A WEAPON DETECTION SYSTEMUSING YOLO V4 FOR SMART SURVEILLANCE

GODBLESS ODHIAMBO OKAKA

A SYSTEM PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE IN THE SCHOOL OF BUSINESS AND TECHNOLOGY IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR'S DEGREE IN COMPUTER SCIENCE.

2024

DECLARATION

I declare that this is my original work and it has not been previously submitted and approved for the award of a bachelor's degree by any University. To the best of my knowledge and belief, the work contains no material previously published or written by another person except where due reference is made in the work itself.

Students' signature:

Name.. Signature..

Date..

This research project has been submitted for examination with my approval as the university **Supervisor.**

Signature.. Date..

Dr.Muchelule Yusuf, PhD

Senior Lecturer Umma University

ACKNOWLEDGEMENT

I take this opportunity to thank the almighty God for enabling me and for his provision throughout this course. I would also like to pass my sincere appreciation to my Supervisor Dr. Yusuf Muchelule for his invaluable guidance, counsel and support ensuring this project is completed. Much appreciation to my immediate and extended families for their support, inspiration and encouragement. I am forever grateful.

ABSTRACT.

Every year, a large amount of population reconciles gun-related violence all over the world. In this work, I develop a computer-based fully automated system to identify basic armaments, particularly handguns and rifles. Recent work in the field of deep learning and transfer learning has demonstrated significant progress in the areas of object detection and recognition. I have implemented YOLO V4 “You Only Look Once” object detection model by training it on my customized data-set. The training results confirm that YOLO V4 outperforms YOLO V2 and traditional convolution neural network (CNN). Additionally, intensive GPUs or high computation resources were not required in my approach as I used transfer learning for training my model. Applying this model in my surveillance system, I can attempt to save human life and accomplish reduction in the rate of manslaughter or mass killing. Additionally, my system can also be implemented in high-end surveillance and security robots to detect a weapon or unsafe assets to avoid any kind of assault or risk to human life.

TABLE OF CONTENTS

DECLARATION i

ACKNOWLEDGEMENT ii

ABSTRACT. iii

TABLE OF CONTENTS iv

CHAPTER 1 1

INTRODUCTION 1

1.0 Background 1

1.1 Problem Statement 2

1.2 Objectives. 2

1.3 Research Questions. 3

1.4 Justification of the study. 3

1.5 Scope of the study 4

1.6 Limitations of the study. 4

CHAPTER 2 5

LITERATURE REVIEW 5

2.0 Introduction. 5

2.1 Weapon Detection 5

2.1.1 History of Object Detection 6

2.1.2 Cumulative result of detecting weapon with precision value. 6

2.1.3 Experimental results for trained deep learning models. 7

2.1.4 Class diagram for DOA layer. 8

2.1.5. Comparison with the existing work. 8

Conclusion 9

2.2 Related works 9

CHAPTER 3 11

SYSTEM METHODOLOGY 11

3.1 Introduction. 11

3.2 Software Development Methodology 11

3.2.1 Sample images from collected dataset 13

3.2.2 Feed forward convolution neural network (CNN) 14

3.2.3 Architectural details of DARKNET-53 layer 15

3.2.4 Architectural description of YOLO v4 16

3.2.5 Experimental Results ... 17
3.2.6 Bounding box around detected object; weapon category GUN. ... 18
3.3 System Analysis ... 18
3.3.1 Research Design ... 19
3.3.2 Location of the study ... 19
3.3.3 Target Population ... 19
3.3.4 Sample Selection ... 19
3.4 System Design ... 20
3.4.1 Data Flow Diagram ... 20
3.4.2 Use Case Diagram ... 21
3.4.3 System Sequence Diagram ... 21
3.4.4 Class Diagram ... 22
3.5 Feasibility Study ... 23
3.5.1 Legal Feasibility ... 23
3.5.2 Operational Feasibility ... 23
3.5.3 Technical Feasibility ... 24
3.5.4 Economic Feasibility. ... 24
3.6 Data collection Method ... 24
3.6.1 Observation ... 25
3.6.2. Review of secondary data and reports ... 25
3.7 System Implementation ... 25
3.7.1 System Component Requirements ... 26
3.7.2 System Development Tools and Techniques ... 26
3.8 Proposed Modules and System Architecture ... 27
3.9 System Testing ... 27
3.10 Validation ... 28
CHAPTER 4 ... 29
SYSTEM ANALYSIS AND DESIGN ... 29
4.1 Introduction ... 29
4.2 System Analysis ... 29
4.2.1.Requirements Determination ... 29
4.3 System Design ... 30
4.3.1 Use Case Diagram ... 30
CHAPTER 5 ... 31
THE SYSTEM CONSTRUCTION ... 31

CHAPTER 6 34
THE SYSTEM TESTING 34
CHAPTER 7 39
THE SYSTEM IMPLEMENTATION 39
7.1 Introduction 39
7.2 System Implementation 39
7.3 System Requirements Specifications 39
7.4. Libraries Installation Process 40
7.5. System Snapshots 40
7.5.1. Client Side 40
7.5.2. Server Side 41
7.5.3. Database 45
CHAPTER 8 48
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS 48
8.1. Introduction 48
8.1.2. Detecting Handguns Via Web Cameras 48
8.1.3. Identifying Performance Factors 48
8.1.4. Conclusions 48
8.1.5. Limitations 48
8.1.6. Recommendations 49
8.1.7. Chapter Summary 49
REFERENCES. 51

List of figures

Figure 1: YOLOv2 6
Figure 2: Table data 7
Figure 3: Image presenting the recorded database for object detection 8
Figure 4: Generalized approach of my research work 12
Figure 5: Handgun 12
Figure 6: Handgun 13
Figure 7: Convolution neural network architecture (CNN) 14
Figure 8: Feed forward convolution neural network 15
Figure 9: DARKNET-53 layer Architecture 16
Figure 10: Handgun with a bounding box 18
Figure 11: Rifle detection image 18
Figure 12: Dataflow diagram (DFD) 20
Figure 13: Use Case Diagram 21
Figure 14: System Sequence Diagram 22
Figure 15: Class Diagram 23
Figure 16: Client Side Login Page 40
Figure 17: Client Side Location Entry 41
Figure 18: Client Side Detection 41
Figure 19: Server Side Sign in page 41
Figure 20: Server Side Registration Page 42
Figure 21: Server Side Alerts Page 42
Figure 22: Detection Alert 43
Figure 23: Detection Alert 43
Figure 24: Detection Alert 44
Figure 25: Detection Alert 44
Figure 26: Detection Alert 44
Figure 27: Users 45
Figure 28:Specific User 45
Figure 29:Upload Alerts 46
Figure 30:Specific Upload Alert 46
Figure 31:User Authentication Token 47

List of Tables

Table 1: Results for trained deep learning models....6
Table 2: Algorithm comparison with existing work....8
Table 3: Unit Testing....35
Table 4: Integration Testing....35

CHAPTER 1

INTRODUCTION

1.0 Background

Violence committed with guns puts significant impact on public, health, psychological, and economic cost. Many people die each year from gun-related violence. Psychological trauma is frequent among children who are exposed to high levels of violence in their communities or through the media. Children exposed to gun-related violence, whether they are victims, perpetrators, or witnesses, can experience negative psychological effects over the short and long terms. Number of studies show that hand-held gun is the primary weapon used for various crimes like -in, robbery, shoplifting, and rape. These crimes can be reduced by identifying the disruptive behavior at early stage and monitoring the suspicious activities carefully so that law enforcement agencies can further take immediate action. In Kenya for instance, there has been a lot of gun related crimes in various areas.

Livestock theft in counties like Baringo for example, has made the use of firearms for both defense and theft looks normal. Some of these weapons are as a matter-of-fact illegal. Increased droughts, floods and invasive species are fueling violent conflicts between pastoralists over livestock in Kenya's central Baringo County, the intensity of which is exacerbated by the proliferation of illegal firearms in the region. Firearms trafficked from civil conflicts in the Horn of Africa have made their way into the hands of pastoralists who now see it as the only way to defend themselves and their cattle during raids and conflicts over grazing land. In 2021, there were 16 deaths from 19 livestock raids; in the first four months of 2022, there have been 39 fatalities from 24 violent clashes, half of them due to livestock raids. Violent conflicts in Baringo are linked to insecurity in neighboring counties, and drought along with politics only heighten the precarious situation.Given the increase in violence in the past few years, many herders say they would be better off defending themselves. Several villagers say having their own weapons is necessary for self-protection. It's a sentiment also expressed by those working in the relief sector within Baringo. That is just an instance associated with Baringo County alone. There are many other places where gun related crimes take place and sometimes go unpunished. There is a great need to solve this problem.

1.1 Problem Statement

Levels of gun-related violence vary greatly among geographical locations and countries. The global death toll from use of guns may be as high as 1,000 dead each day. According to statistics, 4.2 in 100000 people are killed in Pakistan every year in mass shootings. From street crimes to an individual institution attack, many precious lives suffered. This further indicates that manual surveillance system still needs human eye to detect the abnormal activities and it takes a sufficient amount of time reporting to security officials to tackle the situation. Although the human visual framework is quick and precise and can likewise perform complex undertakings like distinguishing different items and recognizing snags with minimal cognizant idea, however, it is common truth that if an individual watches something very similar for quite a long time, there is an opportunity of sluggishness and lack of regard.

Nowadays, with the accessibility of huge data-sets, quicker GPUs, advanced machine learning algorithms, and better calculations, I could then effectively prepare PCs and develop automated computer-based system to distinguish and identify numerous items on a site with high accuracy. Recent developments indicate that machine learning and advance image processing algorithms have played dominant role in smart surveillance and security systems. Apart from this, popularity of smart devices and networked cameras has also empowered this domain. However, human objects or weapon detection and tracking are still conducted at cloud centers, as real-time, online tracking is computationally costly. Significant efforts have been made in recent years to monitor robot manipulators that need high control performance in reliability and speed. The researchers have attempted to improve the response characteristics of the robotic system and to attenuate the uncertainties. The proposed developed robust model-free controller incorporates time delay control (TDC) and adaptive terminal sliding mode control (ATSMC) methods.

1.2 Objectives.

The main objective of the study was to design and develop a weapon detection system that uses machine learning and YOLOv4 object detection algorithm to detect weapons such as handguns and rifles.

The specific objectives were:

i. To identify weapons such as guns and rifles and determine their location in the society.
ii. To send an alert to the relevant authorities regarding the detections done.

iii. To generate a report on the details of the weapon detection including the location and images.

1.3 Research Questions.

The research tries to answer the following questions:

i. How can I detect weapons and determine their locations?
ii. How do I secure the information gotten from a detection from loss or corruption?
iii. How do I engage the authorities in the detection scenario so that immediate and appropriate actions can be taken?

1.4 Justification of the study.

In this research work, I aimed to develop a smart surveillance security system detecting weapons specifically guns. For this purpose, I have applied few compute vision methods and deep learning for identification of a weapon from captured image. Recent work in the field of machine learning and deep learning particularly convolution neural networks has shown considerable progress in the areas of object detection and recognition, exclusively in images. As the first step for any video surveillance application, object detection and classification were essential for further object tracking tasks. For this purpose, I trained the classifier model of YOLO v4, i.e., "You Only Look Once". This model is a state-of-the-art real-time object detection classifier. Furthermore, I am not just detecting the guns, rifles, and fire but also getting the location of the incident and storing the data for future use. I have connected three systems using socket programming as a demonstration for the real-life scenario as camera, CCTV operator, and security panels.

This work was an attempt to design and develop a system which can detect the guns, rifles, and firearms in no time with less computational resources. It was evident from technological advancements that most of the human assisted applications were then automated and computer-based. Eventually, these computer-based systems were replaced by more smart machines, robots, or humanoid robots. In order to provide visionary sense to robots, object detection played fundamental part for understanding the objects and their interpretation. Thus, my weapon detection system could also be implemented in surveillance and security robots to detect any weapon or unsafe assets.

1.5 Scope of the study.

The scope of this project was to develop a weapon detection security system, which allows the authorities to detect unlicensed weapons in the society and the location in which they were held. Further, it was to allow the government to be at a position of stopping crimes such as mass killings through gunshots prior to their happening. The system was also to allow the government and its security agencies to monitor places in the country especially the highly insecure areas in order to ensure security all over the place and the state as a whole.

1.6 Limitations of the study.

Some potential limitations of a weapon detection software research project included:

Cost: Implementing a weapon detection system was expensive, and there were ongoing costs associated with maintaining and updating the software.

Implementation challenges: Implementing a Weapon Detection Software was a significant change for security related organizations, and there were challenges in getting buy in from such enterprises, as well as technical challenges in integrating the software with existing systems.

Data accuracy: The accuracy of the data collected through the Weapon Detection Software were affected by factors such as user error or inconsistent data entry.

Ethical considerations: There were ethical considerations related to gathering and using performance data on security alerts, including privacy concerns and the potential for misuse of the data.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction.

