WOBOT HACKATHON 4.0

Safety Equipment Detection System

1. Stage 1: Evaluation – Deep Learning
    - In this project, I made use of YOLO v5 model for model training. YOLO is an object detection architecture which gives state of the art accuracy on object detection and classification problems. It is fast, efficient and requires minimal GPU support.
    - Transfer learning approach was used to train and test the model.
    - The given annotations were converted from Pascal VOC (xml) format to YOLO (txt) format using the free functionalities provided by Roboflow on doing the same.
    - The entire code was run on Google Colab. The code script is present in *hackathon4.0.ipynb* file.
    - After the model was trained, the provided YouTube video was processed to output detections and prediction confidence scores.
    - *video.mp4* file is the processed YouTube video. Its size on disk is 692 MB.

2. Stage 2: Evaluation – Finding the colour
    - For every detected hardhat in the Stage 1 problem, we were tasked to ensure the bounding box colour should match the colour of the hardhat itself.
    - This can be done by implementing a foreground-segmentation mask using the colour information from the matched boxes and their surrounding background frame-by-frame.
    - Principal Component Analysis can also be used to estimate the frame pixels that are more likely to belong to the foreground object and then match it with the predicted bounding box.
    - Reference: - *Multiple Frames Matching for Object Discovery in Video*

3. Stage 3: Evaluation – mAP
    - During model training, Tensorboard was used to visualize model training and model related metrics. The same can be found in *hackathon4.0.ipynb* file.
    - After running the model for 100 epochs, a mAP value of about 0.62 was obtained.
    - The same model was used to run inference on the test set of 250 images.
    - The predictions were converted from YOLO (txt) format to Pascal VOC (xml) format using Roboflow again.
    - The predicted annotations are in the *labels* folder in the *testset-inference.zip* file. YOLO v5 model output test set images with bounding box and confidence score superimposed on it. These images can be found in *bboximg* folder in *testset-inference.zip* file.

4. Areas of Improvement

    - The model can be made to run for a greater number of epochs to obtain better results. Fine-tuning hyperparameters and weights may help to get higher accuracy and minimal loss.
    - The model at times is not able to correctly identify the class ‘helmet’ and ‘head’. This is reflected in the test set images and processed YouTube video. Using a greater number of training data images (possibly with a different distribution) may solve this problem.

Submitted by: Deepank Girish (+91 8217582948)