Arab Academy for Science, Technology, and Maritime Transport
College of Computing and Information Technology
Smart Village

A System based on AI and real-time object detection and searching

A Thesis submitted in partial fulfillment of the requirements of
Bachelor Degree in Computer Science

By

Alaa Bassem	19105659
Ahmed El-sayed	19107947
Ahmed Hassan	19107898
Mohammed Salah	18107286
Nour Ahmed	19107473
Salma Ayman	19107447

College of Computing and Information Technology, AASTMT

Supervised By

Assoc. Prof. Mohammed Mostafa Fouad

Assoc. Prof. Dr. Essam Elfakharany

July 2023

Abstract

Object detection plays a crucial role in various domains, including security, surveillance, transportation, and social media. However, traditional object detection methods often suffer from limitations such as slow processing speeds and challenges in accurately detecting objects in complex or cluttered scenes. This thesis proposes a real-time object detection and searching model using **YOLO** (You Only Look Once), a popular object detection technique known for its high accuracy and processing speed.

The primary motivation behind this proposed model is to enhance the speed and accuracy of object detection and classification in real-world scenarios. By leveraging the capabilities of YOLO, the model aims to provide an efficient and real-time identification and localization of objects of interest. Unlike traditional methods that require multiple stages for detection, YOLO processes images in a single pass, enabling faster and more efficient processing.

This thesis presents a real-time object detection and searching model using YOLO to overcome the limitations of traditional object detection methods. By leveraging YOLO's high accuracy and processing speed, the model aims to provide fast, reliable, and accurate object detection and classification in real-world scenarios. The model's performance will be evaluated through extensive experiments and comparisons with existing state-of-the-art methods.

Acknowledgment

بسم الله الرحمن الرحيم

وَقُلِ ٱعۡمَلُواْ فَسَيَرَى ٱللَّهُ عَمَلَكُمۡ وَرَسُولُهُۥ وَٱلۡمُؤۡمِنُونَۖ وَسَتُرَدُّونَ إِلَىٰ عَٰلِمِ ٱلۡغَيۡبِ وَٱلشَّهَٰدَةِ فَيُنَبِّئُكُم بِمَا كُنتُمۡ تَعۡمَلُونَ

صدق اللَّهُ العظيم

The completion of this project would not have been possible without the assistance and Support from the following people that we would like to express them our sincere appreciation:

- Our Families, who during the entire study were all incredibly understanding. Even though we had a very difficult time finding the time to finish this project, they gave us the encouragement and support we needed.

- Our Supervisors, Assoc. Prof. Dr. Mohammed Mostafa Fouad, Assoc. Prof. Dr. Essam Elafkarany They fully grasped our needs, concerns, and the challenges encountered while carrying out this project, and they benefited us by sharing their expertise with us and by constantly being patient with us in their guidance.

Contents

List of Figures

List of Tables

Abbreviations

AI	**A**rtificial **I**ntelligence
ML	**M**achine **L**earning
NN	**N**eural **N**etwork
API	**A**pplication **P**rogramming **I**nterface
UML	**U**nified **M**odeling **L**anguage
YOLO	**Y**ou **O**nly **L**ook **O**nce
COCO	**C**ommon **O**bject in **C**ontext

Chapter 1

Introduction

The identification of objects in images or videos is a task that humans excel at, thanks to their advanced and accurate neural visual system. Computers, with their ability to mimic the human visual system, can achieve similar results by training on large datasets, leveraging faster GPUs, and employing sophisticated algorithms. Object detection, which mimics the functions of the human visual system, is a thriving field with numerous applications.

This thesis utilizes YOLO (You Only Look Once) for object detection and classification. YOLO is one of the most popular model architectures and object detection techniques due to its utilization of a highly effective neural network architecture, resulting in high accuracy and efficient processing speeds.

The primary objective of this thesis is to present a real-time object detection and classification system using YOLO. By leveraging the capabilities of YOLO, the model aims to accurately identify and classify objects in real-world scenarios. The use of passive voice throughout the thesis emphasizes the role of the proposed model in imitating the human visual system.

1.1 Motivation

The motivation behind the development of a real-time detection and searching system using YOLO (You Only Look Once) is to enhance the speed and accuracy of object detection and classification in real-world scenarios. Traditional object detection algorithms are often slow and cumbersome, posing challenges for real-time applications. The utilization of YOLO, known for its fast and efficient image processing capabilities, enables real-time detection and searching of objects in images and videos.

By incorporating YOLO into our system, objects of interest can be quickly identified and located. The accurate classification abilities of YOLO contribute to the reduction of false alarms and overall improvement in system performance. The passive voice usage throughout the thesis highlights the aim of providing a fast, reliable, and accurate tool for object detection and classification in real world scenarios, with applications ranging from security and surveillance to search and rescue operations.

1.2 Problem Statement

Traditional object detection methods can be slow and require significant computational resources, making them impractical for many real-time applications. Additionally, traditional object detection methods may struggle to accurately detect and classify objects in complex or cluttered scenes. This can lead to false alarms, missed detections, or misclassifications.

Therefore, the need to develop a system that can accurately and efficiently detect and classify objects in real-time, even in complex or cluttered scenes. This system must be able to process large amounts of data quickly and accurately, and must be able to adapt to changing conditions or environments.

By addressing this problem statement, we can develop a system that has numerous applications in fields such as security, surveillance, and search and rescue, as well as in everyday scenarios such as traffic management and object recognition in social media.

1.3 Objectives

The objective of this thesis is to develop a real-time detection and searching system using YOLO (You Only Look Once) with the aim of improving the efficiency, accuracy, and speed of object detection in various applications. Object detection is a critical task in fields such as security, surveillance, transportation, and medicine. However, traditional object detection methods often suffer from slow processing speeds and high computational requirements, limiting their usability in real-time scenarios.

By leveraging the capabilities of YOLO, this thesis aims to address these limitations and provide an efficient and accurate solution for real-time object detection and classification. The primary objective is to develop a system that can detect and classify objects in real-time, enabling faster decision-making in emergency response situations, traffic management, and security and surveillance operations.

The proposed system seeks to achieve the following objectives:

1. Enhance the speed and efficiency of object detection and classification.
2. Improve the accuracy of object localization and identification.
3. Enable real-time processing for quick decision-making in various applications.

By achieving these objectives, the proposed system using YOLO aims to provide a valuable tool for object detection and searching, offering benefits across a wide range of domains, including security, surveillance, social media, and e-commerce. The system's performance will be evaluated through experiments and comparisons with existing methods to validate its effectiveness in real-world scenarios.

1.4 Problem Complexity

The development of a real-time object detection system using YOLO (You Only Look Once) involves tackling various complexities and challenges. These include technical aspects such as optimizing computational resources, handling complex scenes, and striking a balance between processing speed and detection accuracy. Another significant challenge is the availability of suitable datasets, particularly when targeting specific objects such as handbags.

Obtaining specialized datasets tailored to the system's purpose can be complex and time-consuming, as existing datasets often focus on general object categories rather than specific objects of interest. This dataset complexity requires innovative approaches, such as data augmentation techniques or manual annotation, to curate a dataset that aligns with the system's specific requirements.

In addition to technical challenges and dataset complexity, the interdisciplinary nature of the problem introduces further intricacies. Integrating knowledge from various fields, including computer vision, machine learning, and hardware design, is necessary to create an effective real-time object detection system. These interdisciplinary requirements call for a comprehensive understanding of different domains and methodologies, demanding collaboration and expertise from multiple disciplines.

Mitigating these complexities requires continuous adaptation to evolving requirements and advancements in technology. Rigorous experimentation, evaluation, and feedback from real-world scenarios are essential to ensure the system's robustness, reliability, and adaptability. By addressing these complexities, the aim is to develop a high-performing real-time object detection system that can accurately and efficiently detect objects in various real-world applications.

1.5 Constraints

There could be several constraints that need to be considered when developing a real-time detection and searching system using YOLO. Some of these constraints include:

Computational resources: The real-time processing of images and videos can be computationally intensive, and the available hardware resources may limit the system's ability to perform object detection and classification accurately and efficiently.

Sensor quality and availability: The accuracy of object detection depends on the quality and availability of sensors, such as cameras or LiDAR systems, that are used to capture the data. If the sensor quality is poor or the sensor is not available, the accuracy of the system may be affected.

Environmental conditions: The environmental conditions, such as lighting or weather, can affect the accuracy and reliability of object detection. For example, low-light conditions or heavy rain may make it difficult to detect objects accurately.

Dataset availability and quality: The accuracy of object detection depends on the quality and diversity of the dataset used to train the YOLO model. If the dataset is limited or of poor quality, the performance of the system may be affected.

Regulatory and privacy constraints: In some applications, there may be regulatory or privacy constraints that limit the use of certain types of sensors or data processing techniques.

Cost: The cost of hardware, software, and maintenance can be a significant constraint when developing a real-time detection and searching system using YOLO. The system must be designed to be cost-effective and scalable to meet the needs of the application.

1.6 Standards

The project uses Unified Modeling Language (UML) standards. UML is a general-purpose development modeling language, which aims to provide a standard way to visualize the design of a software. UML models used are Use case diagrams, Activity diagrams, and Sequence diagrams.

1.7 Feasibility Study & Business Canvas

1.7.1 Business Model Canvas

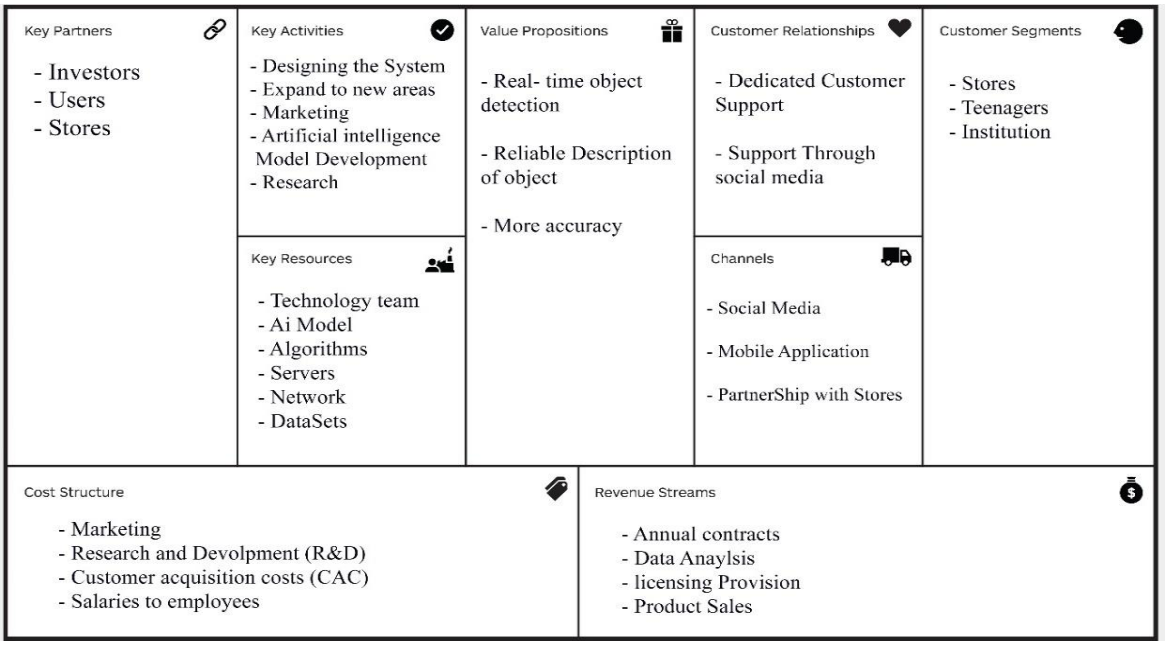


Figure 1.1 Business model canvas

1.7.2 Resource Overview

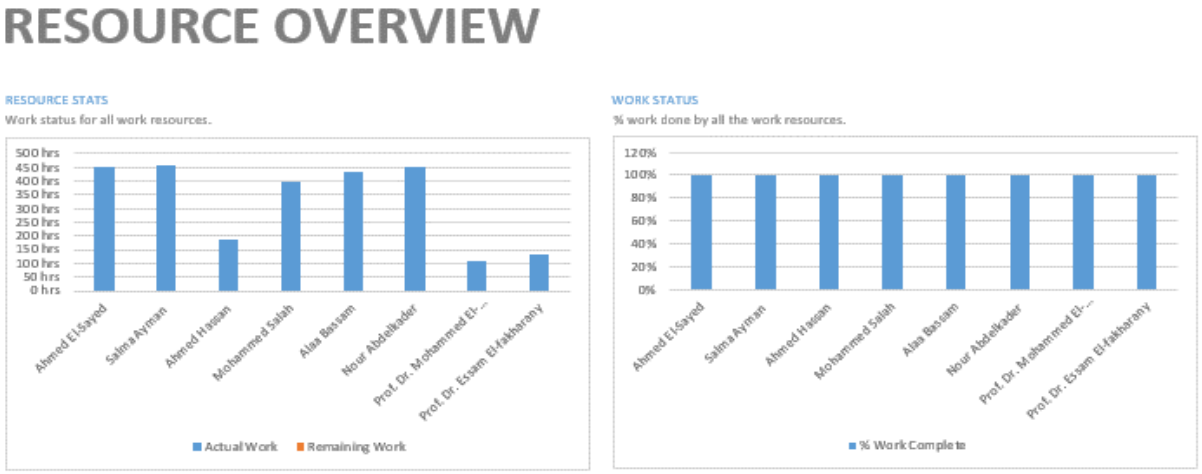


Figure 1.2 Resource Overview

The work status of all people (work resources) who are working on your project, so you'll know how much work is complete and what's left to be done, as we completed the project, we see that all work is done

1.7.3 Cash flow

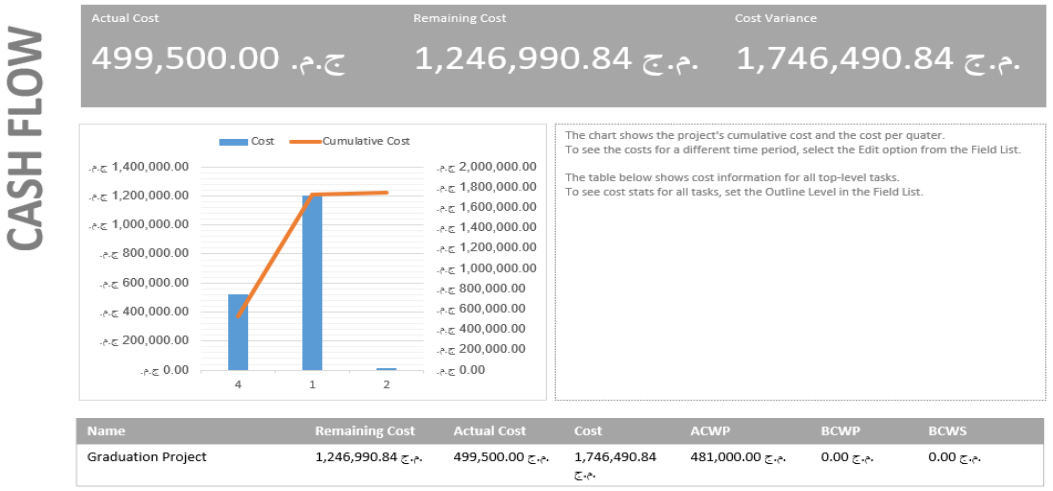


Figure 1.3 cash flow

Cash flow This shows the project's cumulative cost and the cost. As the above figure shows all cost information for all to level tasks.