Reducing the life-threatening acts and providing high security are challenging at every place. Therefore, a number of researchers have contributed to monitoring various activities and behaviors using object detection. In general, a framework of smart surveillance system is developed on three levels: firstly, to extract low-level information like features engineering and object tracking; secondly, to identify unusual human activities, behavior, or detection of any weapon; and finally, the high level is about decision making like abnormal event detection or any anomaly. The latest anomaly detection techniques can be divided into two groups, which are object-centered techniques and integrated methods. The convolution neural network (CNN) spatial-temporal system is only applied to spatial-temporal volumes of interest (SVOI), reducing the cost of processing. In surveillance videos of complex scenes, researchers proposed a tool for detecting and finding anomalous activities.

By conducting spatial-temporal convolution layer, this architecture helps one to capture objects from both time domain and frequency domain, thereby extracting both the presence and motion data encoded in continuous frames. To do traditional functions to local noise and improve detection precision, spatial-temporal convolution layers are only implemented within spatial-temporal quantities of changing pixels. Researchers proposed anomaly-introduced learning method for detecting anomalous activities by developing multi-instance learning graph-based model with abnormal and normal bimodal data, highlighting the positive instances by training coarse filter using kernel-SVM classifier and generating improved dictionary learning known as anchor dictionary learning. Thus, abnormality is measure by selecting the sparse reconstruction cost which yields the comparison with other techniques including utilizing abnormal information and reducing time and cost for SRC.

2.1 Weapon Detection

As far as security is concerned, it is a must for people living in an area or a country to enjoy peace, and harmony without any news of manslaughter using illegally owned weapons in the streets especially guns. This is why it I created a system that would use object detection to detect weapons and figure out if they are licensed or not before raising an alarm to the authorities based on the outcome of the detection. If the weapon detected is a gun for instance and is found to be unlicensed by the government, then there would be an alarm raised to the authorities and information regarding the person possessing the gun such as their actual location.

2.1.1 History of Object Detection

Literature suggests that YOLO v2 often struggled with small object detections. This happened due to loss of fine-grained features as the layers down-sampled the input. In conclusion, YOLO v2 applies an identity mapping, concatenating feature maps from a previous layer to capture low-level features. However, YOLO v2's architecture is lacking some of the influential essentials that are encapsulated in most of state-of-the-art algorithms. The early models were lacking in the residual blocks, skip connections, and up-sampling. On the other hand, YOLO v4 incorporates all of these. The detection of smaller objects can be seen from cumulative results demonstrated in Figure 8. I retrained both YOLO V2 and YOLO V4. Alternatively, I also conducted comparative analysis of the models with traditional CNN which is trained from the very scratch with null weights. The obtained results are summarized in Table 1.

- 

Figure 1: YOLOv2

2.1.2 Cumulative result of detecting weapon with precision value.

Table 1: Results for trained deep learning models

S. no	Models	Accuracy
1	Traditional CNN	95
2	YOLO V2	96.76

3	YOLO V4	98.89

2.1.3 Experimental results for trained deep learning models.

The subsequent part of my research was based on the recording of location where the weapon was detected so that the alarm was generated. For this purpose, at backend I also created a database. A desktop application was also developed in order to provide connectivity with the database system. There are four attributes that are collected from the site where an object like weapon was detected. The collected information needs to be translated into a geographical format of longitude and latitude. For this purpose, geocoding was performed. It was the method of translating addresses to geographical details, longitude, and latitude, to map their positions. As it could be seen, the attributes were latitude, longitude, time, and location where weapons were seen or identified. At backend DAO (Data Access Object) layer was also available to show the user the data from the database. It was component of Java Foundation Classes (JFC), which was a GUI-providing API for Java programs. Swing provided packages that rendered Java programs complex collection of GUI components and platform independent. Figure 10 presents the class diagram and implementation of DAO layer.

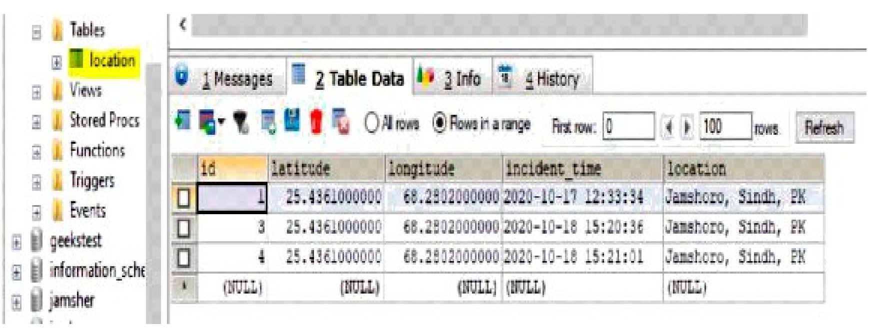

Figure 2: Table data

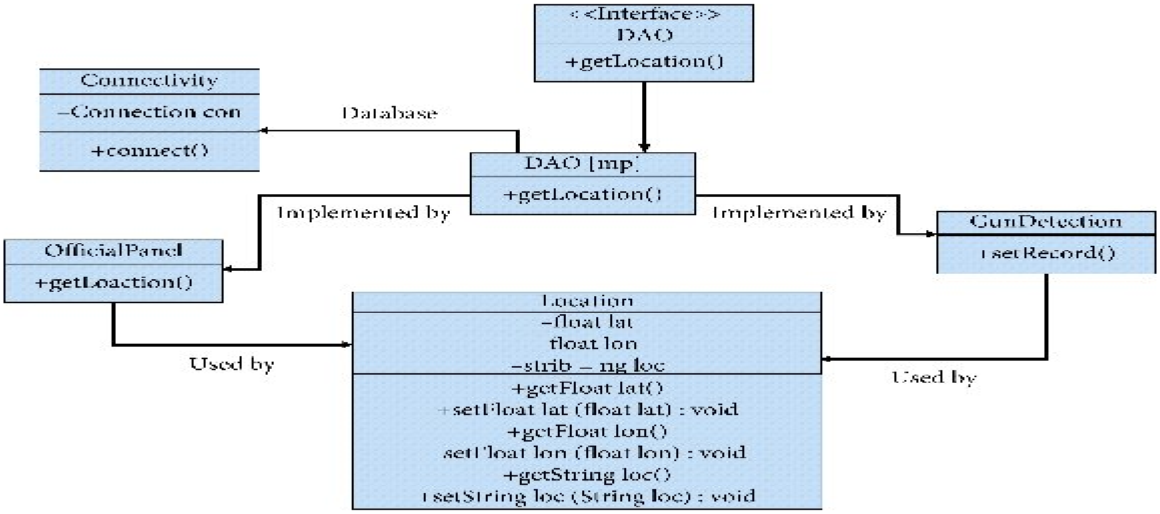

Figure 3: Image presenting the recorded database for object detection

2.1.4 Class diagram for DOA layer.

My system was further compared with the existing literature in Table 2. The system includes CNN-based VGG-16 architecture as feature extractor, followed by state-of-the-art classifiers which were implemented on a standard gun database. Researchers investigated four machine learning models, namely, BoW, HOG + SVM, CNN, and Alexnet + SVM, to recognize the firearms and knifes from a dataset of images. Their work suggests that pretrained Alexnet + SVM performed the best. As it was evident from the previous studies, researchers have widely applied CNN and its variant for weapon or knife identification from CCTV videos. It was obvious from Table 2 that the implemented YOLO v4 outperforms the rest of the other models.

2.1.5. Comparison with the existing work.

Table 2: Algorithm comparison with existing work

S. no	Models	Dataset	Accuracy (%)	
1	My trained model YOLO V4	Image dataset collected for current research	98.89	
2	Alexnet + SVM	Gun video database		95
4	Faster RCNN		Streaming video	95.4

5	CNN VGG-16		IMDB	93.1

Conclusion

In this study, the state-of-the-art YOLO V4 object detection model was implemented and trained over my collected data-set for weapon detection. I created a model that provided a visionary sense to a machine or robot to identify the unsafe weapon and could also alert the human administrator when a gun or a firearm was obvious in the edge. The experimental results show that the trained YOLO V4 had better performance compared to the YOLO V2 model and was less expensive computationally. There was an immediate need to update the current surveillance capabilities with improved resources to support monitoring the effectiveness of human operators. Smart surveillance systems would fully replace current infrastructure with the growing availability of low-cost storage, video infrastructure, and better video processing technologies. Eventually, the digital monitoring systems in terms of robots would fully replace current surveillance systems with the growing availability of cheap computing, video infrastructure, high-end technology, and better video processing.

2.2 Related works

Hu et al. has contributed in detecting various objects in traffic scenes by presenting a method which detected the objects in three steps. Initially, it detected the objects, recognizes the objects, and finally tracks the objects in motion by mainly targeting three classes of different objects including cars, cyclists, and traffic signs. Therefore, all the objects were detected using single learning-based detection framework consisting of dense feature extractor and trimodal class detection. Additionally, dense features were extracted and shared with the rest of detectors which headed to be faster in speed that further needed to be evaluated in testing phase. Therefore, intraclass variation of objects was proposed for object sub categorization with competitive performance on several datasets.

Grega et al. presented an algorithm which automatically detects knives and firearms in CCTV image and alerts the security guard or operator. Majorly, focusing on limiting false alarms and providing a real-time application where specificity of the algorithm was 94.93% and sensitivity was 81.18% for knife detection. Moreover, specificity for fire alarm system was 96.69% and sensitivity was 35.98% for different objects in the video. Mousavi et al. carried out video classifier also referred to as the Histogram of Directed Tracklets which identifies irregular conditions in complex scenes. In comparison to traditional approaches using optical flow which only measured edge features from two subsequent frames, descriptors had been developing over long-range motion projections called tracklets. Spatiotemporal cuboid footage sequences are statistically gathered on the tracklets that moved through them.

Ji et al. developed a system for security footage which automatically identifies the human behavior using convolution neural nets (CNNs) by forming deep learning model which operates directly on the raw inputs. Therefore, 3D CNN model for classification requires the regularization of outputs with high-level characteristics to increase efficiency and integrating the observations of a variety of various models.

Pang et al. presented real-time concealed various object detection under human dress. Metallic guns on human skeleton were used for passive millimeter wave imagery which relied on YOLO algorithm on dataset of small scale. Subsequently, comparison was undertaken between Single MultiBox Detector algorithm, YOLOv4-13, SSD-VGG16, and YOLOv4-53 on PMMW data-set. Moreover, the weapon detection accuracy computed 36 frames per second of detection speed and 95% mean average precision. Warsi A et al. have contributed to automatically detecting the handgun in visual surveillance by implementing YOLO V4 algorithm with Faster Region-Based CNN (RCNN) by differentiating the number of false negatives and false positives, thus, taking real-time images and incorporating with Image Net data-set then training it using YOLO V4 algorithm. They had compared Faster RCNN to YOLO V4 using four different videos and as a result YOLO V4 imparted faster speed in real-time environment.

CHAPTER 3

SYSTEM METHODOLOGY

3.1 Introduction.

Methodology is the combination of best practices, procedure, rules and guidelines of the specific field of science and art by which professionals, specialists and researchers can conduct their projects, research, development and project activities. This chapter describes the methodology used to carry out the research. It presents different fact-finding techniques that were used for achieving the goals and objectives of the project such as requirement of a typical management of details of Weapon detection system, data collection and analysis, system analysis, system design and implementation, system testing and validation.

3.2 Software Development Methodology

In this work, I have developed an integrated framework for reconnaissance security that distinguishes the weapons progressively, if identification is positively true it cautions/briefs the security personals to handle the circumstance by arriving at the place of the incident through IP cameras. I created a model that provides a visionary sense to a machine to identify the unsafe weapon and can also alert the human administrator when a gun or firearm is obvious in the edge. Moreover, I programmed entryways locking framework when the shooter seems to carry appalling weapon. On the off chance conceivable, through IP webcams I can likewise share the live photo to approach security personals to make the move in the meantime. Also, I constructed the information system for recording all the exercises to convey impact activities in the metropolitan territories for a future crisis. This further ends up in designing the database for recording all the activities in order to take prompt actions for future emergency.

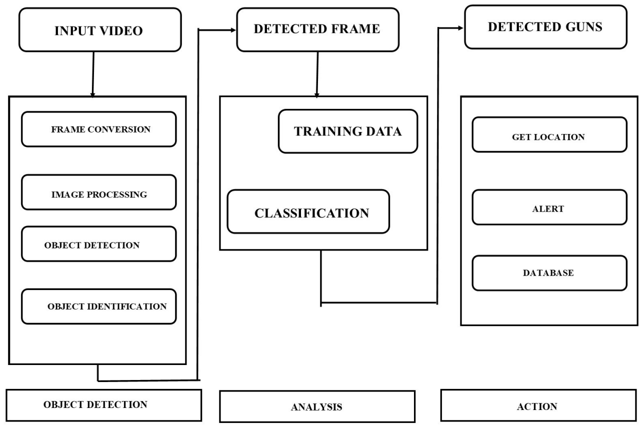

Figure 4: Generalized approach of my research work

The most important and crucial part of any application is to have a desired and suitable dataset in order to train the machine learning models. Therefore, I manually collected huge number of images from Google. A few of the image samples are shown in Figure 2. For each weapon class, I collected at least 50 images. Using google-images-download is one of the best ways to collect images for constructing one's own dataset. I further saved those images to a folder called "images." One must save images in ".jpg" form; if the images are in different extensions, it is a little troublesome andgenerates errors when provided for training. Alternatively, since the images are processed in terms of batches, therefore prior to training, the sizes of all the images are transformed into the same width and height 416 × 416 pixels.

