Minor Project

Title: YOLO Object Detection using OpenCV

Content

Introduction

- YOLO (You Only Look Once) is a real-time object detection algorithm
- YOLO uses a single convolutional neural network (CNN) to detect objects in images
- YOLO architecture consists of three components: input, backbone, and output
- Input layer takes an image and applies convolutional layers to extract features
- Backbone contains convolutional and pooling layers that down sample feature maps and extract high-level features
- Output layer predicts bounding boxes, class probabilities, and confidence scores for each detected object
- YOLO is trained on large datasets such as COCO or VOC with labelled images, object classes, and bounding boxes
- During training, YOLO optimizes loss function that combines localization loss and classification loss
- This results in a model that can accurately detect objects in new images
- YOLO is designed to be fast and accurate, making it popular for real-time applications such as video surveillance, robotics, and autonomous vehicles.

Technical Concepts (Algorithms)

The technical concepts involved in implementing YOLO object detection using OpenCV and Python:

1. Darknet: Darknet is an open-source neural network framework written in C and CUDA. YOLO is built on top of Darknet and uses its neural network architecture.

2. YOLO models: YOLO comes with pre-trained models that have been trained on a large dataset (such as COCO) to detect various objects in different environments.

3. Image preprocessing: Before feeding the image to the YOLO model, it needs to be preprocessed to meet the input requirements of the model.

4. Object detection: The YOLO model takes an input image and outputs bounding boxes around the objects in the image along with their class probabilities. The object detection process involves feeding the preprocessed image to the YOLO model.

5. Non-maximum suppression: Non-maximum suppression is a technique used to eliminate duplicate bounding boxes around the same object.

6. Visualizing the output: Finally, the output of the YOLO model can be visualized by drawing the bounding boxes and class labels on the original image. OpenCV provides functions for drawing rectangles and text on images

Motivation

Motivation for our project is derived from various applications of computer vision in real life scenarios such as:

- Object detection may be used to monitor regions and detect suspicious activity, assisting in the maintenance of security in public and private settings.
- Object detection is an important component of self-driving automobiles since it helps the vehicle to recognise and avoid obstructions, pedestrians, and other vehicles on the road.
- Object detection in retail can be used to monitor consumer activity, optimise shop layouts, manage inventory, and prevent theft.
- Object detection can be used in medical imaging to detect and pinpoint anomalies in pictures, such as tumours or lesions, supporting clinicians in diagnosis and therapy planning.

Problem Statement

To determine, detect and create an efficient and accurate model for detecting objects in real-time images or videos (using a webcam).

Area of Application

1. **Self-driving cars:** Object detection is used to identify and track objects such as pedestrians, cars, and traffic signs on the road.
2. **Surveillance:** Object detection can be used for monitoring and detecting objects and individuals in surveillance cameras and security systems.
3. **Robotics:** Object detection is used to identify and locate objects in a robot's environment, allowing it to navigate and interact with its surroundings.
4. **Medical imaging:** Object detection is used in medical imaging to identify and locate abnormalities such as tumours, lesions, or other medical conditions.
5. **Agriculture:** Object detection can be used to monitor and identify crops, weeds, and pests, enabling precision farming techniques.

Dataset and Input Format

INPUT: Video Frames

1. The YOLO model takes video frames as input and processes them in real-time to detect objects.

2. Each video frame is a 2D array of pixels, with a width and height measured in pixels.

Dataset: Annotated Images

1. To train the YOLO model to recognize objects in videos, we need a dataset of annotated images.
2. An annotated image is an image that has been labeled with bounding boxes and class labels for each object in the image.
3. The bounding boxes define the location of the object in the image, and the class label indicates what type of object it is.
4. The dataset needs to include a wide range of objects in different positions, scales, and orientations to ensure that the model can generalize to new objects in the video.

Literature Review

“Understanding of Object Detection Based on CNN Family and YOLO”, by Juan Du. In this paper, they generally explained about the object detection families like CNN, R-CNN and compared their efficiency and introduced YOLO algorithm to increase the efficiency [2].

Aleksa Ćorović, et al. (2018) implemented a system for detecting the traffic participants using YOLOv3 and Berkley Deep Drive dataset. The number of object classes which can be detected by this system is five (truck, car, traffic signs, pedestrian and lights) in different conditions of the driving (snow, overcast and bright sky, fog, and night). The accuracy was 63% [3].

Omkar Masurekar, et al. (2020) created a model for object detection to help visually impaired people. YOLOv3 and Custom dataset that consists of three classes (bottle, mobile and bus) were used. Google text to speech (gtts) was used for sound generation. [4].

