**Kitchen Safety Guide**

ECE FYP Final Report

**Project Name**: Image fusion of thermal and RGB-depth cameras and its applications

**Project ID**: MWH01

**Supervisor**: Professor Mow Wai Ho

**Author**: CHUI, Chi To Anson; CHUNG, Wai Lok; CHOY, Yu Hin

**Student ID**: 20697419; 20703311; 20697964

**Date**: 9/4/ 2022

TABLE OF CONTENTS

[List of illustrations 2](#_Toc1253802461)

[ABSTRACT 3](#_Toc1319401677)

[SECTION 1 – INTRODUCTION 5](#_Toc895031724)

[1.1 Background and Engineering Problem 5](#_Toc1076659174)

[1.2 Project Objectives 5](#_Toc1596474982)

[1.3 Literature Review 5](#_Toc422564700)

[1.3.1 Combination of images and recognition of the blended image: 6](#_Toc1456040832)

[1.3.2 iGuardStove - Automatic Stove Shut Off: 6](#_Toc426849948)

[1.3.3 Human detection in imaging: 6](#_Toc2085998615)

[1.3.4 Motion Detection System Based on Machine Learning: 6](#_Toc1724915283)

[1.3.5 Fall detection by thermal and depth cameras: 7](#_Toc506717545)

[1.3.6 Thermal imaging for improving kitchen safety: 7](#_Toc1775776196)

[SECTION 2 – METHODOLOGY 7](#_Toc1666419732)

[2.1 Overview 8](#_Toc1295386240)

[2.1.1 product/System Description 8](#_Toc126617862)

[2.1.2 System Block Diagram 9](#_Toc1035875693)

[2.1.3 Components List 9](#_Toc1609224072)

[2.1.4 ECE Knowledge 9](#_Toc1368720962)

[2.2. Objective Statement Execution 9](#_Toc1304355083)

[2.2.1 To develop a Machine Learning Module 10](#_Toc1093368804)

[2.2.2 Selection and building of hardware system to support the software 16](#_Toc1514639695)

[2.3 Evaluation and discussion(new add,not complete) 17](#_Toc141650662)

[SECTION 3 – Conclusion(not complete 0.5 page) 18](#_Toc488939439)

[References 19](#_Toc1537599811)

[Appendices 19](#_Toc368022184)

[A-Final project schedule 19](#_Toc1440793826)

[B- Budget 21](#_Toc2009833305)

[C-Meeting Minutes 22](#_Toc2091046567)

[D- Group Members’ Contributions 28](#_Toc2029613283)

[E- Deviation(s) from the proposal and supporting reason(s) 33](#_Toc1845866698)

# List of illustrations

Figure:

1 figure p.9(Block diagram)

1 figure p.

11(test result)

2 figure p.12(test result)

3 figure p.13(test result)

1 figure p.14(test result)

1 figure p.15(logic gate)

1 figure p.17(test camera)

3 figure p.18(test camera with AI model)

1 figure p.29(contribution)

1 figure p.30(contribution)

2 figure p.31(contribution)

2 figure p.32(contribution)

4 figure p.33(contribution)

Table:

1 table p.9(component)

1 table p.21(project schedule)

1 table p.30(contribution)

# ABSTRACT

This project addresses the engineering problem of detecting stove fires in a kitchen efficiently and reliably. Injury and property damage can result from kitchen safety issues.

The motivation for tackling this problem was to develop a system that could provide early warning of stove fires. This could prevent significant damage and injuries.

The methodology for this project involved creating a dataset of simulated stove fires and writing a model that can detect the real situation from the dataset.

Project results showed that the system detected stove fires accurately in real-time, providing early warning of fires and potentially preventing property damage.

To conclude, this project demonstrated the feasibility of using RGB cameras with AI to detect stove fires in a kitchen. The system could provide an effective method for detecting and preventing stove fires, potentially saving lives and preventing significant property damage.

# SECTION 1 – INTRODUCTION

## 1.1 Background and Engineering Problem

Kitchen safety is always a high-profile issue. Many inventions are related to the above issue, such as Personal Emergency Link Services, Intelligent Fire Sprinklers, Motion Security Cameras, etc. However, Artificial Intelligence or AI-enabled camera monitoring has been rarely developed through these years.

Sometimes tiny accidents may bring dire consequences. For example, teenagers love having entertainment such as watching TV while cooking. They may be over-concentrated on other things and not be aware of the potential dangers of unattained fire. The elderly may forget what they are currently doing while cooking. Current products such as fire extinguishers might be applied to put out the fire, but they could not prevent the accidents, and some elderly may even do not know how to use those products.

The aim of this project is to develop an AI-embedded surveillance system specifically for kitchen safety monitoring. It will detect unattended cooking from now and provide immediate responses to users. To achieve this aim, this project is comprised of two parts. For the AI part, we will train a YOLO module to detect fire from the stove and the hand of the cook. Depending on the result of our module, the system will detect potential danger/ fire hazards.

## 1.2 Project Objectives

Our Final Year Project attempt to develop a Safety Guard, which can detect the existence of fire, human and cooking equipment with embedded AI technology implemented. AI technology is added to classify if there is a human taking care of the stove. The image will be first processed, extracting some parameters for training the AI to identify those conditions. If a stove is cooking without humans taking care of it, a particular alarm function for an emergency event will be activated. The objective statement is listed as follows:

Objective 1:To develop and optimize an Object Detection Module which can identify a human, hand, fire, and cooking utensils (e.g. fire, pot).

Objective 2: To develop unattended cooking algorithm which can identify unattended cooking event.

Objective 3: To embed the AI Module mentioned in Objective 1 into Small Single-Board Computers with RGB camera modules and alarm systems.

Compared to current CCTV, we hope that our project can collect vital data and report dangers to guardians immediately. The main difficulties of this project are the use of the RGB camera as the input source and using AI to perform analysis. We hope that our system can not only help guardians respond to emergencies as soon as possible, but also prevent accidents.

## 1.3 Literature Review

1.3.1 Combination of images and recognition of the blended image:

The combination of thermal image and RGB image is proved to be possible. [1] shows that by using ImageMagick, one can resize, enlarge, and combine the thermal image and RGB image. It relies on the manipulation of RGB channels. The processed image could then be passed to any AI (Artificial Intelligence) recognition system. However, the final image formed by the system implemented in [1] is blurry. In the example, it uses a thermal image of 80x60 pixels and an RGB one of 640x480. Thus, the image is enlarged to 8 times what it was. The zoom ratio will increase if the resolution of the RGB image improves. For example, if a regular camera image of 1280x960 is matched with a thermal image of 80x60, the zoom ratio would jump to 16:1 and the result might become unusable.

1.3.2 iGuardStove - Automatic Stove Shut Off:

There has been development in this field. [2] successfully make use of different kinds of sensors to measure the environment of the kitchen including motion sensors, thermometers, timers, etc. This huge system can measure a lot of data and prevent different kinds of danger. However, the size of the system is our concern. As the limited area of Hong Kong housing, a small size of the kitchen safety system is preferable. As such, our system will not be too bulky.

1.3.3 Human detection in imaging:

[3] tried to achieve such a function by using YOLO (You Only Look Once) and a camera. YOLO is an object detector AI. The input media used imaging with infrared radiation (IR). The research team hoped that moving to an IR camera that will not be affected by environment light level, could avoid the problem of “line of sight” that often happens on RGB cameras. Sometimes RGB cameras would be unable to capture what users want, i.e., in the rain, fog, or in darkness. Normally, YOLO would divide the image into grids, then it turns grids into a class probability map. At the same time, YOLO would develop boundary boxes and their confidence levels. As YOLO is pre-trained for ordinal RGB images, the success rate for imaging recognition falls to 7%. According to the evaluation of [3], if YOLO is to be implemented in other projects, additional and specific training is required. Time investment is expected.

1.3.4 Motion Detection System Based on Machine Learning:

An existing system that can detect falling by detecting the human body in an RGB image with machine learning had published by IEEE [4]. This system can classify the action as whether falling or not falling. According to IEEE’s publishment, the system consists of 4 phases, which are data collection, pre-processing, feature extraction, and classification phases. In the data collection phases, training and testing footage is collected with the label “falls” and “without falls”. In the pre-processing phase, the footage will be further processed in 3 steps, which are foreground detection, shadow removal, and object detection. The final output is a bitmap image, which is only human, and a bounding box of it. After detecting a moving human body, the third phase is to extract 2 features in each frame in the video. One is the Aspect Ratio of the width and height of the human’s bounding box. Two is the Fall Angle, by the values of the Aspect Ratio, it will calculate the angle between the ground and the human. In the last phase, the system will classify this action as a “fall” or a “without falls” by the value of the Falling Angle. The Team of IEEE tried 3 classifiers in the classifier stage: polynomial and Radius Basis Function (RBF), Linear Discriminant Analysis (LDA), and K-nearest neighbour (KNN) classifiers, and the LDA classifier have the best performance.

### 1.3.5 Fall detection by thermal and depth cameras:

There are a few existing systems and research based on “thermal + depth camera” and “fall detection”. One of the successful examples is [5], in which the team treated falling as an “abnormality of human action” and used thermal and depth cameras to achieve fall detection. Their research is very similar to this project, and they are very successful, but we are going to do something different: fire danger detection. The AI would also need to handle object identification.

### 1.3.6 Thermal imaging for improving kitchen safety:

There are some kinds of kitchen safety camera systems to help users detect the danger in the kitchen. However, most of the systems use thermal cameras to detect temperature and fire. For example, burner temperature detection, and pot presence detection. One example is [6]. The major difference in our project is to collect the image source with an RGB camera which can lower the price of the product.

# SECTION 2 – METHODOLOGY

## 2.1 Overview

We will accomplish this project by referencing the technology mentioned in 1.3.6 with the change of the input source, which is an RGB image.

To use the RGB image detecting unattended cooking, we had implemented a python program on the Raspberry Pi 4B with RGB MIPI Camera. The RGB image will be analysed by an object detection model and an Unattended Cooking Algorithm to produce a signal output, which is the likelihood of unattended cooking situation. This section will discuss the detail methodology of our system.

We divide this project into 2 main parts, which are

1. A.I. module training and optimization and Unattended Cooking Algorithm development.
2. Implementing this A.I. into an embedded system with an alarm and other needed features.

### 2.1.1 product/System Description

**Input Data**

The raw data is collected by an MIPI RGB camera. To access the raw data by python, we make use of few existed python libraries like Picamera2 for bullseye OS to interface with MIPI camera, Open CV for visualization of the image. A RGB Numpy array will be obtained.

**Pre-Process**

The image will first be pre-processed for the object detection model input. The pre-process included resize the image to 1 x 3 x 640 x 640, changing the brightness and contrast of the image, and normalization.

**Inference**

By detecting the fire from the stove and human hands, we can identify whether the situation is unattended. So, we trained a custom YOLOv5 model for individually detecting fire, human, hand and pot. We selected “YOLOv5s” model not only due to its simplicity for a limited calculation power, but also its popularity and numerous online resources. YOLO is the currently the fastest open sources object detection AI algorithm from Ultralytics. Custom data training can be achieved by the python file train.py in the official repository. Details will be discussed in section 2.2.1

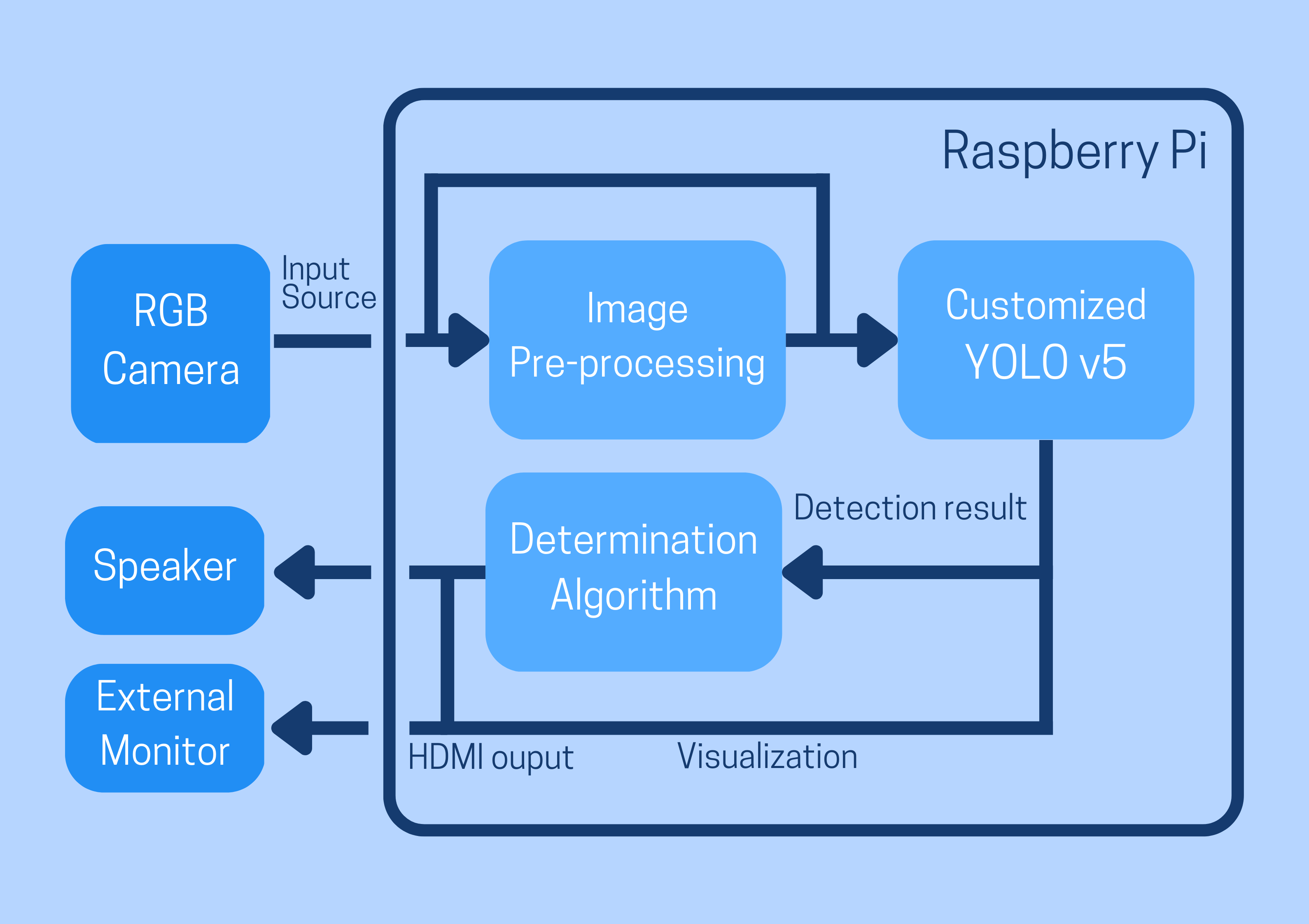
**Unattended cooking Algorithm**

After we obtain the YOLOv5 model detection result. The program takes this result to an algorithm to obtain the final likelihood of unattended cooking event. We had experimented a few algorithms, the details will be discussed in section 2.2.1

**Output**

Based on the final likelihood estimation, the device report to the users by speaker or a buzzer. Hance, a visualization of the detection and algorithm result will be shown in a window as a live video. We currently achieved 1 frame per second (FPS).

### 2.1.2 System Block Diagram



### 2.1.3 Components List

|  |
| --- |
| RGB Camera |
| Raspberry Pi |
| Breadboard |
| Buzzer |
| External Monitor |

### 2.1.4 ECE Knowledge

This project required vastly different kinds of knowledge and skill sets, from hardware to software. The required ECE knowledge is listed as follows:

Before we analyse the raw data of the camera to perform the process mentioned in section 2.1.1, it needed to be extracted from the commercial RGB camera. This requires our knowledge of reading datasheets, communication protocol, and coding skills in C/C++ from ELEC3300 embedded system, ELEC2100 Signal, and system and COMP2011 C++.

Hence, to develop such an AI module, basic knowledge of Machine Learning or Computer Vision is needed. Including knowledge of different classifiers, neural network, and loss functions, which has been covered in COMP2211: python programming with AI and COMP4240 ML & Computer Vision

After we finished the AI module, the next step is to implement it on an embedded AI board with other peripherals, like Buzzer, the Wi-Fi module. Therefore, knowledge of hardware including ELEC1100 Electro-Robot Design, ELEC3300 Embedded System, and ELEC2400 Electronic Circuits is needed.

## 2.2. Objective Statement Execution

### 2.2.1 Object Detection Module Development

**Objective:** To develop a Machine Learning Module (YOLOv5s) which can identify different object (e.g., Human, hand, fire, pot)

To fulfil the objective, we used train a customize YOLOv5s model by Google Colab, Roboflow, OpenCV python library, and ultralytics/yolov5 repository. Detail tasks shown as follows.

**Task 1**

**Aim:** Collect dataset

**Expected Outcome:** 600 images (300 from online,300 from recording by ourselves)

**Member in charge:** CHUNG, Wai Lok; CHUI, Chi To; Choy Yu Hin

**Progress:** We started to record the video in November. Firstly, we try different ratio of recording. Finally, we decide to use 16:9 ratio to record the video. We find out that different angle of recording is also important, so we lastly decide to record the image in front and side view.

**Task 2**

**Aim:** label the object

**Expected Outcome:** label human, hand, fire and pot

**Member in charge:** CHUNG, Wai Lok; CHUI, Chi To; Choy Yu Hin

**Progress:** After collecting the 600 images from both the internet and self-recorded video, we started labelling all the images using Makesense.ai initially. However, we found that Roboflow.com have a more comprehensive functionality compared to Makesense.ai, we switched to using Roboflow.com as our primary tool for labelling. At last, we create a dataset with 600 images with a 70% training set, 20% validation set, and 10% test set.

As we found that the inadequate data, specifically photos of live stove fires, leads us to some flawed results, we increase the number of labelled images to 800. The effect is reflected in the improved results below.

**Task3:**

**Aim:** Visualization with google Colab

**Outcome:** Plot part of the image and label with Google Colab as a demonstration. The Colab Link is shown as follows: <https://colab.research.google.com/drive/1aGRU7MCotNLu76WcDGrYZ17me6RMM1Pd?usp=sharing>

**Member in charge:** CHUI, Chi To Anson

After generating the dataset from Roboflow.com, we wish to visualize the dataset we had made. First, the dataset was downloaded thought Roboflow liberay, with the Jupytor code provided by Roboflow.com. Then, we created two path lists ‘*test\_imgs’* and ‘*test\_labels’*. Since the file in “label” folder and “image” folder dose not synchronize in order. Code had written for synchronize the order in two path list. To draw the image with its own bounding box to, Matplotlib is used for displaying and draw rectangles on the image. Finally, the label and image can be visualized in Google Colab.

**Task 4**

**Aim:** AI training

**Expected Outcome:** Clear bounding boxes are shown on correct items, with at least 70% accuracy and confidence score.

**Progress:** We achieved over 80% accuracy in each class, and only in a few cases would the AI misses the flame.  
<https://colab.research.google.com/drive/1avgUwJ0E-7BhN1OJ4EaQsaaJcENe7qYB?usp=sharing>

**Member in charge:** CHUNG, Wai Lok

**Progress:**

In the first training, as there are too few epochs (rounds of training), there are many false positives and negatives, shown in the graph below. The precision and recall rate are low. The reason is that our dataset has some inconsistencies on labelling. For instance, in the image of pot, some of those bounding box include the handle, and some of them do not. As the result, multiple bounding boxes identified, which are pot with a handle and pot without a handle.

Shown in the confusion matrix, the detection rate of pot and hand are about 80% and 70%. However, it is still difficult for the AI to learn features of fire.

*First Training Results:*

|  |  |
| --- | --- |
| Visualized Results | Validation set |
| Inserting image...  Training Results data | |
| confusion matrix | |

Therefore, we made the following changes modifications:

1. Increasing the Fire class data
2. Re-draw bounding with consistence (i.e. without handle)
3. Increase training epochs

The result is shown as follows:

*Second Training Results:*

|  |  |
| --- | --- |
| Visualized Results | Validation set |
| Training Results data | |
| confusion matrix | |

We can see that the latest model has a remarkable improvement in the accuracy of each class compares to the first training model. And the overall average accuracy is 83% which satisfies our expected outcome.

### 2.2.2 Unattended Cooking Algorithm Development

**Aim:** Build the AI module for classifying “attending” or “non-attending”

**Expected Outcome:** Create an algorithm for classifying “non-attending” or “attending” by using the detection result by YOLOv5.

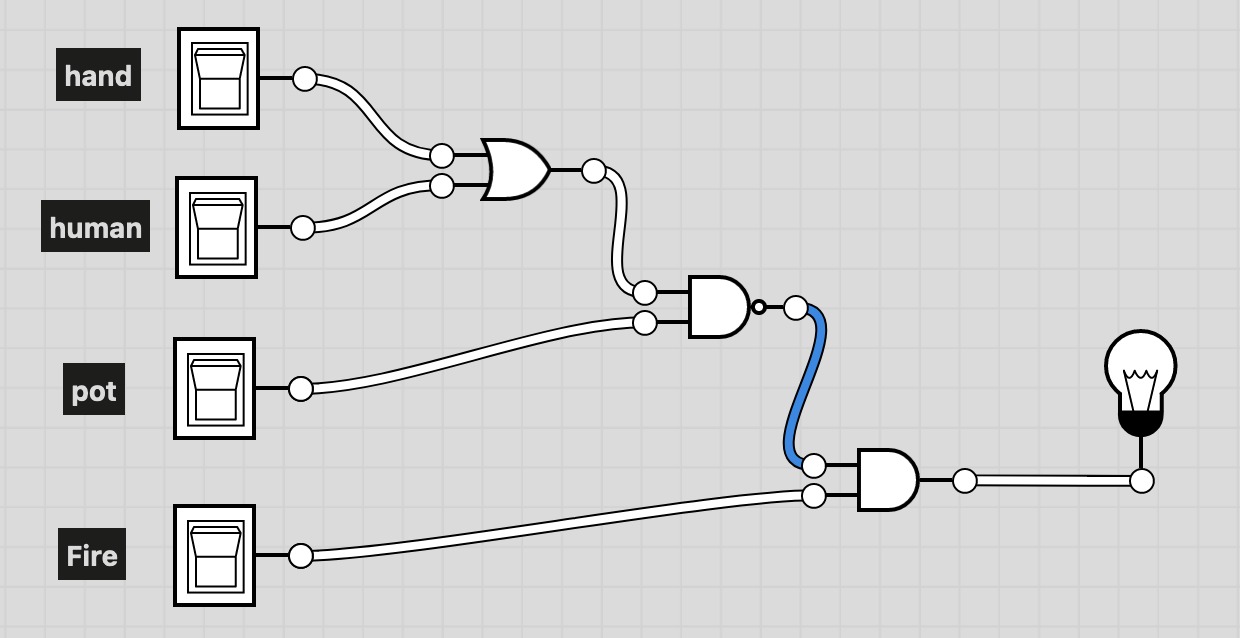
**Member in charge:** CHUNG, Wai Lok; CHUI, Chi To

We had experimented a few algorithms to estimate the unattended cooking event. Pros and cons will list as follows.

**Progress:** evaluated 2 algorithms.

*Algorithm 1: Logic representation of unattended cooking event.*

This algorithm simulates the unattended cooking event of the logic combination of the detection output. Assume the logic input “hand”, ”human”, ”pot ”, ”fire” equal to “True” when that element is present. We can estimate the unattended cooking event occurs by the following logic flow.



The textual algorithm is . When the output equals “True” Implies unattended cooking detected.

Python code:

|  |
| --- |
| flag = (not((log\_dict["human"] or log\_dict["hand"]) and log\_dict["pot"])) and log\_dict["fire"] |

This algorithm only requires a few computational powers due to its simplicity. However, this algorithm oversimplified the real-world situations. For instance, if pot and fire both are present, but the fire is not heating the pot. This algorithm still output False which means safe. So, we did not choose this algorithm as it some other possibility of unattended cooking.

*Algorithm 2: real-time weighted score*

This algorithm gives out weighted score the presence of elements to make decision.

Python code:

|  |
| --- |
| currentScore = (fire \* 2 + hand \* 0.75 + human \* 1 + pot \* 1.2) |

We want to focus on the presence of fire; “fire” element’s weight is increased. Other factors are also of high importance to determine the presence of human; none of factor would receive weight less that 0.5.

However, this algorithm is not perfect. There may be occasions that the cook’s body blocks camera from observing the stove flame. Once it is blocked, the weighted score will sharply decrease.

*Algorithm 3: “forgetful” weighted score*

Python code:

|  |
| --- |
| while true:  pastScore = currentScore > 0 ? currentScore \* 0.5 : 0  currentScore = pastScore + (fire \* 4 + hand \* 0.75 + human \* 1 + pot \* 1.2) |

This algorithm makes use of previous result to introduce function of prediction. As we want to be “forgetful” and not let past results dominate, we scale down its weight (in this case, a factor of 0.5). The idea is to achieve the diminishing effect of past scores.

Suppose we store the currentScore record into an array scoreRecord[], scoreRecord[0] will be the “currentScore” of 1st iteration. Later, the effect of scoreRecord[0] diminishs to one-half (1/2) in 2nd iteration, one-fourth (1/4) in 2nd iteration, that is, i-th element (scoreRecord[i]) will have the effect in the n-th iteration.

“currentScore” is initiated with 0, then it increases due to presence of elements.

We can then react based on the level of score. In this way, we can assess the danger level even if the fire cannot be seen due to light intensity or blockage. We can also divide the danger level further into layers, that is, “High”, “Medium” and “Low”. In this case, we set the high threshold as (fire \* 4 + hand \* 0.75 + human \* 1 + pot \* 1.2)/2, which is around 4 (only flame observed). The system will turn off the alarm if the flame is not “seen” for more than 5 frames (i.e., after around 10 seconds).

*Algorithm 4: “forgetful” weighted score (with negatives)*

Python code:

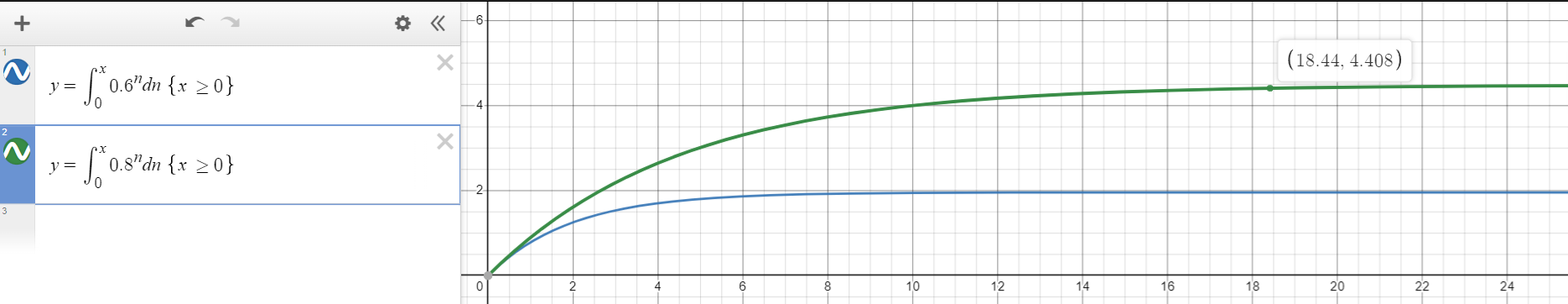
|  |
| --- |
| Threshold = 1 while true:  pastScore = currentScore > 0 ? currentScore \* 0.5 : 0  currentScore = pastScore + (fire \* 1 + hand \* -0.75 + human \* 1 - pot \* 1.2)  currentScore = currentScore if currentScore > 0 else 0 |

One of the concerns of algorithm 3 is that it requires a “warmup” time to aware of fire after long inactivity. Thus, we introduced another mutation of Algorithm 3, which allows a negative score of danger, indicating as safe. To avoid the score from diving too deep, we add stopper functions on pastScore and currentScore (currentScore = currentScore ? currentScore : 0; pastScore = currentScore > 0 ? currentScore \* 0.5 : 0). If there is unattended fire, the system can react quickly, with the lingering effect of past presence of fire.

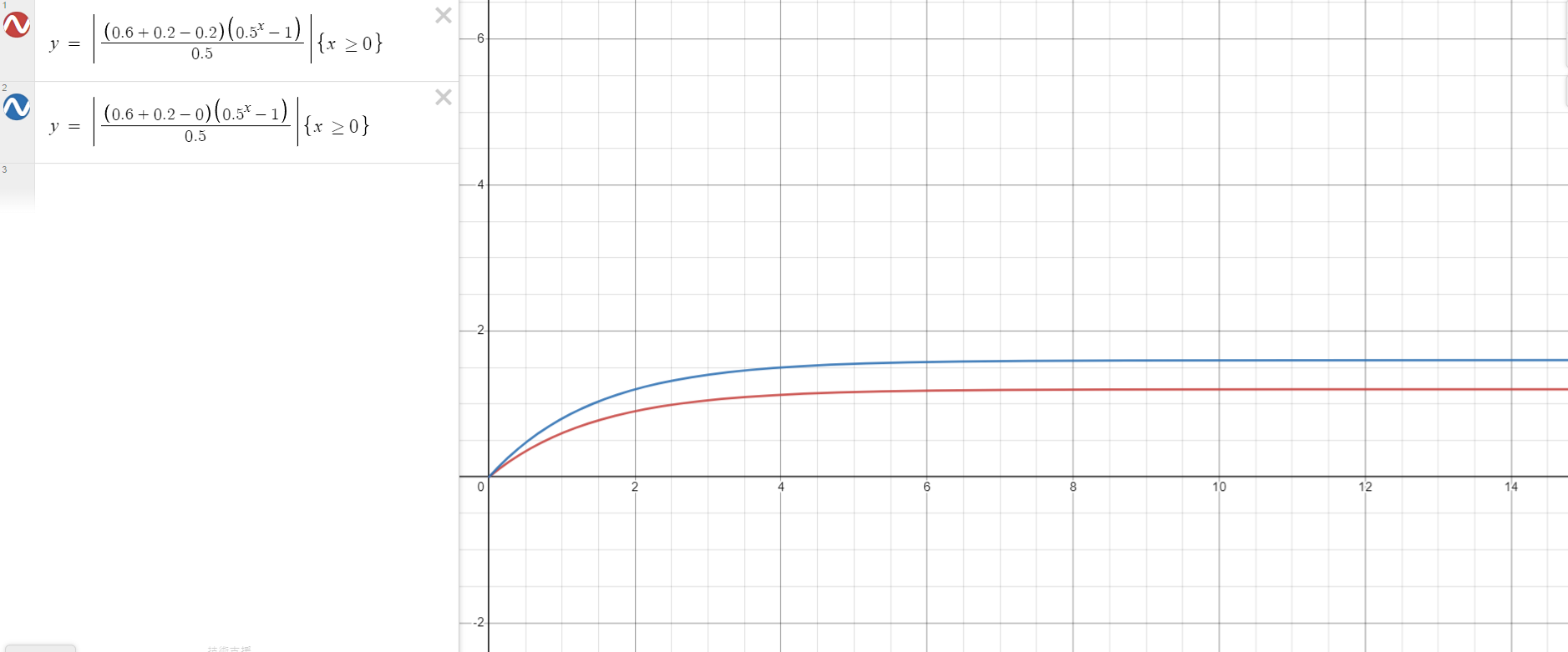
*Algorithm 5*: One/Zero-based Algorithm

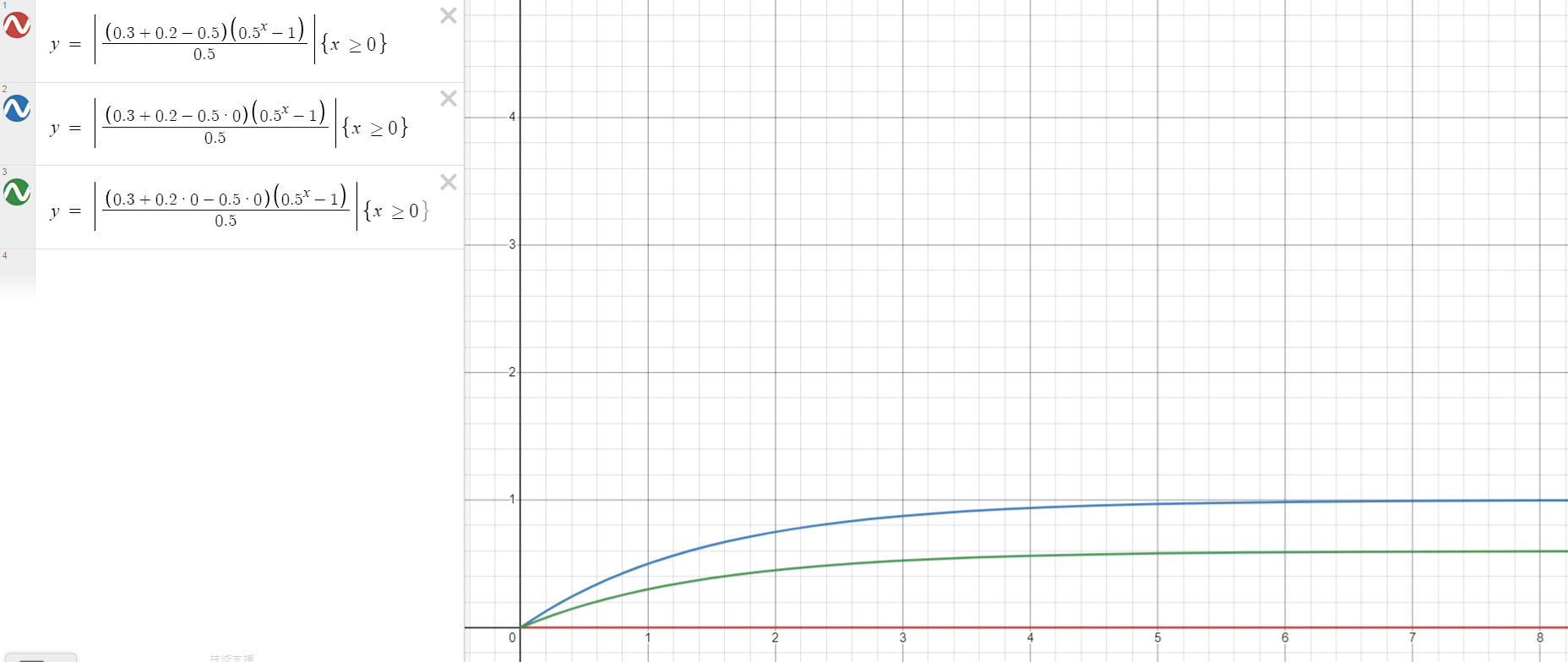
The above algorithms (namely algorithm 2 to 4) may face a similar problem of enlarging absolute value of currentScore. We removed many complicated calcuation in this algorithm.

|  |
| --- |
| Threshold = \_  while true:  pastScore = currentScore > 0 ? currentScore \* 0.5 : 0  currentScore = pastScore + (fire \* 0.3 - (hand or human) \* 0.5 + pot \* 0.2)  currentScore = currentScore > 0 ? currentScore : 0 |

Here, we deliberately make the sum of absolute value of weights to 1 and sum as 0. This can avoid inflation of hazard level (false positive) and sharp add or drop of score (inconsistency). We keep the bottom line of positive scores so as to avoid false negatives due to past safe environment. 

Referring to the mathematic models and the plotted graphs, the existence of all items for a long time will inflate the score to around 2 (i.e., the blue line), while the score will rise till 4.4 if camera only capture fire and pot (i.e., the green line).





We can use the other way (sum of geometric sequence) to show the similar pattern. Variable x represents n-th frame, while variable y represents the accumulated score. The instance of “camera only capture fire and pot” will have higher score. The countereffect of “presence of human” could balance the weight of potential danger (i.e., “fire” and “pot”).

Therefore, we set the threshold to a level that it is just slightly above the normal line and lower than danger zone, and it reserves some buffer.

Task 6

Aim: Connect to a Wi-Fi network if possible; send message via internet

Expected Outcome: The system can send alert messages to guardians if the internet is available.

Member in charge: All

Progress: The module can be connected to Wi-Fi, but it is limited to those with WPA-2 Personal security protocol due to hardware limitations. Raspberry Pi 4 board could only use WPA-2 network. Based on the limitations of unsafe network, we discontinued the development of E-mail functionality. We will focus on other possible outputs instead.

### 2.2.3 Selection and building of hardware system to support the software

Objective: To embed the AI Module mentioned in Objective 1 into Small Single-Board Computers with the camera and alarm system.

**Task 1**

We would use a smaller, portable computer to connect with the camera, to increase the ease of installation in household usage.

**Aim:** Setting up the minicomputer as the hardware part of the system

**Expected Outcome:** Properly install Raspberry Pi OS and necessary such libraries such as Pytorch, Open CV, etc., and the RGB camera functions properly.

**Member in charge:** CHUI, Chi To; CHOY, Yu Hin

**Progress:** Although we reinstall the raspberry OS a few times due to Pytorch compatibility (Pytorch only works in the Bullseye64x version), we successfully install all the required libraries to run our custom YOLOv5 model. On the other hand, we initially interface MIPI camera with Picamera and it failed. The reason is because the Bullseye OS no longer support Picamera. We use Picamera2 and Libcamera as a replacement of Picamera. Finally, MIPI camera are functions properly. We tested it with taking a group photo. AI can also work on the photo and print its prediction.

|  |
| --- |
| Group Photo by PiCamera |

Task 2

We would like to combine our AI model and the camera model. So that the result can show in the photo taken by the camera.

**Aim:** running custom YOLOv5 model inference with MIPI camera as input source.

**Expected Outcome:** the object label can be shown in the photo

**Member in charge:** CHUI, Chi To; CHOY, Yu Hin; CHUNG, Wai Lok

**Progress:** For the YOLO model part, we use the Pytorch and ultralytics/YOLOv5 repository to run inference. For the camera part, we use the Picamera2 and OpenCV to show image and visualize and detection result. In each cycle of the main loop, we first obtain the RGB image as a Numpy array. Then, feed it in our custom model to obtain the detection result. Finally, show the result in a window with bounding box, object name, confidence, and unattended cooking indicator. Until we press the “q” button the program will run the loop and terminate the program.

|  |  |  |
| --- | --- | --- |
| Fig. () Some of demonstrated images with different items | | |
|  |  |  |

### 2.3 Evaluation and discussion

We aim to develop a Safety Guard, which can detect the existence of fire, human and cooking equipment with embedded AI technology implemented, which can provide early warning of stove fires and prevent severe damage and injuries.

The overall result of the project until now is that it can identify the Humans and hands by our system in real-time mode. For the extension of our product, we are trying to increase the FPS by using .onnx yolov5 model format, which is specialize for CPU computation. Our objective of the project is to safeguard. Faster signals sent out can let the user act immediately.

Comparing our results with those of benchmark systems from our literature review, we can meet their result to identify if there is unattended fire with our RGB camera. We have solved the problem with a low-price and basic RGB camera. We proved that we need not have an expensive, high-resolution camera to detect more subtle things (in this case, fire).

Although our model reached 80% accuracy in testing, the accuracy dropped when we tried to move to Raspberry Pi (Linux/Unix-based) platform. This may happen due to dirty input data. We mainly retrieve sample images from real-life and day-to-day situations, in which images contain many unrelated details. The inclusion in images could lead to interpretations different from what we want. We solved the issues by introducing other images having only one to two items and removing duplicate images of original samples. Another reason for low accuracy could be incompatibility of different resolutions. We initially trained the model with 640\*640 images, but we fed it with 2K (2560×1440) images in testing. The issue is solved as PiCamera library supports resolution change.

Give a final assessment of the outcomes of your project and justify your results. Highlight the importance and/or relevance of your work. Discuss what was successful in your project and what wasn’t. Explain why you think you succeeded/failed. Consider what you could have done differently to improve your results or make your work more relevant. (Avoid writing this in a reflective or personal style. Keep your discussion technical.)

We have achieved what we expected, one of which is to detect four elements: fire, human body, hands (palms and fingers), pot. Another achievement is developing an algorithm of hazard assessment other than just using basic logic statements. There is unexpected incompatibility between high-quality RGBA image and camera-generated images. There is also a natural flaw of not having front-facing portraits of cooks. Most images found are of the sides of people cooking.

However, we did not achieve long-distance communication (i.e., the e-mail function). Although we would like to avoid wired connection to existing computer (as that will massively increases the power cost), we could have investigated Wi-Fi extension such as Wi-Fi USB devices and external Network Interface Card (NIC) that can support WPA3 to replace the pre-installed or integrated hardware that only supports less secure WPA2 networks.

In conclusion, we partially succeeded. We completed the fundamental functions, and we investigated diverse ways to optimize our solution, and we tried different solutions. However, there is still room for improvement.

# SECTION 3 – Conclusion(not complete 0.5 page)

Objective:

Our Final Year Project attempt to develop a Safety Guard, which can detect the existence of fire, human and cooking equipment with embedded AI technology implemented. AI technology is added to classify if there is a human taking care of the stove. The image will be first processed, extracting some parameters for training the AI to identify those conditions. If a stove is cooking without humans taking care of it, a particular alarm function for an emergency event will be activated. The objective statement is listed as follows:

Objective 1: To develop and optimize a Machine Learning Module which can identify a human, hand, fire, and cooking utensils (e.g., pot).

Objective 2: To develop and optimize a Machine Learning Module which can identify if those classes exist together.

Objective 3: To embed the AI Module mentioned in Objective 1 into Small Single-Board Computers with RGB camera modules and alarm systems.

Compared to current CCTV, we hope that our project can collect vital data and report dangers to guardians immediately. The main difficulties of this project are the use of the RGB camera as the input source and using AI to perform analysis. We hope that our system can not only help guardians respond to emergencies as soon as possible, but also prevent accidents.

Methodology:

Our product’s methodology includes A.I. module training and optimization, implementing this A.I. into an embedded system with an alert system and other needed features.

Main result:

We achieved a streaming service of computer vision of 1 frame per second (1FPS). The test results show that the model has around 80% accuracy. Embedding system can function at the desired range of 2 to 3 metres away.

## References

1. M. Setchell, “How to blend 80x60 thermal and 640x480 RGB image?” Stack Overflow, 11-Sept-2015. [Online]. Available: https://stackoverflow.com/questions/32493861/how-to-blend-80x60-thermal-and-640x480-rgb-image. [Accessed: 11-Sep-2022].
2. iGuardStove [Overview – iGuardFire](https://iguardfire.com/overview/)
3. M. Ivašić-Kos, M. Krišto, and M. Pobar, “Human detection in thermal imaging using YOLO.” Human detection in thermal imaging using YOLO, Rijeka, Apr-2019.
4. “Falling detection system based on machine learning,” IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/document/7396449. [Accessed: 14-Sep-2022].
5. J. Nogas, “JJN123/fall-detection: Non-invasive fall detection with Keras and TensorFlow,” *GitHub*, 03-Feb-2021. [Online]. Available: <https://github.com/JJN123/Fall-Detection>. [Accessed: 15-Sep-2022].
6. James Robert Green Thermal Imaging for Assisted Living at Home: Improving Kitchen Safety[(PDF) Thermal Imaging for Assisted Living at Home: Improving Kitchen Safety (researchgate.net)](https://www.researchgate.net/publication/267783436_Thermal_Imaging_for_Assisted_Living_at_Home_Improving_Kitchen_Safety)

## Appendices

## A-Final project schedule

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Objective Statements** | 09/22 | 10/22 | 11/22 | 12/22 | 01/22 | 02/22 | 03/22 | 04/22 | 05/22 |
| Research | | | | | | | | | |
| Scope and objective definition |  |  |  |  |  |  |  |  |  |
| Literature review |  |  |  |  |  |  |  |  |  |
| Software and hardware exploration |  |  |  |  |  |  |  |  |  |
| Implementation | | | | | | | | | |
| Improve the accuracy of the camera |  |  |  |  |  |  |  |  |  |
| Analysis of the data of images by AI |  |  |  |  |  |  |  |  |  |
| Collect dataset |  |  |  |  |  |  |  |  |  |
| label the object |  |  |  |  |  |  |  |  |  |
| input the dataset and visualization into google colab |  |  |  |  |  |  |  |  |  |
| Training the AI model |  |  |  |  |  |  |  |  |  |
| Buy Component |  |  |  |  |  |  |  |  |  |
| Hardware finetuning |  |  |  |  |  |  |  |  |  |
| dealing with the background data |  |  |  |  |  |  |  |  |  |
| Documentation | | | | | | | | | |
| Proposal |  |  |  |  |  |  |  |  |  |
| Monthly Report |  |  |  |  |  |  |  |  |  |
| Progress Report |  |  |  |  |  |  |  |  |  |
| Final Report |  |  |  |  |  |  |  |  |  |
| Presentation | | | | | | | | | |
| Midterm Poster Design |  |  |  |  |  |  |  |  |  |
| Midterm Poster Design |  |  |  |  |  |  |  |  |  |
| Final Poster Design |  |  |  |  |  |  |  |  |  |
| Final Presentation |  |  |  |  |  |  |  |  |  |
| Video Making |  |  |  |  |  |  |  |  |  |

## B- Budget

Table 3 shows the expected costs of our project.

Table 3: Budget

|  |  |
| --- | --- |
| **Item** | **Cost (HKD)** |
| RGB Camera | 30 |
| Raspberry Pi | 985 |
| Breadboard | 50 |
| LED | 30 |
| Buzzer | 30 |
| Other components | 300 |
|  |  |
| **Total** | 1430 |

## C-Meeting Minutes

1. Date: 26/7/2022

Time: 6:30pm

Location: Zoom

Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin and Prof. Mow

Absent: All present

Progress Report: NIL

Discussion Items

All the members discussed about what to be expected.

Prof. Mo mentioned some past projects as reference, including auto-VTOL drone and Activity detection.

Future Plan:

Prof. Mo suggested members trying to get familiar with AI algorithms.

Prof. Mo also recommended members to search for information about RGB-D and thermal camera, including the trend and limitation.

Next Meeting: 06/09/2022 1800

1. Date: 06/09/2022  
   Time: 6:00pm

Location: HKUST, Room 6546

Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin and Prof. Mow

Absent: All present

Progress Report:

Members have tried using some of the AI recognition tools, including TensorFlow and OpenCV. One of the members said that he would take a related course for further knowledge.

Discussion Items:

Members discussed about what types of cameras shall be used in the project.

If thermal camera and RGB camera combine, the accuracy may be improved. Prof. Mo suggested some examples, such as camera for elderly, thermal CCTV and thermal gun.

Current products of combined cameras were discussed. The commercial products had different issues that the project might improve, such as the need of light source and violation of privacy.

Prof. Mo mentioned the concept of “line of sight”, and multi-modal training. He also suggested 3 possible projects, including calibration of thermal camera by depth camera, privacy-aware camera, and super-resolution cameras.

Future Plan:

Members agreed to select and complete the project topic and finish the proposal, including survey of camera components, and setting benchmark.

Division of labour was set. Chui would mainly be responsible for the mixed topic, Chung and Choy would be responsible for software and hardware issues respectively.

Next Meeting: 80/09/2022 1630

1. Date: 08/09/2022  
   Time: 4:30pm  
   Location: HKUST, Shaw auditorium  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
   Absent: All present  
     
   Progress Report: NIL  
     
   Discussion Items:  
   FYP topic, focus, and challenge were finalized.   
   Members did online research to check the feasibility of the project topic.  
   After that, we discussed job duty distribution.  
     
   Future Plan:  
   Choy would handle the planning and budget part; Chung and Chui would focus on the technology research and review.  
   Members are reminded to finish and proofread their respective parts and hand them in on time.  
   Continuous communication with Professor via E-mail is expected.  
     
   Next Meeting: 06/10/2022 1800
2. Date: 06/10/2022  
   Time: 6:00pm  
   Location: HKUST, Room 6546  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin and Prof. Mow  
   Absent: All present  
     
   Progress Report:   
   Initial report and documentation had been finished.  
     
   Discussion Items:  
   FYP topic, focus, and challenge were revised.   
   Prof. Mow gave advice of enhancing research processes and pointed out some issues of the report.  
   After that, we discussed the workflow of the project.  
     
   Future Plan:  
   Prof. Mow advised that extra features could be extracted. Further study and test will be done.  
     
   Next Meeting: 23/10/2022 2030
3. Date: 23/10/2022  
   Time: 8:30pm  
   Location: Zoom  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
   Absent: All present  
     
   Progress Report: NIL  
     
   Discussion Items:  
   We finished the first progress report.  
     
   Future Plan:  
   The team will investigate on the uninhabited fire detection with RGB-camera.   
   The team would also investigate the method to achieve the above function.   
   Collecting data for AI training is required.  
     
   Next Meeting: 26/10/2022 2100
4. Date: 26/10/2022  
   Time: 9:00pm  
   Location: Zoom  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
   Absent: All present  
     
   Progress Report:   
   We have sought possible cameras on the market and looked for more possibilities about our project.  
     
   Discussion Items:  
   We edited the first progress report.  
   We have decided to continue our project as “household usage” project.  
     
   Future Plan:  
   We will continue in learning AI and related mechanism to strengthen our base.  
     
   Next Meeting: 17/11/2022 1630
5. Date: 17/11/2022  
   Time: 4:30pm  
   Location: HKUST, Shaw auditorium  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
   Absent: All present  
     
   Progress Report:  
   Chui and Chung have been studying in AI-related courses and brought back some information of Computer Vision and AI architecture, further knowledge expected.  
     
   Discussion Items:  
   We discussed on some reviews of AI and Computer Vision,  
     
   Future Plan:  
   Pre-processing of videos and images are to be done.   
     
   Next Meeting: 27/11/2022 2300
6. Date: 27/11/2022  
   Time: 11:00pm  
   Location: Zoom  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
   Absent: All present  
     
   Progress Report:   
   System architecture is formed, it will make use of YOLOv5 as backbone.  
     
   Discussion Items:  
   We have checked the price of cameras on different online shopping platforms. There are a few viable options.   
   We have learnt about the formal for image processing, similar to the photo analysis by MATLAB in ELEC2100.  
   Lastly, we reviewed some of the existing similar works.  
     
   Future Plan:  
   We will continue to collect images for dataset.  
     
   Next Meeting: 4/12/2022 2300
7. Date: 4/12/2022  
   Time: 11:00pm  
   Location: HKUST, Shaw auditorium  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
   Absent: All present  
     
   Progress Report: NIL  
     
   Discussion Items:  
   Revision of monthly report have been done.  
     
   Future Plan:   
   We will learn from the past model and develop our specific model to detect action patterns (I.e., unattended fire). This will be done by reading more review papers and compare the effectiveness.   
   We will also learn to convert image sources into AI. We will try to use RGB cameras to “teach” the AI about edge detection.   
   We will record videos in our kitchens and also try to introduce some online videos of unattended fire on social medias.  
     
     
   Next Meeting: 30/12/2022 2200
8. Date: 30/12/2022  
   Time: 10:00pm  
   Location: Zoom  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
   Absent: All present  
     
   Progress Report: NIL  
     
   Discussion Items:  
   The mid-term poster is revised for submission.   
   Work and progress were re-assured.   
   Roboflow was selected as the tool of labelling images. The format is set as 640x640 RGB JPEG files.  
     
   Future Plan:  
   Members are expected to record some more cooking videos for training the AI model.   
   The search of online images is also encouraged.  
     
   Next Meeting: 9/1/2023 1200
9. Date: 9/1/2023  
   Time: 12:00noon  
   Location: HKUST  
   Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
   Absent: All present  
     
   Progress Report:   
   Members have recorded, snipped and labelled images to be tested. The images were stored in a cloud storage. Research on using YOLOv5 have been enhanced.  
     
   Discussion Items:  
   We divided the work of the Progress Report.   
   We have been training the model for better accuracy.   
   We also found ways to visualize results for better acquisition of the situation.  
   There were faulty detections (i.e., false positives and repeating labels on the same object) in the system, possible solution are to be found.  
     
   Future Plan:  
   The team will continue to improve the dataset by finding more online images and recording daily video about the usage of stove fire.  
   Labelling of current images will be adjusted and unified for AI to learn easily.  
     
