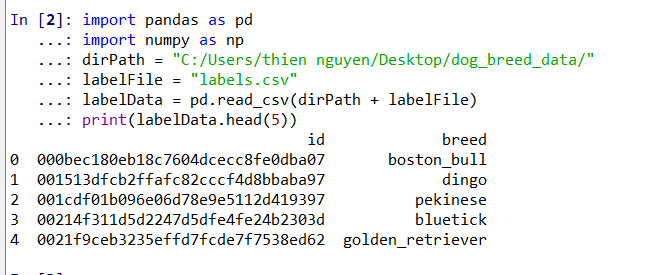
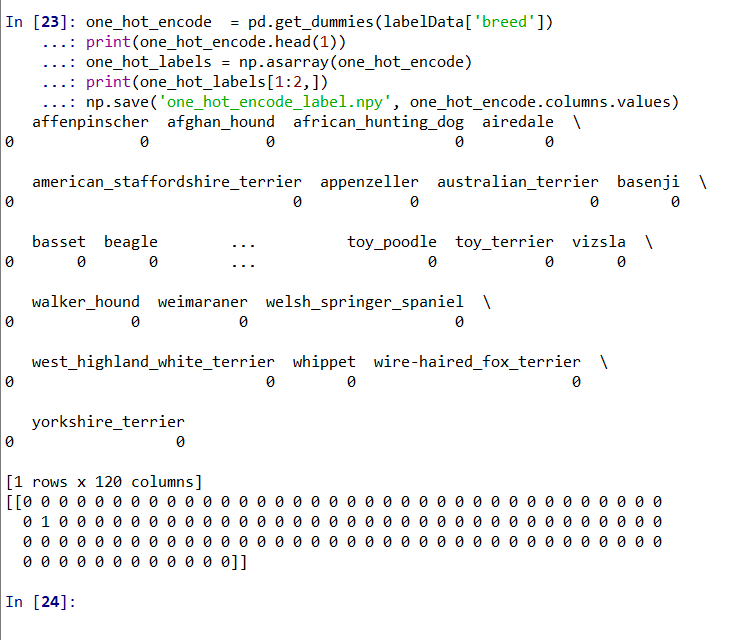
Summary

Explore and prepare the train data

1. Since breed is categorical data we need to one hot encode this



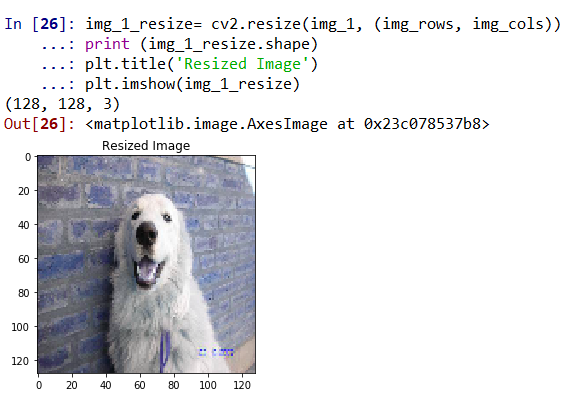
2. one\_hot\_encode now have the dog breed as column and the value of 1 in the column mean the row are classied at the breed. one\_hot\_labels is in an array form of one\_hot\_encode.



3. Try to read an image in and see if we can read it



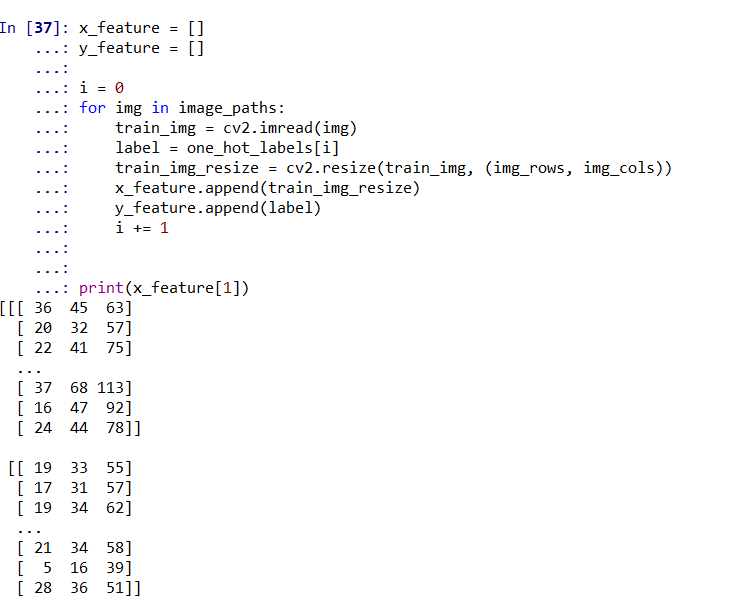
4. Then we resize an image to a smaller dimension we will eventually want to do this for all images so we get the consitent size of all images.



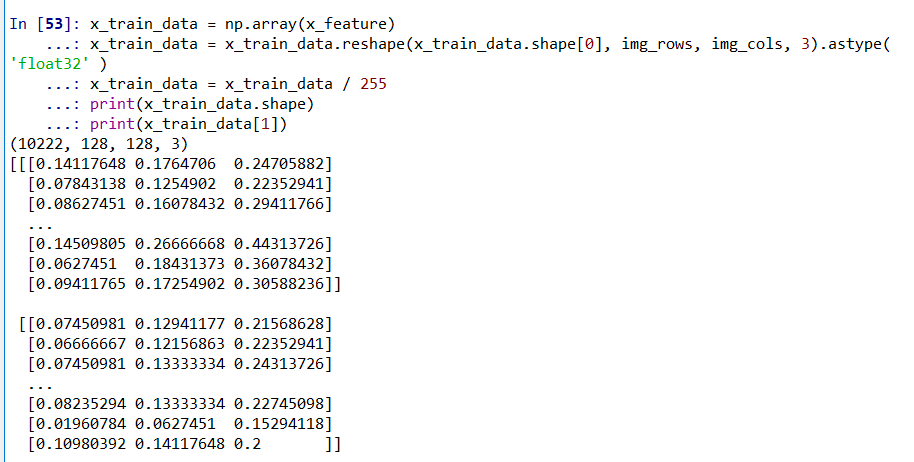
5. Create the full path for all trainning images

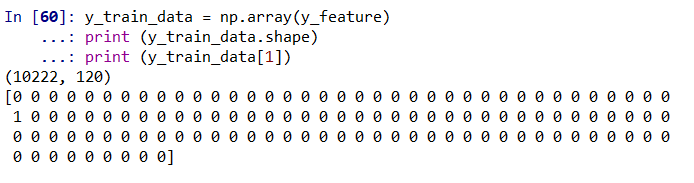


6. Resize and read all training images into x\_feautre and the coresspond one\_hot\_labels into y\_feature . as we can see the trainning images is sotred ad numpy array.

