Capstone Project: Dog Breed Identification Program Using Convolutional Neural Networks

Abdullah Binladin

July 19, 2022

Contents

1 Introduction

It is no secret that Artificial Intelligence is among the most prominent fields of Computer Science today. It is also no secret that big media and tech companies collect your browsing data to train AI (Artificial Intelligence) algorithms to serve various purposes (usually to personalize advertisements to you, the user). For instance, Netflix uses your viewing history to train an AI to predict what movie or T.V. Series you may be interested in watching next.

An arguably more nefarious example may be in the case of the two online marketing giants, Facebook and Google. At the forefront of A.I. research, Facebook and Google are known to log their users browsing history to use or sell it as training data for Artificial Intelligence Algorithms. The A.I in question is then used to display advertisements to the user that the AI predicts the user would be likely to click on. Both firms have been criticized for their allegedly morally dubious business practices regarding how they respect their users' right to privacy. In 2018, Facebook CEO Mark Zuckerberg was required to explain to the U.S. Congress how precisely facebook used their user's data. The interview was broadcast live and lasted an entire eight hours.

While the exact artificial neural network algorithm any given media or tech company uses is an extremely closely guarded trade secret - the tech equivalent of Coca Cola's recipe- They all generally the same abstract subcomponents, like how all sodas are some combi-

nation of fizzy water, flavouring and colour.

As artificial intelligence is so pervasive in the modern world, it is then relevant to understand how it is that AI algorithms are built and how they function. Thus, this paper assumes that the reader is comfortable with multivariate calculus and is somewhat familiar with linear algebra.

2 Project Scope

The project scope, will for the time being be defined primarily by the core AI functionality, as detailed below. A convolutional neural network will be deployed to run the software, and in the future, could be extended to a client-server relationship via an android app and an Amazon Web Services server. Specifically, as of now it is to use a convolutional neural network to find the relationship between the features of a dog and a its corresponding breed

The objective of this project experiment is to model the relationship between the features of various breeds of dogs, and to use a convolutional neural network to find these features and use them as identifying things to classify into dog breeds.

As the network used will be a simple artificial neural network, the neural network in this experiment will only have three layers of neurons:

Following from the previous report, Here we begin analyzing and training our data. With regards to training, It should be possible to use the GPU to train the model, however it is very unstable.

This will form the prototype of a phone application that contains the model trained in the attached source code. The application will do the following:

- take a picture of a dog the user finds
- run it through the model trained in the attached source code file
- inform the user of the dog breed, with the model's stated confidence.
- upload the image to a server, where it will be added to a database
- periodically, humans will sort through the images and verify the breeds of dogs
- the model will be trained with the new data
- the new model will be available as a software update for the phone app, to be downloaded and patched at the user's discretion

3 Objectives

The objective of this prototype is to experiment and determine the best AI model for the final project via experimentation, and trial and error.

4 Overall Functional Description

Currently, the prototype is limited to running locally on a computer. (note, describe the model and explain why you used it by referencing non functional requirements + the callback system used to find the best model, also explain the limitations on image size)

5 User Interface Design

For the time being, the UI is very modest - a simple window with two buttons. One is called "open", and opens the local file explorer and allows you to select an image, and the other is "predict", which will use the image you selected to predict the breed of the dog. Once clicked, the text will change to the prediction and confidence, before prompting the user to select another image. In the future, it would be preferable to have a snapchat like-interface on a phone, with verified images used to further train the model, as described in section 2.

Firstly, upon launching the program, the user sees the window depicted in figure 1. Upon doing so, the program opens a file explorer, as seen in figure 2. Following that, the user navigates to the directory holding the image, at which point they select it. The user will then see the same window as seen in figure 43. After which, the user will press "Predict", at which point the "Predict" button changes to the predicted dog breed, as well as the level of confidence. This is shown in figure 4.

At which point, the user may try again with another image by clicking "Open", selecting another image, and clicking the button below "Open", as shown in figure 4.

