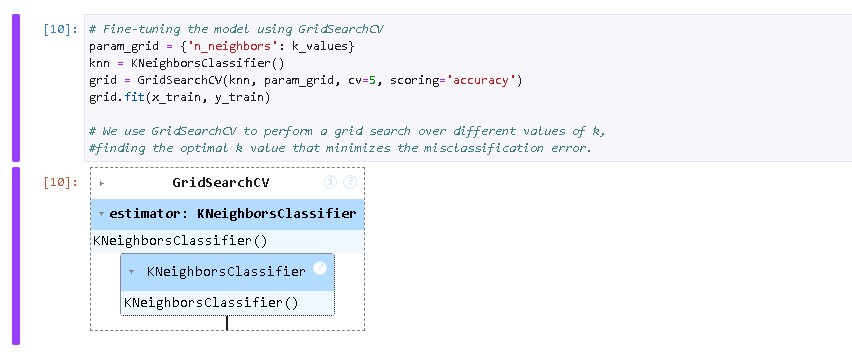
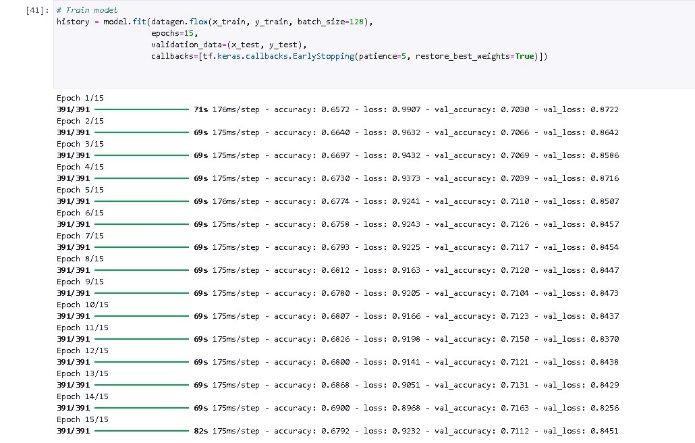
L07 "Code-Driven Object Classification Showdown"

  
Getting to know the process of the and the implementation of an AI based model made me feel challenged and interested with how it works. I can’t say it was the easiest thing to understand as I am a type of to question everything to understand it, like all the coding used in the process even though I am still at the basics of python and is also the weakness I have had for this assignment. As for the libraries, I had to install TensorFlow for the libraries to work, I have also used the default python which could be the reason for the install and then ran normally. I was stumbled for the implementation of my own code to progress the assignment which made me further interested with the assignment. Except the long wait time to training and fine tuning it which took most of my processing time. Lastly, I had issues with killing the Kernal mid training which set me back 2 hours just to retuning and retraining it. For designing my own architecture was mainly in the building stages of the model, pretty much the ending of the assignment. Like adding more conventional and pooling layers to complexify the process and make it faster and better. The augmentation and normalization are normally complementary with each other, normalization is used for making the data preprocess with the same set of data to make the process faster and prevents in biased part of the model and augmentation is for diversifying the data available for the training model increases the robustness and generalization of the model and prevents overfitting that could happen with normalization. As for trying to lower the overfitting, I experimented the batch size and the epochs to determine the training as there were changes need to happen to make sure I didn’t overfit, early stopping also made sure for overfitting, I also played around regularization (dropout) to see how it works and manage to make use of it. Lastly, I made use of a different optimizer called RMSdrop which was the bast out of the rest. Bias is basically the simplification of the model and variance is the diversity of the model. Bias and variance corelates to each other, but can be problematic if you have high of both, you need a balance of bias and variance. As for gridsearchcv, is a hyperparameter tuning which can significantly impact the performance of machine learning model. Cross-validation, is to ensure that the model’s performance is not dependent on a particular train/test split. Multiple metrics, specifying metrics for evaluation. Parallel processing, it supports parallel processing to speed up the search. And after finding the best parameters, it refits an estimator on the whole dataset. And there are some extra positives like scoring and exhaustive search. As for the ending of processing time I had with the gpu, I managed to get an accuracy with 68%. I could’ve done a 69.5% accuracy with more epochs but didn’t have more processing time for something like that.

