

In this 24-hour challenge, I had to build a classifier for a dataset of images that were labelled according to its position (upright, upside-down, rotated-left and rotated-right), and then according to the prediction of the built model on the test set rotate the test set images to its upright position.

First of all, this was my first time dealing with image processing and building a neural network model. So, my biggest challenge during this time was to understand the problem and the steps I had to take in order to reach the objective.

I started by transforming the images into numerical data, which was where I lost most of my time, since I had not done it before, I wasn't aware of image data generator method, and it took me quite a bit of time to get there.

Upfront I tried to downsize the images and transform them to black and white so it my modest local machine could handle it, then I realized that since I was using a pre-built model such as CIFAR10 I had to convert the images to the shape in the model. So then I started to find a way to convert them to the CIFAR10 shape (32,32,3) however during this process I realized that I needed the label for each picture as well.

By now I started to come close to the final step, I realized I had to read the label of the ground truth in a csv file, and attach it to the corresponding image name converted as an array. This actually the hardest part for, finding out how I could do this. After several attempts I discovered the Image Data generator, however I didn't find out straight away about the `flow_from_dataframe` method where I could automate the whole process. So, again I did a couple of failing attempts with image data generators, till I discovered the `flow_from_dataframe` method and it did exactly what I need.

After that the time was already short (more than 12hours passed) so I focused on understanding the parameters I needed to input to use the CIFAR10 model such as the `steps_per_epoch`. Then I set a number of epochs suggested in the CIFAR10 example page, so it would finish running in time.

After that I quickly evaluated and run some tests, and then started producing the expected result which was the rotated images according to its prediction.