Cuculescu Andrei

GR. 511

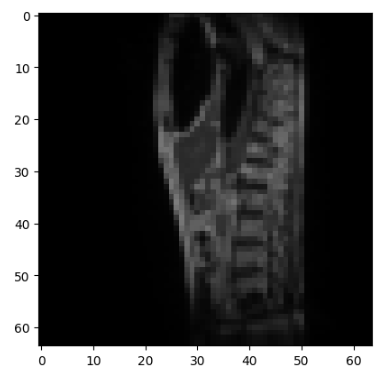
Deep Learning Project

Medical Image Multi-Label Classification

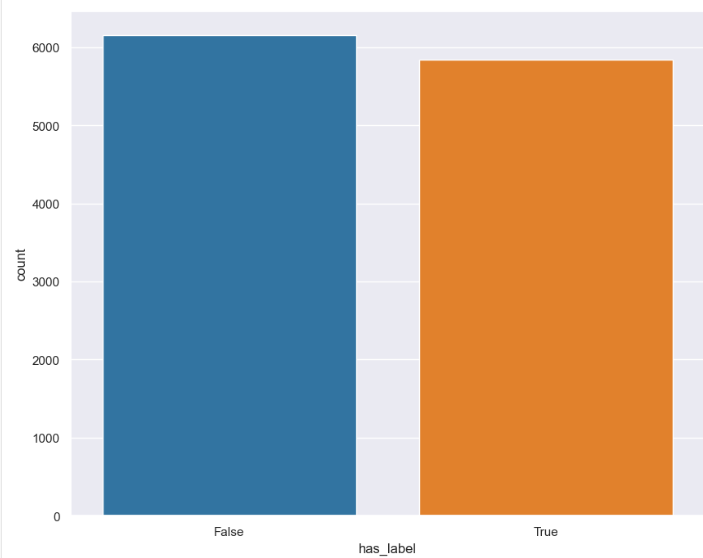
The purpose of this project was to train deep neural network classification models on x-ray images and predict the label 0 or 1 for three classes in multi-label setup.

The dataset was split in 3, the trainset composed of 12.000 images, validation set of 3.000 images and the test set of 5.000 images.

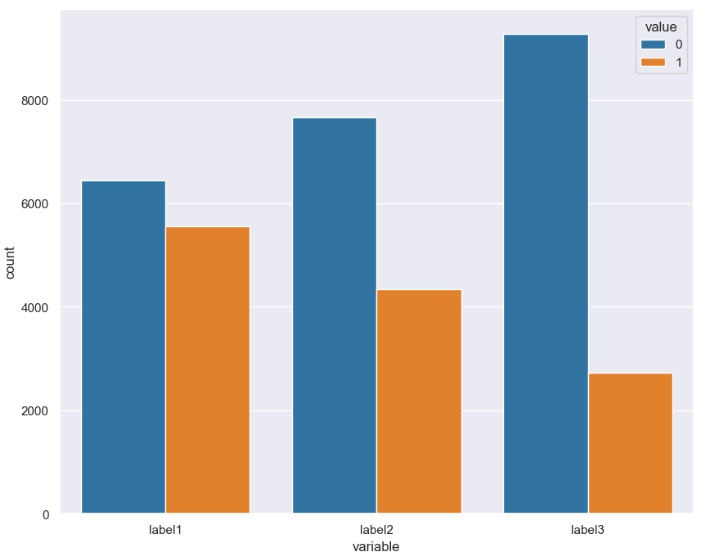
Image format was 64 x 64 pixels. (64,64,3)



In the training set we have 6.153 images that don’t have any label and 5847 that have at least one label.

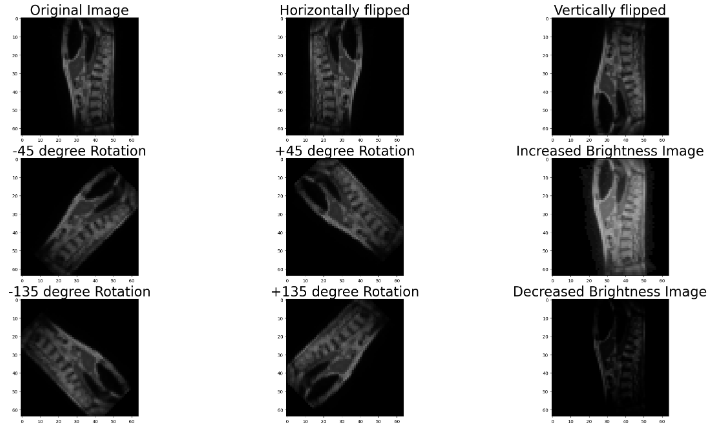


We also can see the labels values:



Data augmentation:

For data augmentation I used more types of augmentation like: flipping the image vertically or horizontally, using different angles of rotations (-45, +45, -135, +135) and increasing or decreasing the brightness of the image.



Due hardware limitation (RAM) I could only apply 4 augmentations at once and the most effective ones were horizontally and vertically flipping and brightness adjustments.

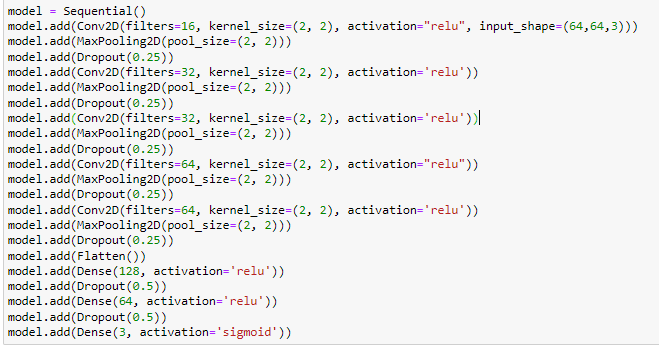
Also, for my best result I added the validation images to train images having now 15.000 images on which I applied augmentation resulting in 75.000 images. I split and shuffled those after in train and validation images with 0.1 ratio having 67.500 train images and 7500 validation images.

For best result I normalized the images to have value between 0 and 1 for better performance of the model.

Models used:

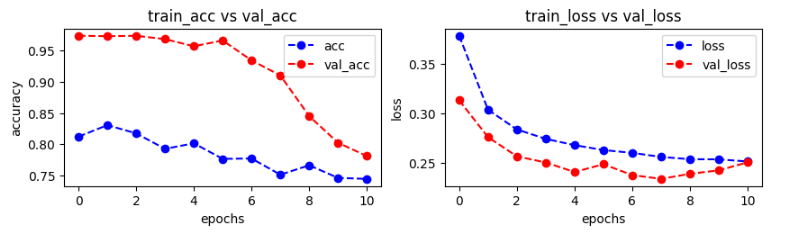
* Convolutional Neural Network
* Dense Neural Network

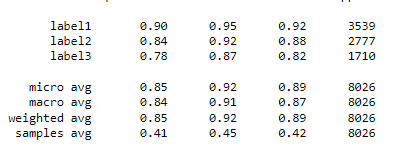
CNN is a network architecture for deep learning very useful for finding patterns in images, to recognize object, classes, patterns and categories.



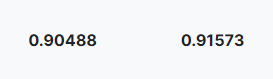
The model I used has 5 convolutional layers and 5 MaxPooling layers and 7 Dropout layers.

For optimezer I used adam and for loss I used binary crossentropy.

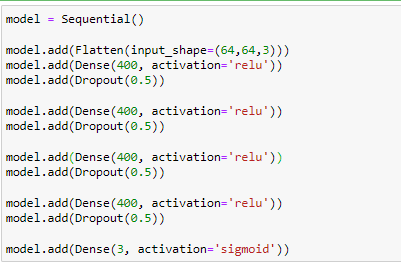




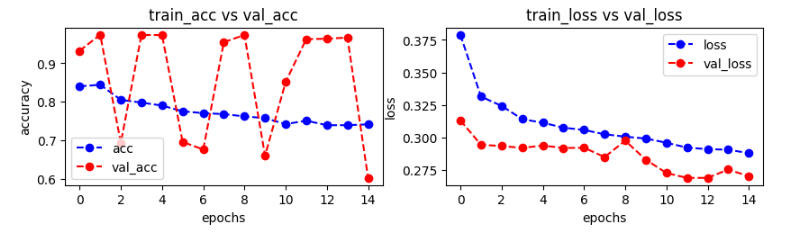
On the test data I obtained 0.90488 on private score on Kaggle and 0.91573. The training time per epoch was around 22s.

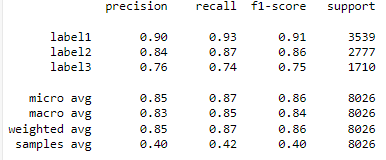


DNN is a neural network that use dense layer that are deeply connected with preceding layer. This layer is the most commonly used layer.



This is the DNN architecture I used with adam optimezer and binary crossentropy loss.

****



I used a high number of units and also increased the dropout rate with a training time per epoch of 17s I obtained a pretty close score to the CNN even if the validation accuracy fluctuates quite a lot.

