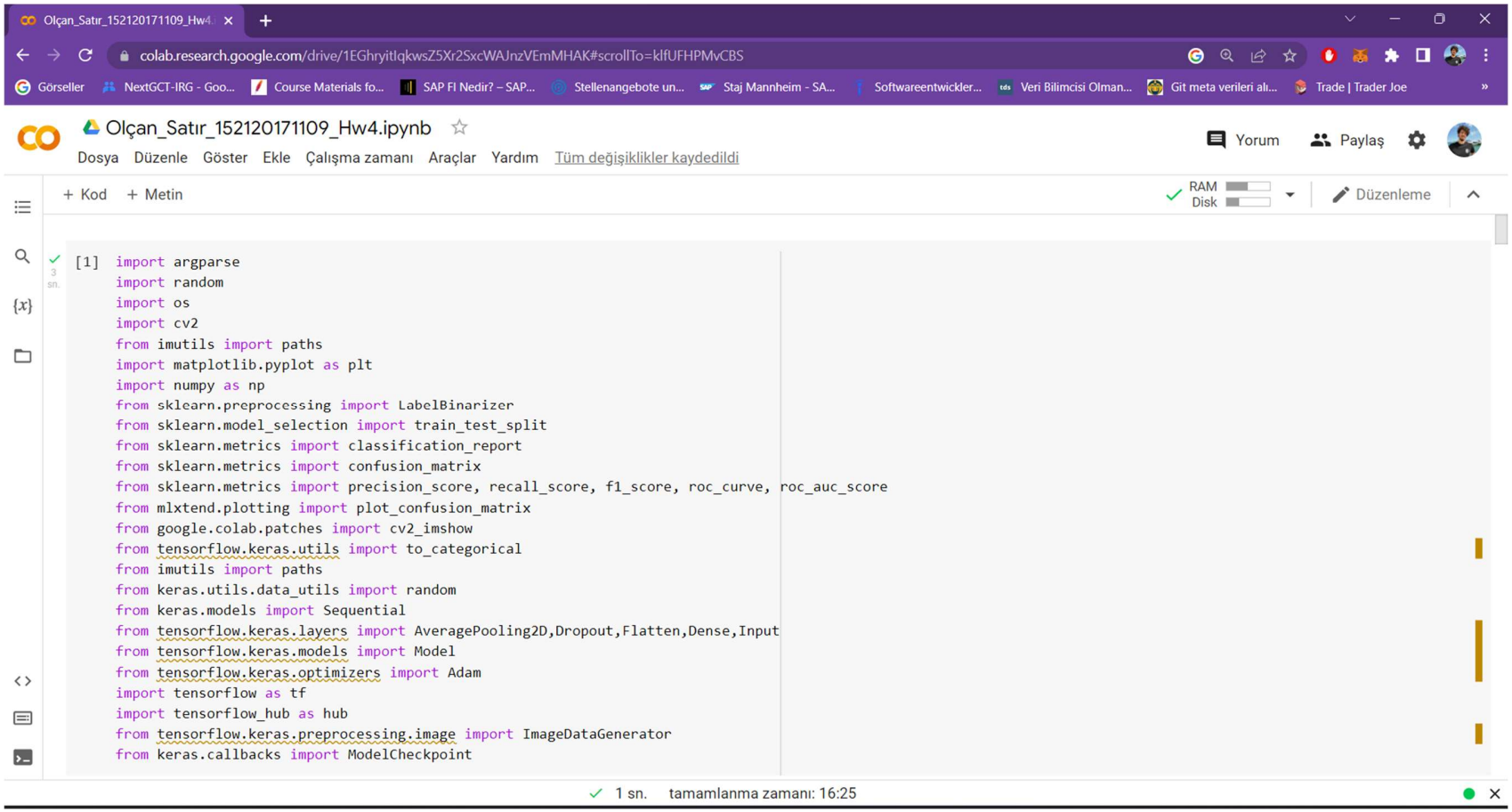
**OLÇAN SATIR**

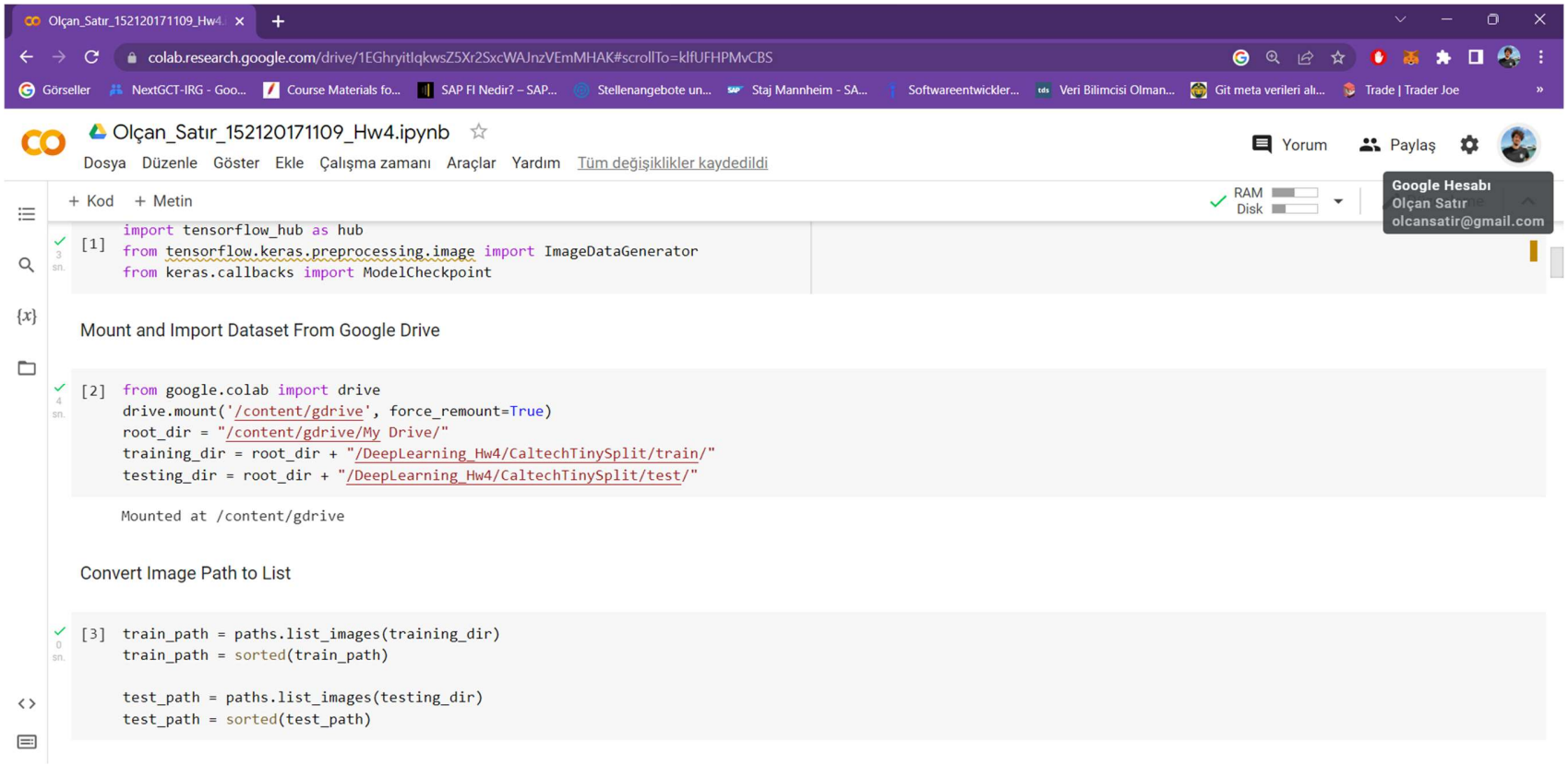
**152120171109**

# Deep Learning HW-4

We define our libraries.

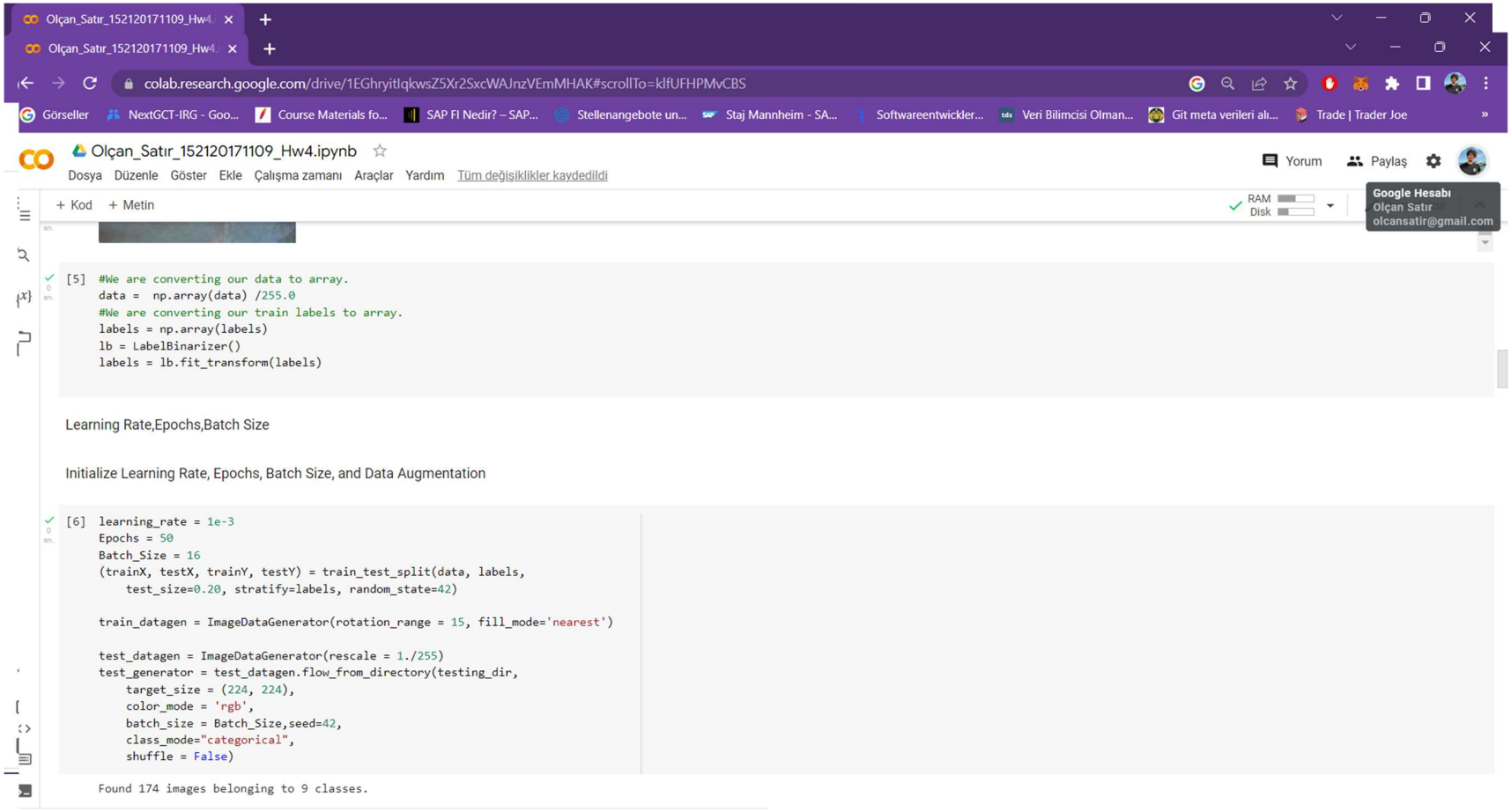


Here, we first do it from the folder we created for the dataset in the drive. Then we create an image list and sort it.

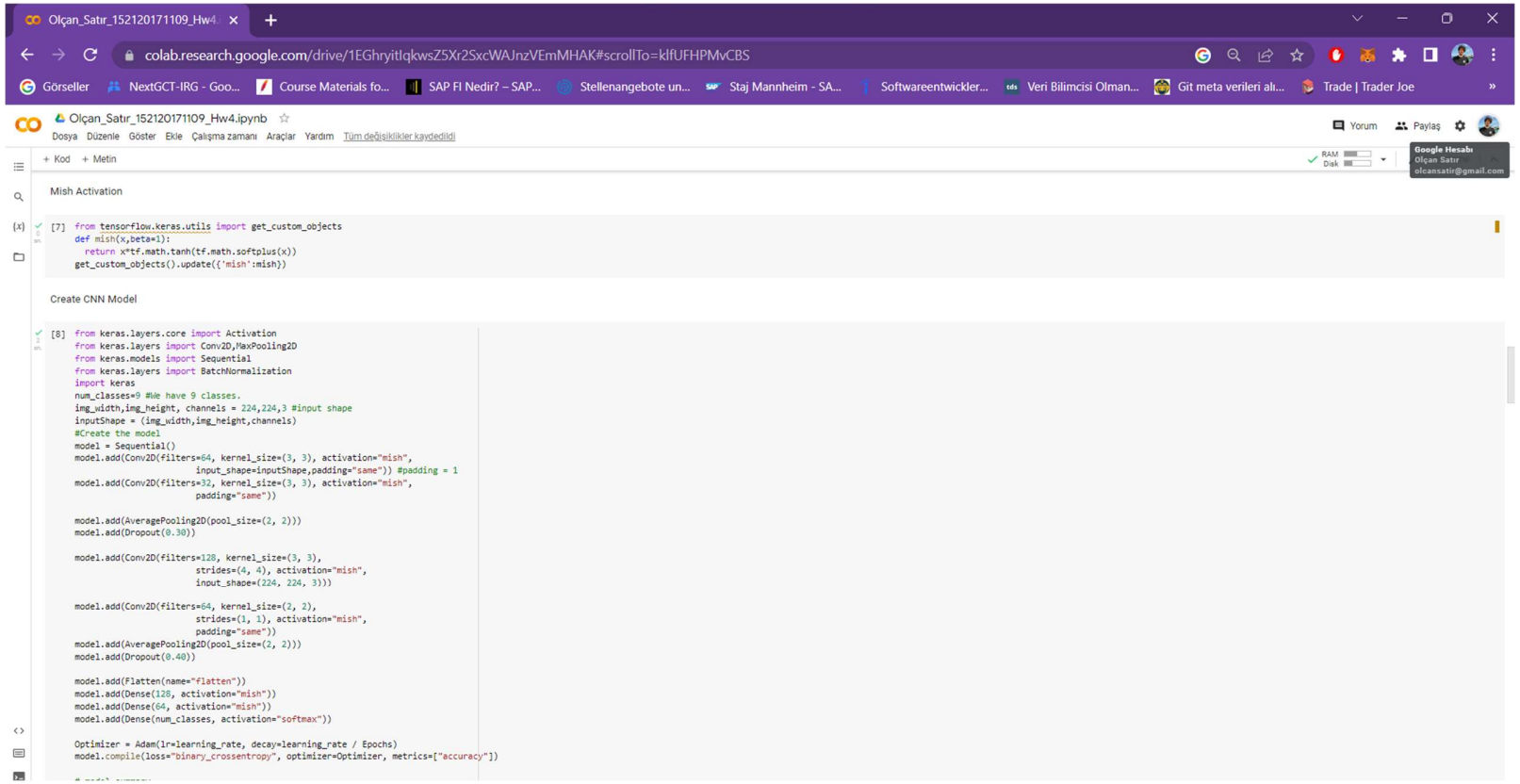


Then we resize the train pictures as 224 by 224 by converting them from BGR to RGB by naming them as inputs and targets. In the for loop below, we print 6 random pictures to see. Here we convert the data and labels into array. Then we define the learning rate epoch number and batch size.

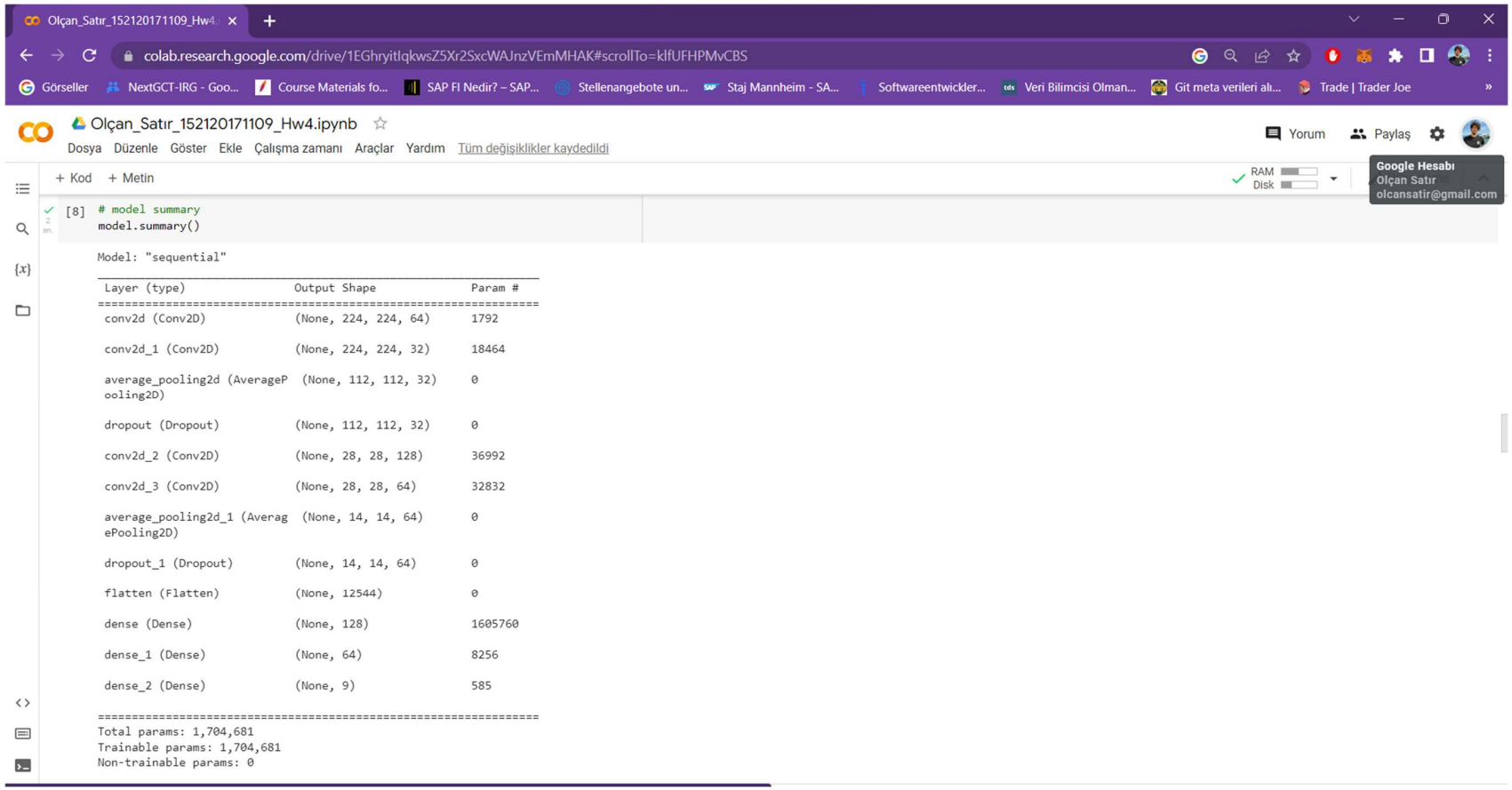
We define train_datagen by editing test datagen and test generator.



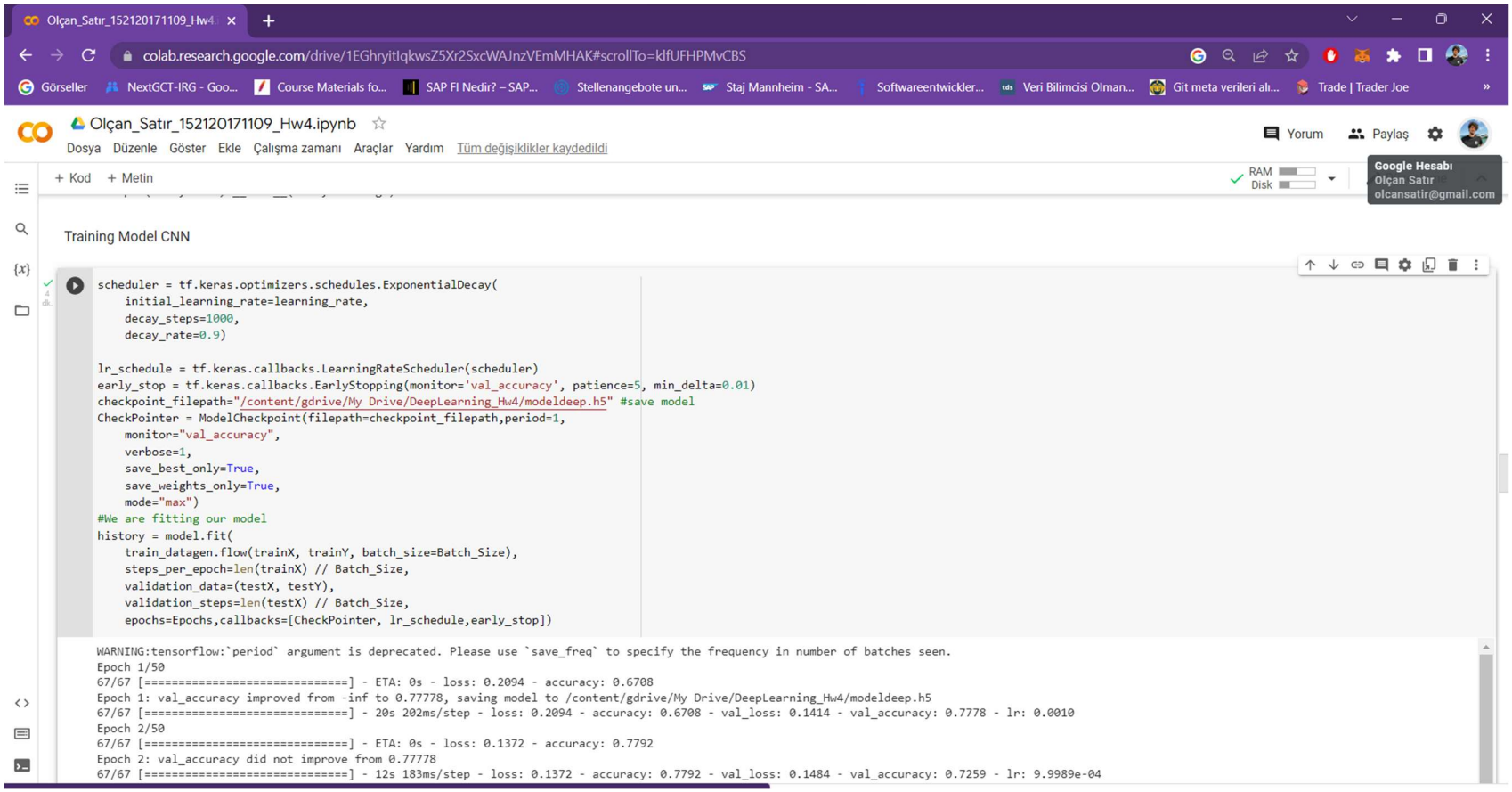
Then we create it as a function to use the mish activation function. Then we create our model. We make conv 2 times while creating the model. In the first conv, we use a 3 by 3 filter and our filter number is 64 padding once and we choose mish as the activation function. Then, in the 2nd conv, we filter 32 times with a 3 x 3 filter, our activation function is the same, we mish and we do padding once again. Then we halve our output size with Average Pooling. We use a dropout to prevent overfitting. Then we do the 3rd conv operation and apply a 3 x 3 filter 128 times. Our stride is 4 and our activation function is mish. Then, for the 4th conv, our filter size is 2x2 and we filter 64 times. Then we reduce our output size by doing AveragePooling again. Then we add a dropout again and convert our output in matrix format to vector with 2 fully connected operations. Afterwards, we create our model by giving our num_classes as 9 since there are 9 classes in total. We use Adam as the optimizer and binary crossentropy as the loss function.



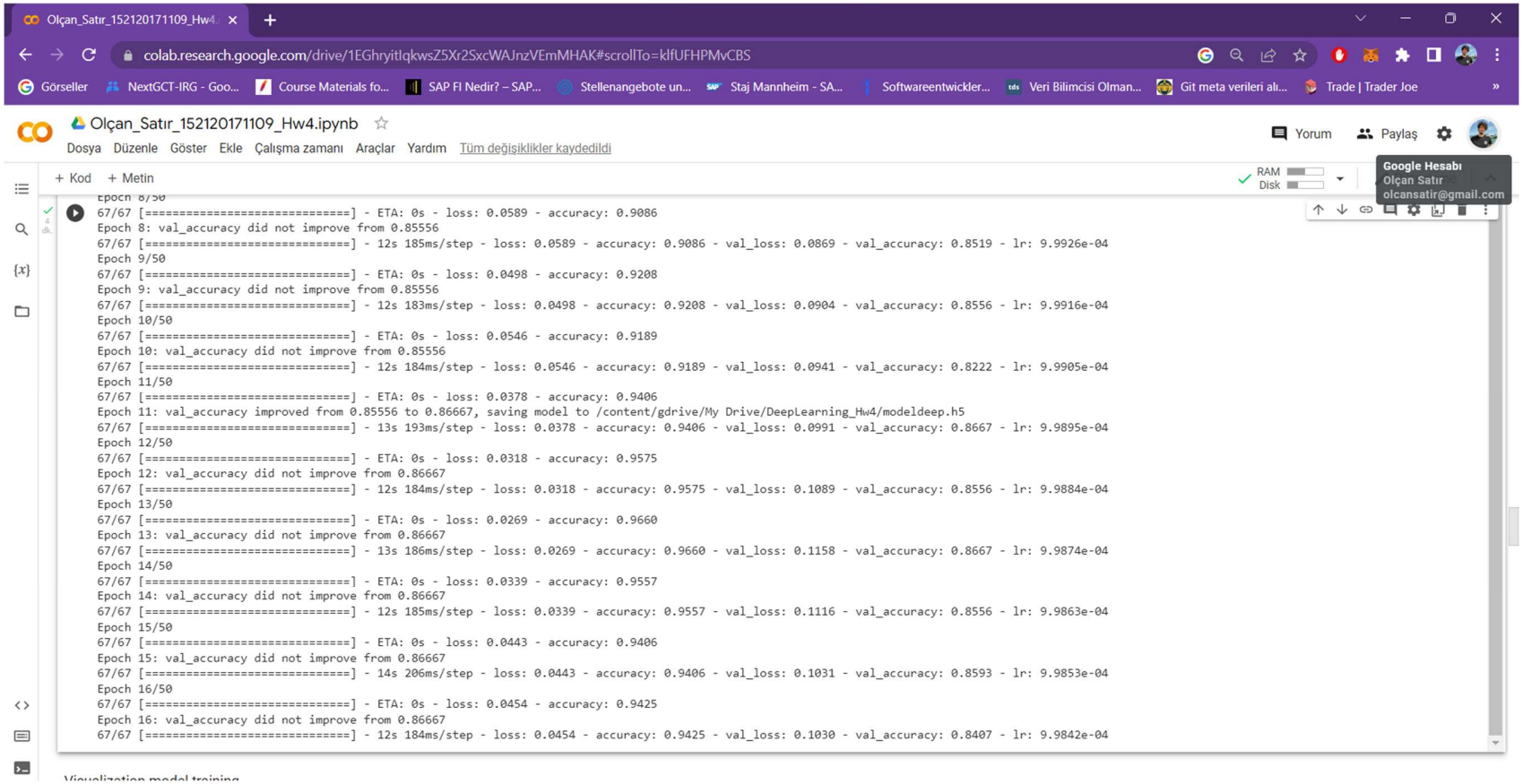
Here we summarize our model.

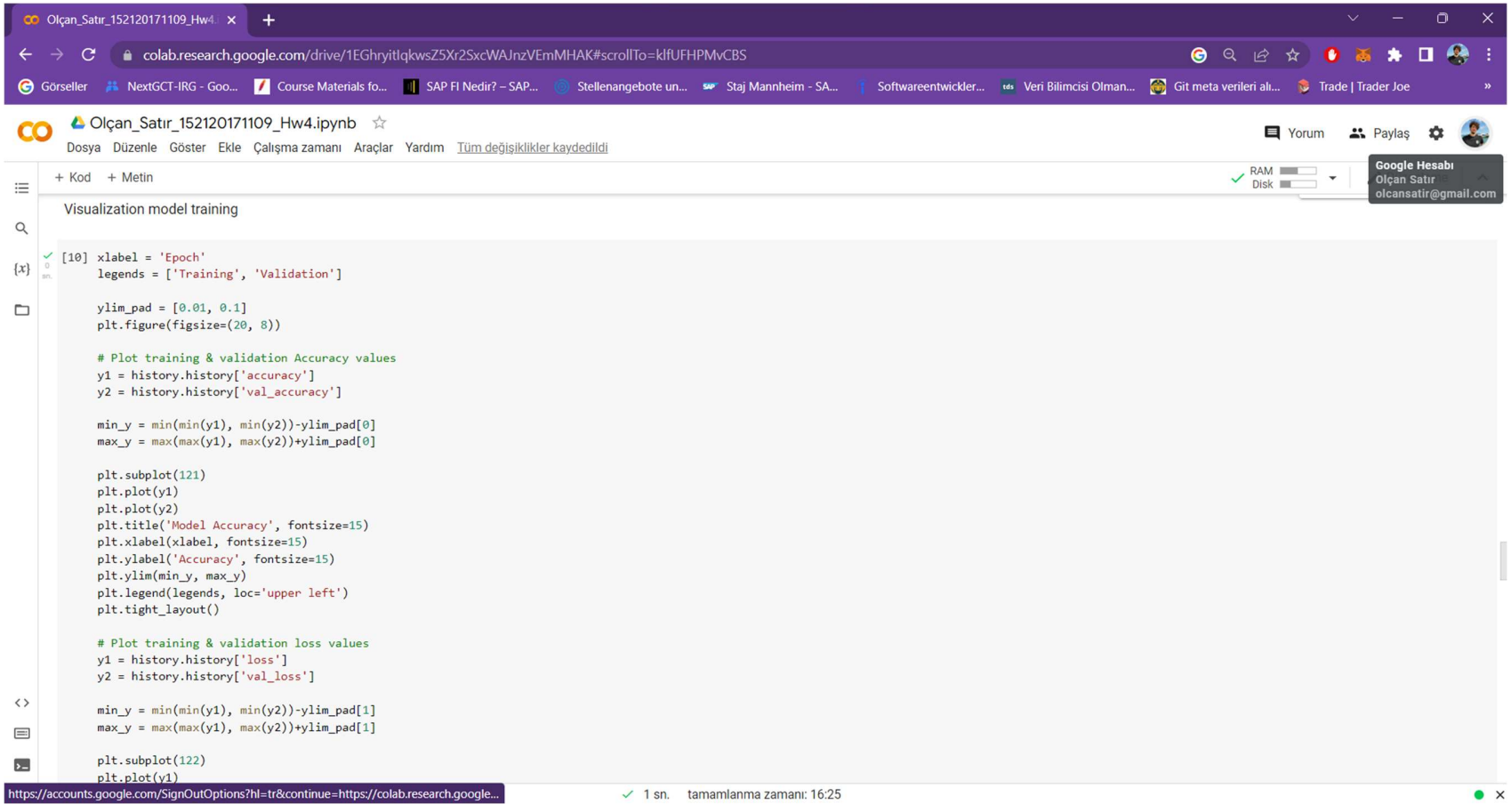


Here we use a CheckPointer to save our model and define an early stop, if the accuracy value of our model gives close values 5 times in a row, it saves the model. So here, our model stopped at the 16th epoch before 50 epochs were defined. Then we save our model by fitting.

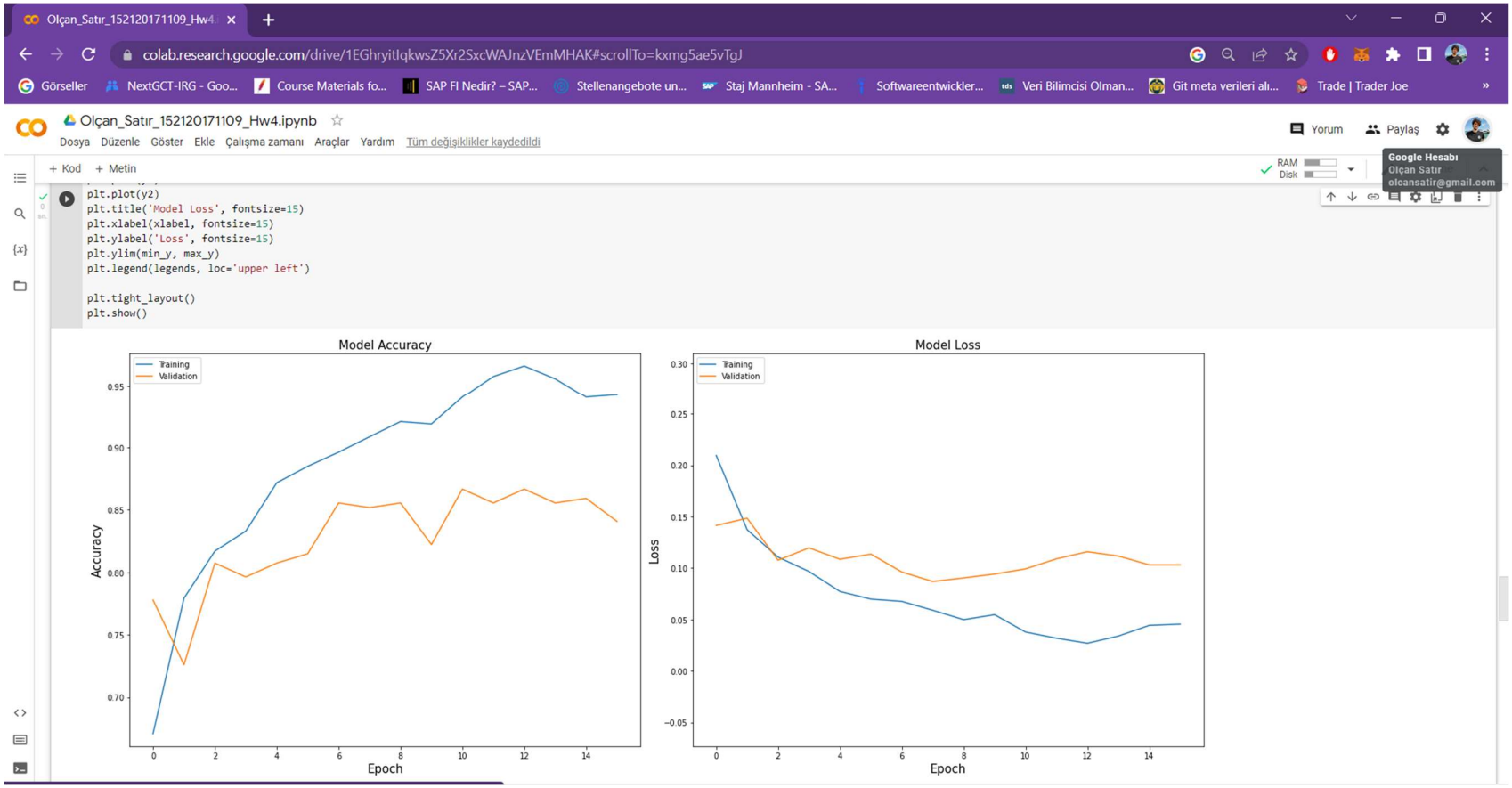


Accuracy and loss values of our model in each epoch.

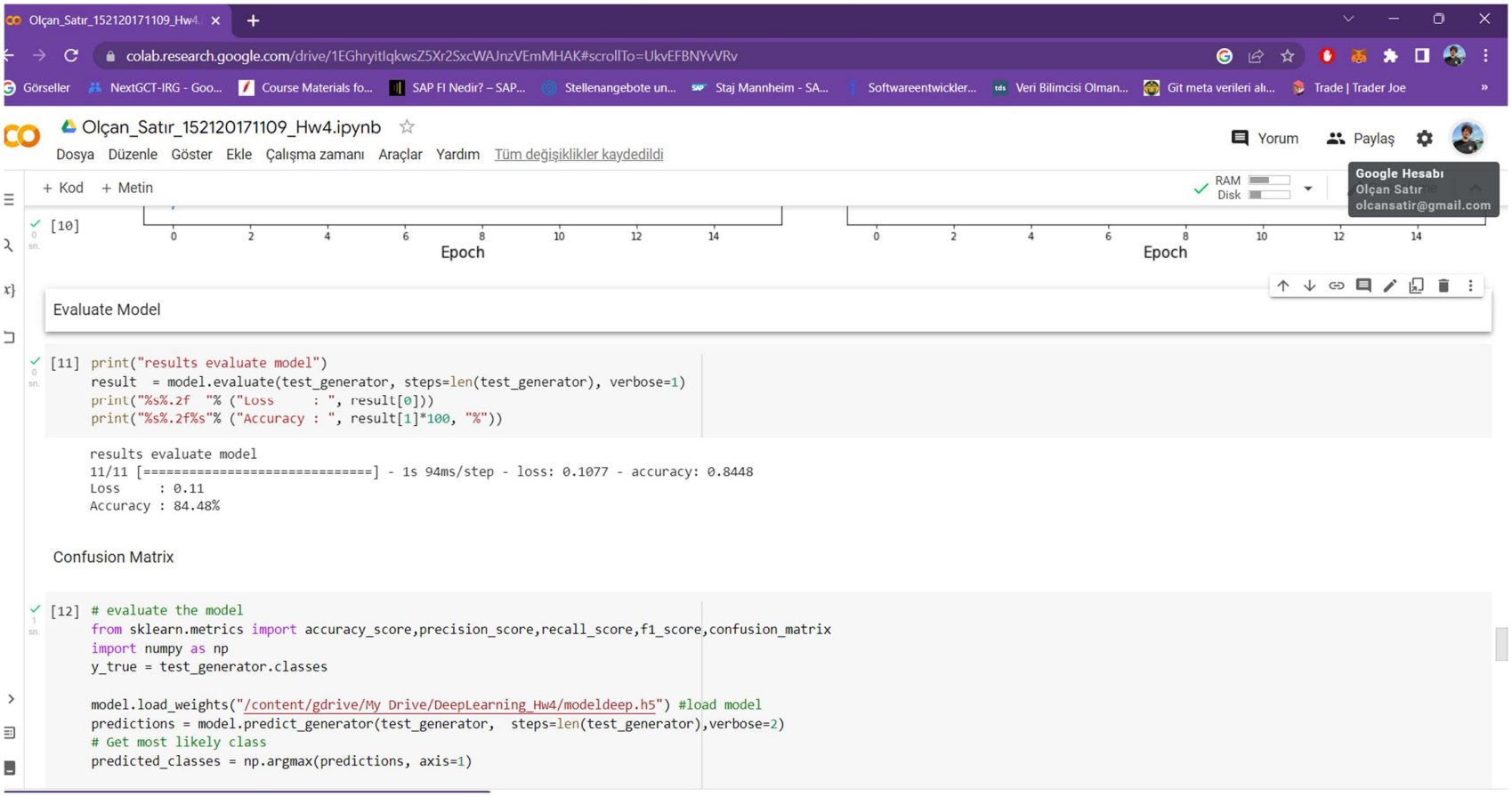



Olçan_Satır_152120171109_Hw4.ipynb
colab.research.google.com/drive/1EGhryitIqkwsZ5Xr2SxcWAJnzVEmMHAK#scrollTo=klfUFHPMvCBS
Görseller
NextGCT-IRG - Goo...
Course Materials fo...
SAP FI Nedir? – SAP...
Stellenangebote un...
Staj Mannheim - SA...
Softwareentwickler...
Veri Bilimcisi Olman...
Git meta verileri alı...
Trade | Trader Joe
Olçan_Satır_152120171109_Hw4.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
+ Kod + Metin
RAM
Disk
Google Hesabı
Olçan Satır
olcansatir@gmail.com
Visualization model training
[10] xlabel = 'Epoch'
legends = ['Training', 'Validation']
ylim_pad = [0.01, 0.1]
plt.figure(figsize=(20, 8))
# Plot training & validation Accuracy values
y1 = history.history['accuracy']
y2 = history.history['val_accuracy']
min_y = min(min(y1), min(y2))-ylim_pad[0]
max_y = max(max(y1), max(y2))+ylim_pad[0]
plt.subplot(121)
plt.plot(y1)
plt.plot(y2)
plt.title('Model Accuracy', fontsize=15)
plt.xlabel(xlabel, fontsize=15)
plt.ylabel('Accuracy', fontsize=15)
plt.ylim(min_y, max_y)
plt.legend(legends, loc='upper left')
plt.tight_layout()
# Plot training & validation loss values
y1 = history.history['loss']
y2 = history.history['val_loss']
min_y = min(min(y1), min(y2))-ylim_pad[1]
max_y = max(max(y1), max(y2))+ylim_pad[1]
plt.subplot(122)
plt.plot(y1)
https://accounts.google.com/SignOutOptions?hl=tr&continue=https://colab.research.google...
1 sn. tamamlanma zamanı: 16:25

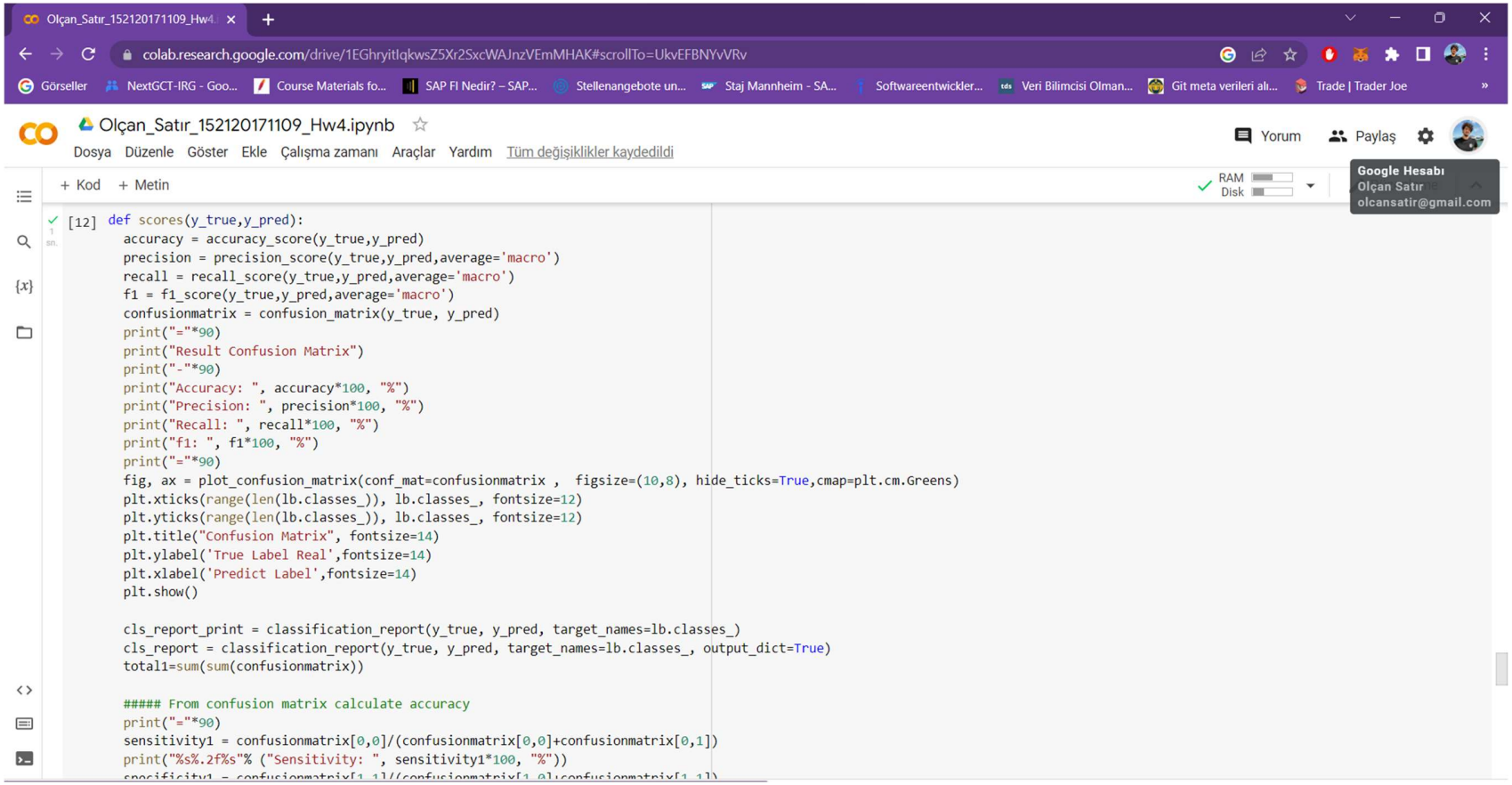
Here we plot the accuracy and loss values of our model for training and validation.



Then we evaluate our model and print the truth value.



Here we create and print the confusion matrix of our model, and then print the F1 score values at the bottom. Our model correctly predicted 44 of the faces in our test data and correctly predicted 77 of the motorbikes.



Olçan_Satır_152120171109_Hw4.ipynb

colab.research.google.com/drive/1EGhryitIqkwsZ5Xr2SxcWAJnzVEmMHAK#scrollTo=IkuEF6livZ6L

Olçan_Satır_152120171109_Hw4.ipynb

Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi

Yorum Paylaş

+ Kod + Metin

RAM Disk

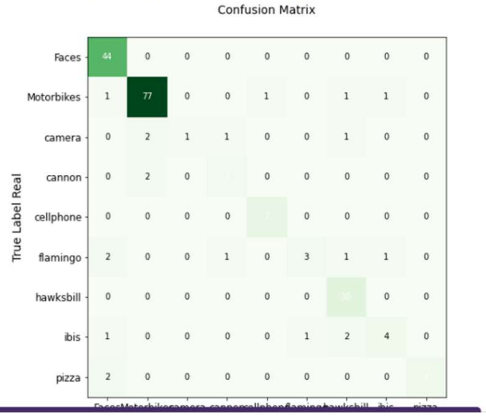
Google Hesabı
Olçan Satır
olcansatir@gmail.com

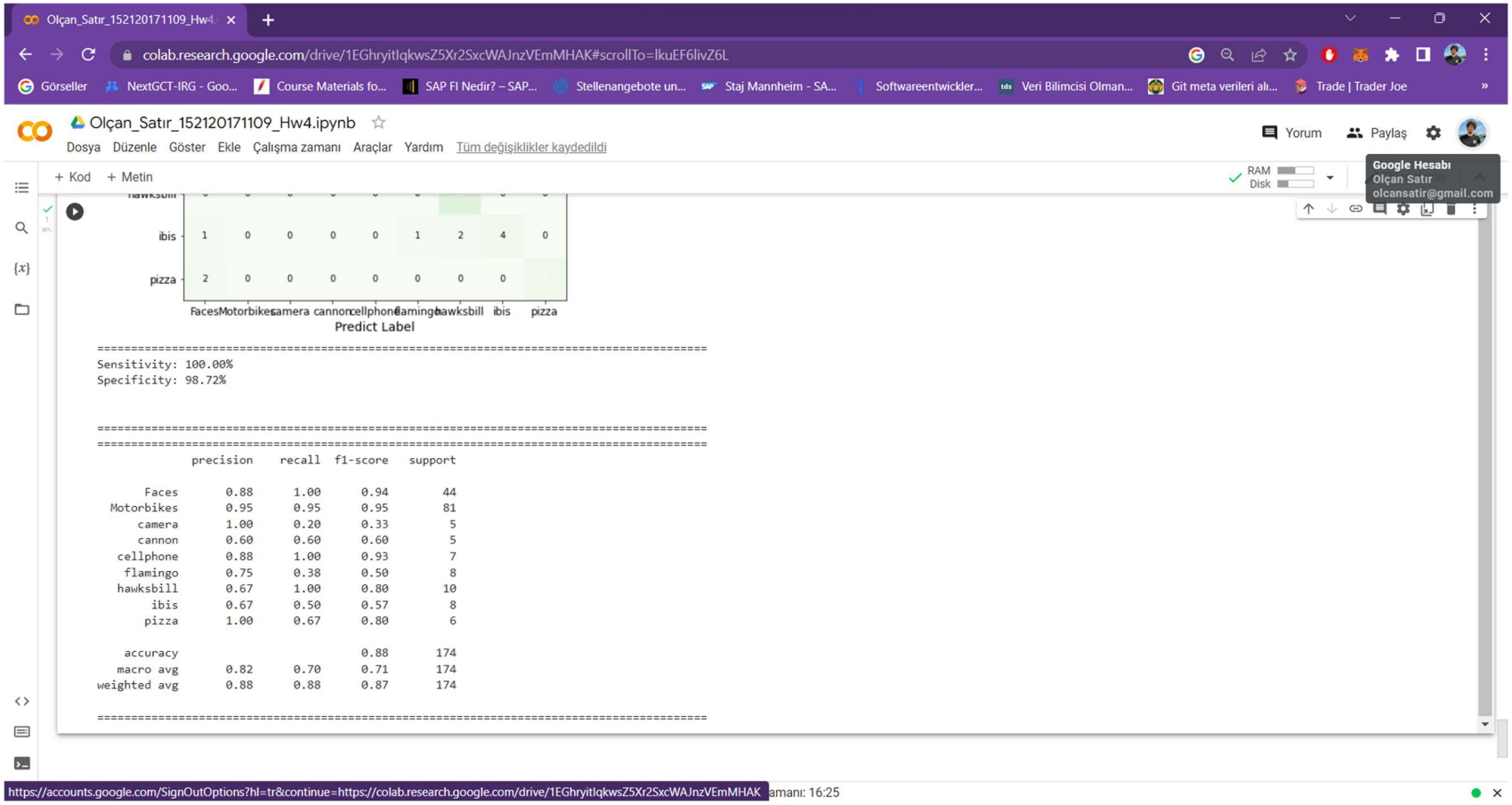
```
##### From confusion matrix calculate accuracy
print("="*90)
sensitivity1 = confusionmatrix[0,0]/(confusionmatrix[0,0]+confusionmatrix[0,1])
print("%s%.2f%s"% ("Sensitivity: ", sensitivity1*100, "%"))
specificity1 = confusionmatrix[1,1]/(confusionmatrix[1,0]+confusionmatrix[1,1])
print("%s%.2f%s"% ("Specificity: ", specificity1*100, "%"))

print("\n")
print("="*90)
print("="*90)
print(cls_report_print)
print("="*90)

scores(y_true,predicted_classes)
```

```
Recall:  69.91426611796983 %
f1:  71.38758594235347 %
==========================================================================================
```




Olçan_Satır_152120171109_Hw4
colab.research.google.com/drive/1EGhryitIqkwsZ5Xr2SxcWAJnzVEmMHAK#scrollTo=IkuEF6IivZ6L
Olçan_Satır_152120171109_Hw4.ipynb
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi
Yorum
Paylaş
+ Kod + Metin
RAM
Disk
Google Hesabı
Olçan Satır
olcansatir@gmail.com
ibis 1 0 0 0 0 1 2 4 0
pizza 2 0 0 0 0 0 0 0
Faces Motorbikes camera cannon cellphone flamingo hawksbill ibis pizza
Predict Label
Sensitivity: 100.00%
Specificity: 98.72%
precision recall f1-score support
Faces 0.88 1.00 0.94 44
Motorbikes 0.95 0.95 0.95 81
camera 1.00 0.20 0.33 5
cannon 0.60 0.60 0.60 5
cellphone 0.88 1.00 0.93 7
flamingo 0.75 0.38 0.50 8
hawksbill 0.67 1.00 0.80 10
ibis 0.67 0.50 0.57 8
pizza 1.00 0.67 0.80 6
accuracy 0.88 174
macro avg 0.82 0.70 0.71 174
weighted avg 0.88 0.88 0.87 174
https://accounts.google.com/SignOutOptions?hl=tr&continue=https://colab.research.google.com/drive/1EGhryitIqkwsZ5Xr2SxcWAJnzVEmMHAK
amanı: 16:25