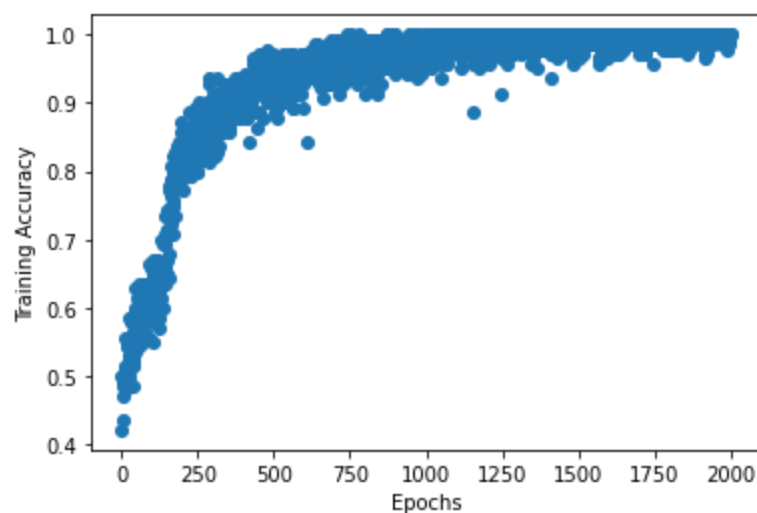
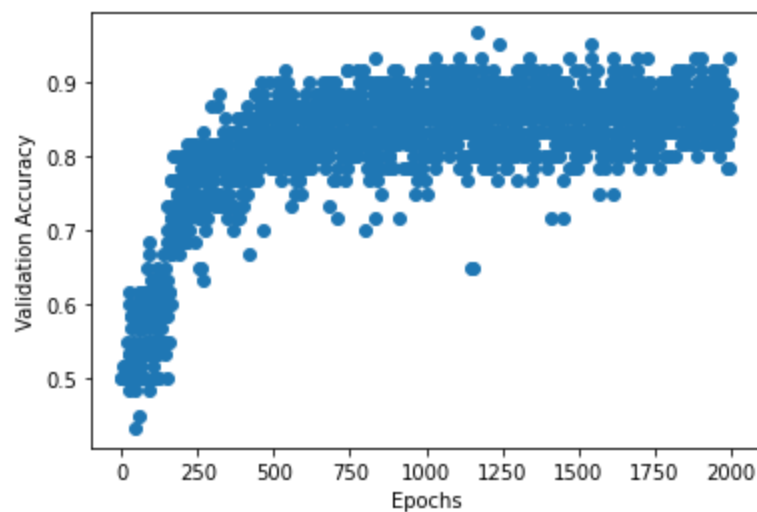


This code project is to create an image classifier to classify between my two cats: Stormy and Domino. To do this, I used the Keras library to construct my Convolutional Neural Network (CNN), with using the os and cv2 libraries to make custom directories and resize the images for making the training faster. This folder is split into 4 parts:

- This document
- The code for preprocessing the data
- The code for training and validating the CNN
- The code for taking a random image and checking whether the network thinks it is Stormy or Domino

Note: All of the codes are in the same google colab document, just that each code file is different code cells (top to bottom is the order in which the cells are executed in the colab file)

### **Some sample graphs I obtained from running the code**



### **Analysis**

There are only 200 images total in the dataset, so the massive variance in validation accuracy is to be expected. To improve this, one would add more images to the dataset so the network gets a better sense of the pattern and so the steps per epoch can be increased and the validation

steps can be increased. The training accuracy converging to 100 is also expected, as this model did not underfit the data. The accuracy curves for training and testing are both very similar, with both curves starting to decrease their slopes at around the same epochs, but not exactly the same. The network could be slightly improved by optimizing the dropout rates and the Dense layers, but something I observed during my time with playing around with the dropout rates and the Dense layers is that if the dropout rate is too high the data would underfit, so to perfectly optimize the network with the little images that there are, one would need to have really precise Dropout rates and layer architecture. So, the best way to improve the network would be both to add more data and to optimize Dropout rates and the network architecture.