# Guided Project Report

## Object Detection and Recognition

Name: Shruti Verma
Course: AI and ML
(Batch 4)
Duration: 10 months

Problem Statement: Implement a simple object detection network using any deep learning framework of your choice for the detection and recognition of COCO dataset.

## Prerequisites

What things you need to install the software and how to install them:

Python 3.8 or higher versions This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not- recognized-as-an-internal-or-external- command/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.8 then run below commands in command prompt/terminal to install these packages pip install -U scikit-learn pip install numpy pip install scipy if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages conda install -c scikit-learn conda install -c anaconda numpy conda install -c anaconda scipy . From Yolo download the Yolo config and weights files.

### Video Link

https://drive.google.com/file/d/1EgHviGbNQL4SR1_yEysIcv4AYmgM0kXt/view?usp=sharing

## Dataset used

Used COCO dataset - common objects in context.COCO is a large-scale object detection, segmentation, and captioning dataset with 330K images and 80 object categories.

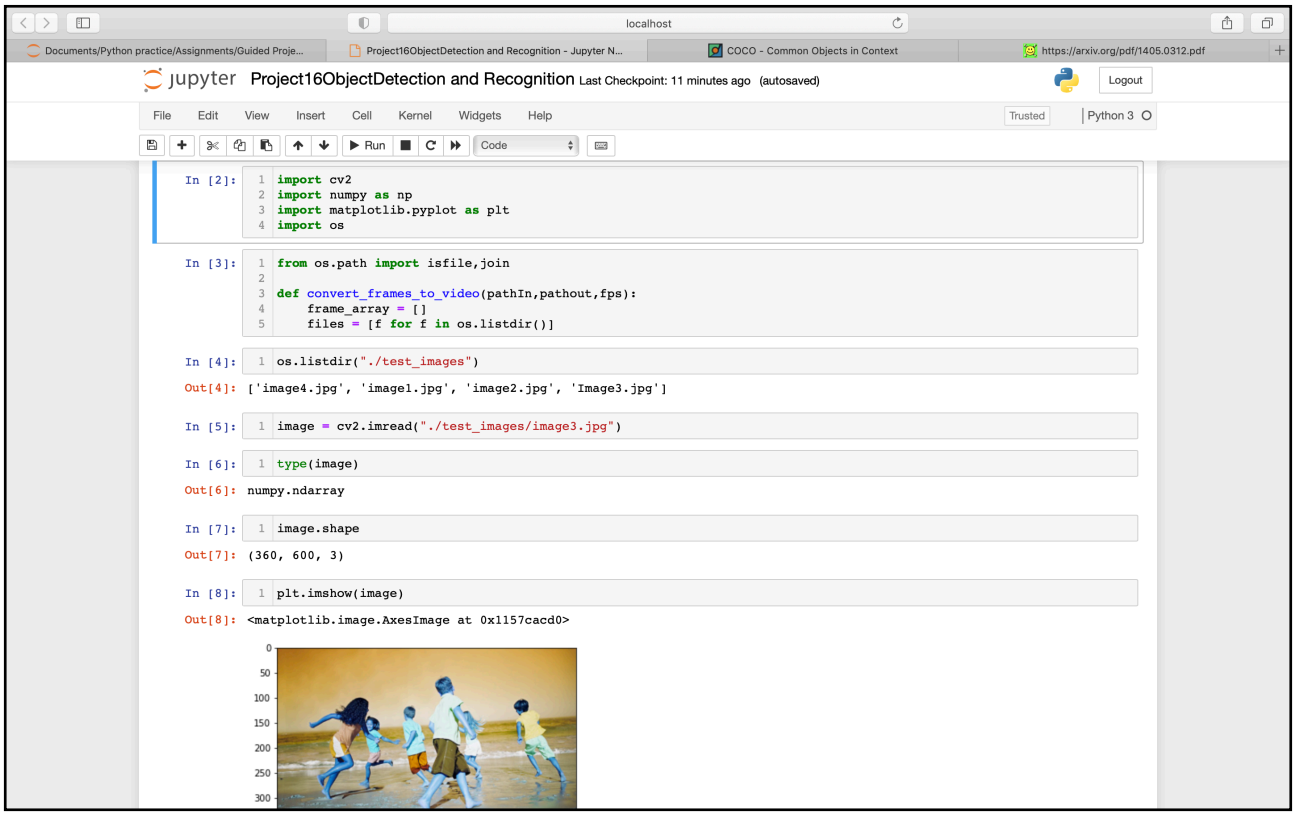
## Method used for detection

- Reading the image3, I have considered 4 images.
- Image preprocessing
- Applying Yolo model

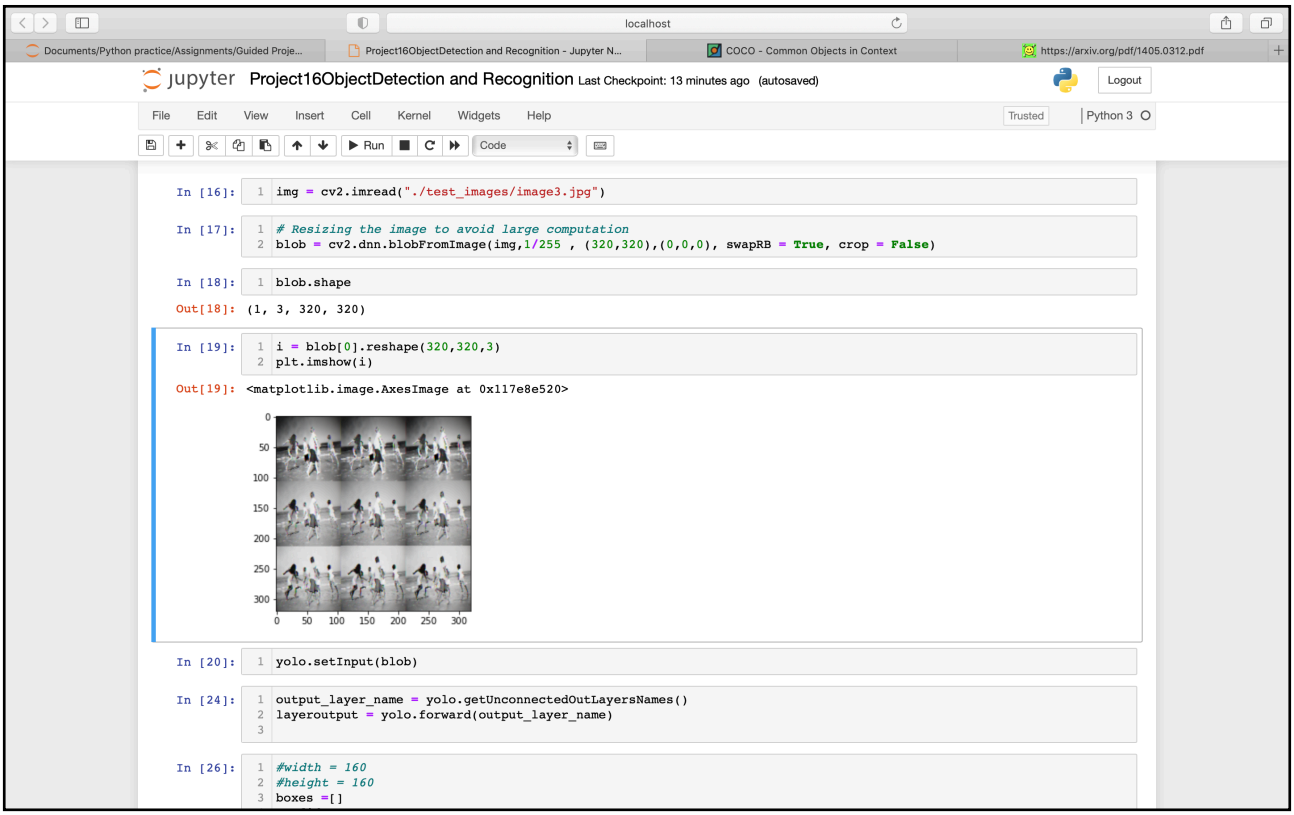
### Importing the libraries and capturing images:

### Importing the necessary libraries and reading the image

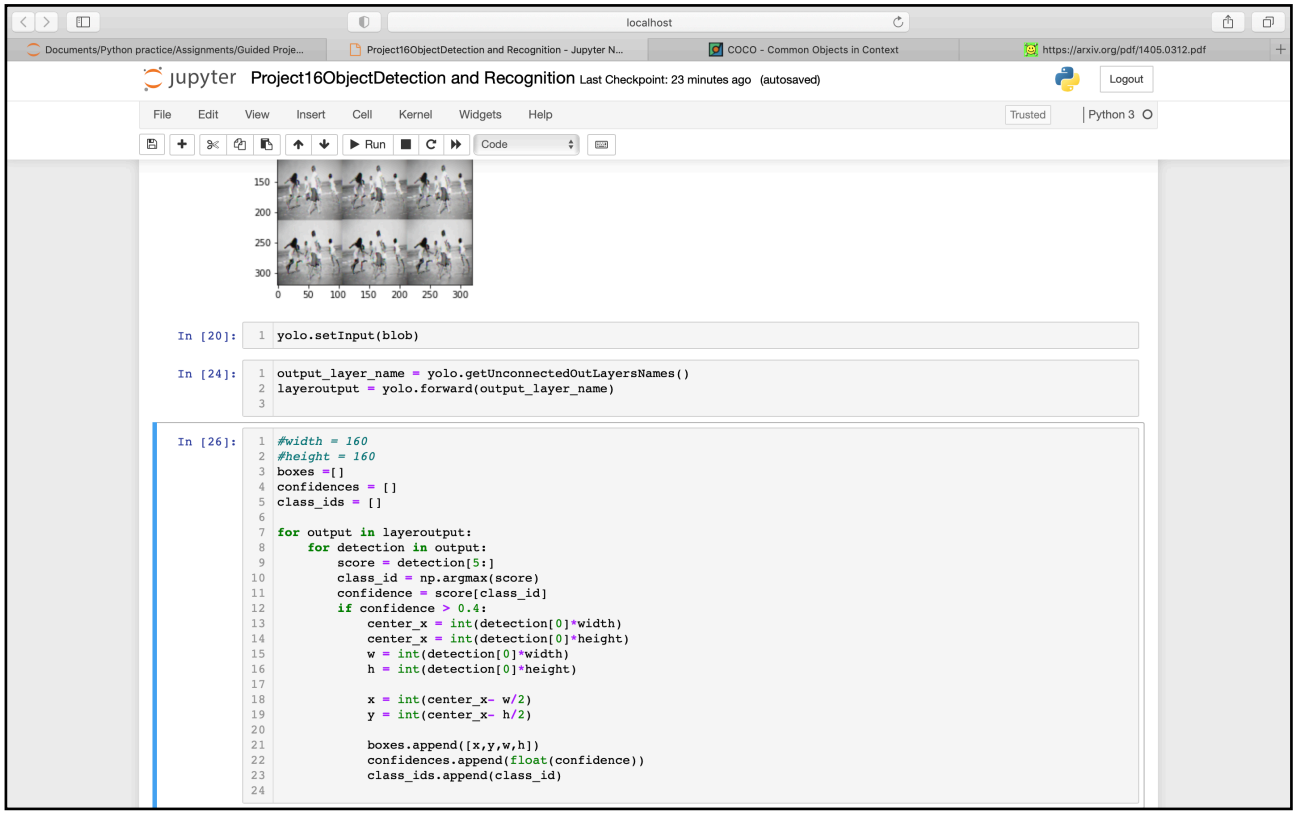
### Splitting the data into train and test and checking the labels and features



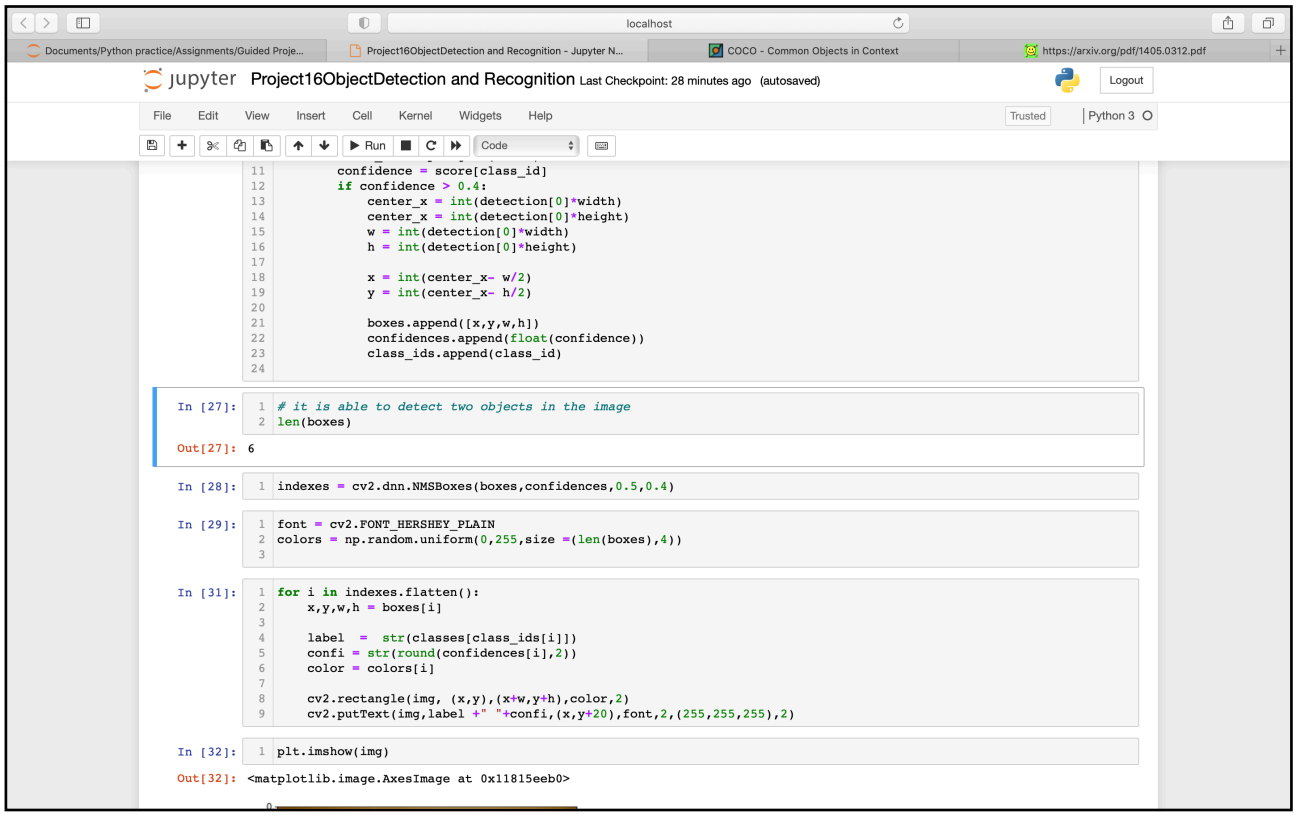
Data processing and resizing the image to 320



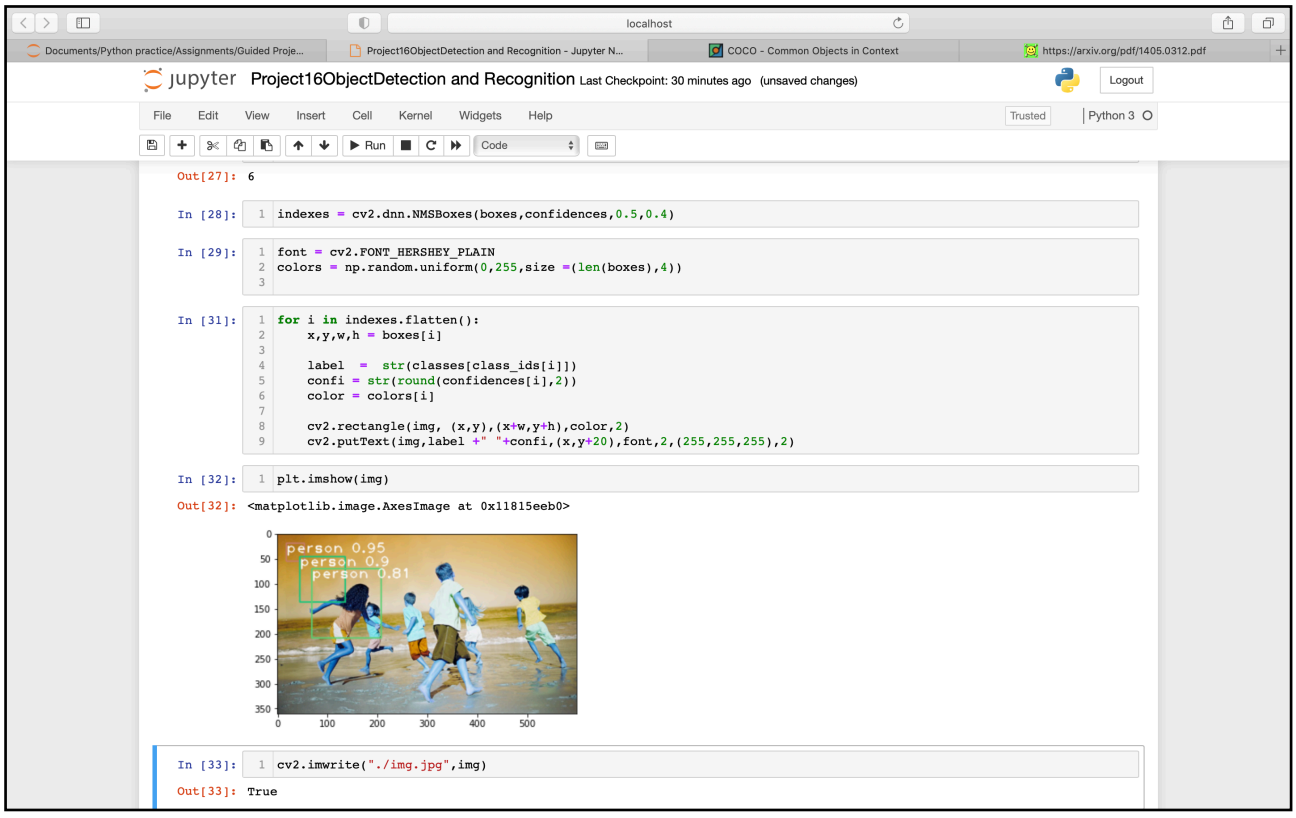

Using Yolo for input , setting the boxes, confidence and class_ids