- 

Figure 5: Handgun

Figure 6: Handgun

3.2.1 Sample images from collected dataset

Object detection is primarily related to computer vision that includes distinguishing objects in computerized images. Object detection is a domain that has benefited immensely from the recent advancements in the realm of deep learning. YOLO is basically a pretrained object detector. It is a CNN model. A CNN is a deep learning algorithm which can take in a raw input image and assign learnable weights and biases to various aspects/objects in the image. A convolution layer in CNN model is responsible of extracting the high-level features such as edges, from the input image. This works by applying kxk filter known as kernel repeatedly over raw image.

This further resulted in activation maps or feature maps. These feature maps were the presence of detected features from the given input. Thus, the preprocessing required is much lower as compared to other classification algorithms, whereas in standard approach, filters are hand-engineered and in CNN these are learned through a number of iterations and training. Figure 7 indicates a basic CNN architecture as classification model for 10 different weapons. Subsequently, the next layer is Max-Pooling or Sub sampling layer, which is responsible for reducing the spatial size of the convolved features. This is to decrease the computational power required to process the data through dimensionality reduction. ReLU is a rectified linear unit activation, which is related to the feature of no saturating activation. It eliminates undesirable values from an activation map effectively by setting them to nil. Finally, the last layers are fully connected layers transforming the data into a 1-dimensional array. To create a particular long feature vector, the flattened output is fed to a feed forward neural network and back propagation was applied to every iteration of training. These layers are liable to learn nonlinear combinations of the high-level features as represented by the output of the convolution layer.

•

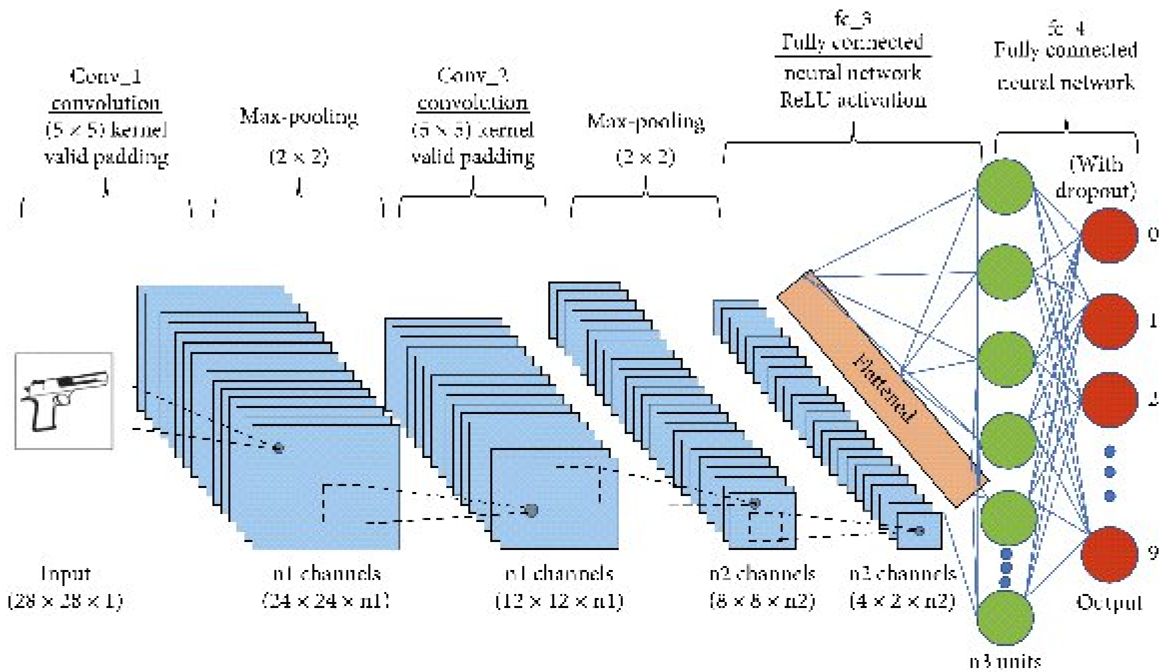

Figure 7: Convolution neural network architecture (CNN)

3.2.2 Feed forward convolution neural network (CNN)

As mentioned earlier that YOLO is a pretrained object detector, a pretrained model simply means that another dataset has been trained on it. It is extremely time consuming to train a model from scratch; it can take weeks or a month to complete the training step. A pretrained model has already seen tons of objects and knows how each of them must be classified. The weights in the abovementioned pretrained model have been obtained by training the network on COCO and ImageNet dataset. Thus, it can only detect objects belonging to the classes present in the dataset used to train the network. It uses Darknet-53 as the backbone network for feature extraction and uses three scale predictions. The DarkNet-53 is again convolution neural network that has 53 layers as elucidated in Figure 4. DarkNet-53 is a fully convolution neural network. Pooling layer is replaced with a convolution operation with stride 2. Furthermore, residual units are applied to avoid the gradient dispersion.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3/2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3/2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3/2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3/2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3/2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 8: Feed forward convolution neural network

3.2.3 Architectural details of DARKNET-53 layer

Initially, CNN architectures were quite linear. Recently, numerous variations are introduced, for example, middle blocks, skip connections, and aggregations of data between layers. These network models have already acquired rich feature representations by getting trained over a wide range of images. Thus, selecting a pretrained network and using it as a starting point to learn a new task is a concept behind transfer learning. In order to recognize the weapons, I took the weights of a pretrained model and trained another YOLO V4 model.

YOLO V4 is designed to be a multiscale detector rather than image classifier. Therefore, for object detection, classification head is replaced by appending a detection head to this architecture. Henceforth, the output is vector with the bounding box coordinates and probability classes. YOLO V4 inherits Darknet-53 as its backbone, a framework to train neural networks with 53 layers as indicated in Figure 4. Moreover, for object detection task additional 53 layers are stacked over it, accumulating to a total of a 106-layer fully convolutional architecture. Due to its multiscale feature fusion layers, YOLO V4 uses 3 feature maps of different scales for target detection as shown in Figure 5.

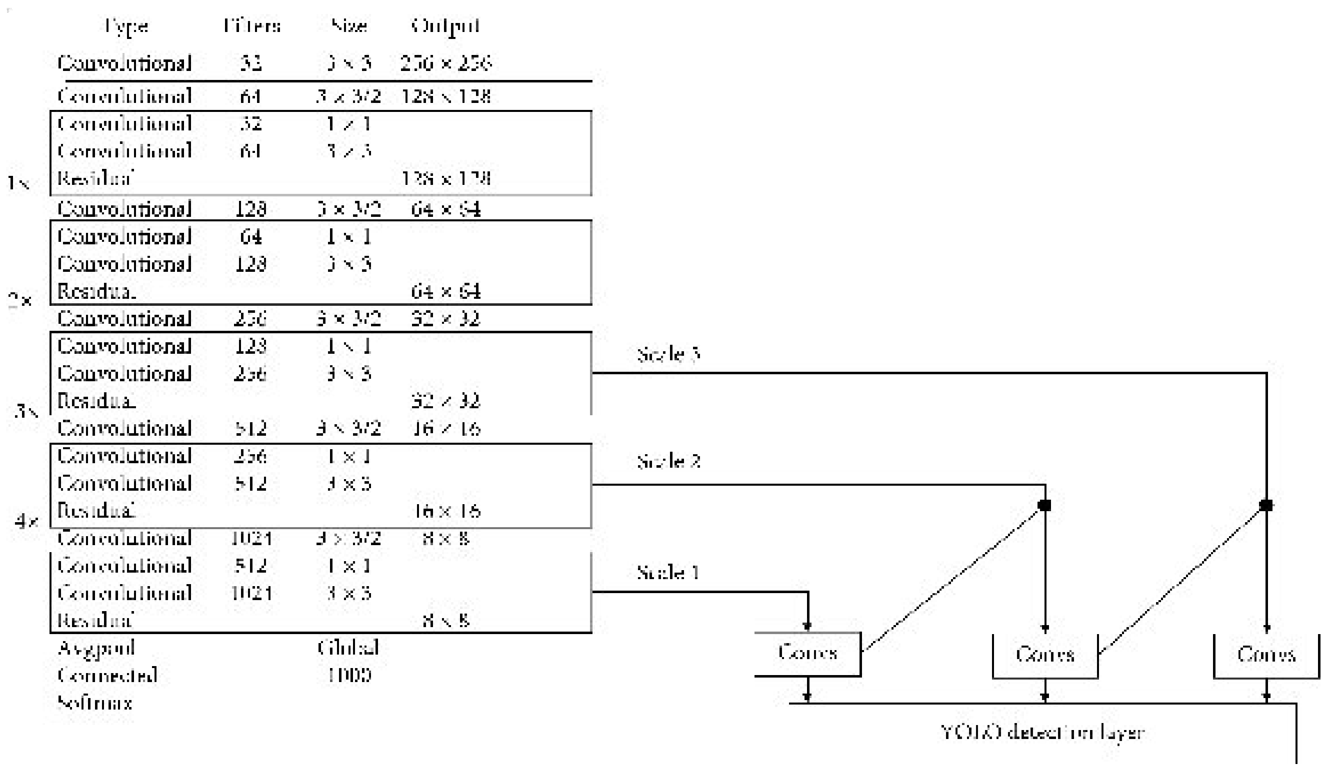

Figure 9: DARKNET-53 layer Architecture

3.2.4 Architectural description of YOLO v4

YOLOv4 makes use of several innovative features that work together to optimize its performance. These include Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross Mini-Batch Normalization (CmBN), Self-adversarial-training (SAT), Mish-activation, Mosaic data augmentation, Drop Block regularization, and CIoU loss. These features are combined to achieve state-of-the-art results.A typical object detector is composed of several parts including the input, the backbone, the neck, and the head. The backbone of YOLOv4 is pre-trained on ImageNet and is used to predict classes and bounding boxes of objects. The backbone could be from several models including VGG, ResNet, ResNeXt, or DenseNetin object detection.

The architecture of YOLOv4 includes CSPDarknet53 as the backbone, PANet as the neck, and YOLOv4 as the The neck part of the detector is used to collect feature maps from different stages and usually includes several bottom-up paths and several top-down paths. The head part is what is used to make the final object detections and classifications. YOLOv4 is designed for optimal speed and accuracy detection head. This design allows YOLOv4 to perform object detection at an impressive speed, making it suitable for real-time applications. YOLOv4 also excels in accuracy, achieving state-of-the-art results in object detection benchmarks.

3.2.5 Experimental Results

Image classification includes, for example, the class of one object in a picture. However, object localization is to recognize the area of at least one article in a picture and drawing a proliferating box around their degree as shown in Figure 10. Moreover, Figure 11 illustrates the detection of rifle from an animated video. The shape of the detection kernel is computed by $1 \times 1 \times (bb \text{ x } (4 + 1 + nc))$. Hence, bb is the number of bounding boxes, “4” is for the 4 bounding box coordinate positions and 1 is object confidence, and nc is the number of classes. The down sampling of the input image is for three scale predictions and is computed by strides 32, 16, and 8. The loss function over here is comprised on three sections, location error (Lbox), confidence error (Lcls), and classification error (Lobj).

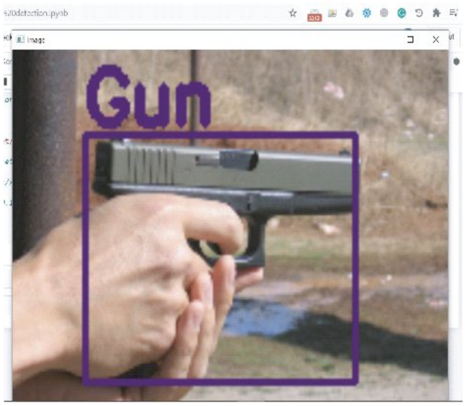

Figure 10: Handgun with a bounding box

3.2.6 Bounding box around detected object; weapon category GUN.

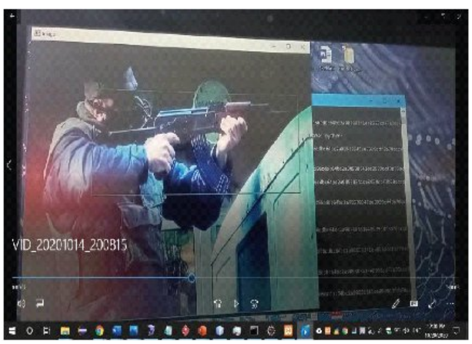

Figure 11: Rifle detection image

3.3 System Analysis

According to Waldo (2006), the activity of system design involves identifying ways to down a large problem by breaking it down into small components which are again decomposed into smaller components. Each of the smaller components can be solved as a single unit, and therefore this is what is referred to as system design. Once components of the system have been identified, it follows that system requirements can be drawn from each of these components which allows a software engineer to envision the functionality and come up with the design of the system under study. Some of the steps followed in system analysis and design include feasibility study, investigation of the current environment, and definition of requirements, technical system options, logical design and physical design (Ramakrishnan, 2012).

3.3.1 Research Design

Research design helps the researcher to organize ideas in a format for collecting and analyzing variables to answer research questions. Research design is the arrangement of conditions for collection and analysis of data in a manner that aims to combine relevance to the research purpose with the research (Tobergte and Curtis, 2013). For this project, the research designconsiders the variables concerning to the baseline, current status of the organization and the future needs of the Weapon detection system to analyze the real-time information to address the need and help the organization for taking appropriate and responsive measures while planning for the future.

3.3.2 Location of the study

The study is carried out in Nairobi and Kajiado County, more specifically at Kajiado town and Kitengela and this location is chosen due to the proximity to the respondents.

3.3.3 Target Population

As defined by (Mugenda and Mugenda, 2003), Population is as a complete set of individual cases or objects with some common observable characteristics, it is the population to which the researcher wants to generalize the results of the study.
The target population is the residents and Citizens around Kajiado and Nairobi. The population also targeted was the authorities and county administrators as well as security personnel.

3.3.4 Sample Selection

Sampling is the process of selecting a few (a sample) from a bigger group (the sampling population) to become the basis for estimating or predicting the prevalence of an unknown piece of information, situation or outcome regarding the bigger group. A sample is a subgroup of the population the researcher is interested in (Kumar, 2015)
A sample design is a definite plan for obtaining a sample from a given population. It refers to the technique or the procedure the researcher would adopt in selecting items for the sample. Sample design may as well lay down the number of items to be included in the sample i.e. the size of the sample. Probability sampling and purposive sampling were employed. Probability sampling is the type of sampling where every item within the sample has equal chances of inclusion in the sample and has no chance of appearing again in the sample once selected whereas purposive sampling

where items are selected deliberately by the researcher (C R Kothari, 2015). A group of the county security personnel considered to participate in the study where they attempt to use the system to detect unlicensed weapons. Furthermore, they try to figure out the actual location of the weapons with the help of the system.

3.4 System Design

System Design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system System Design - SEBoK, n.d.)). The following are the designs that are be used:

3.4.1 Data Flow Diagram

Data flow diagrams define business processes by illustrating processes, data flow, data stores and external entities through the use of graphical symbols. Dennis et al (2012). This illustrated major processes, data stores and inter-related data flows which comprises of the system under construction. Each of the major processes are decomposed to give more details on the movement of data and how some processes interact with data stores.

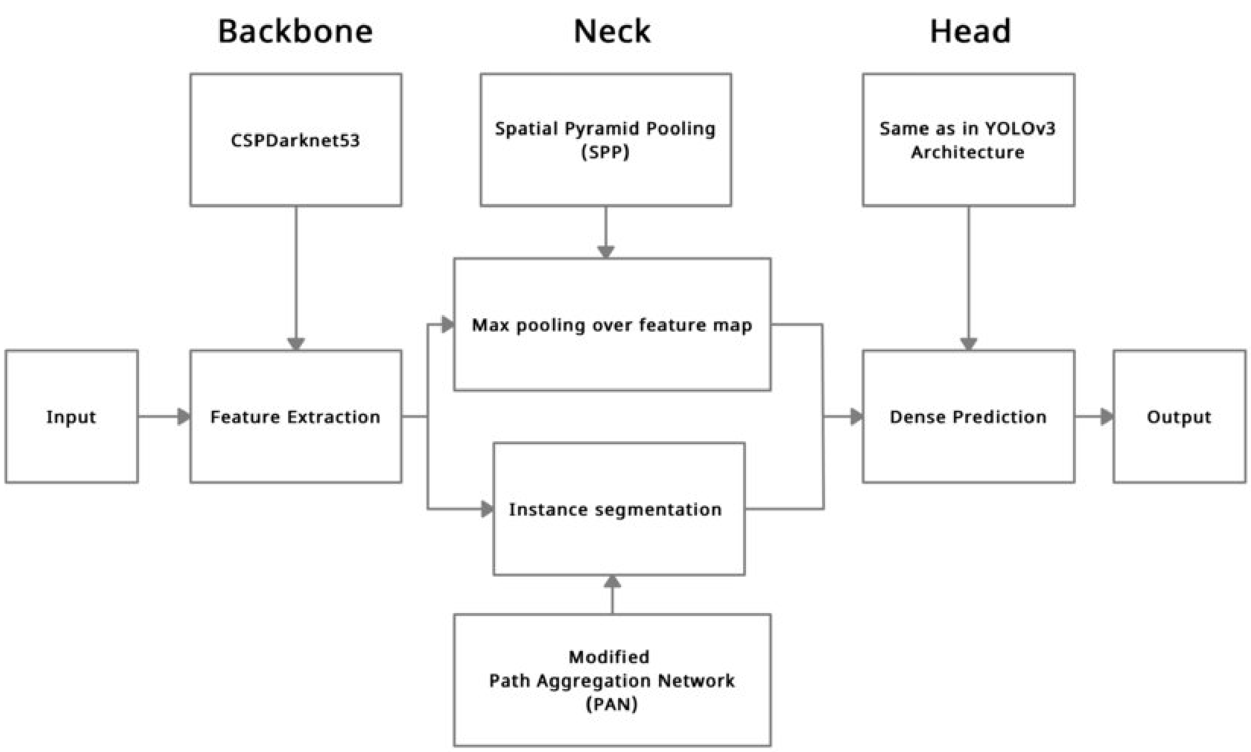

Figure 12: Dataflow diagram (DFD)

3.4.2 Use Case Diagram

A use case is a specified functionality that shows activities that are performed to give a specific result. It follows that a use case diagram shows the relationship between actors of a system and different use cases. Kostoglou et al (2014). The use case diagram illustrates possible set of interactions between the system and it’s actors.

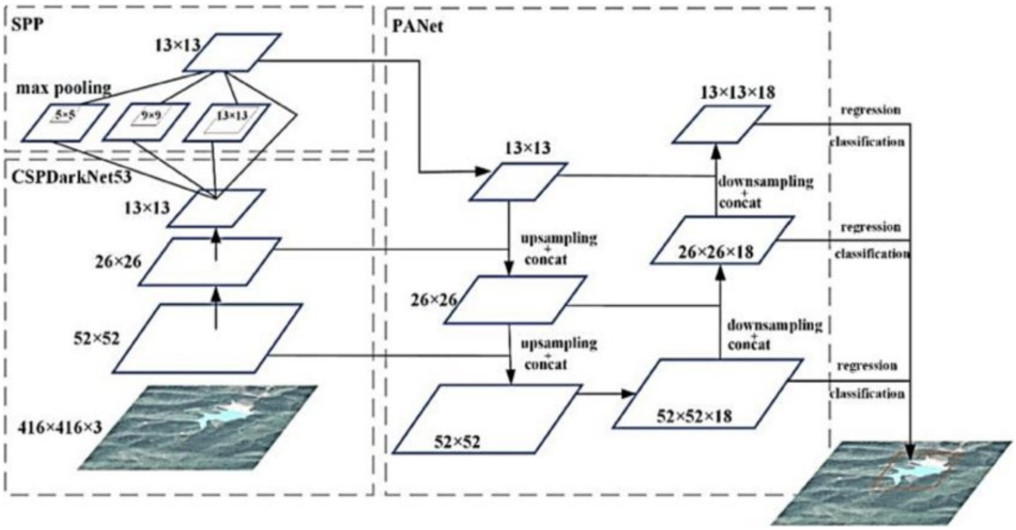

Figure 13: Use Case Diagram

3.4.3 System Sequence Diagram

A sequence diagram shows objects that compromise of a use case and the messages that pass between them for that particular use case. Dennis et al (2012). The system sequence diagram shows the major objects within the system and how information flows between them as well as the messages exchanged between them.

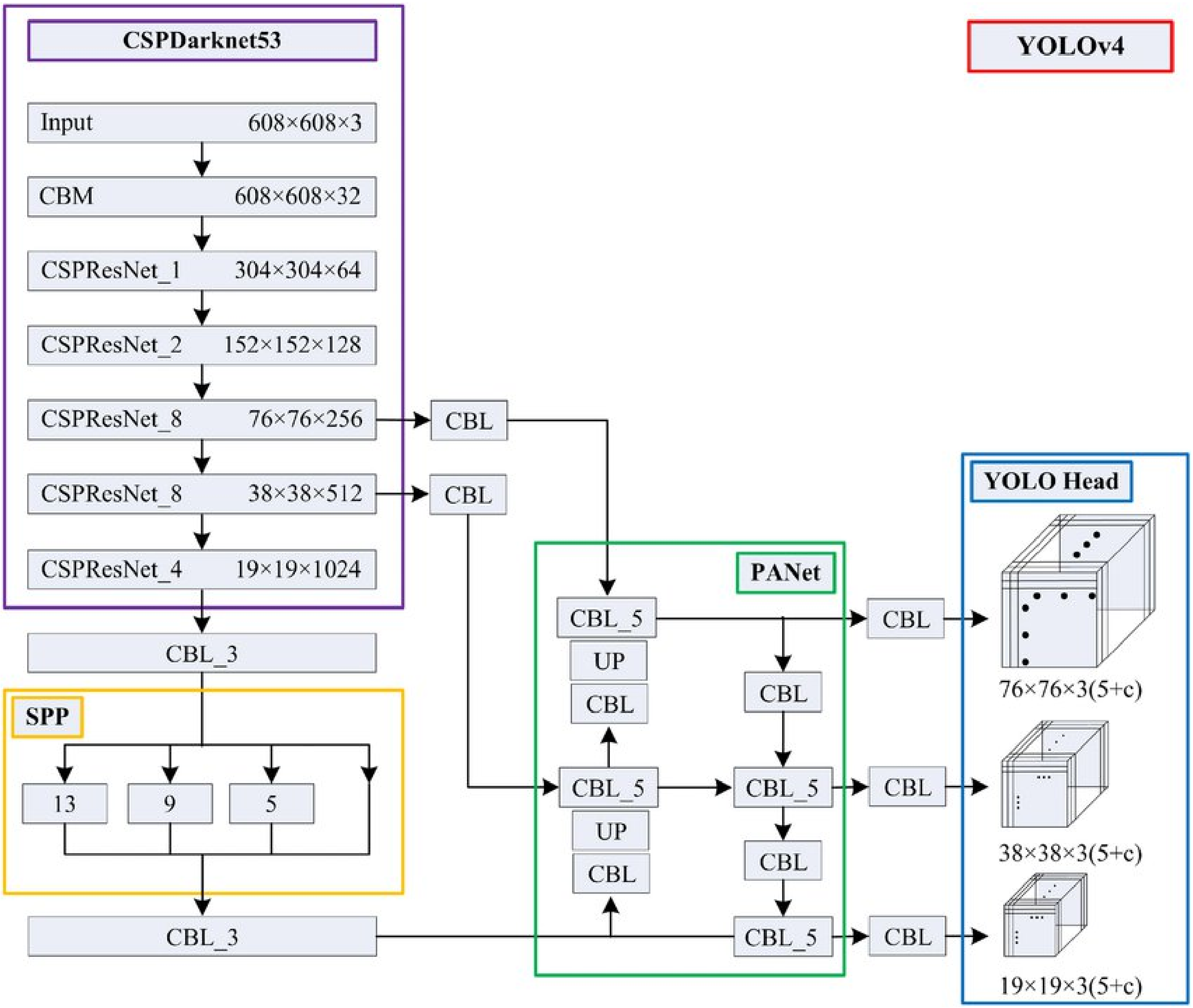

Figure 14: System Sequence Diagram

3.4.4 Class Diagram

Class diagrams present a set of classes, interfaces and associations which present an object-oriented view of the system under discussion. Kostoglou et al (2014). The class diagram provides an overview of the system by describing objects and classes within the system as well as relations between them. It also shows visibility of the different classes in relation to others in the model.

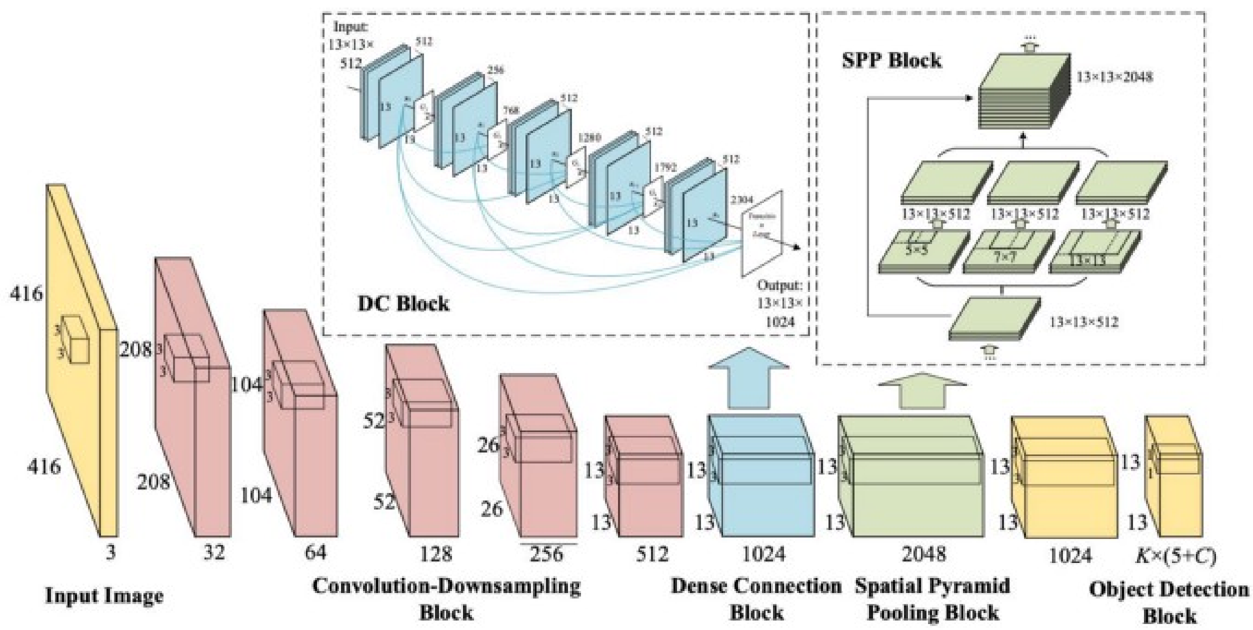

Figure 15: Class Diagram

3.5 Feasibility Study

Feasibility study is a process, in order to discover the strengths, weaknesses, opportunities and threats of a proposed system to full fill the main business needs. Feasibility studies are used to determine whether an intervention is appropriate for further testing; in other words, they enable researchers to assess whether or not the ideas and findings can be shaped to be relevant and sustainable (Bowen et.al, 2009). A detailed study of feasibility has been carried out regarding this system following facts.

