Machine Learning Engineer Nanodegree

Capstone Project

Arnesh Sahay
July 2020

I. Definition

Project Overview

Over the course of the last 30,000 years, humans have domesticated dogs and turned them into truly wonderful companions. Dogs are very diverse animals, showing considerable variation between different breeds. On an evening walk through a given neighborhood, one may encounter several dogs that bear no semblance to each other. Other breeds of dogs are so similar that it is difficult for the untrained eye to tell them apart. Included below are a few different images, see if it would be possible for the average person to determine the the breed of these dogs.

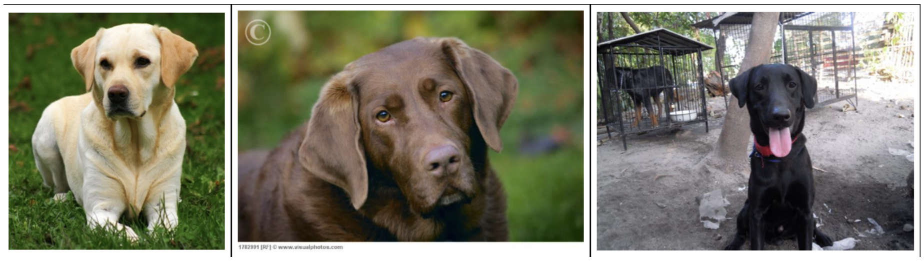

Figure 1: Labrador Retrievers are recognized as having three possible coat colors: yellow, chocolate and black.

> LTHQ et al. Silver Labrador Retriever Facts And Controversy. In *Labrador Training HQ*, 2020.

It may take many weeks or months for a person to learn enough about the physical attributes and unique features of different dog breeds to effectively identify them with a high degree of confidence. It would be interesting to see if a machine learning model can be trained to accomplish the same task in a matter of a few hours. The goal of this project is to use a Convolutional Neural Network (CNN) to train a dog breed classifier across 133 breeds of dogs, using the dogImages dataset, which consists of 8,351 total images split into 133 different categories by dog breed.

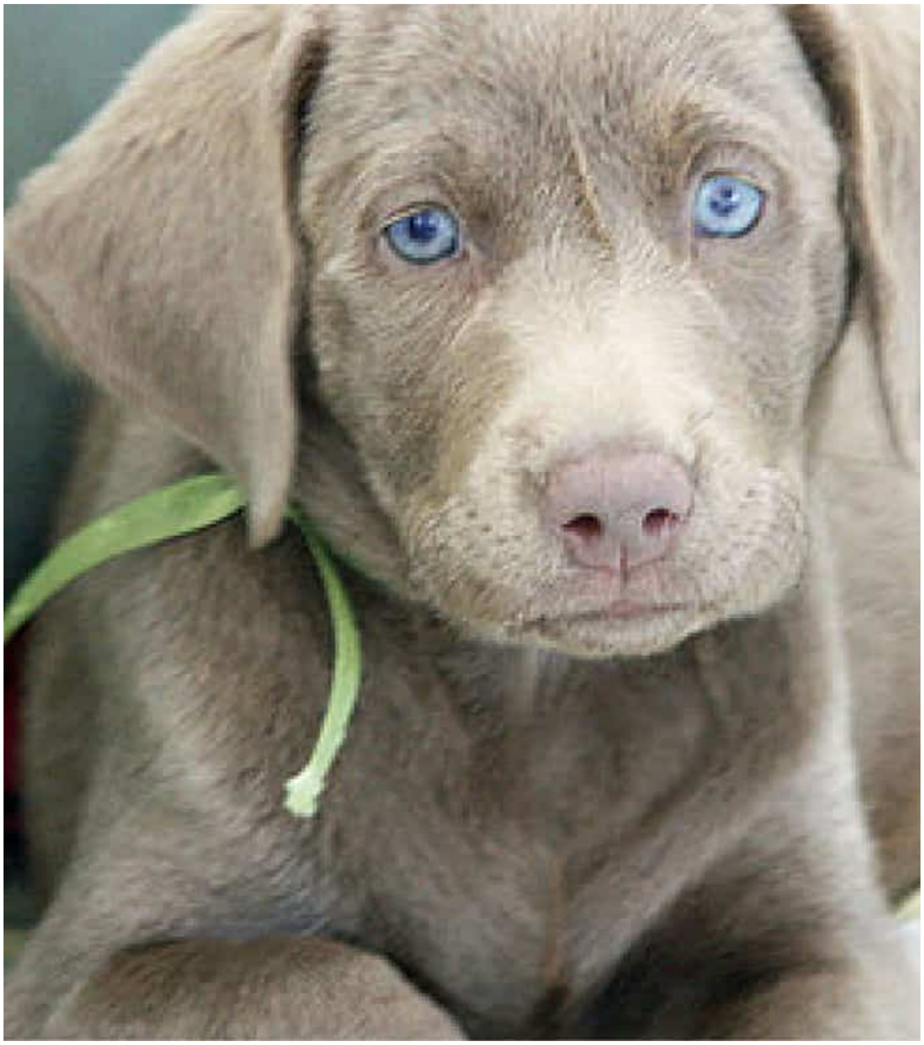

Figure 2: The legitimacy of a fourth color, silver, is widely contested amongst breeders.

