

CookSmart : Photo to Recipe

Interim Report

TU856

BSc in Computer Science

**Jamie Heffernan**

**C20483462**

**Supervisor**

**Richard Lawlor**

School of Computer Science

Technological University, Dublin

**20/11/2023**

Abstract

Studies have shown a clear link between cooking skills and food choices in the home. Those who express the belief that they can’t cook tend to be more reliant on takeaway food, pre-prepared ready meals or rely on another person to provide their food. This reliance on convenience has been shown to lead to a subconscious over-consumption of fats, salts and sugars [2].

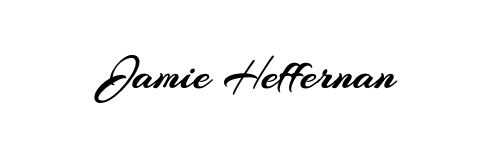
Due to changing family dynamics, the acquisition of cooking skills is in decline, reduced interest in home-economics classes in schools and reduced hours to practice and refine these skills have led to a decline in home cooking and cooking skills, which are essential for regular health. Surveys have shown that time pressure, as well as not knowing what to cook or where to start are driving factors as to why people don’t cook at home. [5][1]

My application seeks to solve this issue by not only giving a user option as to what they can cook at home without making a trip to a shop as well as providing nutritional information and step by step instructions to a simple home-cooked meal that can hopefully improve their cooking skills and overall health if it reduces take-away consumption.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:



Jamie Heffernan

20/11/2023

Acknowledgements

I would like to express my sincere gratitude to my project mentor Mr. Richard Lawlor for his invaluable help and unwavering support throughout the development of my final year project. His expertise, guidance, and thoughtful insights have been instrumental in shaping the trajectory of my application. Mr. Lawlor's willingness to share his knowledge and provide constructive feedback have greatly enriched the quality of this project. I am truly appreciative of his mentorship, which has been a guiding light in navigating the complexities of this academic journey. Thank you, Richard, for your indispensable contribution to the successful completion of my final year project.

[1. Introduction 8](#_Toc152955150)

[1.1. Project Background 8](#_Toc152955151)

[1.2. Project Description 8](#_Toc152955152)

[1.3. Project Aims and Objectives 9](#_Toc152955153)

[1.4. Project Scope 9](#_Toc152955154)

[1.5. Thesis Roadmap 10](#_Toc152955155)

[2. Literature Review 11](#_Toc152955156)

[2.1. Introduction 11](#_Toc152955157)

[2.2. Alternative Existing Solutions 11](#_Toc152955158)

[2.3. Technologies Researched 12](#_Toc152955159)

[2.4. Other Research 13](#_Toc152955160)

[2.5. Existing Final Year Projects 14](#_Toc152955161)

[2.6. Conclusions 15](#_Toc152955162)

[3. System Design 16](#_Toc152955163)

[3.1. Introduction 16](#_Toc152955164)

[3.2. Software Methodology 16](#_Toc152955165)

[3.3. Overview of System 16](#_Toc152955166)

[3.4. Conclusions 19](#_Toc152955167)

[4. Testing and Evaluation 20](#_Toc152955168)

[4.1. Introduction 20](#_Toc152955169)

[4.2. Plan for Testing 20](#_Toc152955170)

[4.3. Plan for Evaluation 20](#_Toc152955171)

[4.4. Conclusions 20](#_Toc152955172)

[5. Prototype Development 21](#_Toc152955173)

[5.1. Introduction 21](#_Toc152955174)

[5.2. Prototype Development 21](#_Toc152955175)

[5.4. Conclusions 27](#_Toc152955176)

[6. Issues and Future Work 28](#_Toc152955177)

[6.1. Introduction 28](#_Toc152955178)

[6.2. Issues and Risks 28](#_Toc152955179)

[6.3. Plans and Future Work 28](#_Toc152955180)

[6.3.1. GANTT Chart 29](#_Toc152955181)

[Bibliography 30](#_Toc152955182)

# 1. Introduction

## Project Background

In 2021, Ireland alone generated over 750,000 tonnes of food waste, with households accounting for 29% of this waste at 221,000 tonnes. That is more than the entire food and beverage sector (215,000 tonnes). [4]

Food waste costs the average household €60 per month, costing the nation a total €1.29 billion annually. This is equal about 120 kg of food waste per household or 44 kg per person (that’s about half the weight of a full brown bin). [4]

The Government of Ireland aims to reduce all food waste by 50% by the year 2030, aligning with the UN sustainable development goals. However, despite progress being made, there's room for improvement in reducing and managing food waste in the household through behavioural changes. [4]

When shopping for food items, retailers typically try to sell as many of an item as possible regardless of its contribution to waste or not, for example loose vegetables have begun to disappear from shelves in recent years and have been replaced with multipacks so that instead of buying one or two loose onions, you now get six or so resulting in higher profits margins for the retailers and higher waste production when these items inevitably go unused and in the waste bin.

The goal of my project is to provide a user with a simple application which will allow them to take a photograph of the ingredients in their fridge left at the end of the week and provide them with a simple recipe that they can make with these leftover ingredients

## Project Description

The aim of my application is to solve the age-old question of “What’s for dinner?” by allowing a user to simply take out their phone and photograph the ingredients they have in their fridge or pantry and use the power of machine learning and artificial intelligence to dynamically generate a recipe based on the ingredients extracted from the photo in a cuisine of their choosing.

This application departs from existing solutions where a user manually types a list of ingredients into a text prompt which then searches through a database to find a pre-existing recipe which they may not have the time nor ingredients for. By leveraging artificial intelligence, the application can generate dynamic and unique recipes based on recognised ingredients and providing users with an instantaneous solution to a meal planning problem.

## Project Aims and Objectives

* Object Detection
  + The app will employ an object detection machine learning algorithm to identify various ingredients when a user takes a photo
  + Real time object recognition will ensure accurate and simultaneous detection of multiple ingredients in real-time
* Recipe Generation
  + Upon identifying ingredients, the app will then clean the ingredients list and pass this to a large language model (LLM) where with the help of some prompt engineering, it will dynamically generate a recipe based on the detected items
  + The user will then receive a new ingredient list with some optional additional ingredients such as seasonings and step-by-step instructions on how to cook the recipe
* Saving a Recipe
  + If the user generates a recipe that they like or would like to cook again they will have the option to save this recipe, this will add the recipe to their cookbook where they can come back to it at a later date if they desire.
* User Friendly Design
  + A Simple and intuitive user interface will help to make the application accessible to all cooking enthusiasts regardless of their culinary experience
  + A minimalistic layout will ensure a seamless and enjoyable user experience
* Mobile Application
  + Develop a visually appealing and responsive mobile interface for the application to replace the command line interface of the proof of concept
  + Optimise this for various device screen sizes and orientations

For the interim submission I plan to have a proof-of-concept prototype, where the application can successfully detect a limited selection of objects and use this list of objects to generate a recipe. Once the proof of concept has been created and deployed, I will work on further training the algorithm to detect more food objects and create a nice user interface so that users can interact with the system in an intuitive and seamless manner

## Project Scope

There are four main aspects to the project, this includes the object detection model, large language model and the application front end and the application back-end. While in an ideal scenario the application would allow users to take a photo of their fridge and generate a recipe from there, in reality, the training of an object detection machine learning algorithm is complex and time consuming so we must limit the scope of the algorithm. For this reason, I have decided to compile a list of the most popular fruits and vegetables the average household has in their fridge and focus primarily on training the algorithm to quickly and accurately detect these.

## Thesis Roadmap

**Literature Review:**

Discusses existing methods of detecting objects as well as generating recipes, their advantages and limitations and overall feasibility within this scope to establish the best methods of executing this project.

**System Design:**

This section breaks the project down, detailing the system architecture and features of the current prototype, its limitations and future work to bring this project from a proof of concept to a full application.

**Testing & Evaluation:**

Discusses the methods of testing the individual components of the application as well as the application as a whole, to ensure a robust and feature rich application that works in all cases.

**Prototype Development:**

This section will explore the development process of initial prototype of the system which serves as a proof of concept for the final application

**Issues & Future Work:**

Explores the proof of concept, what it succeeds in and where its limitations lie, how we can improve upon the current version and work we can complete in the coming months to bring the application to life.

# Literature Review

## 2.1. Introduction

Within the world of computer science, it is crucial to grasp what technologies and solutions have already been explored before starting the development of a project. In this chapter, I will be exploring the current state of object detection, large language models and testing existing solutions and research relevant to my final year project.

By taking a closer look at what is already out there I aim to understand the strengths and weaknesses of similar projects, what works, what doesn’t and trying to make sure that I don’t have the same pitfalls others have. I will compare my project to existing solutions, previous final year projects and peer reviewed research to better understand my project and see what makes it unique, my aim is to sum up what has been explored so far and how it can light the way for the development of my project

## 2.2. Alternative Existing Solutions

**DishGen**

DishGen is an AI powered recipe generator with the goal of transforming the way a user plans their meals. It works by having a user input their desired ingredients, recipe ideas or dietary preferences and using an “advanced algorithm” crafts a unique recipe based on this input.

While the specifics of their recipe database are not disclosed, DishGen likely sources its recipes from a curated or licensed collection of recipes. The platform offers both Basic and Premium accounts, with the latter offering unlimited recipe credits and access to the Idea Generator Tool. This tool can provide 7 different suggestions for any requirements or list of ingredients, adding a creative and personalized touch to meal planning.

**PixFood**

PixFood is a food recognition app which allows users to discover recipes based on pictures of ingredients, by using machine learning, PixFood analyses images of food items and suggests relevant recipes. Its goal is to simplify the recipe search process by allowing users to find recipes through visual input rather than text-based searches.

Although they are secretive about their recipe database PixFood likely sources its recipes from a database or collection of recipes curated or licensed for the application.

**LEGO BrickIt**

BrickIt is a mobile app designed for LEGO enthusiasts, it uses modern computer vision technology to change the way users interact with their LEGO collections. It allows users to photograph their LEGO pieces and the app will then analyse the brick type and colours and give users step by step building instructions for various LEGO models, adding an interactive and imaginative layer to the LEGO building experience.

## 2.3. Technologies Researched

In this section I will discuss possible technologies that can be leveraged to provide a solution to our problem, we will be researching object detection machine learning models to detect food items as well as large language text generation models to generate a recipe from this list

**YOLOv8**

You Only Look Once or YOLO is an advanced object detection model known for its speed and accuracy in real time. YOLOv8 is the eighth version of the YOLO model. It employs a unified neural network that simultaneously predicts bounding boxes and class probabilities for multiple objects within an image [6].

* Grid Division
  + It works by first dividing the image into a grid of cells, creating a structure for analysis
* Bounding Box Prediction
  + It uses pre-defined bounding boxes of various sizes to help plot bounding box coordinates more effectively
* Class Probability Estimation
  + For each grid cell, YOLOv8 predicts bounding box coordinates, the ‘objectness score’ and class probabilities for multiple objects within the camera frame
  + It then applies a score threshold to filter out predictions with a lower confidence. Only predictions above a specified threshold are considered

**Roboflow**

Roboflow is a platform we can employ to facilitate the management of our custom trained YOLOv8 model, it provides us with tools to help us prepare our training data as well as deploy our model once training has been completed

It provides annotation tools which support multiple modern formats such as COCO which will make it easier to work with our custom training data in the YOLOv8 training process as well as allows for various data augmentations such as scaling, rotating and noise addition to diversify the model and improve its robustness [6]

Once our YOLOv8 model has been trained, we can use the Roboflow inference API to make predictions on our trained model through HTTP requests, this will allow us to integrate the model into a web or mobile application while keeping the application itself lightweight and not relying on the processing speed of various mobile devices

ChatGPT 3.5 Turbo

ChatGPT Turbo builds upon its predecessor GPT-3.5 with an increased focus on improving the context recognition issues and response coherence aspects of GPT-3 [9]. The model offers several improvements over previous models that make it suitable for our project such as

* Increased Parameter Count
  + GPT Turbo offers a significantly improved parameter size which enables us to use a vast array of culinary experience from the models' training data, making it well suited for the generation of diverse and high-quality recipes
* Fine Tuning Text Response
  + The GPT Turbo model allows for fine tuning to generate specific outputs, for instance we can specify a JSON schema to the model and limit the response to fit this format
* Contextual Relevance
  + GPT Turbo’s enhanced context handing ensures that generated recipes will maintain their structure and coherence, the model will understand the context of each step and if changes are made to the recipe and it gets regenerated, the original recipe will be altered rather than generating a whole new recipe

For the reasons outlined above and the limitations of current large language models ChatGPT-3.5 Turbo is the best suited LLM for our recipe generation requirements

## 2.4. Other Research

**Theos AI**

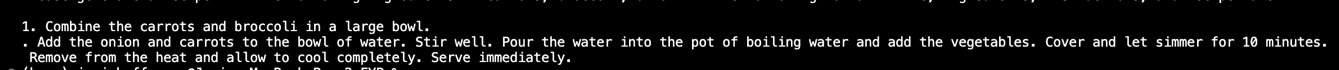
Theos AI is another platform we can possibly use to manage our YOLOv8 model, it offers similar tools to manage and prepare data for the management of our YOLOv8 model.

TheosAI annotation tools also support COCO format to facilitate our training data, and provides a more user-friendly interface for the annotation of the data however they rely on the axios library to make HTTP requests to perform inference on an image or video stream

**Open AI GPT 2**

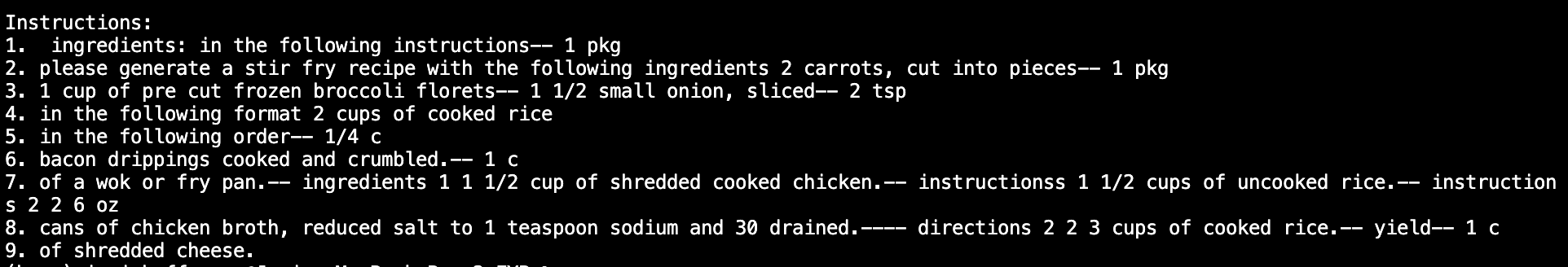
Prior to the release of ChatGPT 3 to the public, OpenAI had created the 2nd iteration of their generative pretrained transformer, it is proficient in generating coherent and contextually relevant text and more importantly the API for GPT2 is offered by OpenAI for free, making it an ideal candidate for our recipe generator [9]

However, upon testing the GPT 2 API I found that the responses were less reliable, accurate and does not offer the same level of fine control that GPT 3 offers. The quality of recipes generated by GPT 2 were quite low, with many instances the user is simply told to add all ingredients to a pot of boiling water for 10 minutes



**Google Flan T5**

T5 or Text-To-Text Transfer Transformer is a transformer-based language model created by Google. It is designed to handle various language processing tasks [8], the “text-to-text" aspect refers to both the input and output being treated as text. This versatility allows for translation and summarization but lacks the robust text generation that the OpenAI models offer. Upon first glance it worked very well for generating recipes however after some further testing it seemed to only generate stir-fry recipes rather than a diverse range of recipes, possibly due to the training data provided by Google



**ChatGPT 3 Davinci**

ChatGPT 3 was the first version of ChatGPT released to the public for experimental use and was the version of GPT which changed the world as we know it. ChatGPT 3 exhibits a remarkable proficiency in understanding and generating human-like text. Its large parameter counts, and contextual awareness makes it a powerful tool for generating recipes however the GPT 3 API plans to deprecate the use of GPT 3 for its later 3.5 and 4 models in January of 2024 which given our development time frame makes this model unsuitable for the task at hand

## 2.5. Existing Final Year Projects

**ClearMyFridge**

ClearMyFridge is a web application that allows a user to enter ingredients they wish to cook with. The aim of the web application is to reduce food wastage in the home, make cooking easy and inspire users to cook new recipes. The user can enter ingredients by the ingredient name or scanning the barcode on its packaging. The user can then view recipes containing the ingredients they entered. When a user logs in the recipes can be saved.

This app takes user input in the form of a text entry and calls an API with a database of recipes, my application seeks to improve on this by allowing the user to enter ingredients simply by taking a picture of them and offers more dynamic recipes by incorporating a large language model to generate recipes from the photo, eliminating the need for a database

**Object Detection Aim Assist**

This project uses pre-trained weights for various video games, to detect selected features in the live screen recording feed it captures. Object detection has been around for many years and is now a common occurrence in many technologies that we take for granted. Detections made in security cameras are done by specially trained weights designed to capture features such as humans and animals. Using another face recognition algorithm, it can detect if the humans are known, residents.

