Objectives

In this assignment, the goal is to fine-tune the pre-trained YOLOv7 model on a custom dataset. After finishing this assignment, you will be familiar with the concept of Object Detection/Recognition in a Transfer Learning framework which enables you to fine-tune large-scale models on simpler downstream tasks.

At the end of this assignment, you need to submit to github repo

- Your code, the code need to be clear and include comments:
- Sample output images and videos
- A short report addressing your design and the questions asked in the assignment.
- Your trained model.
- Instructions to run code.

Problem Statement

Safety of the workers is one of the most important things in working areas. Especially for the people working in the construction zones, it becomes even more crucial. Companies and workers will suffer from the huge costs caused by not wearing the Personal Protection Equipment (PPE) while working in these areas. Therefore, real-time surveillance of the construction areas is necessary in order to make sure everybody is safe. In this assignment, we focus on the safety helmet which is one of the most important PPEs.

YOLOv7 is one of the fastest and most accurate object detection models used widely in real-time applications. The output of YOLOv7 for an example image is shown below:

However, the original model is not trained to detect the personal protection equipment or if a person is wearing them or not. In this assignment, you will train YOLOv7 in a transfer learning framework in order to detect the safety helmets of the workers and determine if they are wearing them or not in the construction zones.

Part-I: Install and Examine the YOLOv7

Clone the official PyTorch version of YOLOv7 repository from their GitHub page in the following link:
https://github.com/WongKinYiu/yolov7

Make sure to install all of the dependencies required for running YOLOv7 experiments. You also need to find the latest checkpoint of the pretrained YOLOv7 and load the weights to the model.

1. Run the detection (inference) on the images in the test set given to you and generate and save the output images which include the bounding boxes around the objects and their labels as well as their confidence scores.
2. What are the main challenges of the YOLOv7 for detecting the safety helmet and detecting the workers who are not wearing them?

Part-II: Data Preparation

In order to fine-tune the YOLOv7 model for the helmet detection task, examples of dataset of safety helmet objects is provided in the following link:

Dataset 1: https://www.kaggle.com/datasets/andrewmvd/hard-hat-detection?select=images
Dataset 2:Hard hat workers Dataset | MakeML - Create neural network with ease

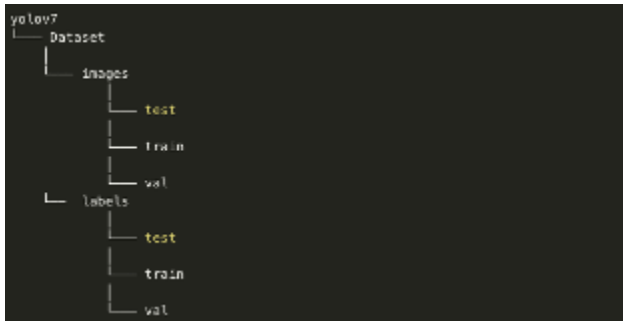
You are given multiple examples of Datasets but you can use and you are entitled to use other Datasets

The first step in fine-tuning a deep learning model is to convert the dataset format to the original format they have been trained with. The dataset provided to you is in PASCAL-VOC format, which is a very common object recognition format. However, YOLO has its own data format and therefore, we need to convert it to the YOLO style. An example code that can be used for transferring PASCAL-VOC format to YOLOv7 is provided in the following link:

https://towardsdatascience.com/convert-pascal-voc-xml-to-yolo-for-object-detection-f969811ccba5

1. Use the code provided above to transform the data format to YOLOv7.
2. Randomly split the dataset into 80% training, 10% validation and 10% test sets and save them in separate folders.