3.5.1 Legal Feasibility

Since the system is monitoring other entities to detect unlicensed weapons, a detailed legal due diligence of the system has been done to analyze and ensure it is in accordance to the security policy of Kenya.

3.5.2 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of the system development. The proposed system functions were analyzed to see whether they accomplish the business requirements and all the functional and non-functional requirements can be fulfilled by this system. Hence my system is operationally feasible.

3.5.3 Technical Feasibility

The purpose of the technical feasibility is to determine whether the company or the project development team has the technical expertise to handle completion of the project. The system is developed using Python as the main development language and Django is be used on the back-end due to high reliability, flexibility and security. MySQL database is chosen to be the DBMS of the application. The chosen technologies are broadly used in software development sector and are available for free use. These technologies are all open-source software and doesn't require any registering or purchasing of any kind. So, this project is technically feasible.

3.5.4 Economic Feasibility.

The purpose of an economic feasibility study is to determine the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency, other state agencies, and the general public as a whole. It includes quantification and identification of all the benefits expected. The estimated cost for this project development is the cost of the following:

i. Cost invested on the technical expertise to develop the software.
ii. Cost of the tools required to develop the software
    a Operating system
    b Additional software
iii. Software Installation

1. Server to install the software
2. Training cost
    i. Operational Cost.
    ii. Software maintenance cost.
    iii. Software upgrading cost.

3.6 Data collection Method

The data was performed from the primary as well as the secondary data sources through observation, review of secondary data and reports. The physical processes under investigation are viewed from different perspective point of views such as theoretical, practical, theories, etc. The dataset for training the YOLOv4 model consists of gun related photos obtained from Kaggle.com.

Using the above-mentioned techniques, the data collection processes is carried out in different phases from the multiple sources of data using quantitatively and qualitatively. The details of each data collection tools used as stated as follow.

3.6.1 Observation

One of the instruments that the researcher used to collect the necessary data is direct observation. It is a planned, systematic and selective way of watching and listening to activities as the actual working processes takes place. The researcher used this tool to obtain information about current functional process system of the different security systems. Moreover, it helps to crosscheck the views that are given by the respondents in the questionnaire and interview. 10 direct observations with sub sections were observed by using the following item checklist: Security system's organizational structure, Security agencies' layout, Quality standards and control activities, Human resource and development, Information Technology system, setting standard time application for every process, Management Capabilities, Store management. Therefore, the project carried out this tool to check out the collected data to eliminate the bias and to observe what is currently happening in the case of the social security sector.

3.6.2. Review of secondary data and reports

Secondary data include both quantitative and qualitative data and they are being used principally in both descriptive and explanatory research. I examined the existing documents, data from files, forms, journals, books and magazines to provide additional information where appropriate with the concept of competitiveness. Besides, variety of published and/or unpublished systems and the documents, reports and newsletters were reviewed to make the project fruitful. Sometimes records are believable than what people say and thus I decided that it would be good to get facts from the documents as way to counter check what had been interviewed. The records reviewed included: written policy documents, rules and regulations, standard operations, procedures used in the social security systems.

3.7 System Implementation

The system comprised of a weapon detection system that would track all unlicensed weapons and figure out their actual locations. The weapon detection system is based on Machine learning and object detection. For successful implementation, the minimum requirements of the computing device on which the application is developed were:

3.7.1 System Component Requirements

i. RAM 16GB minimum
ii. Disk Space 500GB
iii. Processor–Intel CORE i5

iv. Python 3
v. Google Colab.
vi. Qt Designer App.
vii. Postman App.
viii. MySQL Database.
ix. Google Drive.

3.7.2 System Development Tools and Techniques

This refers to the tools that are used to ensure that the system is created in an efficient manner. The following are the tools:

3.7.2.1 PyCharm

This is the code editor in which both the client-side and the server-side code is written. It contains sufficient features for running python code.

3.7.2.2 MySQL Database

The database consists of tables that stores records implemented using MySQL database. The reason behind using MySQL database is that it is fast and easy to configure. In addition, it can store a very large record and requires little configuration.

3.7.2.3 Django

This is the python framework that is used to write the backend of the system.

3.7.2.4 Postman

This is the Api software that is used to test the back-end functionality of the system by sending Api with data to the backend.

3.7.2.5 Qt Designer

This is the software that is used to design the user interface for the system's client-side application. The user interfaces to be designed include: login window, settings window and detection window.

3.7.2.6 Google Colab

This is the software is used to train the dataset for the Model. It is convenient since it has a built in GPU which makes it way faster than a CPU.

3.7.2.7 Google Drive

Once the dataset has been trained, it needs to be backed up and Google drive is be a powerful platform to back up trained dataset.

3.8 Proposed Modules and System Architecture

The following are the modules for the proposed system:

a. **Administrator Module-** The administrator has full control of the application and oversees all operations in the system.

b. **User Module-** This is a module that allows authorized personnel such as police to use the system in the weapon detection process.

3.9 System Testing

System testing is an integral phase of the software development lifecycle as it is the procedure through which quality of the system is assessed. It is the activity of evaluating the system to determine whether it meets the requirements which were specified during the requirements determination phase (Hooda &Chhillar, 2015). Testing is performed throughout the entire process of software development. The types of testing strategies which were applied to this research are as follows:

1. Functional Testing – this involved the selection of test cases based on requirements that were specified before implementation began. This is because the main quality factor in software is that it meets the required functionality and behavior (Hooda &Chhillar, 2015). Each software unit is individually tested and thereafter integrated units were tested to ensure that communication between the different modules occurred as expected. Finally, both the developer and system users were involved in performing system testing where they interacted with the entire system and analyzed the functionality which it provided. The developer is keener on ensuring that all software bugs were resolved so that the application is stable.
2. Compatibility Testing – this is conducted to ensure that the software created could work well in different platforms and without compatibility issues.

3. Usability Testing – this evaluated the extent to which users could learn to use the system. This involved observing the human interaction between a test group and the system. A survey is also be administered following the user testing phase to obtain feedback from the test group.

3.10 Validation

Validation is the process through which a system is evaluated to ensure that its features have been built to meet the requirements which were set at the beginning of implementation (Williams, 2006). Validation of the system is done by the use of functional testing procedures to ensure that the system performed based on the requirements that had been set. These procedures were unit testing to test each individual component, integration testing to test different linked components and system testing to test the overall system with all interlinked components working together. During the testing phase of the system, the researcher is keen to obtain feedback on how the features of the application solve the security problem.

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

This chapter presents the overall design structure of the proposed system. The analysis part entails presenting the functional and non-functional requirements of the system. Thereafter, design diagrams are presented to illustrate the interactions between users of the system through the use of diagrams. Different components of the system are illustrated as well as how they contribute in solving the security related problem.

4.2 System Analysis

System analysis refers to the process of studying procedure or a business to identify its goals so as to be able to create successful systems in an efficient way.

4.2.1.Requirements Determination

The process of requirements determination involves breaking down a high-level statement of software requirements into finer detail, stating what a system should perform in order to deliver the needed functionality to its users. Requirements address not only the functionality of a system, but also non-functional issues which affect the usability of a system (Brackett, 1990). In this regard, Dennis et al. (2012) define a requirement as a characteristic which a system must have.

4.2.1.1Functional Requirements

A functional requirement is a specification of a function or capability that a system should perform for its users (Brackett, 1990; Dennis, Wixom, & Roth, 2012). The functional requirements which were identified by the respondents of the study indicated the following:

User registration – the system should allow for registration of new users.

Login/logout – registered users should be able to access both the system application and web backend by way of their user credentials.

4.2.1.2 Non-functional Requirements

Non-functional requirements are those which relate to the operation and performance of a system

(Glinz, 2011). The requirements identified for this system are:

a) Intuitive application design – the design of the screens should allow for easy use by users of the system.

b) Availability – the weapon detection system requires an internet connection to provide full application functionality.

c) Feedback – the system should provide feedback to users so that they can understand what process is happening in the background.

d) Performance and reliability – the application should stay live at all times to allow users access at their convenience.

4.3 System Design

System design is the process of designing elements of system architecture, modules and components, the different interfaces of the components and the data that goes through that system. The following are the design diagrams:

4.3.1 Use Case Diagram

The use case diagram shows possible interactions between the different actors. In this case, the different actors are the county administrator, security personnel and citizens. Each actor plays a different role. The county administrator is responsible for overseeing operations. Security personnel on the other hand is responsible for operating the system to detect unlicensed weapon and bring the suspects involved into custody. Citizens are the reason for the whole project. To provide them security from criminals and prevent mass killings through shootings.

CHAPTER 5

THE SYSTEM CONSTRUCTION

5.1 Introduction

In the journey of the Weapon Detection System's development, Chapter Six stands as a phase, where the app evolves from a concept to a tangible reality through systematic construction and iterative enhancements. This chapter offers insights into the meticulous steps taken to breathe life into the system, consistently refining it based on valuable user feedback and evolving requirements.

5.2 System Development Tools and Techniques

5.2.1 Programming Languages

Adoption the Python programming language as the foundation for development. This decision was influenced by its simplicity and convenience as one is able to execute complex tasks with just a few lines of code. The choice ofPython was aimed to enhance productivity, boost developer satisfaction, and ensure code safety. The modern language features of Python allowed me to focus on expressing my creative ideas while reducing the need for repetitive boilerplate code. Co routines was effectively employed, a feature of Python, to streamline asynchronous programming, making tasksmore efficient and responsive.

5.2.2 PyCharm

To streamline the development process, PyCharm was heavily utilized, the official integrated development environment (IDE) for python applications. This IDE, provided an ideal environment for crafting the system. By integrating Python with Qt Designer Application, there was a swift generation of user interfaces for the client side using intuitive APIs. A standout feature of Pycharm was its capability to implement real-time code and resource changes, significantly expediting the development cycle. PyCharm also allowed for the development of the system's backend using the Django web framework The Google Colab allowed for the dataset training to take place and automatic backup of trained datasets in the Google Drive software. Moreover, Pycharm's smooth incorporation with MySQL database substantially enhanced the system's data management capabilities.

5.2.3 MySQL

MySQL played a pivotal role in this project by offering a comprehensive suite of services to streamline system development, management, and growth. Leveraging MySQL facilitated the system creation process and bolstered security measures. The platform's cross-platform services accommodated web interfaces, extending the system's potential reach.

The authentication service within MySQLin conjunction with the Django web framework introduced user-friendly UI libraries and SDKs, simplifying user authentication processes. Consequently, this significantly minimized the development effort and time required to establish and maintain robust authentication systems.

5.3 The Waterfall Development Process

The development process of the Weapon Detection System employs a waterfall model, commencing with the construction of core features as outlined in the requirements. This foundational version serves as a basis for sequential development. It sets distinct goals or endpoints for each phase of development. Using the waterfall development process has allowed for construction of each phase of the system to completion before moving the next phase.

5.4 Embracing Ongoing Improvement

The Weapon Detection System is a testament to constant evolution, demonstrated through three key principles: user-centric enhancement, flexible adaptation, and harmonizing change and stability. By actively soliciting and acting upon user feedback, the system continuously refines itself to cater to user needs and preferences. The iterative strategy allows seamless adjustments to changing requirements and emerging trends, ensuring the system's sustained relevance. This balance is struck by maintaining core functionalities while accommodating improvements, fostering user familiarity and usability.

5.5 Conclusions

As this chapter concludes, the journey of constructing the Weapon Detection System showcases how it transformed from an idea into a functional and evolving tool. By constructing the system step by step, actively integrating user feedback, and consistently refining it, I ensure that the system remains responsive and impactful for security enhancement. This chapter sets the stage for the subsequent phases, as I continue to enhance and adapt the system to meet the evolving needs of its users.

