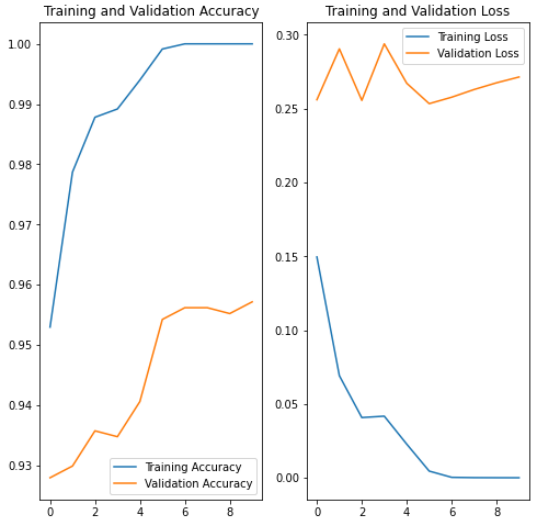
I decided to change my project since I was having difficulty to understand the model for edge detection.
Therefore, I decided to do image classification of bird species. I am using this dataset https://www.kaggle.com/datasets/gpiosenka/100-bird-species. However, instead of using the 500 species for my model which will take a lot of time to train the model, I am using 11 species which had over 200 images in them.
Also, to upload my dataset in my google collab document, I used the drive.mount function. (https://towardsdatascience.com/different-ways-to-connect-google-drive-to-a-google-colab-notebook-pt-1-de03433d2f7a)
I followed the code recommended in https://www.tensorflow.org/tutorials/images/classification to build my model. Then, I used the build-in model Sequential from the Keras library and added the same layers that were proposed. I also used https://www.tensorflow.org/guide/data_performance to try to make my model to compile faster by using the cache. By doing that, I got an accuracy of 82% which is not bad. I also used the accuracy loss metric and the adam algorithm which is a gradient descent algorithm as recommended in the tensor flow website. (https://keras.io/api/optimizers/adam/, https://www.tensorflow.org/api_docs/python/tf/data/Dataset)

Since for each class I only had 200 images, I decided to augment my dataset by using cv2 function flip and convertScaleAbs. These functions will create two new images, one that is flipped and the other that has more contrast and is brighter.(https://note.nkmk.me/en/python-opencv-numpy-rotate-flip/, https://www.tutorialspoint.com/how-to-change-the-contrast-and-brightness-of-an-image-using-opencv-in-python) By doing so, I have 95.72% of accuracy.


ORIGINAL


FLIP


BRIGTH



Then, I tried to add Conv2D layers to the model. As we can see in the following graph, the prediction accuracy on the validation set is similar to the previous results but the validation lost as decreased of at least 5%. It is this model that I will be using.

I will be using epoch=7, since for the last three test it is around that epoch that we have the lowest validation loss and accuracy. My final results are a validation accuracy 96.69% and a validation loss of 16.22%.