1.7.4 Task Cost Overview

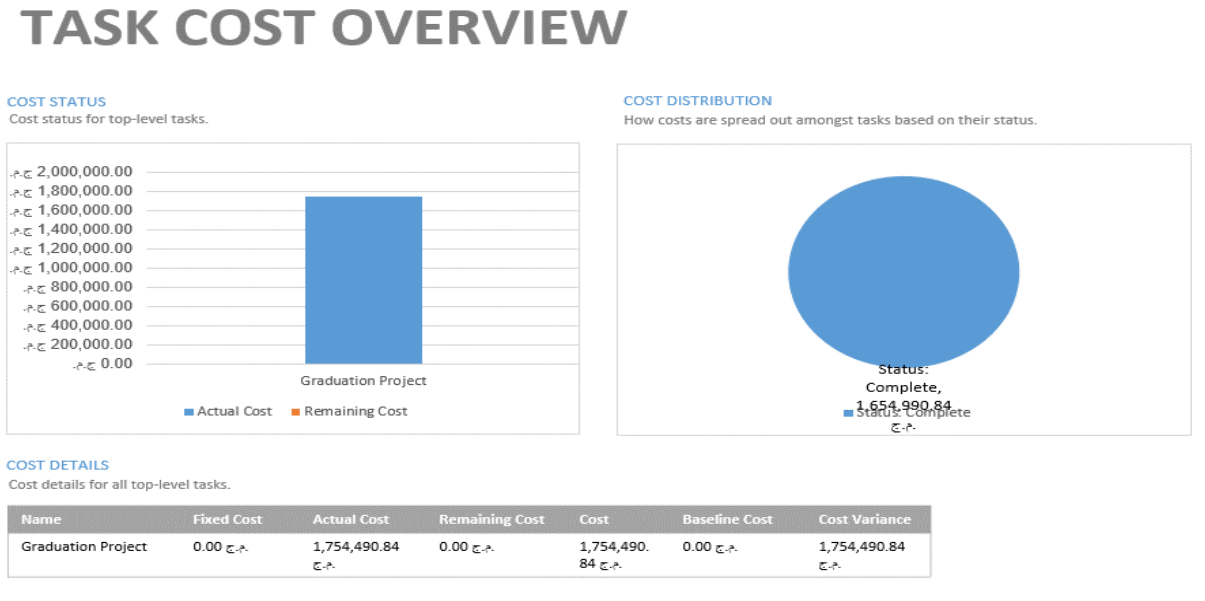
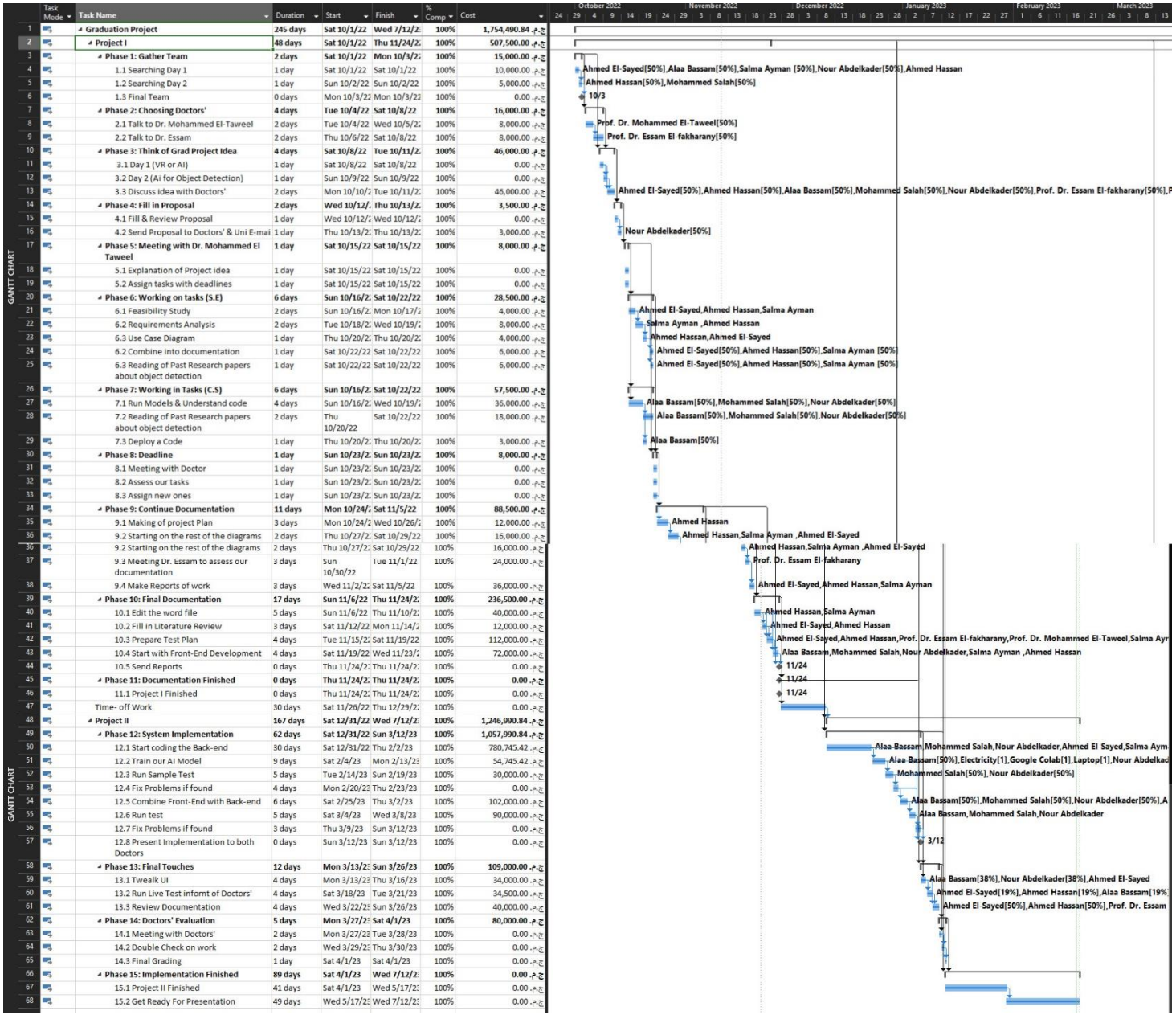


Figure 1.4 Task Cost Overview

The cost of a task represents the added costs of all the assignments which a task has, plus any fixed cost set for the task. In the case of summary tasks, the total cost is the added costs of the subtasks plus any fixed cost set for the summary task.

1.8 Project Plan



	Task Mode	Task Name	Duration	Start	Finish	% Comp	Cost
1		Graduation Project	245 days	Sat 10/1/22	Wed 7/12/2	100%	1,754,490.84
2		Project I	48 days	Sat 10/1/22	Thu 11/24/2	100%	507,500.00
3		Phase 1: Gather Team	2 days	Sat 10/1/22	Mon 10/3/2	100%	15,000.00
4		1.1 Searching Day 1	1 day	Sat 10/1/22	Sat 10/1/22	100%	10,000.00
5		1.2 Searching Day 2	1 day	Sun 10/2/22	Sun 10/2/22	100%	5,000.00
6		1.3 Final Team	0 days	Mon 10/3/22	Mon 10/3/22	100%	0.00
7		Phase 2: Choosing Doctors'	4 days	Tue 10/4/22	Sat 10/8/22	100%	16,000.00
8		2.1 Talk to Dr. Mohammed El-Taweel	2 days	Tue 10/4/22	Wed 10/5/2	100%	8,000.00
9		2.2 Talk to Dr. Essam	2 days	Thu 10/6/22	Sat 10/8/22	100%	8,000.00
10		Phase 3: Think of Grad Project Idea	4 days	Sat 10/8/22	Tue 10/11/2	100%	46,000.00
11		3.1 Day 1 (VR or AI)	1 day	Sat 10/8/22	Sat 10/8/22	100%	0.00
12		3.2 Day 2 (Ai for Object Detection)	1 day	Sun 10/9/22	Sun 10/9/22	100%	0.00
13		3.3 Discuss Idea with Doctors'	2 days	Mon 10/10/2	Tue 10/11/2	100%	46,000.00
14		Phase 4: Fill in Proposal	2 days	Wed 10/12/2	Thu 10/13/2	100%	3,500.00
15		4.1 Fill & Review Proposal	1 day	Wed 10/12/2	Wed 10/12/2	100%	0.00
16		4.2 Send Proposal to Doctors' & Uni E-mai	1 day	Thu 10/13/2	Thu 10/13/2	100%	3,000.00
17		Phase 5: Meeting with Dr. Mohammed El Taweel	1 day	Sat 10/15/22	Sat 10/15/22	100%	8,000.00
18		5.1 Explanation of Project idea	1 day	Sat 10/15/22	Sat 10/15/22	100%	0.00
19		5.2 Assign tasks with deadlines	1 day	Sat 10/15/22	Sat 10/15/22	100%	0.00
20		Phase 6: Working on tasks (S.E)	6 days	Sun 10/16/2	Sat 10/22/22	100%	28,500.00
21		6.1 Feasibility Study	2 days	Sun 10/16/2	Mon 10/17/2	100%	4,000.00
22		6.2 Requirements Analysis	2 days	Tue 10/18/2	Wed 10/19/2	100%	8,000.00
23		6.3 Use Case Diagram	1 day	Thu 10/20/2	Thu 10/20/2	100%	4,000.00
24		6.2 Combine into documentation	1 day	Sat 10/22/22	Sat 10/22/22	100%	6,000.00
25		6.3 Reading of Past Research papers about object detection	1 day	Sat 10/22/22	Sat 10/22/22	100%	6,000.00
26		Phase 7: Working in Tasks (C.S)	6 days	Sun 10/16/2	Sat 10/22/22	100%	57,500.00
27		7.1 Run Models & Understand code	4 days	Sun 10/16/2	Wed 10/19/2	100%	36,000.00
28		7.2 Reading of Past Research papers about object detection	2 days	Thu 10/20/22	Sat 10/22/22	100%	18,000.00
29		7.3 Deploy a Code	1 day	Thu 10/20/2	Thu 10/20/2	100%	3,000.00
30		Phase 8: Deadline	1 day	Sun 10/23/2	Sun 10/23/2	100%	8,000.00
31		8.1 Meeting with Doctor	1 day	Sun 10/23/2	Sun 10/23/2	100%	0.00
32		8.2 Assess our tasks	1 day	Sun 10/23/2	Sun 10/23/2	100%	0.00
33		8.3 Assign new ones	1 day	Sun 10/23/2	Sun 10/23/2	100%	0.00
34		Phase 9: Continue Documentation	11 days	Mon 10/24/2	Sat 11/5/22	100%	88,500.00
35		9.1 Making of project Plan	3 days	Mon 10/24/2	Wed 10/26/2	100%	12,000.00
36		9.2 Starting on the rest of the diagrams	2 days	Thu 10/27/2	Sat 10/29/22	100%	16,000.00
36		9.2 Starting on the rest of the diagrams	2 days	Thu 10/27/2	Sat 10/29/22	100%	16,000.00
37		9.3 Meeting Dr. Essam to assess our documentation	3 days	Sun 10/30/22	Tue 11/1/22	100%	24,000.00
38		9.4 Make Reports of work	3 days	Wed 11/2/22	Sat 11/5/22	100%	36,000.00
39		Phase 10: Final Documentation	17 days	Sun 11/6/22	Thu 11/24/2	100%	236,500.00
40		10.1 Edit the word file	5 days	Sun 11/6/22	Thu 11/10/2	100%	40,000.00
41		10.2 Fill in Literature Review	3 days	Sat 11/12/22	Mon 11/14/2	100%	12,000.00
42		10.3 Prepare Test Plan	4 days	Tue 11/15/2	Sat 11/19/22	100%	112,000.00
43		10.4 Start with Front-End Development	4 days	Sat 11/19/22	Wed 11/23/2	100%	72,000.00
44		10.5 Send Reports	0 days	Thu 11/24/2	Thu 11/24/2	100%	0.00
45		Phase 11: Documentation Finished	0 days	Thu 11/24/2	Thu 11/24/2	100%	0.00
46		11.1 Project I Finished	0 days	Thu 11/24/2	Thu 11/24/2	100%	0.00
47		Time- off Work	30 days	Sat 11/26/22	Thu 12/29/2	100%	0.00
48		Project II	167 days	Sat 12/31/22	Wed 7/12/2	100%	1,246,990.84
49		Phase 12: System Implementation	62 days	Sat 12/31/22	Sun 3/12/23	100%	1,057,990.84
50		12.1 Start coding the Back-end	30 days	Sat 12/31/22	Thu 2/2/23	100%	780,745.42
51		12.2 Train our AI Model	9 days	Sat 2/4/23	Mon 2/13/23	100%	54,745.42
52		12.3 Run Sample Test	5 days	Tue 2/14/23	Sun 2/19/23	100%	30,000.00
53		12.4 Fix Problems if found	4 days	Mon 2/20/23	Thu 2/23/23	100%	0.00
54		12.5 Combine Front-End with Back-end	6 days	Sat 2/25/23	Thu 3/2/23	100%	102,000.00
55		12.6 Run test	5 days	Sat 3/4/23	Wed 3/8/23	100%	90,000.00
56		12.7 Fix Problems if found	3 days	Thu 3/9/23	Sun 3/12/23	100%	0.00
57		12.8 Present Implementation to both Doctors	0 days	Sun 3/12/23	Sun 3/12/23	100%	0.00
58		Phase 13: Final Touches	12 days	Mon 3/13/23	Sun 3/26/23	100%	109,000.00
59		13.1 Tweaik UI	4 days	Mon 3/13/23	Thu 3/16/23	100%	34,000.00
60		13.2 Run Live Test infornt of Doctors'	4 days	Sat 3/18/23	Tue 3/21/23	100%	34,500.00
61		13.3 Review Documentation	4 days	Wed 3/22/23	Sun 3/26/23	100%	40,000.00
62		Phase 14: Doctors' Evaluation	5 days	Mon 3/27/23	Sat 4/1/23	100%	80,000.00
63		14.1 Meeting with Doctors'	2 days	Mon 3/27/23	Tue 3/28/23	100%	0.00
64		14.2 Double Check on work	2 days	Wed 3/29/23	Thu 3/30/23	100%	0.00
65		14.3 Final Grading	1 day	Sat 4/1/23	Sat 4/1/23	100%	0.00
66		Phase 15: Implementation Finished	89 days	Sat 4/1/23	Wed 7/12/2	100%	0.00
67		15.1 Project II Finished	41 days	Sat 4/1/23	Wed 5/17/23	100%	0.00
68		15.2 Get Ready For Presentation	49 days	Wed 5/17/23	Wed 7/12/23	100%	0.00

