Image Analysis for Person-Object Distance Measurement using YOLO (You Only Look Once)

Patrick Mark Andrei M. Castro
Department of Computer Science
Adamson University
Manila, Philippines
patrick.mark.andrei.castro@adamson.edu.ph

Angelo D. Erasmo
Department of Computer Science
Adamson University
Manila, Philippines
angelo.erasmo@adamson.edu.ph

Julius Fredrick Nocon
Department of Computer Science
Adamson University
Manila, Philippines
julius.fredrick.nocon@adamson.edu.ph

Jhon Henry S. Bayquen
Department of Computer Science
Adamson University
Manila, Philippines
jhon.henry.bayquen@adamson.edu.ph

***Abstract*—One of the problems that we face today is a problem in security, specifically in our own households. Many homeowners are installing security monitoring to prevent common crimes like larceny, burglary, theft, etc. Usage of CCTV and similar devices can be improved when applied with image analysis. Object detection is a field in computer technology that is used in detecting instances of a certain class within an image; this is used in a wide range of applications such as self-driving cars, robotics, healthcare, advertisement, security and more. This study generally aims to design and develop a home monitoring application using deep learning and to determine the distance measurements between person and object using image analysis. This will lessen the need of manual checking in the feed for proof and violations of a person intruding the place. The research used an object detection model called YOLOv4 (You Only Look Once). This algorithm has a great performance compared to other algorithms especially in detecting real-time. The research also used an algorithm to measure distance between objects and a person. The main steps for this research is gathering datasets, training the object detection model, testing the performance of the object detection model applying the distance measurement on the objects, and developing the home monitoring application which contains various functions and alerts. The output is a home monitoring application incorporating object detection and person-object distance measurement. The results of this study show that applying these algorithms on a home monitoring application lessens the need of manual checking when checking for intruders. It is recommended to train and improve the object detection with more datasets to improve the accuracy**

Keywords— Object detection · Convolutional neural network · Deep learning · Image analysis · Security Monitoring · Desktop application · CCTV · YOLOv4*

I. Introduction

Distance can be measured using different technologies that arose in the world today. These different technologies can serve its purpose to measure distance in different fields in the real-world. By measuring the distance using technology, it can solve specific problems.

One of the problems that we face today is a problem in security, specifically in our own households. The safety of a person and their possessions is a priority. That's why many homeowners are installing security monitoring to prevent common crimes like larceny, burglary, theft, etc. This type of technology can be a big help to this matter especially when it will be used efficiently. Over the past years, we apply capturing devices to our security. One of these is Close Circuit Television or CCTV cameras. Using CCTV cameras is a secure way to monitor or guard a specific establishment, building, street, and especially in a household. Nowadays, people who own properties are more likely to install CCTV cameras to protect them or to prevent crimes. There are numerous benefits that CCTV has: it can be used as evidence for investigation or crime prevention, or for surveillance purposes. In the capturing devices, it can be implemented by using object detection together with a distance measurement. These kinds of cases will briefly be tackled in the research by applying object detection and image analysis to a home monitoring camera system.

This research is based on a deep learning approach on object detection and distance measurement. To this day, the popular and used object detection models and algorithms are based on improved implementations of convolutional neural networks. such as CNN, R-CNN and FasterCNN.

This research will be using a model based on the YOLOv4 [1] (You Only Look Once) model, which is an algorithm based on regression, a model which is also based on the convolutional neural network (CNN) model for doing object detection. The algorithm of YOLO uses only a single neural network to the image and then divides the image into several regions which then simultaneously predicts multiple bounding boxes and probabilities for those regions. The YOLO model trains on full sized images and only requires one forward propagation pass to make its predictions. YOLO is an ideal model for real-time object detection as it has fast performance and high accuracy [2].

A. Background of the Study

Object detection is one of many pieces that classifies computer vision. It is an image processing that deals with detecting instances of objects from a particular class in an image or video. It aids in many ways as it is a general tool for specific uses.

This research will show how useful object detection is in Home Monitoring by applying object detection with the YOLO method. However, one of the problems in object detection is that it only identifies the object but doesn't indicate where the object is specifically located in the image [3]. And by that, this study will implement a distance formula for measuring distance between person to object. It can be useful in a home base 3 security monitoring system to provide us with additional monitoring features. This study can be a big help in detecting each one's home from threats or other unfortunate events. In other parts of the gathered related research, like Detection of Malaysian Traffic Signs via Modified YOLOv3 Algorithm, the said study implemented the YOLOv3 framework for identification of Malaysian traffic signs in the real environment. It enables the identification of their study far and small traffic signs in the real environment [4]. The researchers choose this study because this will be a huge benefit in the fields of computer science specifically in deep learning as well as in home monitoring efficiency.

There are several researches regarding object detection, "Security System and Surveillance using Real Time Object Tracking and Multiple Cameras" is a research that uses real time tracking in security cameras or CCTV systems. The study aims to provide security and detect the moving object in real time video sequences and live video streaming which draws out the research by applying it on multiple stable cameras to track a person in an indoor environment without sacrificing security features of the surveillance cameras [5]. Another is “Overhead View Person Detection Using YOLO" is a research that uses YOLO. The research trained the model using a frontal view data set and tested it on a person data set focused on overhead view. Based on the results of training and testing, it generated a good output model with FPR of around 0.2%, and TPR of 95%[6].

Most existing studies and applications have not been implemented. This research addresses the need for distance measurement from person to objects. The research aims at finding out the feasibility of a Person-Object distance measurement that can be applied to Home Monitoring.

B. *Objectives of the Study*

This study generally aims to design and develop an application using deep learning with the aim to determine the distance measurements between objects using Image Analysis.

Therefore, the project must achieve the following specific goals:

- Train a model which would distinguish people and objects in an image.
- Test and implement the application on an image capturing device.
- Develop the Image Analysis application which will detect and measure the distance from one person to another object in real time.

C. *Significance of the Study*

Below is a list of significant contributions of the study.

- In the global context, this study will be an additional evidence of the optimum algorithm in determining the measurements between objects using image analysis. With the study, Institutions worldwide will be able to identify the significance of distance measurement between objects with the aid of image analysis to create decisions and plan strategically. And the significance of distance measurement and image analysis in the field of security.
- To the Image Analysis Researchers, this study will be significant to the Image Analysis Researchers as this research takes a look at Image Analysis and its challenges in a specific setting. The findings of this research can impact the future of Image Analysis.
- In the field of Security, this study will be significant in the field of Security, since this research will take a look at the implementation of image analysis into home using CCTV cameras that may require object detection and distance measurement between objects in order for home owners to monitor if someone is getting too close to a specific belonging that they want to monitor. The finding of this research can benefit the future of implementing image analysis into Security
- To the Researcher Students, the study will help them find a solution to the problem. This research will also serve as a resource for students who are researching Image Analysis, Object Distance Algorithms.

D. *Scope and Delimitation*

The study covers the development of an application using deep learning to determine the distance measurements between person and objects. The application will be a home security monitoring desktop application. This application will be set up in a home 6 environment. The application is intended for people who would like to monitor or set a restriction on an object in their homes in real-time. The application will have features to alert or alarm if a certain condition set by the user is violated, record or log the timestamps of an event, capture a screenshot, record the video feed, send a message to contact set by the user.

Pedestrian dataset from Kaggle and INRIA Person dataset will be the source of datasets of persons that will be used in the study. Handpicked household objects from MS COCO (Microsoft Common Objects in Context), will be used to train to detect the objects. In determining the person and object, an object detection algorithm will be used namely YOLOv4 (You Only Look Once). The study will implement the algorithm in an image capturing device, and will analyze a real-time video feed.

Due to design constraints, the study is limited to distinguishable objects only and definite materials. Image capturing devices are limited to CCTV cameras and IP cameras only. The recommended and minimum hardware specifications needed will not be discussed.

E. *Conceptual Framework*

Specific procedures, requirements, and ideas were carefully discussed to conceptualize the design and development of the project in order to successfully achieve the desired results of this study. After a lot of deliberation and brainstorming, ideas that led to the concept were agreed.

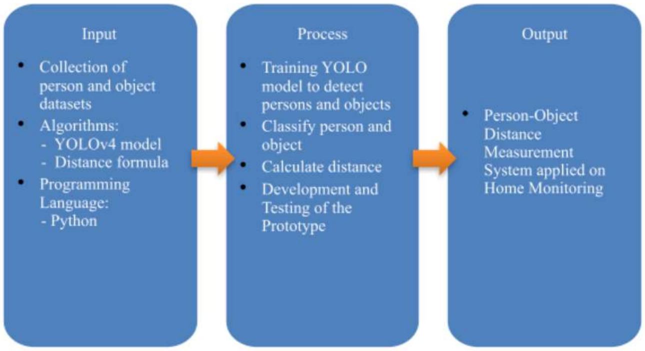

Input
Collection of person and object datasets
Algorithms:
- YOLOv4 model
- Distance formula
Programming Language:
- Python
Process
Training YOLO model to detect persons and objects
Classify person and object
Calculate distance
Development and Testing of the Prototype
Output
Person-Object Distance Measurement System applied on Home Monitoring

Fig. 1 Conceptual framework

Figure 1 presents the conceptual framework of the study. It consists of three major parts; The input phase includes the gathering of datasets of person and object, and the configurations of software and hardware. The algorithms will be used for object detection and the distance measurement, and it will be implemented in Python.

The process phase covers the training and implementation, and it is separated into two parts. First part of the process phase is training the YOLO model to correctly distinguish persons from objects (non person) using the dataset, and implementing the distance algorithm to the detected images and objects. Second part is to develop a prototype that uses the algorithms and testing the prototype using images and to test it on a real time feed using a camera. The main features of the system will be applying the algorithm into the monitoring and reporting system. This includes tracking the distance of objects, alerting when a certain distance is met, and recording the instances when a violation of distance occurs. Recording includes screenshots and videos, which includes timestamps.

The output phase is where the researchers produce a Person-Object distance measurement algorithm applied on Home Security Monitoring Application which monitors the distance from person to objects in real time

F. Operational Definition of Terms

- Algorithm - Algorithm is a sequence of processes that is followed in calculations and other problem-solving operations.
- Application - Application is a computer software that carries a specific task to be executed.
- CCTV - Closed-circuit Television (CCTV) is a video surveillance system that enables monitoring of areas using security cameras.
- CNN - Convolutional Neural Network (CNN) is a deep learning neural network that is used to analyze and process data
- Dataset - Dataset is a collection of organized data that stores specific information about the data.
- Deep learning - Deep learning is a subfield of machine learning that is based on learning on its own by examining algorithms.
- Distance - Distance is a scalar quantity that measures the space between two objects or people.
- Framework - Framework is an abstraction in which software provides generic functionality and can be selectively changed by additional user-written code.
- Image analysis - Image analysis is a subfield in Machine Learning that involves processing an image to extract meaning information.
- Image classification - Image classification is a method in Image Analysis that involves the process of categorizing and labeling images or groups of images.
- Machine learning - Machine Learning is a field in Computer Science that allows computer applications and programs to learn and improve from experience using data.
- Model - Model is a description of a system that uses mathematical concepts and formulations.
- Neural network - Neural Network is a mathematical model that uses learning algorithms to recognize patterns in the data.
- Object - Object is an abstract data that refers to a value or variable.
- Object detection - Object Detection is an image analysis technique that allows computers to identify and classify objects in an image.
- Prototype - Prototype is an experimental version of a product that is created to test or simulate a process.

