R-CNN Model:

Summary:

Total params: 277,642
Trainable params: 276,266
Non-trainable params: 1,376

I have built my own R-CNN model for this project before using the pre-trained models as mentioned in the problem statement.

Dependencies:

OpenCV
Keras
Pandas
Numpy
Matplotlib
Sklearn confusion matrix

Data Preparation:

- I have prepared the data based on the previous data preprocessing results.
- Based on the 10 classes we have in the brand names we have one-hot encoded the classes.
- Normalization os the data. Using /255 way.

Building the layers and model.

I have imported the Resnet model from the Keras example and didn't do many changes to the model.

Used a stack of layers

- Conv2D resnet layer,
- BatchNormalization
- Activation for activation I have used 'relu'.

End i have used

- AveragePooling layer with pool size = 8
- Dense with softmax as we have more than 2 categories.

Image Data Preprocessing:

- I have done 5 rotations: The data we got has some images that are reversed and rotated in all directions so to generate good amount of understanding in all directions i find the need to rotate images…
- I have also used width shift range and height shift range with 0.1. As i have images that are having brand names at the edge so I find this necessary to do this.

Modeling.

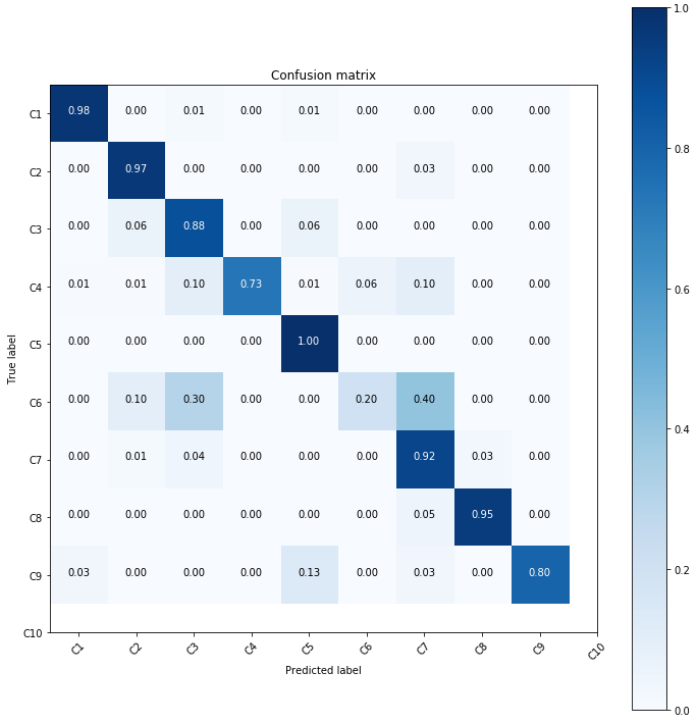
- While modeling i have used batch sizes ranging from 42 - 50
- Epochs ranged from 5 - 20
- My best model as of now having batch size 50 and epochs = 20.

Validations:

I have used Accuracy as my primary measure.

- Training Accuracy = 0.99
- Testing Accuracy = 0.87

Confusion Matrix:



Images based on CM: