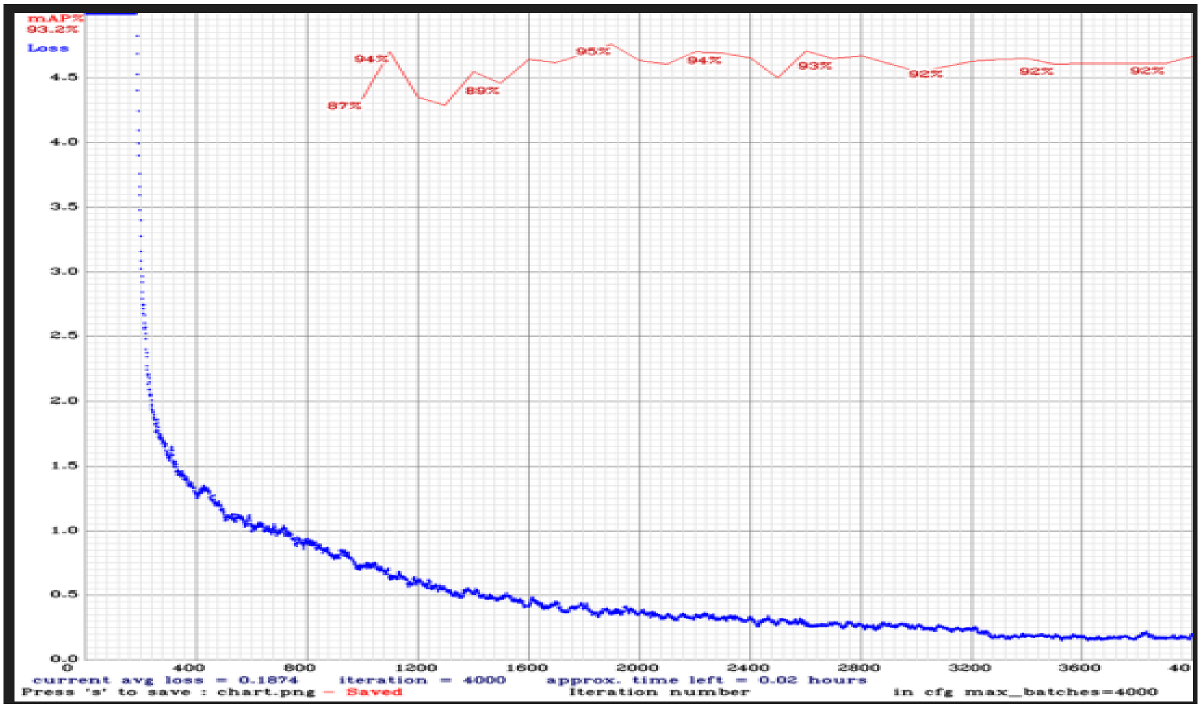

Figure 4-1: A graph of mAP and loss against number of iterations for elephant detection

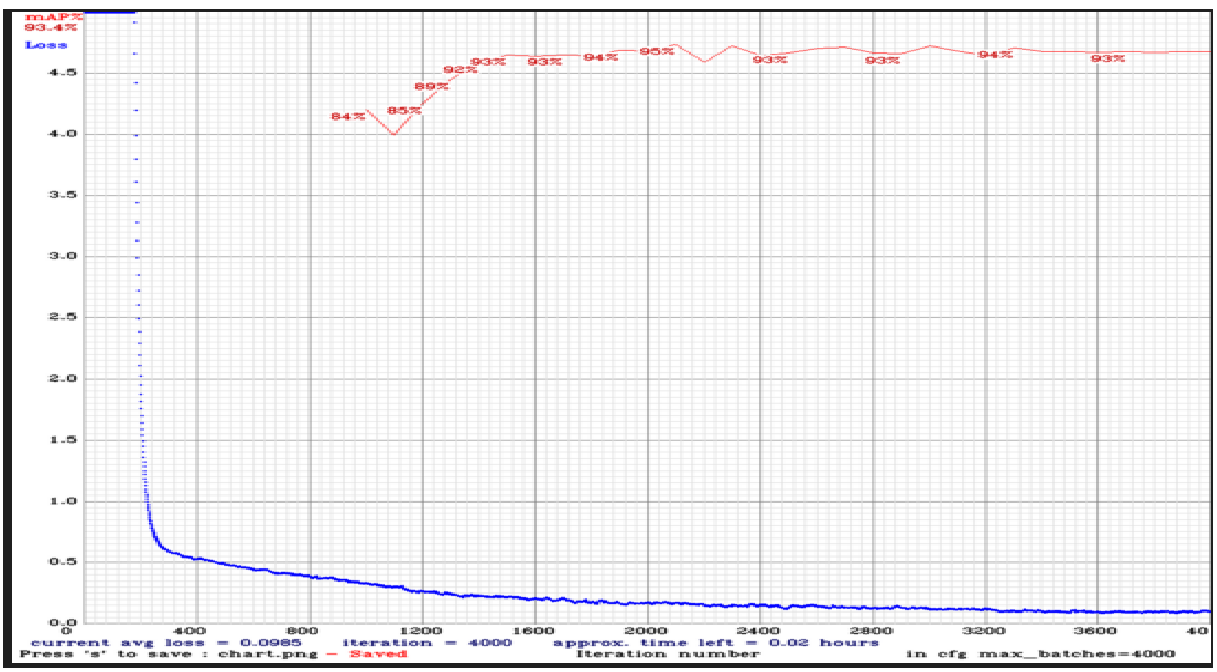

Figure 4-2: A graph of mAP and loss against number of iterations for elephant signpost detection

After obtaining optimal results above for the training, we performed testing on the object detector using random images from the test dataset so as to determine the object detector accuracy as depicted in the images below.

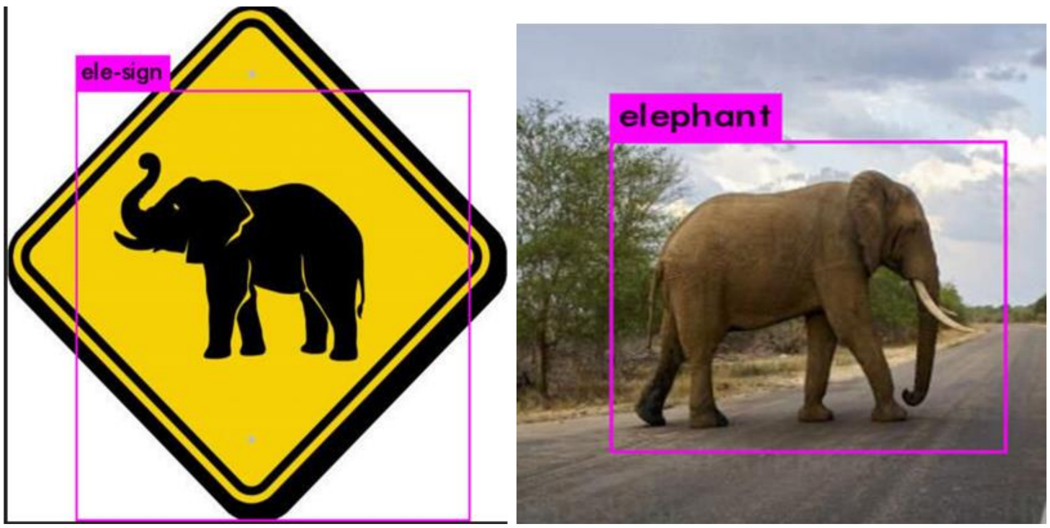

Figure 4-3: Test images for the object detectors after inference

4.1.1.1 Comparison with other models

After training the object detector; we also trained other models such as YOLO S coco and Detectron 2 for performance comparison with our trained model basing on two major parameters @mAP_0.5 and Average loss. This is as shown in the table below;

A table showing a comparative performance of YOLOv4 against other models

Table 4-2: A table showing a comparative performance of YOLOv4 against other models

Model	**mAP@0.50**	**Average loss**
YOLO v4 tiny darknet	0.93	0.0987
YOLO S COCO	0.79	0.097
Detectron 2	0.60	0.22

4.1.2 Distance and speed estimation integration results

After integrating the distance, speed and direction estimation we ran several videos through the object detector to test the integrated system on as much data as possible while including other different object classes like humans, cars and cows. The test on other classes was aimed to determine how easily we can alter the object classes while maintaining accuracy in speed and distance estimation as evidenced below;

Figure4-4: A test on elephants in Kruger National Park South Africa

Figure4-5: Using humans' class to test the distance and speed estimator

We then set a threshold distance of 10 m from the camera for testing whether the alert would be generated if the estimated distance is less than 10m so after simulation; results showed that the red highlight in the image indicates that the critical distance has been exceeded as evidenced below

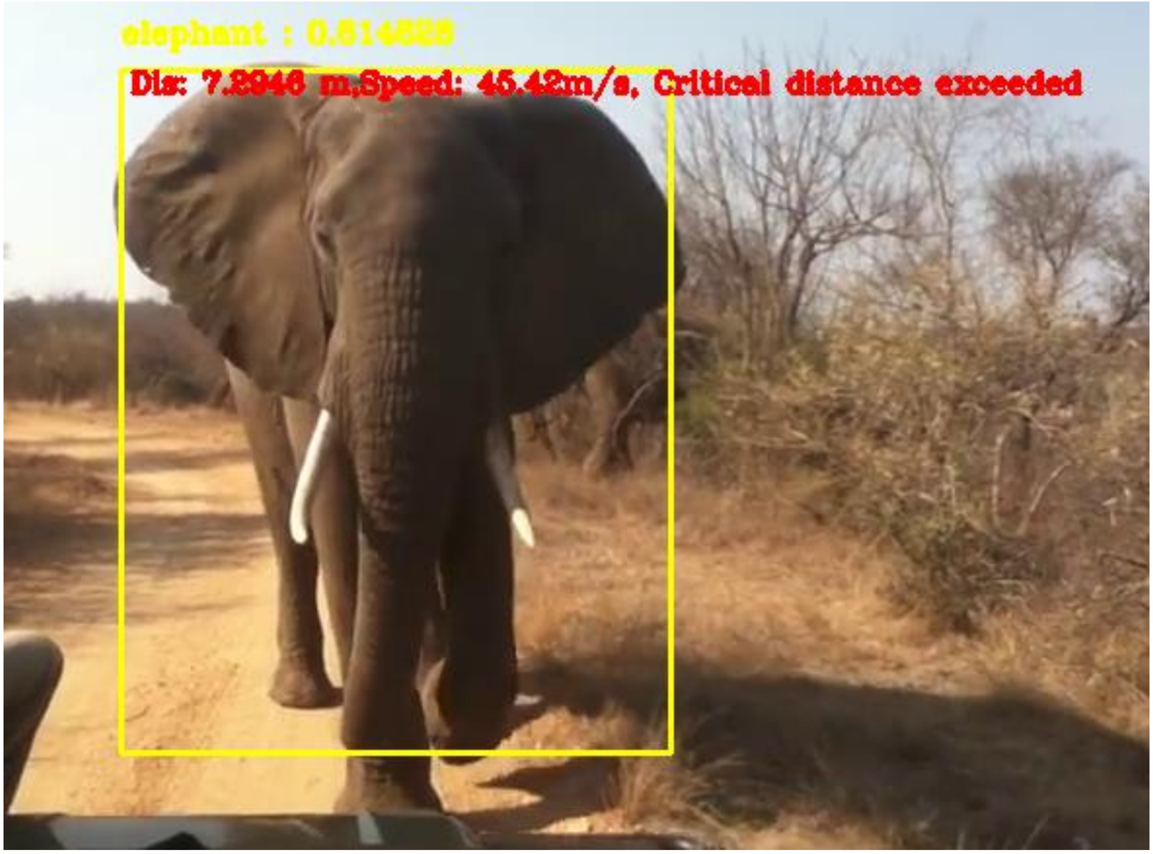

Figure4-6: A detection with the threshold distance of 10m exceeded

4.1.3 Deployment on Raspberry Pi 4 and webcam results

We used the raspberry Pi 4 module and our PC webcam to test the performance of the integrated detector in real world scenarios, where we changed the class for detections to humans in order to achieve an approximation of the desired results. We measured and averaged the accuracy of detecting a person moving to and from the camera at low speeds. The real-world human detection results obtained from deployment are tabulated below.