II. Review of Related Literature and Studies

A. Object and Person Detection

Object detection is a common problem in computer vision since it often entails object recognition and localization. [7]. One-stage object detection algorithms, including YOLO [2] and SSD [8], Because of their high speed, they are often used in real-time detectors and on embedded systems. The study [18] states that one of the most important software components in the next generation of self-driving cars is object detection. The study also aims to address the issue of slow response time in traditional computer vision and machine learning approaches for object detection by implementing the YOLO algorithm which can be processed in real-time to address the problem. The network was trained to detect objects of 5 classes which are cars, trucks, pedestrians, traffic signs, and lastly lights. Despite different weather conditions the approach was successful [18].

In addition to the YOLO model, many object detection algorithms using different approaches and models have also achieved surprising results. Since then, the two concepts of building recognition have merged.

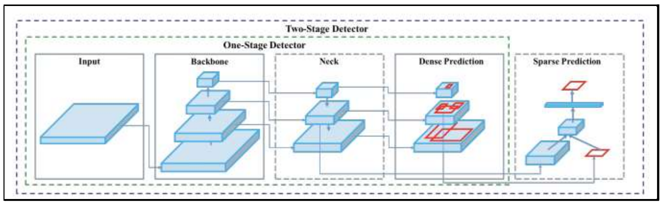


Fig. 2 One-stage detector and Two-stage detector

The primary object detection architecture is that the features of the input image will be reduced through a feature extractor called Backbone. And then go to the object detection (including detection neck and detection head) as shown in figure 2. The detection neck (or neck) acts as a set of features specified for mixing and merging features; are trained in Backbone to prepare the detection step in Detection Head (or Head). The difference emerges here that Head is responsible for detection including localization and classification for each bounding box, while the two-stage detector performs these two tasks separately and combines the results later (Sparse Prediction), while the one-stage detector does it at the same time. (Dense prediction) as shown in Figure 2. YOLO is a one-stage detector, that's why it is called You Only Look Once [9].

Along with classification and localization, Detection is one of the primary works under computer vision algorithms. These three fundamental tasks present both interesting opportunities and challenges to the field of computer vision [10[. It has become a locomotive for researchers and has led to a significant increase in work in these fields. One of the reasons for this tremendous growth is the integration of deeply integrated neural networks (CNNs) into the field of computer vision. Another important role in computer vision is object detection. It detects the presence of the entire object in an image. Examples of this are driverless cars or so-called self-driving cars. They use a combination of software and sensors, not only to detect the presence of other cars, but also the presence of trees, people, animals, other vehicles, and more. on his way.

Deep learning-based object detection has had a great journey over the past few years in developing and using new advanced object detection methods in the field of computer vision. The study [10] provides a concise and excellent idea of the different object detection techniques. The study also focuses on various applications of object detection. In addition, the paper also mentions the promising future of object detection in the field of remote sensing images.

B. Real Time Object Detection Research

Many researches and studies have been conducted when it comes to real time object detection specifically that aims to be more accurate and more robust than the older ones. In the study [11], it presents a multi-resolution framework for generalized object detection that may be utilized in real-time and with a short training time, similar to a single resolution method. The study method is modeled after a human search behavior and biological vision in that humans direct eye movements during object recognition, based on a poor quality description of the object. The results of the study show that as the search dynamics range from coarse to fine, many target candidates can be excluded based solely on their low resolution descriptions, thus creating an efficient search behavior. high efficiency and save time of neural computation. The study applies the Histograms of Oriented Gradient (HOG) framework to detect objects. HOG features are first extracted from smaller spatial regions and then normalized over larger regions known as "blocks". The concatenation of all block features is used as the feature descriptor for the entire detection window. The study is divided as follows: First they will present the general multi-resolution framework, then a crisp and precise explanation of the training and detection algorithms, applying it to HOG features for general object detection. In the said study's experiment there is a comparative performance in both speed and accuracy for pedestrian detection. The researchers performed experiments on the VOC2006 challenge database for motorcycles and cars. In all cases, their method improved both detection accuracy and speed, which has a detection speed of 25 ~ 30 fps for 320 × 240 images.

Another study [12] uses YOLO as a target detection system because it is fast and suitable for real-time detection, and they applied it in face detection. The results showed that the face detection based on YOLOv3 has a shorter detection time as well as being more robust which reduces the miss and error rate compared to the more traditional algorithms. Real-time detection speed requirement was also achieved.

A research regarding real-time object detection that used a different approach [13]. The research paper aimed to detect pedestrians from a camera in the car which may be implemented in smart cars, the research used a multi-stage segmentation technique. Image segmentation is a technique of dividing a digital image into multiple segments in order to make it more meaningful [14]. To visualize image classification techniques, see Figure 3.

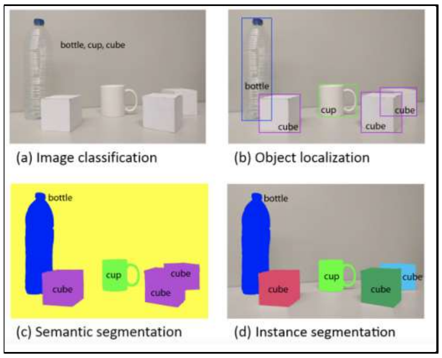


Fig. 3 Difference between (a) image classification, (b) object localization, (c) semantic segmentation (d) instance segmentation [15]

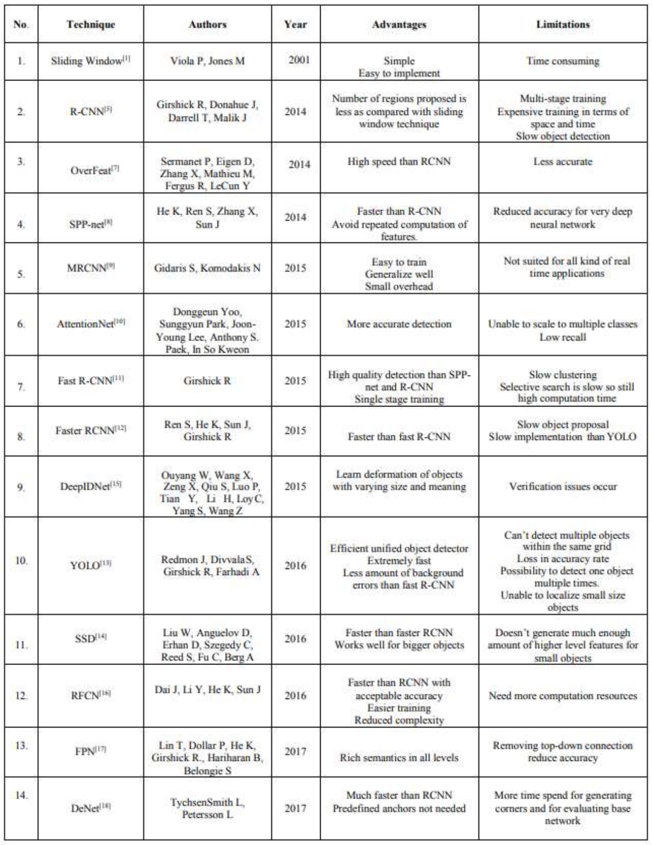
The said research used a method incorporating a multi-stage segmentation with distance transforms. The algorithm is said to apply a "K-means"-like algorithm that uses a bottom-up approach at each hierarchy [13]. The research trial used data of around 700 pedestrian images and was also trained to detect traffic-signs using 1000 pedestrian shapes. The research had a detection rate of 75-85% per image which was fairly accurate. However the research had problems in performing detection when used from a moving vehicle. The authors also mentioned that this approach may not be the best in detecting pedestrians very close to the camera. Because of this approach using image segmentation, the researchers considered taking a look at other implementations of image segmentation.

The research paper [15] also used a segmentation technique creating a modified YOLOv3 model. The study used instance segmentation which is a combination of three image detection techniques, namely classification, localization and segmentation. The goal of the research was to develop a real-time segment that generates masks on people to highlight them. More specifically, the research focuses on the development of a particular segment for a particular sport, golf. They implemented a modified YOLOv3, ROI Align that takes bounding box inputs from the YOLOv3 and a Fully Convolutional Convolutional Network that transforms regions of interest (ROI) inputs to semantic mask outputs. The model was tested on a COCO dataset which measures that performance and the accuracy of the model. The model was tested and compared with the existing instance segmentation Mask R-CNN [16] which generated great results. Even though SEG-YOLO generated less accurate results especially on smaller objects, it has fairly similar in terms of performance compared to the Mask R-CNN however when compared on real-time application SEG-YOLO runs fast 30.8 FPS while Mask R-CNN only runs 5-9 FPS on a single GPU. The drawback in SEG-YOLO is that if the environment has the same color as the outfit the mask prediction sometimes fails like when a hat has the same color as the sky.

C. *YOLO Model*

A study discussed how YOLO started, and its development through the years [17]. Back in 2016, Researcher Joseph Redmon and his other colleagues presented for the first time an object detection system that performs all the essential steps for detecting an object using a single neural network., YOLO algorithm (You Only Look Once) [2]. It reworks the object detection as a single linear regression problem, starting from the image resolution to a called bounding box coordinate and class probability. This unified model predicts multiple bounding boxes and class probabilities simultaneously for the objects covered by the boxes. At the time of release, the YOLO algorithm produced impressive specifications that surpassed the original algorithms in speed and accuracy in detecting and determining the coordinates of objects.They compared it with a popular neural network approach and real time.

The introduction of YOLO in 2016 introduced great promise for the model, outperforming other existing models and with its application to real time detection made it impactful to the field of computer vision. The study concludes that their model is significantly faster in real time and its huge benefit is training directly using a full image. Figure 4 shows a comparative study about various techniques and methods used for object detection together with its authors, year, and advantages and disadvantages.



No.	Technique	Authors	Year	Advantages	Limitations
1.	Sliding Window[1]	Viola P, Jones M	2001	Simple Easy to implement	Time consuming
2.	R-CNN[5]	Girshick R, Donahue J, Darrell T, Malik J	2014	Number of regions proposed is less as compared with sliding window technique	Multi-stage training Expensive training in terms of space and time Slow object detection
3.	OverFeat[7]	Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y	2014	High speed than RCNN	Less accurate
4.	SPP-net[8]	He K, Ren S, Zhang X, Sun J	2014	Faster than R-CNN Avoid repeated computation of features.	Reduced accuracy for very deep neural network
5.	MRCNN[9]	Gidaris S, Komodakis N	2015	Easy to train Generalize well Small overhead	Not suited for all kind of real time applications
6.	AttentionNet[10]	Donggeun Yoo, Sunggyun Park, Joon-Young Lee, Anthony S. Paek, In So Kweon	2015	More accurate detection	Unable to scale to multiple classes Low recall
7.	Fast R-CNN[11]	Girshick R	2015	High quality detection than SPP-net and R-CNN Single stage training	Slow clustering Selective search is slow so still high computation time
8.	Faster RCNN[12]	Ren S, He K, Sun J, Girshick R	2015	Faster than fast R-CNN	Slow object proposal Slow implementation than YOLO
9.	DeepIDNet[15]	Ouyang W, Wang X, Zeng X, Qiu S, Luo P, Tian Y, Li H, Loy C, Yang S, Wang Z	2015	Learn deformation of objects with varying size and meaning	Verification issues occur
10.	YOLO[13]	Redmon J, Divvala S, Girshick R, Farhadi A	2016	Efficient unified object detector Extremely fast Less amount of background errors than fast R-CNN	Can't detect multiple objects within the same grid Loss in accuracy rate Possibility to detect one object multiple times. Unable to localize small size objects
11.	SSD[14]	Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg A	2016	Faster than faster RCNN Works well for bigger objects	Doesn't generate much enough amount of higher level features for small objects
12.	RFCN[16]	Dai J, Li Y, He K, Sun J	2016	Faster than RCNN with acceptable accuracy Easier training Reduced complexity	Need more computation resources
13.	FPN[17]	Lin T, Dollar P, He K, Girshick R., Hariharan B, Belongie S	2017	Rich semantics in all levels	Removing top-down connection reduce accuracy
14.	DeNet[18]	TychsenSmith L, Petersson L	2017	Much faster than RCNN Predefined anchors not needed	More time spend for generating corners and for evaluating base network

Table 1. Brief comparison between object detection techniques [10]

The research [19] proposes the use of an improved YOLOv4 in detecting uneaten pellets despite low-quality underwater images. The real-time monitoring of feed pellet consumption is important in formulating scientific feeding strategies that can reduce the food wasted and also reduce water pollution. In a fish farm the results showed that the average precision increased by 27.21% from 65.40% to 92.61%, the amount of computation was also reduced by almost 30% [19].

D. *YOLO vs other Object Detection algorithms*

Compared with real time detectors YOLO has the greatest mean average precision when it was first implemented in 2016 [2] and its Fast YOLO implementation boasts a very high FPS and mAP compared to the 100HzDPM (deformable parts model) implementation. For the Image Detection using region based proposals (RCNN, fast RCNN, faster RCNN), YOLO has lower mAP, but it greatly outperforms them in real time detection as these region based proposals have a really low FPS. Significant improvements have been made to the model like the YOLOv3 [20] and YOLOv4 [1].

The research paper [1] is an improvement of the model based on the YOLO. The main goal of the researchers of the YOLOv4 is to design an object detector which can really be used in production systems where it can be applied to different technologies when the detection is required real time and can be run on a conventional GPU. YOLOv4 consists of CSPDarknet53 [21] as the backbone, SPP [22], and PAN [23] for the Neck and uses YOLOv3[20] as the head. Also included processing techniques including Mosaic data augmentation, DropBlock regularization, Class label smoothing, Cross-stage partial connections, Self-Adversarial Training [1]. The YOLOv4 model is tested with MS COCO dataset, and is compared with other real time object detectors. YOLOv4, compared to the other models like YOLOv3 [20] , LRF [24], SSD [8] which resulted in YOLOv4 being superior in terms of accuracy and FPS. See Figure 4.

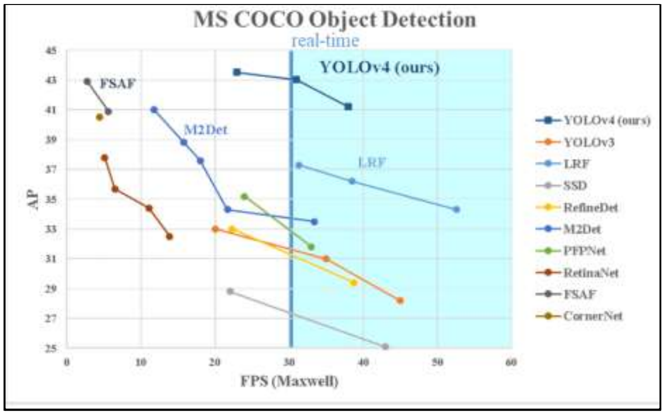


Fig. 4 Performance of YOLOv4 compared to other real-time image detectors [1]

In the study [25], they compared YOLO to other object detection algorithms for real-time vehicle type recognition, Faster-RCNN, SSD which are algorithms that can also be processed in real time with a relatively high accuracy, They found that the YOLOv4 outperformed the other methods with an accuracy of 93% in recognizing the model of the vehicle [25].

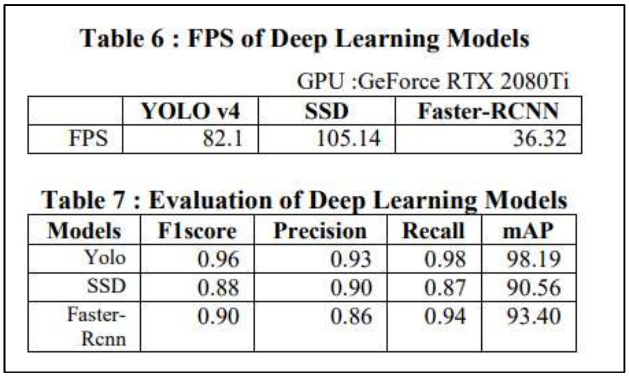


Table 6 : FPS of Deep Learning Models

GPU :GeForce RTX 2080Ti

	YOLO v4	SSD	Faster-RCNN
FPS	82.1	105.14	36.32

Table 7 : Evaluation of Deep Learning Models

Models	F1score	Precision	Recall	mAP
Yolo	0.96	0.93	0.98	98.19
SSD	0.88	0.90	0.87	90.56
Faster-Rcnn	0.90	0.86	0.94	93.40

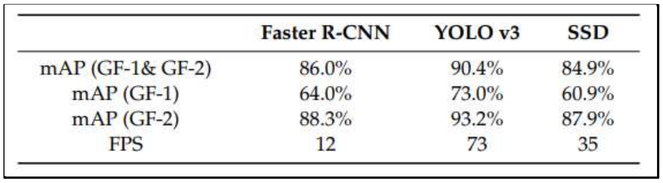
Fig. 5 YOLOv4 compared to other detection algorithms in vehicle type recognition

In Figure 5, YOLOv4 was able to predict the features in each layer using Feature Pyramid Network, they also measured the performance with the use of a single GPU, YOLO produced the most optimal result with the highest accuracy and a relatively high FPS.

In addition to the research above, a research paper [26] aimed to compare the performance of using these algorithms by testing it on tennis videos online. They used action decision networks along with these models to estimate the ball location. They trained the model using images and videos from tennis court datasets. They then measured different parameters to determine which of the algorithms perform well like serve speed, hit speed. The research concluded that SSD is more efficient and more accurate than YOLO and faster RCNN when applied to action decision networks that detects tennis ball tosses.

[27] is a study that compares various object detection algorithms in accurately detecting Agricultural Greenhouses (AGs) which is required in the strategic planning of modern agriculture. Among the other object detection algorithms YOLOv3 was faster and more accurate in detecting AGs, YOLOv3 had an mAP of 90.4% and an FPS of 73. They also tested the latest version of YOLO which was YOLOv4 and

produced a 91.8% mAP as well as 98FPS which was the highest among the compared detectors [27].



	Faster R-CNN	YOLO v3	SSD
mAP (GF-1& GF-2)	86.0%	90.4%	84.9%
mAP (GF-1)	64.0%	73.0%	60.9%
mAP (GF-2)	88.3%	93.2%	87.9%
FPS	12	73	35

Table 2. Metrics comparison of different models.

Based on Table 2, it shows that the three models used had an mAP above 75% with YOLO having the highest mAP of 90.4%. Faster R-CNN and SSD have a similar mAP with only a difference of 1.1%. YOLOv3 also has a much better FPS compared to the other model.

E. Distance Measurement

Several studies estimated and calculated the distance to objects using image analysis. The purpose of the research [28] is to show how to calculate the distance to the detected object. In the research they demonstrate it in an image taken using a camera placed in front of a car. The object is classified using a CNN namely YOLO. One method used to calculate the distance is that the distance is inversely proportional to the number of pixels. They said it is safe to use the reverse rule of 3 to calculate the distance to an object if the number of pixels of the object at a given distance is known. The distance is estimated by counting the number of pixels in a bounding box that matches the object [28].

$$D = \frac{D_{init} P_{init}}{P_{run}}$$

Fig. 6 Distance formula

Figure 6 shows how the distance can be calculated in a more precise way using the formula where D$_{i\ \ i}$ is the distance of the calibration image, P$_{i\ \ i}$ is the width of the object in pixels of the calibration image, and P$_{ru}$ is the width of the object in pixels at runtime.

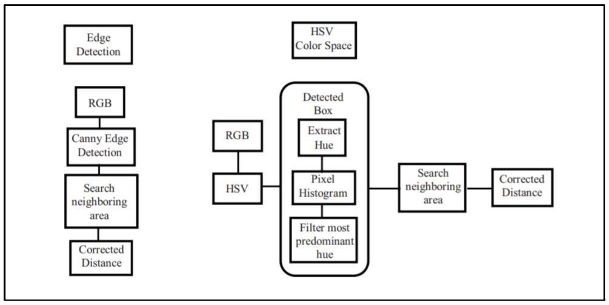


Fig. 7 Edge detection and HSV color space algorithms are used in parallel

Figure 7 shows that canny edge detection and HSV color space are used to further correct the distance. The study proved that the absolute error achieved by YOLO detection is less than 1 meter up to a distance of 8 meters and less than 2 meters up to a distance of 15 meters. After correction, the worst absolute error is reduced from 4 meters to 2 meters. For distances greater than 8 meters, the error is reduced by half. There is no need for correction for distances between 2 and 7 meters since the error is usually less than 0.5 meter [28].

Another research [29] evaluated the ability of image processing in embedded systems based on the LPC1768 microcontroller that works in conjunction with the TCM8230MD CMOS camera and laser set to detect and calculate distance allows this technology to contribute to the implementation of devices such as electrical device floors.

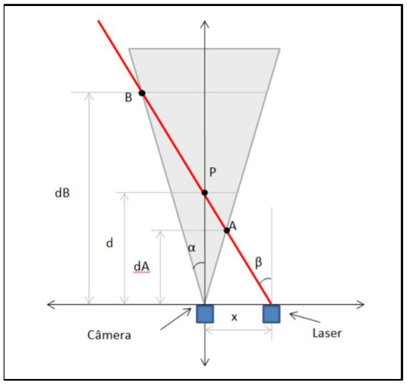


Fig. 8 Mathematical model for calculating distances between obstacle and camera.