The project uses YOLO, an object detection model which was trained to detect player movements in game similar to how a smart security camera might capture a person's movements to aid players in their gaming skills

The idea for using object detection in video games is a unique one which really captured my attention by training YOLO models to detect player movement and improve one's overall skills. This could be very similar to how I plan to identify ingredients within my own application however the moral aspect of the project must be examined as it can be easily adapted to cheat in online multiplayer games and many game developers use technology to detect modifications such as this which may lead to unassuming users simply trying to practice receiving bans from game developers

## 2.6. Conclusions

In conclusion, this comparative analysis has provided me with valuable insights into what makes my project unique, how it differs from previous similar projects and the improvements it makes on them. While drawing inspiration from previous endeavours and following a proven software methodology and aligning with industry standards, I believe my project stands out as a unique combination of innovative features which showcase the projects capacity for pushing the boundaries in the domain of computer science.

# 3. System Design

## 3.1. Introduction

In this chapter we will delve into the design of the application, and the integration of an object detection and large language machine learning model to generate a unique recipe for the user. The application will leverage the capabilities of the YOLOv8 object detection algorithm and pair it with the power of Open AI’s chat GPT language model. In this chapter I will discuss the various aspects of the application, the role they play in the overall functionality of the system and the relationships between them.

## 3.2. Software Methodology

Due to the fast-paced nature of the project and weekly meetings with my project supervisor Mr. Lawlor, I have decided to develop the project using the Scrum methodology as it focuses on iterative and incremental development, allows for flexibility and adaptability and returns a minimum viable product at the end of each sprint cycle

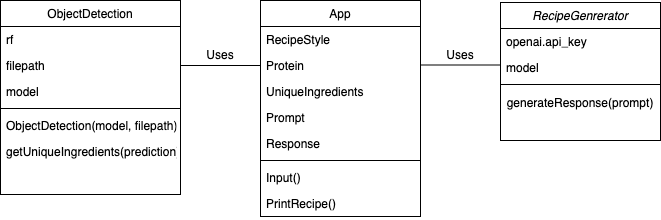
Scrum promotes an incremental development approach meaning that the product is slowly developed in incremental steps to slowly build a larger system, this is beneficial in our specific case as it allows us to make continuous improvements and adjustments based on feedback from my mentor and peers, ensuring that the project stays in line with its requirements

Scrum is also known for its high flexibility and ability to adapt to changing project goals. In the context of a final year project where we can have ever changing requirements, having the ability to dynamically adapt to changes ensures that the project stays in line with its objectives and goals

With the frequent reviews and retrospectives by peers and my project mentor we can use scrum to quickly address any risks in the developmental process, this proactive risk management approach is crucial in mitigating potential risks or issues that may arise during the development of the project

## 3.3. Overview of System

The system is made up of three main components, the object detector, the recipe generator and the main app.py



**App.py**

The App class serves as the entry point into the application. It interacts with the user and uses methods from the RecipeGenerator and ObjectDetection classes

* Attributes
  + RecipeStyle – The style of cuisine specified by the user
  + Protein – A boolean value indicating to the user if they wish to add a protein
  + UniqueIngredients – a list of unique ingredients for the recipe
  + Prompt – A string that forms the prompt for the recipe generator
  + Response – The final generated recipe
* Methods
  + Input – Takes user input
  + PrintRecipe – Prints the final recipe to the user

**RecipeGenerator.py**

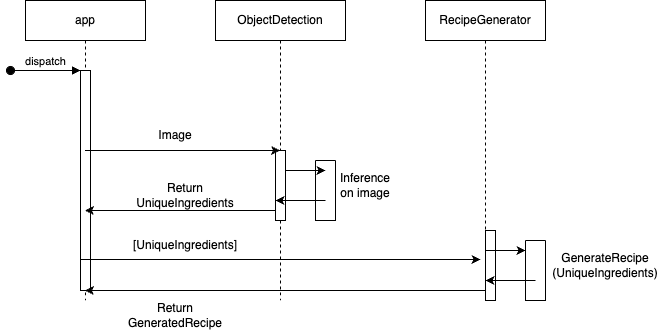
The RecipeGenerator class is responsible for generating a recipe based on the objects detected in the image given by the user using the OpenAI API

* Attributes
  + OpenAI.api\_key – Stores the API key for OpenAI authentication
  + Model – Specifies the GPT 3.5 Turbo model for generating the recipe
* Methods
  + GenerateResponse(prompt) - Generates a recipe based on the given input

**ObjectDetection.py**

The object detection class uses the RoboFlow API to detect objects in an image and extract a list of unique ingredients based on our trained model from the prediction results

* Attributes
  + Rf – Roboflow API key for authentication
  + Model – The model to be called for detecting objects in the image
  + Filepath – specifies the path to the location of the image to run inference on



From this diagram we can see that the ObjectDetection and RecipeGenerator do not directly interact with each other, instead only interacting through the main app.py file. This is to ensure low coupling so that with each iteration of each, the changing of one will not affect the other

The app.py calls the recipe generator file which takes an image in .jpg format where it runs inference on this image over the RoboFlow API, this then in turn detects any objects within the image and returns a cleaned list of these ingredients which is then returned to the main app file.

Once the main app receives the list of unique ingredients from the object detector it then passes this list to the recipe generator which creates a prompt for the list and passes this prompt to the OpenAI API where it runs the prompt through ChatGPT 3.5 Turbo and returns a recipe, this is then passed back to the main application where it is returned to the user in a print statement.

While the current prototype of the application does not incorporate a database for persistent storage, I do plan to include this in the final version so that a user can save a recipe they liked for a later date as well as share recipes with other users.



This database schema will allow the application to store multiple users that can save generated recipes to their accounts for later use. The entities and their relationships are defined as follows:

* **Users –** Represents the users with accounts on the platform
  + UserID – Primary Key
  + Username
* **Recipes –** Represents individual recipes
  + RecipeID – Primary Key
  + UserID – Foreign Key referencing Users table
  + Title
  + Calories
* **Ingredients –** Stores a list of ingredients
  + IngredientID
  + Name
* **RecipeIngredients –** Represents the ingredients in each recipe to associate the ingredients with their respective amounts
  + RecipeID
  + IngredientID
  + Amount
* **Instructions –** Contains the step-by-step instructions for a recipe
  + InstructionID – Primary Key
  + RecipeID – Foreign Key
  + StepNumber
  + Description
* **Notes –** Contains additional notes related to the recipe
  + NoteID – Primary Key
  + RecipeID – Foreign Key
  + AdditionalNotes

From this schema we can see that each recipe is associated with a user through the **UserID** foreign key in the **Recipes** table, allowing users to save their own recipes

The **RecipeIngredients** table connects recipes to their respective ingredients and amounts

The **Instructions** table links recipes to their instructions and the step number of these instructions

The **Notes** table associates additional notes with specific recipes

This schema will allow us to represent our recipes in a comprehensive way so that each recipe has its own ingredients, instructions, and additional notes while also facilitating users to have their own personalised cookbooks with their own recipes stored within.

## 3.4. Conclusions

In conclusion this early version of the application proves that the project is indeed feasible and doable within the given timeframe if an appropriate software development methodology is followed. We now have a fully normalised database schema, as well as a prototype of the application which leaves lots of room for expansion in the coming months as each component can be tweaked and upgraded without having a knock-on effect on the rest of the system.

# 4. Testing and Evaluation

## 4.1. Introduction

In this chapter I will be detailing the testing strategy used within the project to help develop a robust piece of software that is not prone to crashing or errors. By establishing a test plan where we outline our testing strategy as well as defining the user acceptance criteria; we can develop a piece of software that not only functions well but is impervious to user and software errors.

## 4.2. Plan for Testing

We can split our testing of the prototype into two sections, unit testing and integration testing. The focus of unit tests is to validate the individual components of a piece of software to ensure that they function correctly and accurately. While integration testing focuses on how these individual units can be integrated into a larger system. Once all components have been tested individually and passed their unit tests, we can then perform integration tests to ensure that all the components function as intended when combined to form the larger system of units.

## 4.3. Plan for Evaluation

Once our software has passed all tests and is functionally sound, we can begin the evaluation of the software, how well it performs, how scalable it is and how the overall user experience is with the application.

In its current form, our system is integrated into a python environment as it was the language, I have the most experience in addition to this Python offers various methods of cleaning strings and lists which will be beneficial to us in this early stage of the application. However, this will impact the performance of the program as python takes longer to execute than other more efficient languages.