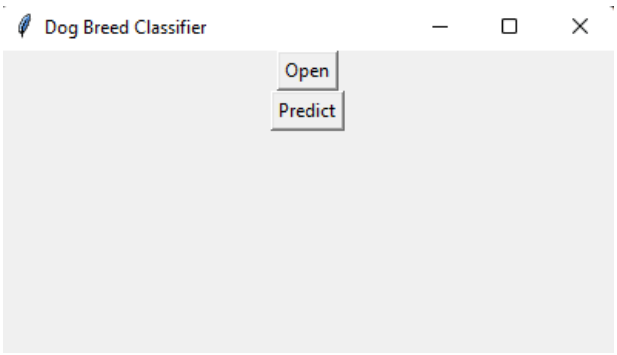

Figure 1: The initial program user interface

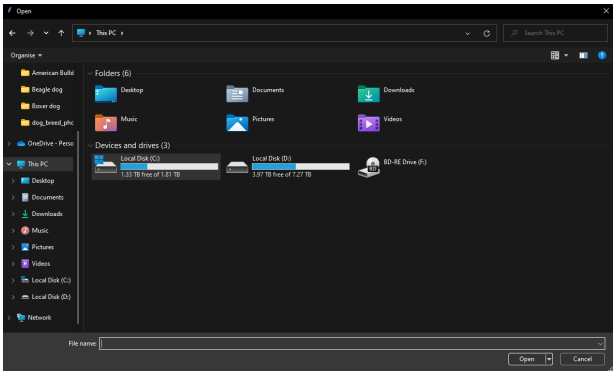

Figure 2: The file explorer

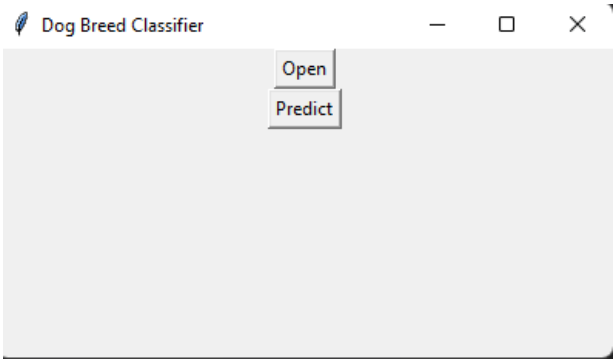

Figure 3: Upon selecting the image, the user is shown the initial UI

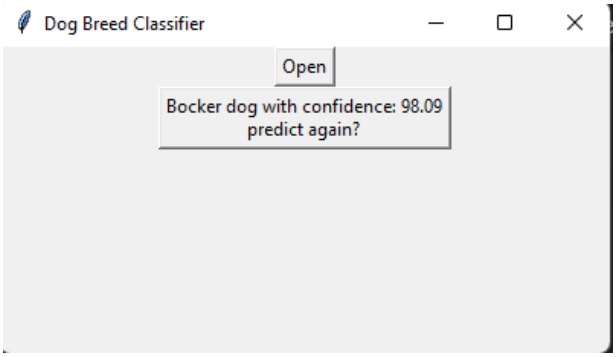

Figure 4: "Predict" is selected, and returns the dog breed with the level of confidence

6 Database Design

For the time being, a database has yet to be implemented- as the project does not require one at this rudimentary stage. Figure 5 illustrates how such a model may exist: The client will will upload images to the server, where it will be cleaned as data and added to the training dataset. Periodically, the server will run the model and train on the new data. The new model, saved in .h5 format, will be available as a patch/software update to the client. Thus the project will increase in accuracy as more data is supplied to train the model.

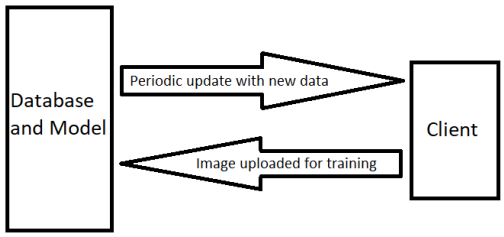

Figure 5: The high-level database model, applicable to both being run locally as well as through a client-server relationship

7 Non-Functional Requirements

The greatest current limitation is the lack of data to train on, resulting in overfitting. However that should be alleviated with more data to work with as the project. additional fine-tuning and data augmentation show that the accuracy improves as well. The figures below illustrate how fine-tuning the model (in other words, adjusting the types of layers, the parameters of the layers, and the number of neurons per layer), as well as increasing the quality of the dataset, improves the final result- justifying the need to continuously collect more high quality data over the project's lifespan.

Figures 6 and 7 refer to the the validation and loss sets over the period of training the model, both pre- and post- optimization. The major concern for commercial use is the validation set, as that is how the model reacts to new data- such as those submitted to the program via an image capture. As seen in the aforementioned figures, after adjusting the model hyperparameters and supplying higher-quality data, the point of overfitting (where the orange line approaches a horizontal asymptotse) is much closer to being correct, and the asymptote of the error graph is far lower as well. As better data is supplied, these factors will also improve.

Figure 6: Pre-Optimization accuracy and loss charts

Figure 7: Post-Optimization accuracy and loss charts

8 Implementation Plan

As previously mentioned, the most critical factor to the success of the project is the gathering of quality data. The current dataset used to train this model was sourced from https://www.kaggle.com/datasets/enashed/dog-breed-photos, kindly compiled by Emaad Nasheed.

As for the construction of the model, a series of layers of 2D convolutional neural networks will be required, as the images need to be seperated into RGB channels in order to be analyzed, then pooled together and flattened. This will require experimentation via trial and error, and the results of such should be seen in section 7. A simple mockup UI

will also be required, detailed in section 5. For the time being, this prototype will only train and save the model weights and biases locally on the host computer.