Figure 8 shows the method used to measure distances based on a geometric model using a laser and a CMOS camera. The model considers an embedded system with LPC1768 microcontroller to calculate the distance from the obstacle and detect the laser on the camera image. This methodology applies to a low-cost white cane design to assist the visually impaired. Research has shown the effectiveness of measuring distances and using vibration motors as distance feedback to the user. While equipment is performing as expected, new strategies can be implemented to improve power consumption performance. [29].

Another study [30] presented the possibility of using a camera instead of LIDAR for distance estimation, they used two slightly moving cameras to obtain two images, through an algorithm based on the stereoscopy system measurement and calculate the distance to detected objects. The study has demonstrated that the algorithm can be useful for estimating distances within 20 meters. It can be used in many ADAS applications, such as adaptive cruise control, automatic parking, and more. It has been proven that the camera, as the primary sensor, can replace LIDAR in some cases. Inaccurate YOLO bounding boxes cause problems mainly in test situations when the subject is more than 20 meters away from the camera. Regarding measurement accuracy, each pixel error leads to large deviation [30].

A new distance measurement technique was created [31] in which a camera was put in front of a car that captured images of the next vehicles. A system checks the position of the plate of the car, and sends and compares the data in the database to obtain the correct distance. The study showed that it encountered some problems and was solved during the experimentation. Overall, the presentation of the study is just the concept and the presentation of the preliminary stage and it is necessary to do a functional test of the device, some method contents and some steps of machine operation need to be improved. [31].

A research [32] used a novel image-based method that measures various types of distance from a single image captured by a smart mobile device. The embedded accelerometer is used to determine the display orientation of the device, which helps back-projecting the image pixel to the ground. Rear projection was performed using a new camera calibration method that can accurately estimate the focal length using two known distances.

The magnification ratio can be calculated to convert pixel distance to actual distance. Various types of distances, including ground distance, depth and height, can be accurately measured using magnification ratios and back-projection.The study showed the effectiveness of the method only if the distances are known [32].

F. *Application on Security with Object Detection*

Several researches applied object detection in the field of security. The research [33] discussed the implementation of a low-cost, intelligent security system that overcomes the shortcomings of traditional security cameras. This is achieved by using machine learning and the Viola Jones algorithm in the image processing to identify intruders and detect multiple objects in real time. The research presented the design and implementation of an intelligent object detection-based security system using a Raspberry Pi 3B single board computer in two different computing environments, MATLAB and Python, respectively. The model has effectively demonstrated that this detection-based security framework works fine. In the study, machine learning-based facial recognition security systems recognize human faces only after extracting human facial features and warn the owner when a person enters a restricted area.[33].

Another research that used object detection for security is [34]. The paper proposed an algorithm for detecting unmanned and unknown objects using background subtraction and morphological filtering. The purpose of the algorithm is to automatically detect suspicious objects and improve the security of public areas. The system captures frames from the video captured by the static camera while inputting and subtracting the foreground and background using thresholding techniques. Morphological operations are used to enhance the detected region. The main strength of their approach lies in the ability to separate background and foreground in an accurate way from video. The speed of the proposed system is directly proportional to the HD/Full HD/4K surveillance system. They calculated the system's accuracy by comparing the actual and obtained number of frames. They have observed that the accuracy of the system is between 70% to 75% [34].

G. *Synthesis and Antithesis*

These are the related literature and studies that the researchers have gathered based on the study's ideas and concepts.

Estimating the distance to an object based on image processing [28], the study shows how the distance to a detected object is calculated. The distance is estimated by counting the number of pixels in the bounding box that match the object. The distance is further modified using canny edge detection and HSV color space. The similarity of this research is that this research also aims to calculate the distance of objects to a person. However their study has a different approach, which detects the distance from a camera to an object.

The Real-Time Detection of Traffic Participants Using YOLO Algorithm [18], the study aims to address the issue of slow response time in traditional computer vision and machine learning approaches for object detection by implementing the YOLO algorithm. The network was trained to detect objects of 5 classes. Despite different weather conditions the approach was effective. The similarity is that this research would also be using the YOLO model for object detection and aims to detect specific objects only. However the version that will be used is YOLOv4 which has a better performance and higher accuracy compared to YOLOv3, this research would also be applied not only for object detection but also detecting distance between objects.

Object Detection Based Security System Using Machine learning algorithm and Raspberry Pi [33], the paper describes the use of machine learning and the Viola Jones algorithm in the image processing to identify intruders and detect multiple objects in real time for low-cost intelligent security that overcomes the shortcomings of traditional security cameras. Their research is similarly used in security, using image analysis to improve the features of a traditional security application. The difference is the implementation. Their research is focused on checking if a person seen on a camera has entered the premises, while this research will be focusing on checking if a person is near a particular object which will indicate a warning and a notification.

III. Methodology

A. *General Method*

This study will use applied research method. The research paper aims to solve specific problems regarding distance measurement that requires detection of a person to other objects. This research aims to apply the distance measurement and YOLOv4 algorithm to develop an application which will be used to track the distance of person to object and will alert when the distance between them reaches a certain distance. The researchers will use the following diagram to classify the different phases of the research. (See Fig. 9)

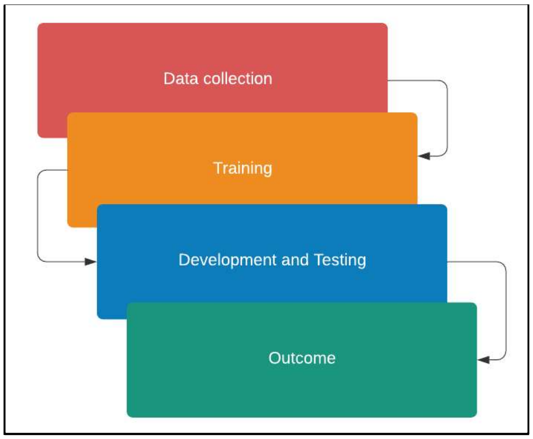


Fig. 9 Main phases of the research

B. *Data Collection*

The purpose of conducting in this phase is to gather the images and convert them into the desired image size, which will then be used to train the object detection model. For data collection, the researchers will gather a person dataset, specifically the dataset that will be used is the INRIA Person Dataset [35] and an open source Pedestrian image dataset from Kaggle [36]. And for object dataset, handpicked household items will be used from MS COCO dataset [37]. These datasets will be used to train the person and object detection model.

C. *Training of Model*

For the training of the person and object detection model, the researchers will use the gathered dataset (Inria Person Dataset [35], Kaggle Dataset [36] and MS COCO dataset [37]) to train the model. A model will be trained to detect persons, common objects and household items will be used to determine the objects in the application.

D. *Development*

The development phase will be separated into two major parts. The development of the algorithm and the development of application.

For the development of the algorithm, the researchers will create the image detection algorithm for the trained objects as well as the distance detection which can be used as a module / library similar to an API, which the GUI application will use to detect through the CCTV.

E. Algorithms

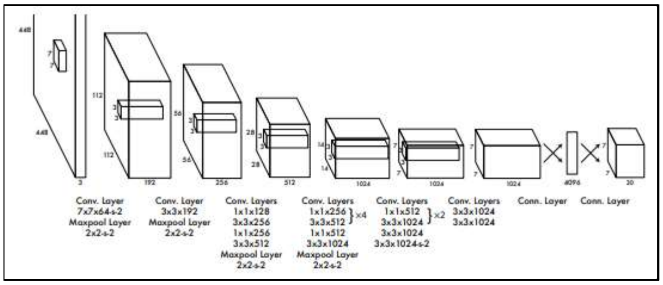


Fig. 10 YOLO architecture [2]

As shown in Figure 10, the YOLO (You Only Look Once) architecture consists of 27 CNN (Convolutional Neural Network) layers, 24 convolutional layers, two fully connected layers, and a final detection layer. It divides the input image into S × S grid cells and predicts the B bounding box and the score of each C class in each grid cell. Each bounding box consists of five predictions: center x, center y, width, height, and bounding box confidence. For each grid cell, there is only one set of class scores C for all bounding boxes in that region. Therefore, the output of the YOLO network is a vector of S × S × (5B + C) numbers for each image. The fully connected layers use the features extracted from the convolution layer and use that information to predict the probability of the object and at the same time build the bounding box. Next, the last layer, the YOLO detection layer, is a regression that maps the output of the last fully connected layer to the final bounding box and class assignments.

1. YOLOv4

The main components of YOLOv4 uses Darknet53 [21], SSP [22], PAN [23] as the neck and the for the head it uses YOLOv3[20]. As well as other BoF (Bag of Freebies) and BoS (Bag of Specials). Bag of freebies are training methods which change the strategy for training and increase training cost. Bag of specials are methods that will increase inference cost of the model but will improve its performance [1]. Darknet P53 is an effective backbone for extracting features. It has a deep backbone with 53 convolutional layers and several advanced structures, such as: (a) The remaining blocks. Add a layer of links to simplify network training. (b) Inception structure containing 3x3, 1x1 convolutional kernels to maintain each field while reducing computational time. The SPP-block, and PAN path-aggregation block, are used as post-processing methods to improve accuracy. The YOLOv3 head uses a function pyramid network (FPN)-like structure. In terms of the various scales, it makes three predictions. The bounding box coordinates, confidence scores for each class, and object confidence (1 for object and 0 for non-object) are all included in the output tensor. The YOLO head's outputs are post-processed with non-maximum suppression to eliminate noise.

2. Pairwise Euclidean distance formula

The researchers will use the pairwise Euclidean distance formula on each bounding box in order to generate a distance between each object and create a line for each object.

$$d(x,\, y) \;=\; \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Fig. 11 Pairwise Euclidean distance formula

3. Object Detection

The image detection API is made with Python and the trained model to produce bounding boxes along with the object name based on the objects detected. The object detection function will return the frame with the bounding boxes along with the data of the detected objects.

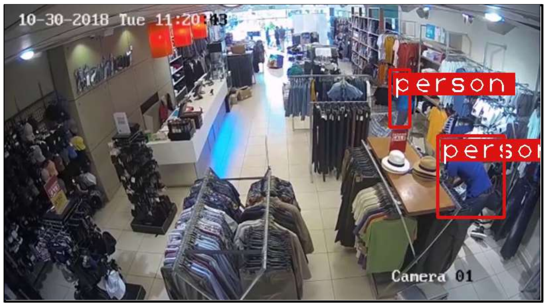
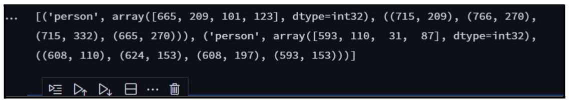


Fig. 12 Object detection

The object data returned is an array of objects (shown below) detected in an image or a frame, in the format (class name, coordinates, bounding boxes). For example, the first object detected is person, the coordinates is [center x=665; center y=209; width=101; height=123;], the bounding boxes coordinates are in a tuple (715,209), (766,270), (715,332), (665,270) in (x, y) format. The object data will be used in the distance algorithm to be explained later. The object detection used for the video feed will apply this same algorithm and will run the detection per frame.



```
...   [('person', array([665, 209, 101, 123], dtype=int32), ((715, 209), (766, 270),
      (715, 332), (665, 270))), ('person', array([593, 110,  31,  87], dtype=int32),
      ((608, 110), (624, 153), (608, 197), (593, 153)))]
```

Fig. 13 Object data returned

4. Distance Algorithm

The distance algorithm API uses the data coordinates obtained from the object detection, in order to compare and check the distances between the objects. The algorithm will be comparing a person to other objects only since it will be used for home security. The Euclidean distance formula (Fig. 11) is used for the detection where the algorithm will check the distance between the coordinates of the detected object and person, the minimum distance computed for each of the points will be used for checking the distance or if it violates a certain distance which is configurable in the function.