We must also evaluate the robustness of the application, since the user will be interacting with a text generation large language model, we must ensure that they are unable to generate any responses that may go against the guidelines of the application, for instance if they are inputting any extra ingredients we must make sure that they cannot make the model respond in an unsavoury way

## 4.4. Conclusions

In conclusion, our testing and evaluation plan is in place to make sure we develop robust and reliable software. We will test each individual component in the system in isolation to ensure its functionality while also integrating them into a larger system and testing this. We will also be evaluating the performance metrics of the software as well as the user experience within the finished application. By incorporating this testing strategy, I aim to not only deliver functional and fast software but also to allow for scalability and user satisfaction

# 5. Prototype Development

## 5.1. Introduction

In this section I will discuss the actual development of the CookSmart prototype.

Before development could begin, I sat down with my project mentor and discussed what aspects of the application are realistic to develop within the given time frame and what was needed for the interim submission. We decided it would be best to tackle the most difficult aspect of the project first, the object detection model and recipe generator, to see if :

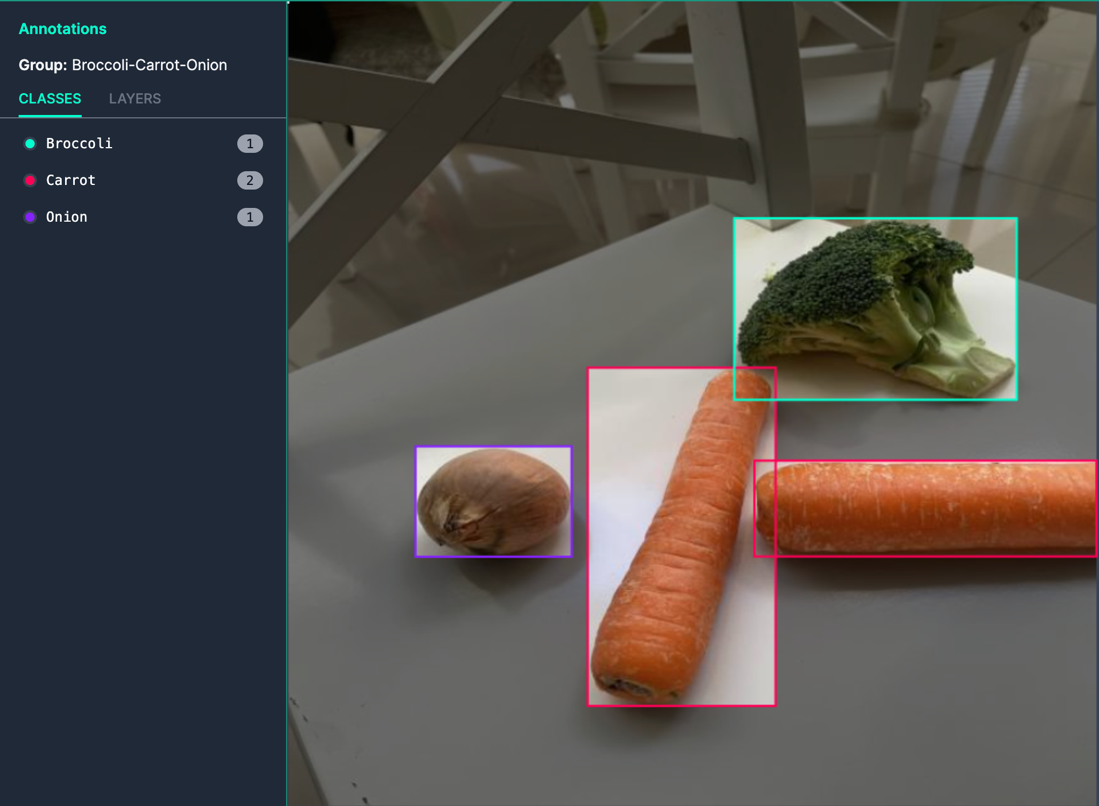
1. The project is feasible for development from a student perspective
2. The technology exists to execute this
3. The trained model can simultaneously detect multiple objects
4. These objects can be returned to the user in a list form
5. This can be passed to a large language model through an engineered prompt
6. The large language model output can be restricted to fit a database schema

Once we had decided on what could be done before the interim submission, I began researching how we can develop this and the best practices to do so.

## 5.2. Prototype Development

After researching different object detection models, the first step is to gather the data we will need to train our first version of the object detection model. This comes in the form of many photographs of the objects we want to train the model on annotated in the COCO format.

I began by purchasing a diverse selection of different vegetables for training before taking 50 photographs of each one and adding them to the project repository. Once the photos were loaded I opened them on the roboflow dashboard and began annotating them. This was a very tedious process which involved drawing a bounding box around each object and giving it a class.

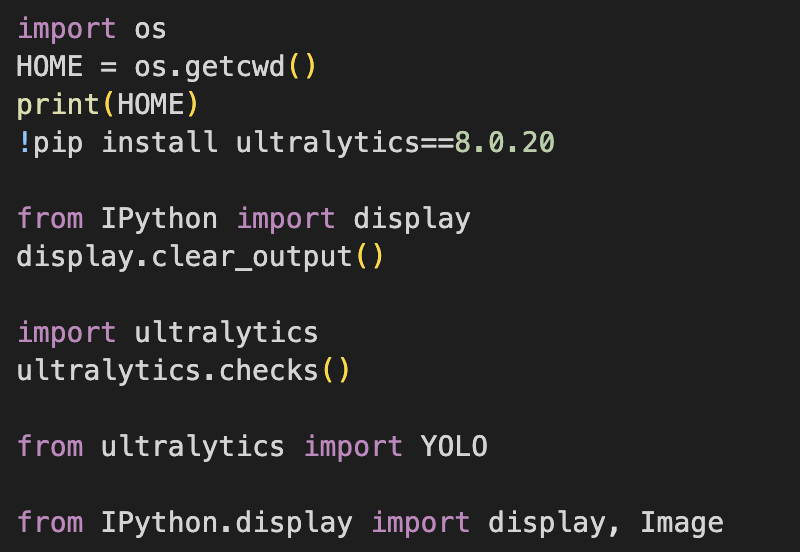


After repeating this step for the 49 remaining photos, I then flipped each of the images both horizontally and vertically to give us a total of 150 images that we can use for the training of the model. We then split these images into three sections:

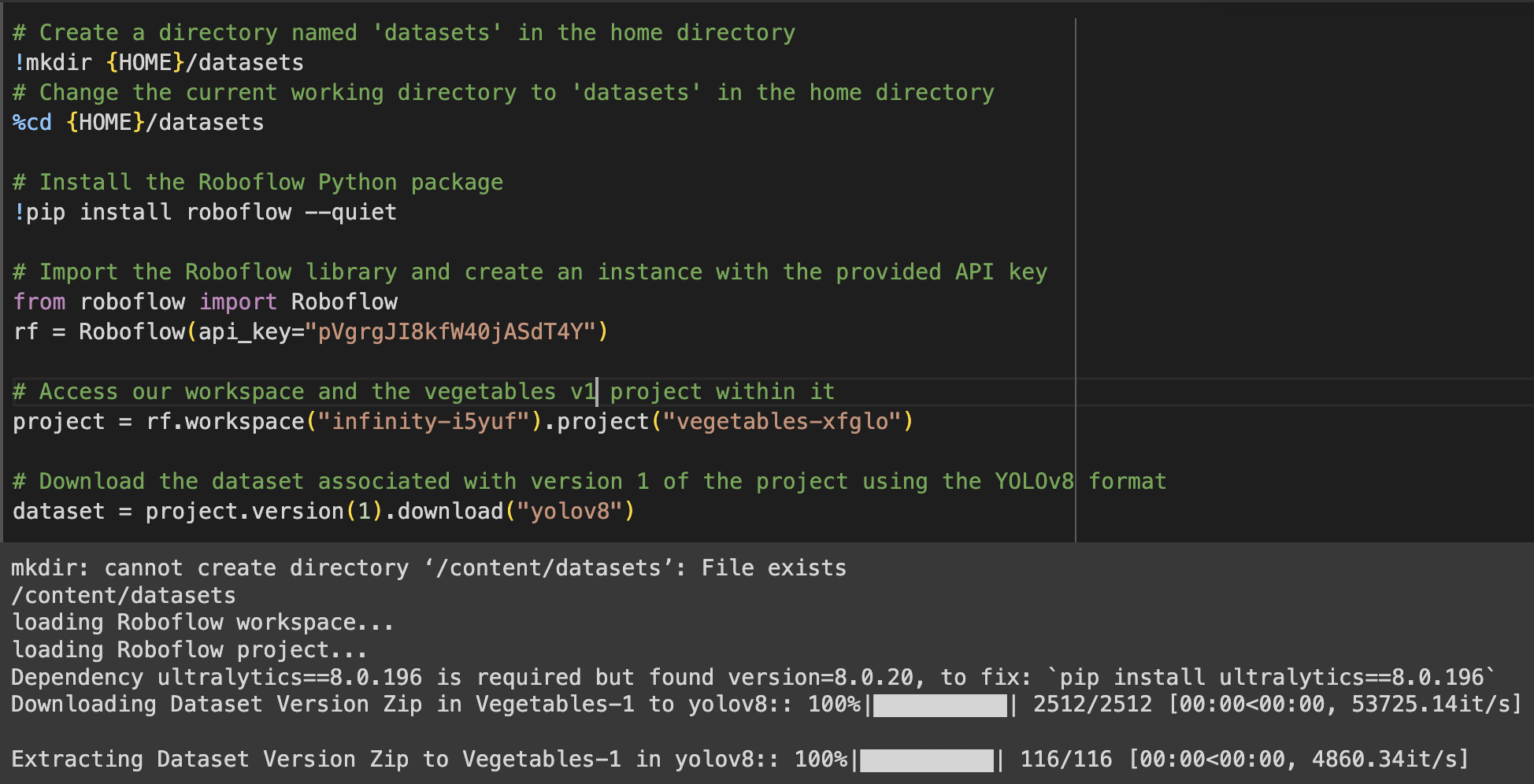
* Train – Used to train the model
* Test – Used to test the model to see how well it functions
* Validate – Control data to validate the accuracy of the model