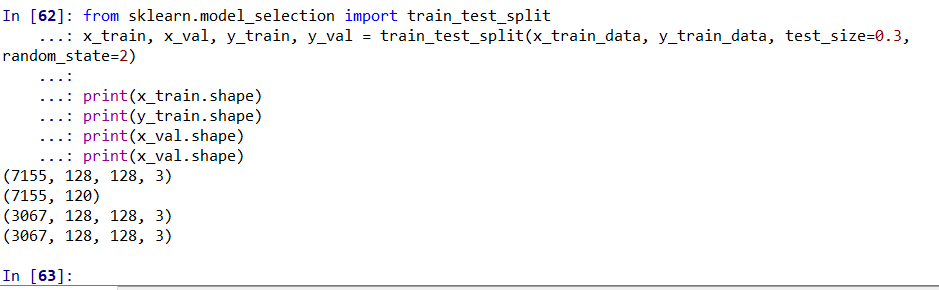


7. normalize the RGB values



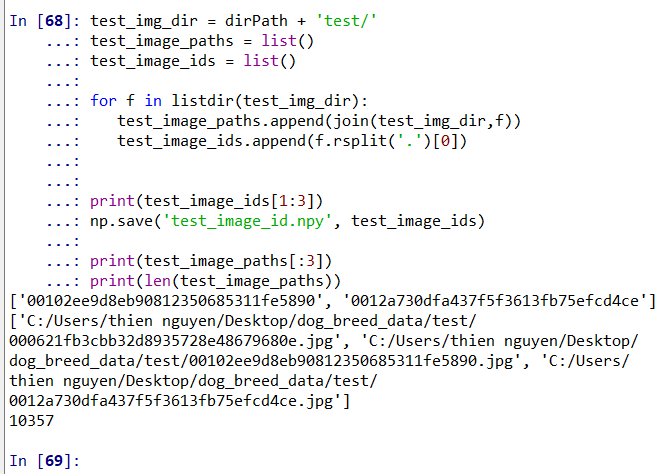


8. split train data to train and validtaion 70% train and 30% validation

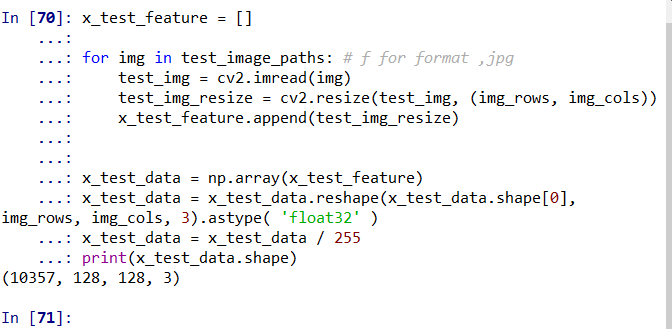


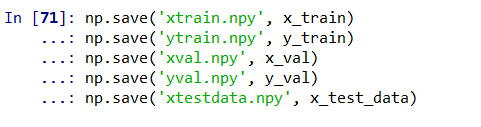
Prepare test data

9. Get all the test images full path and the unique id



10. Read and normalize test images into x\_test\_feature

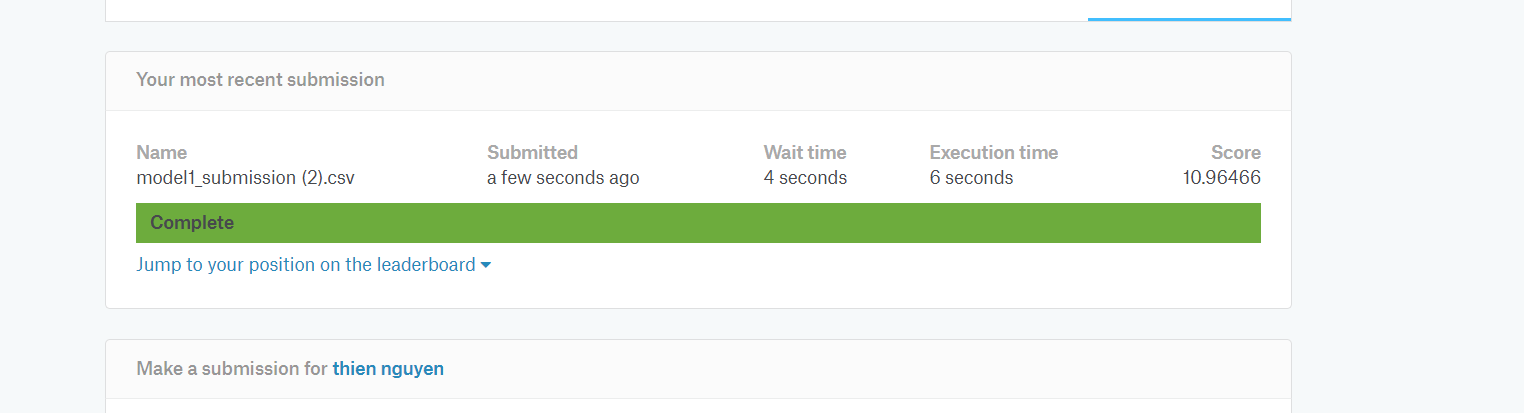
  
11. Save all the necessary data so we can put the prepared data to google drive and use google colab to build deep learning models



12. Building the baseline model1. This model is way overfited since the accuracy loss is near 0 but the validation loss is around 10. And here is the score from kaggle.

See model details in

<https://github.com/minhthien/dog-breed-classification/blob/master/dog_breed_classification.ipynb>



13. After multiple days of tuning the parameters and modify the cnn topology the model2 score around 4.48 con kaggle

The detail of the model is in

<https://github.com/minhthien/dog-breed-classification/blob/master/dog_breed_classification.ipynb>

