# University of Maryland

### ENPM808X

# PROJECT PROPOSAL

# **Human Detector and Tracker**

Abhinav Garg 117553000 Naveen Mangla 117376944 Mahima Arora 117530377



# **Proposal Details**

#### Overview

In this project, we will be working on *Human Detector and Tracker*. To achieve this, we plan to use the pre-trained model Yolo-V5 (as suggested in the lecture). We will freeze the last layer of the Yolo neural network so that it only detects humans present in the camera image. To achieve this goal, we will implement four classes, in which the master class will be **Human-Detector**. We will import a pre-trained neural network Yolo-V5. All the image manipulation (resize, grayscale) would be handled by image class which will be using **OpenCV**.

#### Scope

We are doing this to deliver the required module to Acme Robotics who needs it so that its robots can avoid human obstacles in its path as well as track the people in its surrounding and to know the location of humans with respect to the robot.

#### Quad Chart:

#### **Human Detector and Tracker**

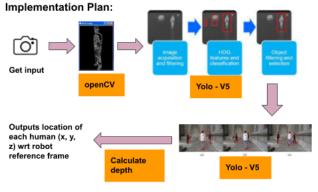
PI: Mahima Arora, Naveen Mangla, Abhinav Garg

#### Overview:

- We aim to develop a Human detecting and tracking module for Acme Robotics. The module will detect multiple humans in a video and assign different IDs and colors to each person and output the location(x, y, z) of each person with respect to the robot frame. This module will also help track the humans as they change their position.
- We are doing this to deliver the required module to Acme Robotics who needs it so that we can avoid human obstacles in the path of the robot as well as track the people to know the exact location of the humans..

#### Methodology:

- We are using libraries such as YOLO V5 for human detection and tracking, image manipulation using OpenCV, standard C++ libraries for input output and vector library to store image pixel output.
- The aim is to filter out human label from the given Yolo-V5 available and detect as well as track.
- Risk
  - New to using Yolo-V5 being used with C++
  - Output video might have low resolution
  - Computation might be high depending on the GPU specifications of the computer



Task	Date
Create Stubs	12/07/22
Implementing Yolo class and image class	16/07/22
Test cases for image class	17/07/22
Merging Yolo and image class in GitHub and test	21/07/22
Feeding image to Yolo Network Testing human detection through video file	23/07/22
Depth calculation for human and final testing	27/07/22

Figure 1: Quad Chart

# Methodology

Our program will consist of three classes, listed as:

**Image:** This class will incorporate all the image related functions, such as *reading*, *resizing*, *visualising* and other modification.

**Human Detector:** This is the linking class for the other two classes, This class will take Image and Yolo as input and merge both of them to get desired human detection. The transformation will take place in this class as well.

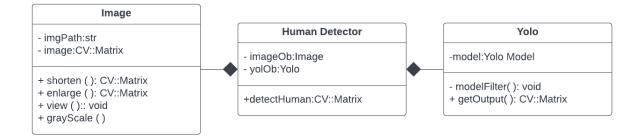


Figure 2: Class Diagram

**Yolo:** This class will take care of all the Yolo-V5 related functions, such as *calling model, filtering* for human detector and giving detected output.

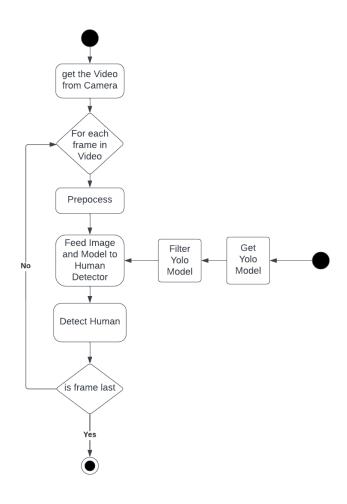


Figure 3: Activity Diagram

## Click here for video