Now that our training data has been collected and organised, we can train the model on this data to hopefully result in a YOLOv8 model that can successfully distinguish and categorise, carrots, broccoli and onions.

We begin this process by setting up our development environment and installing any dependencies required



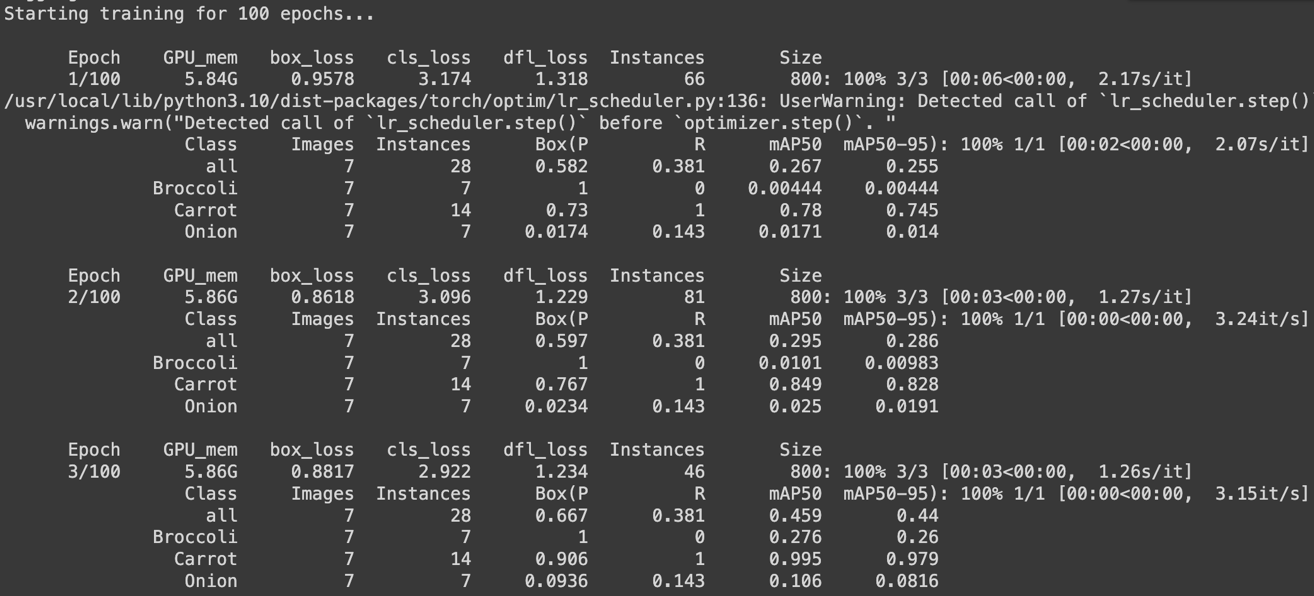
Once our dependency requirements have been satisfied, we can import our annotated dataset in COCO format into the computer memory for training of our model



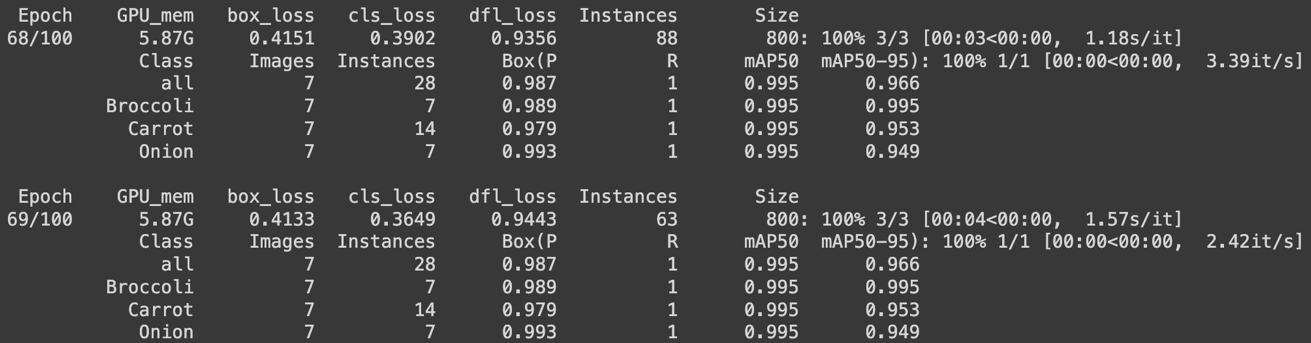
Once our dataset has been loaded into memory, we can begin the process of training the model on this dataset, we can specify how many epochs of training where the higher the number of epochs the more accurate the model will be at predicting the objects in the image.



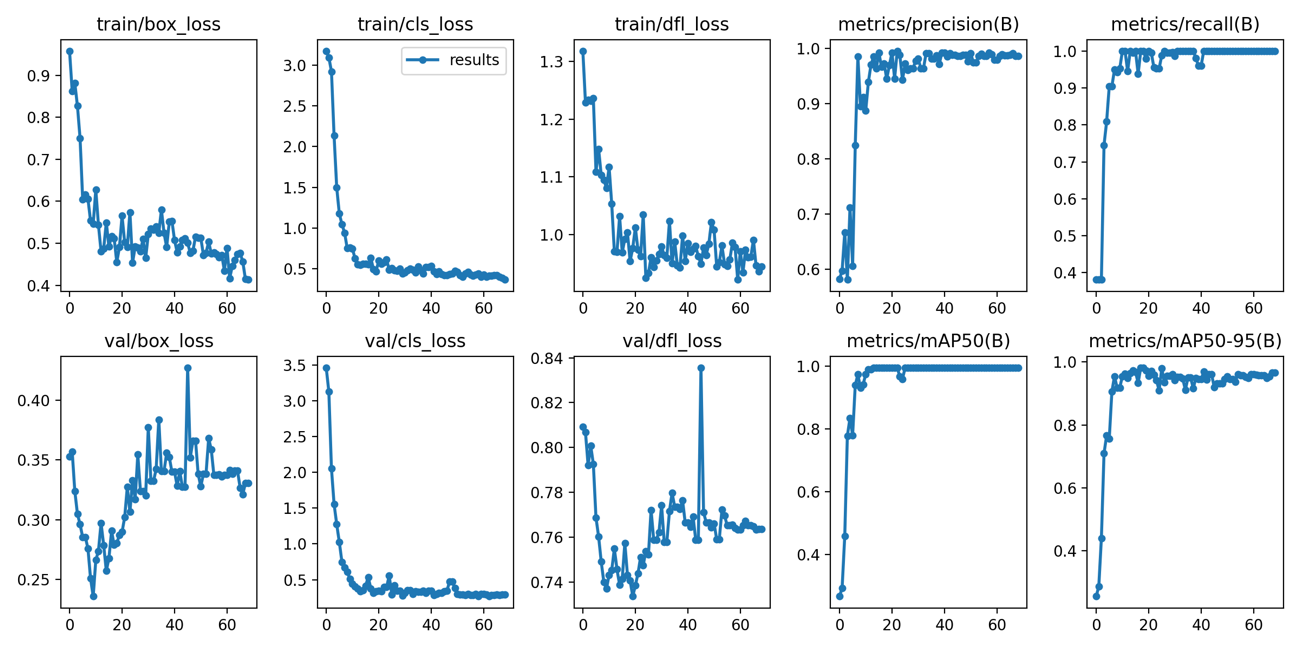
After running this code, we can see the training take place and the confidence of the model improving with each epoch. This section proved to be very time consuming and GPU intensive so for training on newer datasets, I will use Google Collab to run the code from there servers rather than my home computer to hopefully improve the training time



