SILIGURI INSTITUTE OF TECHNOLOGY

**PROJ- CS881**

##### **Comparative analysis of popular and publicly available datasets for object detection in challenging environment**

##### **BY**

##### **CSE\_PROJ\_2023\_10**

|  |  |
| --- | --- |
| Name of Students | Roll No. |
| 1. Soumya Majumder | 11900119073 |
| 1. Sayantan Bhowmick | 11900119078 |
| 1. Sayani Ghosh | 11900119047 |
| 1. Suvaditya Gupta | 11900119025 |

**Under the Guidance**

**of**

**Prof. Dr. Uddalak Mitra**

Submitted to the Department of **Computer Science & Engineering** in partial fulfillment of the requirements for the award of the degree Bachelor of Technology in **Computer Science & Engineering.**

**Year of Submission: 2023**



**Siliguri Institute of Technology**

**P.O. SUKNA, SILIGURI, DIST. DARJEELING, PIN:734009**

**Tel: (0353)2778002/04, Fax: (0353) 2778003**

**DECLARATION**

This is to certify that Report entitled “**Comparative analysis of popular and publicly available datasets for object detection in challenging environment”** which is submitted by me in partial fulfillment of the requirement for the award of degree B.Tech. in **Computer Science Engineering** at **Siliguri Institute of Technology** under **Maulana Abul Kalam Azad University of Technology**, West Bengal. We took the help of other materials in our dissertation which have been properly acknowledged. This report has not been submitted to any other Institute for the award of any other degree.

Date:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Name of the Student | Roll No | Signature |
| 1 | Soumya Majumder | 11900119073 |  |
| 2 | Sayantan Bhowmick | 11900119078 |  |
| 3 | Sayani Ghosh | 11900119047 |  |
| 4 | Suvaditya Gupta | 11900119025 |  |

**CERTIFICATE**

This is to certify that the project report entitled **Comparative analysis of popular and publicly available datasets for object detection in challenging environment** submitted to **the Department of Computer Science & Engineering of Siliguri Institute of Technology** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science & Engineering** during the academic year **2022-23,** is a bona fide record of the project work carried out by them under my guidance and supervision.

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Group Number:** CSE\_PROJ\_2023\_10 | | | |
| **SN** | **Name of the students** | **Registration No** | **Roll No** |
|  | Soumya Majumder | 035911 | 11900119073 |
|  | Sayantan Bhowmick | 035880 | 11900119078 |
|  | Sayani Ghosh | 036187 | 11900119047 |
|  | Suvaditya Gupta | 036430 | 11900119025 |

**------------------------------------------**

**Signature of the HOD**

**Department of Computer Science & Engineering**

**---------------------------------------**

**Signature of Project Guide**

**Name of the Guide: Dr. Uddalak Mitra**

**Acknowledgement**

We would like to express our heartfelt gratitude to all those who played a significant role in the successful completion of our group project "**Comparative analysis of popular and publicly available datasets for object detection in a challenging environment**". Without their valuable contributions, this endeavor would not have been possible.

First and foremost, we extend our deepest appreciation to our project supervisor, **Dr. Uddalak Mitra**. Their unwavering support, insightful guidance, and constructive feedback were instrumental in shaping our project and enhancing its quality. Their expertise and encouragement motivated us to push the boundaries and strive for excellence. We are truly grateful for their mentorship throughout this journey.

Furthermore, we would like to acknowledge the faculty members at **Siliguri Institute of Technology**. Their dedication to academic excellence and commitment to providing us with the necessary resources and facilities greatly facilitated our project. We are grateful for their tireless efforts in creating a conducive learning environment and encouraging us to pursue innovative projects.

Additionally, we would like to thank our classmates who actively participated in our project. Their collective efforts, valuable insights, and collaborative spirit significantly contributed to the overall success of the project. The countless hours spent discussing ideas, sharing knowledge, and working together as a team were invaluable, and we appreciate their commitment and enthusiasm.

Finally, we would like to express our gratitude to our friends and family members who supported us throughout this project. Their encouragement, understanding, and patience were vital in keeping us motivated and focused during challenging times.

Signature of all the group members with the date

1.

2.

3.

4.

Table of Contents

Introduction8

System Analysis8

Identification of need 8

Preliminary Investigation 8

Feasibility Study 13

Project Planning 15

Project Scheduling 16

Software Requirement Specification 17

Software Engineering Paradigm applied19

System Design19

Modularization details 19

About Datasets 21

User Interface Design 24

Coding 25

Testing29

Testing Report 30

Inference Analysis 35

Cost Estimation 41

PERT Chart 42

Conclusion and recommendation42

Reference and appendices44

**List of Tables and Figures**

Table 1: Comparison of various object detection datasets used in this study

21

Table 2: Comparison of all models based on mAP@0.5 and mAP@0.5:0.95

35

Fig 1: Overview of the project Timeline 16

Fig 2: Sample of OpenImage 22

Fig 3: Sample of OpenImage 22

Fig 4: Sample of Microsoft COCO 22

Fig 5: Sample of Microsoft COCO 22

Fig 6: Sample of Person-v1 23

Fig 7: Sample of Person-v1 23

Fig 8: Sample of Person-v2 24

Fig 9: Sample of Person-v2 24

Fig 10: Download code snippet 26

Fig 11: Convert Code Snippet 26

Fig 12:Exporting the dataset 27

Fig 13: Choosing the Export format 27

Fig 14: Download Initiation 28

Fig 15: f1 curve of COCO model 30

Fig 16: f1 curve of OpenImage 30

Fig 17: f1 curve of Person-v1 31

Fig 18: f1 curve of Person-v2 31

Fig 19: f1 curve of EnsembleX4 31

Fig 20: f1 curve of EnsembleX2 31

Fig 21: P curve of COCO model 31

Fig 22: P curve of OpenImage 31

Fig 23: P curve of Person-v1 32

Fig 24: P curve of Person-v2 32

Fig 25: P curve of EnsembleX4 32

Fig 26: P curve of EnsembleX2 32

Fig 27: R curve of COCO model 32

Fig 28: R curve of OpenImage 32

Fig 29: R curve of Person-v1 33

Fig 30: R curve of Person-v2 33

Fig 31: R curve of EnsembleX4 33

Fig 32: R curve of EnsembleX2 33

Fig 33: PR curve of COCO 33

Fig 34: PR curve of OpenImage 33

Fig 35: PR curve of Person-v1 34

Fig 36: PR curve of Person-v2 34

Fig 37: PR curve of EnsembleX4 34

Fig 38: PR curve of EnsembleX2 34

Fig 39: Detection using the COCO model 35

Fig 40: Detection using OpenImage 35

Fig 41: Detection using the Person-v1 model 36

Fig 42: Detection using the Person-v2 model 36

Fig 43: Detection using EnsembleX4 36

Fig 44: Detection using EnsembleX2 36

Fig 45,46: Underwater detection of COCO 37

Fig 47: Low visibility detection of COCO 37

Fig 48,49: Underwater detection of OpenImage 38

Fig 50: Low visibility detection of OpenImage 38

Fig 51,52: Underwater detection of Person-v1 38

Fig 53: Low visibility detection of Person-v1 39

Fig 54,55: Underwater detection of Person-v2 39

Fig 56: Low visibility detection of Person-v2 39

Fig 57,58: Underwater detection of EnsembleX4 40

Fig 59: Low visibility detection of EnsembleX4 40

Fig 60,61: Underwater detection of EnsembleX2 41

Fig 62: Low visibility detection of EnsembleX2 41

Fig 63: Pert Chart 42

**Abstract:**

This project aims to develop an efficient object detection system for challenging environments using the YOLOv7 model. The main objective is to enhance the accuracy and effectiveness of object identification and tracking in various scenarios, including low visibility conditions, underwater environments, and disaster zones. The project utilizes open-source tools and datasets to train and evaluate the performance of the YOLOv7 model.

The project begins with a preliminary investigation to identify the specific challenges and requirements of object detection in challenging environments. Extensive research is conducted to gather information about existing object detection algorithms, and the YOLOv7 model is selected for its efficiency and robustness. The project leverages Google Colab for model training and testing, ensuring easy accessibility and computational resources.

A custom dataset has been created, consisting of diverse images from different sources, including Google Open Images, MS COCO, person-v1, person-v2, and self-collected images. The dataset has been annotated to provide ground truth labels for evaluation purposes. The YOLOv7 models were extensively tested using this dataset to evaluate their performance in challenging environments.

The results demonstrate the effectiveness of the YOLOv7 model in detecting and tracking objects in challenging scenarios. The model shows promising results in low visibility conditions, underwater environments, and disaster zones, enabling applications in search and rescue operations and military operations. Additionally, ensembled models are created to further improve the accuracy and robustness of the object detection system.

Overall, this project provides valuable insights and contributions to the field of object detection in challenging environments. The developed system offers enhanced capabilities for identifying and tracking objects, enabling efficient and effective operations in various domains. The findings of this project have significant implications for rescue operations, military applications, and other fields that require accurate object detection in challenging conditions.

**Keywords**: object detection, deep learning, YOLOv7, challenging environments, ensemble learning classifier.

1. **Introduction**

**1.1 System Analysis**

**1.1.1 Identification of Need**

In challenging environments such as disaster zones, rugged terrain, or harsh weather conditions, there may be a critical need for object detection [1] to support search and rescue operations, environmental monitoring, or infrastructure inspection. The identification of the need for an object detection project in such environments requires a deep understanding of the specific challenges and constraints that need to be addressed. For example, in disaster zones, it may be necessary to identify and locate survivors amidst the rubble and debris of collapsed buildings. In such scenarios, the use of object identification technology can help to detect and track signs of life, such as movement or body heat. In remote or inaccessible areas, object identification can be used to monitor wildlife populations or track environmental changes. Therefore, the identification of a need for an object detection project in challenging environments requires a careful analysis of the problem at hand and the potential benefits that can be achieved through the use of technology. Additionally, the project must be designed and implemented to operate reliably in the harsh conditions of the environment, with appropriate measures for safety and security.