Figure 1.5 project plan

This is a roadmap for achieving the result of our project. It shows the different tasks and how is the materials (Electricity, google Colab subscription, etc.) and how the human resources are allocated (Team Members and Supervising professors). We divided our team to 2 groups S.E(Ahmed Hassan, Ahmed Elsayed, Salma Ayman) which work on Project I and C.S (Alaa Bassem, Nour Abdelkader, Mohammed Salah) which work on Project II.

1.9 Thesis Organization

The thesis is organized into eight chapters to provide a logical and comprehensive exploration of the real-time detection and searching system using YOLO. Chapter 1: Introduction introduces the problem, motivation, objectives, and constraints of the system. Chapter 2: Background information covers the fundamental concepts and theories relevant to object detection. Chapter 3: Related work reviews existing literature and research in the field. Chapter 4: Analysis conducts a detailed analysis of system requirements and constraints. Chapter 5: Design presents the system's architecture, algorithms, and methodologies. Chapter 6: Implementation describes the practical aspects of developing the system. Chapter 7: Testing evaluates the system's performance and effectiveness. Finally, Chapter 8: Conclusion and Future Work summarizes the findings, discusses implications, and provides suggestions for future research. This organizational structure ensures a coherent progression, from problem identification to solution design, implementation, testing, and future directions.

Chapter 2

Background

2.1 Computer Vision

Computer vision is an interdisciplinary field that deals with the science of enabling machines to interpret, understand, and analyze digital images and videos from the world around us. It involves a combination of techniques from various fields such as mathematics, computer science, physics, and neuroscience.

The goal of computer vision is to enable machines to perceive the world in a manner similar to how humans do, using visual information as the primary input source. By analyzing and processing images and videos, machines can extract meaningful information, detect patterns, and make decisions based on this data.

There are several techniques used in computer vision, and one of the most popular is deep learning, specifically Convolutional Neural Networks (CNNs). CNNs have become a popular technique for image recognition tasks because they can automatically learn and extract features from images without human intervention. The networks learn to identify different patterns and features in the images such as edges, corners, and textures, which are then used to classify or recognize objects.

Computer vision is used in various fields, including medical imaging, robotics, security and surveillance, agriculture, and autonomous vehicles. Some examples of computer vision applications include:

Medical imaging: Computer vision is used in medical imaging to analyze medical images and identify abnormalities or potential diseases. This includes identifying tumors, detecting broken bones, and analyzing brain activity through Magnetic Resonance Imaging (MRI) scans.

Robotics: Computer vision is used in robotics to enable machines to perceive and navigate their environments. This includes detecting objects in their environment, recognizing obstacles, and identifying the position and orientation of objects.

Security and surveillance: Computer vision is used in security and surveillance systems to monitor public areas, identify potential threats, and detect criminal activity. This includes facial recognition, object detection, and tracking.

Agriculture: Computer vision is used in agriculture to monitor crop growth, detect plant diseases, and optimize crop yields. This includes analyzing crop images to detect changes in

plant health and growth patterns.
Autonomous vehicles: Computer vision is a critical component of autonomous vehicles. By analyzing the environment in real-time, autonomous vehicles can make decisions on how to navigate and avoid obstacles.

While it's getting easier to obtain resources to develop computer vision applications, an important question to answer early on is: What exactly will these applications do? Understanding and defining specific computer vision tasks can focus and validate projects and applications and make it easier to get started.

Here are a few examples of established computer vision tasks:

Image classification sees an image and can classify it (a dog, an apple, a person's face). More precisely, it is able to accurately predict that a given image belongs to a certain class. For example, a social media company might want to use it to automatically identify and segregate objectionable images uploaded by users.

Object detection can use image classification to identify a certain class of image and then detect and tabulate their appearance in an image or video. Examples include detecting damages on an assembly line or identifying machinery that requires maintenance.

Object tracking follows or tracks an object once it is detected. This task is often executed with images captured in sequence or real-time video feeds. Autonomous vehicles, for example, need to not only classify and detect objects such as pedestrians, other cars and road infrastructure, they need to track them in motion to avoid collisions and obey traffic laws

Content-based image retrieval uses computer vision to browse, search and retrieve images from large data stores, based on the content of the images rather than metadata tags associated with them. This task can incorporate automatic image annotation that replaces manual image tagging. These tasks can be used for digital asset management systems and can increase the accuracy of search and retrieval.

Overall, computer vision is a rapidly growing field with endless applications in various industries. With the increasing availability of data and advances in deep learning techniques, it is expected that computer vision will continue to be an essential technology in many fields, improving efficiency, and enabling new capabilities.

2.2 YOLOv5 vs. YOLOv8: The Comparison

2.2.1 What is Yolov5

YOLOv5 is indeed a popular and powerful object detection model that builds upon the success of its predecessors. Developed by Ultralytics, YOLOv5 was introduced in 2020 as an upgrade to YOLOv3. It is built on the PyTorch deep learning framework and has gained significant attention and adoption in the computer vision community.

One of the key strengths of YOLOv5 is its speed. It leverages a single-shot detection approach, which means it can detect objects in an image in a single pass. This makes it faster than many other object detection models while still maintaining good accuracy. YOLOv5 achieves this speed through its anchor-based approach, where anchor boxes of different scales and aspect ratios are used to predict object bounding boxes and class probabilities.

YOLOv5 offers a user-friendly and flexible architecture. It comes with pre-trained models that can be easily fine-tuned on custom datasets, allowing developers to adapt it to their specific object detection tasks. The model architecture is designed to be lightweight and efficient, making it suitable for real-time applications on devices with limited computational resources.

The deep learning-based object detection technique known as YOLOv5 (You Only Look Once version 5) uses a convolutional neural network (CNN) to identify and classify items in images or videos.

A general step of YOLOv5's operation is provided below:

1. **Input image:** The input image is first resized and converted into a format that the neural network can understand.
2. **Backbone network:** To extract features from the input image, YOLOv5 uses a CNN as its backbone network. The network has numerous convolutional layers that can be used for dimensionality reduction and feature extraction. A collection of feature maps with various levels of visual detail is what the backbone network produces as its output.
3. **Neck network:** The neck network uses additional convolutional layers to include data from various scales and further enhance the feature maps created by the backbone network. A series of feature maps with a higher level of abstraction are produced by the neck network.
4. **Head network:** The head network predicts bounding boxes and class probabilities for objects in the image using feature maps created by the neck network. The head network is formed up of a number of convolutional layers as well as a number of output layers that generate the results. A predetermined number of bounding boxes are predicted by

each output layer, which are then filtered and improved based on their confidence scores and overlap with other boxes.

5. **Non-maximum suppression (NMS):** is used to get rid of overlapping and redundant boxes after, the head network generates predicted bounding boxes and class probabilities. This process makes sure that only the most accurate and relevant boxes are kept.
6. **Output:** The YOLOv5 algorithm generates a set of bounding boxes and class probabilities for the items found in the input image as its final output.

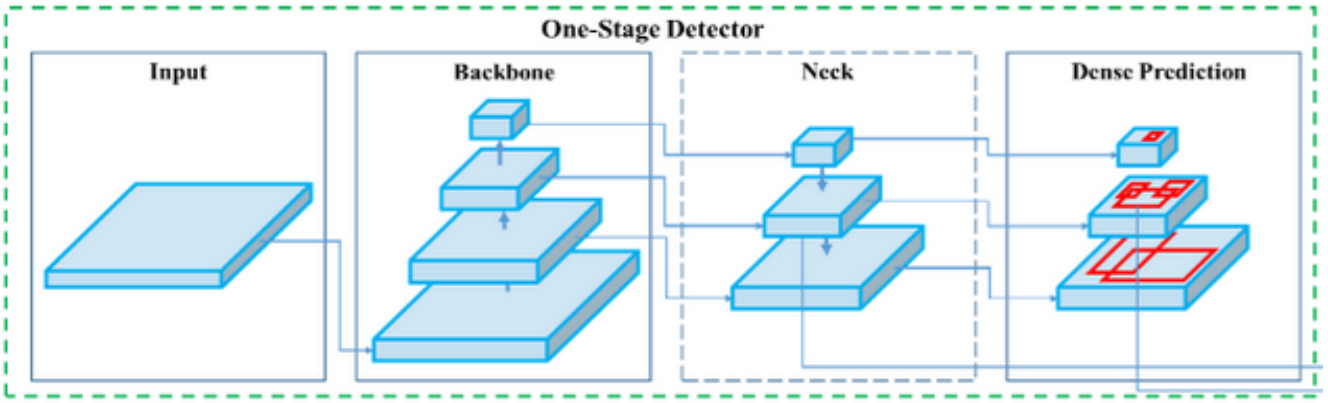


Figure 2.1 yolov5 architecture

Compared to its predecessors, YOLOv5 has improved accuracy. It introduces a novel backbone network, called CSPDarknet53, which enhances feature extraction capabilities. Additionally, it incorporates various advanced techniques such as focal loss, label smoothing, and data augmentation, contributing to better object detection performance.

YOLOv5 provides a comprehensive set of tools and utilities for training, evaluation, and deployment. The Ultralytics team has developed a user-friendly Python library that simplifies the process of training and deploying YOLOv5 models. The library includes functionalities for data preprocessing, model configuration, training, inference, and result analysis.

The model has achieved state-of-the-art results on popular object detection benchmarks, demonstrating its effectiveness and competitiveness. However, it's important to note that the performance of any object detection model, including YOLOv5, heavily depends on the quality and diversity of the training data, as well as the fine-tuning and hyperparameter settings.

Overall, YOLOv5 offers a combination of speed, accuracy, ease of use, and flexibility, making it an attractive choice for developers and researchers working on object detection tasks. Its continuous development and active community support contribute to its growing popularity and utility in various applications such as autonomous driving, surveillance systems, and robotics.

2.2.2 What is Yolov8

YOLOv8 is the latest iteration of the YOLO family of models. YOLO stands for You Only Look Once and these series of models are thus named because of their ability to predict every object present in an image with one forward pass.

The main distinction introduced by the YOLO models was the framing of the task at hand. The authors of the paper reframed the object detection task as a regression problem (predict the bounding box coordinates) instead of classification.

YOLO models are pre-trained on huge datasets such as COCO and ImageNet. This gives them the simultaneous ability to be the Master and the Student. They provide highly accurate predictions on classes they are pre-trained on (master ability) and can also learn new classes comparatively easily (student ability).
YOLO models are also faster to train and have the ability to produce high accuracy with smaller model sizes. They can be trained on single GPUs, making them more accessible to developers like us.

YOLOv8 is the latest iteration of these YOLO models (as of early 2023). It has undergone a few major changes from its ancestors, such as anchor free detection, the introduction of C3 convolutions, and mosaic augmentation.

Here is an image showing the timeline of YOLO object detection models

Figure 2.2 Yolo timeline

Some of the critical features of YOLO v8 include the following:

1. Improved accuracy and speed compared to previous versions of YOLO.
2. An improved Efficient Net-based backbone network that increases the model's capacity to capture high-level features.
3. A new feature fusion module that integrates features from multiple scales.
4. Improved methods for data augmentation, such as MixUp and CutMix.

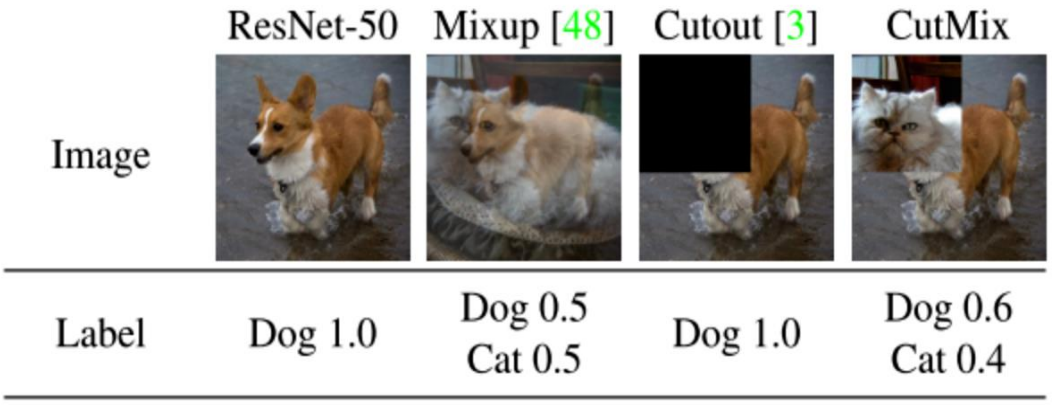


Figure 2.3 Image augmentation techniques

2.2.3 Models Available in YOLOv8

The YOLOv8 models for detection, segmentation, and classification each contain five models. The smallest and fastest is the YOLOv8 Nano, while the most accurate and slowest is the YOLOv8 Extra Large (YOLOv8x).

YOLOv8n	YOLOv8s	YOLOv8m	YOLOv8l	YOLOv8x

Table 2.1 Yolov8 models

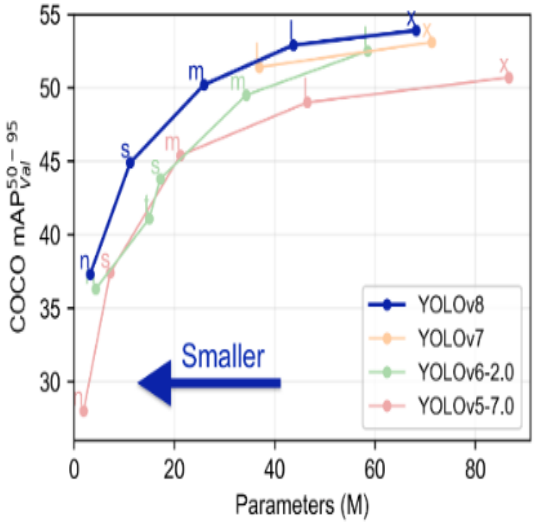


Figure 2.4 Yolo size comparison

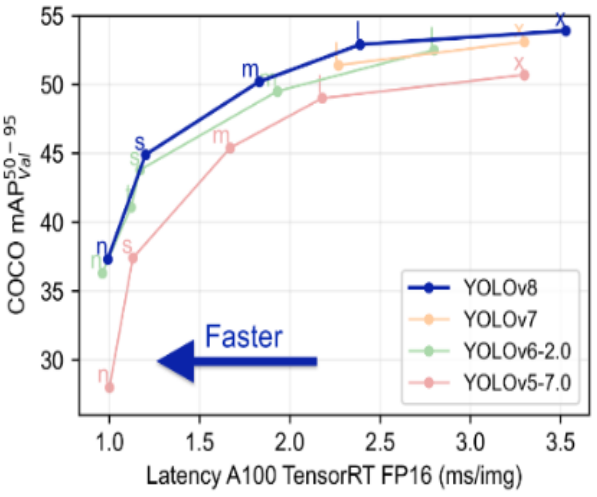


Figure 2.5 Yolo inference speed

Pre-trained models included with YOLOv8 include the following:

1. Object Detection checkpoints trained on the COCO detection dataset with an image resolution of 640.

2. Instance segmentation checkpoints trained on the COCO segmentation dataset with an image resolution of 640.

3. Image classification models pretrained on the ImageNet dataset with an image resolution of 224.

2.2.4 Which one is better

When it comes to object detection, there are many models available. However, YOLOv8 and YOLOv5 are two of the most popular and state-of-the-art models created by Ultralytics. YOLOv8 is the latest addition to the YOLO family, which builds upon the success of previous versions and introduces new features and improvements to boost performance and flexibility. YOLOv5, on the other hand, is known for its speed, simplicity, and accuracy.

The new architecture used by YOLOv8 combines both FAN and PAN modules. While PAN aggregates features from many layers of the network to increase accuracy, FPN generates feature maps at various scales and resolutions. The combined FAN and PAN modules produce better results than YOLOv5, which employs a modified CSPDarknet architecture.

Another difference the two models have is their training data. In comparison to YOLOv5, YOLOv8 was learned on a larger and more varied dataset. While YOLOv5 was trained primarily on the COCO dataset, YOLOv8 was trained on a combination of the COCO 2017 dataset and several other datasets. Because of this, YOLOv8 performs better on a larger variety of photos.

The COCO 2017(Common Objects in Context) dataset is a widely used benchmark dataset in computer vision for object detection, segmentation, and captioning tasks. It comprises a vast collection of images covering diverse everyday scenes and objects. With annotations for 80 common object categories, including bounding boxes, segmentation masks, and annotations, it provides a comprehensive resource for training and evaluating models with over 200,000 labeled images.

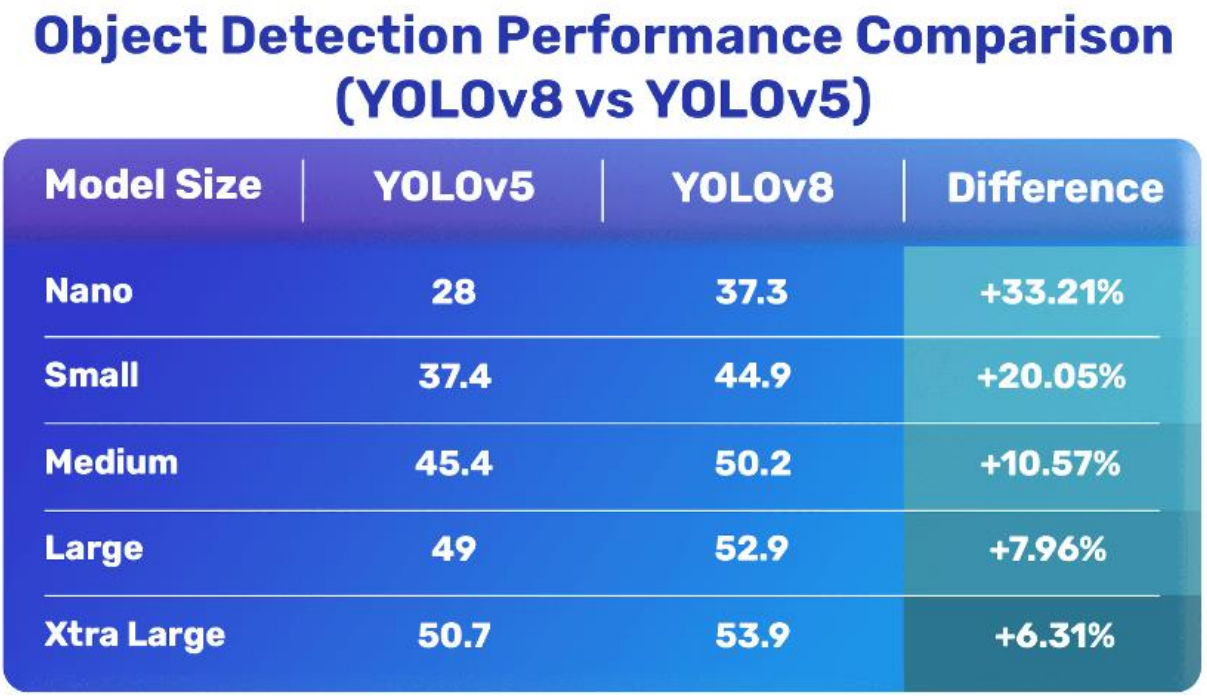


Figure 2.6 Yolov5 vs Yolov8

- Speed**:**
  Both YOLOv8 and YOLOv5 are fast object detection models, capable of processing images in real-time. However, YOLOv8 is faster than YOLOv5, making it a better choice for applications that require real-time object detection.

- Accuracy:

  Accuracy is a critical factor to consider when choosing an object detection model. In this regard, YOLOv8 is more accurate than YOLOv5, thanks to the several improvements made in its architecture.

Based on the above information, we opted to use Yolov8 for better detection, based on the improved accuracy and improved speed, provided by such model and we also have chosen Yolov8x to be a bit lighter while also giving us a better performance.

Right away, YOLOv8 models appear to perform much better than the previous YOLO models. Not only YOLOv5 models, YOLOv8 is ahead of the curve against YOLOv7 and YOLOv6 models also.

Chapter 3

Related Work and Similar Systems

Our system solves the problem of manual and time-consuming detection and search. This system adds the capability to quickly process video and image data with high accuracy, The use of YOLO enables parallel processing and efficient predictions, and the system's customizable search capabilities enable efficient retrieval of relevant information. Overall, this system improves efficiency and accuracy in object detection and search, making it a valuable tool for various applications.

3.1 Related Work

3.1.1 Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions paper

This study has improved the YOLOv5 architecture and training process, mainly by increasing the efficacy and accuracy of object detection for drone photographs. The authors offer methods for resolving the unique characteristics of drone photos, such as using data augmentation methods and enhancing the model architecture. These modifications aim to enhance the model's ability to accurately and successfully detect items under a variety of environmental conditions.

The study provides valuable information on the challenges and potential solutions for object detection in drone photographs, even though it does not go into detail regarding the suggested improvements. The affiliations of the authors highlight their experience and competence in the field of electrical and computer engineering.

These are the results:

- Model: YOLOv5_Ours
- Backbone: CSPdarknet
- mAP (0.5): 95.5

3.1.2 Improved YOLOv5 network for real-time multi-scale traffic sign detection paper

To solve the challenges of identifying signs of varying sizes, the authors offer modifications to the YOLOv5 model.
By altering the YOLOv5 architecture and training process, the authors hope to increase the multi-scale traffic sign identification accuracy and efficiency. Even if the details of the suggested improvements are not covered in the provided information, the study significantly advances the field of computer vision and object detection, particularly in the context of traffic sign recognition. The author show the potential significance of the authors' study results.
The following are the YOLOv5 model's real-time multi-scale traffic sign identification performance metrics.

- Model Size: 16.3 million parameters
- Parameters: 8.039 million
- Floating Point Operations (FLOPs): 17.9 billion
- Frames Per Second (FPS): 95

3.1.3 Real-Time Flying Object Detection with YOLOv8

The author describes the problem of in-flight object detection in real time and the requirement for effective object detection systems.

Discuss the YOLOv8 model architecture, data pre-processing, and training procedure used in the real-time flying object detection method.

Summarize the results of the research, including the performance criteria used to evaluate the YOLOv8 model's accuracy and speed for flying object recognition. Give samples or examples of the model in use to show the way it works in real-world scenarios.

Analyze the results and evaluate the efficacy of the YOLOv8 model for the detection of flying objects, taking consideration both its advantages and disadvantages. Discuss the model's possible uses in various contexts and identify any areas that require additional study or development.

Include recommendations for using the YOLOv8 model for real-time flying object identification in practical applications together with a summary of the study's key findings and an explanation of their significance.

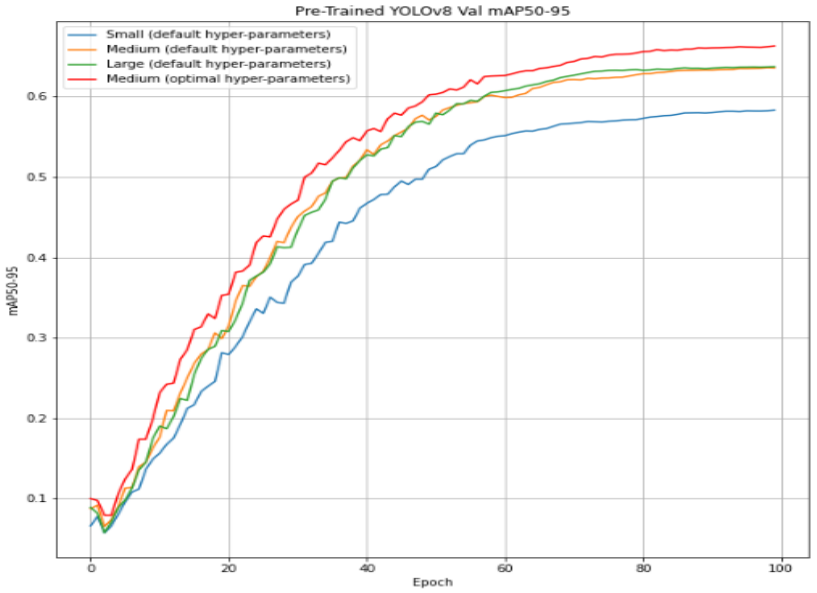


Figure 3.1 Yolov8 validation mAP50-95

3.1.4 Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8 paper

In this paper the author describes the issue with helmet violation detection and the requirement for quick detection tools. For the purpose of ensuring traffic safety, real-time helmet detection is essential in a variety of weather and time-of-day situations. For instance, varying weather and lighting may significantly affect motorcycle riders' visibility and their helmets, making it difficult for traditional systems to accurately detect helmet usage violations.

Using a few-shot data sampling technique and YOLOv8, describe the methods used for multi-class helmet violation detection. Describe the few-shot data sampling technique, the YOLOv8 model architecture, and the data preparation and training procedures employed.

Summarize the results of the research, including the performance measures used to evaluate the YOLOv8 model's accuracy and speed for multi-class helmet violation detection. Also, include representations or scenarios of the model in use to show its value in real-world scenarios.

Analyze the results and evaluate the effectiveness of the YOLOv8 model for detecting multi-class helmet violations, taking note of both its advantages and disadvantages. Discuss the model's possible uses in various contexts and identify any areas that require additional study or development.

Conclude the research's main results, highlight their importance, and provide suggestions for using the YOLOv8 model in real-world settings to identify multi-class helmet violations.

Model	mAP	fps
yolov5	0.3988	160
yolov7	0.4061	163
yolov8	0.5136	162
yolov5+TTA	0.5346	100
yolov8+TTA	**0.5861**	**95**

Table 3.2 Experimental results on the test dataset

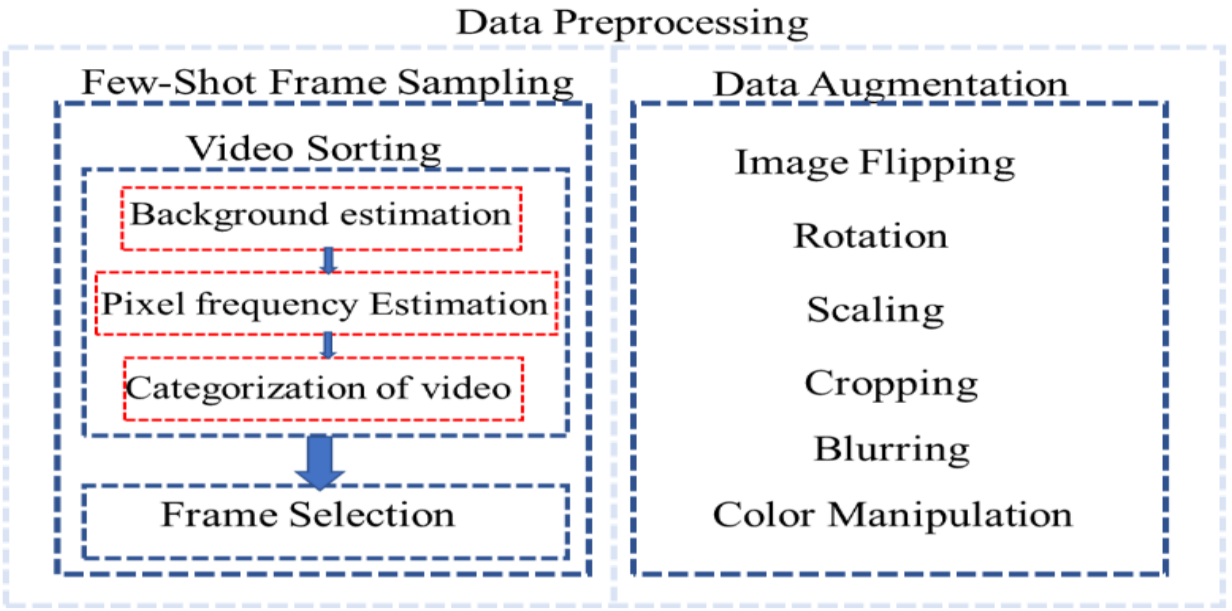


Figure 3.3 Data preprocessing: frame sampling and data augmentation

Chapter 4

Analysis

The primary goal of analysis is to provide insights and make informed decisions based on the results. The process of analysis requires critical thinking, creativity, and a deep understanding of the subject matter. Effective analysis can lead to significant improvements in various areas, such as product design, process optimization, and risk management.

4.1 Functional Requirements

User:

- Signup in the application providing some personal information

  Users at first need to create an account (Signup) when first using the application; thus, providing their first and last name, setting a username, Email ID, phone number and so on till the account is created.

Login

- After creating an account, users then login with their Username and password. Users are granted access to the application if the username and password is correct.

- Upload from gallery (video/photo)

  This is when you want to upload a photo/video you already have on your phone.

- View Product Description

  This is perhaps the most important feature for the user, it's when the object is finally is detected and all available info is shown up on your display. A user can buy direct through our app, not only that, but also getting a discount for doing that.

- Share link with friends

  This is a share feature when you want to notify your friends about a product you detected and is happy with. Sharing is Caring

- Recent Scans

  This is basically where you would find a list of all your recent scans

Admin:

- Signup in the application providing some personal information

  Users at first need to create an account (Signup) when first using the application; thus, providing their first and last name, setting a username, Email ID, phone number and so on till the account is created.

- Login

  After creating an account, users then login with their Username and password. Users are granted access to the application if the username and password is correct.

- Manage users' data

  This is to ensure that all data input from the users are in the correct format; thus, the admin is responsible for making the system accept data in certain formats, so we don't use many resources and also have validation through our system.

- Manage Database Schema

  An admin is responsible for managing and maintain the database, ofc there will be different teams perhaps for this, but overall, they are responsible for the database architecture and retrieval speeds and for any optimization needed.

- Responsible for System Maintenance

  This also part of the admins' team responsibility, they should maintain the whole system either from attacks or failures or optimizing servers computing power or installing of new servers.

- Manage Payment Process

  As mentioned above the user can purchase items directly through our application, this means we need a higher level of security for those transactions.

Store:

- Send available items to application

  To do business with store, we must come to agreement that we need a duplicate of their database to retrieve items from. So, we need to ensure that no harm is done to the stores, conduct our business with top quality and earn their trust.

- Send percent of every paid item

  We as business partners must come to a deal so that we can guarantee users their discount coupon and seek some incentive from the stores for making new wave of people purchase from them.

4.2 Non-functional Requirements

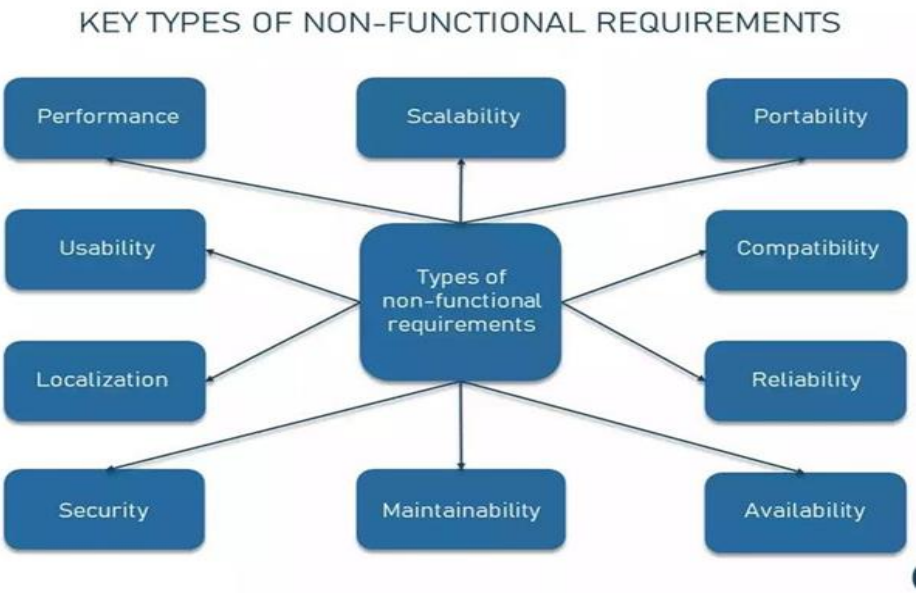


Figure 4.1 Non-Functional Requirements

Usability

- How easily a user can achieve their goal in a single page visit.
- How user can ease to use application
- How quickly they perform the tasks in the store application
- How memorable and intuitive the design of application.

Security

- Only the system data administrator can assign roles and change access permissions to the system.
- The application must be resilient to any kind of attacks, including DDoS and XSS attacks.
- The payment processing gateway must be PCI DSS compliant.
- How to secure data of user

Performance

- The application should load in less than 4 seconds for scanning video photo on iOS 10+, Safari on 4G.
- Testing application. Performance will help you understand whether you've achieved your KPIs set out in non-functional requirements.

Scalability

- Our main goal for the next two years is internationalization, so the application shall have multiple views for each country
- The system must be scalable enough to support 200,000 users at the same time while maintaining optimal performance

Reliability

- The database update process must roll back all related updates when any update fails

4.3 Diagrams

4.3.1 Use-Case Diagram

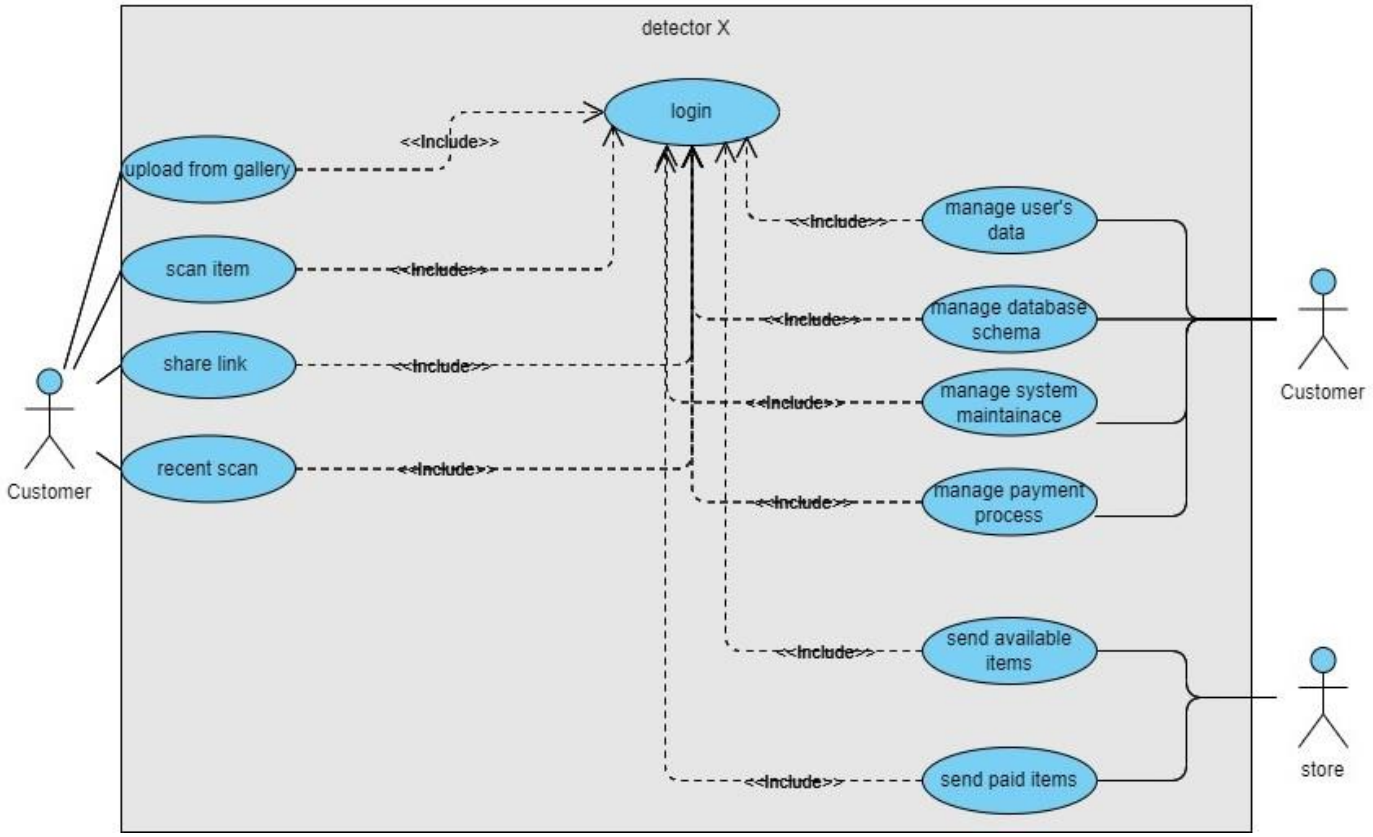


Figure 4.2 Use-case diagram

Primary Actors	Use Cases
User	1. Upload Image/Video 2. Share Link 3. Recent Scan
Admin	1. Manage User Data 2. Manage Database Schema 3. Maintain System 4. Secure method of payment Process
Store	1. Send Available Item 2. Sent Discount Percent of Items

Table 4.1 actors and use cases

4.3.2 UC-1 (Upload Image/Video)

ID and Name:	UC-1 Upload Image/Video		
Created by:	Ahmed EL-Sayed	Date Created:	December 27,2022
Primary Actor:	User	Secondary Actor:	N/A
Description:	A user will open the application, and from gallery upload a picture/video, the app automatically the items found in the uploaded material		
Trigger:	A user indicates that he/she wants to upload from gallery		
Preconditions:	PRE-1 User already registered into the application PRE-2 he/she have picture/video in gallery		
Postconditions:	Post-1 Image/Video is uploaded successfully Post-2 Image/video is processed by the system		
Normal Flow:	1.0 Upload Image/Video 1. User opens the application 2. User select upload from gallery option 3. System confirm image/video uploaded 4. System start processing image/video		
Alternative Flows:	N/A		
Exceptions:	1.0 E1 Uploaded Image/video isn't clear 1. System informs user to upload a well-lit Image or asks for high-res video 2. If user cancels the process, system will return to home screen		
Priority:	High		
Frequency of Use:	Approximately 1 time per user		
Business rules:	BR-1, BR-2, BR-3		
Other Information:	N/A		

Table 4.2 Use case 1

4.3.3 UC-2 Share link

ID and Name:	UC-2 Share Link		
Created by:	Ahmed Hassan	Date Created:	December 27,2022
Primary Actor:	User	Secondary Actor:	N/A
Description:	Feature that allows the user to share links of scanned item to family and friends		
Trigger:	After successful scanning user can share the link or press recent scan and select a scan and share the link		
Preconditions:	PRE-1 User already registered in the application PRE-2 User wants to share scanned items		
Postconditions:	N/A		
Normal Flow:	3.0 UC-3 Share link 1. User is registered into the application 2. User already scanned item 3. User generated share link 4. User send link to family and friends		
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	Approximately once per user		
Business rules:	BR-1, BR-2, BR-3, BR-4		
Other Information:	N/A		
Assumptions:	User already scanned items to be able to share		

Table 4.3 Use case 2

4.3.4 UC-3 Recent Scans

ID and Name:	UC-3 Recent Scans		
Created by:	Salma Ayman	Date Created:	December 27, 2022
Primary Actor:	User	Secondary Actor:	N/A
Description:	A Feature that allows the user to see a list of recent scans		
Trigger:	A user indicates that he/she wats to see list of recently scanned items		
Preconditions:	PRE-1 User already registered PRE-2 User scanned multiple objects		
Postconditions:	N/A		
Normal Flow:	4.0 UC-4 Recent Scans 1. User scanned items 2. User wants to see a list of scanned items 3. User press recent scans button		
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	At least once per user		
Business rules:	BR-1, BR-2, BR-3		
Other Information:	N/A		
Assumptions:	Multiple items already scanned		

Table 4.4 Use case 3

Chapter 5

Design

5.1 Engineering design process

By adopting Agile principles and practices, we can effectively manage the development process and enhance our project outcomes. Here's why we believe Agile is beneficial for our team:

1. Collaboration and Teamwork: Agile promotes collaboration and teamwork among our project members. We can actively engage in regular communication, knowledge sharing, and problem-solving sessions. This collaborative environment fosters a sense of unity, allowing us to work together efficiently and leverage our individual strengths to overcome challenges.

2. Iterative Development and Feedback: With Agile, we can break down our project into smaller iterations or milestones. Each iteration focuses on specific objectives and delivers tangible results. By receiving feedback at regular intervals, we can incorporate suggestions, make improvements, and ensure our project is on the right track. This iterative development approach helps us build a refined final outcome.

3. Flexibility and Adaptability: Agile's flexibility enables us to adapt our approach based on changing requirements or research goals. We understand that graduation projects often involve evolving objectives, and Agile allows us to adjust our plans accordingly. We can embrace new insights, make necessary changes, and ensure our project remains aligned with the desired outcomes.