We can see from the output that with each additional epoch during training, the object detection model refines its understanding of the objects and fine-tunes its accuracy to predict objects in different contexts



Now that we have completed the training of our model, we can visualise the training data in graphs

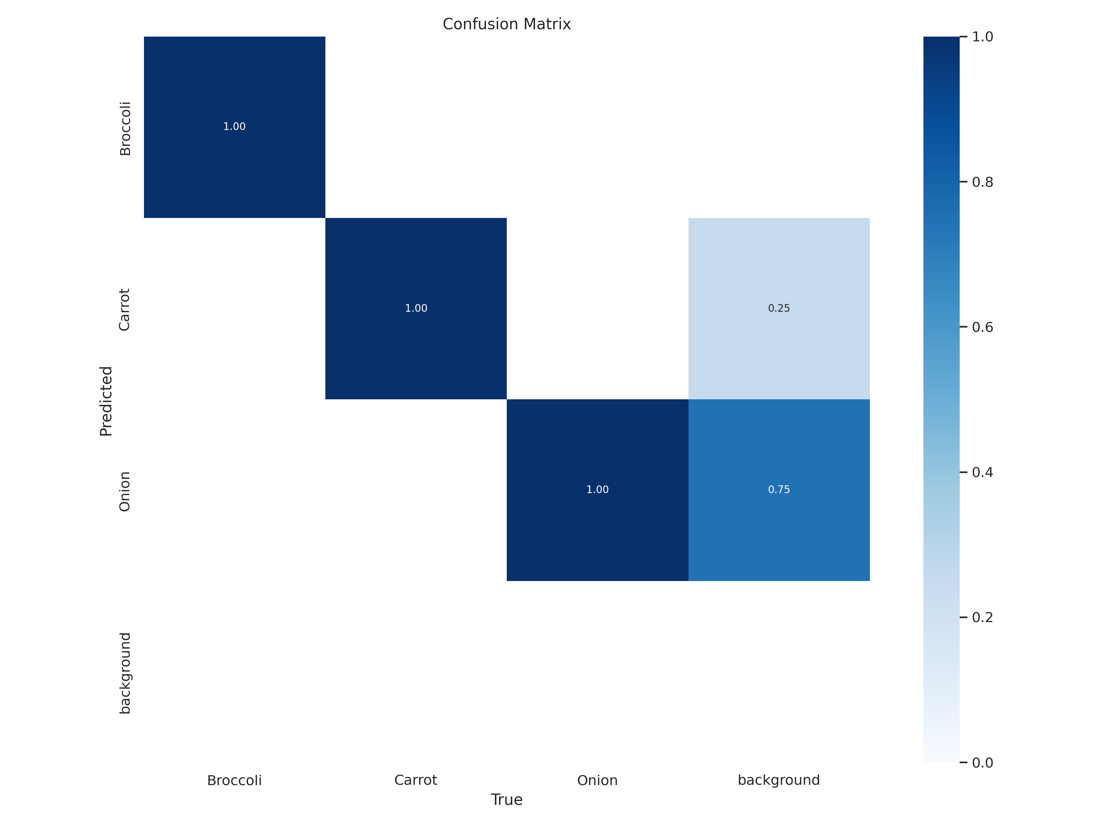


We can visualise the training in a series of graphs, to analyse how well the training went, while all the data is important, we want to focus on Box Loss, Classification Loss and Dynamic Feature Loss of this early model.

Box Loss is a crucial component in object detection models, it measures the disparity between the predicted bounding box and the control bounding box. This loss function ensures that the model learns to accurately predict objects by minimising the differences between this and the previous epoch, we can see that the box loss lowers each epoch which is in line with accurate training.

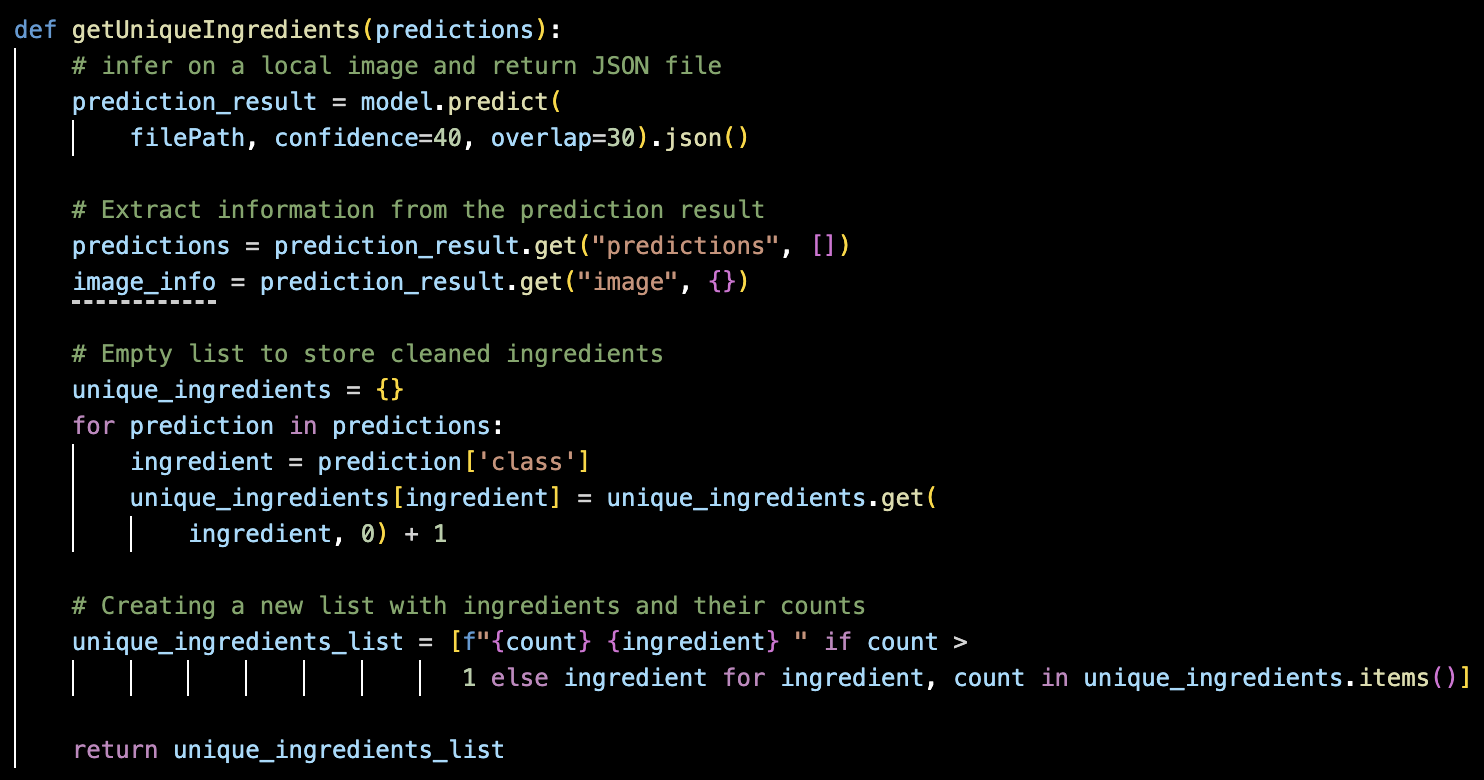
Classification Loss or cls\_loss is responsible for assessing the accuracy of the classification of different objects detected in the image. By penalising the model when it makes an incorrect classification and rewarding it when it makes accurate ones, we can guide the model to accurately categorizing the objects