   The dataset folder structure should look like the following:



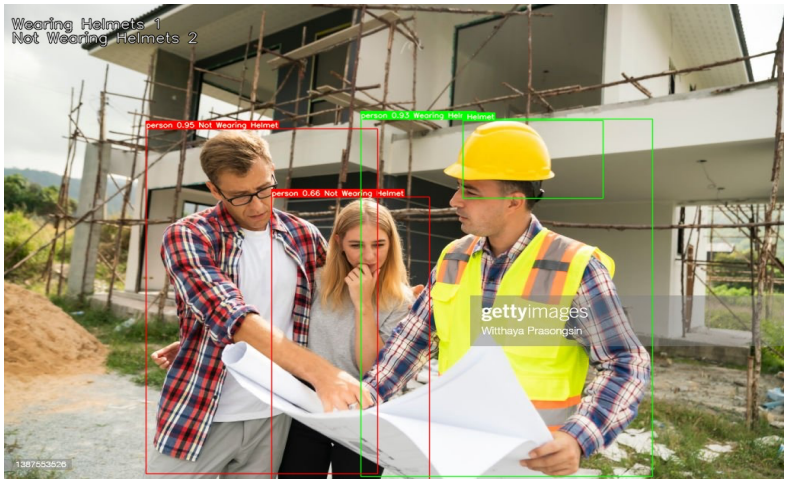
Part-III: Transfer Learning

1. Fine-tune (train) the model using the new helmet dataset so that the model is able to detect all of the safety helmets in the images. After training is finished, apply your model to the images in your test set and save the output images in a new folder. Make sure the test images are not used during the training process (your model will be evaluated using a secret test set not provided to you!). The saved images should include bounding boxes around the workers and **their helmets (if it was detected)** with labels indicating whether he/she is wearing a helmet or not.

Notice the image has 2 bounding boxes, the first identifying the person the second identifying the hardhat within this bounding box, the person identified should have a bounding box around them either Red if no helmet was detected within the bounding box area and Green in case there was a helmet within the bounding box area. You are required to have the same output.The image also has the description of the frame top left of the image.

Original Image

Expected Output



2. How many YOLOv7 models do you need to do these tasks? Can this be done using only one single model or more?
3. What are the challenges in this task? What if the people are far away from the camera and appear tiny in the images?

4. What if a worker holds his/her helmet in his/her hands?

Part-IV: Video Analysis

1. Using the model trained in the previous step, create a system that analyzes a video clip of the workers and generates and saves a video clip including all of the bounding boxes and labels (people wearing/not wearing helmets and the helmets).
2. Modify the previous system so that it raises an alarm(ex a beep sound) whenever it detects a worker not wearing a helmet. The user should be able to press any key to continue reviewing the rest of the video.
3. Report the number of people in the video at each frame wearing and not wearing their helmets in a separate text file. For people not wearing their helmets, the report should also include the time of the event.
4. Calculate and report the speed of your system in terms of frame per second. In other words, how many frames of the video it can process (detects all of the objects) in one second.

Part-V: Evaluation Rubrics

Criteria	Meets Expectations	Does Not Meet Expectations
Part-I Original YOLOv7 (10%)	The code detects all the objects in the test images with bounding boxes and labels and confidence scores. The question is answered	The code does not run correctly or the objects are not detected properly. Answer is missing in the report
Part-II Data Preparation (20%)	The dataset is transferred into YOLO format Dataset iis splitted into train, val and test sets according to the structure provided.	The dataset is not converted to YOLO format Dataset is notl organized properly or the test/val sets are mixed with the train set.
Part-III Transfer Learning (40%)	All the output images in their test set are provided where the helmets are detected with bounding boxes and labels and confidence scores. The workers without helmets are identified. Their code runs for the secret test set and detects the helmets and whether the workers are wearing them. The questions are addressed in their report	The test images are missing or the helmets in them are not detected or the workers without helmets are not identified. The code doesn’t generate results similar to the images they provided from their experiments (bad generalization) Answers are missing
Part-IV Video Analysis (30%)	A video including all the bounding boxes and labels are provided from their model. Their code should raise an alarm when a worker is not wearing a helmet. The number of workers with and without helmets in each frame is reported on the frame and also a text file with each frame description .	Output video is not provided or the workers with and without helmet are not detected property. Inference speed is not

	Show the inference speed of the system in frame per second. The code is submitted and it runs with the secret test video and generates the same results.	calculated properly. The code is not provided or it does not run properly.

Good Luck!