|  |  |
| --- | --- |
|  | Project Plan |
|  |  |
|  | Natalia Palej |

A00279259  
Software Design with Artificial Intelligence for Cloud Computing  
Year 4

Contents

[Introduction 2](#_Toc182942147)

[Purpose and Scope 2](#_Toc182942148)

[Purpose 2](#_Toc182942149)

[Scope 2](#_Toc182942150)

[Goals and Objectives 2](#_Toc182942151)

[Goals 2](#_Toc182942152)

[Objectives 2](#_Toc182942153)

[Success Criteria 3](#_Toc182942154)

[Work Breakdown Structure (WBS) 3](#_Toc182942155)

[High Level Breakdown 3](#_Toc182942156)

[What’s In and Out of Scope 4](#_Toc182942157)

[Assumptions 4](#_Toc182942158)

[Milestones 5](#_Toc182942159)

[Project Schedule 6](#_Toc182942160)

[Estimates 6](#_Toc182942161)

[Approach 7](#_Toc182942162)

[Resources and Budget 8](#_Toc182942163)

[Costs 8](#_Toc182942164)

[Resources 8](#_Toc182942165)

[Risks and Mitigation 8](#_Toc182942166)

[Acceptance Criteria 8](#_Toc182942167)

[References 9](#_Toc182942168)

# Introduction

The project, "Recipe Vision", is designed to reduce food waste by using Machine Learning and Artificial Intelligence to identify food items and generate recipe ideas. The app uses YOLO for object detection and Groq API to suggest recipes, including at least one healthy option. The project is built with Python, PyTorch and Flask, with focus on creating a simple prototype where users can upload images, detect food items and view recipe suggestions. The goal is to prove the concept while keeping it easy to use.

# Purpose and Scope

## Purpose

The purpose of this project is to develop a user-friendly application that identifies raw food products and its quantities using image recognition. Based on recognized items and their quantities, the application provides recipe suggestions (where at least one is healthy) which will reduce food waste and household expenses. The application joins object detection with AI recipe generation.

## Scope

The application will recognize common raw products from categories such as vegetables, fruits, meat, and dairy. It will use the Groq API to generate recipes, ensuring at least one healthy option. A basic front-end will allow users to upload images and view the results. After processing, the application will display a pop-up showing the detected items, which users can manually confirm or adjust by adding any missing products in a text field. The backend will filter recognized products based on a confidence threshold of 80% or higher and integrate with an external AI API for recipe generation.

# Goals and Objectives

## Goals

* Develop an AI-powered application using YOLO for object detection
* Integrate Groq AI API for recipe generation
* Create flask-based interface for demonstration

## Objectives

* Collect and annotate raw food datasets with focus on common products from categories like vegetables, fruit, meat, diary
* Custom train and fine-tune YOLO model to get accurate object detection
* Build Flask-based backend to handle image processing and API “post” request
* Develop simple frontend to allow users to upload images, view detected items and add missing ones if necessary
* Ensure seamless integration between the image recognition model, Groq API and frontend components

# Success Criteria

* Fully annotated and validated raw food dataset
* Custom trained YOLO model with at least 80% confidence threshold
* Functional Flask backend capable of filtering recognized items based on confidence levels
* Successful interaction with Groq AI API for recipe generation
* Basic but functional frontend for user interactions and result display
* Comprehensive documentation, including final year project thesis with detailed explanations of the system’s functionality and outcomes

# Work Breakdown Structure (WBS)

## High Level Breakdown

|  |  |
| --- | --- |
| **Phase** | **Description** |
| Planning | Gather and refine requirements |
| Data Preparation | Find and combine appropriate food datasets  Annotate images (if not already done so)  Validate final, custom dataset |
| Model Development | Train YOLO custom model on annotated dataset  Evaluate and fine-tune the model |
| System Integration | Develop backend for object detection and recipe generation |
| Frontend Development | Build basic Flask frontend |
| Testing | End-to-end system testing |
| Refinement | Debugging and performance optimization |