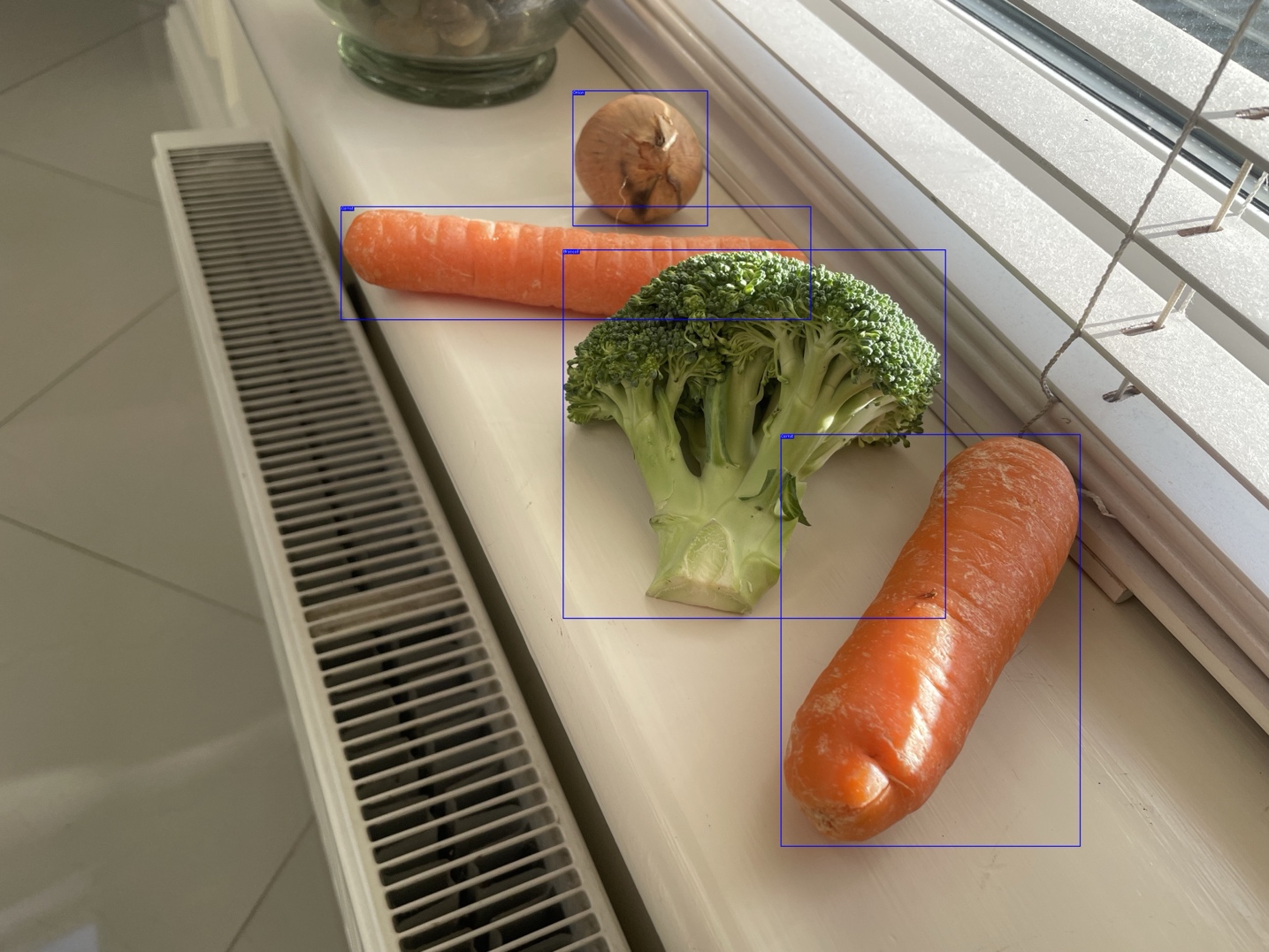
Dynamic Feature Loss or dfl\_loss refers to the adaptability of the object detection model to dynamically adjust itself to different features learned during training. This loss function serves to refine the model's ability to understand context of objects and adapt to variations in an object's appearance. By incorporating dynamic feature loss, the model can handle a diverse selection of object making it fit for real world conditions where different food items will differ in appearance

After training the data, the program creates a confusion matrix where we can visualise exactly how our model handles the prediction of different classes as well as the confidence in these predictions.

For example, this early version of the model can detect a carrot, broccoli and onion with great accuracy (according to itself) but sometimes incorrectly classifies the background areas as an onion 75% of the time and a carrot the other 25% of the time, this can be improved with greater epochs in training and a more diverse dataset as well as by experimenting with weights and biases.

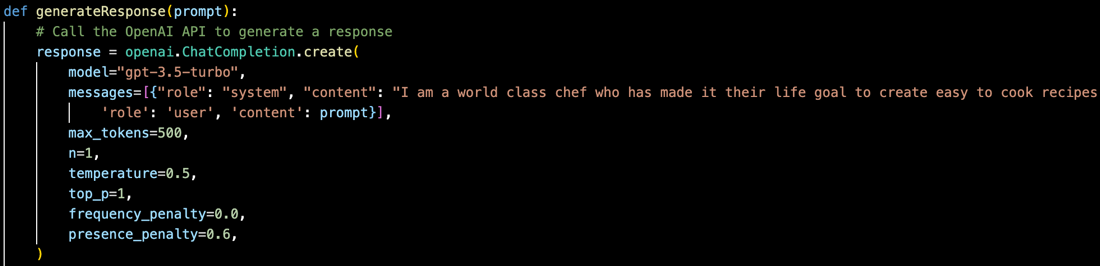
Once training was completed and the model was verified to be working we can run inference on am image through HTTP requests hosted on the roboflow API which will allow us to select an image and check if the model can detect any of the objects within the image and return us a list of what was found.





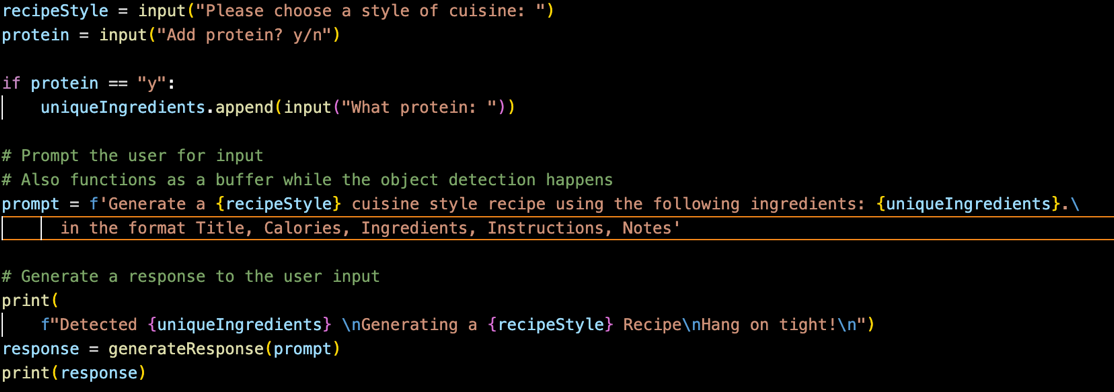
Once we had the code for our object detection model running inference on a local image and returning a list of detected objects we could then begin the development of the Recipe Generator section of our application. After much trial and error testing various text generation models, we settled on the OpenAI ChatGPT 3.5 Turbo Model for the recipe generation segment of our application.

We interact with this model differently than previous GPT models as the GPT Turbo model treats the interaction as a conversation between the AI and the user rather than simply completing texts like previous versions

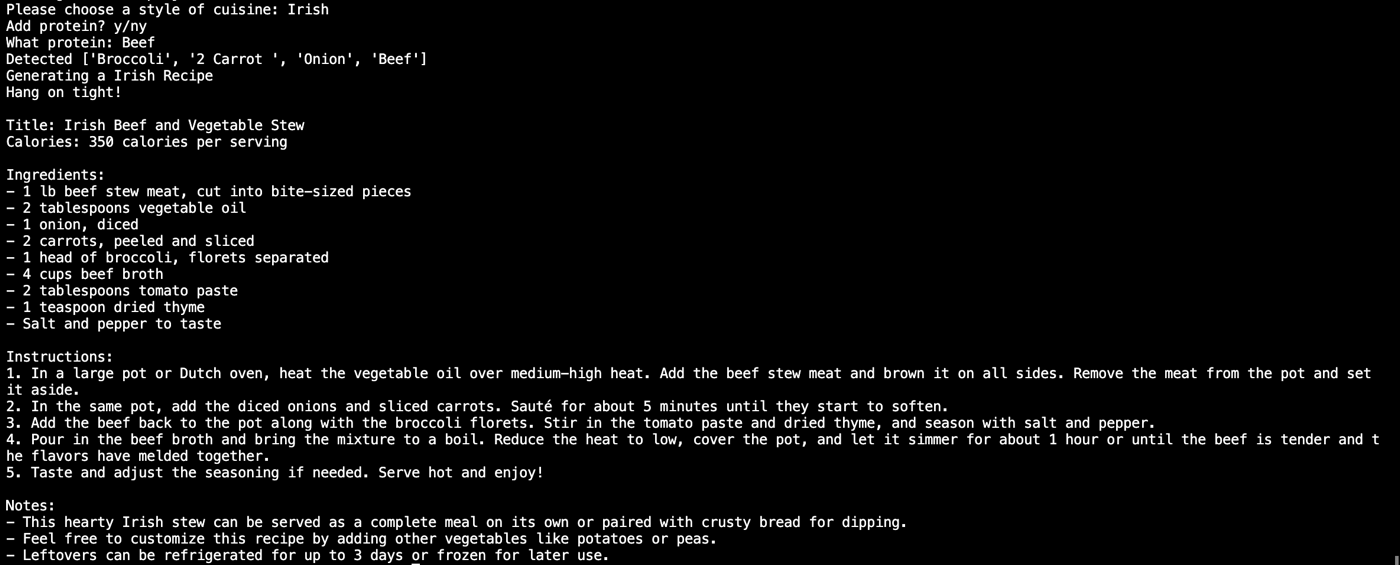


To do this we call the API and pass it a prompt to give it some context of the conversation stating that it is a chef with the sole purpose of generating recipes in our preferred format.

