We have used yolo v3 model to make the classification. You can get more info from <https://pjreddie.com/darknet/yolo/>

1. Press Alt+F12 to open terminal within PyCharm, then write in the command and press enter.
2. scale = 0.00392 for the cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False) function. Scale value for this function is 1/255 = 0.00392.
3. image size = (416,416) for the yolo model default image size is 416x416. You can check that in configuration file(.cfg files in each folder we delivered to you). Although it is used 416x416, when training the network, it automatically changes the image sizes for maximum accuracy.
4. “center\_x = int(detection[0] \* Width)

center\_y = int(detection[1] \* Height)

w = int(detection[2] \* Width)

h = int(detection[3] \* Height)

x = center\_x - w / 2

y = center\_y - h / 2”

This bit of code is used to draw the bounding box around the detected image. W = width, h = height, x,y are the coordinates of upper left corner.

1. In Annotating we don’t crop images. we just mark the relevant image by drawing a box around it. We used “LabelImg” software for annotation. Here is the link <https://github.com/tzutalin/labelImg>.Each image has a text file mentioning the coordinates of the objects. Ex: 1 0.716797 0.395833 0.216406 0.147222
2. We have used all the images for except for about 20 in each class for training. Those 20 is used for testing (this testing is happened in model training process, not a process we do manually). Images in the Test results shown in the doxc file are the images used to test. Since we cant use the same images for training and testing we need to use more images to train the yolo network. with the limited number of images in the dataset we had to use few images for testing. Those are included in each of the folders we delivered to you. There test results are shown in the docx.