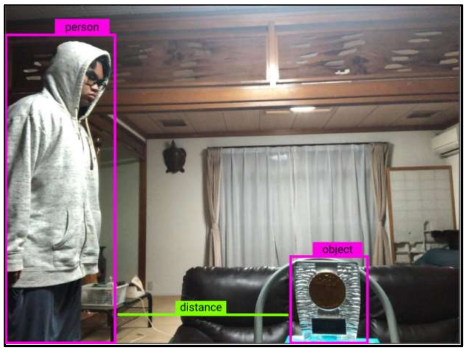


Fig. 14 Person-object distance measurement

The distance checking only checks for pixels as cameras have different depth and distance from the object. The warning distance and critical distance is configurable and can be passed as arguments in the function as well as whether the frame should indicate the distance in between.

The algorithm checks the level of the detection per frame where level 0 = no violation, level 1 = warning, level 2 = critical, the bounding box will also have a different color upon the trigger of the different levels. On level 0 the bounding box color will be green, on level 1 the bounding box will be level orange, and on level 2 the bounding box will be shown in red to really alert the user. The function will return the frame with distance checking and the level of the detection. The level of the violation will be used in the home monitoring application to enable the application to notify, log, and send alerts based on the level of violation detected in the feed.

F. Home Monitoring Application

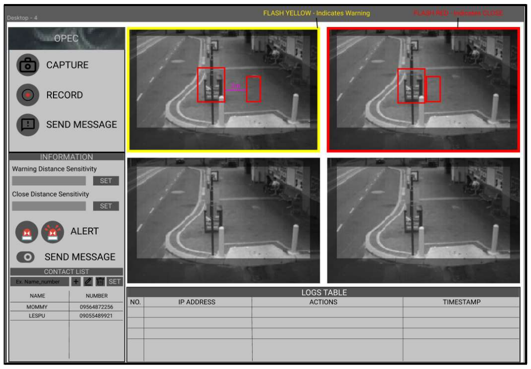
For the development of the application, the programming language used is Python; the researchers will be using Qt Designer and PyQt5 for the GUI that will support the features and the functionalities of the algorithm. The researchers will also be using SQLite3 as the database for the home security monitoring application back-end system. The application can be run on a desktop (Windows) which will be explained in the Outcome.

G. Testing

For the testing phase, the researchers will test the model on the testing set which is 10% from the dataset The results are presented in Chapter 4 of the study.

H. Outcome

The expected outcome will be the Person-Object distance measurement algorithm applied to a Home Security Monitoring Application. Other details will be discussed in Chapter 4 of the paper.



NAME	NUMBER
MOMMY	09564872256
LESPU	09055489921

NO.	IP ADDRESS	ACTIONS	TIMESTAMP

Fig. 15 Home Monitoring Application Prototype

IV. Results and Discussions

A. The Developed System

Hardware

The developed system requires hardware components such as: desktop computer, one (1) or two (2) IP cameras, Wi-Fi router, and an internet connection.

The developed system was run in a desktop computer that consisted of the following such as storage drive, graphics card specifically Nvidia Geforce GTX 1050 Ti, CPU specifically AMD Ryzen 5 3500X, and a total RAM of 16GB DRR4.

The IP cameras used in the developed system were V380 IP Cameras namely Q6 Wifi Smart Net Camera, and Q5 V380s (V-105R). Both were initially set up using the manufacturer's instructions. After that, both of the cameras need to install a file ceshi.ini to enable the RTSP to be able to access the RTSP stream by other devices, in this case, is the developed system. Then the login credentials that were configured in the manufacturer's application were used to connect to the developed system. By joining the camera's credentials and IP address in order to complete the stream URL in such manner rtsp://<username>:<password>@<192.168.xx.xx>:554/live/ch00_1 where 554 was the default port the two cameras, and live/ch00_1 was the path.

Software

The developed software was run in a Windows 10 Professional operating system. The Home Monitoring System is developed mainly on Python, while the GUI of the software was developed using QtDesigner and PyQt5. And SQLite3 for back-end local storage. The software also incorporates other languages such as JSON and HTML to support the requirements of the software.

The application will check the video feed for violations of distances in the feed. The application is similar to those of CCTV application as shown in the program flow (Fig. 16) which allows the user to do basic record functions but with the added features of distance checking, object detection and alerts. Ideally this would lessen the need of a person manually monitoring the feed. The home security monitoring application will use both the object and distance detection algorithm that will track the distance of the person to the object. The person will be automatically detected and the objects can either be detected automatically or let the users set the bounding box around it. This option will help for some objects in the camera that might be unrecognizable to the application because of its shape or form.

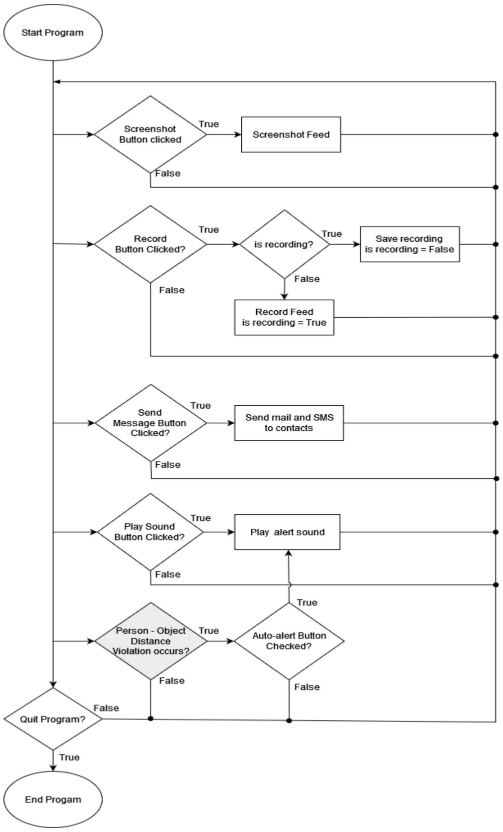
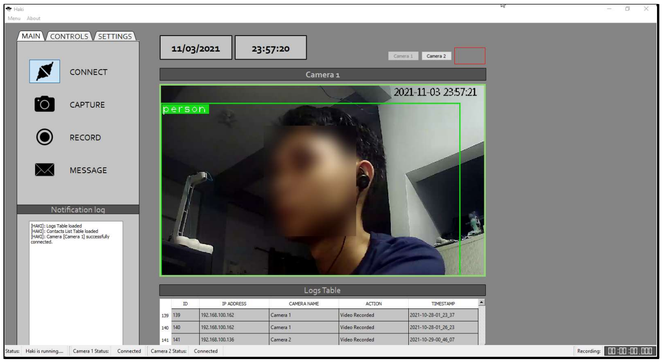

Start Program
Screenshot Button clicked
True
Screenshot Feed
False
Record Button Clicked?
True
is recording?
True
Save recording is recording = False
False
Record Feed is recording = True
False
Send Message Button Clicked?
True
Send mail and SMS to contacts
False
Play Sound Button Clicked?
True
Play alert sound
False
True
Person - Object Distance Violation occurs?
True
Auto-alert Button Checked?
False
False
False
Quit Program?
True
End Progam

Fig. 16 Home Monitoring System Program Flow

The application will track the distance between the person and the object and when a certain distance is met. i.e. when a person is too close to an object, the application will respond with an alert. The user of the application will be prompted with a sound which can be also manually configured and will receive an SMS message as well as an E-mail. The monitoring application will also capture the event with a timestamp in the logs. It will be stored on the local storage of the application.



[HAKI]: Logs Table loaded
[HAKI]: Contacts List Table loaded
[HAKI]: Camera [Camera 1] successfully connected.

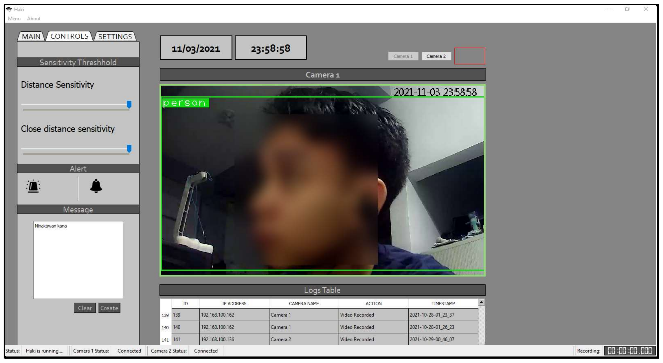
	ID	IP ADDRESS	CAMERA NAME	ACTION	TIMESTAMP
139	139	192.168.100.162	Camera 1	Video Recorded	2021-10-28-01_23_37
140	140	192.168.100.162	Camera 1	Video Recorded	2021-10-28-01_26_23
141	141	192.168.100.136	Camera 2	Video Recorded	2021-10-29-00_46_07

Status: Haki is running.... Camera 1 Status: Connected Camera 2 Status: Connected Recording: 00:00:00.000

Fig. 17 Haki: Home Security Monitoring Application (Main Tab)

For the application's GUI, the Main window (Fig. 17) will include multiple tabs for the categorization of features and functionalities. The Main window will consist of a camera window for camera view, current date and time, camera buttons that can be switched to display camera, and the logs table for the detections in the current video feed.

In the Main tab (Fig. 17), it will present main functions of the home monitoring application like Connect, Capture, Record, and Message buttons. As well as the Notification Log which will show the actions of the program similar to a command prompt.



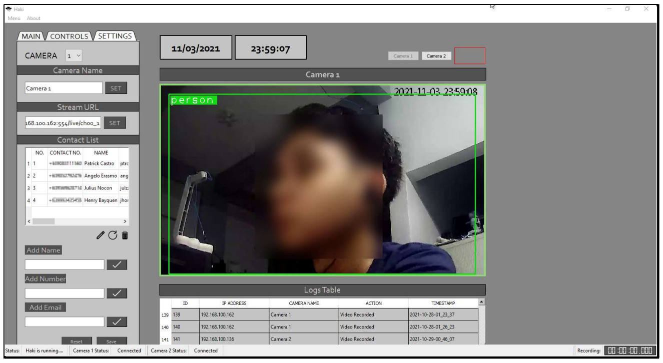
Ninakawan kana

	ID	IP ADDRESS	CAMERA NAME	ACTION	TIMESTAMP
139	139	192.168.100.162	Camera 1	Video Recorded	2021-10-28-01_23_37
140	140	192.168.100.162	Camera 1	Video Recorded	2021-10-28-01_26_23
141	141	192.168.100.136	Camera 2	Video Recorded	2021-10-29-00_46_07

Status: Haki is running.... Camera 1 Status: Connected Camera 2 Status: Connected Recording: 00:00:00.000

Fig. 18 Haki: Home Security Monitoring Application (Controls Tab)

Under the Control tab (Fig. 18), the user can adjust the distance sensitivity for the critical and warning distances from the sliders. These slider objects will connect to the distance detection algorithm for checking the distances and the application will show the appropriate warnings for each violation. Along with the slider, manual and automatic alert buttons are included for optionalities of the user which will play a sound when clicked or when a violation of distance occurs in the feed. And lastly, the message box that can be interacted with by the user to create their specified message that will be sent through email and SMS of the contacts in the Settings tab (Fig. 19), once the algorithm detects a critical violation.



CAMERA 1

Camera 1

168.100.162:554/live/choo_1

	NO.	CONTACT NO.	NAME	
1	1	[redacted]	Patrick Castro	ptrc
2	2	[redacted]	Angelo Erasmo	ang
3	3	[redacted]	Julius Nocon	julz
4	4	[redacted]	Henry Bayquen	jhor

	ID	IP ADDRESS	CAMERA NAME	ACTION	TIMESTAMP
139	139	192.168.100.162	Camera 1	Video Recorded	2021-10-28-01_23_37
140	140	192.168.100.162	Camera 1	Video Recorded	2021-10-28-01_26_23
141	141	192.168.100.136	Camera 2	Video Recorded	2021-10-29-00_46_07

Status: Haki is running.... Camera 1 Status: Connected Camera 2 Status: Connected Recording: 00:00:00.000

Fig. 19 Haki: Home Security Monitoring Application (Settings Tab)

And lastly the Settings tab (Fig. 19), this is where configurations of the cameras' information can be set such as the Stream URL which includes the IP address and device name of the cameras. Up to two (2) cameras can be set and connected to the application. The Contact List is also shown where the user can create, edit, and remove contacts to be alerted.

Integration

The integration of the model to the system was quite a challenge. The integration of the YOLOv4 model to the hardware requires the OpenCV Python library. The captured frames from the hardware were sent per frame which will be processed by the object and distance detection and sent to the system. (See Fig. 20)

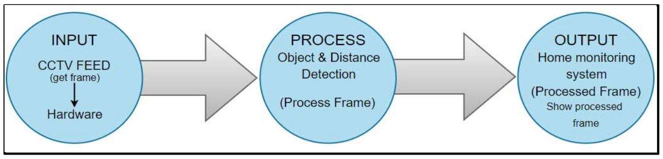
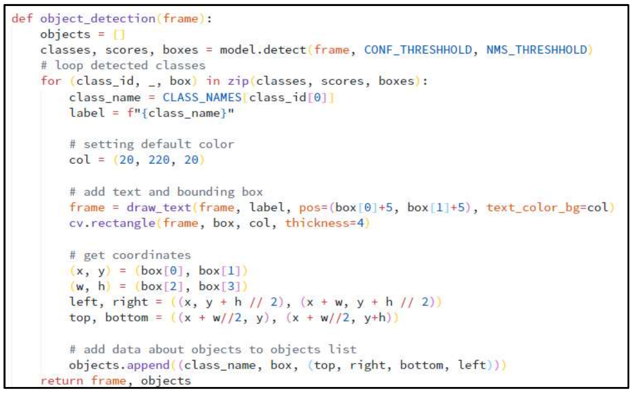


Fig. 20 IPO model for object detection

The object detection (Fig. 21) and the distance detection algorithm (Fig. 22) processes the frame check for object and distance detection and sends the processed frame to be rendered on the system. It also returns the appropriate data which includes the level of violation and the bounding boxes in order that these can be used for alerts and for rendering of the manipulated frame to the home monitoring system.

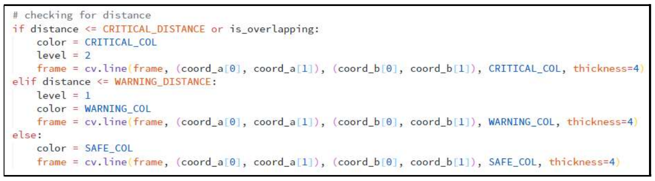
Important Codes



```
def object_detection(frame):
    objects = []
    classes, scores, boxes = model.detect(frame, CONF_THRESHHOLD, NMS_THRESHHOLD)
    # loop detected classes
    for (class_id, _, box) in zip(classes, scores, boxes):
        class_name = CLASS_NAMES[class_id[0]]
        label = f"{class_name}"

        # setting default color
        col = (20, 220, 20)

        # add text and bounding box
        frame = draw_text(frame, label, pos=(box[0]+5, box[1]+5), text_color_bg=col)
        cv.rectangle(frame, box, col, thickness=4)

        # get coordinates
        (x, y) = (box[0], box[1])
        (w, h) = (box[2], box[3])
        left, right = ((x, y + h // 2), (x + w, y + h // 2))
        top, bottom = ((x + w//2, y), (x + w//2, y+h))

        # add data about objects to objects list
        objects.append((class_name, box, (top, right, bottom, left)))
    return frame, objects
```

Fig. 21 Object Detection function

The object detection function will create a label for the object, as well as create a rectangle around it based on the coordinates returned from detection. The default color of the object label is green, as well as the bounding boxes around it. The frame is returned as well as other coordinates for each object which is used in the distance checking to compare the distance from these coordinates as seen in Figure 22.



```
# checking for distance
if distance <= CRITICAL_DISTANCE or is_overlapping:
    color = CRITICAL_COL
    level = 2
    frame = cv.line(frame, (coord_a[0], coord_a[1]), (coord_b[0], coord_b[1]), CRITICAL_COL, thickness=4)
elif distance <= WARNING_DISTANCE:
    level = 1
    color = WARNING_COL
    frame = cv.line(frame, (coord_a[0], coord_a[1]), (coord_b[0], coord_b[1]), WARNING_COL, thickness=4)
else:
    color = SAFE_COL
    frame = cv.line(frame, (coord_a[0], coord_a[1]), (coord_b[0], coord_b[1]), SAFE_COL, thickness=4)
```

Fig. 22 Distance checking

The distance is calculated from the closest two points using the Euclidean distance formula. Depending on the minimum distance between, the function will set a color on to the frame and set the level of the detection as seen in (Fig. 22). Depending on the distance if it is less than or equal to the critical distance defined by the user in the home monitoring system, it will return a level value of 2 and the color will be set as red (critical color), the frame is also updated with the following lines. For showing the warning, the distance is checked less than or equal to the close distance defined in the slider, when the distance is met the level will be 1 and the label color and the bounding box will be colored as the orange (warning color).

The frame is returned with appropriate color and the level of warning which is used by the home monitoring system.

Training

The model is trained with a total of 12,056 images, with 30,011 persons, 9,361 chairs, 5,001 handbags, 5,000 televisions, 4,960 laptops, 2,634 refrigerators from the gathered dataset. Model is trained with darknet YOLOv4 with 14,000 training iterations for the images. The model is trained on a CUDA-enabled GPU hardware Nvidia GTX 1080 along with CPU hardware of Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz. The training took around 120 hours using the GPU. The best weight is generated from the 13894 iteration.

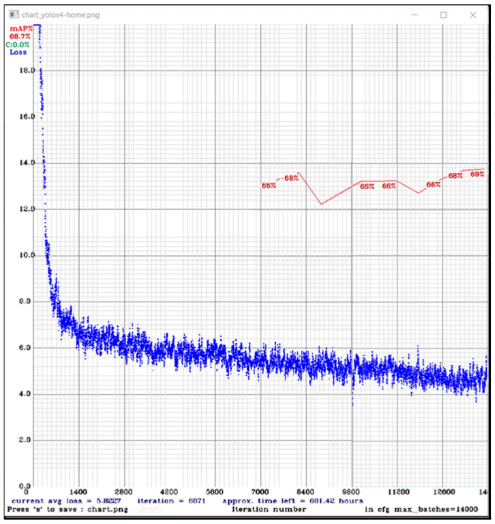


Fig. 23 Model Training

Testing

The trained model is tested on a CUDA-enabled GPU hardware Nvidia GTX 1080 along with CPU hardware of Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz. There are 1016 images used as the validation set to calculate the accuracy of the model. From iteration 7000, the mAP started at 66% then ended with 69%. Table 3 shows the result of the testing of the model. The object detection model is also tested on its integration on the video feed which generated 17-25 FPS (frames per second).

Class	AP (Average Precision)	TP (True Positives)	FP (False Positives)
Person	75.76%	1650	1170
Handbag	36.41%	147	198
Chair	57.66%	307	288
TV	82.83%	301	98
Laptop	80.95%	326	84
Refrigerator	78.72%	161	30

Table 3. Results from testing dataset

From the data above the overall accuracy of the model can be seen with certain formulas. Precision and recall is calculated from getting the true positives, false positives, and false negatives from an image. True positives occur when the model correctly predicts the objects that exist in the model. False positives occur when the model predicts an object but there is no object in the actual image, while false negatives occur when the model fails to predict what it is in the actual image.

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Fig. 24 Precision and Recall Formula

The formula (Fig. 24) for precision is calculated by dividing the number of true positives by the total number of true and false positives. For calculating recall, the true positives are divided by the sum of the true positives and the false negatives. The precision and recall of each class is averaged which resulted in the precision of 0.61 and a recall of 0.59 as seen in Table 4.

For the IoU (Intersection over Union), it is calculated by using the area detected from the model and the actual area of the object in the testing data (Fig. 25).

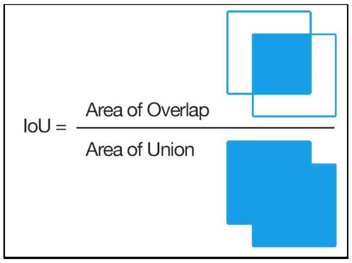


Fig. 25 IoU formula [38]

The Area of Overlap or the overlapping area from the detected bounding box coordinates and the actual bounding box coordinates defined in the testing set, is divided into their Area of Union which results in an IoU per detection. The IoU per detection is averaged to get the overall IoU of the model. For the mAP (mean Average Precision), it is calculated by taking the mean of the AP of each class from Table 3.

Precision	Recall	Average IoU	mAP	Average Detection Speed
61%	59%	49.23%	68.72%	89.57ms

Table 4. Overall results from testing dataset

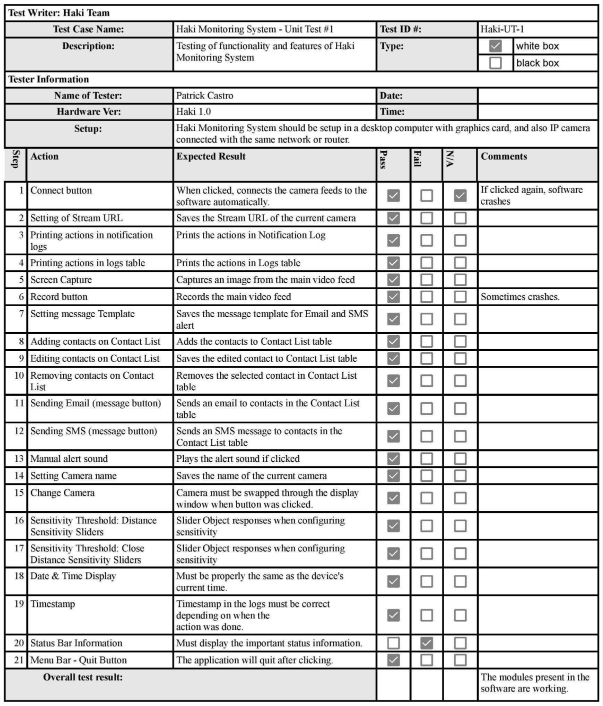
From these results, the Precision is 61%, Recall is 59%, Average IoU is 49.23% and mAP is 68.72%, The testing for 1016 images took 91 seconds, so the average detection speed per image in the testing dataset is 89.57ms. The accuracy of the model is acceptable at 68.72% and can run well in real time on 17-25 FPS which is sufficient for applying on a home security application. The mAP of the model is higher than the YOLOv4 pre-trained model (43.2%) since it focuses on selected objects found in home security.

B. Verification and Testing Result

a. Unit Testing

Unit Testing evaluates the individual modules or features and functionalities of the home monitoring system.

TABLE 5
RESULT OF UNIT TEST OF HAKI MONITORING SYSTEM



Test Writer: Haki Team						
Test Case Name:		Haki Monitoring System - Unit Test #1	**Test ID #:**			Haki-UT-1
Description:		Testing of functionality and features of Haki Monitoring System	**Type:**			☑ white box ☐ black box
Tester Information						
Name of Tester:		Patrick Castro	**Date:**			
Hardware Ver:		Haki 1.0	**Time:**			
Setup:		Haki Monitoring System should be setup in a desktop computer with graphics card, and also IP camera connected with the same network or router.				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect button	When clicked, connects the camera feeds to the software automatically.	☑	☐	☑	If clicked again, software crashes
2	Setting of Stream URL	Saves the Stream URL of the current camera	☑	☐	☐	
3	Printing actions in notification logs	Prints the actions in Notification Log	☑	☐	☐	
4	Printing actions in logs table	Prints the actions in Logs table	☑	☐	☐	
5	Screen Capture	Captures an image from the main video feed	☑	☐	☐	
6	Record button	Records the main video feed	☑	☐	☐	Sometimes crashes.
7	Setting message Template	Saves the message template for Email and SMS alert	☑	☐	☐	
8	Adding contacts on Contact List	Adds the contacts to Contact List table	☑	☐	☐	
9	Editing contacts on Contact List	Saves the edited contact to Contact List table	☑	☐	☐	
10	Removing contacts on Contact List	Removes the selected contact in Contact List table	☑	☐	☐	
11	Sending Email (message button)	Sends an email to contacts in the Contact List table	☑	☐	☐	
12	Sending SMS (message button)	Sends an SMS message to contacts in the Contact List table	☑	☐	☐	
13	Manual alert sound	Plays the alert sound if clicked	☑	☐	☐	
14	Setting Camera name	Saves the name of the current camera	☑	☐	☐	
15	Change Camera	Camera must be swapped through the display window when button was clicked.	☑	☐	☐	
16	Sensitivity Threshold: Distance Sensitivity Sliders	Slider Object responses when configuring sensitivity	☑	☐	☐	
17	Sensitivity Threshold: Close Distance Sensitivity Sliders	Slider Object responses when configuring sensitivity	☑	☐	☐	
18	Date & Time Display	Must be properly the same as the device's current time.	☑	☐	☐	
19	Timestamp	Timestamp in the logs must be correct depending on when the action was done.	☑	☐	☐	
20	Status Bar Information	Must display the important status information.	☐	☑	☐	
21	Menu Bar - Quit Button	The application will quit after clicking.	☑	☐	☐	
Overall test result:						The modules present in the software are working.

Table 5 shows the result of the Unit Testing of functionalities and features of the home monitoring system. For step 1, the connect button connects the camera feeds to the software automatically. However, if clicked again, the software crashes. Step 2 is the setting of the stream URL, it passes the test and saves the stream URL of the current camera when it was connected. In step 3, printing actions in the notification log was a success. Every time an action is done, the notification log automatically displays it. Step 4 is printing actions in the logs table, it also automatically prints the actions in the logs table. Step 5 which is a screen capture button. When clicked, it captures an image form the main video feed and is functional. Step 6 is the record button. The button was functional and was able to record the main video feed. Although sometimes it crashes. In step 7, setting the message template saves the message template for email and SMS. It is working and it has a clear and save button which is convenient. Step 8, step 9 and step 10 are adding, editing, and removing contacts on the contact list. They are operational and can add, edit and remove contacts on the contact list. It also has a contact table so it can view all the information. For step 11, sending an email which is a message button, when tested, it actually sends an email to contacts in the contact list table.

For the testing of email (step 11) and SMS (step 12). The message is sent correctly to a list of contacts through email and SMS sent from the account credentials set, which met the expected result. The alert sound (step 13) plays the alert sound when the button is pressed and stops after. For step 14, the setting of camera name is correctly implemented as it changes the reflected camera name on the top of the main video feed. Change camera (step 15) swaps the main feed and the secondary video feed view, this works properly when switching between video feeds.

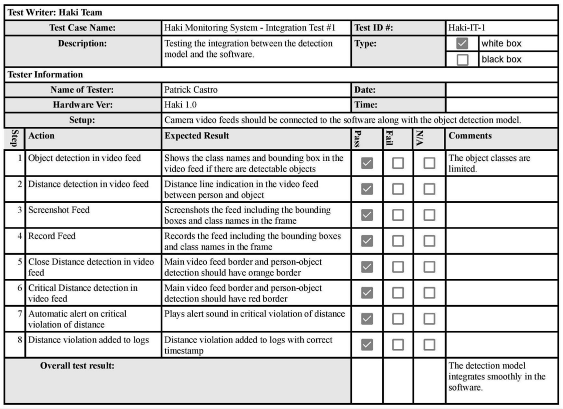
The distance sensitivity sliders (step 16 and step 17) were tested by moving the sliders and if the values from the sliders were correct. For step 18, the displays for the date and time were tested if it reflects the current date and time and if the time updates correctly. In step 19, the timestamp correctly reflects the actual time the action is done. The status bar (step 20) is expected to show the status of the CCTV feed like the connections, is recording, and etc. These are not yet working. For the last step (step 21) Quit button, the application quits when this is clicked.

The overall test result of the features and functionalities of the main modules used in the software are working properly with minor bugs.

b. Integration Test

Integration Test verifies the interaction between the detection model and the home monitoring software and its modules.

TABLE 6
RESULT OF INTEGRATION TEST OF THE DETECTION MODEL AND THE SOFTWARE



Test Writer: Haki Team						
Test Case Name:		Haki Monitoring System - Integration Test #1	**Test ID #:**			Haki-IT-1
Description:		Testing the integration between the detection model and the software.	**Type:**			☑ white box ☐ black box
Tester Information						
Name of Tester:		Patrick Castro	**Date:**			
Hardware Ver:		Haki 1.0	**Time:**			
Setup:		Camera video feeds should be connected to the software along with the object detection model.				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Object detection in video feed	Shows the class names and bounding box in the video feed if there are detectable objects	☑	☐	☐	The object classes are limited.
2	Distance detection in video feed	Distance line indication in the video feed between person and object	☑	☐	☐	
3	Screenshot Feed	Screenshots the feed including the bounding boxes and class names in the frame	☑	☐	☐	
4	Record Feed	Records the feed including the bounding boxes and class names in the frame	☑	☐	☐	
5	Close Distance detection in video feed	Main video feed border and person-object detection should have orange border	☑	☐	☐	
6	Critical Distance detection in video feed	Main video feed border and person-object detection should have red border	☑	☐	☐	
7	Automatic alert on critical violation of distance	Plays alert sound in critical violation of distance	☑	☐	☐	
8	Distance violation added to logs	Distance violation added to logs with correct timestamp	☑	☐	☐	
Overall test result:						The detection model integrates smoothly in the software.

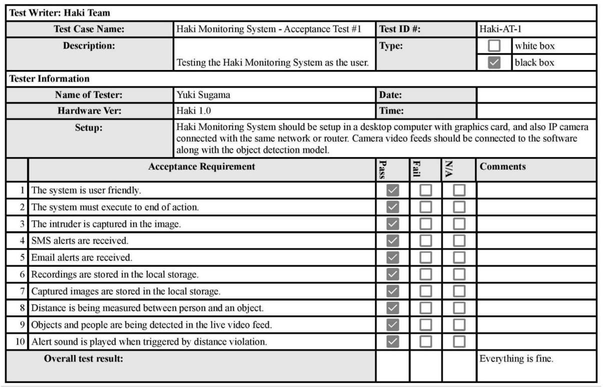
Table 6 shows the result of the integration test of the detection model and the software. In step 1, the object detection in the video feed shows the object names and bounding box in the video feed in real time if the object is detected. While some objects are not detected. In step 2, the distance between person and object are shown in the video feed. In step 3, taking screenshots in the video feed includes the bounding boxes, the object detected class names, and the distance between the person and the object. In step 4, recording the video feed also includes the objects detected, the bounding boxes, and distance between the person and the object in real time. In step 5, the main video feed's border and the person-object detected have an orange border when in close to critical sensitivity. In step 6, the main video feed's border and the person-object detected have a red border when in critical condition. In step 7, an alert plays automatically when the main video feed detects a critical situation. In step 8, when a close or critical incident occurs in the main video feed, the incidents are recorded in the logs table with a timestamp of the incident and the camera details where the feed has an incident.

The overall test result is that the object and distance detection algorithm trained to detect objects and distance in a frame, interacts with the main video feed which is connected to the system. The system shows the information and sends alerts based on the detection in the frame.

c. Acceptance Test

Acceptance test ensures that the home monitoring system meets the requirements specification.

TABLE 7
RESULT OF ACCEPTANCE TEST OF HAKI MONITORING SYSTEM



Test Writer: Haki Team					
Test Case Name:	Haki Monitoring System - Acceptance Test #1	**Test ID #:**			Haki-AT-1
Description:	Testing the Haki Monitoring System as the user.	**Type:**			☐ white box ☑ black box
Tester Information					
Name of Tester:	Yuki Sugama	**Date:**			
Hardware Ver:	Haki 1.0	**Time:**			
Setup:	Haki Monitoring System should be setup in a desktop computer with graphics card, and also IP camera connected with the same network or router. Camera video feeds should be connected to the software along with the object detection model.				
	Acceptance Requirement	**Pass**	**Fail**	**N/A**	**Comments**
1	The system is user friendly.	☑	☐	☐	
2	The system must execute to end of action.	☑	☐	☐	
3	The intruder is captured in the image.	☑	☐	☐	
4	SMS alerts are received.	☑	☐	☐	
5	Email alerts are received.	☑	☐	☐	
6	Recordings are stored in the local storage.	☑	☐	☐	
7	Captured images are stored in the local storage.	☑	☐	☐	
8	Distance is being measured between person and an object.	☑	☐	☐	
9	Objects and people are being detected in the live video feed.	☑	☐	☐	
10	Alert sound is played when triggered by distance violation.	☑	☐	☐	
Overall test result:					Everything is fine.

Table 7 shows the result of the acceptance test of the Haki Monitoring System. The tester accepted every acceptance requirement. Therefore, the system is accepted by users with the acceptance requirements being fulfilled by the system. The overall test result is that the system fulfilled the basic goal of a home monitoring system.

V. SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

A. Summary of Findings

This study summarizes the following findings:

- The image detection model trained for home security has a mean average precision of 68.7% for object detection. The model has a higher mAP compared to the pre-trained YOLOv4 (43.2%).
- The video feed runs around 17-25 FPS with object detection real-time.
- The unit test of the home monitoring system shows that the modules are working properly, with minor bugs.
- The integration test of the detection model and home monitoring system shows that the home monitoring system has integrated the object and distance detection properly and used the data to generate results.
- The acceptance test of the home monitoring system shows that the system fulfilled the basic goal of the monitoring system, and can be operated by non-technical people.

B. Conclusions

This study concludes the following

- The application of the object detection and person-object distance measurement to the video feed in real time is feasible, this is because of the acceptable frame rate (17-25 FPS) in real time.
- The model has an improved precision from the pre-trained YOLOv4 model for home-security objects and persons.
- In order to train a model which would distinguish persons and objects in an image, the researchers

gathered datasets [35][36][37] that contain persons and objects found in home for training the model. We conclude that after training on 12,056 images, the model has an acceptable precision when detecting people and objects.

- In order to test and implement the application on an image capturing device, the researchers require to connect the IP camera on the object and distance measurement algorithm. We conclude after creating the algorithms and integrating it in the IP camera, that it has a good performance of 17-25 FPS and can be used in a home monitoring application.
- In order to develop an advanced home monitoring system, the functions and features of the application should be made for a home based environment, and the application will integrate the model, object and distance detection algorithms. We concluded that after doing the unit test, integration test, and acceptance test, the researchers have achieved the objective and that the application fully served its purpose.
- Overall, the study achieved its objectives on applying the person-object distance measurement on a home security application. And the researchers conclude that this project is feasible and can greatly improve the field of security by incorporating computer vision through a home environment. This will lessen the need for manual checking of the video feed when checking for intruders.

C. *Recommendations*

This study recommends the following:

- To improve the home monitoring system's performance, the hardware requirement may add or upgrade to a more powerful CPU, RAM, and GPU.
- To improve the precision of the model, add more household object images, valuable objects found common in home images, and common objects in a low-light environment to the dataset. And train the model for a longer time.
- For testing multiple feeds, It is recommended to disable the camera feed UI to improve performance of the detection system.

References

[1] Bochkovskiy, A., Wang, C.-Y., and Liao,` H.-Y. M., “YOLOv4: Optimal Speed and Accuracy of Object Detection”,arXiv:2004.10934

[2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[3] O. D. S. C.- O. D. Science, “Overview of the YOLO Object Detection Algorithm,” Medium, 25-Sep-2018. [Online]. Available: https://medium.com/@ODSC/overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0.

[4] W. -N. Mohd-Isa, M. -S. Abdullah, M. Sarzil, J. Abdullah, A. Ali and N. Hashim, "Detection of Malaysian Traffic Signs via Modified YOLOv3 Algorithm," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-5, doi: 10.1109/ICDABI51230.2020.9325690.

[5] R. Kachhava, V. Srivastava, R. Jain, and E. Chaturvedi, “Security System and Surveillance Using Real Time Object Tracking and Multiple Cameras,” AMR, vol. 403–408, pp. 4968–4973, Nov. 2011, doi: 10.4028/www.scientific.net/amr.403-408.4968.

[6] M. Ahmad, I. Ahmed and A. Adnan, "Overhead View Person Detection Using YOLO," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2019, pp. 0627-0633, doi: 10.1109/UEMCON47517.2019.8992980

[7] Z. Wang, “SEG-YOLO: Real-Time Instance Segmentation Using YOLOv3 and Fully Convolutional Network”, Dissertation, 2019.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A.C. Berg. “SSD: Single shot multibox detector”. 2016 European Conference on Computer Vision (ECCV), 2016, pages 21–37, 2016. 2, 11 doi: 10.1007/978-3-319-46448-0_2

[9] J. Solawetz, “Breaking Down YOLOv4,” Roboflow Blog, 04-Mar-2021. [Online]. Available: https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/. [Accessed: 21-May-2021].

[10] C. Bhagya and A. Shyna, "An Overview of Deep Learning Based Object Detection Techniques," 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2019, pp. 1-6, doi: 10.1109/ICIICT1.2019.8741359.

[11] W. Zhang, G. Zelinsky and D. Samaras, "Real-time Accurate Object Detection using Multiple Resolutions," _2007 IEEE 11th International Conference on Computer Vision_, 2007, pp. 1-8, doi: 10.1109/ICCV.2007.4409057.

[12] W. Yang and Z. Jiachun, "Real-time face detection based on YOLO," 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), 2018, pp. 221-224, doi: 10.1109/ICKII.2018.8569109.

[13] D. M. Gavrila and V. Philomin, "Real-time object detection for "smart" vehicles," Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, pp. 87-93 vol.1, doi: 10.1109/ICCV.1999.791202.

[14] Y.-J. Zhang, “A Review of Image Segmentation Evaluation in the 21st Century,” in Encyclopedia of Information Science and Technology, Third Edition, IGI Global, 2015, pp. 5857–5867.

[15] Z. Wang, “SEG-YOLO: Real-Time Instance Segmentation Using YOLOv3 and Fully Convolutional Network”, Dissertation, 2019.

[16] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[17] T. Do, “EVOLUTION OF YOLO ALGORITHM AND YOLOV5: THE STATE-OF-THE-ART OBJECT DETECTION ALGORITHM”, 2021, [Online], Available: http://urn.fi/URN:NBN:fi:amk-202103042892

[18] A. Ćorović, V. Ilić, S. Đurić, M. Marijan and B. Pavković, "The Real-Time Detection of Traffic Participants Using YOLO Algorithm," 2018 26th Telecommunications Forum (TELFOR), 2018, pp. 1-4, doi: 10.1109/TELFOR.2018.8611986.

[19] X. Hu et al., “Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network,” Computers and Electronics in Agriculture, vol. 185, p. 106135, Jun. 2021, doi: 10.1016/j.compag.2021.106135.

[20] J. Redmon and A. Farhadi. “YOLOv3: An incremental improvement”. 2018, arXiv preprint arXiv:1804.02767, 2018. 2, 4, 7, 11

[21] C. Wang, H. Mark Liao, Y. Wu, P. Chen, J. Hsieh and I. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1571-1580, doi: 10.1109/CVPRW50498.2020.00203.

[22] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015, doi: 10.1109/TPAMI.2015.2389824.

[23] S. Liu, D. Huang, and Y. Wang. “Learning spatial fusion for single-shot object detection”. 2019 arXiv preprint arXiv:1911.09516, 2019. 2, 4, 13

[24] T. Wang, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang and L. Shao, "Learning Rich Features at High-Speed for Single-Shot Object Detection," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1971-1980, doi: 10.1109/ICCV.2019.00206.

[25] J. -a. Kim, J. -Y. Sung and S. -h. Park, "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020, pp. 1-4, doi: 10.1109/ICCE-Asia49877.2020.9277040.

[26] R. Deepa, E. Tamilselvan, E. S. Abrar, and S. Sampath, “Comparison of Yolo, SSD, Faster RCNN for Real Time Tennis Ball Tracking for Action Decision Networks,” presented at the 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE), Apr. 2019, doi: 10.1109/icacce46606.2019.9079965.

[27] M. Li, Z. Zhang, L. Lei, X. Wang, and X. Guo, “Agricultural Greenhouses Detection in High-Resolution Satellite Images Based on Convolutional Neural Networks: Comparison of Faster R-CNN, YOLO v3 and SSD,” Sensors, vol. 20, no. 17, p. 4938, Aug. 2020, doi: 10.3390/s20174938.

[28] G. Natanael, C. Zet and C. Foşalău, "Estimating the distance to an object based on image processing," 2018 International Conference and Exposition on Electrical And Power Engineering (EPE), 2018, pp. 0211-0216, doi: 10.1109/ICEPE.2018.8559642.

[29] S. V. F. Barreto, R. E. Sant'Anna and M. A. F. Feitosa, "A method for image processing and distance measuring based on laser distance triangulation," 2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), 2013, pp. 695-698, doi: 10.1109/ICECS.2013.6815509.

[30] B. Strbac, M. Gostovic, Z. Lukac and D. Samardzija, "YOLO Multi-Camera Object Detection and Distance Estimation," 2020 Zooming

Innovation in Consumer Technologies Conference (ZINC), 2020, pp. 26-30, doi: 10.1109/ZINC50678.2020.9161805.

[31] J. Phelawan, P. Kittisut, and N. Pornsuwancharoen, “A new technique for distance measurement of between vehicles to vehicles by plate car using image processing,” Procedia Engineering, vol. 32, pp. 348–353, 2012, doi: 10.1016/j.proeng.2012.01.1278.

[32] Mr. G. A. Ashok, Dr. J. J, and Prof. M. R.M, “Android Application Implemented for Distance Measurement using Image Processing,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 6, no. 6, pp. 406–409, Jun. 2017, doi: 10.17148/ijarcce.2017.6673.

[33] H. Hashib, M. Leon and A. M. Salaque, "Object Detection Based Security System Using Machine learning algorithm and Raspberry Pi," 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), 2019, pp. 1-4, doi: 10.1109/IC4ME247184.2019.9036531.

[34] T. M. Pandit, P. M. Jadhav and A. C. Phadke, "Suspicious object detection in surveillance videos for security applications," 2016 International Conference on Inventive Computation Technologies (ICICT), 2016, pp. 1-5, doi: 10.1109/INVENTIVE.2016.7823224.

[35] INRIA Person Dataset -
N. Dalal, “INRIA Person Dataset,” http://pascal.inrialpes.fr/data/human/

[36] Pedestrian Dataset - N. J. Karthika and S. Chandran, “Addressing the False Positives in Pedestrian Detection,” Lecture Notes in Electrical Engineering. Springer Singapore, pp. 1083–1092, 2020. doi: 10.1007/978-981-15-7031-5_103. originally used on the study [33]

[37] T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” Computer Vision – ECCV 2014. Springer International Publishing, pp. 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48.

[38] A. Rosebrock, "Intersection over Union (IoU) for object detection - PyImageSearch", PyImageSearch, 2021. [Online]. Available: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.