# What’s In and Out of Scope

|  |  |
| --- | --- |
| **In Scope** | **Out of Scope** |
| Annotate sufficient portion of dataset to train and validate custom YOLO model | Detecting packaged or processed food like cans or pre-made meals |
| Recognizing common raw food categories   * Vegetables (eg. carrots, potatoes, broccoli) * Frutis (eg. apples, bananas, oranges) * Diary (eg. milk, cheese, cream cheese) * Meats, classified into general types (eg. “chicken” instead of specific cuts like chicken fillet or diced chicken) | Detecting all possible food products |
| Image detection confidence level of at least 80% using YOLO custom model | Classifying specific food varieties (eg. distinguishing between chicken fillet, dices chicken or whole chicken – all will be grouped as “chicken”) |
| Generating 3 recipe suggestions via Groq API where at least one is a healthy option | Full-scale deployment or optimization for mobile platforms (focus is on desktop or local prototype functionality). |
| Functional prototype to prove the concept   * Backend for processing images and integrating API * Frontend to upload images, view results, manually add missing items | Full-scale deployment or optimization for mobile platforms (focus is on desktop or local prototype functionality). |

# Assumptions

1. Open-source datasets are accessible and meet quality standards for model training.
2. MakeSense online annotation tool will function effectively throughout the dataset preparation phase.
3. College PC will have necessary resources (GPU, Memory) to train custom YOLO model without delays or limitations.
4. Groq AI API remain available and reliable for recipe generation during the project timeline and final presentation.
5. ChatGPT will be accessible as a backup for generating recipes during the final presentation.
6. Time allocation will sufficiently balance project with other academic responsibilities including exams and other assignments.
7. The project will focus only on common food products.
8. The custom trained model will achieve confidence threshold of at least 80% for detecting food items accurately.

# Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone** | **End Date** | **Description** | **Output** |
| M1: Project Proposal | 24/09/2024 | The project proposal is finalized, including goals, objectives, methodologies and expected outcomes. | Formal project proposal document submitted for review and approval. |
| M2: Feasibility Study | 30/09/2024 | Feasibility study is conducted to ensure that the proposed technologies and methodologies are viable. | Feasibility study document showing evidence of the project’s viability. |
| M3: Requirements Specification Finalized | 07/10/2024 | Functional and non-functional requirements are detailed in the Software Requirements Specification (SRS). | Finalized SRS document outlining system functionality and constraints. |
| M4: Dataset Preparation | 06/01/2025 | Raw food images are collected, annotated and validated for custom YOLO model training. | A fully annotated and validated dataset ready for model training. |
| M5: Model Training and Evaluation | 03/02/2025 | YOLO model is trained using the annotated dataset and evaluated against performance metrics, eg. mAP, precision, recall. | Trained custom YOLO model meeting at least 80% confidence threshold. |
| M6: Backend Development | 17/02/2025 | Backend is developed to process images, filter results based on confidence thresholds and integrated with Groq API for recipe generation. | A functional backend capable of image processing and recipe generation. |
| M7: Frontend Development | 17/03/2025 | User -friendly frontend is developed for image upload, manual adjustments of detected items and display of recipe suggestions. | A fully functional frontend integrated with the backend. |
| M8: Integration Testing | 28/03/2025 | End-to-end testing ensured smooth communication between the backend, frontend, and Groq API. Debugging and optimization are performed. | A fully tested and operational system ready for final evaluation. |
| M9: Thesis Writing | 11/04/2025 | The project thesis documenting the background, methodology, implementation, results and conclusions is completed and proofread. | A comprehensive thesis ready for submission. |
| M 0: Poster Design | 18/04/2025 | Professional project poster summarizing the key aspects of the project is created for demo. | A finalized poster ready for display. |

# Project Schedule

A graph on a white sheet

Description automatically generated

Figure Gantt Chart

Gantt Chart in Excel Format: [[CLICK HERE](https://d.docs.live.net/cad1acabf18afac6/Desktop/Software%20Design%20with%20AI%20for%20Cloud/Year_4/management-and-organizational-behaviour/project-plan/Online%20Gantt%2020241119.csv)]

A timeline of milestones

Description automatically generated

Figure Project Estimate Timeframe

# Approach

This project uses Python as the main programming language and PyTorch with YOLO for object detection. A custom YOLO model will be trained using datasets collected from open-source platforms, like the FruitVeg-81 Dataset (Paletta, Georg Waltner, Michael Schwarz, Stefan Ladstätter, Anna Weber, Patrick Luley, Meinrad Lindschinger, Irene Schmid, Walter Scheitz, & Horst Bischof, Lucas, 2017) [[1]](#_References), Roboflow [[2]](#_References) or Kaggle [[3]](#_References). The images will be annotated using MakeSense [[4]](#_References), an online tool for annotating data. Since my local machine cannot handle the training process due to hardware limitations, the college PC will be used for model training. The outputs will be validated and the model will be fine-tuned to improve accuracy. The backend will be developed using Flask to process images, handle detections and integrate with the Groq [[5]](#_References) API for recipe generation. A simple frontend will allow users to upload images, view the detected items and manually add any missing products.

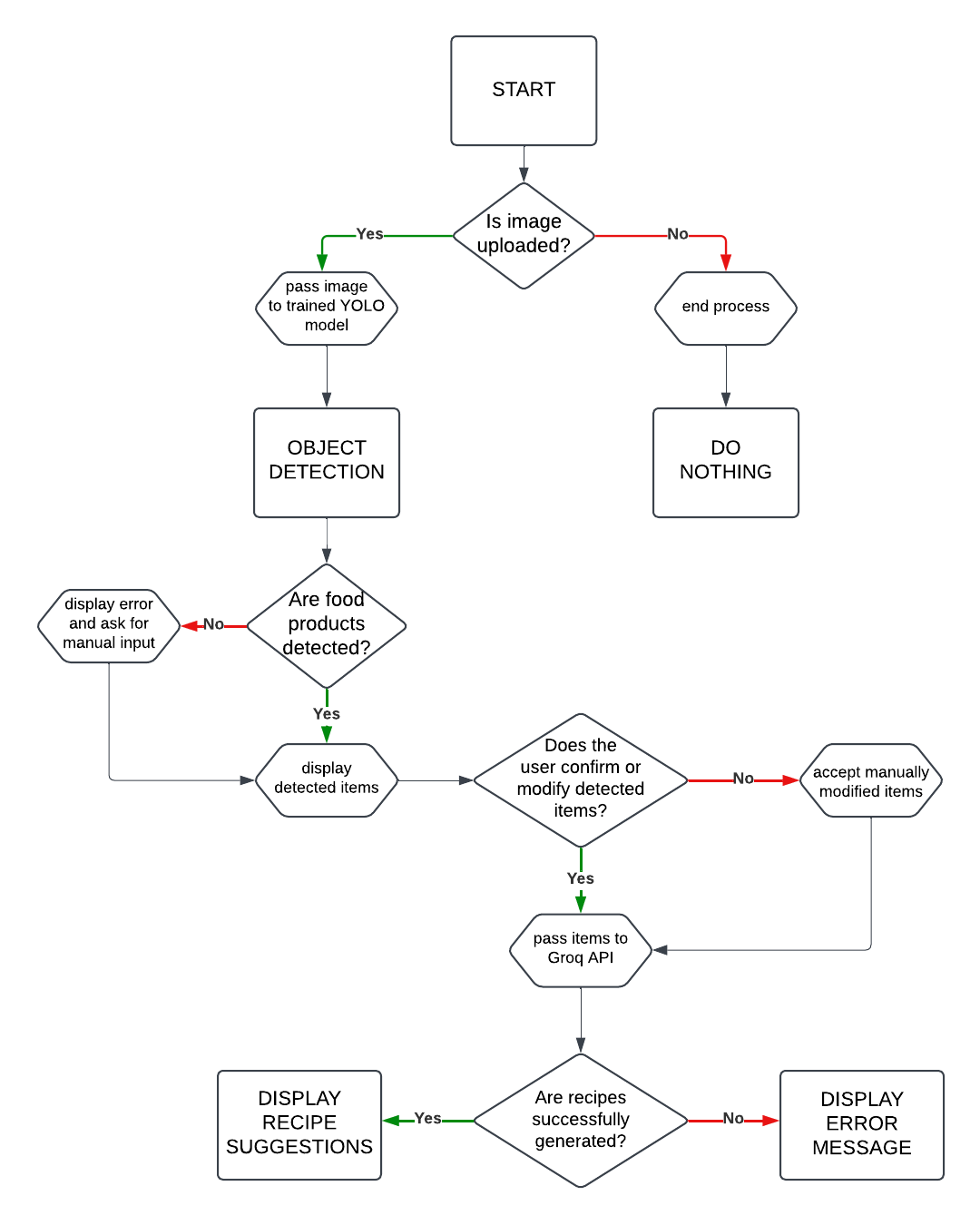


Figure 3 Flow Chart

# Resources and Budget

## Costs

**Estimated Budget:** €0 (utilizing open-source tools and college-provided resources)

The project has minimal costs due to the use of open-source tools and datasets. Access to the FruitVeg-81 dataset eliminates the need to manually take photos of various fruits and vegetables, saving time and effort. The Groq AI API is free for recipe generation, and ChatGPT will be used as a backup only during the final presentation, reducing the cost of tokens required.

## Resources

**People:** 1 developer

**Tools:** Open-source software (YOLO, PyTorch, Python, Flask, Groq AI), college hardware

College PC will be used for model training and other resource-intensive tasks. Open-source datasets of raw food products are used and annotated with MakeSense. Flask will be used for backend and frontend development.

# Risks and Mitigation

|  |  |  |
| --- | --- | --- |
| **Description** | **Risk** | **Mitigation** |
| Image Annotation Taking Too Long | Manually annotating thousands of images could delay the project timeline | Focus on annotating key food products first, ensuring that common items are included |