Before passing the ingredients to the API we must engineer a prompt with our compiled list of ingredients, the style of cuisine and finally any additional protein that the user might want to add to the recipe and setting it to work. Once the API is called it takes a couple seconds to run before returning us our recipe



Output:



## 5.4. Conclusions

In conclusion the training of the object detection model was very tedious and time consuming for how little objects it can detect. To improve upon the training time, I will train future models using Google Collab as it will allow us to use Google’s servers to train the data across multiple powerful GPU’s rather than my RTX 3060Ti.

The object detection model works well in some cases but not in others, it returns some false positives which can be mitigated by increasing the confidence parameter in the function however since the model was trained on one specific vegetable it loses confidence the more the detected vegetable differs from the one the model was trained on. To improve this, I plan to use multiple different variations of vegetables in future training datasets

The recipe generator, after much trial-and-error works extremely well with the GPT 3.5 Turbo API and is the most powerful and cost-effective method thus far. In its current state it returns a recipe in the correct format however this is not rigid enough to store this data in a persistent manner as of yet. The LLM currently mixes measurements between imperial and metric and sometimes specifies methods of preparing the ingredient in the ingredient section when it should be in the instruction section instead. This can be improved by further testing and engineering the prompt and context before making the API request.

# 6. Issues and Future Work

## 6.1. Introduction

In this chapter I will examine the application in its current state, explore known issues that have arisen during the development of the program, its limitations and its future development. This chapter not only explores the limitations of the application but how we can improve it in future iterations and lay the groundwork for a roadmap to develop a more refined, feature rich version of the application

## 6.2. Issues and Risks

Although the development of the current prototype of the application serves as a great skeleton for the final application there are some limitations in this early version of the app to be improved upon before the final application submission

**Limited Object Detection Classes**

The current Object Detection algorithm serves as a proof of concept that it is possible to detect these objects, for this reason I limited the algorithm to only broccoli, carrots and onions to ensure its feasibility before spending more time on it. In future versions I plan to expand this algorithm to detect the most popular fruits and vegetables in Ireland rather than these three

**Limited Training Time**

Due to the limited nature of the object detection algorithm, after 50 epochs of training, there was little improvement in the object detection scores, with a larger dataset I estimate that training will improve significantly along with a greater number of epochs. This should in turn reduce the number of false positive detections within the system

**Recipe Generator Format**

The current version of the large language model is good for now, but it will need to be restricted in the future, this will serve to make the generator more robust and allow us to restrict the output to the format of our database schema so that recipes generated can be returned to at a later date, the current version simply follows the format of Title, Ingredients, Instructions but does not stick to one specific unit of measurement (cups and tablespoons instead of grams) and can add the measurements and instructions in the wrong order or place (“2 Cups Onion, Diced” instead of “500g Diced Onion”) as well as this we will need to restrict it if the user is to interact directly with the generator to prevent it from generating harmful or misleading content.

## 6.3. Plans and Future Work

**Persistent Storage**

The application in its current state has no form of persistent storage, recipes generated are lost upon restart and cannot be restored once the application is closed. This does not serve the overall goal of the application so the next phase of the application will be to add a database to the backend of the application where generated recipes will be stored. I have designed a database schema for this but cannot implement it until the output of the recipe generator has been restricted to fit the schema

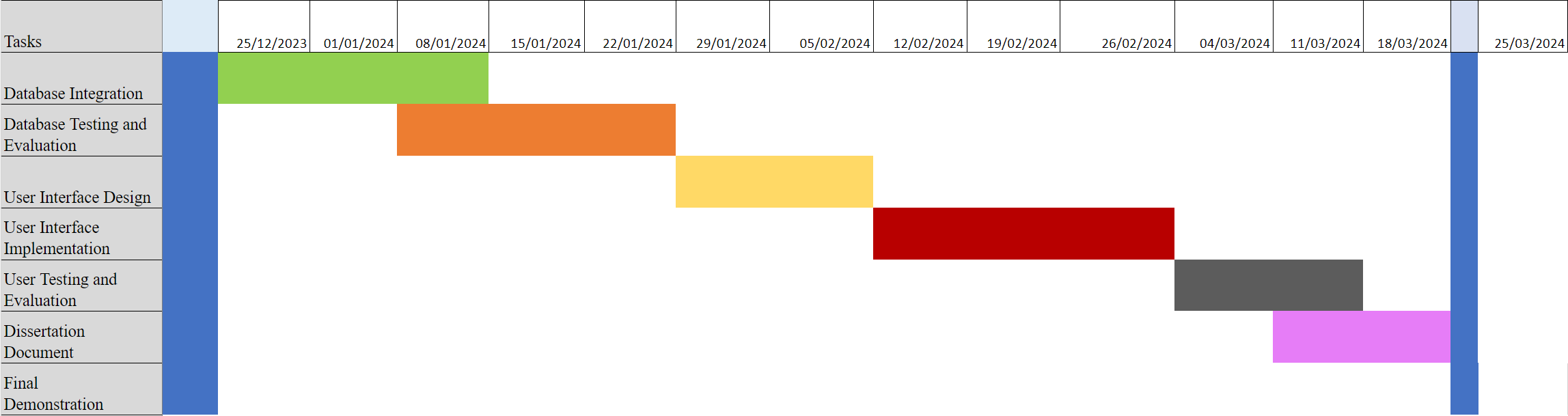
**Lack of User Interface**

The current prototype has no user interface and is exclusively interacted with through the terminal, this is obviously not in line with my plan or industry standards. I plan to add a user interface that can be interacted with in a seamless manner in either a React Native application or JavaScript webapp. This will allow users to use the camera on their device and to take the photos on their mobile device camera

**Improved Detection Model**

As the current detection model serves only to prove that the project is indeed feasible, there will be many further iterations of the model which include more training data, classes of food that are returned with a higher confidence than the current version

### 6.3.1. GANTT Chart

This GANTT chart serves as a roadmap for the future development of this project

# Bibliography

1.

Taher AKQA. The effects of takeaway (fast) food consumption on UK adolescent’s diet quality and BMI [Internet] [phd]. University of Leeds; 2020 [cited 2023 Dec 7]. Available from: <https://etheses.whiterose.ac.uk/27881/>

2.

Lydon J. The Current State of Cooking in Ireland: The Relationship between Cooking Skills and Food Choice.

3.

Flanagan A, Priyadarshini A. A study of consumer behaviour towards food-waste in Ireland: Attitudes, quantities and global warming potentials. Journal of Environmental Management. 2021 Apr 15;284:112046.

4.

Agency EP. Food Waste Statistics [Internet]. [cited 2023 Dec 3]. Available from: <https://www.epa.ie/our-services/monitoring--assessment/waste/national-waste-statistics/food/>

5.

Is Cooking Still a Part of Our Eating Practices? Analysing the Decline of a Practice with Time-Use Surveys [Internet]. [cited 2023 Dec 7]. Available from: <https://journals.sagepub.com/doi/epub/10.1177/1749975518791431>

6.

Graham-Rowe E, Jessop DC, Sparks P. Identifying motivations and barriers to minimising household food waste. Resources, Conservation and Recycling. 2014 Mar 1;84:15–23.

7.

COMPUTATIONAL MODEL OF COCONUT MATURITY DETECTION USING YOLO AND ROBOFLOW | Redshine Archive [Internet]. [cited 2023 Dec 7]. Available from: <https://chapters.redshine.in/index.php/redshine/article/view/130>

8.

Zhao ZQ, Zheng P, Xu ST, Wu X. Object Detection With Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems. 2019 Nov;30(11):3212–32.

9.

Ni J, Ábrego GH, Constant N, Ma J, Hall KB, Cer D, et al. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models [Internet]. arXiv; 2021 [cited 2023 Dec 7]. Available from: <http://arxiv.org/abs/2108.08877>

10.

Wu T, He S, Liu J, Sun S, Liu K, Han QL, et al. A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development. IEEE/CAA Journal of Automatica Sinica. 2023 May;10(5):1122–36.