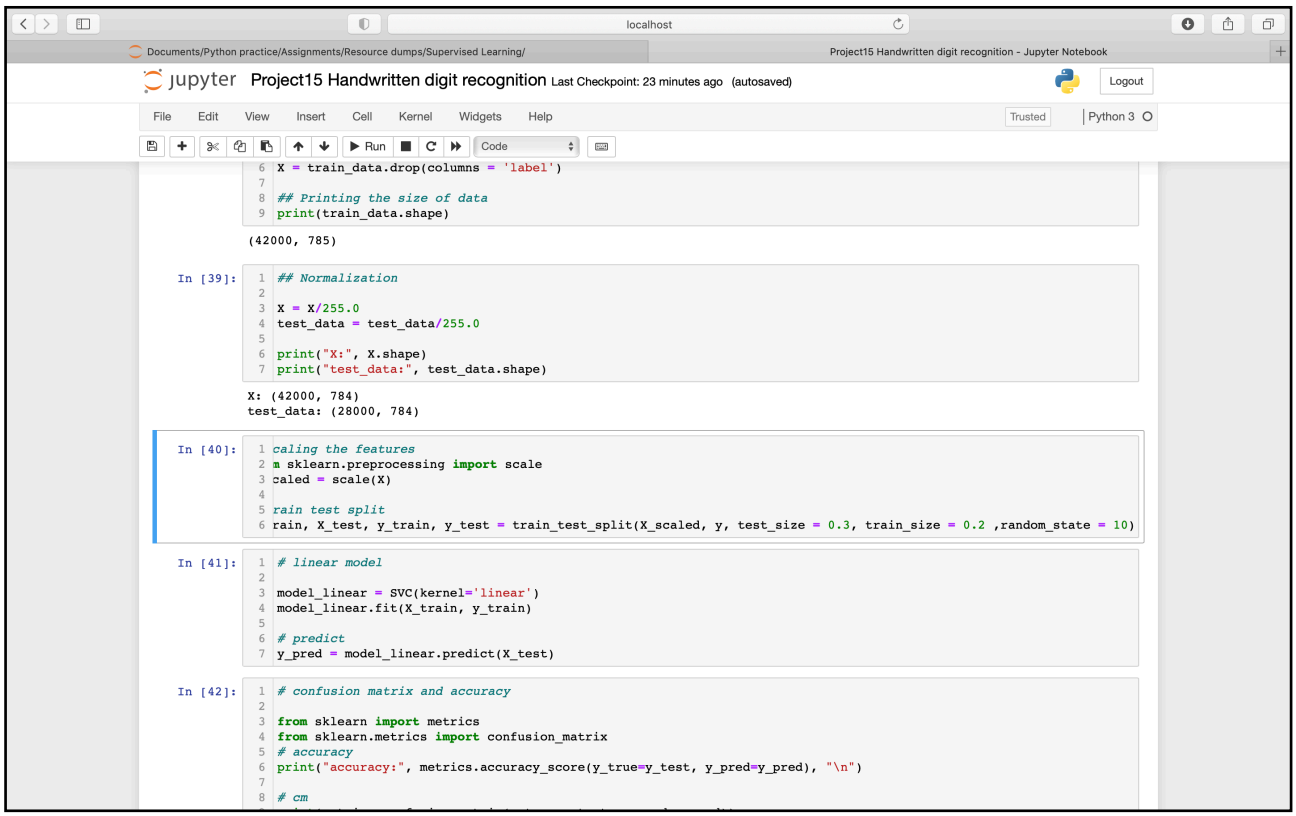


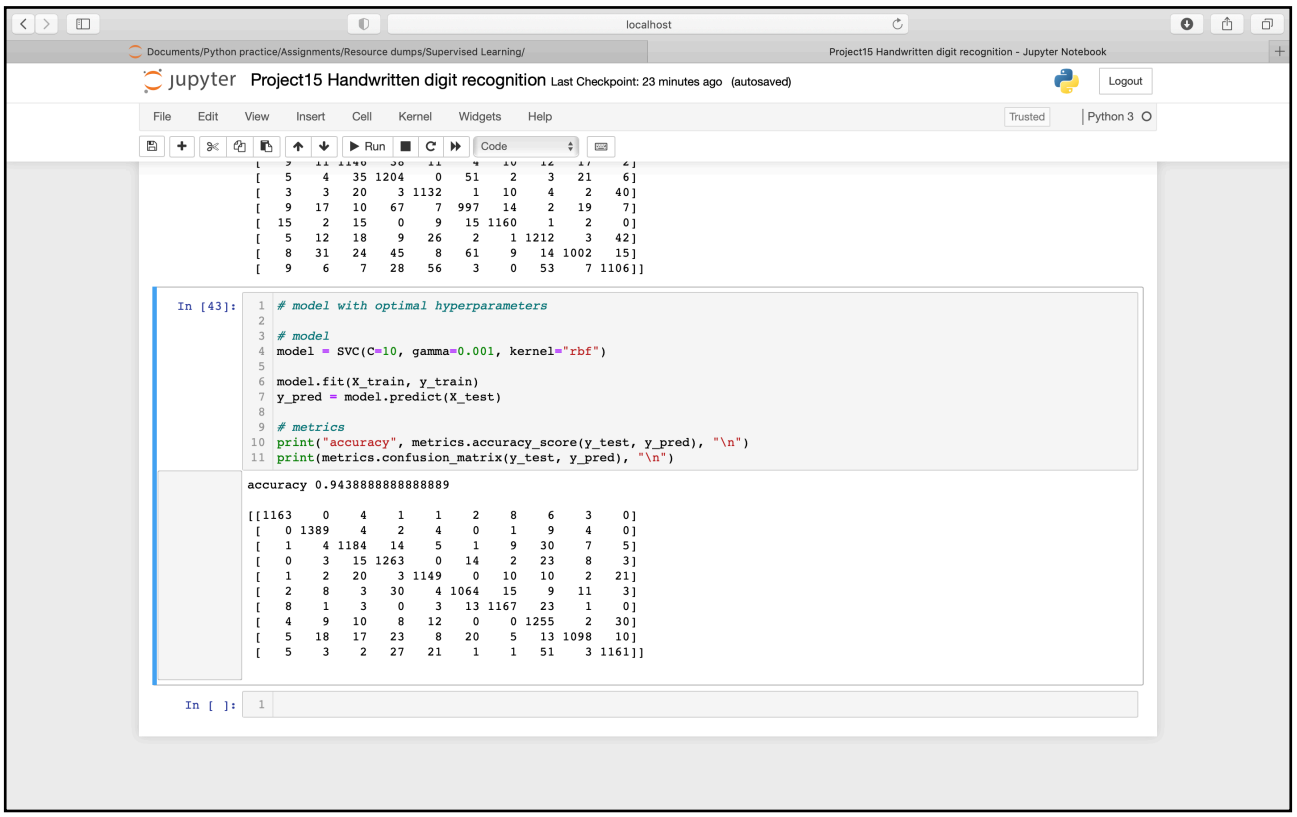
It will detect 6 objects In the given image




Setting the Font and colour of the boxes and then write the image




localhost
Documents/Python practice/Assignments/Resource dumps/Supervised Learning/
Project15 Handwritten digit recognition - Jupyter Notebook
jupyter Project15 Handwritten digit recognition Last Checkpoint: 23 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
X = train_data.drop(columns = 'label')
## Printing the size of data
print(train_data.shape)
(42000, 785)
In [39]:
## Normalization
X = X/255.0
test_data = test_data/255.0
print("X:", X.shape)
print("test_data:", test_data.shape)
X: (42000, 784)
test_data: (28000, 784)
In [40]:
caling the features
m sklearn.preprocessing import scale
caled = scale(X)
rain test split
rain, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size = 0.3, train_size = 0.2 ,random_state = 10)
In [41]:
# linear model
model_linear = SVC(kernel='linear')
model_linear.fit(X_train, y_train)
# predict
y_pred = model_linear.predict(X_test)
In [42]:
# confusion matrix and accuracy
from sklearn import metrics
from sklearn.metrics import confusion_matrix
# accuracy
print("accuracy:", metrics.accuracy_score(y_true=y_test, y_pred=y_pred), "\n")
# cm


localhost
Documents/Python practice/Assignments/Resource dumps/Supervised Learning/
Project15 Handwritten digit recognition - Jupyter Notebook
jupyter Project15 Handwritten digit recognition Last Checkpoint: 23 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
[ 5 4 35 1204 0 51 2 3 21 6]
[ 3 3 20 3 1132 1 10 4 2 40]
[ 9 17 10 67 7 997 14 2 19 7]
[ 15 2 15 0 9 15 1160 1 2 0]
[ 5 12 18 9 26 2 1 1212 3 42]
[ 8 31 24 45 8 61 9 14 1002 15]
[ 9 6 7 28 56 3 0 53 7 1106]]
In [43]:
# model with optimal hyperparameters
# model
model = SVC(C=10, gamma=0.001, kernel="rbf")
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
# metrics
print("accuracy", metrics.accuracy_score(y_test, y_pred), "\n")
print(metrics.confusion_matrix(y_test, y_pred), "\n")
accuracy 0.9438888888888889
[[1163 0 4 1 1 2 8 6 3 0]
[ 0 1389 4 2 4 0 1 9 4 0]
[ 1 4 1184 14 5 1 9 30 7 5]
[ 0 3 15 1263 0 14 2 23 8 3]
[ 1 2 20 3 1149 0 10 10 2 21]
[ 2 8 3 30 4 1064 15 9 11 3]
[ 8 1 3 0 3 13 1167 23 1 0]
[ 4 9 10 8 12 0 0 1255 2 30]
[ 5 18 17 23 8 20 5 13 1098 10]
[ 5 3 2 27 21 1 1 51 3 1161]]
In [ ]: