

[ MLOps - Task - 4 ]

Face Recognition Using VGG16 and MobileNet.

**Face Recognition is a computer vision task of identifying and verifying a person based on a photograph of their face.**

**Recently, deep learning convolutional neural networks have surpassed classical methods and are achieving state-of-the-art results on standard face recognition datasets. One example of a state-of-the-art model is the VGGFace and VGGFace2 model developed by researchers at the Visual Geometry Group at Oxford.**

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another.

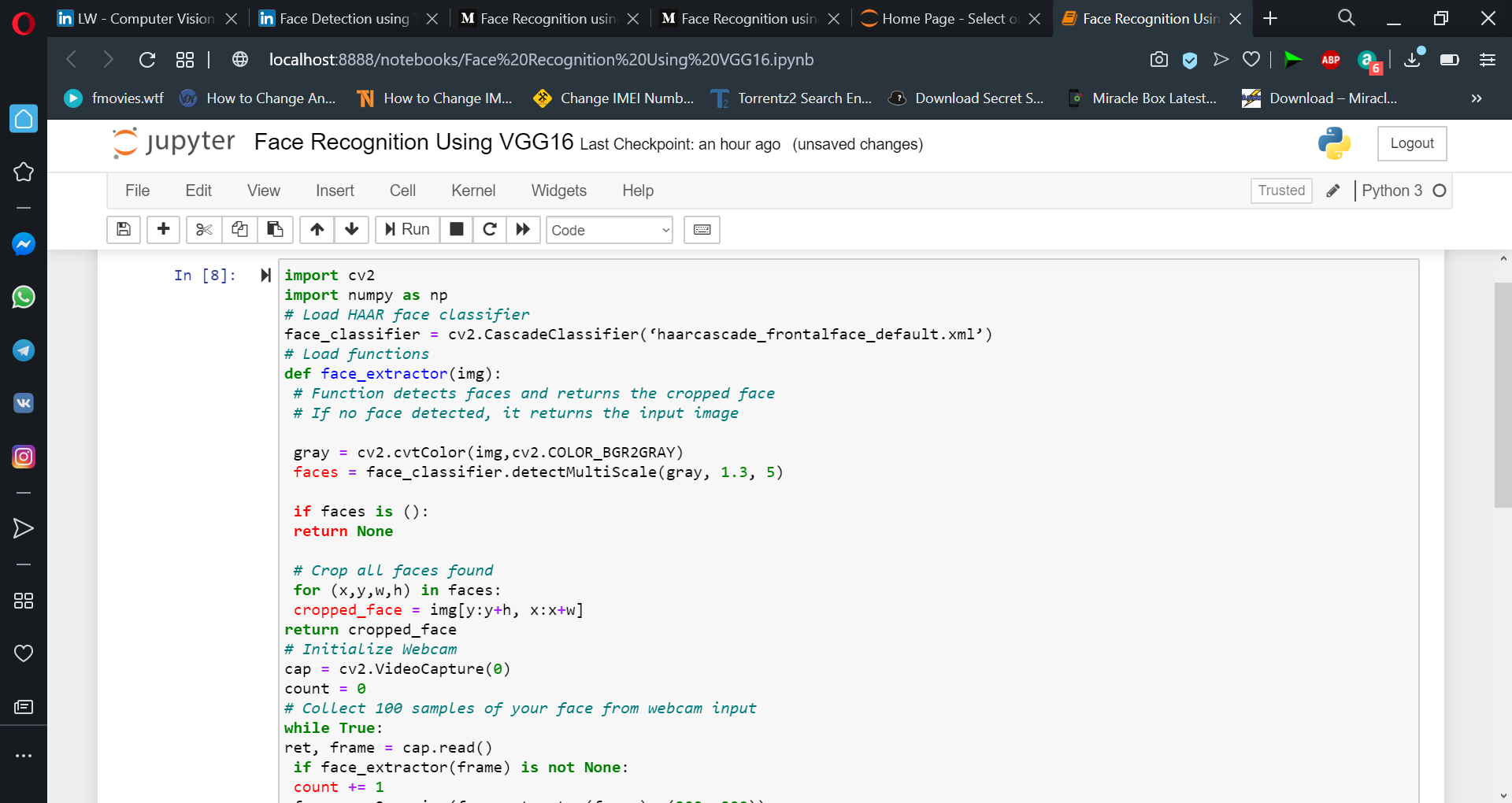
Basic Required libraries:

- conda install tensorflow keras opencv-python pillow numpy.

#Creation of Dataset :

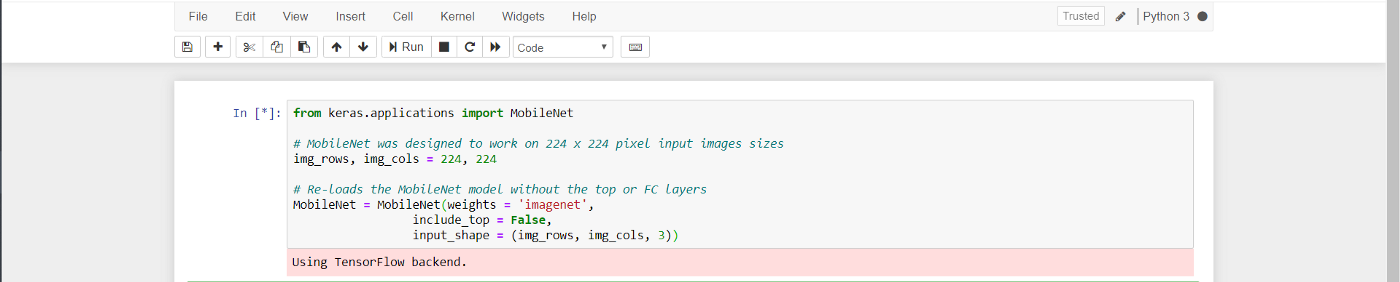
import cv2  
import numpy as np# Load HAAR face classifier  
face\_classifier = cv2.CascadeClassifier(‘haarcascade\_frontalface\_default.xml’)# Load functions  
def face\_extractor(img):  
 # Function detects faces and returns the cropped face  
 # If no face detected, it returns the input image  
   
 gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)  
 faces = face\_classifier.detectMultiScale(gray, 1.3, 5)  
   
 if faces is ():  
 return None  
   
 # Crop all faces found  
 for (x,y,w,h) in faces:  
 cropped\_face = img[y:y+h, x:x+w]return cropped\_face# Initialize Webcam  
cap = cv2.VideoCapture(0)  
count = 0# Collect 100 samples of your face from webcam input  
while True:ret, frame = cap.read()  
 if face\_extractor(frame) is not None:  
 count += 1  
 face = cv2.resize(face\_extractor(frame), (300, 300))# Save file in specified directory with unique name  
 file\_name\_path = r” C:\Users\Public\Pictures” + str(count) + ‘.jpg’  
 cv2.imwrite(file\_name\_path, face)# Put count on images and display live count  
 cv2.putText(face, str(count), (50, 50), cv2.FONT\_HERSHEY\_COMPLEX, 1, (0,255,0), 2)  
 cv2.imshow(‘Face Cropper’, face)  
   
 else:  
 print(“Face not found”)  
 passif cv2.waitKey(1) == 27 or count == 100: #27 is the Esc Key  
 break  
   
cap.release()  
cv2.destroyAllWindows()   
print(“Samples Taken”)

Here ” C:\Users\Public\Pictures” is the path where the samples will be stored.

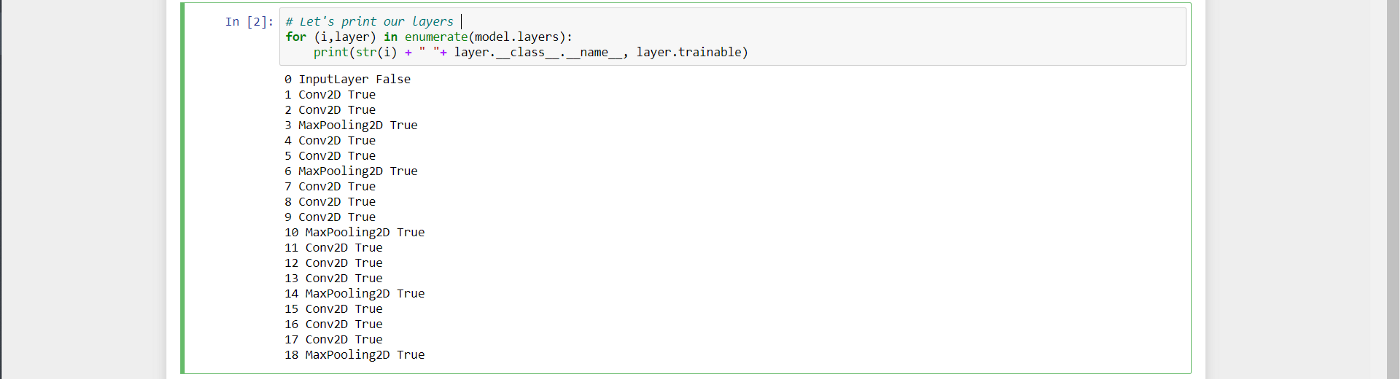


Now, concept of transfer learning will be used on the model MobileNet.

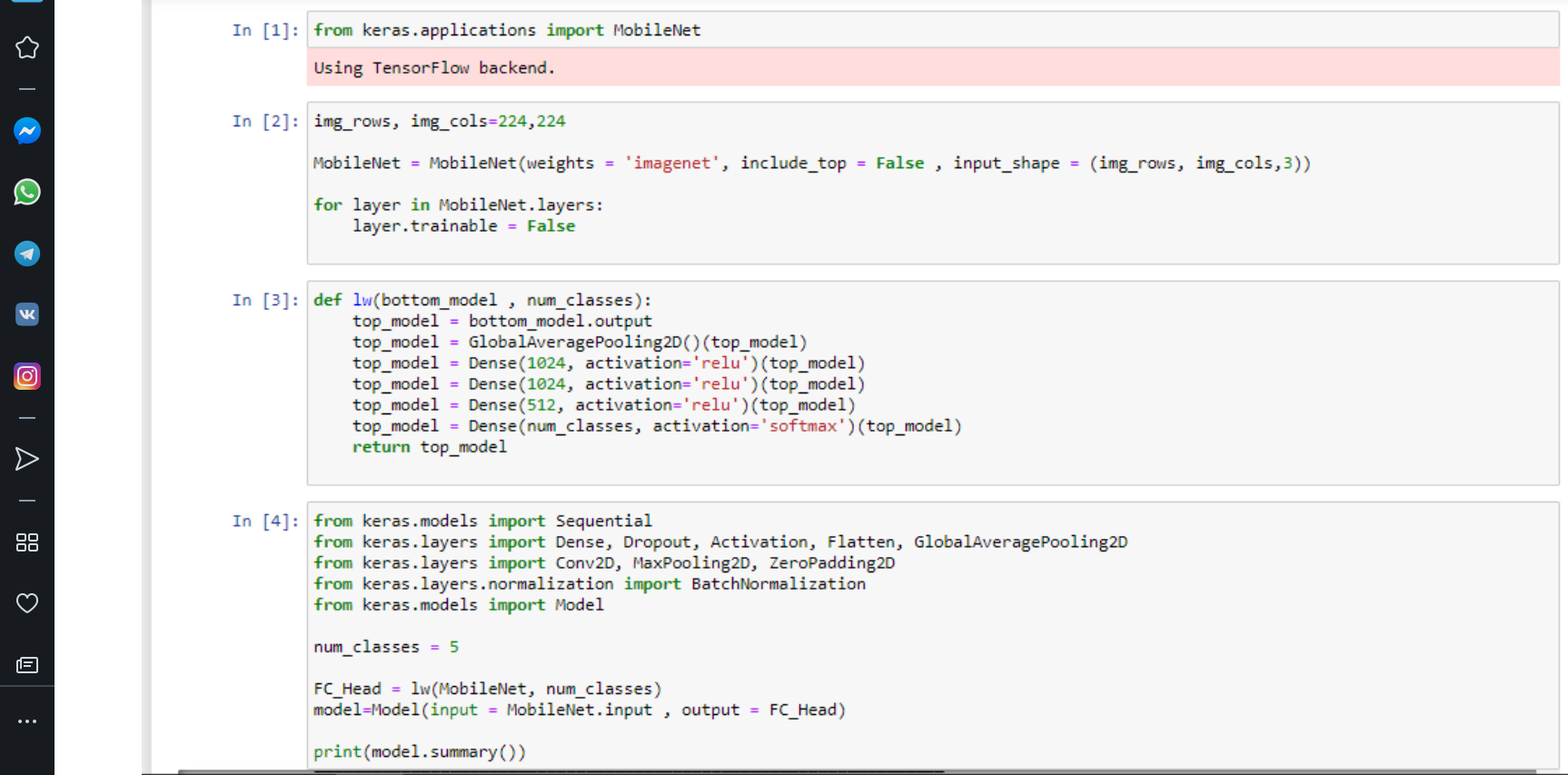
The pre-trained model or download it locally, you can use.

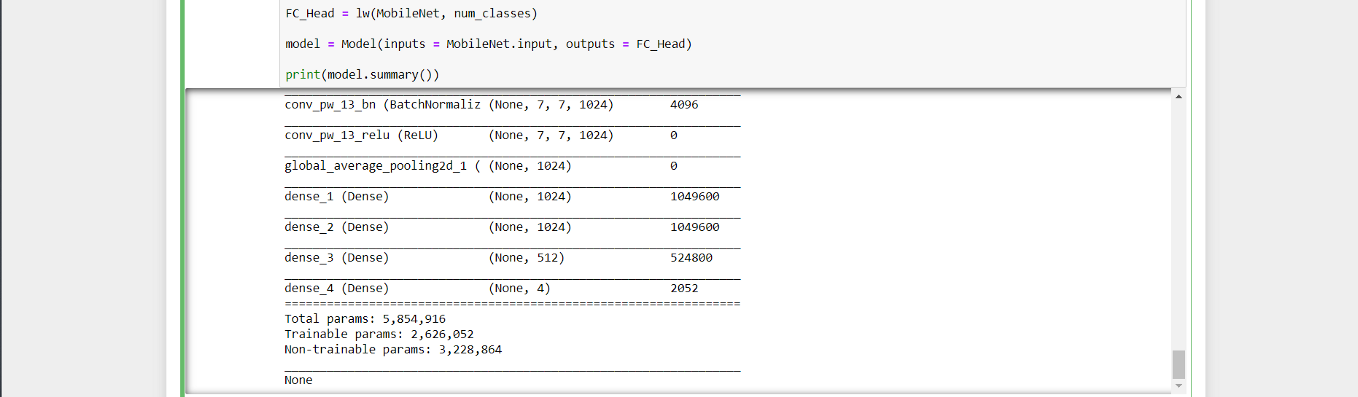


Pre-trained layers that had been used in this model by default. Means that they are ready to be trained again.



Set all the pre-trained layers to False. And add the input layer for training, and after adding all the layer printing summary of the model.





Testing amd modifying the dataset.