1. Top of Form

**1.1.2 Preliminary Investigation**

The preliminary investigation of a project on object identification in challenging environments [2] typically involves the following steps:

1. Problem identification: We identified the specific problem of detecting persons in challenging environments such as foggy, underwater, and dark conditions. The objective was to develop an efficient and accurate object detection system that could be used in rescue operations.
2. Feasibility assessment: Determine the feasibility of implementing an object identification system. This involves considering factors such as the available technology and resources, the complexity of the problem, and the environmental conditions.
3. Information gathering: To begin, we conducted thorough research to understand the current state of object detection algorithms [3] and their limitations. We explored various algorithms and techniques available in the literature, evaluating their performance in challenging environments. After careful analysis, we identified YOLOv7 [4] as the most suitable algorithm for our project due to its efficiency and effectiveness. We dived into an in-depth study of YOLOv7, exploring its architecture, underlying principles, and implementation details. This research enabled us to gain a comprehensive understanding of how YOLOv7 operates and its potential applicability in challenging environments. Armed with this knowledge, we proceeded to plan and structure our project accordingly. In addition to algorithmic research, we also focused on gathering information about the datasets required for training and evaluation. We extensively explored available datasets in the open source, assessing their suitability for our project's objectives. We identified and selected appropriate datasets that encompassed a diverse range of challenging environmental conditions, including dark, foggy, and underwater scenarios. This ensured that our models were trained and evaluated on data representative of real-world challenges. Furthermore, we researched to gather information about the infrastructure and tools required for our project and their availability. We explored different platforms and frameworks that could support the implementation of YOLOv7 and facilitate the training and testing processes. This included investigating cloud-based platforms, such as Google Colab, and version control systems like GitHub, to ensure seamless collaboration and efficient project management. By extensively gathering information about object detection weaknesses, the YOLOv7 algorithm, suitable datasets, and the necessary infrastructure, we laid a solid foundation for our project. This comprehensive information-gathering process enabled us to make informed decisions and develop a robust plan to address the challenges associated with object detection in challenging environments.
4. Benefit and risk analysis:

To ensure a comprehensive evaluation of our project, we conducted a thorough benefit and risk analysis. This analysis allowed us to assess the potential advantages and drawbacks associated with implementing our object detection model in challenging environments. We focused on three key considerations: technical aspects, operational implications, and financial factors.

4.1. Technical Considerations: By implementing an object detection algorithm, we aimed to achieve several technical benefits. These included improved accuracy and efficiency in detecting objects in challenging environments, which can be critical in scenarios such as rescue operations and surveillance tasks. Additionally, our system aimed to overcome the limitations of existing detection algorithms by specifically addressing challenges like low visibility and lighting conditions, foggy weather, and underwater conditions. By leveraging the YOLOv7 algorithm, we anticipated achieving a higher detection rate with minimized false positives, leading to enhanced reliability and performance in challenging environments.

4.2. Operational Implications: The successful implementation of our object detection model had significant operational implications. It would enable more efficient and effective detection and tracking of objects, particularly in challenging environments. This could streamline various operations, such as search and rescue missions and security surveillance. By automating the detection process, our system would reduce human effort and enhance operational productivity. Moreover, it would enable real-time object identification, providing timely information for decision-making and enabling proactive response in critical situations.

4.3. Financial Considerations: We conducted a detailed financial analysis to assess the costs to implement our object identification system. This analysis encompassed factors such as hardware and software requirements, dataset acquisition, and computational resources. We considered the feasibility of utilizing cloud-based platforms, such as Google Colab [5] or Amazon Sagemaker that provide free computational resources but with limitations and open source software, such as Fiftyone, to keep costs within our capabilities. Additionally, successful implementation of the project can lead to potential financial benefits arising from the improved efficiency and accuracy of the model, which could lead to cost savings in various domains, including search and rescue operations and surveillance.

Throughout the benefit and risk analysis, we also identified potential risks and challenges. These included technical complexities during the implementation phase, potential issues with dataset quality and diversity, and the limitations of the algorithm and object detection model.

Overall, the benefit and risk analysis highlighted the significant potential benefits of implementing our object detection model, including improved accuracy, enhanced operational efficiency, and potential cost savings. By recognizing the associated risks and challenges, we were able to devise strategies to mitigate these risks and ensure the successful implementation of the system.

1. Stakeholder analysis:

To ensure the successful implementation of our object identification system, we conducted a stakeholder analysis to identify the key individuals and organizations that would be involved or affected by the project. This analysis allowed us to understand their needs, expectations, roles, responsibilities, and the potential impact they could have on the project's outcome.

5.1. Stakeholders Identified: Based on our analysis, the following stakeholders were identified:

a) Research Institutions: Academic and research institutions specializing in computer vision, object detection, and artificial intelligence. These stakeholders are interested in advancing the field of object identification in challenging environments and may contribute to the project through collaboration, research support, knowledge, and resource sharing.

b) Rescue and Emergency Services: Stakeholders involved in search and rescue operations, disaster management, and emergency response. They have a significant interest in accurate and efficient object identification systems that can aid in locating individuals in challenging environments. Their requirements may include real-time detection, robustness in adverse conditions, and integration with existing rescue technologies.

c) Government Agencies: Government organizations responsible for public safety, security, and surveillance. These stakeholders may have regulatory requirements and guidelines that need to be considered during the implementation of the object identification system. They also play a crucial role in funding, policy support, and adoption of such technologies for public use.

d) Military Application: Our object detection model holds significant potential for military applications. It can play a crucial role in enhancing military operations by enabling real-time detection and tracking of enemy combatants and identifying suspicious activities. The model's ability to provide valuable insights into enemy positions contributes to improved situational awareness and aids in mission planning. Additionally, it can be utilized in search and rescue operations, assisting in locating missing personnel and ensuring their timely recovery. Furthermore, the model can enhance border security efforts by detecting unauthorized border crossings and identifying smuggling activities. By integrating our model into military systems, we can strengthen national security measures and enable more effective and efficient military operations.

5.2. Needs and Expectations: Through consultations, interviews, and discussions with the stakeholders, we identified their common needs and expectations:

Accurate and Reliable Detection: Stakeholders expect the object detection system to provide accurate and reliable detection results in challenging environments, including low visibility, foggy weather, and underwater conditions.

Real-time Performance: There is a need for real-time or near real-time object detection capabilities to enable timely decision-making and response in critical situations.

Adaptability and Flexibility: Stakeholders expect the system to be adaptable and flexible, allowing for customization based on specific environmental conditions, target objects, and deployment scenarios.

Integration and Compatibility: The system should be compatible with existing rescue, monitoring, or surveillance technologies, allowing for seamless integration into the stakeholders' operational workflows.

User-Friendly Interface: Stakeholders require an intuitive and user-friendly interface that enables easy system operation, configuration, and result interpretation, even for non-technical users.

5.3. Stakeholder Impact: The stakeholders identified can have a significant impact on the project's success. Research institutions can contribute expertise, provide resources and feedback on algorithmic advancements, and validate the system's performance through rigorous evaluation. Rescue and emergency services stakeholders play a critical role in field testing, providing real-world feedback, and validating the system's effectiveness in challenging environments. Government agencies can influence policy, regulations, and funding decisions, which can significantly impact the system's deployment and adoption. Technology providers can offer technical support, integration solutions, and access to cutting-edge technologies to enhance the system's capabilities.

By considering the needs, expectations, roles, and responsibilities of the identified stakeholders, we can ensure that our object identification system meets their requirements and aligns with their objectives

The findings of the preliminary investigation will provide the foundation for the project plan, including the scope, budget, and timeline. A well-conducted preliminary investigation can help ensure that the project is feasible, well-informed, and aligned with stakeholder needs and expectations.

Top of Form

**1.1.3 Feasibility Study**

YOLOv7 is an object detection algorithm that has gained popularity in recent years due to its high accuracy and efficiency. However, training multiple YOLOv7 models can be resource-intensive and require significant computational power. In this report, we will assess the feasibility of training four YOLOv7 models based on resource constraints.

Hardware Requirements:

A study of required Hardware specifications is necessary for conducting the Feasibility study. Training for multiple YOLOv7 models requires significant computational resources, including a high-end GPU, sufficient RAM, and fast storage. A GPU with at least 16GB of VRAM is recommended for training YOLOv7, and a CPU with at least 8 cores is ideal. Additionally, the system should have at least 32GB of RAM, and preferably an SSD with high read and write speeds.

Feasibility Assessment:

The feasibility of training four YOLOv7 models depends on several factors, including the size of the dataset, the number of classes, and the number of epochs required for convergence. For a small dataset with a limited number of classes, it may be feasible to train four models simultaneously on a single GPU. However, for larger datasets with many classes, training multiple models simultaneously may require more GPUs or distributed training methods.

Another factor that could impact the feasibility of training multiple YOLOv7 models is the number of epochs required for convergence. If each model requires a large number of epochs to converge, training multiple models simultaneously may not be feasible due to the extended training time required. In this case, it may be necessary to reduce the number of epochs or use transfer learning to reduce the training time and resource requirements.

Feasible Specification for our Project:

For our project, we used both Google Colab Pro and Colab free plans for training our models. We subscribed to Colab Pro, which costs $11.79 per month and provides options to select Hardware Accelerators [6], either GPUs (Graphics Processing Units) or TPUs (Tensor Processing Units). For GPUs, options available were NVIDIA T4, NVIDIA A100, and NVIDIA V100.

We used Colab Pro and free plans in a hybrid manner to minimize costs. In the free plan, only the NVIDIA T4 GPU was available, while in the Pro plan, either the NVIDIA T4 or V100 was available. For System RAM, in the free plan, we were restricted to just 12.7GB of usable RAM, whereas in the Pro plan with the "HIGH RAM" runtime shape selected, we had 25.5 GB of usable RAM.

Each session of Colab with Hardware Accelerators was restricted to approximately 6-8 hours of continuous usage. Colab has inbuilt integration with Google Drive. Due to Drive being restricted to 15 GB for free, we subscribed to Google One monthly plans which extended the storage capacity to 100 GB.

In conclusion, the feasibility of training four YOLOv7 models depends on various factors such as dataset size, class number, and required epochs for convergence. For our project, we utilized a hybrid approach by using both Colab Pro and free plans and selecting hardware accelerators like GPUs and TPUs. We also subscribed to Google One to extend our storage capacity. Although each Colab session was restricted to approximately 6-8 hours, we managed to train our models effectively by making the most of the resources available. Overall, we completed our project by optimizing the available resources and implementing an effective training strategy.

**1.1.4 Project Planning**

When planning a project on Object detection in challenging environments, it is important to consider several key factors. Here's a detailed breakdown of our project planning:

1. Project Objectives: The primary objective of the project was to develop a robust object detection model that can accurately detect a person in challenging environments such as underwater, foggy weather, and low lighting condition. The project aimed to achieve this objective by training and testing multiple YOLOv7 models on five different datasets, four for training the models and one for testing, and making an ensembled supermodel to improve the accuracy and efficiency.

2. Scope: The project's scope was limited to the detection of a single class (Person) in challenging environments. The project did not explore other classes or environmental conditions due to resource restrictions.

3. Deliverables:

The primary objective of our project was to develop and deliver a comprehensive solution utilizing the YOLOv7 model for object detection tasks. As part of this endeavor, we aimed to provide the following deliverables to the users and customers:

**Trained YOLOv7 Models**: We have successfully trained multiple YOLOv7 models using diverse datasets. These models are specifically designed for object detection in challenging environments. Each model is capable of detecting and localizing various objects with high accuracy.

**Custom Dataset for Testing**: To ensure robust evaluation and validation of our models, we have curated a custom dataset consisting of diverse images and videos. This dataset includes challenging scenarios such as low visibility, underwater conditions, and partial object visibility. It serves as a reliable benchmark for testing the performance of our models. This dataset can also be used for training purposes

**Ensembled Super Models:** In addition to individual models, we have developed ensembled supermodels [7] that combine the strengths of multiple YOLOv7 models. These ensembled models exhibit enhanced accuracy and reliability in object detection tasks. By leveraging the collective intelligence of multiple models, they provide improved results and increased confidence in the detection outcomes.

These deliverables collectively provide users and customers with a powerful and reliable object detection solution based on the YOLOv7 model. Our trained models, custom datasets, and ensembled supermodels empower users to effectively identify and track objects of interest in challenging environments.

4. Timelines: The project timeline was determined based on the availability of resources and the complexity of the task. The team allocated 1 week for collecting and pre-processing the datasets, several months for training and testing the individual models, and additional time for ensembling the models and testing on custom datasets.

5. Resources: The project required various resources, including hardware (e.g., GPUs), cloud platform (e.g., Google Colab), Software(e.g., FiftyOne, Visual Studio Code), and personnel (e.g., data annotators and developers). The team also had to allocate resources for data collection, pre-processing, training, testing, and ensembling.

6. Budget: The project budget was determined based on the resources required and the timeline for completing the project. The team had to consider the costs of resources needed and any other expenses related to data collection and pre-processing.

Overall, the project planning was well thought out and considered various factors that could impact the success of the project. The report provided a clear summary of the findings, including the strengths and weaknesses of each model and the benefits of ensembling them to improve accuracy and efficiency. The application of our project in rescue and military operations represents a significant and valuable contribution to the respective fields. Top of Form

**1.1.5 Project Scheduling**

Fig 1: Overview of the project timeline

**1.1.6 Software Requirement Specifications (SRS)**

Software Requirement Specification (SRS) for an Object Identification project in a challenging environment:

1. Introduction: The software is designed to identify objects in a challenging environment using image processing and machine learning techniques. The software will be used in various industries, including construction, engineering, and security.
2. Scope: The software will process images captured in challenging environments with low lighting, cluttered backgrounds, or occlusions. The software will extract features from the images, detect objects of interest, and classify them into predefined categories.
3. Functional Requirements: The software must perform the following functions:

* Image preprocessing to enhance image quality and remove noise
* Feature extraction to identify distinctive features in the images
* Object detection to identify objects of interest in the images
* Object classification to categorize the objects based on predefined categories
* The user interface displays the results of the object identification process

1. Non-functional Requirements: The software must meet the following non-functional requirements:

* Performance: The software should be able to process images in real-time or near real-time. In the case of our benchmark dataset (MS COCO), the performance recorded was with batch 1 fps of 161 fps, and the batch 32 average time was 2.8ms [8]
* Usability: The software should be easy to use and require minimal training.
* Security: The software should protect sensitive data and prevent unauthorized access.
* Reliability: The software should be reliable and operate without errors or crashes.
* Compatibility: The software should be compatible with different operating systems and hardware configurations.

1. System Architecture: The software will be developed using Python and deep learning frameworks such as TensorFlow or PyTorch and requires the following packages for basic functionality:

matplotlib >= 3.2.2

numpy>= 1.18.5, < 1.24.0

opencv-python >= 4.1.1

Pillow >= 7.1.2

PyYAML>= 5.3.1

requests >= 2.23.0

scipy>= 1.4.1

torch>=1.7.0,!=1.12.0

torchvision>=0.8.1,!=0.13.0

tqdm>=4.41.0

protobuf<4.21.3Top of Form

1. Data Management: The software will store and manage images and object categories in a database hosted in the cloud. The database will be secured using encryption and access control measures.
2. Testing and Validation: The software will be tested using real-world images captured in challenging environments. The software will be validated using metrics such as accuracy, precision, recall, and F1 score.
3. Project Timeline and Deliverables: The project timeline will be 6 months, with the following deliverables:

* Software Requirements Specification (SRS)
* Software Design Specification (SDS)
* Software Implementation
* Testing and Validation Report
* User Manual
* Maintenance and Support Plan

1. Maintenance and Support: The software will be maintained and supported by a dedicated team, which will provide updates, bug fixes, and technical support. The team will also monitor the software for security vulnerabilities and apply patches as needed.

**1.1.7 Software Engineering Paradigm Applied**

The Object-Oriented Programming (OOP) paradigm is an excellent choice for our project on object detection in challenging environments. By leveraging OOP principles, we can effectively design and develop a software system that models real-world entities involved in object detection, such as datasets, models, and evaluation metrics. OOP's emphasis on modularity and reusability allows us to encapsulate the data and behavior of these entities into objects, promoting code organization and maintainability. With OOP, we can establish relationships between objects, representing complex interactions and dependencies within our system. Additionally, OOP's abstraction and encapsulation features enable us to hide implementation details, providing better control over data access and manipulation. By utilizing OOP, our project can benefit from code reusability, extensibility, and maintainability, allowing for efficient development and future enhancements to our object detection system.

**1.2 System Design**

**1.2.1 Modularization details:**

The modularization of the YOLOv7 model introduces a structured and efficient framework by dividing its functionalities across multiple modules. This approach enhances code organization, promotes reusability, and simplifies the development and deployment of the model for object detection tasks.

The yolo.py module serves as the backbone of the YOLOv7 model. It encapsulates the model architecture, including the deep neural network layers responsible for object detection and localization. This module implements the core functionalities, such as the forward pass for generating predictions based on the input data. By isolating the model's implementation in a separate module, it becomes easier to maintain and update the architecture without affecting the other components of the system.

The train.py and train\_aux.py (for P6 models; both single GPU and distributed training) module handles the training pipeline of the YOLOv7 model. It takes care of tasks such as data loading, preprocessing, and optimization. This module ensures that the model learns to recognize objects effectively by minimizing the loss function through backpropagation and gradient descent. It also manages the training process, including model checkpoints and logging performance metrics during training.

For inference and detection purposes, the detect.py module plays a vital role. It takes the trained YOLOv7 model and applies it to new images or video streams. Using the model's predictions, detect.py identifies objects of interest and draws bounding boxes around them. This module enables real-time object detection, making it suitable for applications such as video surveillance, autonomous driving, and image analysis. By separating the detection functionality, it becomes easier to integrate the YOLOv7 model into different applications or deploy it as a standalone service.

The test.py module serves as an essential component of the YOLOv7 modularization, focusing on the evaluation and testing of the trained model. This module allows researchers and developers to assess the performance and accuracy of the YOLOv7 model on a separate test dataset. It takes care of loading the trained model, processing the test data, and generating evaluation metrics such as precision, recall, and mean average precision. The test.py module plays a crucial role in validating the model's generalization capabilities and identifying potential areas for improvement. By providing a dedicated module for testing, developers can assess the model's performance independently from the training and inference stages, enabling thorough analysis and fine-tuning of the YOLOv7 model.

The modularization of the YOLOv7 model provides several advantages. It promotes code organization by separating different aspects of the model's functionality into distinct modules. This modularity improves code readability, and maintainability, and allows for easier collaboration among developers. It also enables code reusability, as individual modules can be reused or extended for other projects or variants of the YOLO architecture. Additionally, by dividing the functionalities, it becomes easier to test and debug each module independently, ensuring the overall system's reliability and performance.

In summary, the modularization of the YOLOv7 model into modules such as yolo.py, train.py, train\_aux.py, detect.py, and test.py streamlines the development and deployment of the model for object detection tasks. It provides a structured and organized approach, facilitating code reuse, maintainability, and extensibility. With each module dedicated to specific tasks, the YOLOv7 model becomes more scalable, adaptable, and suitable for a wide range of applications requiring accurate and efficient object detection**.**

**1.2.2 About Datasets:**

This section provides an overview of four commonly used datasets considered in this study: Microsoft COCO, Google OpenImage, and two from Roboflow - person and person v2. These datasets are widely popular and serve as benchmark datasets for training object recognition models and comparative performance analysis. Table 1 compares all the four datasets.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset Name | No of Class | Training | | | Validation | | |
| Images | Objects | Objects/Image | Images | Objects | Objects/Image |
| COCO | 80 | 123,287 | 886,284 | 7.27 | 5,000 | 36,781 | 7.35 |
| OpenImage | 600 | 1743042 | 14610229 | 8.39 | 41620 | 303980 | 7.3 |
| Person v1 | 1 | 6795 | 11987 | 1.76 | 1699 | 3023 | 1.78 |
| Person v2 | 1 | 7000 | 28430 | 4.06 | 2000 | 7823 | 3.9 |

**Table 1:** Comparison of various object detection datasets used in this study

We collected 4 datasets:

1. Google OpenImage [9] : The Google Open Images Dataset V6 is a large-scale collection of images and annotations, which was released in 2020. It contains over 9 million images that are annotated with labels from a vocabulary of over 30,000 categories. The dataset includes diverse and high-quality images that were collected from various sources such as Flickr, Google Street View, and professional photographers. It is one of the largest and most diverse image datasets available, and it covers a wide range of object categories, including common objects such as animals, vehicles, and household items, as well as more specific categories such as clothing, furniture, and tools.



