

Start

Import all the necessary modules, libraries and methods.

Prepare the argument parser.

Parse five arguments(input, output, yolo model, threshold and confidence).

Load the yolo class labels in "labelspath"(string).

Load class names in "LABELS"(list) from "labelspath".

Initialize a list of colors, "COLORS" for every class in the YOLO model.

Derive the paths to the YOLO weights and model configuration in "weightsPath" and "configPath" respectively.

Load our YOLO object detector trained on COCO dataset (80 classes).

Determine only the output layer names that we need from YOLO.

Input the video in "vs"(variable) by cv2.VideoCapture.

Initialize our output video "writer" to NONE.

Initialize dimensions of frame as NONE.

Determine the total number of frames in video using cv2.CAP_PROP_FRAME_COUNT and save in "total" variable.

Loop over the frames and read every frame until the end.

Initialize a set, "violate", to store number of violations in a frame.

Grab the dimensions of the current frame i.e., W and H.

Construct a blob from the input frame.

Perform a forward pass of the YOLO object detector, giving us our bounding boxes and associated probabilities in a list, "layerOutputs".

Calculate the time taken to process one frame and save it in "elap" variable.

Initialize four lists("boxes", "confidences", "classIDs", "centroids") to store the information in "layerOutputs".

Loop over "layerOutputs".

Loop over each detections.

Collect classIDs and probabilities from the detection in the ndarray, "scores", for the current object .

From "scores" get the classID and confidence associated with the classID.

Get the index of the class "person" in "personID".

Filter detections for person as object and valid confidence.

Extract coordinates for centroids and dimensions of bounding boxes from "detection" into "box".

Calculate coordinates for top left corner of bounding boxes using centroids and dimensions.

Now append top left coordinates and dimensions in "boxes" list.

Append confidence of the current object in "confidences" list.

Append centroids of bounding boxes in "centroids" list.

Append classID of current detection in "classIDs" list.

Exit the loop on line 22.

Verify that atleast there are two identifiable objects by "if" statement.

Calculate euclidean distance between the centroids of all the objects detected.

Check if any set of centroids have less than the advised distance and add them to "violate" set.

Exit "if" statement on line 4.

Apply non maxima suppression to narrow down detections in 'idxs' ndarray.

Ensure atleast one violation by "if" statement.

Loop over "idxs" after converting it to one dimensional array.

Extract bounding box coordinates with respective centroids.

Assign color to the clasIDs from "COLORS" on line 7.

If current deetection is in "violate", assign color red to the classID i.e., (0,0,255) in BGR.

Draw the bounding boxes and mark the centroids.

Display classID and confidence over respective boxes.

Exit "if" statement on line 35.

Display number of violations i.e., length of set "violate at the bottom of the frame.

Initialize VideoWriter.

Display the time taken by a single frame to process, "elap" on line 20.

Display the time to be taken by all frame using "total" and "elap" on lines 14 and 20 respectively.

Write the frame to output video file.

Clean up and release the pointers(writer and "vs" on line 11).