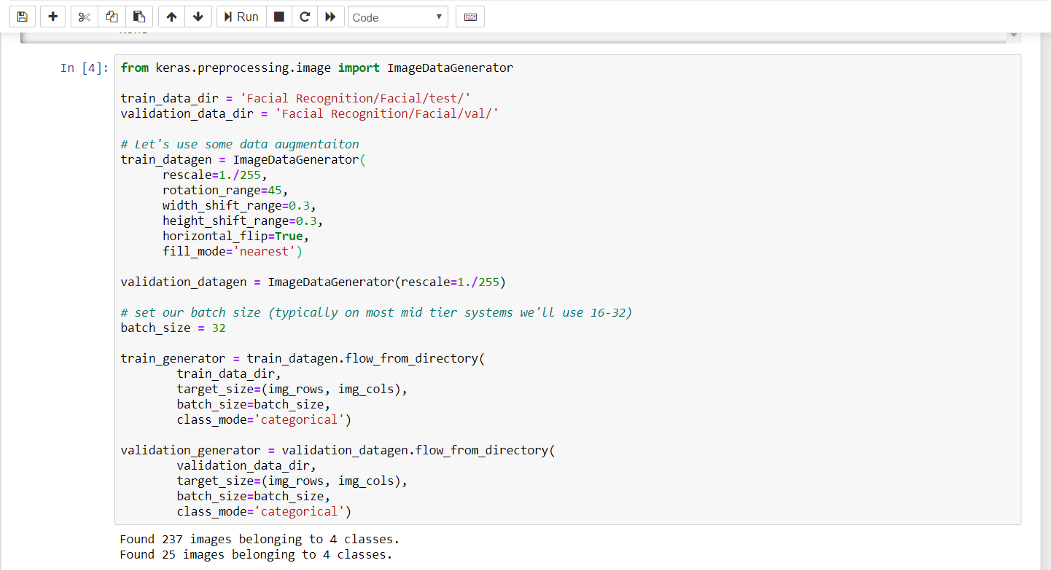
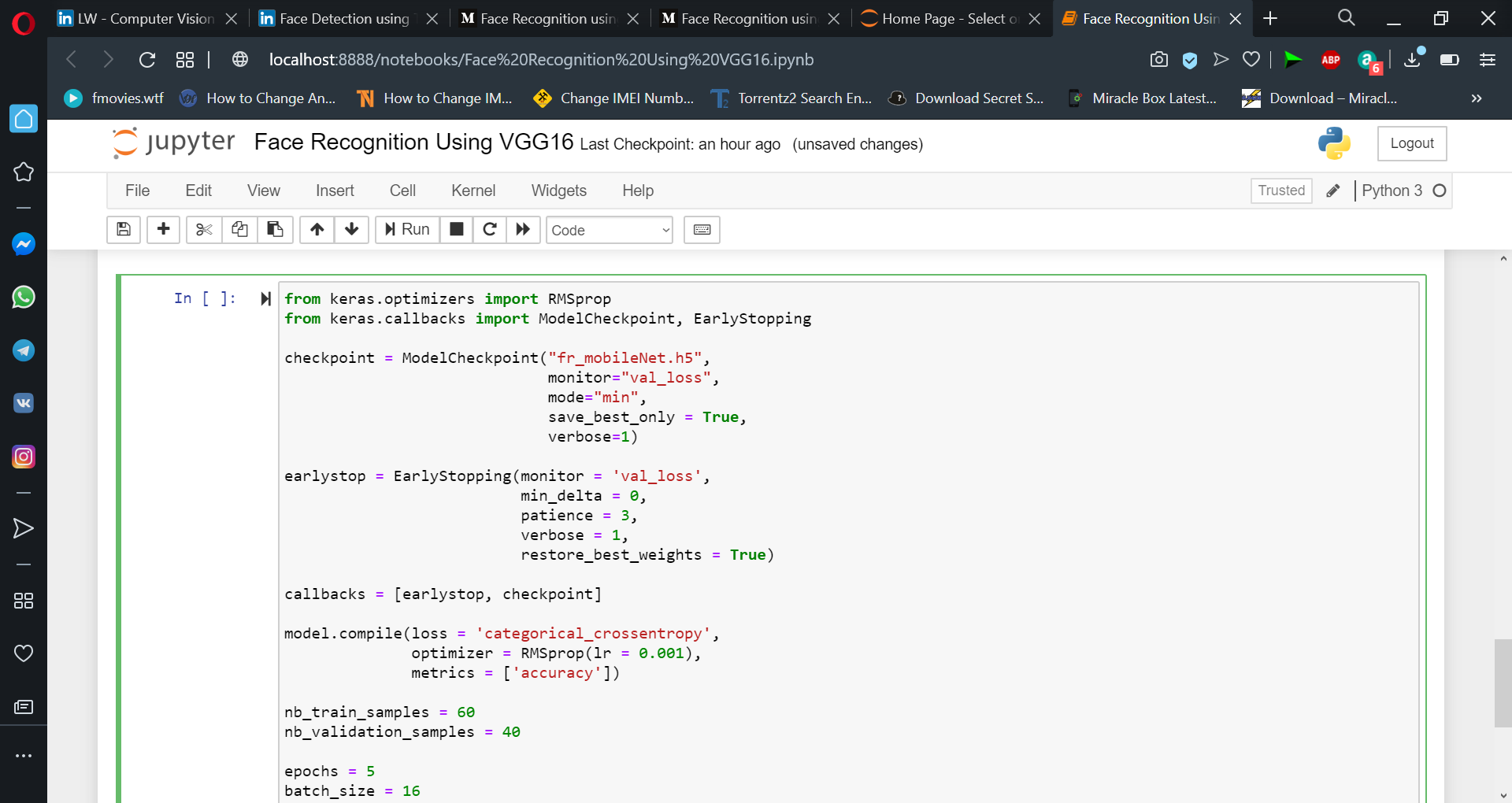
