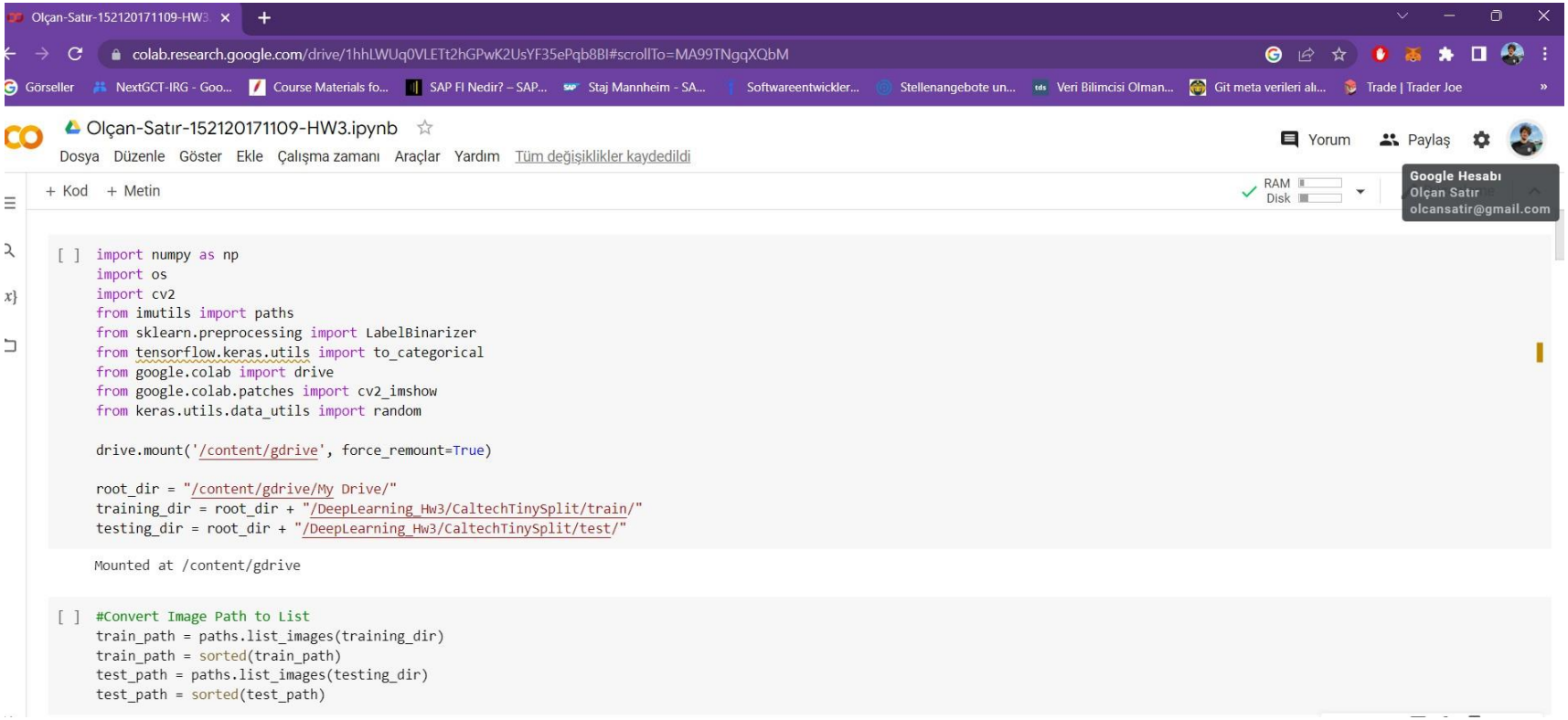
OLÇAN SATIR

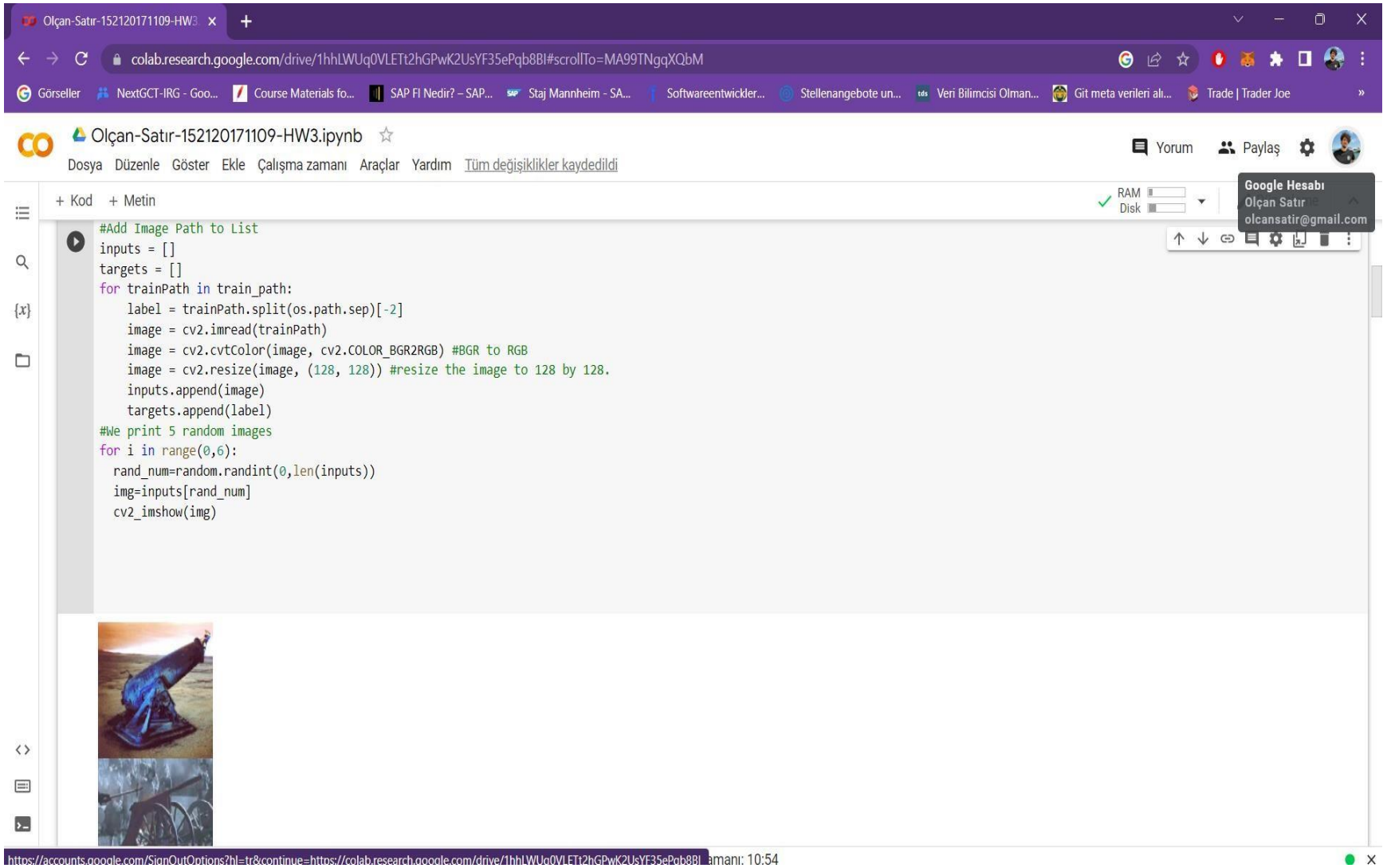
152120171109

Deep Learning HW-3

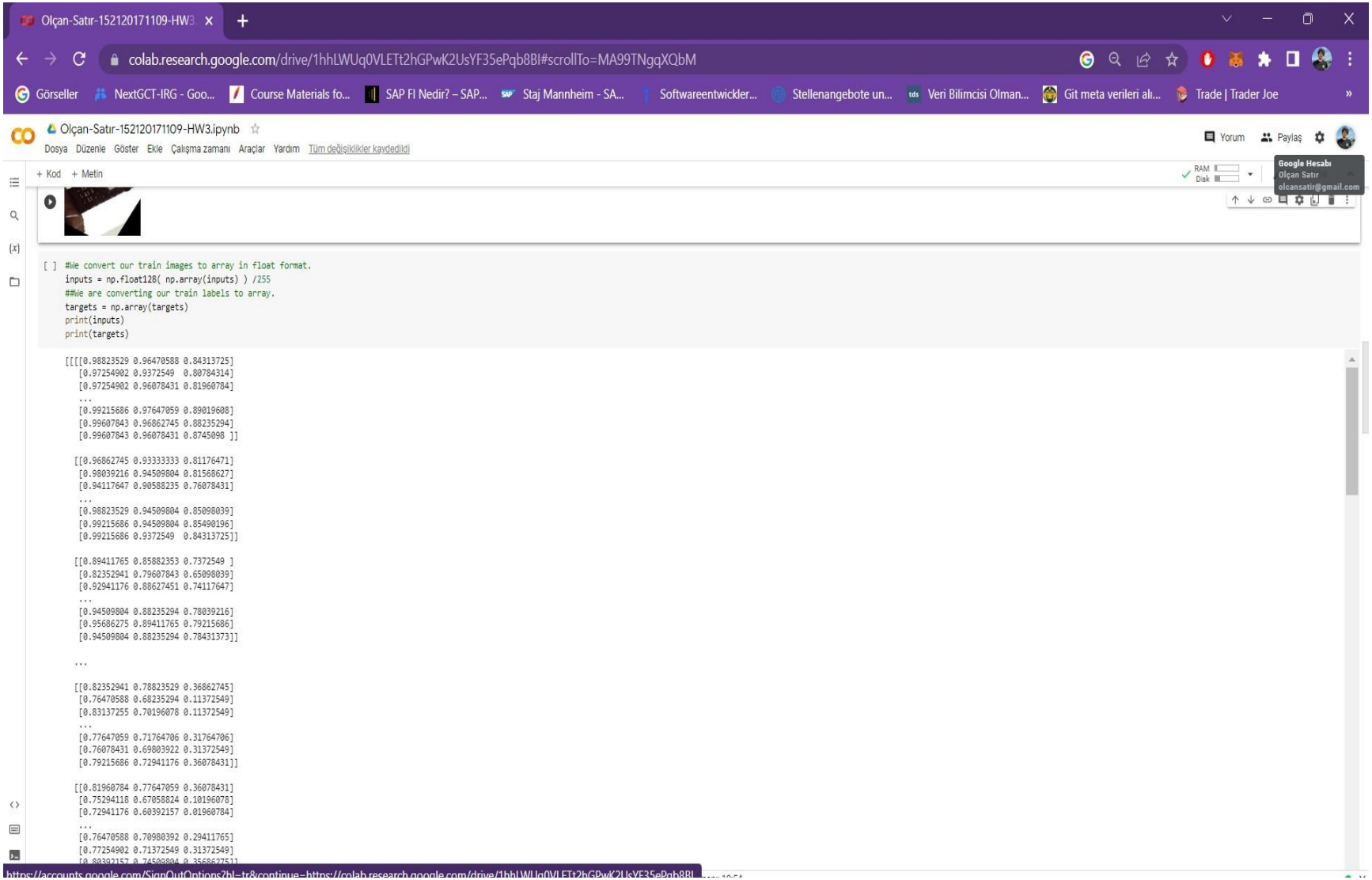
Here, we first do it from the folder we created for the dataset in the drive. Then we create an image list and sort it.