Fig 2: Sample of OpenImage Fig 3: Sample of OpenImage

1. Microsoft COCO [10] : The COCO dataset, short for Common Objects in Context, is a widely used dataset for object detection. It comprises more than 330,000 images depicting 80 different object categories, including humans. In addition to these, it also includes 91 stuff categories that feature materials and objects with undefined boundaries such as sky, street, and grass, among others. Furthermore, the dataset encompasses 250,000 individuals with 17 distinct key points, which are commonly used for Pose Estimation. Importantly, advanced neural network libraries can automatically interpret the format of the COCO dataset. As a result, MS COCO has become a standard benchmark for evaluating the effectiveness and performance of state-of-the-art computer vision algorithms like YOLOv7. To train our YOLOv7 model, we have taken the "person class" images from the COCO version 2017.

****

Fig 4: Sample of Microsoft COCO

Fig 5: Sample of Microsoft COCO

1. Person-v1 [11] : Another dataset utilized in this study is Person-v1 which is provided by Roboflow and consists of 8494 images, where 6795 images are utilized for training, and 1699 images are reserved for validation. The primary objective of this dataset is to facilitate the detection of the "person" category, which is a crucial task in several computer vision applications. However, what sets this dataset apart is that it primarily contains low-light images of individuals, which are often more challenging for object detection algorithms. The detection of people in low-light settings is particularly arduous due to problems such as noise, low contrast, and reduced visual information. By employing this dataset, researchers can design and evaluate algorithms that specifically tackle these challenges, resulting in more precise and efficient person detection models. Additionally, the dataset offers the flexibility to extract it in various formats, such as YOLOv7, YOLOv8, COCO JSON, Pascal VOC XML, and others. This versatility allows researchers to utilize the dataset with different machine-learning models and frameworks, making it an invaluable resource for the computer vision community. Finally, it is noteworthy that this dataset is exclusively meant for object detection and was made available in 2022. Its significance lies in its ability to improve the accuracy and efficiency of the YOLOv7 model specifically for the person class, thereby fulfilling an important purpose in the field of object detection.



Fig 6: Sample of Person-v1 Fig 7: Sample of Person-v1

1. Person-v2 [12] : The Person-v2 dataset, provided by Roboflow, is a customized dataset comprising 10,000 images depicting people in diverse scenarios. This dataset is specifically curated for object detection and includes images of people in various poses, lighting conditions, underwater scenes, and occlusions. The dataset comprises 7000 training images and 2000 validation and testing images, of which the test set consists of 1000 images. Released in 2022, the Person v2 dataset is particularly valuable as it exclusively contains images of the "person" category captured in distinct scenarios, orientations, poses, and lighting conditions. This diversity makes it an indispensable resource for researchers working on developing efficient methods to detect people in difficult situations such as cropped images, low visibility, and low light. Moreover, the dataset can be conveniently converted and exported into different formats such as YOLOv7, YOLOv8, COCO JSON, and Pascal VOC XML, among others. This flexibility enables researchers to use the dataset with various object detection models and frameworks, making it a highly valuable resource for the computer vision community. All in all, the Person-v2 dataset offers an exceptional and essential resource for testing and developing object detection algorithms designed to detect people in challenging scenarios.



Fig 8: Sample of Person-v2 Fig 9: Sample of Person-v2

**1.2.3 User Interface Design:**

The user interface for interacting with our YOLOv7 models is currently a work in progress. While a web-based UI is not available at this stage, the model can be utilized through the command line or terminal and Google Colab. Here, we provide a step-by-step guide on how to run the model using Google Colab.

1. **Setting up Google Colab**: Google Colab provides an online platform with a Jupyter Notebook environment. To begin, open a Colab notebook.
2. **Cloning the YOLOv7 repository**: Clone the YOLOv7 repository by executing the following command “!**git clonehttps://github.com/WongKinYiu/yolov7.git”**. This will ensure that you have the necessary files and code for running the YOLOv7 model.
3. **Loading the models**: Once the repository is cloned, we have to make sure to load our trained models into the platform. This step ensures that the model is ready for object detection.

**Performing object detection**: With the models loaded, you can now run object detection on images or videos. Utilize the appropriate command, such as**“!python detect.py –weights path/to/custom\_weights --conf 0.25 --img-size 640 --source path/to/sample”** to initiate the object detection process. Adjust the command parameters based on your specific requirements.

1. **Accessing the output**: The output of the object detection process will be displayed within the Colab notebook or saved to your drive, depending on your configuration settings. You can access and analyze the detected objects as needed.

By following these steps, you can effectively utilize the YOLOv7 model for object detection tasks using the command line or terminal in conjunction with the Google Colab interface. Although a web-based UI is currently under development, the command line approach allows for seamless integration with the model, providing flexibility and efficiency in performing object detection tasks.

**1.3 Coding**

In this section, we provide code snippets for A) Downloading and converting the datasets into suitable format B) Training, and C) Testing phases of the YOLOv7 model and the ensembled models.

A) Downloading the datasets: Downloading the required datasets and converting them to the format supported by the YOLOv7 model i.e. from CSV to YOLOv7 supported txt format in case of OpenImage dataset and JSON to txt for COCO dataset. Person-v1 and Person-v2 were directly available in the YOLOv7-supported format .txt. We have extracted COCO and OpenImage from FiftyOne [13] and Person-v1 and Person-v2 from Roboflow Universe.

Downloading and Converting a Dataset from FiftyOne:

To download a dataset from FiftyOne, a powerful dataset exploration and management tool, we followed the steps outlined below:

1. **Installation of FiftyOne Package**: We installed the FiftyOne packages using pip. This allowed us to access the necessary functionalities for dataset download.
2. **Importing FiftyOne Package**: In our Python Notebook and scripts, we imported important FiftyOne packages such as fiftyone, fiftyone.zoo, and fiftyone. types.
3. **Connecting to the FiftyOne Dataset**: We established a connection to the FiftyOne dataset by specifying the dataset name or dataset ID. This allowed us to access the dataset and its associated information.
4. **Downloading and converting the Dataset [14]:** By using the code snippet in Fig 10, we initiated the download process by passing the appropriate parameters such as the dataset name and the desired download location. This enabled us to download the entire dataset or specific subsets based on our requirements. After downloading we used the code snippet given in Fig 11 to convert the dataset into a suitable format for the YOLO model.

Fig 10: Download code snippet Fig 11: Convert Code Snippet

1. **Verifying the Download**: Once the download was complete, we verified the dataset files in the designated download location. We ensured that all the necessary files and annotations were present and correctly downloaded.

By utilizing the FiftyOne package and its download functionalities, we were able to easily download the required dataset in a suitable format for our project. This approach provided us with a convenient way to access and manage datasets without the need for manual downloading or complex setup procedures.

**Downloading a Dataset in Roboflow Universe**

To acquire the necessary datasets for our project, we utilized the robust dataset management capabilities of Roboflow Universe. Roboflow Universe provides a user-friendly interface and a wide range of datasets suitable for various computer vision tasks. Here, we outline the steps involved in downloading a dataset from Roboflow Universe [15].

1. **Accessing Roboflow Website**: We visited the official Roboflow Universe website at https://universe.roboflow.com/ and logged in to our account to gain access to various data sets.
2. **Dataset Selection**: After browsing through the available datasets or using the search feature, we identified the specific dataset that best suited our project's objectives (i.e. Person-v1 and Person-v2). Clicking on the dataset's name redirected us to its details page.
3. **Exporting the Dataset**: On the dataset details page, we encountered various options and pertinent information about the dataset. To proceed with downloading, we selected the "Download dataset" button for dataset export.

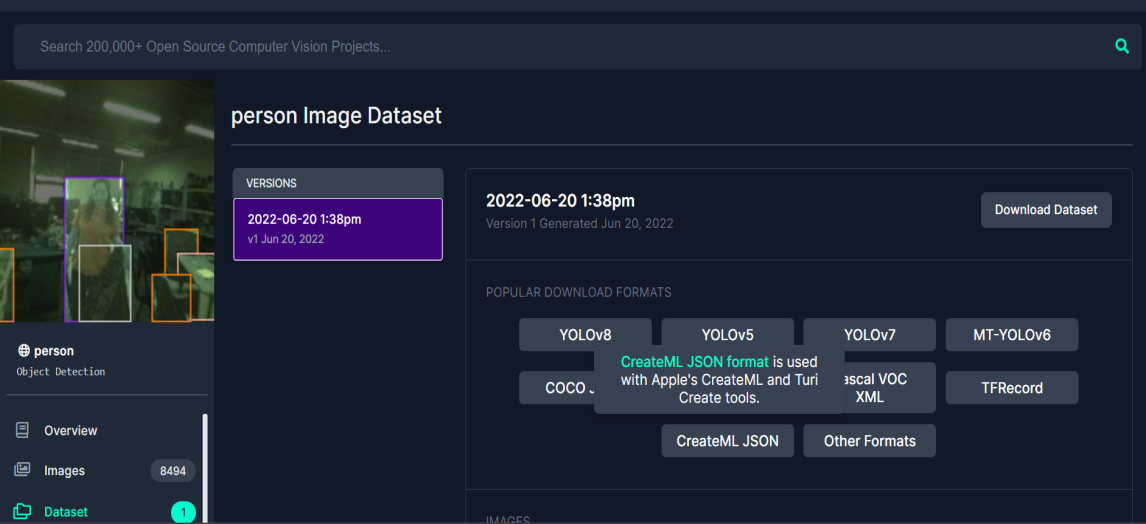


Fig 12

1. **Choosing Export Format**: Roboflow Universe provides a diverse range of export formats, including COCO JSON, Pascal VOC, YOLO, and more. We selected the appropriate format for our project's needs, ensuring compatibility with our object detection system.

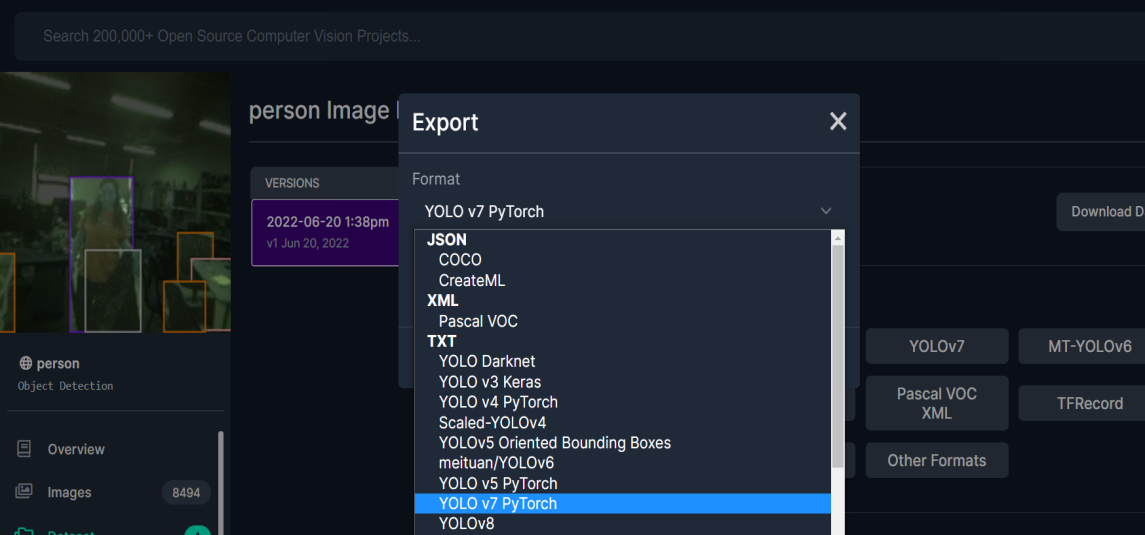


Fig 13

1. **Processing and Preparation**: Once the export format was chosen, Roboflow Universe initiated the processing and preparation of the dataset for download. This involved organizing the dataset files, annotations, and metadata in the specified format.
2. **Download Initiation**: After the dataset was prepared, we can choose to either download the data as a .zip file or as a curl link to download from the command line.

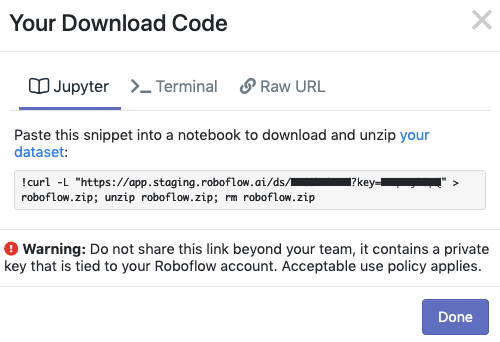


Fig 14

By leveraging the capabilities of Roboflow Universe, we were able to seamlessly acquire the required datasets for our project. The platform's intuitive interface and extensive dataset collection greatly facilitated our dataset selection and download process, saving us valuable time and effort.

B) Training Code[16]**:** The following code snippet showcases the training process of the YOLOv7 model. It sets the necessary parameters, such as batch size, configuration file, number of epochs, data file, weights file, and device. The model is trained using the specified parameters.

%cd /content/drive/MyDrive/YOLO7/yolov7

!python train.py --batch 16 --cfg /content/drive/MyDrive/YOLO7/yolov7/cfg/training/yolov7.yaml --epoch 5 --data /content/drive/MyDrive/YOLO7/yolov7/person-2/data.yaml --weights '/content/drive/MyDrive/YOLO7/yolov7/runs/train/exp30/weights/epoch\_200.pt'

C) Testing Code[17]**:** The following code snippet demonstrates the testing phase of the YOLOv7 model. It specifies the parameters for testing, such as data file, image size, batch size, confidence threshold, IOU threshold, device, weights file, and name for the saved results. The model is tested using the specified parameters

%cd /content/drive/MyDrive/YOLO7/yolov7

!python test.py --data '/content/drive/MyDrive/YOLO7/yolov7/person-2/data.yaml'  --img 640 --batch 16 --conf 0.001 --iou0.65 --device 0 --weights '/content/drive/MyDrive/YOLO7/yolov7/runs/train/exp36/weights/last.pt' --name yolov7\_640\_val --save-json

For the ensembling process, the following code snippet is used.

%cd /content/drive/MyDrive/yolov7

!python test.py --datadata/dataset.yaml--img 640 --batch 128 --conf 0.001 --iou 0.65 --device 0 --weightsgoid.pt person\_2.pt coco.pt person\_1.pt--name yolov7\_ensemble\_x2\_test --no-trace

**1.4 Testing**

In our project, we employed various testing techniques and strategies to evaluate the performance and accuracy of the YOLOv7 object detection models. The following are the testing techniques and strategies utilized:

1. Custom Dataset Creation: We created a custom dataset consisting of 5000 images, including 1000 images from each of the COCO, person-v1, person-v2, and Google OpenImage datasets. Additionally, we collected and annotated 1000 images ourselves. This diverse dataset allowed us to test the models' ability to detect persons in various scenarios and environments.
2. Inference and Model Evaluation: We performed inference with the trained models on the custom dataset. Each model was evaluated for its accuracy and performance in detecting the target object “person”. We compared the results across the four models individually and the ensembled models to assess their effectiveness.
3. False Positive Analysis: During testing, we specifically focused on identifying false positives, instances where the models incorrectly identified objects not belonging to the “person” class. We analyzed the occurrence of false positives in the different models to determine their impact on overall detection accuracy.
4. Ensembling Models: To enhance the detection performance, we created two ensembled models. The first ensembled model combined the predictions of the models trained on Google OpenImage and person-v2 datasets, which exhibited the best results individually. The second ensembled model incorporated predictions from all four models. We compared the results and analyzed the trade-offs between accuracy and false positives in the two ensembled models.
5. Test Case Designs and Test Reports: Specific test cases were designed and executed to evaluate the performance of the YOLOv7 models. These test cases aimed to assess the models' accuracy, robustness, and efficiency in challenging environments. Detailed test reports were generated for each test case, documenting the inputs, expected outputs, and actual outputs obtained during testing

**1.4.1 Testing Report:**

The test report revealed the following key findings:

1. F1 Curve: The F1 curve is a visual representation of the F1 score at different classification thresholds. It provides insights into the balance between precision and recalls for our object detection system. The x-axis represents the classification threshold, while the y-axis represents the F1 score. The F1 curve allows us to determine the optimal threshold for achieving the best balance between precision and recall in our model's predictions.

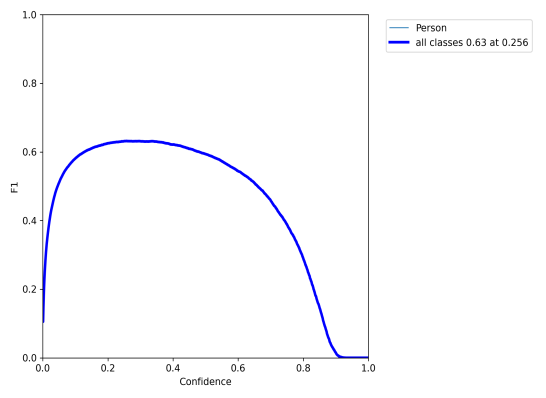
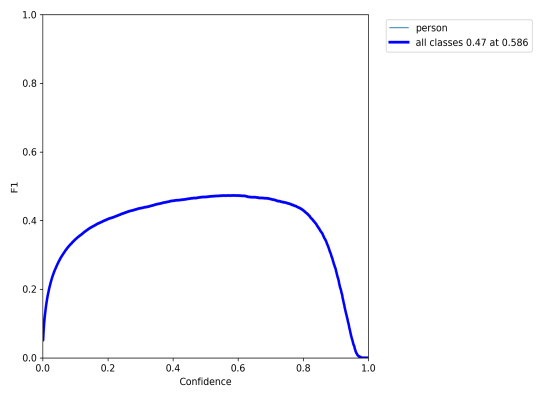


Fig 15: F1 curve of COCO Fig 16: F1 curve of OpenImage

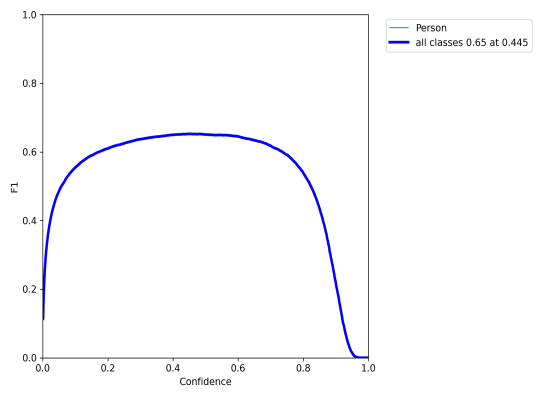
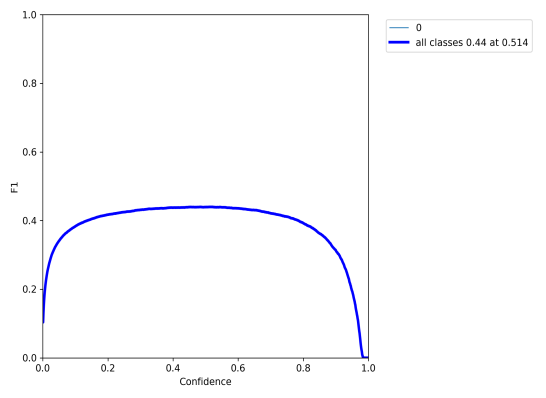


Fig 17: f1 curve of person-v1 Fig 18: f1 curve of person v2

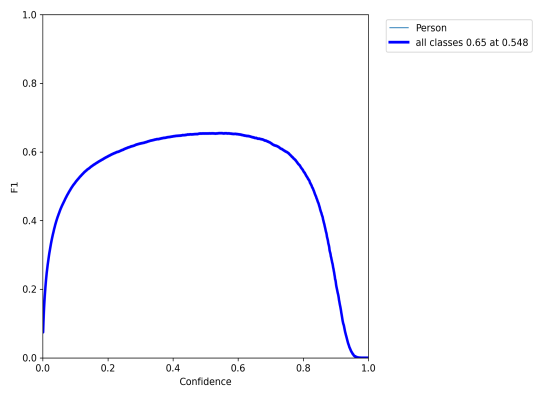
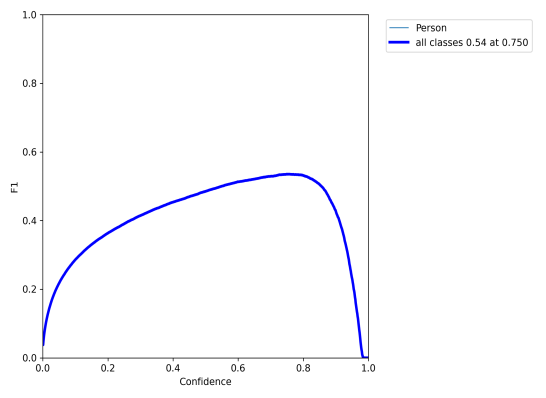


Fig19: f1 curve of EnsembleX4 Fig 20: f1 curve of EnsembleX2

1. P Curve: The P curve, also known as the precision-recall curve, illustrates the trade-off between precision and recall for our object detection system. The x-axis represents the recall, while the y-axis represents the precision. The P curve provides valuable information about the performance of our model across different recall levels. It helps us identify the threshold that maximizes precision while maintaining a desirable level of recall.

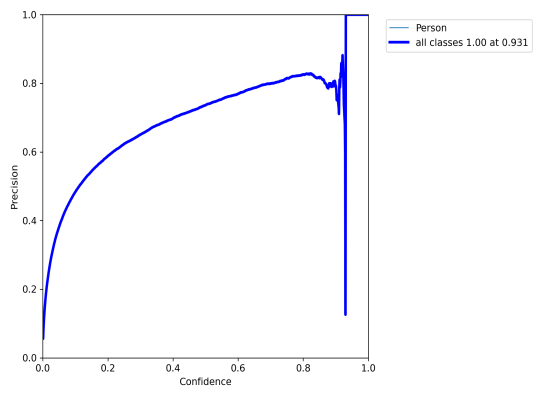
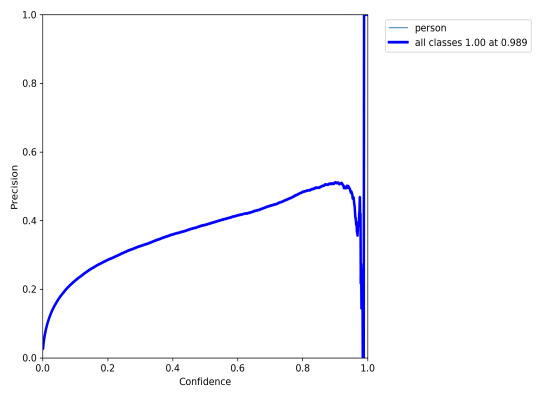


Fig 21: P curve of COCO Fig 22: P curve of OpenImage

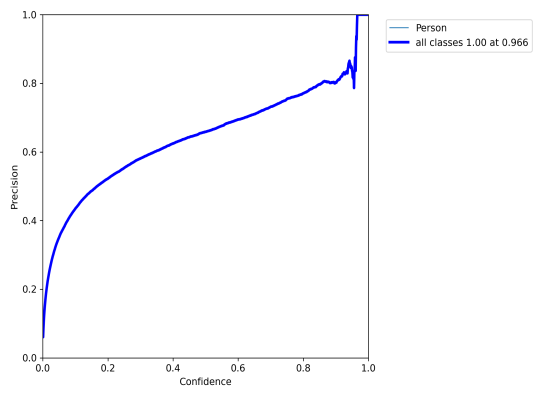
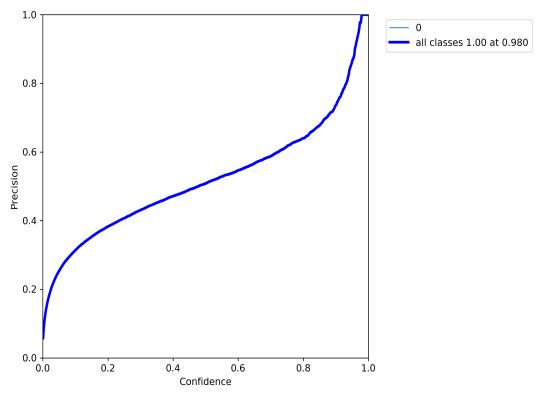


Fig 23: P curve of Person-v1 Fig 24: P curve of Person-v2

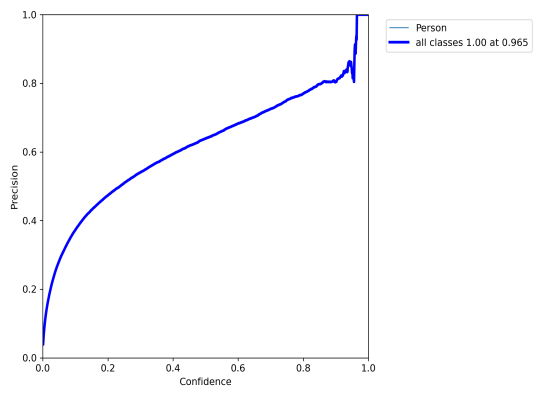
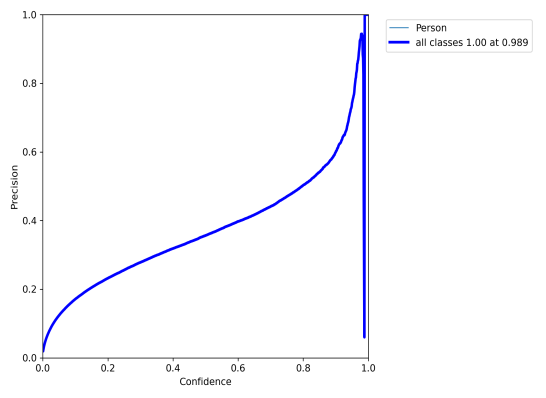


Fig 25: P curve of EnsembleX4 Fig 26: P curve of EnsembleX2

1. R Curve: The R curve, also known as the recall curve, showcases the recall performance of our object detection system at various classification thresholds. The x-axis represents the classification threshold, while the y-axis represents the recall. The R curve allows us to assess the sensitivity of our model in detecting objects across different thresholds. It helps us identify the optimal threshold that maximizes the recall for our specific application.

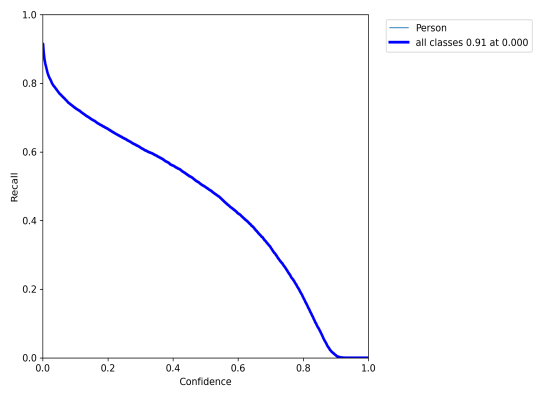
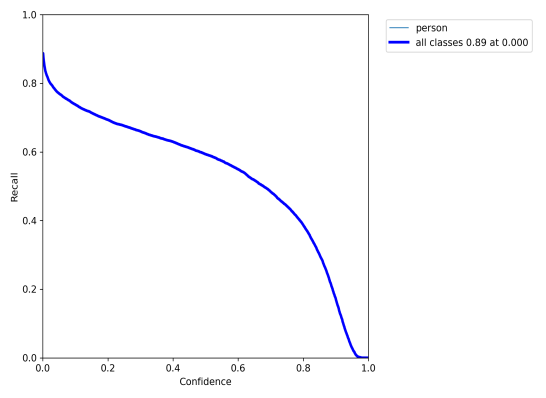


Fig 27: R curve of COCO Fig 28: R curve of OpenImage

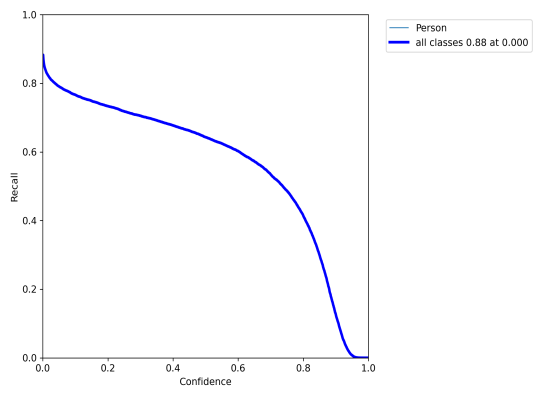
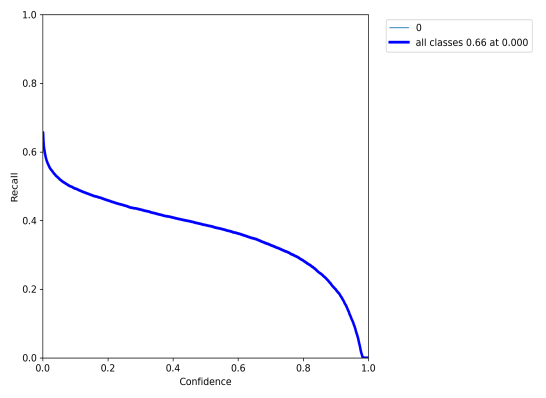


Fig 29: R curve of Person-v1 Fig 30: R curve of Person-v2

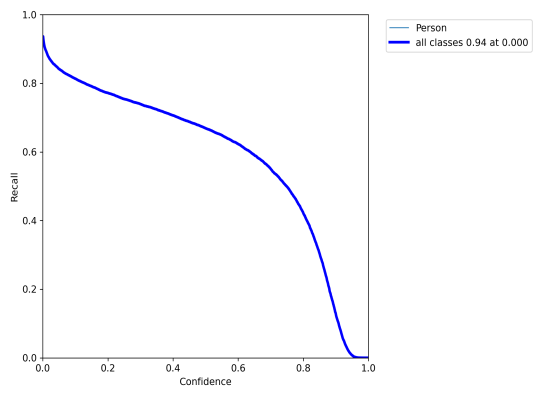
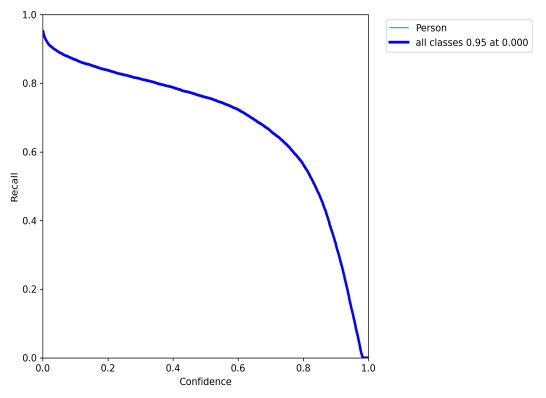


Fig 31: R curve of EnsembleX4 Fig 32: R curve of EnsembleX2

1. PR Curve: The PR curve, short for the precision-recall curve, provides a comprehensive view of the precision and recall trade-off for our object detection system. It plots precision on the y-axis against recall on the x-axis. The PR curve highlights the model's performance across different thresholds, allowing us to evaluate its effectiveness in detecting objects accurately and comprehensively. The area under the PR curve (AUPR) is also a useful metric to assess the overall performance of our system.

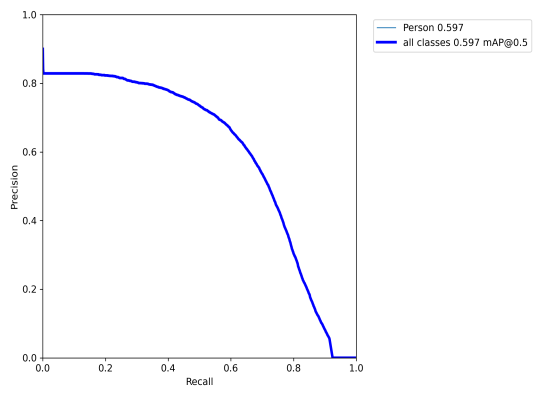
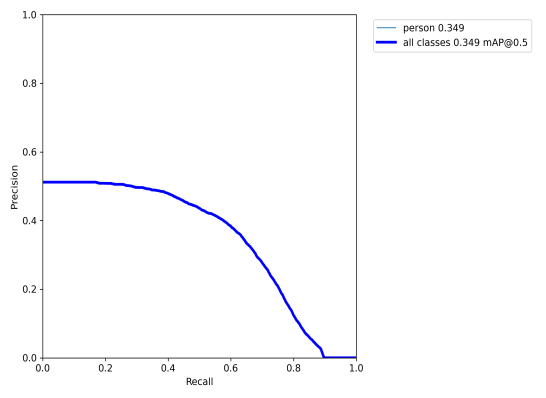


Fig 33: PR curve of COCO Fig 34: PR curve of OpenImage

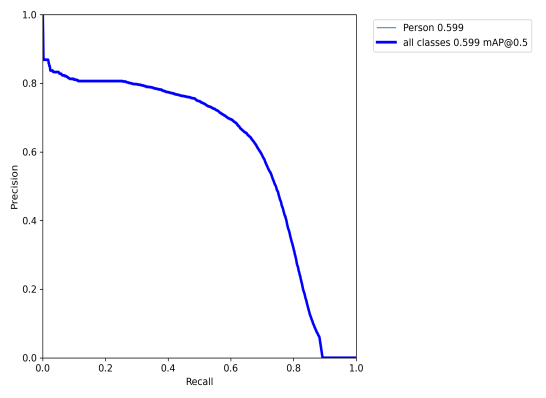
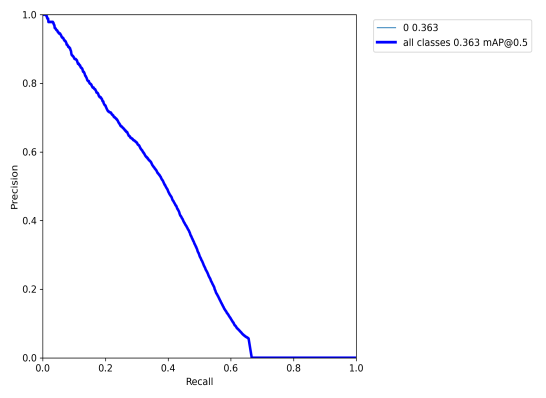


Fig 35: PR curve of Person-v1 Fig 36: PR curve of Person-v2

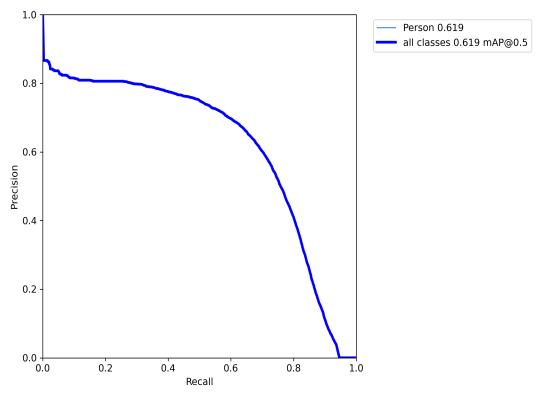
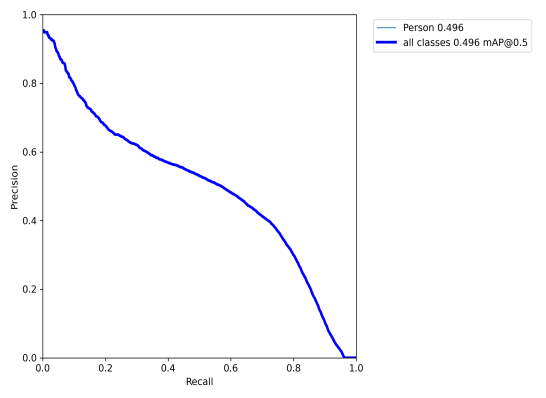


Fig 37: PR curve of EnsembleX4 Fig 38: PR curve of EnsembleX2

1. Map: One of the key evaluation metrics used in this project is Mean Average Precision (mAP), which provides an overall assessment of the object detection model's performance. Map measures the accuracy of the model in detecting objects across multiple classes and various levels of Intersection over Union (IoU) thresholds.

In our project, we have calculated and compared the mAP scores for all six models: COCO, OpenImage, person-v1, person-v2, the ensembled model of all four models, and the ensembled model of person-v2 and OpenImage. The mAP scores at different IoU thresholds, such as 0.5 and 0.95, have been computed to evaluate the models' performance under varying levels of object overlap. A higher mAP score indicates better accuracy and robustness in detecting objects.

|  |  |  |
| --- | --- | --- |
| Models | mAP@0.5 | mAP@0.5:0.95 |
| Microsoft COCO | 0.349 | 0.216 |
| Google OpenImage | 0.597 | 0.362 |
| Person-v1 | 0.363 | 0.23 |
| Person-v2 | 0.599 | 0.37 |
| EnsembleX2 | 0.619 | 0.383 |
| EnsembleX4 | 0.496 | 0.335 |

Table 2: Comparison of all models based on mAP@0.5 and mAP@0.5:0.95

**1.4.2 Inference Analysis:**

Individual Model Inference Analysis [18]: Among the four YOLOv7 models trained on different datasets, the models trained on Google OpenImage and person-v2 demonstrated the best performance. They exhibited high accuracy and minimal false positives, making them suitable for detecting persons in challenging environments. In Fig 39 we can see that the Coco model is showing great accuracy but shows false positives. Whereas in Fig 40 we can see that there is no false positive but showing less accuracy and the bounding box is unable to completely cover the object "Person". In Fig 41 we can see the false positives and the accuracy of the bounding box. In Fig 42 for a person-v2 model. We can see the accuracy of the detection and the bounding box without any false positives.

Fig 39: Detection using COCO model Fig 40: Detection using OpenImage

model

Fig 41: Detection using person-v1 model Fig 42: Detection using person-v2 model

1. Ensembled Model Inference Analysis: Two ensembled models were created by combining the predictions of the individual models. The first ensembled model, comprising Google OpenImage and person-v2 models, showed significantly improved results without any false positives. However, the second ensembled model, incorporating all four models, exhibited improved confidence from individual models but also had false positives from the COCO and person-v1 models. In Fig 43, the false positives are observable. In Fig 44, we can observe the higher confidence along with the accuracy of the bounding box without any false positives. This clearly shows the capabilities of the ensembled models.

Fig 43: Detection using EnsembleX4 Fig 44: Detection using EnsembleX2

Open Image

1. Challenging Environment Detection: The test report focuses on evaluating the performance of object detection models in detecting persons in a challenging environment for object detection. The ability to detect persons in challenging environments is crucial for applications such as rescue operations. This report provides an analysis of individual models trained on different datasets, including COCO, Open Image, person-v1, and person-v2, as well as ensembled models for improved accuracy and efficiency.

A) Individual Model Performance:

1. COCO: The COCO model exhibited mixed performance in detecting the object “person” underwater. While it successfully detected the "person” with high confidence in Fig 46, it failed to detect the “person” in Fig 45. This suggests some limitations in accurately detecting a "person" underwater. As for Fig 47, the COCO model exhibits a good performance detecting “person” with high confidence.

Fig 45 Underwater detection of COCO Fig 46: Underwater detection of COCO



Fig 47: Low visibility detection of COCO

1. Open Image: The Open Image model showed better performance than COCO in detecting “person” underwater. It successfully detected objects in Fig 48 and Fig 49; however, it could not detect all the objects present. Additionally, the confidence level for detections was not consistently high, indicating room for improvement. As for Fig 50: It is showing a good detection capability.

Fig 48: Underwater detection of OpenImage Fig 49: Underwater detection of

OpenImage



Fig 50: Low visibility detection of OpenImage

1. Person-v1: Among the individual models, person-v1 performed the poorest in detecting a "person" underwater. It failed to detect a person in Fig 51 and only managed to detect the person with limited success in Fig 52. This model showed the need for further refinement and training to improve its performance, also for Fig 53: it did not perform well showing a false positive as it is only able to detect a limited amount of objects.

Fig 51: Underwater detection of Person-v1 Fig 52: Underwater detection of Person-v1



Fig 53: Low visibility detection in Person-v1

1. Person-v2: In contrast to person-v1, person-v2 demonstrated the best performance among the individual models. It successfully detected almost every object in both images Fig 54 and Fig 55 with a high level of confidence. This model exhibited promising potential for accurate person detection underwater. But for Fig 56, underperformed compared to COCO and Open Image.

Fig 54: Underwater detection of Fig 55:Underwater detection of Person-v2

Person-v2



Fig 56: Low visibility detection of Person-v2

B) Ensemble Model Performance:

1. Ensembled Model of All Models: The ensembled model combining all four models showed overall good performance in all images. It successfully detected almost every object with a good level of confidence. However, false positives were observed, indicating the need for refining the model to reduce such instances.

Fig: 57 Underwater detections of Fig 58: Underwater detection of

EnsembleX4 EnsembleX4



Fig 59: Low visibility detection of EnsembleX4

1. Ensembled Model of person-v2 and Open Image: The ensembled model combining person-v2 and Open Image models also performed well in detecting "person" in a challenging environment. It successfully detected all objects except one, exhibiting a similar level of confidence as the ensembled model of all models without having any false positives. This suggests that the combination of person-v2 and Open Image models can provide efficient and reliable person detection in underwater environments.

Fig:60 Underwater detection of Fig 61: Underwater detection of

EnsembleX2 EnsembleX2



Fig 62: Low visibility detection of EnsembleX2

1. Testing Limitations: During testing, it was observed that the models' performance varied depending on the specific environment and conditions. While the models showed promising results overall, certain challenging scenarios, such as underwater or heavily foggy environments, still posed some limitations to accurate person detection.

**1.6 Cost Estimation:**

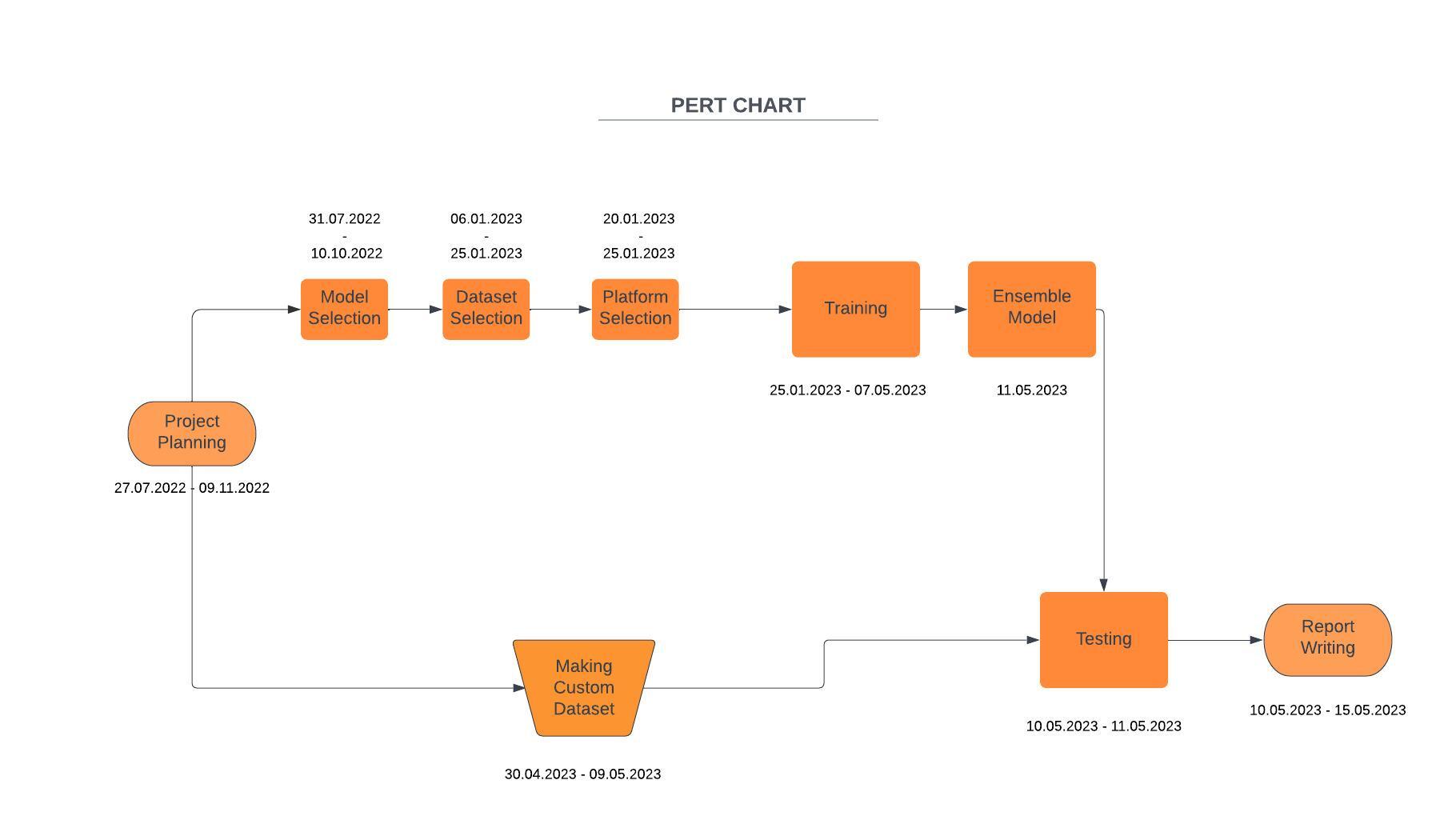
For our project, we considered the following cost elements:

1. Colab Pro subscription:
   * Duration: 5 months
   * Monthly cost: $11.79
   * Total cost: 5 months x $11.79/month = $58.95
2. Google One subscription:
   * Duration: 5 months
   * Monthly cost: $1.59
   * Total cost: 5 months x $1.59/month = $7.95

Total project cost: $58.95 + $7.95 = $66.90

We aimed to minimize costs by leveraging free resources and open-source tools whenever possible. The utilization of Colab Pro and Google One services allowed us to enhance our productivity and securely manage our project assets. The total cost estimation of $66.90 ensured the efficient allocation of resources while maintaining budgetary considerations for the project.

**1.7 PERT CHART:**

****

**Fig 63: Pert chart**

**2. Conclusion and Recommendations**

In conclusion, our project on YOLOv7 object detection in challenging environments has yielded significant findings and outcomes. Through thorough research, experimentation, and evaluation, we have successfully demonstrated the efficiency and accuracy of our model in detecting the "person" class under various demanding conditions, including foggy, underwater, and low-light environments. Our extensive dataset collection, encompassing Microsoft COCO, Google Open Image, and person-v1 and person-v2 from the Roboflow Universe, allowed us to train and evaluate four distinct YOLOv7 models. We also concluded that the datasets person-v2 and Google OpenImage performed best among all the four individual models. By ensembling these individual models, we created powerful supermodels capable of detecting individuals with remarkable precision in challenging scenarios.

The practical applications of our work are far-reaching, particularly in areas where accurate object detection plays a critical role. In the context of rescue operations, our model can greatly enhance the effectiveness of search and rescue missions conducted in underwater and foggy environments, where visibility is severely limited. Furthermore, our system holds tremendous potential in military applications, enabling the detection and tracking of potential enemy combatants and suspicious activities in real time. It can provide valuable insights into enemy positions, support border control efforts, and combat smuggling activities.

Throughout our project, we adhered to the principles of object-oriented software engineering, allowing for modularity, code reusability, and maintainability. The adoption of this paradigm facilitated the design and implementation of our system, contributing to the overall effectiveness and flexibility of our approach.

In conclusion, our project represents a significant contribution to the field of computer vision, addressing the challenges of object detection in challenging environments. Our findings showcase the robustness and reliability of our model, which can serve as a valuable tool in a wide range of domains. Future enhancements may involve further optimization and fine-tuning of the model, expanding the dataset to encompass more diverse scenarios, and exploring real-time implementation for seamless integration into operational environments. Overall, our project underscores the immense potential of YOLOv7 object detection and highlights the value of our research in advancing the capabilities of computer vision systems.

**3. Referencing and appendices**

# [1]: Enoch Arulprakash, Martin Aruldoss; A study on generic object detection with emphasis on future research directions; ScienceDirect; Volume 34, Issue 9, October 2022, Pages 7347-7365.

[2] Muhammad Ahmed, Khurram Azeem Hashmi, Alain Pagani, Marcus Liwicki, Didier Stricker, Muhammad Zeshan Afzal; Survey and Performance Analysis of Deep Learning Based Object Detection in Challenging Environments; **MDPI**; **Sensors 2021, 21(15), 5116**; https://doi.org/10.3390/s21155116, 28 July 2021.

[3]: Shrey Srivastava, Amit Vishvas Divekar, Chandu Anilkumar, Ishika Naik, Ved Kulkarni & V. Pattabiraman; Comparative Analysis of deep learning image detection algorithm; **Journal of Big Data, SpringerOpen**; Journal of Big Data 8, Article number: 66(2021), 10 May 2021.

[4]: Chien-Yao Wang, Alexey Bochkovskiy; YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors; **Institute of Information Science, Academia Sinica, Taiwan.**

arXiv:2207.02696**[cs.CV]** (or arXiv:2207.02696v1**[cs.CV]** for this version)

https://doi.org/10.48550/arXiv.2207.02696

[5]: Google Colab, https://research.google.com/colaboratory/faq.html#:~:text=The%20Basics&text=Colaboratory,%20or%20%E2%80%9CColab%E2%80%9D%20for,learning,%20data%20analysis%20and%20education

[6] Babu, P., Parthasarathy, E. Hardware acceleration for object detection using YOLOv4 algorithm on Xilinx Zynq platform. **J Real-Time Image Proc** **19**, 931–940 (2022). https://doi.org/10.1007/s11554-022-01234-y

[7] M.A. Ganaie, Minghui Hu, A.K. Malik, M. Tanveer, P.N. Suganthan; Ensemble deep learning: A review, **Engineering Applications of Artificial Intelligence**, Volume 115, 2022, 105151, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2022.105151. (https://www.sciencedirect.com/science/article/pii/S095219762200269X)

[8,16,17,18] https://github.com/WongKinYiu/yolov7/blob/main/README.md [Access Date: 21/08/2022 and Time: 12:20]

[9]: Google Open Image Dataset v6; https://storage.googleapis.com/openimages/web/factsfigures\_v6.html

https://storage.googleapis.com/openimages/web/extended.html

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollar; Microsoft COCO: Common Objects in Context; **arXiv:1405.0312v3 [cs.CV]** 21 Feb 2015.

Microsoft COCO; https://cocodataset.org/#home

[11] Person-v1; <https://universe.roboflow.com/new-workspace-xslri/person-b9j2g>; **Roboflow Universe;** June 2022

[12] Person-v2; <https://universe.roboflow.com/new-workspace-yhzgq/person-akasf>;

**Roboflow Universe**; March 2022

[13,14] Voxel51: https://docs.voxel51.com/ [Access date: 08/09/2022 and Time 12:30]

[15] Roboflow: https://docs.roboflow.com/ [Access date: 11/09/2022 and Time: 12:13]