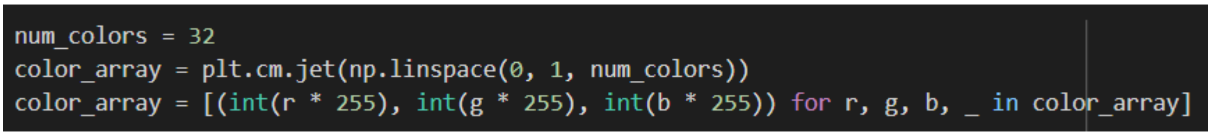
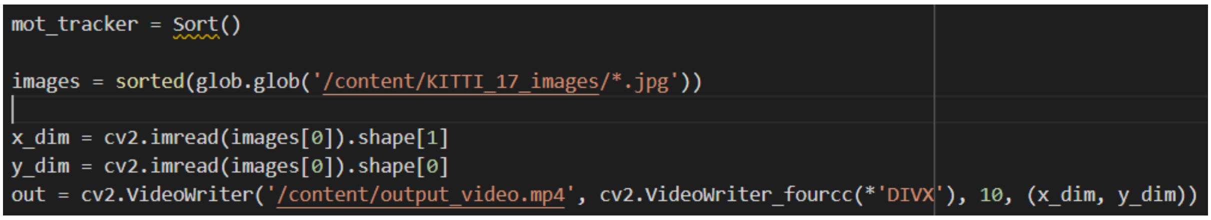


```
num_colors = 32
color_array = plt.cm.jet(np.linspace(0, 1, num_colors))
color_array = [(int(r * 255), int(g * 255), int(b * 255)) for r, g, b, _ in color_array]
```

The Jet colormap from the matplotlib library is used in this part to produce a 32-color array. The colours are transformed into 8-bit RGB tuples, which are then kept in a list called color_array.

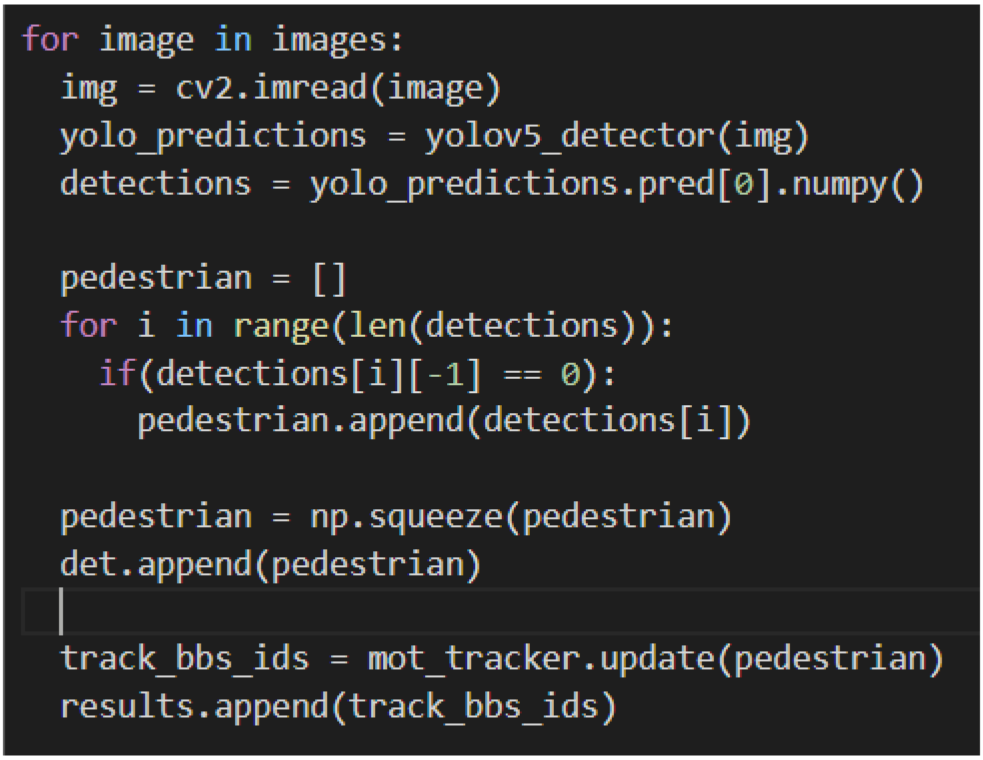


```
mot_tracker = Sort()

images = sorted(glob.glob('/content/KITTI_17_images/*.jpg'))

x_dim = cv2.imread(images[0]).shape[1]
y_dim = cv2.imread(images[0]).shape[0]
out = cv2.VideoWriter('/content/output_video.mp4', cv2.VideoWriter_fourcc(*'DIVX'), 10, (x_dim, y_dim))
```

The SORT (Simple Online and Realtime Tracking) algorithm is created by this line. It is expected that the Sort class is imported and accessible in the code. This component reads every.jpg image file present in the given folder and organises them alphabetically using the glob library. The variable images contains the sorted list of picture file paths. The OpenCV imread function is used to read the dimensions of the first picture in the sorted list in this case. The DIVX codec, a 10-frame per second frame rate, and the input picture dimensions are then used to build a VideoWriter object.



```
for image in images:
  img = cv2.imread(image)
  yolo_predictions = yolov5_detector(img)
  detections = yolo_predictions.pred[0].numpy()

  pedestrian = []
  for i in range(len(detections)):
    if(detections[i][-1] == 0):
      pedestrian.append(detections[i])

  pedestrian = np.squeeze(pedestrian)
  det.append(pedestrian)

  track_bbs_ids = mot_tracker.update(pedestrian)
  results.append(track_bbs_ids)
```

This section's code loops through the sorted images, reads each one using OpenCV, and uses the YOLOv5 detector to find objects (assumed to be available and configured correctly). The detections variable contains the results of the object detection.

Assuming that pedestrians have a class ID of 0, it then removes detections that are not pedestrians and adds them to the det list.

The SORT tracker receives the pedestrian positions after filtering the detections and updating the tracking data. The results list is appended with the tracking results, including the bounding boxes and track IDs.