Sunit Vaidya, et al. (2020) implemented a web application and an android application for the object detection. YOLOv3 and the coco dataset were used in these systems. The authors found that the maximum accuracy in web applications is 89 % and 85.5% for mobile phones[5].

Zhimin Mo1, Liding Chen1, Wen-jing You in 2019 “Identification and Detection of Automotive Door Panel Solder Joints based on YOLO”. This YOLO algorithm, proposed identifies the position of the solder joints accurately in real time [6].

Cites Related Work

[2] Juan Du , "Understanding of Object Detection Based on CNN Family and YOLO", Citation Juan Du 2018 J. Phys.: Conf. Ser. 1004 012029

DOI 10.1088/1742-6596/1004/1/012029

[3]Aleksa .et.al "The Real-Time Detection of Traffic Participants Using YOLO Algorithm" , Corovic, Aleksa & Ilic, Velibor & Đurić, Siniša & Marijan, Mališa & Pavkovic, Bogdan. (2018). 10.1109/TELFOR.2018.8611986.

[4] Akila S .Et.Al "Indoor And Outdoor Navigation Assistance System For Visually Impaired People Using Yolo Technology" P-Issn: 2395-0072,International Research Journal Of Engineering And Technology (Irjet)

These are the cites used as the reference for our work.

Inference from Literature Review

Based on the literature review here, it can be inferred that YOLO (You Only Look Once) algorithm is a popular approach for object detection in various applications. YOLO is efficient in real-time object detection and can detect multiple objects in a single pass. This algorithm uses a single CNN to simultaneously predict the bounding boxes and the class probabilities for objects in an image.

The reviewed papers show how YOLO can be used in a variety of fields, including traffic detection, helping persons who are blind, locating solder joints on car door panels, and creating online and mobile applications for object identification.

The YOLO algorithm's accuracy varies based on the quantity of classes to be identified, the surrounding circumstances, and the quality of the training dataset. In web applications, some studies reported accuracy rates as high as 89%, while others reported lower rates of about 63%.

The reviewed papers demonstrate the YOLO algorithm's potential for fast and precise object recognition across a wide variety of applications and areas.

SWOT Analysis

Strengths -
YOLO (You Only Look Once) is a real-time object detection system that can detect objects in images and videos quickly and accurately.
It is a deep learning algorithm,so it can be trained to recognize new objects and improve its accuracy over time.

Weakness -

YOLO requires a significant amount of computational power, which can be a challenge for some systems.

It may also struggle with detecting small objects, especially if they are in a cluttered scene.

Opportunity -

With the growing popularity of deep learning and computer vision, there is a growing demand for object detection systems like YOLO.

Threat -

There are other object detection algorithms and software libraries available, which could compete with YOLO.

Objective

Main Objective

- To identify or detect objects in real-time images or videos using YOLO architecture.

Sub Objective

- To identify the bounding boxes of the object and to correctly predict the object inside those bounding boxes and then show the output in real time.

Methodology

Reference Software model -

https://docs.google.com/document/d/1LQvX5UZ-PGpJbjTsEVA0hbCqV68AC5HsbeGfmXBDBak/edit?usp=sharing

Steps –

Phase 1 - Requirement analysis

Phase 2 - Designing and development

Phase 3 - coding

Phase 4 - Testing

Phase 5 - Bringing of all the above phases together and putting them into practice .

Deliverable of Each Steps or Phase

Phase 1 - Requirement analysis

- Study the concepts of Python programming.
- Study of YOLO (You Only Look Once) algorithm, OpenCV.
- Study of Basic Neural Networks.

Phase 2 - Designing and development

The stages of design and development are further broken down.

- Our project development begins with the information gathered from the requirement and analysis phase of the project, then we will advance to the model creation phase where we will build a model based on the algorithms that we will use i.e. using the yolo algorithm.
- After the algorithm design phase is complete, the focus will shift to algorithm analysis and implementation for this project i.e., the accuracy on which the model predicting.

Deliverable of Each Steps or Phase

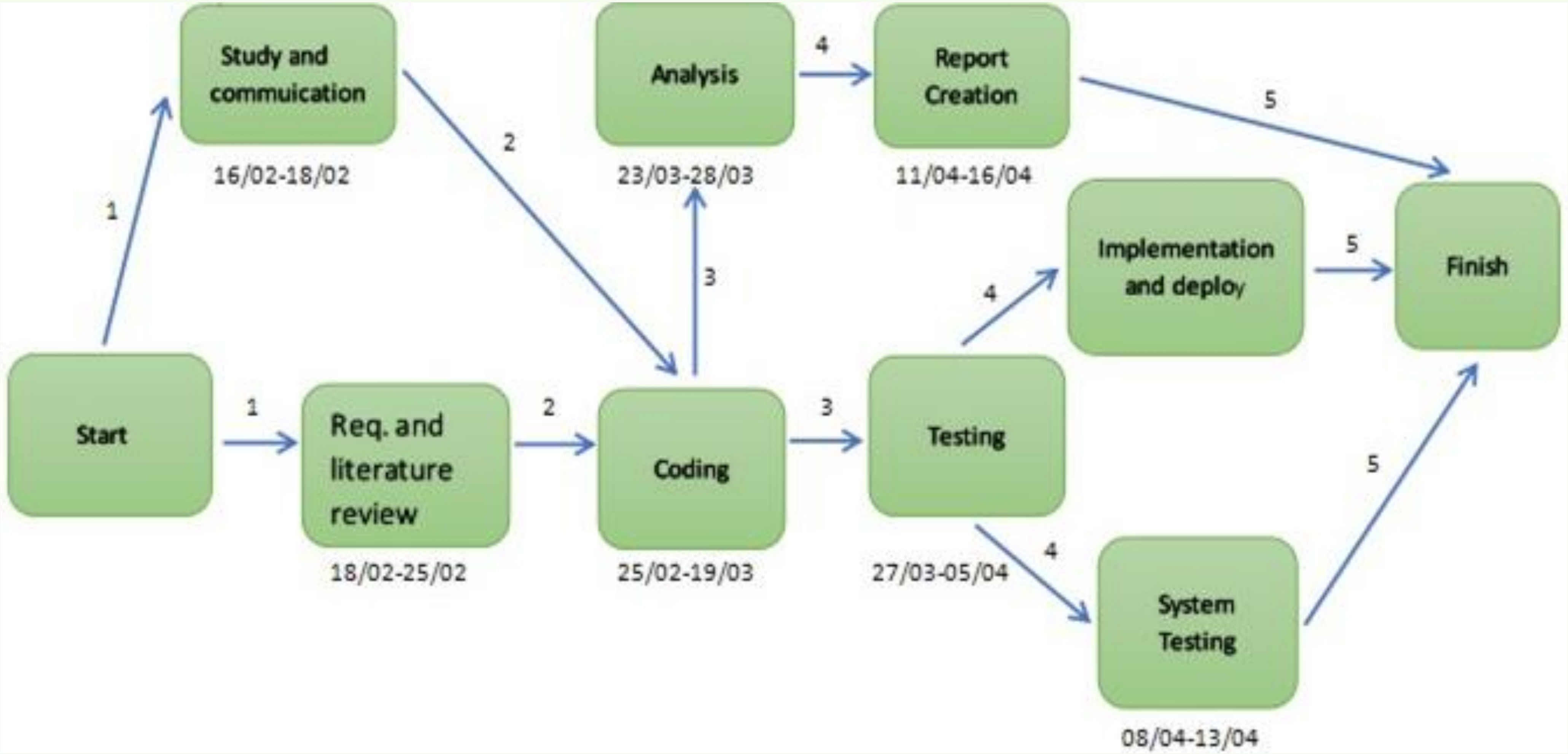
Phase 3 - Coding

After receiving system design papers, the work is divided into modules or units and assigned to team members before actual coding starts. The developers' major attention is on the code because it is being written during this stage. The project's most time-consuming phase will be this. Then making a program in the Python programming language and successfully producing an executable program are the first steps in putting this project into practice, i.e., by using Python, OpenCV and YOLO architecture.

Phase 4 - Testing

Now in order to assess and test the accuracy of our project, the testing will be done with multiple inputs of data i.e. to improve the accuracy score of the model.

Timeline



Study and commuication
16/02-18/02
Analysis
23/03-28/03
Report Creation
11/04-16/04
Implementation and deploy
Finish
Start
Req. and literature review
18/02-25/02
Coding
25/02-19/03
Testing
27/03-05/04
System Testing
08/04-13/04
1
1
2
2
3
3
4
4
4
5
5
5

The steps proposed for the project are outlined below:-

1. Install OpenCV and Python on your system and download the pre-trained YOLO model from the official website.
2. Load the YOLO model using OpenCV's DNN module.
3. Load the image or video that you want to perform object detection on.
4. Preprocess the image by resizing it and normalizing the pixel values.

Pass the preprocessed image through the YOLO model to get the object detections.

5. Postprocess the detections to filter out low-confidence detections and non-maximum suppression to remove overlapping detections.
6. Draw bounding boxes and labels around the detected objects on the image or video.
7. Display the output image or video with the detections overlaid.

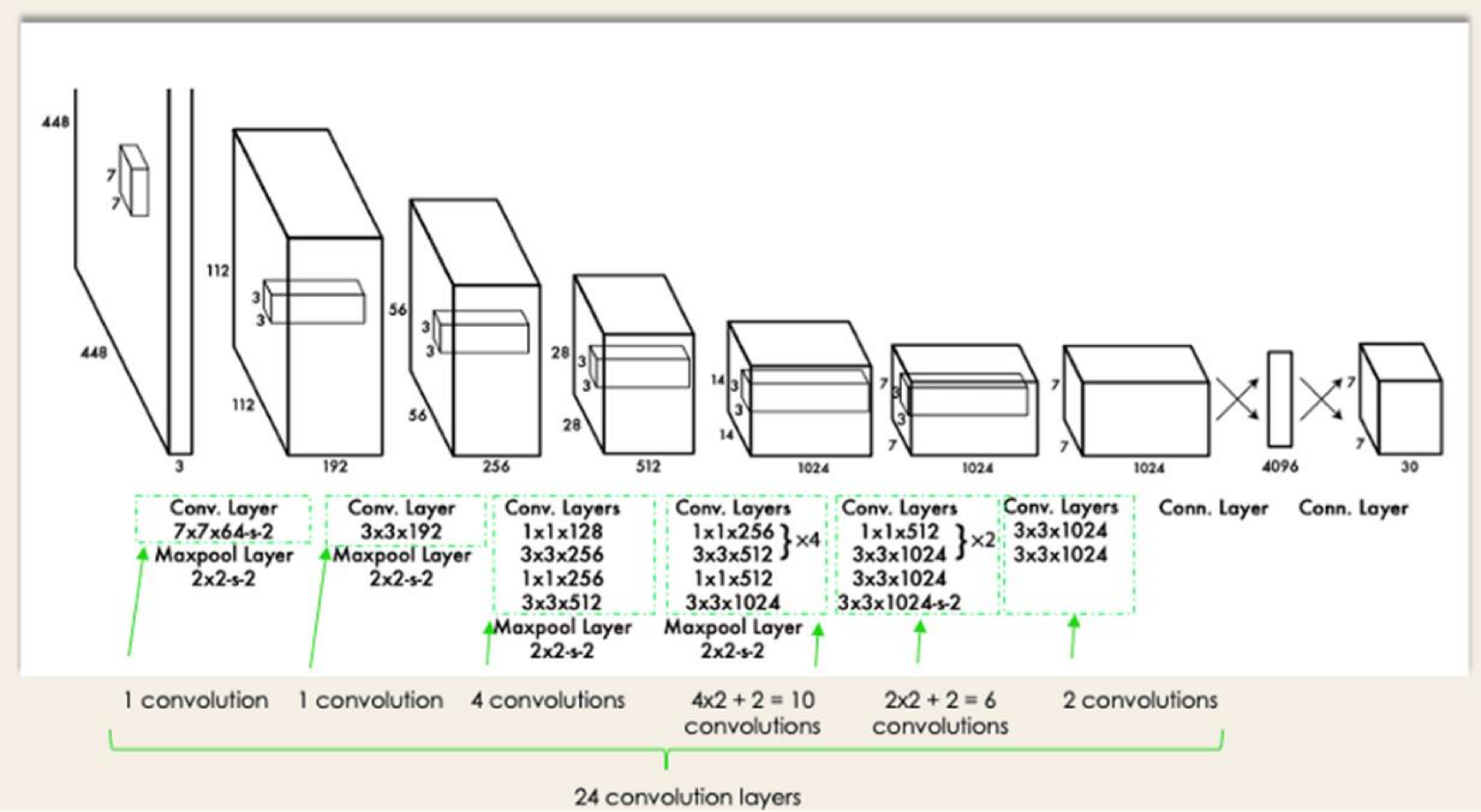
Optionally, you can save the output video or image with the detections overlaid.

Working Model

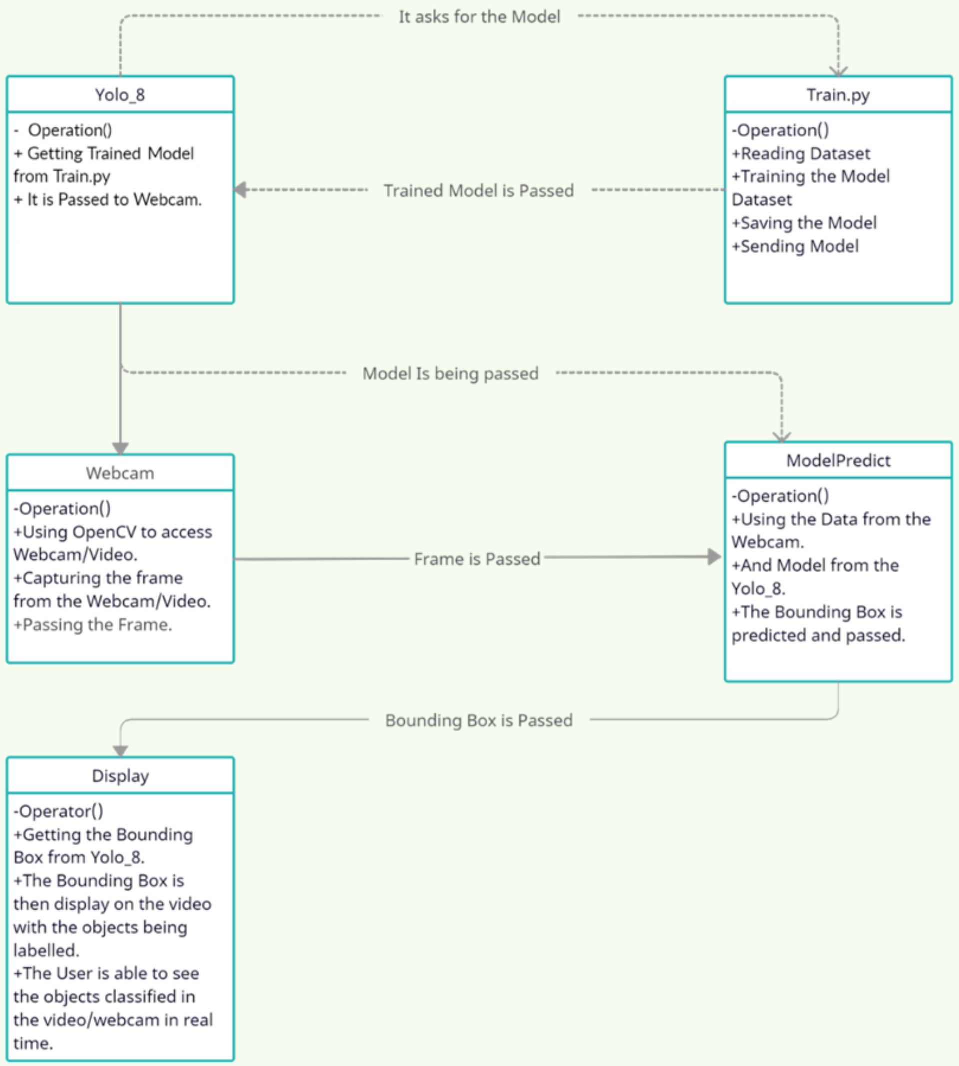
Requirement Analysis

1. **Image and video input:** The system should also be able to handle various file formats and image resolutions.
2. **Accuracy :** It **s**hould be predetermined on set of evaluation metrics such as precision, recall, and F1 score.
3. **Object detection speed**: Should be able to process the input images and videos in real-time ornear-real-time, with a minimum frame rate.
4. **Object detection scalability:** Should be able to handle large-scale object detection tasks efficiently
5. **User interface:** Should have a user-friendly interface that allows users to interact with the system.
6. **Integration:** Should be easily integrable with other software systems and hardware devices.
7. **Hardware requirements**: Should be designed to operate on the appropriate hardware.

Technical Diagram



UML Diagram



It asks for the Model
Yolo_8
- Operation()
+ Getting Trained Model from Train.py
+ It is Passed to Webcam.
Train.py
-Operation()
+Reading Dataset
+Training the Model Dataset
+Saving the Model
+Sending Model
Trained Model is Passed
Model Is being passed
Webcam
-Operation()
+Using OpenCV to access Webcam/Video.
+Capturing the frame from the Webcam/Video.
+Passing the Frame.
Frame is Passed
ModelPredict
-Operation()
+Using the Data from the Webcam.
+And Model from the Yolo_8.
+The Bounding Box is predicted and passed.
Bounding Box is Passed
Display
-Operator()
+Getting the Bounding Box from Yolo_8.
+The Bounding Box is then display on the video with the objects being labelled.
+The User is able to see the objects classified in the video/webcam in real time.

Working Module

- **Image input:** The first step entails entering a frame from the video or webcam. Any size and resolution are possible for the Frame.
- **Pre-processing:** As part of the pre-processing, the image is fixed in size and the pixel values are normalised.
- **CNN:** After the image has been pre-processed, it is run through a CNN.
- **CNN backbone:** It generates a number of feature maps at various scales to represent various levels of abstraction.
- **Anchor Boxes:** YOLO creates a set of potential bounding boxes for each grid cell in feature maps using anchor boxes.
- **Bounding box:** The bounding box prediction made by YOLO includes the objectness score and the bounding box coordinates in relation to the grid cell.
- **Non-maximum suppression:** YOLO uses non-maximum suppression (NMS) to eliminate overlapping bounding boxes before choosing the most certain bounding boxes for each object.
- **Output:** A set of bounding boxes with associated class probabilities and objectness scores make up the final output of YOLO.

Attained Deliverable

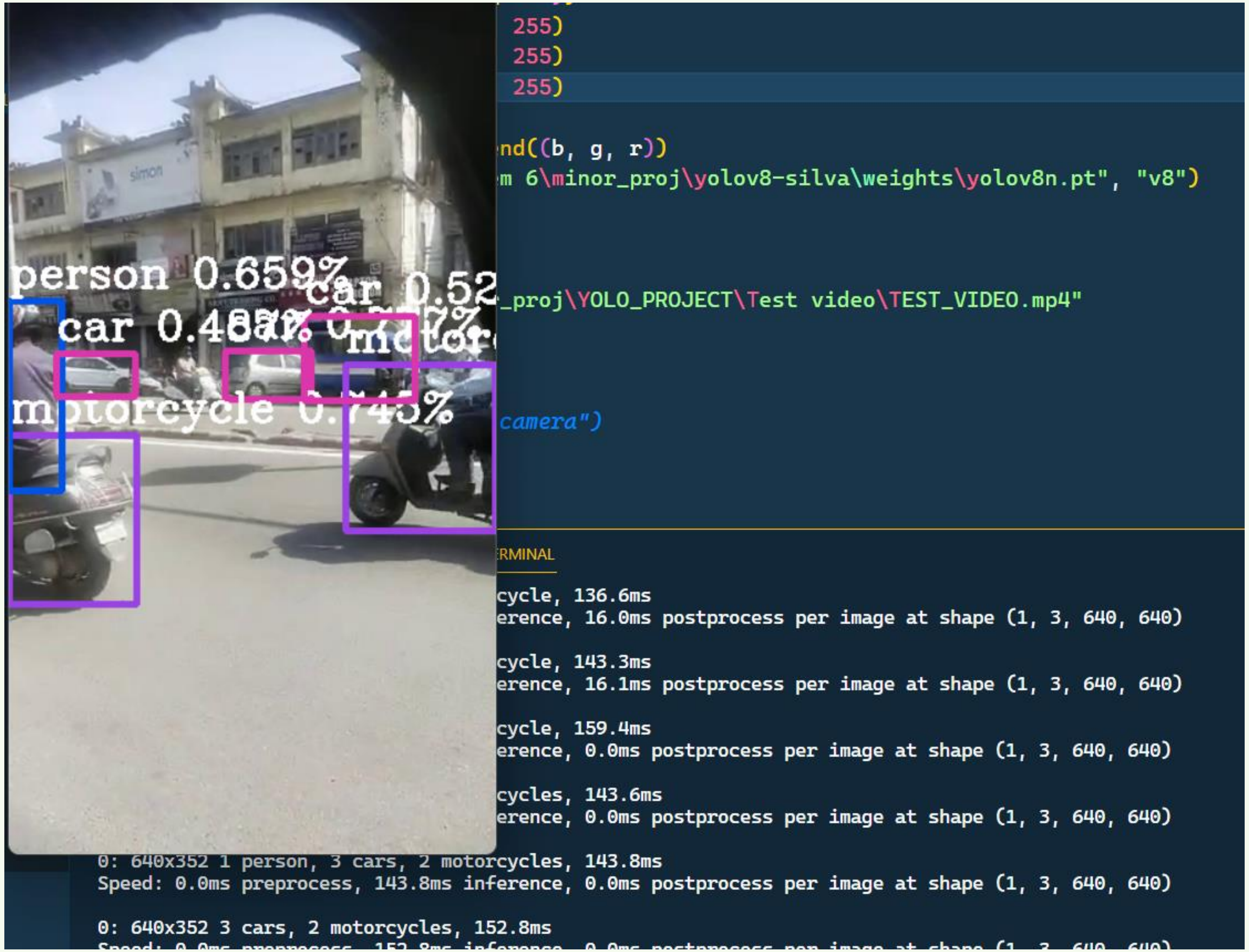
- **Custom Dataset -** A custom dataset is created, which includes images and annotations of objects specific to the object detection task. This dataset is used to calibrate and train the YOLO model.
- **Trained YOLO Object Detection Model -** The main deliverable of the project is the YOLO object detection model, which is trained on the custom dataset to detect objects accurately in real-time.
- **Software Implementation -** The YOLO object detection model is implemented using software, which entails incorporating the concept into an object detection program or system.
- **Technical Report -** The project's approach, including the datasets used, the model architecture, and the evaluation measures, is documented in a technical report. The report also discusses the project's outcomes and constraints.
- **Presentation -**The goals, approach, and outcomes of the project have been compressed into a presentation. To give a general summary of the project's progress and results, the presentation is given to the project stakeholders.

Results-

A box will appear where the video from the webcam is shown. In that video, the program will identify the objects in the video. The objects will be surrounded by a rectangle boundary with the percentage of what the program thinks that object is.

We press “q” key to stop the program.

Results-

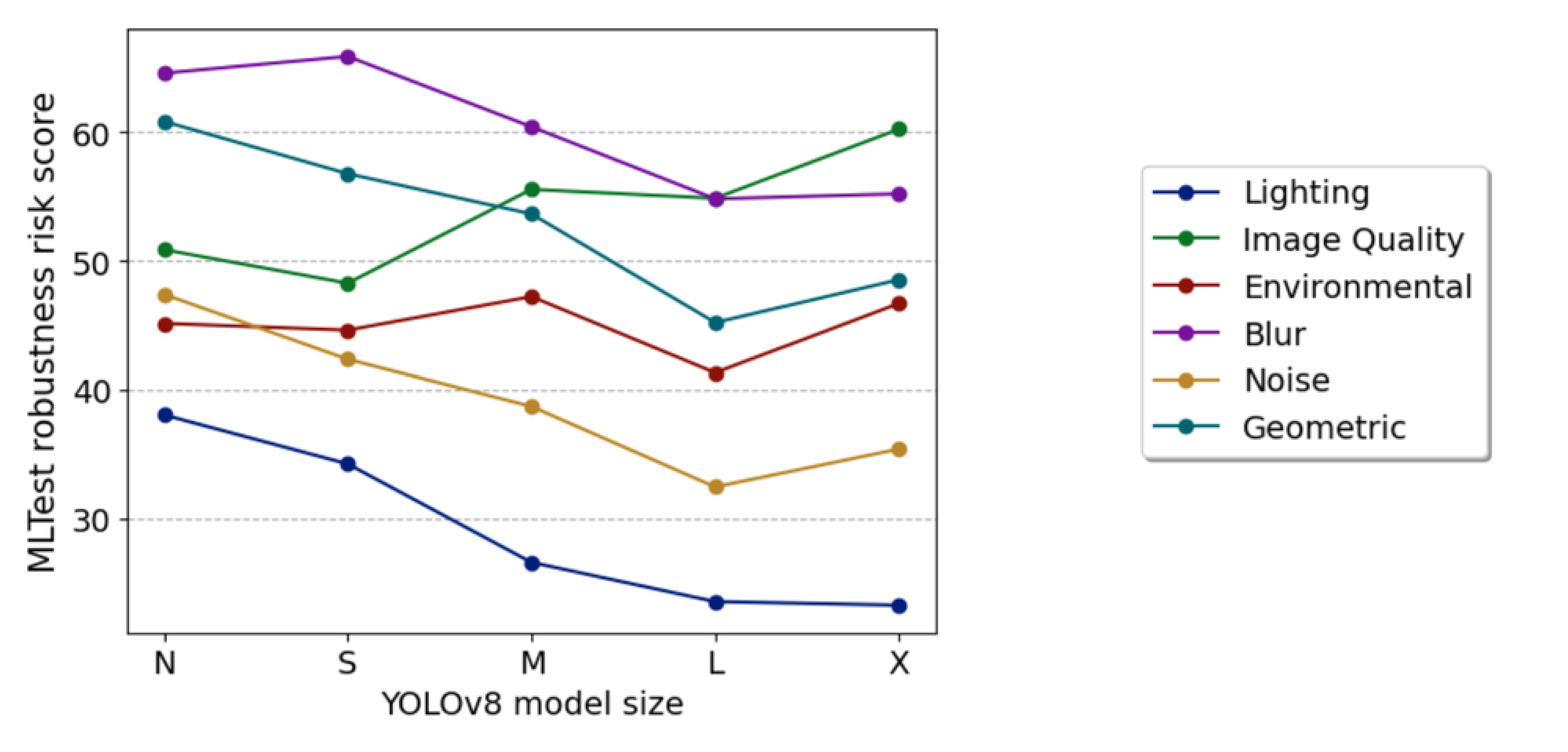


person 0.659%
car 0.52
car 0.48
motorcycle 0.745%
255)
255)
255)
nd((b, g, r))
m 6\minor_proj\yolov8-silva\weights\yolov8n.pt", "v8")
_proj\YOLO_PROJECT\Test video\TEST_VIDEO.mp4"
camera")
RMINAL
cycle, 136.6ms
erence, 16.0ms postprocess per image at shape (1, 3, 640, 640)
cycle, 143.3ms
erence, 16.1ms postprocess per image at shape (1, 3, 640, 640)
cycle, 159.4ms
erence, 0.0ms postprocess per image at shape (1, 3, 640, 640)
cycles, 143.6ms
erence, 0.0ms postprocess per image at shape (1, 3, 640, 640)
0: 640x352 1 person, 3 cars, 2 motorcycles, 143.8ms
Speed: 0.0ms preprocess, 143.8ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)
0: 640x352 3 cars, 2 motorcycles, 152.8ms

Testcase-

- **Functionality testing:** Test cases can be created to make sure that every component of the software or system is functioning as it should. Data input is properly validated, the system is capable of handling various data types, and data is correctly stored and retrieved.
- **Performance testing:** Test cases can be created to make sure that the system or software operates as expected under various dataset conditions. Testing the system's response time, efficiency, scalability, illumination of surroundings, and camera quality can be a part of this.
- **Testing for compatibility:** Test cases can be created to make sure that the software or system is compatible with different versions of software, libraries, and programming languages.

Outcome Graphs-



Conclusion

Justification of Objectives

Our project of YOLO Object detection is a real-time object detection algorithm as it is much faster compared to other algorithms while being able to maintain a good accuracy . The YOLO network understands generic object representation, however the precision is limited for nearby and smaller objects due to spatial constraints.

Overall, YOLO is a popular approach for real-time object recognition because to its speed and accuracy.

Comparative Study

YOLO (You Only Look Once) and SSD(Single Shot Detector) are two popular real-time object detection algorithms. Here is a comparative study of both:-

1.Architecture: While SSD employs two separate networks, one for object detection and the other for classification, YOLO uses a single neural network that completes both tasks in a single pass.

2.Speed: YOLO performs detection in a single pass through the network, whereas SSD requires multiple passes, making it faster than SSD. Real-time performance on YOLO can reach up to 45 frames per second, compared to 20 to 30 frames per second on SSD.

3.Accuracy: Although YOLO and SSD are both good object detection algorithms, the accuracy of each algorithm varies with the dataset and the particular configuration being used. YOLO generally outperforms SSD on large objects, while SSD outperforms YOLO on small and medium-sized objects.

4.Training: YOLO is simpler to train than SSD because it optimises both object detection and classification using a single loss function, whereas SSD uses different loss functions for each task.

Future work

Here are some potential areas which we could consider for future work for YOLO object detection -

- **Object detection in video -** While YOLO can identify objects in individual frames of video, it cannot follow them across frames.
- **Real-time implementation -** Although YOLO is intended to be a quick and effective object detector, there is still potential for growth in these areas. The implementation of YOLO for real-time applications, such as driverless vehicles or surveillance systems, may be one promising field for future research.
- **Multi-object tracking -** The practise of following many objects across time and determining their interactions is known as multi-object tracking.
- **Adversarial attacks -** Malicious alterations to input data can lead deep learning models to behave improperly. Future research into YOLO's vulnerabilities to adversarial attacks and the creation of defences against them could be profitable.

Reference

[1]https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.datacamp.com%2Fblog%2Fyolo-object-detection-explained&psig=AOvVaw0cytg0YSptuh6Vj_otDfiW&ust=1676485548854000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCliu8b7RlfOCFQAAAAAdAAAAABAR

[2] Juan Du , "Understanding of Object Detection Based on CNN Family and YOLO", Citation Juan Du 2018 J. Phys.: Conf. Ser. 1004 012029

DOI 10.1088/1742-6596/1004/1/012029

[3]Aleksa .et.al "The Real-Time Detection of Traffic Participants Using YOLO Algorithm" , Corovic, Aleksa & Ilic, Velibor & Đurić, Siniša & Marijan, Mališa & Pavkovic, Bogdan. (2018). 10.1109/TELFOR.2018.8611986.

[4] Akila S .Et.Al "Indoor And Outdoor Navigation Assistance System For Visually Impaired People Using Yolo Technology" P-Issn: 2395-0072,International Research Journal Of Engineering And Technology (Irjet)

[5]https://www.academia.edu/75269169/You_Only_Look_Once_YOLOv3_Object_Detection_and_Recognition_for_Indoor_Environment

[6] https://ieeexplore.ieee.org/document/8833257

Thank You