| Not Enough High-Quality Images | Lack of quality images could impact the model’s performance | Look for additional open-source datasets or use data augmentation to create more diverse images |
| Overwhelmed with College Projects and Exams | Balancing this project with other coursework may cause delays | Stick to the Gantt chart, focus on priority tasks and allow extra time for important phases |
| Personal Exhaustion or Stress | Long working hours may lead to burnout | Take regular breaks, maintain a balanced schedule, seek help from supervisors when needed |
| Unexpected Issues or Delays | Unexpected problems like technical failures or personal emergencies may impact progress | Regularly back up data and maintain version control for code |

# Acceptance Criteria

1. Fully annotated and validated dataset, ready for model training.
2. Custom YOLO model trained and capable of detecting food items with at least 80% confidence.
3. Backend that processes images, filters result and integrates with the Groq API for recipe generation.
4. Simple frontend where users can upload images, view results and add missing items.
5. Comprehensive documentation including final year thesis, system architecture diagrams, project outcomes.
6. Professional project poster summarizing the problem, methods, results and conclusions.

# References

1. G. Waltner, M. Schwarz, S. Ladstätter, A. Weber, P. Luley, M. Lindschinger, I. Schmid, W. Scheitz, H. Bischof, and L. Paletta, “Personalized Dietary Self-Management using Mobile Vision-based Assistance,” in *Proc. International Workshop on Multimedia Assisted Dietary Management (MADIMA, in conjunction with ICIAP)*, 2017. [Online]. Available: <https://www.tugraz.at/institute/icg/research/team-bischof/learning-recognition-surveillance/downloads/fruitveg>. [Accessed: Oct. 14, 2024].
2. Roboflow, "Open Source Computer Vision Community," 2024. [Online]. Available: <https://universe.roboflow.com/> [Accessed: Nov. 7, 2024].
3. Kaggle, "The world’s largest data science community with powerful tools and datasets," 2024. [Online]. Available: <https://www.kaggle.com/> [Accessed: Nov. 5, 2024]
4. MakeSense, "Free-to-use tool for annotating images," 2024. [Online]. Available: <https://www.makesense.ai/> [Accessed: Oct-Dec, 2024].
5. Groq AI, "Fast AI Inference," 2024. [Online]. Available: <https://www.groq.com/> [Accessed: Oct, 2024].