CHAPTER 6

THE SYSTEM TESTING

6.1 Introduction

In this section, I delve into the core aspects of constructing the Weapon Detection System. My goal is to provide a comprehensive understanding of the system's components and functionalities. I 'll explore the purpose of various sections within the system and highlight the critical role of system testing in determining its success. The primary objective here is to identify and address any potential system failures and defects before its full deployment. Through rigorous testing, I'll assess the system's functionality and its ability to handle a range of tasks effectively.

6.2 System Testing

This segment is dedicated to an in-depth exploration of the Weapon Detection System functionality and performance. My primary focus is on comprehensive System testing to gauge its success and uncover any potential shortcomings. By subjecting the system to rigorous testing, I aim to detect and address system failures or defects before the system's full deployment. This testing phase serves as a critical quality assurance step to ensure the system functions as intended and meets the standards set. It also affords me the opportunity to refine and optimize the system's performance, making it a reliable and effective tool for users, administrators, and companies alike.

6.3Test Plan

Software testing was employed as a systematic procedure to assess the system's functionality and quality. The overarching aim of software testing was to pinpoint any defects, errors, or discrepancies that might arise in the system's behavior. This process focused on comparing the actual outcomes generated by the system against the anticipated results.

6.3.1 Unit Testing

Throughout the development phase, unit testing was conducted on individual components or units of code, performed by the developer. Tests were crafted to validate the functionality of specific functions or methods within the app. Utilizing automated testing frameworks that were compatible with Python, the creation and execution of these tests became straightforward. The process of unit testing played a pivotal role in the identification and resolution of bugs or issues during the early stages of development, ensuring the proper functioning of each component before integration. For instance, a module like Detection is tested to see whether it carried out the full process correctly.

Table 3: Unit Testing

Test	Test Area	Expected Results	Results
Unit Testing	Testing the individual modules of the We apon detection system.	Every module to have the capability to run independently and give the expected feedback	The modules work appropriately and are able to give the expected feedback.

6.3.2 Integration Testing

Following the successful completion of unit testing and confirming the absence of errors within individual components, the focus shifted to integration testing. In this phase, the objective was to scrutinize the interactions among various app units when they were combined. This encompassed the testing of scenarios where multiple components collaborated, simulating real-world usage conditions. The practice of integration testing enabled the identification of potential issues stemming from the interplay between these components, allowing for their resolution before progressing further in the development process.

In this stage, different modules of the system were combined together and tested as a whole. In this process, it verified that the individual components integrated are able to work together and interact well without any conflicts. It is tested that the integrated modules are able to meet the stated user needs.

Table 4: Integration Testing

Test	Test Area	Expected Results	Results

Integration Testing	Module's relationship	Different modules in the Weapon Detection system working together in an expected way.	The modules were able work together and produce the expected results.

6.3.3 Regression Testing

During the development journey, and particularly following modifications such as bug fixes or feature additions, regression testing played a crucial role. Its purpose was to confirm that any alterations introduced into the codebase didn't inadvertently introduce new flaws or adversely affect the existing functionality. This was achieved by rerunning previously conducted tests, enabling rapid identification of any unforeseen issues that might have emerged as a result of recent changes.

6.3.4 System Testing

Approaching the system's final stages, system testing was initiated to assess its performance across diverse operating systems. Multiple Android devices with varying specifications and versions were utilized for app testing. Examination was carried out to understand how the app responded under different circumstances, including distinct screen sizes, orientations, and hardware capabilities. This comprehensive evaluation aimed to ensure uniform user experiences across a wide array of devices and environmental settings.

6.3.5 White box testing

During the system's development, active engagement in white box testing was employed. This approach utilized access to the source code and internal architecture to comprehensively inspect the logic, flow, and structure of the application. Techniques such as code coverage analysis were utilized to ensure the execution and validation of all code components. Furthermore, branch and decision testing were applied to confirm the thorough testing and proper handling of various code paths and conditions.

6.4 *Registration and Login Test Case*

This verifies that the user's login form is working correctly. For it to work, one must have a valid email and password which is acquired during registration.

i. Test procedure - Enter the email and password then click the login button.

ii. Test data - email, password.

iii. Expected results - The user logs in on entering correct email and password.
iv. Actual results - If the user details are correct, the user logs in, otherwise he/she is denied access to the system during login.

This page introduces the user to the system. It allows a user to login into the system. The user selects his/her user type and then enters the user identification number and password. On clicking the login button, they system verifies their credentials and provide the appropriate dashboard. In the event that the user provides wrong details, the system provides the appropriate error messages.

6.5 Software Maintenance

Software maintenance was a key element in ensuring the system's long-term viability, functionality, and adaptability to changing requirements. This encompassed various types of maintenance, each implemented to address specific needs:

6.5.1 Types of maintenance

Corrective maintenance: Continuous monitoring of user feedback and regular testing allowed for the identification and resolution of reported errors or bugs. Swift action was taken to analyze root causes, fix code, and release updates to ensure a seamless user experience, promptly addressing issues like crashes or unexpected behavior.

Adaptive maintenance: With the emergence of new systems, efforts were made to maintain the system's compatibility. Adjustments were made to accommodate different platforms and integrate with new hardware or software, involving modifications to the codebase for smooth operation across diverse environments.

Perfective maintenance: Periodic introductions of new features and enhancements were undertaken to elevate the system's functionality. Existing features were refined, and new capabilities were added in alignment with user preferences, enriching the overall user experience.

Preventive maintenance: A proactive approach was adopted to prevent future issues. Regular code reviews and the implementation of best practices ensured the software's stability. This strategy involved identifying and addressing potential problem areas to minimize the risk of significant challenges arising in the future.

6.6 User Experience Enhancement

Within the context of implementation and testing, this segment illuminates the steps taken to elevate the user experience in the Weapon Detection System. Acknowledging that user satisfaction is intrinsic to successful implementation, I meticulously crafted strategies to ensure an intuitive, engaging, and user centric interface. I proactively addressed how users navigate the software, interact with its features, and engage with its content. By integrating real user feedback and embracing iterative design refinements, I aimed to cultivate an app that not only meets but surpasses user expectations. This enhancement focused approach within the implementation and testing phases reflects my commitment to delivering a high-quality user experience.

6.7 Future Scalability and Expansion

In relation with implementation and testing, this section casts a forward looking on the system's scalability and potential expansion. I recognize that a system's journey extends beyond its initial launch, encompassing growth, evolving user needs, and emerging technologies. Within this context, I delve into my strategic blueprint for scalability and expansion. I detail how the system's architecture has been meticulously fashioned to seamlessly accommodate future enhancements. By adhering to modular coding practices and flexible design principles, I've positioned the system to incorporate updates and expansions without disrupting its core functionality. This forward-thinking mindset underscores my dedication to enduring success and adaptability within the ever-evolving landscape of implementation and testing.

CHAPTER 7

THE SYSTEM IMPLEMENTATION

7.1 Introduction

This chapter delves into the pivotal phase of system implementation for the Weapon Detection System. With the groundwork laid by meticulous planning, this chapter explores the critical step of translating concepts into a functional reality. In particular, I focus on the System Requirements Specifications that form the blueprint for ensuring the system's seamless operation and compatibility.

7.2 System Implementation

System implementation materializes as the bridge between conceptualization and functional reality in the Weapon Detection System process. This crucial phase incorporates the critical use of coding and development, translating design blueprints into tangible features and interactions. The user interface (UI) emerges as an artful fusion of design elements, creating an intuitive and visually captivating platform that resonates with users.

Database integration lends the system its data management prowess, storing detection alert profiles, and vital information securely. Feature implementation breathes life into planned functionalities, ensuring seamless interactions that empower users' feedback and engagement. Rigorous testing scrutinizes every corner for bugs and bottlenecks, while user training resources and support mechanisms ease users into the system's ecosystem. With these efforts, the Weapon Detection System emerges as a testament to meticulous implementation.

7.3 System Requirements Specifications

The Weapon Detection System stands as a testament to technological advancement, requiring a compatible operating system such as windows or Linux. It necessitates a minimum required version as outlined in the system's specifications, ensuring optimal compatibility and utilization of the latest features. The device's processor is of paramount importance; it must meet or surpass the system's stipulated minimum requirements in terms of speed and performance. This ensures that the software operates fluidly, offering users a responsive and engaging experience.

To uphold an uninterrupted user experience, the device's Random Access Memory (RAM) must be adequate. This attribute is crucial in preventing lags or crashes during usage, even under demanding tasks. As the app involves installation and data storage, the device must provide substantial storage capacity. A minimum of 8GB of available storage space is imperative to accommodate the system's components, data, and any future updates.

7.4. Libraries Installation Process

Transitioning beyond the realms of development, this section navigates through the installation of Open Computer Vision library that allows the system to make use of the computer web cam during the detection process. The installation of python PyQt5, Pandas and NumPy libraries that allows me to put together a machine learning model.

Establishment of Virtual environments using PIP was also of great use. Pipenv made a good virtual environment for running both the client and the server side effectively.