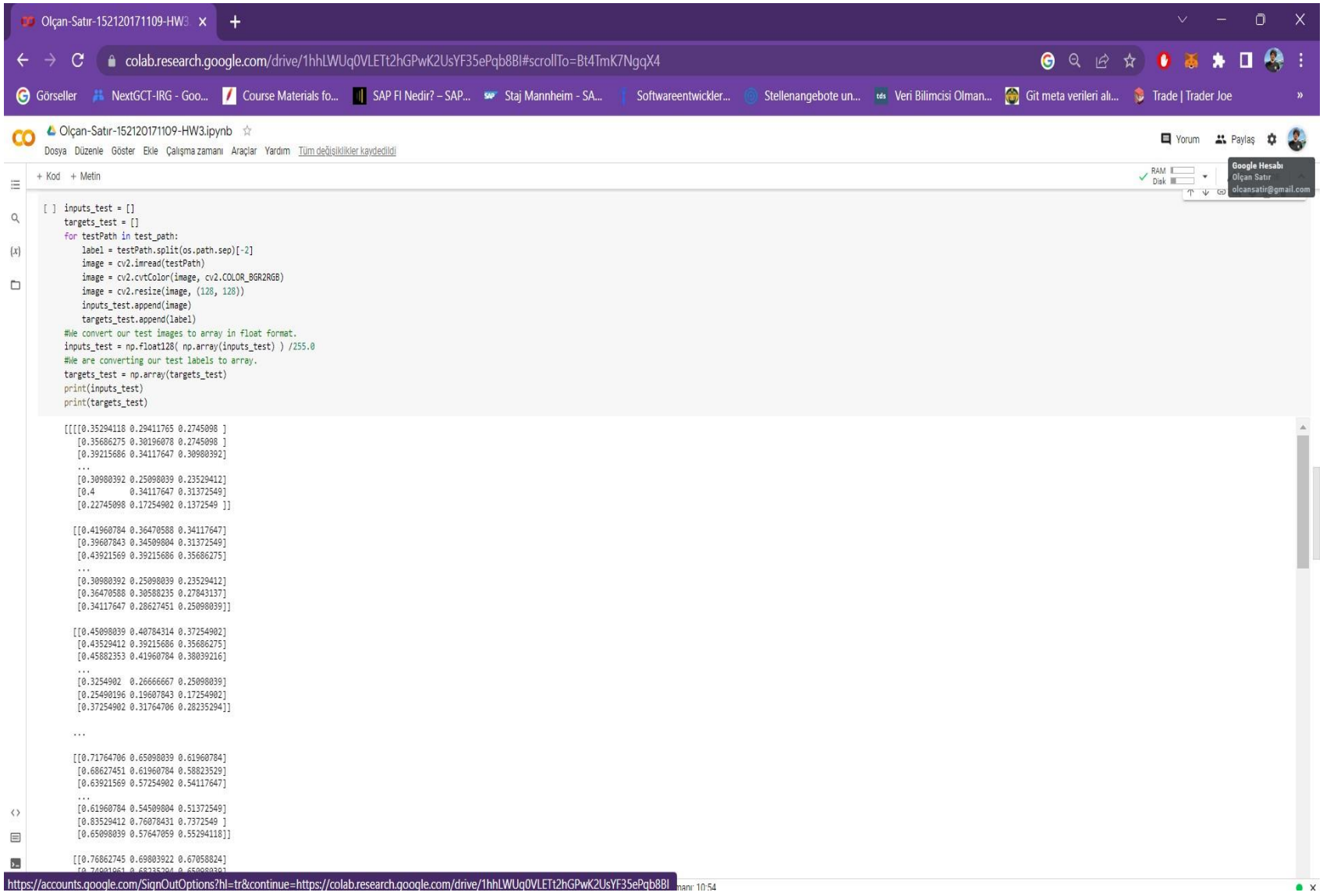
Then we resize the train pictures as 128 by 128 by converting them from BGR to RGB by naming them as inputs and targets. In the for loop below, we print 5 random pictures to see.



Here we convert the inputs and targets into array.

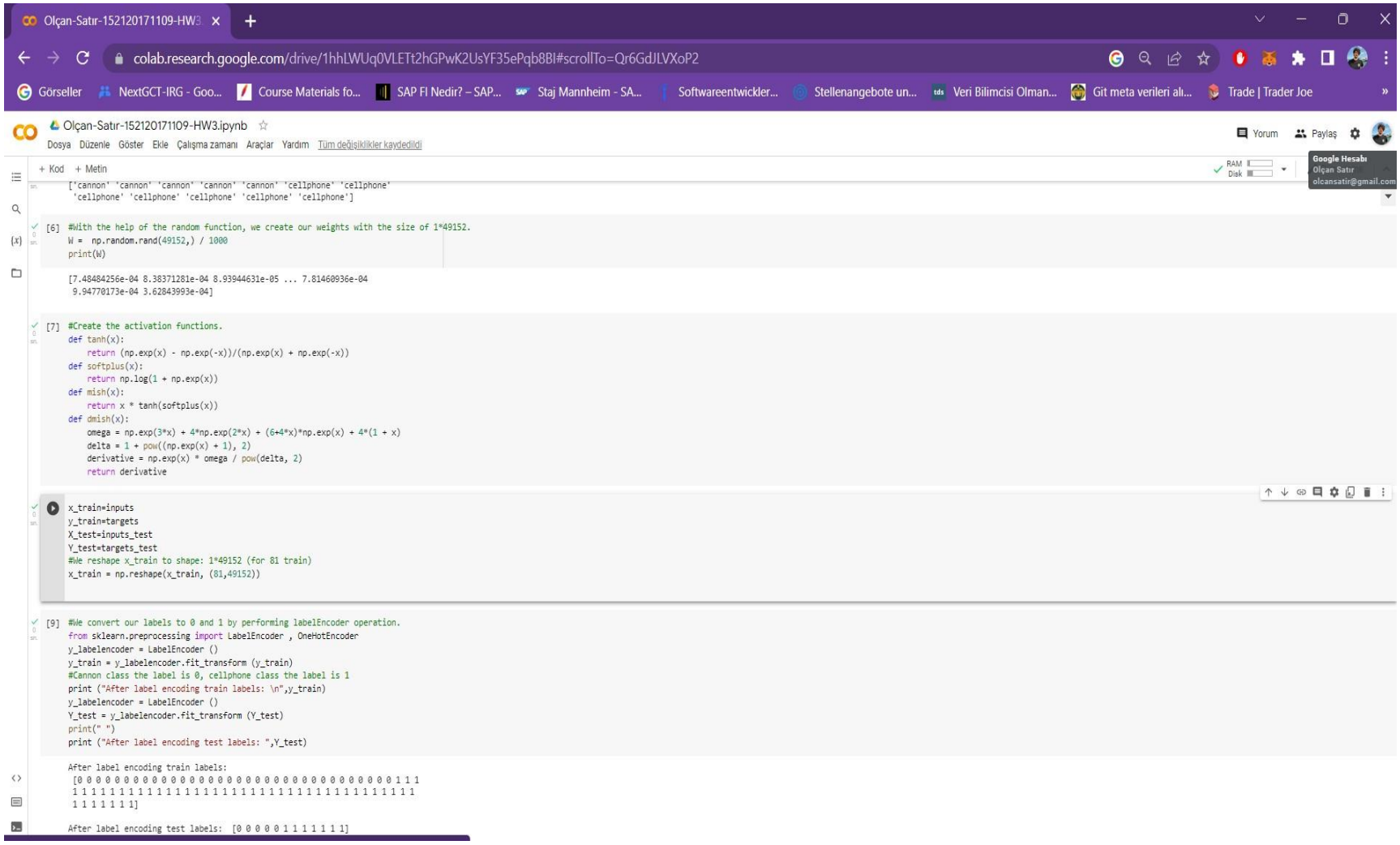


Then we resize the test pictures as 128 by 128 by converting them from BGR to RGB by naming them as inputs and targets. In the for loop below, we print 5 random pictures to see.



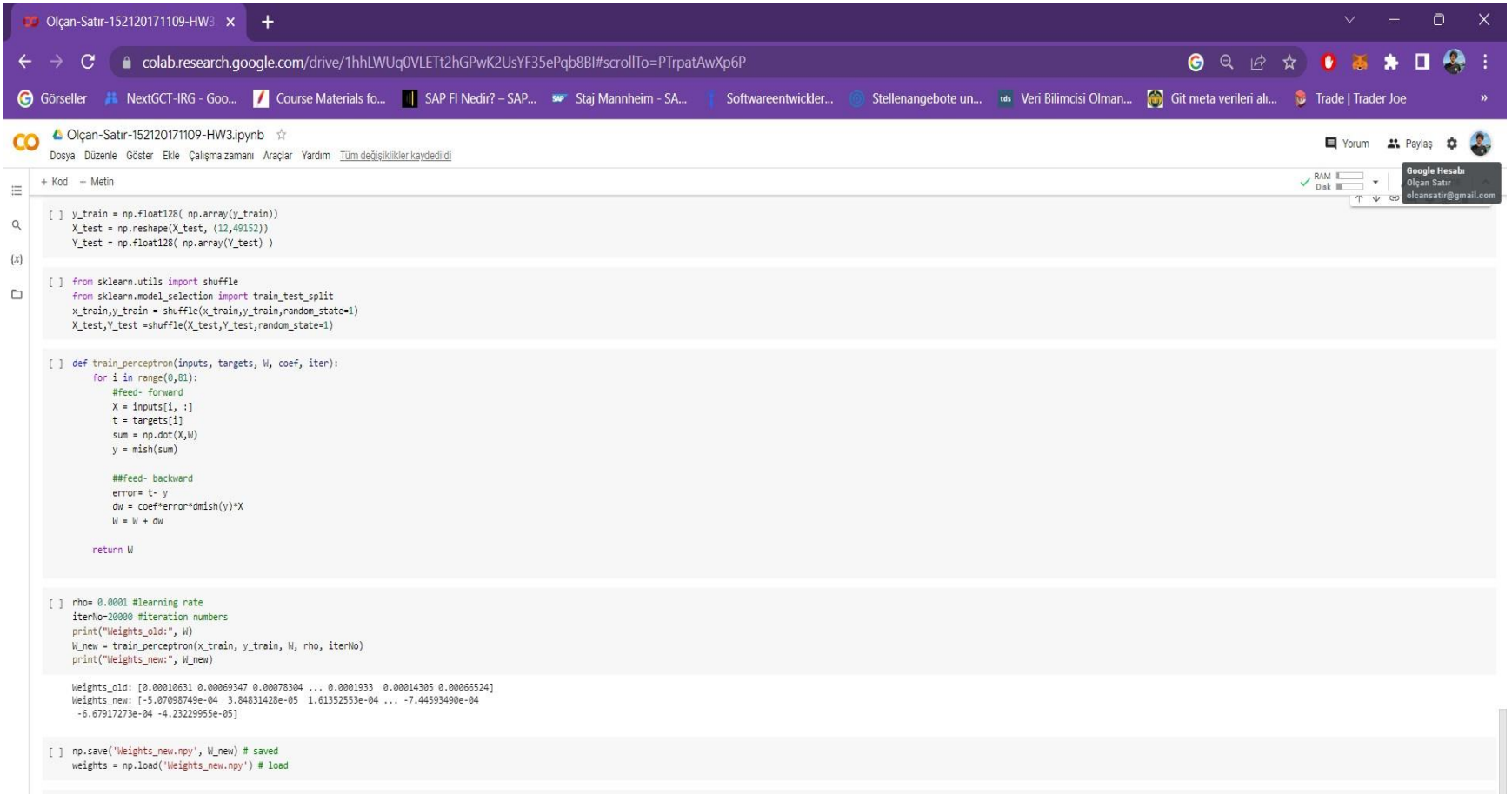
Here, we define the weights we will use for feed forward with random values in size 1x49152. Then we create the activation functions .

Then we label our images as 0 and 1 using the labelencoder.



Here we first convert y_train to an array of float128 type values. We then resize X_test for 1x49152. Then we convert Y_test into an array of float128 type values in the same way, then we perform shuffle operation for our data in the train dataset. After that, we apply feed forward to our training data using mish function, then we update the weights by calculating the error and applying feed backward.

Then, we define the learning rate and the number of iterations that we will use in the testing phase, save and load the weights.



Then we test our images with the testPerceptron function here. Here, if the relevant values are greater than 1, it takes it as 1, and if it is less than 0, it takes it as 0. When we test our 12 images, we see that they are all correctly categorized. In this case, our accuracy value is 100%.