   Next Meeting: TBC
10. Date: 9/2/2023  
    Time: 15:00  
    Location: HKUST  
    Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
    Absent: All present  
      
      
    Discussion Items:  
    setting up the background system of the raspberryPi  
    Future Plan:  
    install the camera and yolov5  
    Next Meeting: 23/2
11. Date: 23/2/2023  
    Time: 15:00  
    Location: HKUST  
    Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
    Absent: All present  
      
      
    Discussion Items: install the camera, we many methods to install the camera.However it cannot work.The image captured is a picture with a lot of pink vertical line.  
    Future Plan: find out the reason it cannot capture image
12. Date: 7/3/2023  
    Time: 12:00noon  
    Location: HKUST  
    Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
    Absent: All present  
      
    Discussion Items:  
    We cannot find the reason of the error, so we consider the problem could be incompatitable camera software. We buy a new camera. The problem has solved. We can capture the image by the camera  
    Future Plan:  
    We will download the yolo to the raspberryPi and run the AI model.  
    Next Meeting: TBC
13. Date: 18/4/2023  
    Time: 12:00noon  
    Location: HKUST  
    Attendees: Chung Wai Lok, Chui Chi To, Choy Yu Hin  
    Absent: All present  
      
    Discussion Items:  
    After separate investigation and communication to the vendor, we assured that the camera is faulty, and its circuit could be burnt.  
    We combined what we have made and finished the project. We investigated on the reason behind low FPS and slow performance.  
    Chung suspected if the Raspberry Pi CPU computation power be the limiting factor   
    .Chui tried with other model format for improving performance.  
    Future Plan:  
    We will finish the documentation, report and poster.  
    Next Meeting: TBC

## D- Group Members’ Contributions

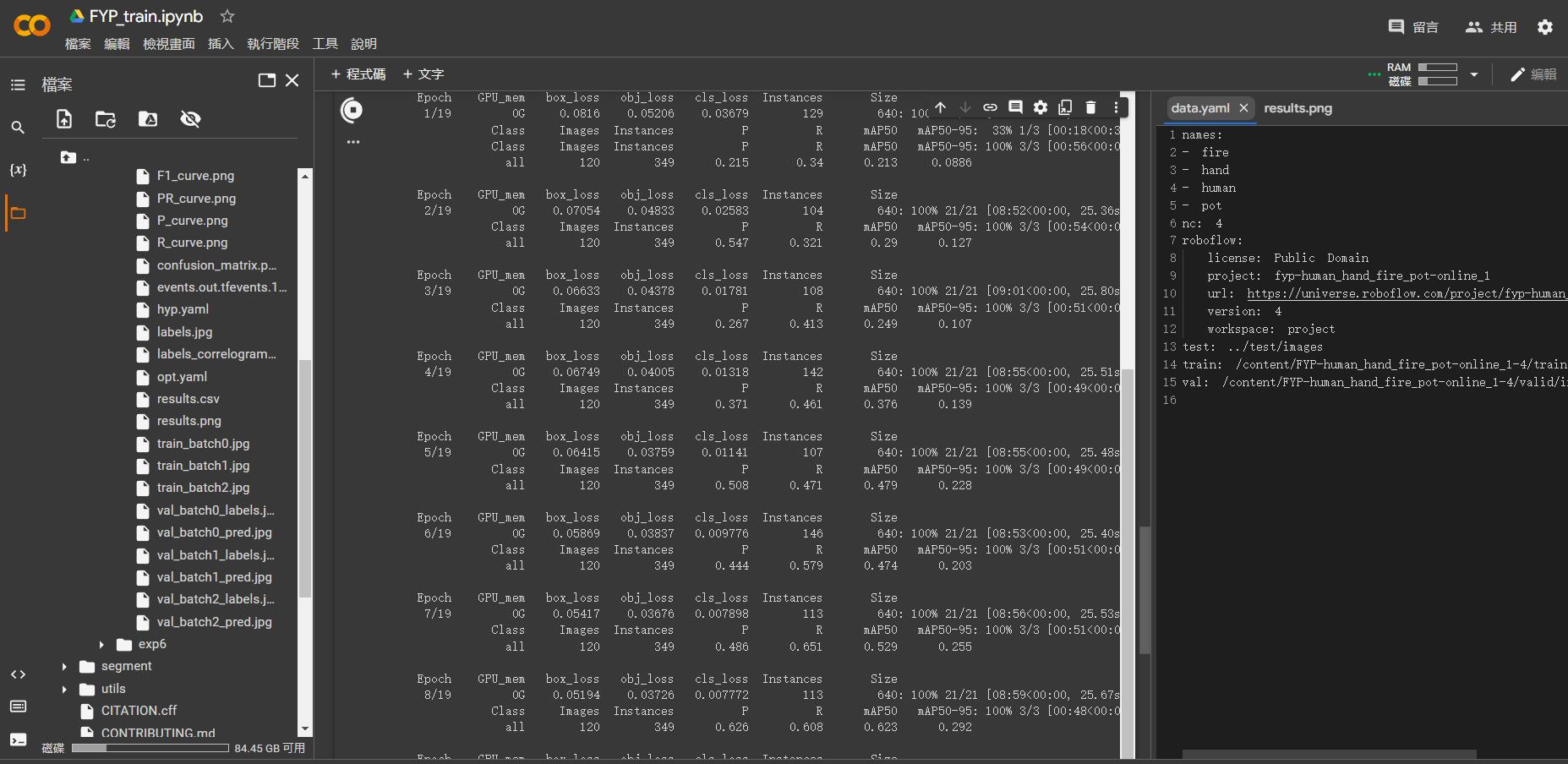
**Chung Wai Lok:**

I am responsible for the AI training in Google Colab. To compare the result, I used Roboflow’s integrated AI training tools as the project’s opponent.

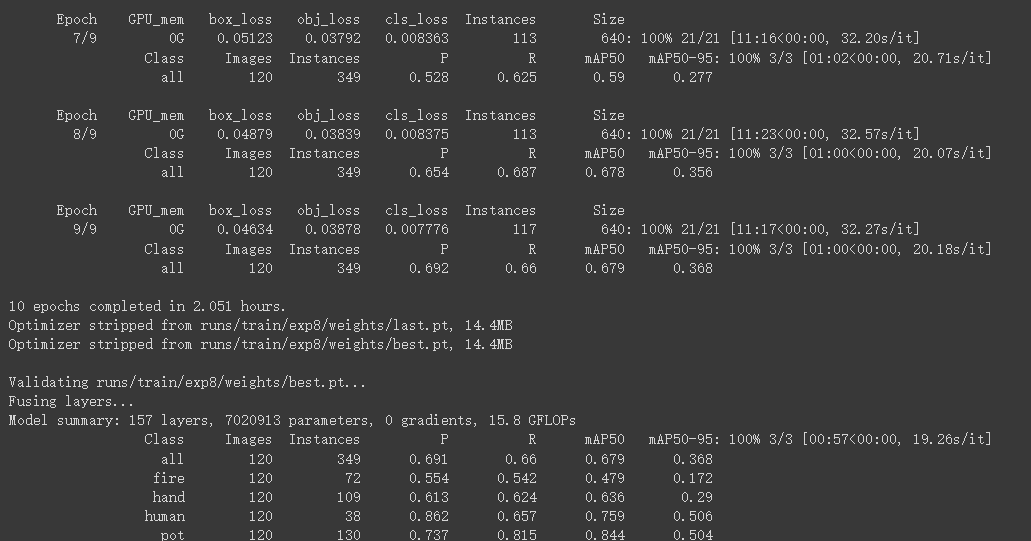
The main difficulty is the combination of different AI tools. AI training is very time-consuming, and it requires many computing resources. Thus, I chose Google Colab as it provides the function of cloud computing. Even so, some built-in features (inactivity alerts, resource limit, etc.) limit our attempts.

Another problem is related to coding. Current tools rely on different styles of addressing and there are conflicting settings that cause unique errors, which require manual “try and error”. Eventually, I solved the issue by updating the address of files manually each time it got updated.

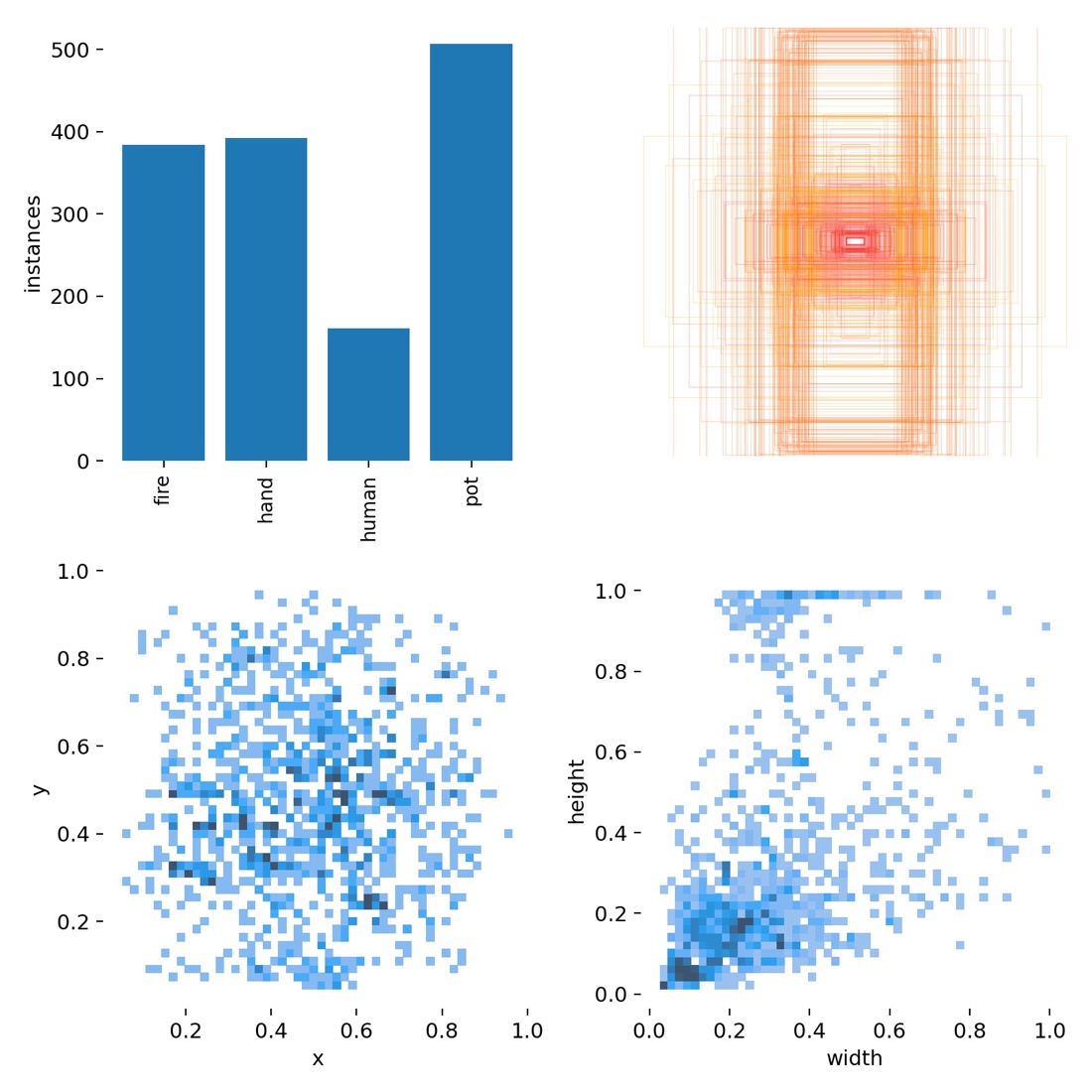
The first batch of results is non-satisfactory due to low epochs and inadequate samples. We communicated and decided to unify our labeling style. I also increased the variables for better results.



The attempt was successful. However, due to the imbalances between tags, the performance for detecting pots remains outstanding, which is unexpected.



I will continue to improve and test the model on two perspectives: input and training, including training procedure and the variety of photos. After some correction to images, the model performance improves.



During the embedding stage, I am responsible for software development, including checking and optimization of the program. During the testing, the program can take a video input and process frame by frame.

|  |
| --- |
| import cv2  cap = cv2.VideoCapture("./out.mp4") pos\_frame = cap.get(cv2.CAP\_PROP\_POS\_FRAMES) #initialization while True:  flag, frame = cap.read()  if flag:  cv2.imshow('video', frame)  pos\_frame = cap.get(cv2.CAP\_PROP\_POS\_FRAMES)  print str(pos\_frame)+" frames"  else:  # The next frame is not ready, so try to read it again  cap.set(cv2.CAP\_PROP\_POS\_FRAMES, pos\_frame-1)  cv2.waitKey(1000)   if cv2.waitKey(10) == “q”:  break  if cap.get(cv2.CAP\_PROP\_POS\_FRAMES) == cap.get(cv2.CAP\_PROP\_FRAME\_COUNT):  # end If the number of captured frames is equal to the total number of frames (end of file)  break |

After proving that the YOLO model is successful, I helped in converting streaming input from the MIPI camera to usable RGB array. The camera’s format is slightly different to our trained images and tested videos. Thus, I introduce Alpha (contrast) and beta (brightness) change to the captured array. In the testing, I found that increasing alpha helps improving the performance, as high-contrast images can show clearer features on image.

|  |
| --- |
| #constant alpha = 10 beta = 0.95  while True:  flag, frame = cap.read()   im= cv2.convertScaleAbs(frame, alpha, beta) |

I have also worked on the detection algorithms, including the weighted scores and “forgetful” functions. By having a “past” component, the program can keep the awareness to visually blocked flame, and only disarm if there is no fire. The difficulty is that it is hard to balance between “long inactivity” and “long active” situations. While I prefer a more conservative mode (being sensitive), there could be false alarms, reducing the credibility of the system. At last, I compromised and reducing the timed factor and setting upper limits of currentScore.

**Chui Chi To:**

In this FYP group my most contribution is coordinate each part of the project, such as goal setting, leading the discussion, work distribution, deadlines planning, etc. In order to have an organised workflow, a todo-list word file was created for clear aims and steps to achieve the aims. The word document link is shown as follows: <https://hkustconnect-my.sharepoint.com/:w:/g/personal/ctachui_connect_ust_hk/Ed3HKmWxz6JFjLVIv1lQ0I8BCQWwDoh_cUltFuT6MzqrHA?e=o6aWYQ>.

Hance, each of our groupmate have found 200 images and label they though Roboflow.com for the dataset. I found 100 images online, from a copyright free website, called Unsplash. The other 100 images were obtained by extracting frame in our recordings, which is recorded from different angle, pot and human in the recordings.



Figure 1 online images of cooking found on Unsplash



Figure 2 self-record video screen cap

To demonstrate a visualization for our dataset, a Google Colab Jupytor notebook was created. The link was shown as follows: <https://colab.research.google.com/drive/1aGRU7MCotNLu76WcDGrYZ17me6RMM1Pd?usp=sharing>. This Colab PyTorch notebook are able to a) plot some image of our dataset, b) plot the labels from the correct file of each image, c) show multiple images at once. To do so, first create two path lists ‘*test\_imgs’* and ‘*test\_labels’,* whichholds all the file path of test images or labels. Then reorder all path in *‘test\_labels’,* to be synchronised with *‘test\_img’,* so that test\_labels[0] is the label of test\_imgs[0]. Then, using “Matplotlib” to plot multiple image a one figure. To draw the bounding box from string in label file, we first transform the YOLOv5 format, which shows the Box centre of x, y, width and high in 0, 1 scale, to left bottom conner x, y width and high in real pixel. Finally, plot them out with matplotlib.patches.Rectangle() functions to visualize the labels with their bounding box and class names.



Figure 3 YOLOv5 bounding box format

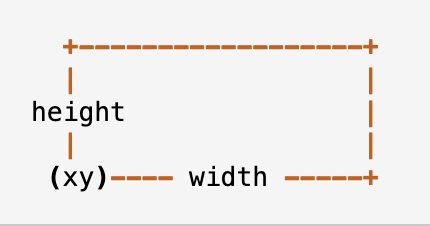


Figure 4 matplotlib.patches.Rectangle() input format

**Choy Yu Hin:**

****I think the most challenging task until now is handling and collecting the data source. When I first time record the video, I don't know which part is important to capture for the AI to identify. At first, we collect 200 photos per person. After our first time AI training, we find out that fire is difficult for the AI to Identify. After our discussion, we decide to increase the sample of fire or photos including fire.

****From this experience, I learn that there are some measures of identification e.g., loss, and accuracy. Moreover, I learn how to improve the accuracy of the system, which is to make the sample boundary more concise and increase the sample of the class which has low accuracy. We should have the same standard to label the class, for example, the hand. When we compare our photos, we find out some hand labels include the front arm, and some only include the palm. As such we must have the same standard when lableling the class. I also learn that 600 photo is not enough for the AI system to recognize the object correctly.

I am also responsible for buying the material. I find out that there are a lot of AI boards in the market. After comparing the price and function, we decide to buy the RaspberryPi4.After I start to learn about the RaspberryPi, I find out it has a huge difference between the Arduino board we use in ELEC1100 and 2400. It can connect to much more components which include the RGB camera that we need. Now, I am learning how to use the RaspberryPi to run our AI. Because we need to finish the software part first. After the software part is finished, I will start to find the other component such as the buzzer and the circuit for our last part, which is the circuit hardware part.

## E- Deviation(s) from the proposal and supporting reason(s)

C1. Re-definition of the usage

We redefined our project from “household security camera” to “unattended fire detection and warning”. The reason is that we underestimated the universality of security cameras on the market and the development of thermal cameras in the AI field. The root cause is the imbalanced search for information. The error was corrected, so we adjusted the project to a more specific and less common one.

C2. Discarding the function of fall detection

We chose not to introduce “fall detection”. There are already viable databases and AI models that can achieve that. Their results were impressive, and we would like to try something that is challenging.