7.5. System Snapshots

7.5.1. Client Side

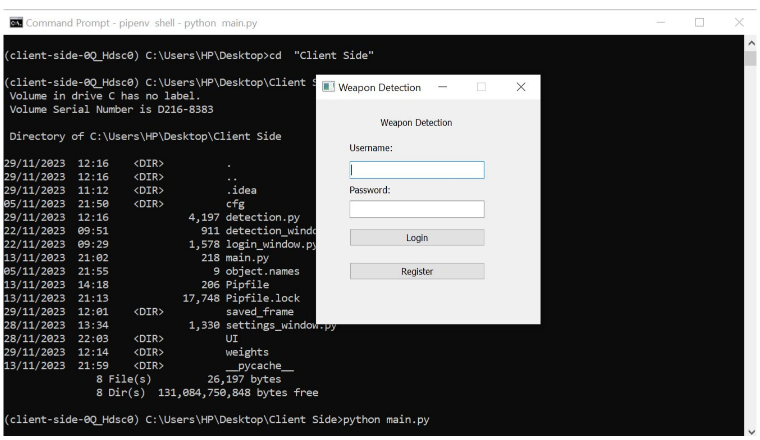

Figure 16: Client Side Login Page

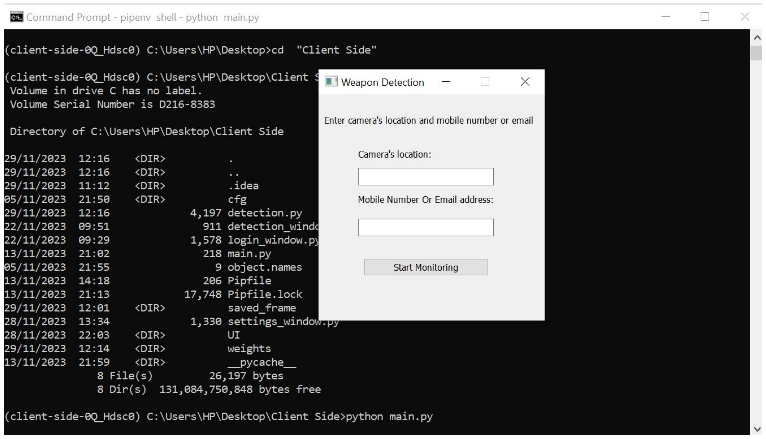

Figure 17: Client Side Location Entry

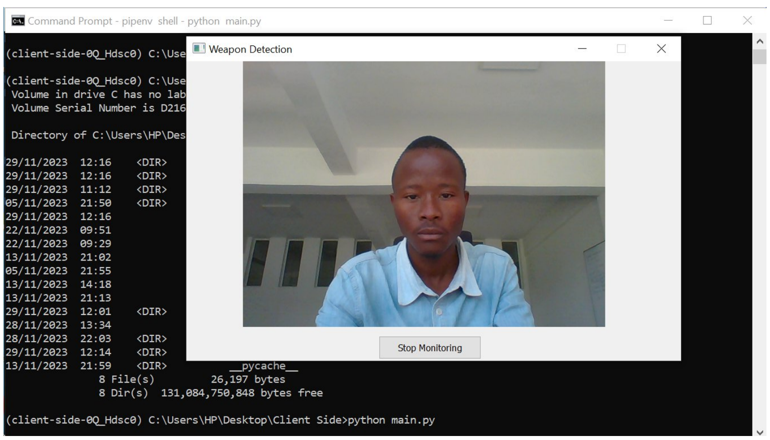

Figure 18: Client Side Detection

7.5.2. Server Side

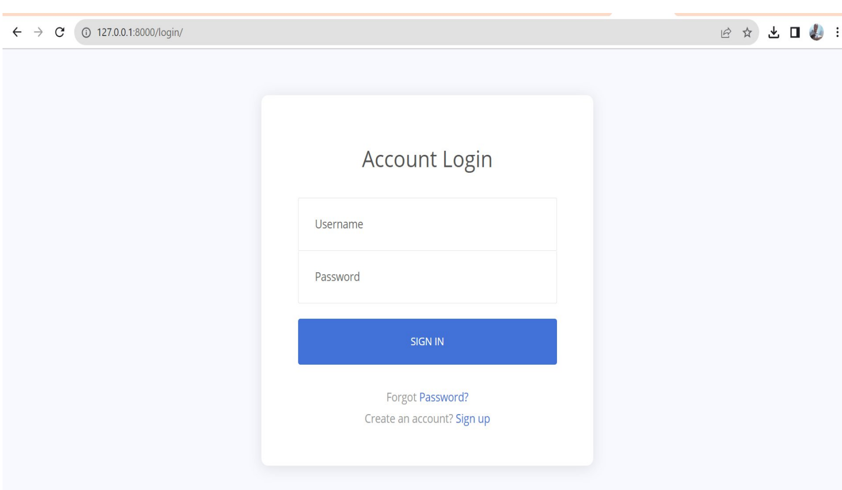

Figure 19: Server Side Sign in page

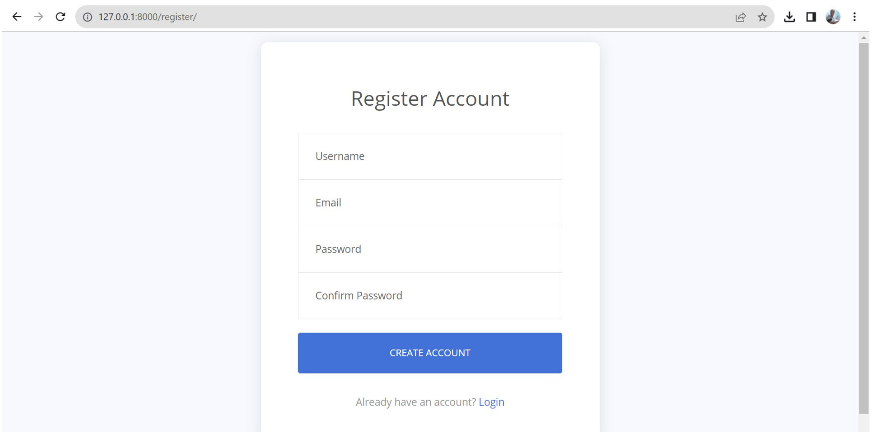

Figure 20: Server Side Registration Page

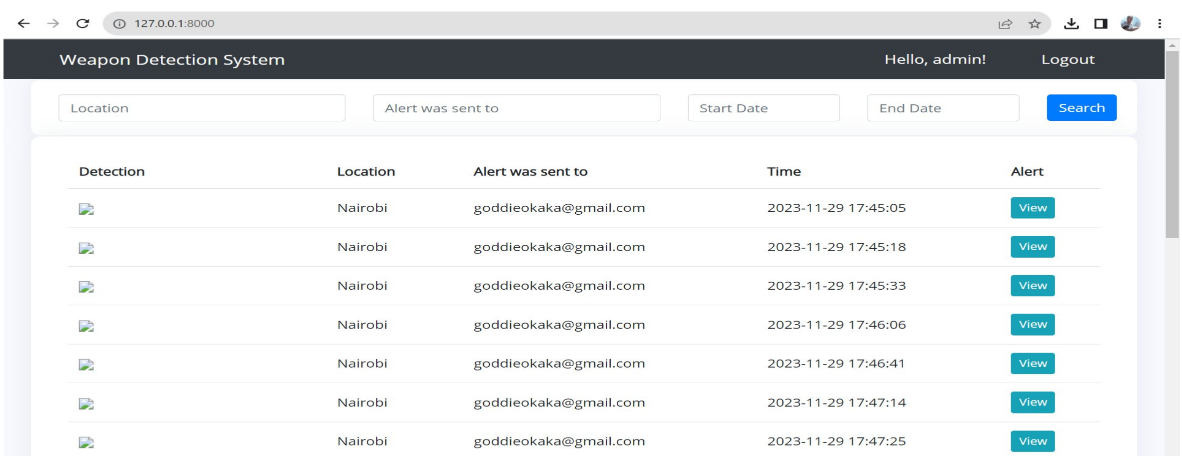

Figure 21: Server Side Alerts Page

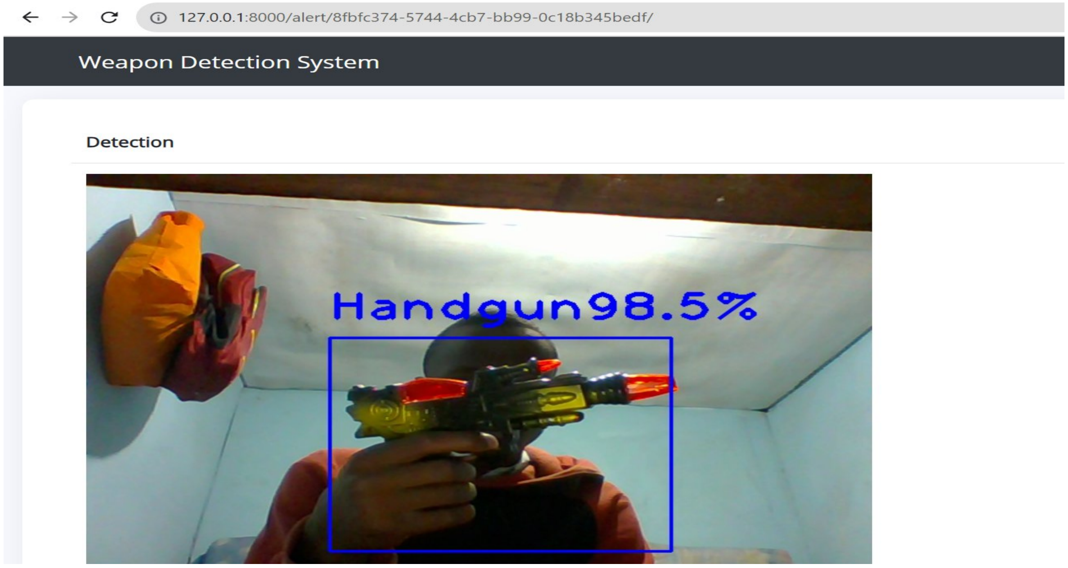

Figure 22: Detection Alert

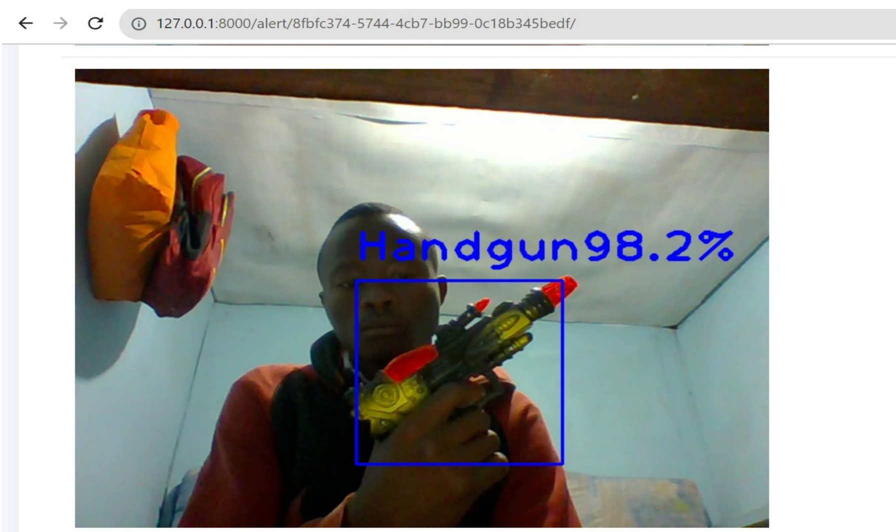

Figure 23: Detection Alert

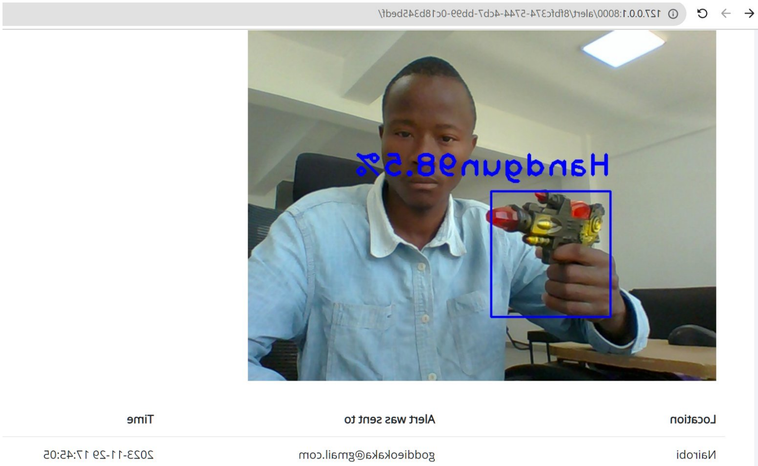

Figure 24: Detection Alert

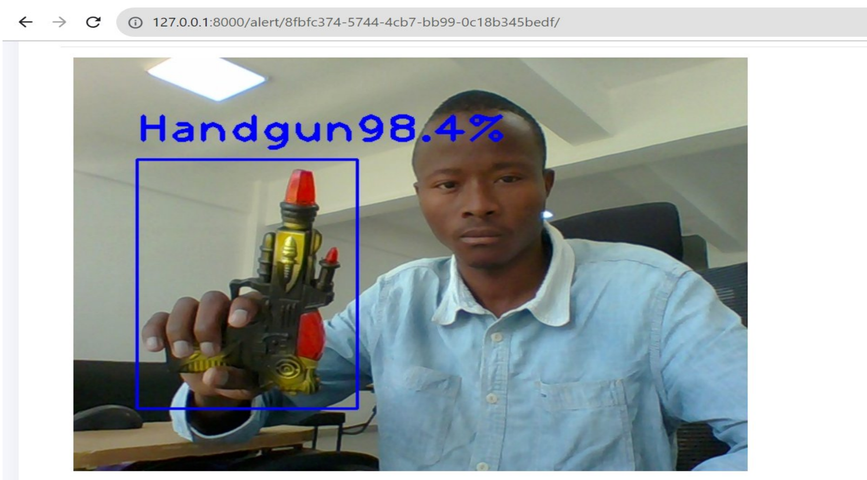

Figure 25: Detection Alert

Figure 26: Detection Alert

7.5.3. Database

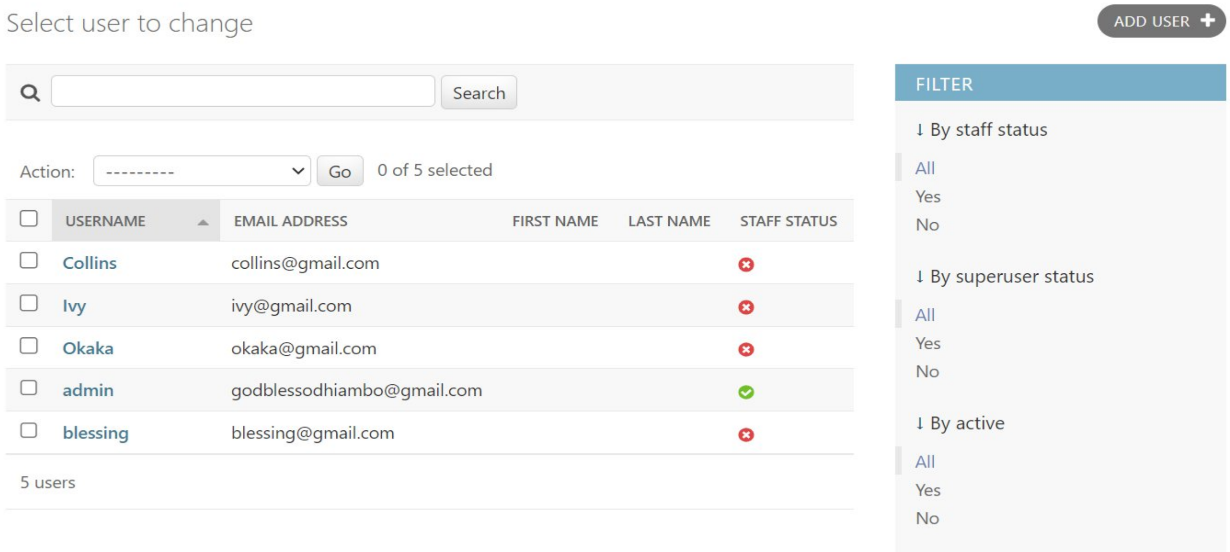

Figure 27: Users

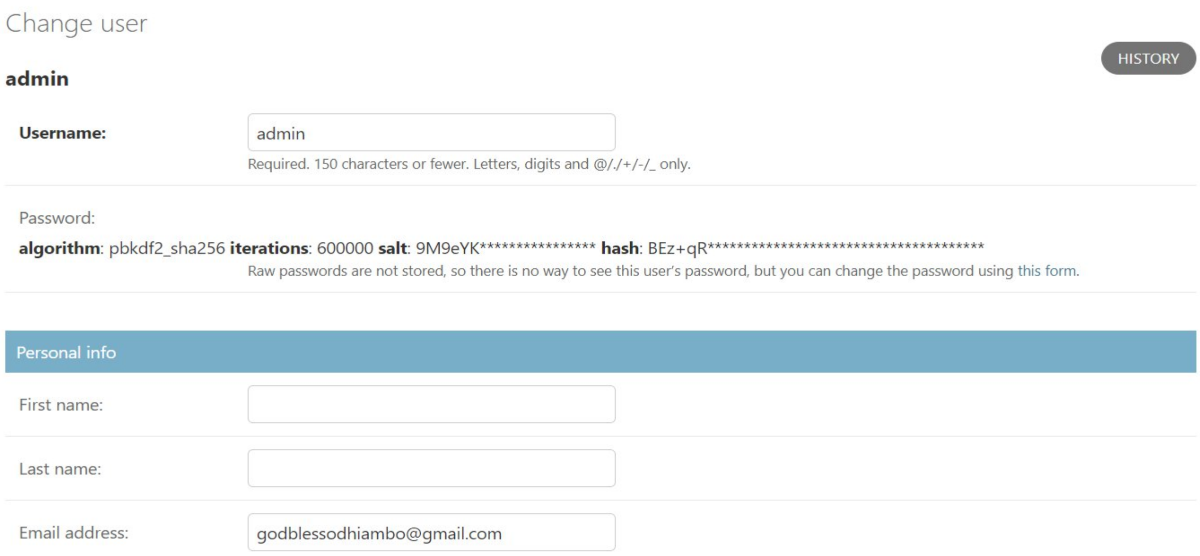

Figure 28:Specific User

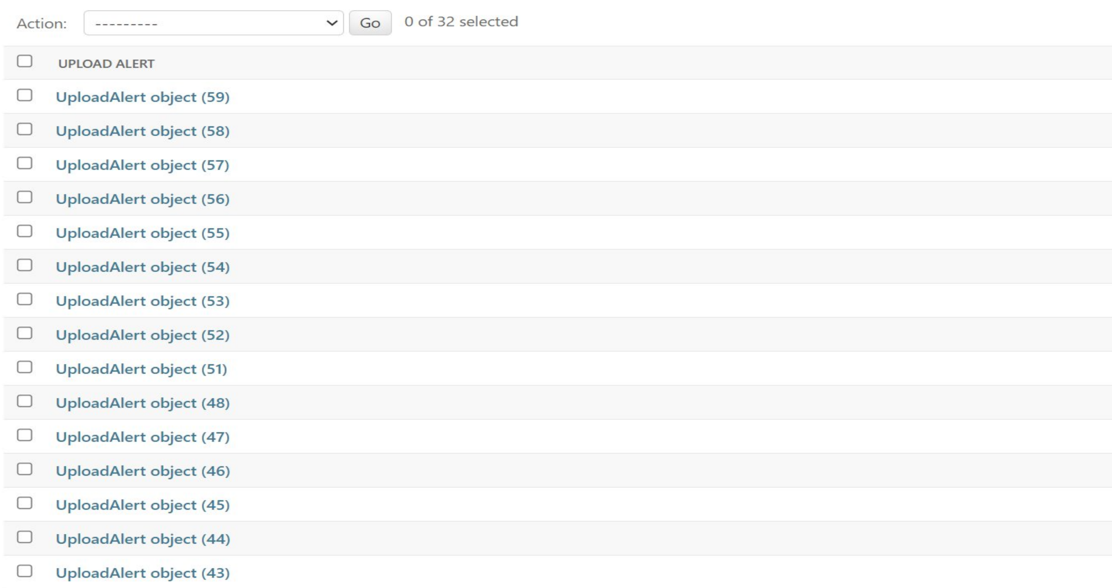

Figure 29:Upload Alerts

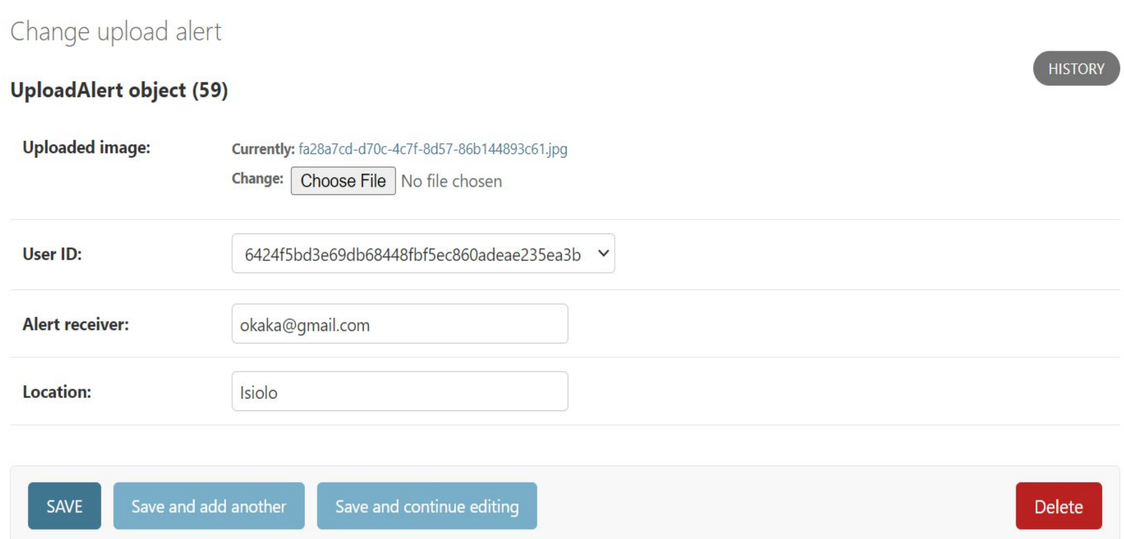

Figure 30:Specific Upload Alert

Select token to change

ADD TOKEN +

KEY	USER	CREATED
b66108f870a2f996b1194b333aa399dc0896d5f8	Collins	Jan. 22, 2024, 11:40 a.m.
afa459f8cf38982b9b80f20fe3e18991fd8eb24d	blessing	Nov. 27, 2023, 7:31 a.m.
52fc2bcd7fdfcb38594e720e75914999c8f2fccb	Okaka	Nov. 22, 2023, 3:29 a.m.
087fab5463aab9e7e116990c2bc217590d90a6e0	Ivy	Nov. 14, 2023, 12:48 a.m.
6424f5bd3e69db68448fbf5ec860adeae235ea3b	admin	Nov. 14, 2023, 12:03 a.m.

5 tokens

Figure 31:User Authentication Token

CHAPTER 8

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

8.1. Introduction

8.1.2. Detecting Handguns Via Web Cameras

Objective one aimed to detect handguns using a computer webcam and draw a bounding box around the weapon detected to point to the detection's success. You Only Look Once version four (YOLOv4) was used successfully in developing the model.

8.1.3. Identifying Performance Factors

Objective two focused on sending an alert to the database containing the time and place in which the detection took place alongside a snapshot of the handgun detection within a bounding box.

8.1.4. Conclusions

My research questions have been effectively addressed through the comprehensive implementation and testing of the Weapon Detection System. I have gained insights into the system's performance, identified influential factors, and verified the system's robustness. These achievements contribute to the successful realization of my research objectives and provide a solid foundation for informed decision making.

This project involved the development of a system that detects weapons, guns, to be specific. Weapon Detection using YOLOv4 for smart surveillance is designed to enhance maximum security. One of the primary goals for this system is to enable authorities detect guns and have data concerning them uploaded to their databases.

8.1.5. Limitations

i. Challenge in testing the system – Real guns were required in order to test whether they were licensed or not alongside their whereabouts. But obtaining real guns required a lot of procedures and it is practically impossible. Fake guns had to be used to test the system.

ii. High computer specification such as memory, and processing speed is required to get the system up and running.

8.1.6. Recommendations

While the current version of the YOLOv4 weapon detection system has already introduced a range of convenient features, with future enhancements the software improves the user experience and expand its capabilities. This application has the potential to play a pivotal role in transforming and enhancing security in the future.

The future enhancements for the system include:

i. Using real guns for testing.

ii. Tracking the weapons and determining whether they are licensed or not.
iii. Incorporate a real time GPS tracking.

iv. Incorporating sensors instead of datasets consisting of photos only.

8.1.7. Chapter Summary

The YOLOv4 weapon detection system represents a significant advancement in the smart surveillance industry, offering a more convenient and efficient way for the authorities to detect weapons hence reducing the rate of crime in the country. It addresses a crucial need for security agencies by simplifying the process of locating and detecting weapons as well as sending relevant information concerning them to the authorities. This system harnesses the power of technology to provide maximum security.

The software's features, including registration, login, detection, and alert upload, offer significant benefits to both civilians and the government.

REFERENCES.

S. A. Velastin, B. A. Boghossian, and M. A. Vicencio-Silva, “A motion-based image processing system for detecting potentially dangerous situations in underground railway stations,” Transportation Research Part C: Emerging Technologies, vol. 14, no. 2, pp. 96–113, 2006.

United Nations, Office on Drugs and Crime, Report on “Global Study of Homicide”, https://www.unodc.org/documents/data-and-analysis/gsh/Booklet1.pdf.

P. M. Kumar, U. Gandhi, R. Varatharajan, G. Manogaran, R. Jidhesh, and T. Vadivel, “Intelligent face recognition and navigation system using neural learning for smart security in internet of things,” Cluster Computing, vol. 22, no. S4, pp. 7733–7744, 2019.

V. Babanne, N. S. Mahajan, R. L. Sharma, and P. P. Gargate, “Machine learning based smart surveillance system,” in Proceedings of the 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 84–86, IEEE, Palladam, India, December 2019.

A. Joshi, N. Jagdale, R. Gandhi, and S. Chaudhari, “Smart surveillance system for detection of suspicious behaviour using machine learning,” in Intelligent Computing, Information and Control Systems. ICICCS 2019. Advances in Intelligent Systems and Computing, A. Pandian, K. Ntalianis, and R. Palanisamy, Eds., vol. 1039, Springer, Cham, Berlin, Germany, 2020.

K.-E. Ko and K.-B. Sim, “Deep convolutional framework for abnormal behavior detection in a smart surveillance system,” Engineering Applications of Artificial Intelligence, vol. 67, pp. 226–234, 2018.

S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. Faughnan, “Smart surveillance as an edge network service: from harr-cascade, SVM to a lightweight CNN,” in Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), pp. 256–265, Philadelphia, PA, USA, April 2018.

R. Xu, S. Y. Nikouei, Y. Chen et al., “Real-time human objects tracking for smart surveillance at the edge,” in Proceedings of the 2018 IEEE International Conference on Communications (ICC), pp. 1–6, Kansas City, MO, USA, May 2018.

S. Ahmed, A. Ahmed, I. Mansoor, F. Junejo, and A. Saeed, “Output feedback adaptive fractional-order super-twisting sliding mode control of robotic manipulator,” Iranian Journal of Science and

Technology, Transactions of Electrical Engineering, vol. 45, no. 1, pp. 335–347, 2021.

S. Ahmed, H. Wang, and Y. Tian, “Adaptive fractional high-order terminal sliding mode control for nonlinear robotic manipulator under alternating loads,” Asian Journal of Control, 2020.

S. Ahmed, H. Wang, and Y. Tian, “Adaptive high-order terminal sliding mode control based on time delay estimation for the robotic manipulators with backlash hysteresis,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 2, pp. 1128–1137, 2021.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788, Las Vegas, NV, USA, June 2016.

A. Farhadi and R. Joseph, “Yolov4: an incremental improvement,” Computer Vision and Pattern Recognition, 2018.

C. He, J. Shao, and J. Sun, “An anomaly-introduced learning method for abnormal event detection,” Multimedia Tools and Applications, vol. 77, no. 22, pp. 29573–29588, 2018.

Q. Hu, S. Paisitkriangkrai, C. Shen, A. van den Hengel, and F. Porikli, “Fast detection of multiple objects in traffic scenes with a common detection framework,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 1002–1014, 2015.

M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, “Automated detection of firearms and knives in a CCTV image,” Sensors, vol. 16, no. 1, p. 47, 2016

H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino, “Analyzing tracklets for the detection of abnormal crowd behavior,” in Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, pp. 148–155, IEEE, Waikoloa, HI, USA, January 2015.

S. Ji, W. Xu, M. Yang, and K. Yu, “3 D convolutional neural networks for human action recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 221–231, 2012.

L. Pang, H. Liu, Y. Chen, and J. Miao, “Real-time concealed object detection from passive millimeter wave images based on the YOLOv4 algorithm,” Sensors, vol. 20, no. 6, p. 1678, 2020.

A. Warsi, M. Abdullah, M. N. Husen, M. Yahya, S. Khan, and N. Jawaid, “Gun detection system using YOLOv4," in Proceedings of the 2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), pp. 1–4, IEEE, Kuala Lumpur, Malaysia, August 2019.

G. K. Verma and A. Dhillon, "A handheld gun detection using faster r-cnn deep learning," in Proceedings of the 7th International Conference on Computer and Communication Technology, pp. 84–88, Kurukshetra, Haryana, November 2017.

S. B. Kibria and M. S. Hasan, "An analysis of feature extraction and classification algorithms for dangerous object detection," in Proceedings of the 2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE), pp. 1–4, IEEE, Rajshahi, Bangladesh, December 2017.

A. Castillo, S. Tabik, F. Pérez, R. Olmos, and F. Herrera, "Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning," Neurocomputing, vol. 330, pp. 151–161, 2019.

V. Gun, "Database," http://kt.agh.edu.pl/grega/guns/.