4. Continuous Learning and Improvement: Agile encourages continuous learning and improvement throughout the project. By reflecting on our progress, identifying areas for enhancement, and implementing feedback, we can enhance our skills and deliver higher-quality work. This iterative learning process empowers us to develop our abilities and produce a project that reflects our growth.

5. Demonstrating Progress and Achievements: Agile's emphasis on incremental development allows us to demonstrate our progress and achievements at various stages of the project. We can present tangible outcomes, share our evolving work, and receive valuable input from our academic advisors and mentors. This regular showcasing of our progress enables us to gather feedback, refine our ideas, and ensure the successful completion of our graduation project.

5.2 Diagrams

5.2.1 Activity Diagram

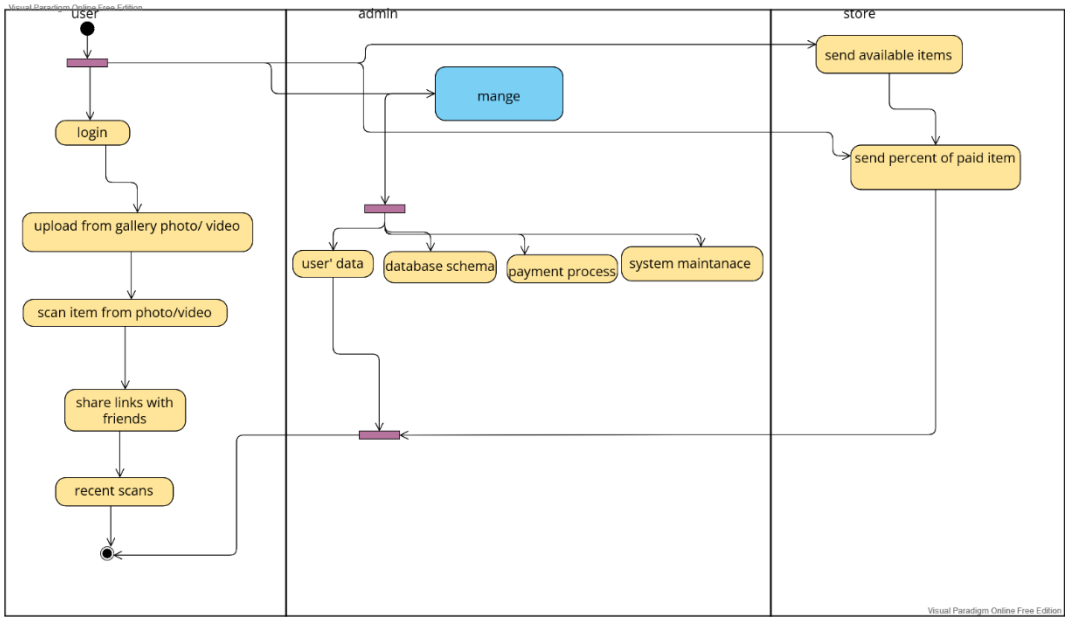


Figure 5.1 activity diagram

As seen in figure 5.1, The diagram explains the collection of activities a user has to do from login to exit function, it specifies what the user, admin and store can do while inside the system.

5.2.2 Class Diagram

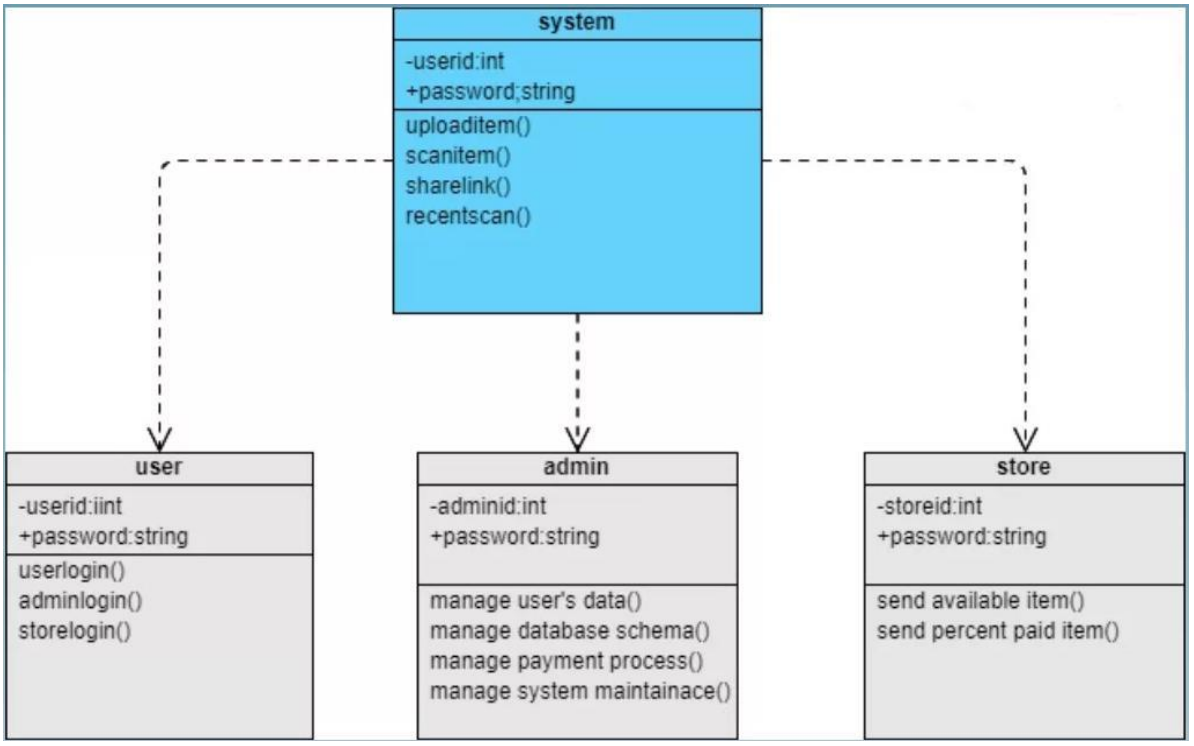


Figure 5.2 class diagram

As seen in figure 5.2, This diagram shows every role in our system and their respective functions, and their relationship.

5.2.3 Sequence Diagram

5.2.3.1 Sequence Diagram (Registration)

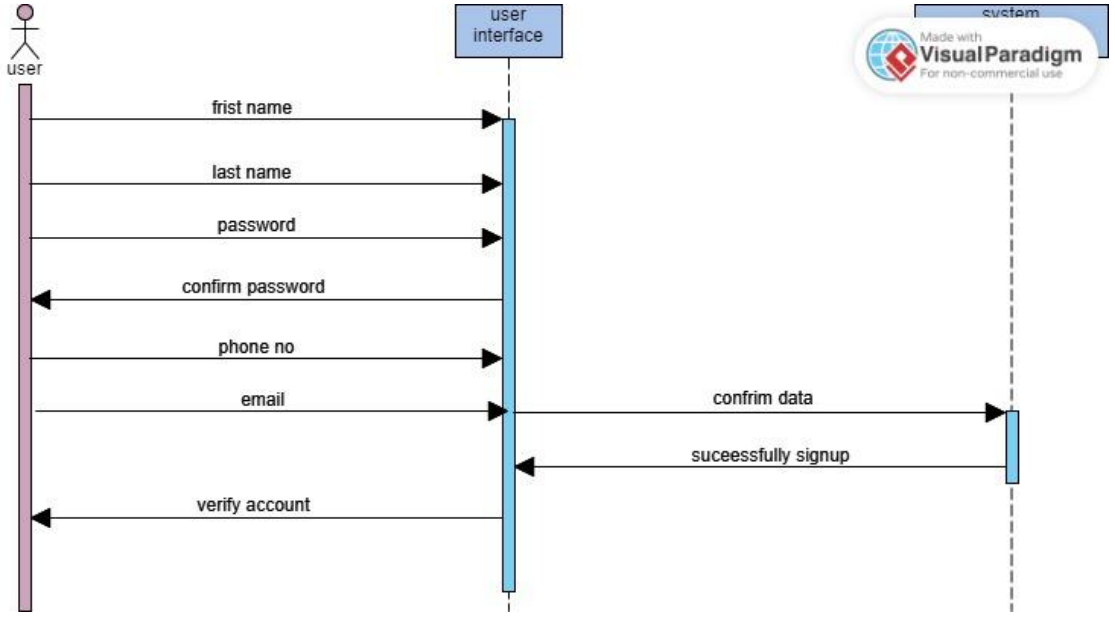


Figure 5.3 registration sequence diagram

As seen in figure 5.3, This describes the steps a user should take to register for an account, also show visually the reply from the system

5.2.3.2 Sequence Diagram (User)

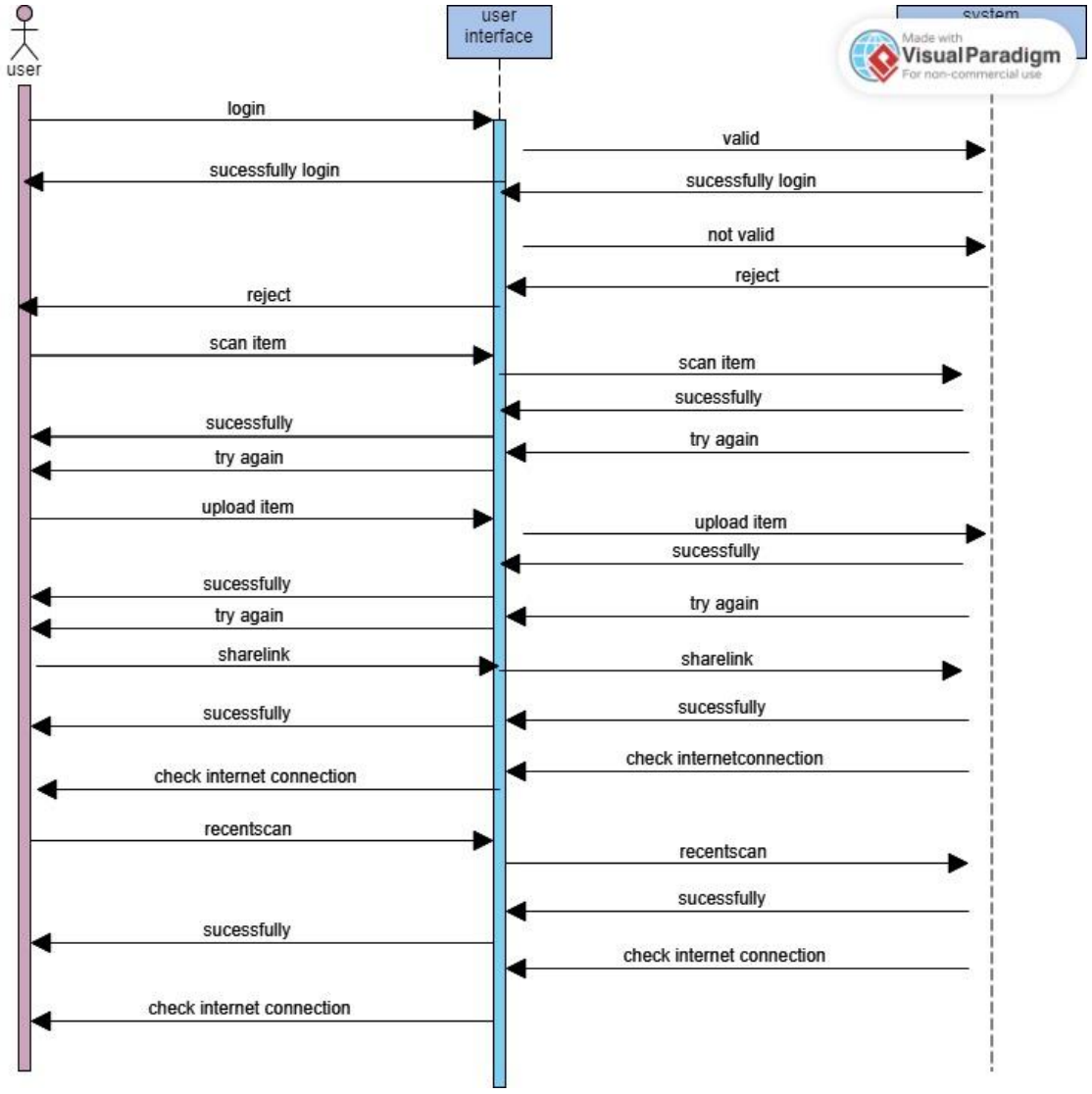


Figure 5.4 Functions sequence diagram

As seen in figure 5.4, This diagram explains the four functions of the user visually and how the system should respond, if there were no errors made.

5.2.3.3 Sequence Diagram (Admin)

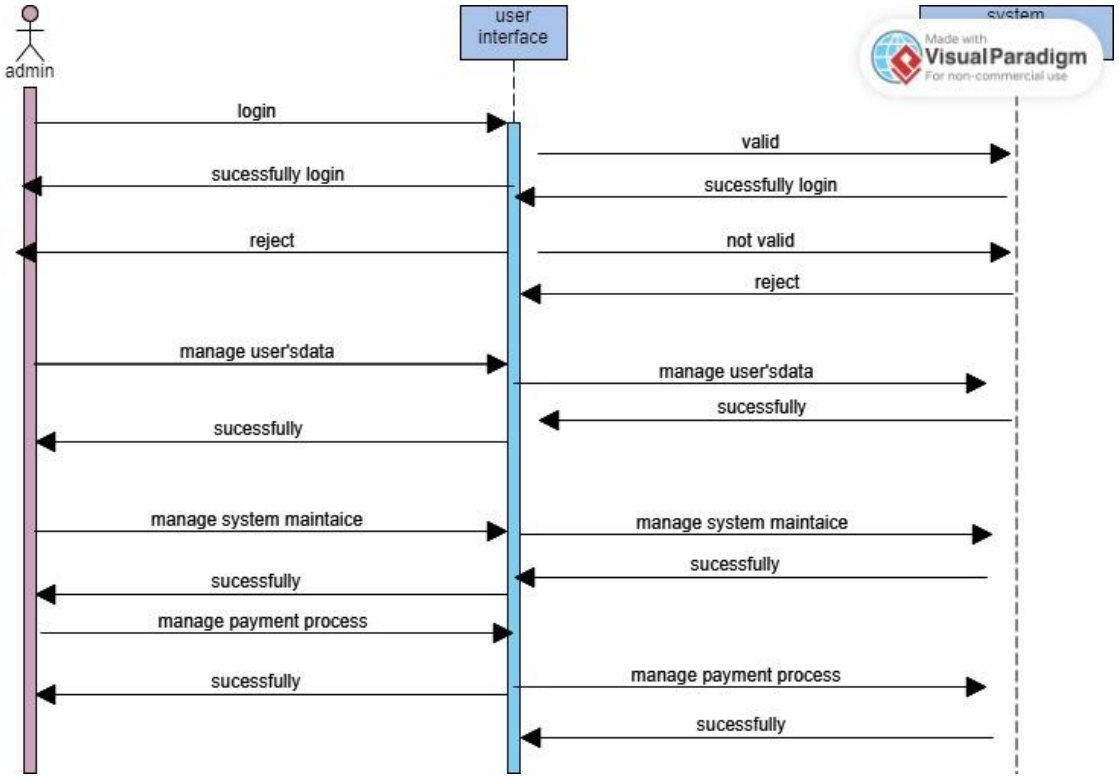


Figure 5.5 Admin sequence diagram

As seen in figure 5.5, This describes the sequence of action an admin has access to while managing the system.

5.2.3.4 Sequence Diagram (Stores)

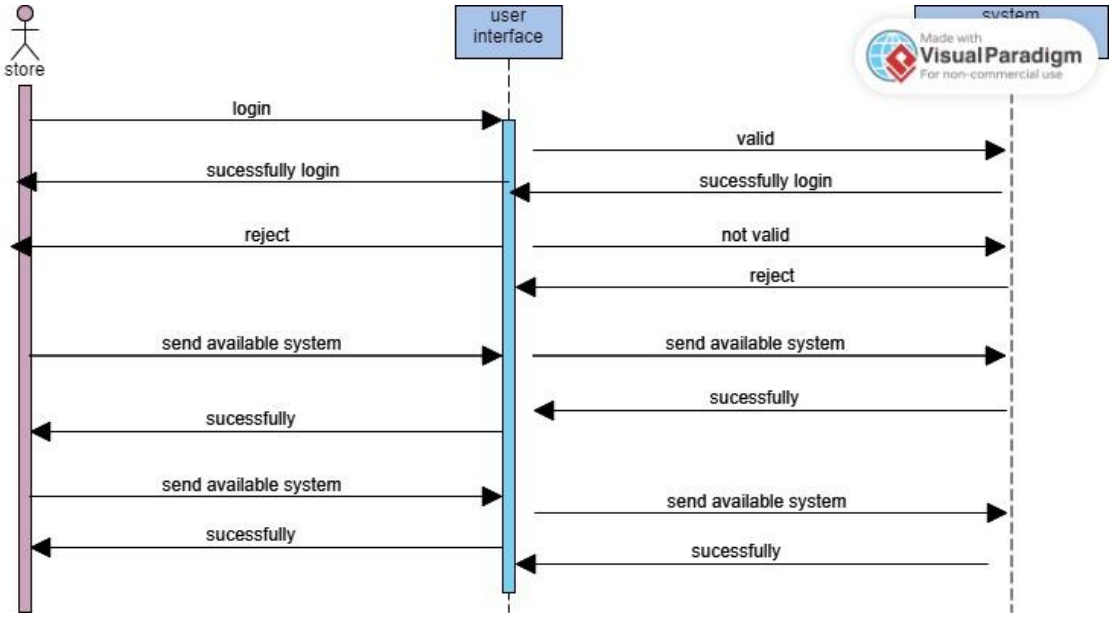


Figure 5.6 store sequence diagram

As seen in figure 5.6, this explains the sequence of action should take on the system.

5.3 Technologies and Tools Used

5.3.1 Artificial Intelligence

Image processing is the use of algorithms to process digital images. Image processing techniques are particularly useful in extracting features that would otherwise be difficult to obtain. By extracting these features, it could then be used to train artificial intelligence models. There are several image processing techniques used thresholding, Gaussian Blur, and rotating.

Deep learning is a subset of machine learning that uses Neural Networks with multiple layers. Neural networks are a set of algorithms that recognize features in a way that mimics human learning and the way biological neurons signal to one another. Deep learning is used in our application, where it classifies the documents given based on input data.

The data set is split into train set that is used to train the AI model, and the test set, which is used to test the model after it has been trained. Epoch is the total number of iterations of the all the training data.

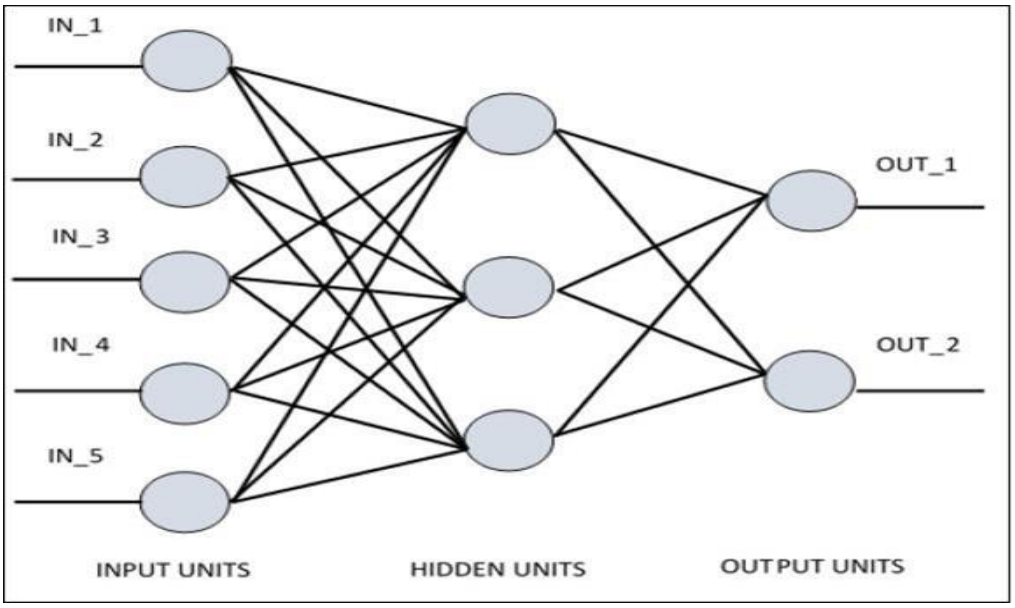


Figure 5.6 Artificial neural network

5.3.2 Yolov8

YOLOv8, or You Only Look Once version 8, is an object detection model that belongs to the YOLO (You Only Look Once) family of algorithms. YOLOv8 is designed to detect and localize objects in images or video frames with real-time performance. It utilizes a single neural network architecture that simultaneously predicts bounding boxes and class probabilities for multiple objects within an image. YOLOv8 builds upon the previous versions of YOLO by incorporating various improvements, including feature pyramid networks, anchor boxes, and a more efficient backbone network. These enhancements enable YOLOv8 to achieve higher accuracy and faster inference speed, making it a popular choice for a wide range of computer vision tasks, such as object detection, tracking, and autonomous driving.

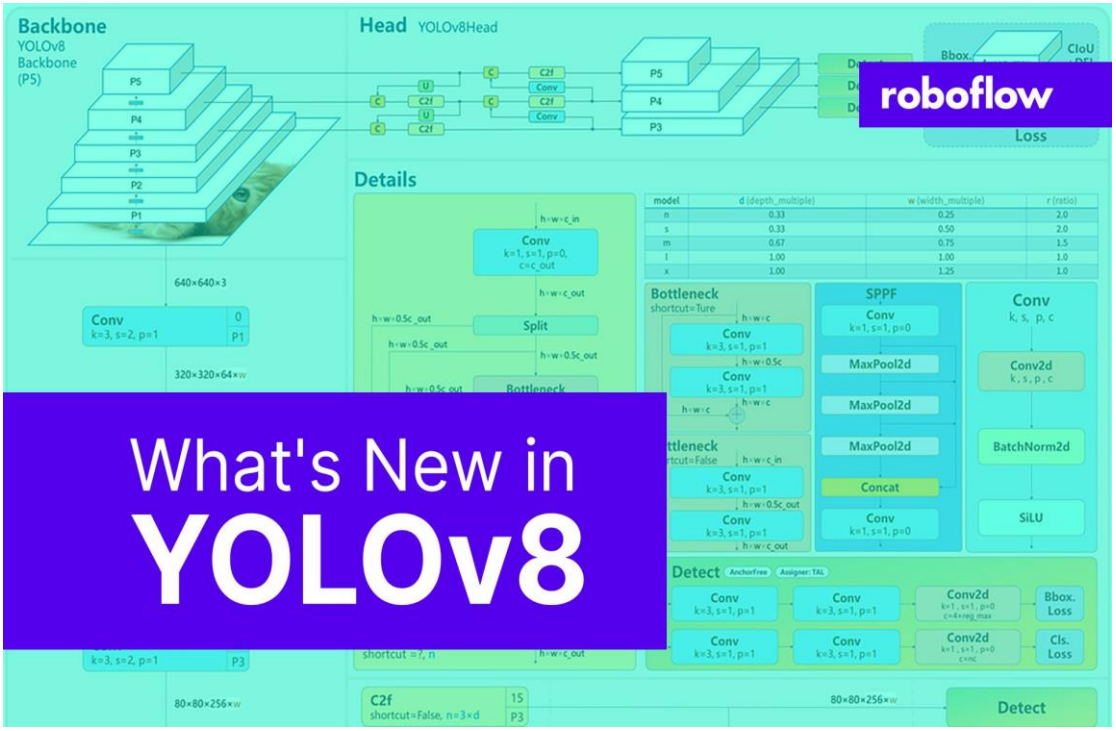


Figure 5.7 Yolov8

5.3.3 Google Colab

Google Colab is a cloud-based development environment that offers a Jupiter Notebook-like interface for writing and executing Python code. It provides pre-installed libraries, GPU support, and seamless collaboration, allowing users to work on projects, share notebooks, and edit code collaboratively. With access to Google's powerful cloud infrastructure, Colab eliminates the need for local setup and provides convenient storage and data access through Google Drive. It is a user-friendly and versatile platform for Python programming, data analysis, and machine learning tasks.

Figure 5.8 Google Colab

5.3.4 Google Vision API

Google Vision API is a powerful cloud-based service provided by Google that allows developers to integrate image recognition and analysis capabilities into their applications. It utilizes advanced machine learning models to identify and classify objects, detect faces and emotions, extract text from images, and perform various image-based tasks. By leveraging the Vision API, developers can automate image processing tasks, improve user experiences, and unlock new possibilities in fields like e-commerce, content moderation, and visual search. The API provides an easy-to-use interface and supports multiple programming languages, making it accessible to developers with varying levels of expertise.

Figure 5.9 Google Api

5.4 Prototype

In Google Colab, we used the `ipywidgets` and `IPython.display` packages to create a graphical user interface (GUI). The `ipywidgets` package provides interactive HTML widgets for GUI components like buttons and sliders, which can respond to user input and update dynamically. The `IPython.display` package's `display` function was used to show the GUI components and the `clear_output` function helped in refreshing the display area for updated content. This combination allowed us to create an interactive GUI in Google Colab, enabling users to interact with the code and visualize real-time output.

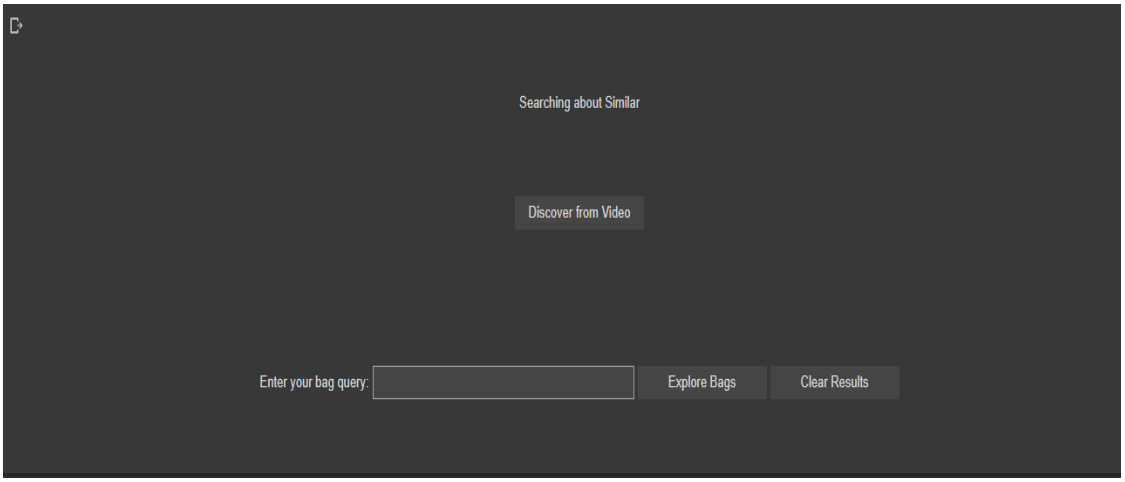


Figure 5.10.1 GUI in Google Colab

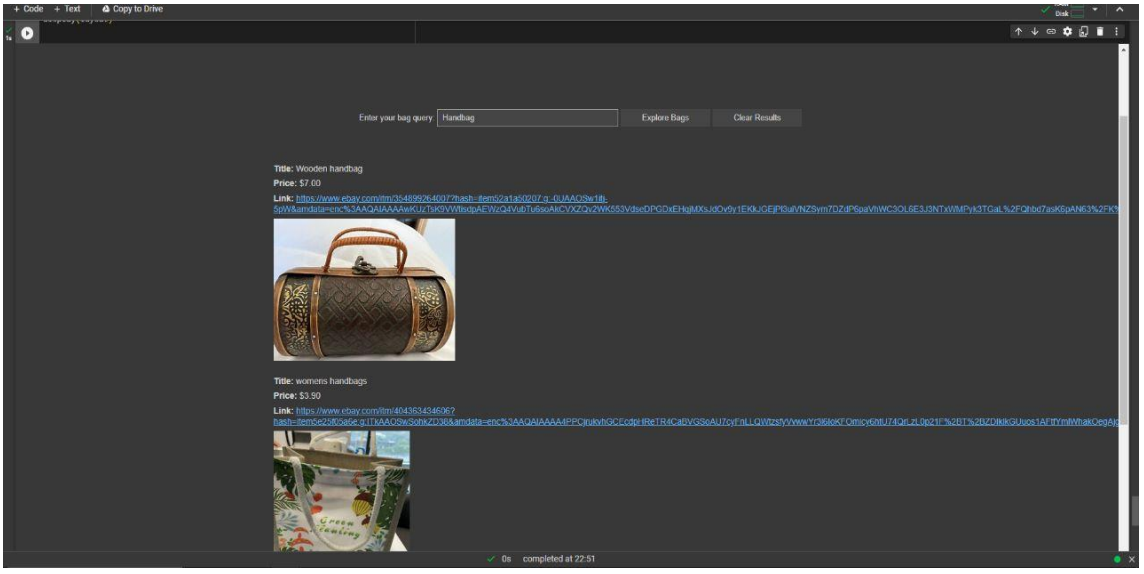


Figure 5.10.2 GUI in Google Colab

Chapter 6

Implementation

6.1 Implementation Environment

The Detector-X platform aims to auto-detect and search across the internet to find the exact match or the similarities of the items detected, to ease the process for the user and make it as simple as it gets.

1. The user first goes in the application, then uploads a video of a required item, but the object must be visible in the frames for the detection process to be successful.

2. After uploading the video successfully, our detection model begins to work, specifically Yolov8, detect the class of the object given.

3. After detecting the object, in our case a (handbag, suitcase, backpack), it goes to the 2nd part of the system, which is to classify the brand.

4. We used Google Vision Api, as it gives the best accurate classification, and is quite fast in the retrieval of the extracted features.

5. 2nd to last step, we gather all the extracted features, and the detected object name to build a custom search query.

6. Last step, is retrieval of the results of the query and show the output to the user.

Here's a breakdown of the technologies used:

1. **Python:** Python is a widely-used programming language, and it's commonly used for machine learning and computer vision tasks due to its extensive libraries and frameworks.

2. **Google Colab:** Google Colab is a cloud-based Jupiter notebook environment provided by Google. It allows you to run Python code and provides access to GPUs, which is beneficial for training deep learning models.

3. **YOLOv8:** YOLOv8 is an object detection algorithm that stands for "You Only Look Once." It is a popular and efficient real-time object detection model that can detect multiple objects within an image.

4. **Google Vision API:** Google Vision API is a cloud-based service that provides pre-trained models for various computer vision tasks, including object detection, image labeling, and text recognition. It can be integrated into Python code to leverage its capabilities.

5. **PyTorch:** PyTorch is a deep learning framework that provides tools and modules for building and training neural networks. It offers a dynamic computational graph, making it flexible and suitable for research purposes.

6. **OpenCV:** OpenCV (Open-Source Computer Vision Library) is a widely-used computer vision library that provides tools and algorithms for image and video processing tasks. It offers functions for reading and manipulating images, as well as implementing computer vision algorithms.

7. **NumPy:** NumPy is a Python library that provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions. It's commonly used for numerical computations and data manipulation in scientific computing.

8. **PIL (Python Imaging Library):** PIL is a library for image processing tasks in Python. It provides functions for opening, manipulating, and saving various image file formats.

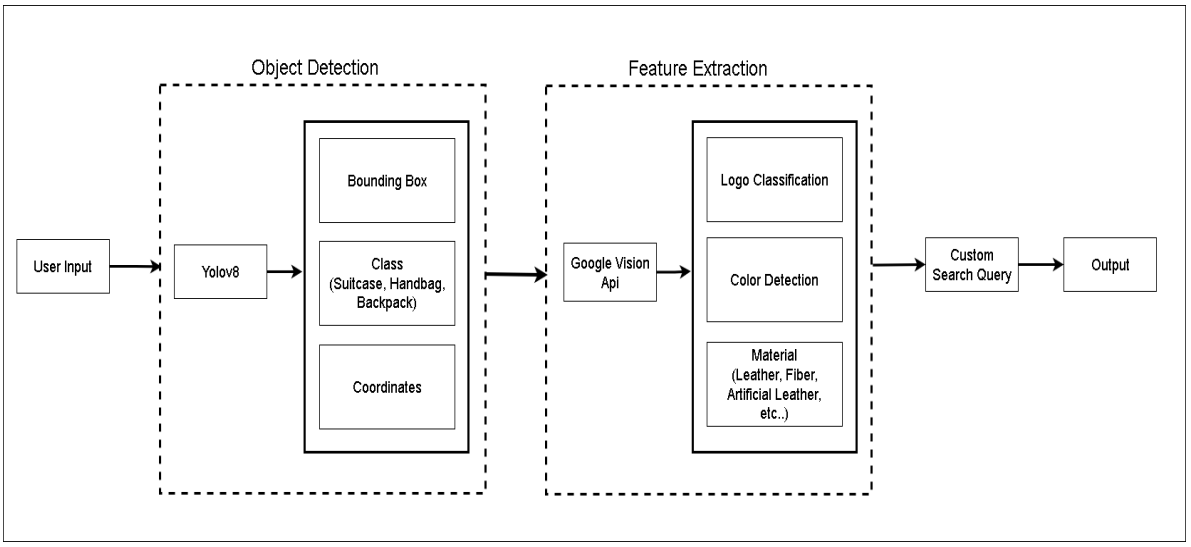


Figure 6.1 proposed model

6.2 Results and Discussion

The system implemented in this project combines YOLOv8 for object detection with the accuracy of the Google Vision API for feature extraction. YOLOv8 achieved a mean Average Precision (mAP) of 53.8%, indicating reasonably good object detection performance. The Google Vision API, known for its accuracy ranging from 80% to 90%, provides reliable results for feature extraction. The system leverages the extracted features to build custom queries, enabling internet searches for similarities or exact matches. This integration enhances the system's capabilities, offering robust object detection, accurate feature extraction, and the ability to explore related instances of detected objects on the internet. Continuous evaluation and refinement can further improve the system's performance and adapt it to specific use cases.

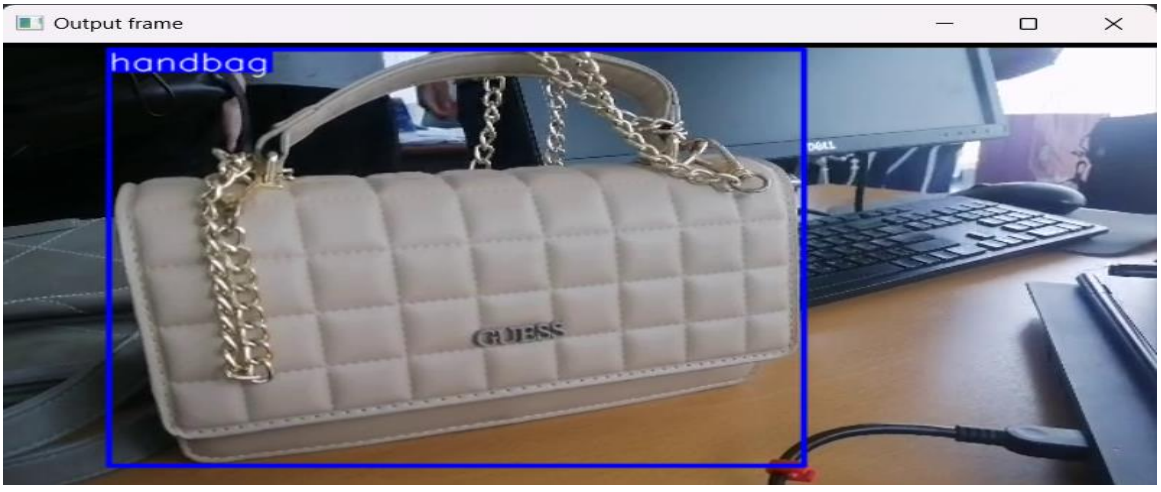


Figure 6.2 detect type of bag

Chapter 7

Testing

Software testing is the process of evaluating software and to ensure that it meets the required standards. There are different standards and tests for software testing, but they all share the same goal, that being highlighting failures in software and ensuring that software performs as intended.

7.1 Model Testing

During the testing phase of the YOLOv8 model, which achieved a known mean Average Precision (mAP) of 53.8%, the model demonstrated accurate object detection by correctly identifying each object. Additionally, the model exhibited efficient performance, completing the detection process quickly. The combination of accurate detection and fast processing makes the YOLOv8 model a suitable choice for real-time or time-sensitive applications where both accuracy and speed are crucial

Model	Size (pixels)	mAP^{val}_{50-95}	Speed CPU ONNX (ms)	Speed A100 Tenser RT (ms)	Params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Table 7.1 Performance of yolov8

7.2 Feature extraction Testing

During the testing phase, the Google Vision API model demonstrated accurate results, providing reliable object detection and extracting useful features for the detected objects. We evaluated the accuracy of the model and found it to perform well in identifying objects with a high level of precision. The extracted features were found to be informative and relevant to the detected objects, enhancing the overall effectiveness of the system. The accurate results and extracted features obtained from the Google Vision API model validate its suitability for object detection and highlight its potential for various applications.

7.3 System Testing

During system testing, the proposed model effectively combined the Google Vision API and YOLOv8, demonstrating accurate object detection and valuable feature extraction. The Google Vision API provided precise results and extracted meaningful features, while YOLOv8 achieved a 53.8% mAP and exhibited efficient processing speed. The seamless integration and collaboration between these components were verified through system testing, where the model successfully built custom queries and redirected users to the output screen after completing all tasks, ensuring a streamlined user experience.

Chapter 8

Conclusion and Future Work

In conclusion, the system leveraging YOLOv8 and Google Vision API for object detection and feature extraction, along with a custom query-based internet search, offers a powerful solution. With YOLOv8 achieving a mAP of 53.8 and the Google Vision API providing accuracy levels ranging from 80% to 90%, the system enables accurate detection and detailed feature extraction from objects. By constructing custom queries, it facilitates searching the internet for similarities or exact matches, enhancing the understanding of the detected objects. This integrated approach holds promise for a wide range of applications, such as security surveillance, autonomous vehicles, and content moderation, where precise object detection, feature extraction, and contextual information retrieval are essential.

For future work, we have several exciting directions to enhance our system. Firstly, we plan to develop our own custom CNN model to have more control and customization, potentially improving accuracy compared to relying solely on the Google Vision API. Additionally, we aim to create a Flutter app, enabling our system to be available across different platforms such as iOS, Android, PC, and Mac, thereby expanding its accessibility to a wider user base. Customizing the YOLOv8 model to achieve a better mean Average Precision (mAP) is another goal, as it would significantly enhance our system's performance. Moreover, we intend to establish our own database and expand beyond handbags and backpacks to include other fashion items, allowing users to search for information on a diverse range of fashion products. These future enhancements will empower us to deliver a more accurate, versatile, and comprehensive solution, providing an enhanced user experience for individuals seeking information about various fashion items

Chapter 9

Bibliography

1. IQ.opengenus.org. (n.d.). YOLOv5. Retrieved from https://iq.opengenus.org/yolov5/

2. Ultralytics. (n.d.). YOLOv8. Retrieved from https://ultralytics.com/yolov8

3. Ultralytics. (n.d.). YOLOv8 Documentation. Retrieved from https://docs.ultralytics.com/yolov5/

4. Labellerr. (n.d.). Evolution of YOLO Object Detection Model from v5 to v8. Retrieved from https://www.labellerr.com/blog/evolution-of-yolo-object-detection-model-from-v5-to-v8/

5. Learn OpenCV. (n.d.). Ultralytics YOLOv8. Retrieved from https://learnopencv.com/ultralytics-yolov8/

6. Roboflow Blog. (n.d.). What's New in YOLOv8? Retrieved from https://blog.roboflow.com/whats-new-in-yolov8/

7. Tsang, S. H. (n.d.). Paper: CutMix - Regularization Strategy to Train Strong Classifiers with Localizable Features. Retrieved from https://sh-tsang.medium.com/paper-cutmix-regularization-strategy-to-train-strong-classifiers-with-localizable-features-5527e29c4890

8. Aisent.io. (2023, February 6). YOLO: Real-Time Object Detection Explained. Retrieved from https://aisent.io/en/blog/2023-02-06/yolo/

9. Wang, J., Chen, Y., Gao, M., & Dong, Z. Improved YOLOv5 network for real-time multi-scale traffic sign detection. Retrieved from https://arxiv.org/abs/2112.08782

10. Jung, H.-K., & Choi, G.-S. (2022). Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions. Applied Sciences, 12(14), 7255. https://doi.org/10.3390/app1214725

11. Aboah, A., Wang, B., Bagci, U., & Adu-Gyamfi, Y. (2023). Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8. Retrieved from https://arxiv.org/abs/2304.08256

Senior Project Summary Report

Project Title	Real-time detection and searching system	
Supervisor(s)	Assoc. Prof. Mohammed Mostafa Fouad Assoc. Prof. Essam Elafkarany	
Team members:	Names: Alaa Bassem Ahmed Elsayed Ahmed Hassan Mohammed Salah Nour Abdelkader Salma Ayman	Registration Numbers: 19105659 19107947 19107898 18107286 19107473 19107447
Project Deliverables	**System Requirements Specification:** A document that outlines the functional and non-functional requirements of the system. **Software Implementation:** The actual implementation of the system. **Testing and Quality Assurance:** This involves conducting various tests to ensure the system functions as intended.	
Team Organization	S.E Group: • Ahmed Hassan • Ahmed Elsayed • Salma Ayman C.S Group: • Alaa Bassam • Mohammed Salah • Nour Abdelkader Each member in the team approached problem evaluation by drawing upon their specialized knowledge and skills in their respective domains.	
Ethical Considerations	User data should be protected Responsible deployment User Control and Autonomy	
Social Impact	Efficiency in E-commerce and Retail Reduce time wasted in detection and searching	
Professional Responsibility	The System is able to Perform functions without any failures. Conduct multiple tests and handle all cases Prioritize security	

Supervisor Name	**Signature**
Assoc. Prof. Mohammed Mostafa Fouad	
Assoc. Prof. Essam Elafkarany	