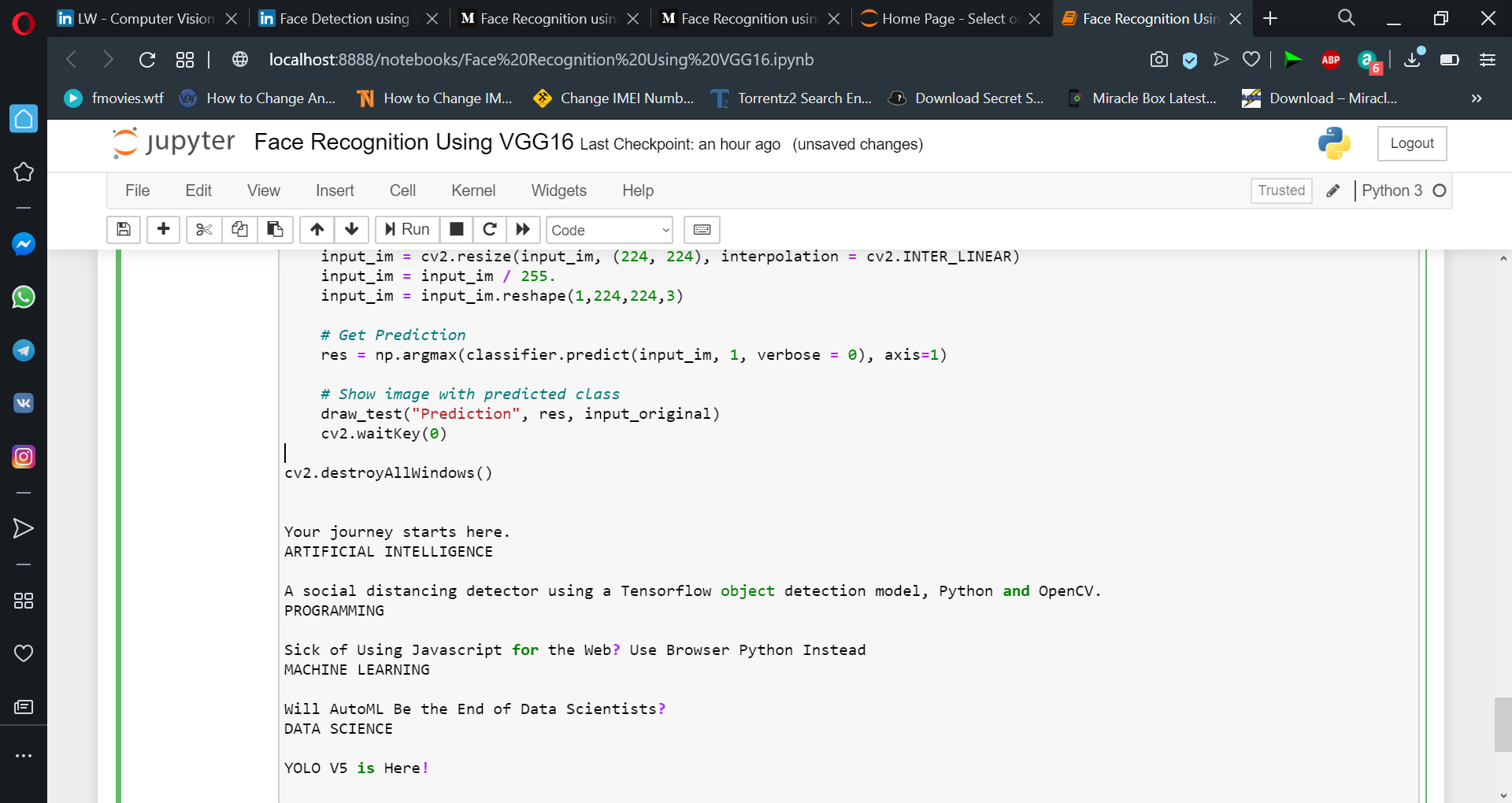


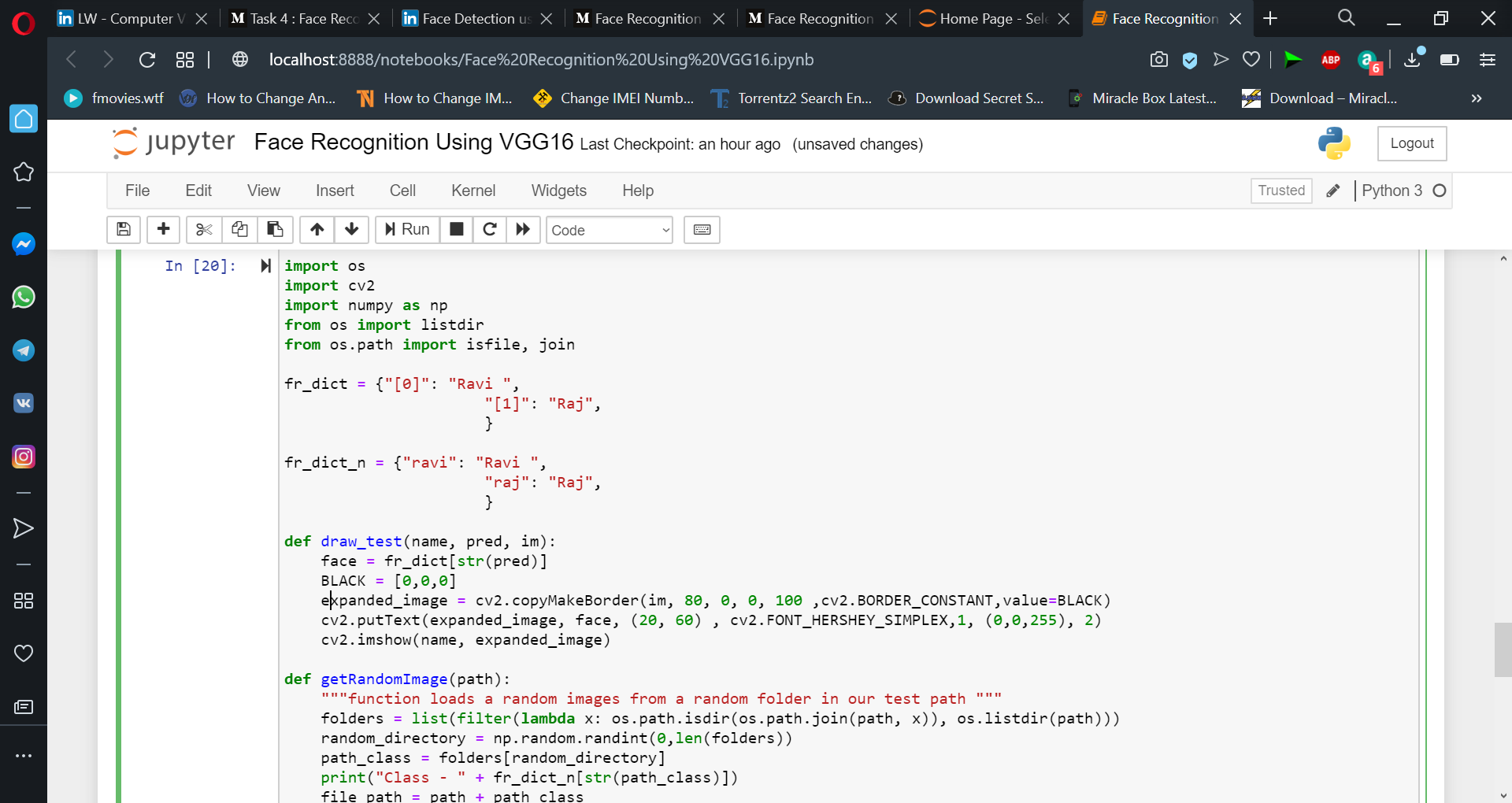
Image Data Generator module. Image Data Generator accepts an input batch of **images**, randomly transforms the batch, and then returns both the original batch and modified data.

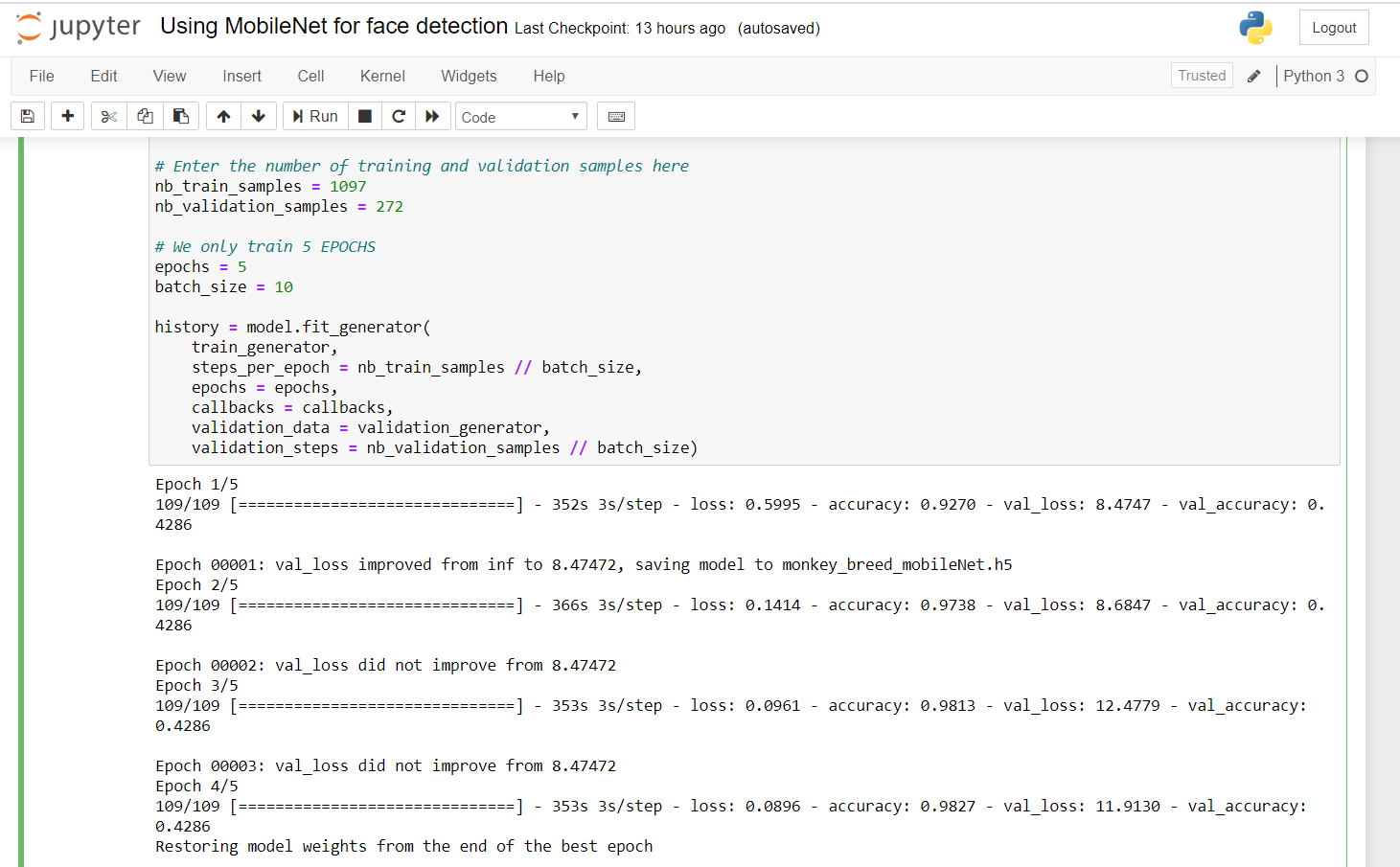
Now, we’ll train the model and save it.



 Loading the model and prediction of the model for facial recognition.







The accuracy of the model which is almost 91% by having such few epochs and using less computation power. The codes are complete checked and you can use it easily.