Human Detector and Tracker

Proposal

Introduction:

Detection of objects or things in an image is not so trivial for machines as it is for Humans. The process of identification and localization of objects in images is called Object Detection. Object Detection has a wide range of applications in the field of computer vision. In this project we propose to develop a Human Detector and Tracker in C++ using a Monocular Camera which computes the location of Human obstacles which can be directly used in the Robot's frame of reference.

Objective:

To develop a human detector and tracking system which can track and detect human objects in images and live camera feed with proper C++ coding and documentation practices.

Project Organization:

The project development is planned to be completed in two sprints as per the Agile methodology. As a part of this project, we will be tracking all the tasks with a backlog table and all the tasks are outlined in the iteration backlog for every iteration. Development shall follow pair programming method where the driver and navigator roles and the code maintainer roles interchange each sprint.

Assumptions:

- All the obstacles i.e Humans are on the flat ground.
- Height of every human object fixed to an average height of 1.7m.
- Object detection algorithm is pre-trained on the COCO dataset.
- The camera matrix is assumed for the calculation of object location in the robot's frame.

Procedure:

In this project, to achieve end to end object detection we will use the YOLOv5**)** (You Only Look Once) neural network trained on the COCO dataset. Yolov5 is a fast convolutional neural network which has Binary cross entropy and logit loss as loss function. The version 5 of YOLO is faster and easier to use than the previous versions. The entire code development is planned to be carried out in C++ using the OpenCV and other imaging libraries.

The process starts from the loading the model into the program, following that the image needs to be converted to BLOB (binary large object). The next step would be setting up the weights and initializing the network input. Then the neural network using forward propagation outputs the detected values. Now we must process the output values of the

network and check for genuine results. Once all the bad detections are rejected based on the class score, we will be sorting the detections and perform non-maximum suppression on the results to select the correct bounding box.

After applying NMS (Non-Maximum Suppression) the coordinates of the object given out by the network are then transformed into robot's frame of reference using a transformation matrix.

Finally, we will add labels according to the classification and showcase the results.

Project Timeline:

TASK	DATE
SPRINT 1	
Process input frames in correspondence to the YOLOv5 model	10/15/22
Computing the prediction weights for the processed frames with the YOLOv5 weights from the pre-trained model.	10/16/22
Modify the predicted weights to get output of Class ID: 1 i.e. Humans	10/17/22
Complete developer level documentation for sprint 1	10/18/22
SPRINT 2	
Get pixel co-ordinates and draw bounding boxes to identify humans	10/20/22
Apply non-maximum suppression and get one box per human.	10/22/22
Transform the pixel co-ordinates to robot's frame.	10/23/22
Complete developer level documentation for the whole project	10/25/22

Deliverables:

The deliverables will include a new module for detection & tracking in C++ using high-quality software engineering practices, class and activity diagrams, an up-to-date GitHub repository with complete documentation and unit tests integrated with ICOV package.

References:

[1] You Only Look Once: Unified, Real-Time Object Detection by Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi.

[2] YOLOv4: Optimal Speed and Accuracy of Object Detection by Alexey Bochkovskiy et al.