Table 4-3: A table showing the results obtained from the webcam deployment

Parameter	**Accuracy**	**Comments**
Speed estimate	62%	Average
Distance estimate	95%	Excellent
Range of detection	(0.1-20) m	Good
Direction	98%	Excellent

Table 4-4: A table showing the results obtained from the raspberry Pi 4 module deployment

Parameter	**Accuracy**	**Comments**
Speed estimate	56%	Average
Distance estimate	90%	Excellent
Range of direction	(0.5-8) m	Good
Direction	98%	Excellent

CHAPTER 5 : CONCLUSION

5.1 DISCUSSION AND CONCLUSION

We created our own datasets for elephant and elephant signposts detection tasks to train the YOLOv4-tiny model. After that we conducted experiments using the model for both detections and analyzed their results. There of which the robustness of the model was assessed through performance comparison using precision and average loss metrics for different object detection models for training over 4000 iterations. We also obtained the average IoU and average loss for each of the model detectors. Finally, we evaluated the overall accuracy of the model using mAP metric at 0.5 IoU threshold value and average loss. Therefore, both experimental results and performance evaluation evidence that YOLO v4-tiny object detection model is faster and more accurate which is desirable and more efficient.

We integrated our object detection model with a distance, speed and direction estimation algorithm using python programming language by utilizing saved weights of the object detectors from model training and achieved desirable results which evidence good system performance and accuracy.

We deployed our system on a raspberry Pi 4 module and webcam implementing it in real world scenarios while detecting humans as our object class at low speeds which produced better and promising results for actual implementation of our system

5.2 LIMITATIONS

- Low processor speeds for real time detection and operations thus limited system reliability.
- Relatively higher costs for purchasing better and faster processors for more system efficiency.
- Poor visibility of obstacles under dark or insufficient light conditions especially during night times.
- Narrow image resolution sizes for model detection of large objects such as elephants

5.3 RECOMMENDATIONS

- Using faster processors such as Rock 64, Orange Pi prime for improved real time efficiency. This is because of their faster processing speeds for analysis of real time scenarios.
- Increasing image resolution for better accuracy in model detection and improved system efficiency. This improves image visibility of the spatial features of obstacles by the system.
- Utilization of lighter models such as YOLO V7 and Fast R-CNN for cheaper and better system performance. This is because they require smaller memory space allocation for faster model training on larger custom datasets.
- Ensuring correct and actual measurements of the reference image relative distances. This ensures and maintains accuracy of the system in distance estimations of obstacles from the host vehicle.

REFERENCES

[1] P. Horn, "Electric vehicle-Wikipedia," 1 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/Electric_vehicle. [Accessed 2 September 2022].

[2] R. Matulka, "The History of Electric Car," 15 September 2014. [Online]. Available: https://www.energy.gov/articles/history-electric-car. [Accessed 12 September 2022].

[3] Nora, "Kiira Motors: Electric vehicles made in Uganda by 2021," 29 August 2020. [Online]. Available: https://www.electrive.com/2020/08/29/kiira-motors-electric-vehicles-made-in-uganda-by-2021/. [Accessed 9 September 2022].

[4] K. Kazibwe, "Uganda: 4159 People Died in Road Acidents in 2021-Boda Bodas Killed Five Each Day - Police Report," 18 May 2022. [Online]. Available: https://nilepost.co.ug/2022/05/18/4159-people-died-in-road-accidents-in-2021-boda-bodas-killed-five-each-day-police-report/. [Accessed 10 September 2022].

[5] [Online]. Available: https://www.google.com/search?q=lane+departure+warning+system&client=ms-android-transsion-infinix. [Accessed 20 09 2022].

[6] [Online]. Available: https://technoupdate2017.wordpress.com/2017/06/17/night-vision-and-pedestrian-detection-system-in-cars/. [Accessed 20 09 2022].

[7] [Online]. Available: https://www.google.com/search?q=radar+sensor+operation+in+cars&client=ms-android-transsion-infinix. [Accessed 20 09 2022].

[8] [Online]. [Accessed 20 09 2022].

[9] J. Jansson, Collision Avoidance Theory, Linkoping: Jonas Jansson, 2005.

[10] L. Jiao, *A Survey of deep learning-based Object Detection*, vol. 4, no. 5 Jul, 2022, p. 32, 2016.

[11] V. Rajput, "YOLO v4 explained in full detail," AlGuys, 23 December 2021. [Online]. Available: https://medium.com/aiguys/yolo-v4-explained-in-full-detail-5200b77aa825. [Accessed 22 September 2022].

[12] [Online]. [Accessed 20 09 2022].

[13] [Online]. [Accessed 20 09 2022].

[14] [Online].

Appendix A

MAKERERE UNIVERSITY

FULLPROPOSALREPORT

PROJECT TITLE: A DETECTION AND ALERT SYSTEM FOR ANIMAL ROAD CROSSINGS IN ELECTRIC VEHICLES USING COMPUTER VISION

BY: RONALD SEKASI 18/U/497

MARVIN BENJAMIN KIVUMBI 18/U/498

SUPERVISORS

MAIN SUPERVISOR: Mr. DERRICK SEBBAALE

CO-SUPERVISOR: Ms. AGATHA TURYAGYENDA

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Background

Electric vehicles are the future. In an aim by the government of Uganda to improve on the transport sector, it has allocated to locally based Kira Motors space to begin production of electric vehicles by 2021. As we speak, the company had set a target of 5,000 electric vehicles each year by the end of last year. Though the pandemic was attributed to the failure to meet the set targets due to a lack of funding. This year the company has had multiple tests runs of the KAYOOLA EVS, a fully electric bus that offers almost a 300 km range on full charge. In addition to the electric buses, the company has also proposed plans to produce solar buses, diesel buses and all electric SUVs that they hope to make with locally based materials.

Kiira Motors attributed its increased investment in electric vehicles to the ever-increasing global demand. In the third quarter of 2021, the Alliance for Automotive Innovation reported that the sales of electric vehicles had reached 6% of the US light duty automotive sales with 187,000 vehicles being the highest number of sold EVs ever recorded. Over 65 million electric vehicles were sold worldwide by the end of 2021 amidst the global COVID -19 pandemic. This is evidenced by the 109% increase as per the year 2020, the main consumer being mainland China. This comes as no surprise that the state-owned Kiira Motors partnered with CHTC, a China based part manufacturing plant to aid the production of the electric vehicles. The company was allocated land by the government in Jinja to manufacture all the close to 3200 parts needed by each electric bus using locally made steel.

1.2 Problem

Though the country is trying to go electric, the challenge of road safety is still lingering on the minds of many Ugandans. Just in a month [September,2021], the Uganda police reported almost 1407 road accidents out of which 229 of these were fatal. The police further cited that reckless driving was the main cause of the high rate and encouraged extra diligence. Another report further cites that almost 80% of the road accidents are due to human error and thus the number of lives lost are due to inherent human behavior like drunk driving and many more factors.

Few of these accidents are attributed to animal road crossings because most times the driver is alert to avert any impending collisions with animals. The impact of the road crossings needs to be brought into perspective because Uganda is primarily based on an agricultural background with tourism a supporting sector to the economy. These two sectors have animals as the common link between them and road safety. Suburban and urban farming are common in many of Uganda's cities including Kampala, the country's capital. This implies that the chances of collisions with animals which have been free ranged are high and likely and have been noticed and recorded numerous times. This is evidenced in a report released by the Uganda Police for the year ending 2020, where they attributed 2% of the road accidents to animal road crossing including the Trinity bus incident involving an elephant in Pakwach.

1.3 Aim

In this paper therefore we propose the need for a system that can reduce the impact of animal road accidents on the number of accidents. We hope to design a system that can alert and warn the driver in case of eminent danger. The impact in some cases may not be averted but we also hope to reduce the impact of collisions in case they are unavoidable. The main target being the need for this system to be able to detect abrupt crossings using computer vision. The development of models and deep machine learning will aid our detection and alert system, making us freer from the effect of human error since this is eliminated by the use of intelligent systems.

The basis of the detection is a highly sensitive and accurate YOLOv5 model, the basis of many real time detection systems. This will aid the detection and is based on a Python interactive environment by Google Co-laboratory. The road sign detection system is also based on the model and with training with custom datasets we hope to improve on the accuracy and sensitivity. Thus, more to this we hope to do some parameter estimation for instance distance estimation, speed estimation and direction estimation to actually give the real time information on the car's environment. Therefore, computer vision and object tracking are the aids we hope to successfully implement the system.

1.4 Justification

The significance of collision warning / assistance systems is to curb the number of road accidents that happen all through the world. It is estimated that the US alone spends almost 1 billion dollars in damages and insurance policies due to road accidents due to animal crossings (deer). Thus, such systems have recorded enormous improvements when used in vehicles and therefore implementation with electric vehicles would not only support the increased use of the vehicles but also increased road safety. The UN has also adopted a decade for change to cut down by 50 % of all road accidents and using intelligent systems integrated with cars would work a long way in trying to meet that target.

1.5 Report Scope

Chapter 1 comprises of the project background, problem, project aim and justification. Chapter 2 introduces the literature review that comprises of theory about collision avoidance systems and sensor technology. Chapter 3 comprises of the project objectives and techniques necessary in achieving them as well as the expected results and plan for implementation. Chapter 4 includes conclusions and recommendation which comprise of analysis and proof of the expected results, project limitations and what should be done to achieve an optimal system design

CHAPTER 2: LITERATURE REVIEW

2.1 Vehicle detection systems for Collision avoidance systems.

Almost 1.24 million people die worldwide die in road accidents. This value if the current trend is to go by is estimated to increase by 65% in 2030. The National Highway systems indicates that 25 % of the road accidents are due to driver inattention. Driver in attention could be inform of distractions (cognitive, visual and auditory distractions), fatigue and immature driving like reckless driving and over-speeding.

Since human errors are the most common reasons for road accidents, design of vehicle systems that are safer is on the agenda of many engineers. Safety systems are of mainly two types; active and passive. These have various examples of both categories as shown in the figure below.

Under the development of active systems, the use of optical sensor, radar and sonar have found wide range application. Their use has greatly improved car safety with a reduction of accidents in the number of vehicles with such systems included. There are still issues on sensor effectiveness and hardware compatibility. Therefore, the requirements of robustness, accuracy and cost efficiency are key considerations while developing such systems.

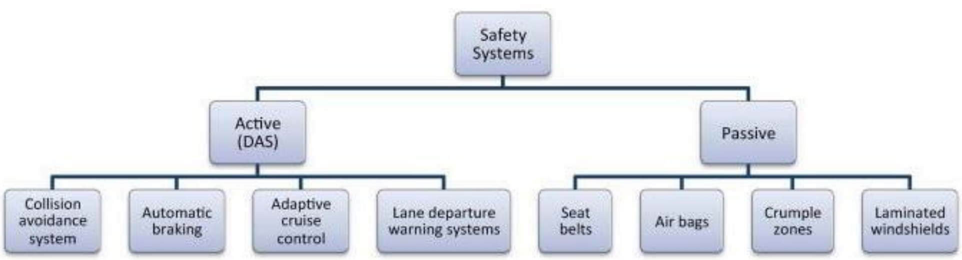

figure 1 : Types of safety systems [source: [1]]

The algorithmic side of the active systems is the most important and therefore main focus is dedicated to developing the software. The best features of accuracy, classifier (identifier) and training the identifier are all key features to consider.

2.1.2 Sensors for collision avoidance systems.

Sensors collect information about road conditions to aid in decision making for the driver and automatic vehicles. Sensors are of two main categories which are Active sensors and Passive sensors.

Active Sensors: These operate by emitting signals and collecting the reflection of the emitted signals to detect an object or obstacle. There two major categories; Radar based and laser/Lidar (Light detection and ranging based. Radar based examples include the Pulse Doppler Radar framework that was mounted on the lower front side of car. It detected and measured the relative velocity of the car and the target by reading echoes from the target. This system worked well for different weather conditions and for a target distance of 150m. Laser /Lidar emit ultraviolet and infra-red waves. On bouncing off of obstacles these waves hit the collector and are counted as a function of time. The time is considered for a round trip giving a distance estimation of the target. The SICK system is an example of such systems and is shown in figure. Lasers are low cost and figure. therefore, widely adopted.

Figure2 : A figure showing a 3D laser sensor and it mounted on a Toyota Priuse [Source: [2]]

Passive sensors: These in contrast do not emit waves to collect information and they include optical sensors (cameras) and acoustic. Acoustic sensors extract a robust spatial feature from noisy acoustic environments by employing the gradient method and the extract is then filtered using a particle filter to avail the information required. Optical sensors use cameras and usually offer more accurate object tracking as compared to active sensors as they give a brief description of the environment around the vehicle. Multiple cameras can be installed and infra-red cameras can be used at night due to their poor vision under low brightness.

Fusion of sensors

To achieve more reliability and accuracy in object detection a combination of passive and active systems or active and active or passive and passive fusion. In such systems each system may validate and compare results with each other to offer decision making information or one sensor may detect while the other validates. Examples of these include Radar and Vision, Lidar and Vision, Acoustic and Vision and Radar and Lidar.

2.2 Forward collision warning systems

These are systems that are installed on vehicles to avoid rear end collisions with other vehicles. The system sends an alert to the driver when the front vehicle slows down or the risk of collision increases. Forward collision systems are based on a variety of technologies including Time to collision models, Time headway models (THW), kinematic based systems, Road Friction and Relative motion between adjacent vehicles. With the rapid growth of wireless systems, newer FCW models that are based on short range communication are also under research. Below we look at some of the come models widely used today;

Time to collision-based systems: Time to collision refer to the time remaining before the rear end accident occurs if the speed and course of the vehicles is maintained. The idea was brought forward by Hayward in 1971 as the time span left before the vehicles collide if no evasive action is taken. TTC has been applied in car- following systems and traffic management systems to improve on road safety. The design of some collision avoidance systems is based on TTC with assessing some other systems based on the same design. The TTC for a vehicle is shown below, and if its value is high the situation is considered safer. [3] Whether the collision is going to occur is based on the trajectory parameters of the leading and following vehicles.

$$TTC_F(t) = \frac{X_L(t) - X_F(t) - l_L}{\dot{X}_F(t) - \dot{X}_L(t)}, \qquad \forall \dot{X}_F(t) - \dot{X}_L(t)$$

Where X denotes the position, denotes the derivative of X with respect to time or speed, denotes the leading vehicle length, L and F refer to leading and following respectively. The above equation is only valid if the speed of the following vehicle is higher than the leading vehicle.

The TTC based are limited because it assumes constant velocities for vehicles during collision which is impractical because it ignores the actual acceleration or deceleration of these vehicles.

TTC also indicates the actual occurrence of dangerous situations and cannot capture potentially dangerous situations.

Time Headway systems: These not only consider the speed of the leading vehicle but the distance between the two vehicles.

Wireless sensor systems: These are based on active sensors like RADAR (Radio Detection and Ranging) as shown in the figure below, LiDAR (Light detection and ranging). These are used to detect images by sending out signals and using the reflections of these signals to extract information about obstacles front of the car. Cheng - Yi Yu, Lung Tsai Lee proposed a vehicle mounted detection and collision-based system that uses dedicated short-range communications. GPS and 3G or Wi-Fi were also used on this system for determining the extent of the driver warning, car current state parameters (like the relative speed) and the lane information. The information for various scenarios was uploaded to a dedicated cloud server for analysis. These extents the application of internet is the development of smart vehicles, Internet of things and intelligent transport systems.[4]

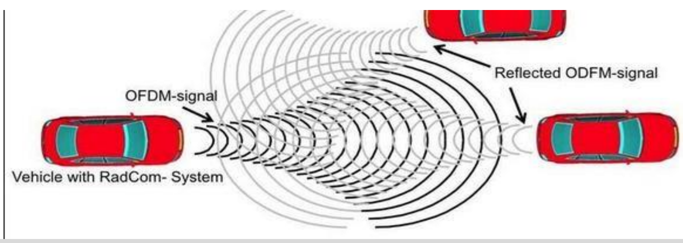

figure 3. The figure shows a radar system and its operation [source: researchget.net]

Pedestrian detection systems.

Pedestrian fatalities are on the rise in conjunction to road usage. Though intelligent transport systems propose a way of reducing on this increasing rate and therefore their development is on the agenda of many engineers and designers. These systems should not only be able to detect pedestrians in varying environments but also the possibility of collision and convey the information in a manner that does not affect the driver's maneuvered avoidance habits. These systems use imaging sensors (visible light /infra-red sensors) and time of flight sensors (Radar and Lidar). Image sensors capture images and require large processing to detect the required target object. Most application are using a fusion of both image sensing and time of flight sensors to offer more robust detection.

Visible light sensors process a large amount of data to extract information on the environment making them more accurate and the best choice for intelligent transport systems. The only limitation is the cost of processing such large amounts of data. Shaped based approaches can be used to extract the characteristic features of the images while using a trained classifier to separate pedestrians from other objects using a large number of samples to provide a better accurate prediction.

A trainable object detection system was developed by Poggio and Papageorgiou. The feature is based on over complete Haar wavelet transform that provides a rich description of the pattern. It then applies a Support Vector Machine (SVM) that is trained for detecting faces, people and vehicles. This also applies real time detection.[5] Havasi et al proposed a system that use symmetry characteristics of the legs of a walking person in order to detect pedestrians. Morphological operators are then used to detect the symmetries which are then temporarily tracked followed by classification of traces.[6]

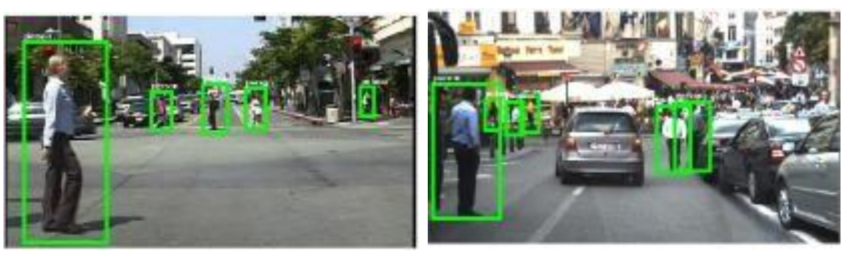

Figure 4: shows the boxes around pedestrians detected by a pedestrian Detection system [source: Pdfdollar.github.io]

Lane departure systems

Lane departure systems are high on the list of technologies that are geared towards improving road safety and reducing accidents. These use the lanes of the read ahead and the position of the vehicle with respect to the lanes to generate alerts indicating departure from the nominal lane. To detect the lanes, systems like embedded magnetic markers, digital misinform-red sensors and highly accurate Global positioning systems have been adopted in the many applications coupled with intelligent systems like the Support Vector Machine (SVM) to make them more robust and accurate.

E Salari and D Ouyang Proposed an in-car camera system, that used image processing to analyse the position of the lanes with respect to the car and generate a warning if the car breaks a lane without the use of indicators. In addition, they fused the Lane Departure system with the Forward collision warning system to make the system more robust. The Lane Departure system was made up of 2 modules, one which used the Hough Transform to identify the lane parameters of a lane mark line which best fits a set of given edge points (Lane detection). The departure warning system first locates the intersection points between the two lanes and the bottom line of the image to determine whether the system is departing to the left or right.[7]

figure 5: This shows a simulation with a lane departure system demarcating the lanes a black color. Source [ieee.org]

Deer collision avoidance using LiDAR (3D light detection and ranging)

J Chen et al proposed a LiDAR technology that can give a 360-degree surrounding of the car with high accuracy in-turn providing a real time deer detection. The LiDAR sensors can work day and night without influence from light conditions. J Chen et al as proposed a new method for detecting deer from the data collected by the LiDAR sensor. The new deer detection algorithm contains three main parts which are Background filtering, object clustering and object classification. The background filtering stage eliminates all other objects from the image except the deer, which was based on the 3D density static filtering (3D-DSF). Object clustering was intended for detecting the exact location of the deer by collecting all points belonging to the deer in one group. The group can accurately represent the deer and thus can be continually tracked to detect the behavioral characteristics. This was done using a DBSCAN based clustering algorithm with MinPts. The final stage in the involved the use of a classifier for three different targets I.e., deer, vehicles and pedestrians

LiDAR sensor has been adopted by some designers but it’s limited by cost, since LiDAR sensors are very expensive. He also suggested there was room for improvement of the deer detection algorithm [9]

CHAPTER 3: THE PROJECT PLAN.

3.1 The requirements specifications

3.1.1 Hardware specifications

Dash Camera

For this system we hope to use a camera to capture all the relevant information ahead of the vehicles and along the sides. Cameras have already been deployed in many cars and we hope that we can connect the system to the already installed dash cameras. The system will take the camera feed in real time and detect the locations, position and speed of the target items to make a comprehensive decision on whether to alert or not alert the driver of the danger ahead. Some of the examples of the current dash cameras include Garmin dash cam (67 W) and the Think way dash camera.

This is also known as an event data recorder (EDR), front facing camera connected the car battery and attached to the inner windscreen with mobile drive technology with storage capacity of 500GB and up to 180 days of continuous recordings. It has a 170° field angle, a full HD (1080p) resolution, f/1.8 aperture and night mode. This camera is used to continuously record the view through the vehicle's front view and also record the target relative distance, speed, GPS data and the voltage of the power source. These dashcam recordings necessitate sensor data for the YOLO V5s model input.

Global Positioning system (GPS) tracking technology

The technology is based on GPS trackers mounted on the target animals. This has already been implemented by the wildlife protection agencies because of poaching to track the number of threatened wildlife species like elephants and rhinos. This is to the system’s advantage because we can therefore track the animal behavior and critical zones where these animals carry out the most crossings. The GPS system is based on tracking the exact position and give data on migration and movement of these animals.

3.1.2 Software specifications

Python

The design of the system is based on digital image processing and therefore we opt for python a free and easy to use high level programming language. The specification of the latest python module we have already acquired is python 3.10.4. With the emergence of AI technology, that requires very fast processing time and speed, low processors may not be able to run the python code so for implementation we hope to use free online GPU’s (Graphics Processing Units) provide by google across their platform the Google co-laboratory. This therefore requires a dedicated internet connection to run and simulate the system.

YOLOv5 model libraries

To carry out object detection and classification we hope to import the YOLOv5 libraries that are free and easy to use across the Google co-laboratory platform. The model is the state-of-the-art neural network that has multiple layers that have been stacked together to achieve a high level of accuracy in detecting different objects. Though the model has been trained on open datasets, we hope to further train it on local datasets that to improve on the accuracy and minimize the loss and operation time. Furthermore, we hope to also include more for distance, speed and direction estimation for collision detection/ alerts.

The basic model network structure is made up of two parts namely; main network part (input side, the backbone part and the detection network) and the Neck and prediction part with the main method functions used in the input side being mosaic data enhancement. Similarly, the backbone (focus) structure is to enhance the learning ability of the convolution network to reduce budget costs and Neck (FPN†PAN) mainly adjust the number of layers to transfer shallow features and minimize the risk of data loss.

Car simulation software.

The emergency of autonomous vehicles has led to wide need and development of car movement simulation software. Among many other software we hope to upload a complete a fully functional system and monitor its performance in various scenarios to an online car simulator for example CARLA

3.2 Technical specifications

Cameras are the enablers to computer vision systems, they have become cheaper, smaller and of higher quality making their use extensive. This coupled with the improvement in computing technology; development of multi core processors and graphical processing units (GPU) has greatly made real time accurate data collection very possible.

3.2.1 System Architecture

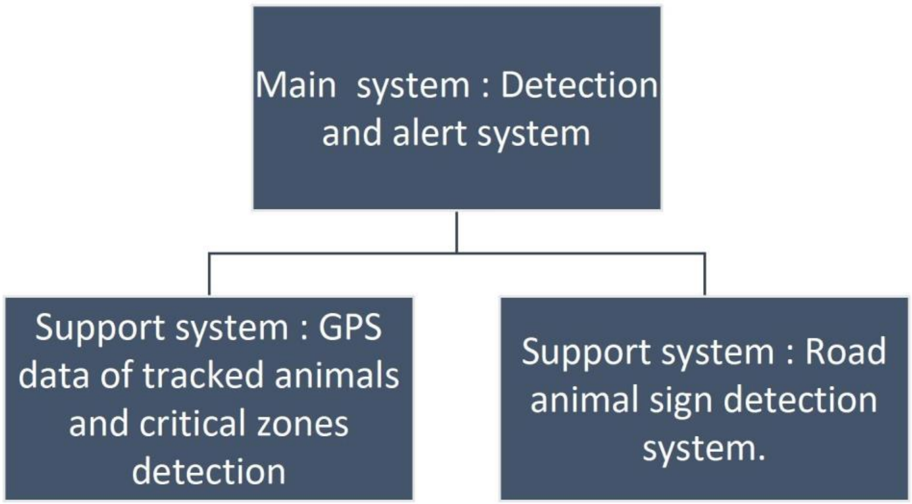

Figure 6: A diagram showing the system architecture

The system is made of the main system and two support systems. The main system is the detection and alerting system. This is a camera-based system that takes the feed from the dash board camera connected to the car and analyses it on frame-by-frame basis to track target and non-target items. We hope that the main system will also be able to carry out distance and speed estimation of the targets from the car so as to offer reliable alerts to the drive.

The 2 support systems help the main system in achieving its objective. The GPS tracking system will help detect critical zones of the tracked animals and places where they tend to cross the roads from the most. The data will then be used to give the main system exact locations of critical zones so that it can automatically switch on the system when in such zones. The road side detection system will adopt the already placed road signs to also automatically switch on the system and then switch if off if the car is out of the critical zones.

3.2.2 System Development

The development of any vision-based system should involve the stages shown below to achieve foreseeable results

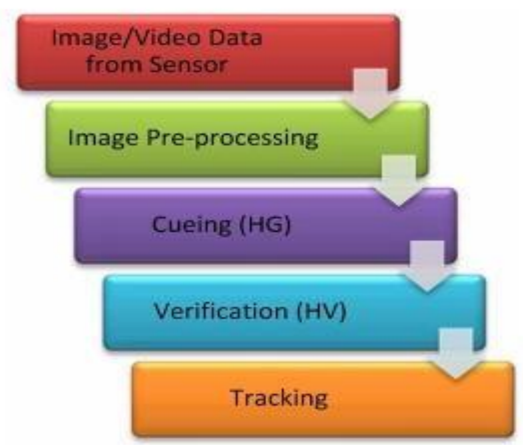

Figure 7: A diagram showing the stages in system development

Stage 1: Image/ Video data from sensor. This involves the collection of data from a sensor installed on the car. As specified earlier, these can be active, passive or a fusion of both. For vision-based systems a camera (optical sensor) provides the required data for the Collision Avoidance Systems. (CAS). The data can be in form of images or videos

Stage 2: Image Pre- Processing. When raw data from the camera is collected, noise is till incorporated. Therefore, the data collected needs to be filtered and adjusted in some cases to be compatible with the target system. Image processing mainly uses digital signal processing techniques to improve image/ audio quality.

Stage 3: Cueing (HG). During this stage, the main aim is to have a location for our Regions of Interest (Animals for this case).

Motion based techniques: These collect data samples inform of data frames. This therefore supports real- time scenario-based system design. These exploit the temporal data from optical sensor data frames to calculate and match feature points (pixels) between consecutive data frames to determine the location of the target. Dense and sparse optical flow techniques have been suggested with implementations of Scene Segmented Establishing Tracking (ASSET _2) and Smallest Unit Value Segment

Assimilating Nucleus (SUSAN).[2]

Stage 4: Verification (HV). During this stage, the aim is to validate which target. (Identification whether animal on road / animal by the wayside).

Learning based approaches using a classifier. A classifier is trained to differentiate between target ad non- target items. A set of images is provided to the classifier and through training the detection and accuracy of the classifier is gradually improved. This is done through selecting critical features that are similar to all targets. Schemes like the Artificial Neural Networks (ANN), Ad boost and SVM have so far registered significant success in vision-based detection systems.

Stage 5: Tracking is done using an Artificial Neural network, to keep monitoring the position of the target as time goes on. This is done on frame-by-frame analysis of the camera output.

3.3 Objectives

Main objective

To develop a system that can detect and alert or control the electric vehicle to curb on the number and impact of animal road crossings.

Specific objectives

- To develop a model that can detect and alert the driver on animals encroaching the road during driving.
- To develop a model, determine the critical zones on road networks based on the GPS data of tracked animals.
- To analyse the results and apply them in real world simulator to achieve comparable and promising results.

3.4 Expected Results

- We expect to develop a real time detection and alert system for animal detection:
- We expect to develop model for sign post detection and recognition
- We expect to develop real time location detection system based on GPS
- We expect to carry out a real-world simulation evidencing a high level of accuracy detection.

3.5 Work plan (Gantt chart)

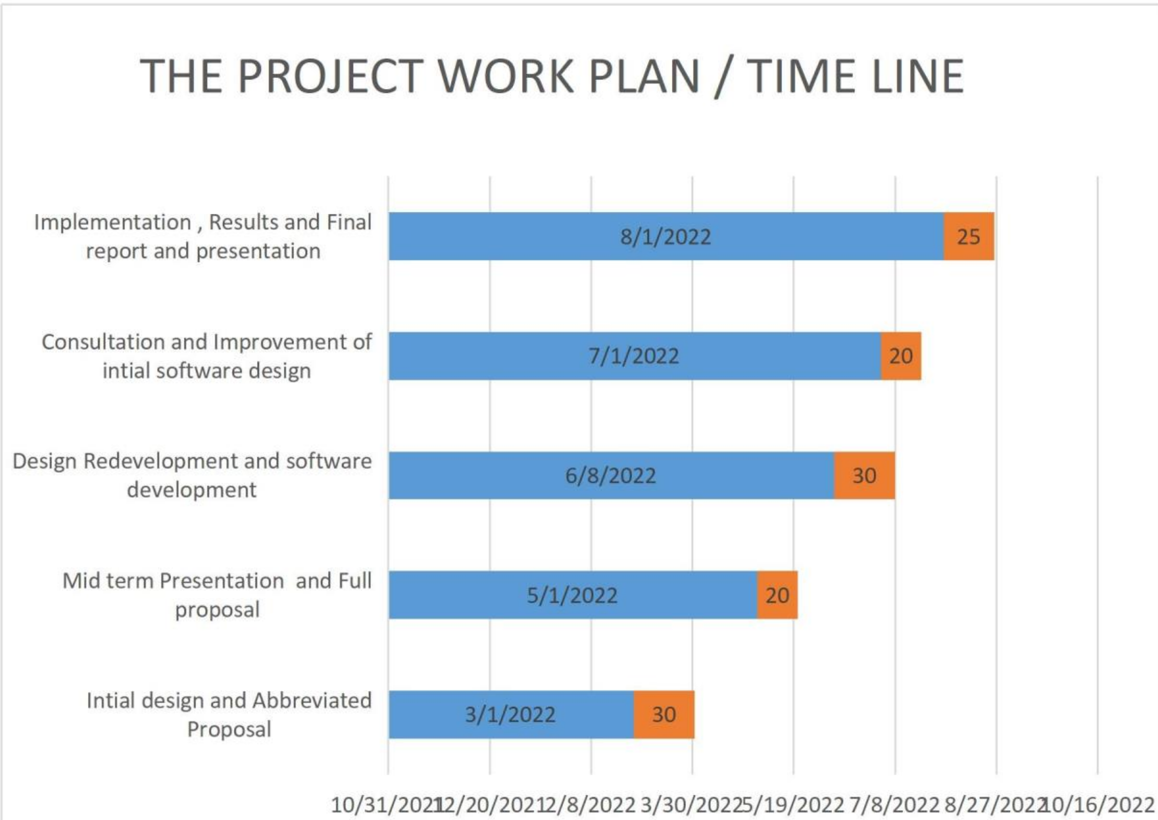

Figure 8: A work plan illustration using a Gantt chart

3.6 Budget.

ITEM /TASK	QUANTITY	UNIT PRICE (UGSHS)	TOTAL (UG SHS)
DASH CAMERA	2	230,000	460,000
INTERNET	2	150,000	300,000
GPS TRACKER	2	120,000	240,000
KIIRA MOTORS PLANT VISITS	1	400,000	400,000
TOTAL			1,400,000

Table 1: A table showing the estimated working budget

3.7 Project scope

Collision Avoidance Systems

Collision Avoidance systems are systems designed and implemented to minimize the impact of collision with objects in vehicles during driving. This project shall cover the design of a collision avoidance system based on computer vision i.e., use of an optical sensor (dash cam) for capturing video recorded data enhanced by artificial intelligence (AI) of the YOLOv5 model and algorithm for vehicle detection and tracking capability so as to alert the driver.

CHAPTER 4: CONCLUSION

4.0 Conclusion

Despite the continued development and adoption of electric vehicles in Uganda, road safety has still to a greater extent not been put into consideration yet the risks associated with utilization of these electric vehicles is on a rise but little has been affected to curb their impact. These electric vehicle related accidents and damages have resulted into a rise in the total mortality rates and serious injuries in turn high costs and expenditures on insurance policies and compensations.

Furthermore, very little of these road traffic accidents have been considered to be associated with animal involvements since most of them are attributed to human driving errors and misbehavior so this calls for care to be taken for electric vehicle animal related accidents and consideration should be made in accordance during the continual developments of these electric vehicles.

Therefore, the development of a detection and alert system in electric vehicles for animal road crossings shall be a great milestone in curbing on the impact of animal related accidents thus minimizing the related costs and expenditures on insurance policies as well as compensations thus improved GNP (Gross National Product) of Uganda.

REFERENCES

[1] Peter Seiler, Bongsob Song, J. Karl Hedrick. “Development of a Collision Avoidance System”. Published under the University of California-Berkeley. Published 1998. Available [online]:

[2] Amir Mukhtar, Likun Xia, Tong Boon Tang. “Vehicle Detection Techniques for Collision
Avoidance Systems: A Review” IEEE transactions on intelligent transportation systems, vol. 16, no. 5, october 2015

[3] Naseralavi, Seyed & Nadimi, Navid & Saffarzadeh, M. & Mamdoohi, Amir Reza. (2013). A general formulation for time-to-collision safety indicator. Proceedings of the ICE - Transport. 166. 294-304. 10.1680/tran.11.00031. Pg 2-5.

[4] Yi - Ying Chang, Cheng - Yi Yu, Lung-Tsai Lee, “Research ofVehicle-Mounted Detection and Collision Avoidance System based on Dedicated Short-Range Communications”, 2016 IEEE.

[5] C. Papageorgiou and T. Poggio, “A Trainable System for Object Detection,” International Journal of Computer Vision 38(1), 15–33, 2000.

[6] L. Havasi, Z. Szlávik, and T. Szirányi, “Pedestrian Detection Using Derived Third-Order Symmetry of Legs,” in Proc. Int. Conf. on Computer Vision and Graphics, 2004.

[7] E. Salari and D. Ouyang, “Camera-based Forward Collision and Lane Departure
Warning Systems Using SVM” Department of Electrical Engineering and Computer Science University of Toledo, Toledo, Ohio, USA 43606, 2013

[8] Arya Ketabchi Haghighat, Varsha Ravichandra-Mouli, Pranamesh Chakraborty, Yasaman Esfandiari, Saeed Arabi, AnujSharma. “Applications of Deep Learning in Intelligent Transportation Systems” Pg 4-6.

[9] JINGRONG CHEN , HAO XU , JIANQING WU , RUI YUE , CHANGWEI YUAN , AND LU WANG. “Deer Crossing Road Detection With Roadside LiDAR
Sensor” , date of publication May 14, 2019, Pg 2-7