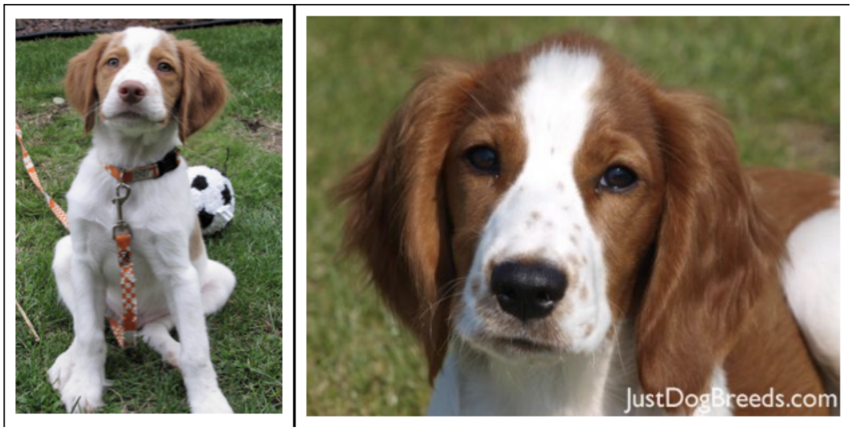

Figure 3: It is not easy to distinguish between the Brittany (left) and the Welsh Springer Spaniel (right) due to similarities in the patterned fur around their eyes.

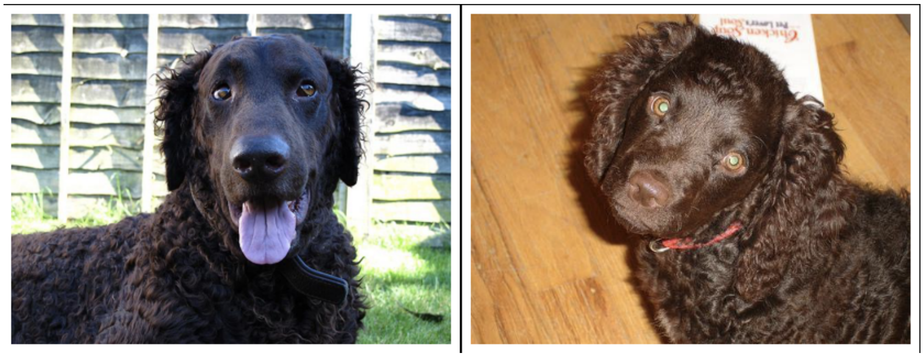

Figure 4: Another pair of dogs, the Curly-coated Retriever (left) and the American Water Spaniel (right), that are difficult to tell apart from the texture of their coats.

Problem Statement

In order to identify the breed of a dog, a model would first need to identify if the image even contains a dog. Once the presence of a dog in the image is confirmed, a multi-class classifier can then begin to determine which breed of dog the image contains. For the scope of this project, 133 breeds were included in the model.

In addition to classifying images of dogs by breed, it would be interesting if the model passed human images as valid input to the classifier as well and found the closest matching breed depending on the human's face. For that purpose, another model can be used to identify if the image contains a human. Once a human has been identified as being present in the image, it can be fed into the multi-class classifier to determine which of the 133 breeds is the closest match.

All in all, the solution should have three models: two binary classifiers to determine if the image contains a human or a dog, respectively, and a third multi-class classifier to identify the dog breed of the human or dog in the image.

Metrics

Using a test set from the `dogImages` dataset, it is possible to measure accuracy by counting `true positives` identified by the model and weighing them against the total number of predictions made. This calculation is outlined below.

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure 5: Accuracy

Accuracy is a decent measure for evaluation of machine learning models, but a far better approach is to evaluate the precision and recall of a model. Accuracy, alone, is not a very helpful metric for understanding what changed between iterations of models. For instance, if a model has very high precision but low recall, then it might be necessary to retrain the model on more data; on the other hand, if a model has good recall but lower precision, then it might be necessary to tune the features being used. Accuracy might be identical across both of the previous scenarios and does not lend very much information for improving model performance. A graphic below explains the calculation for both precision and recall.

In this section, you will need to clearly define the metrics or calculations you will use to measure performance of a model or result in your project. These

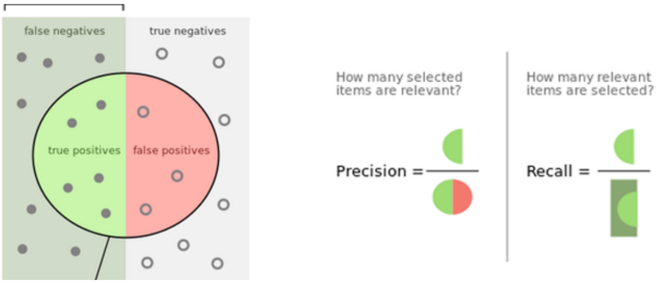

Figure 6: Precision and Recall

calculations and metrics should be justified based on the characteristics of the problem and problem domain. Questions to ask yourself when writing this section: - *Are the metrics you've chosen to measure the performance of your models clearly discussed and defined?* - *Have you provided reasonable justification for the metrics chosen based on the problem and solution?*

II. Analysis

(approx. 2-4 pages)

Data Exploration

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section: - *If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?* - *If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?* - *If a dataset is* ***not*** *present for this problem, has discussion been made about the input space or input data for your problem?* - *Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)*

Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section: - *Have you visualized a relevant characteristic or feature about the dataset or input data?* - *Is the visualization thoroughly analyzed and discussed?* - *If a plot is provided, are the axes, title, and datum clearly defined?*

Algorithms and Techniques

There are two components to the model in this project. The first is a binary classification task that will consist of determining whether an image is of a human or a dog. The second is a multi-class classification task - if the image is of a dog, the model will need to identify the breed to which it belongs; otherwise, if the image is of a human, the model will need to determine the breed that the human most closely resembles. The binary classification step will detect if a dog is present in the image and output `true` if a dog is detected or `false` otherwise. Afterwards, multi-class classification will match the image to a particular breed out of 133 possibilities.

Given an image of a dog or human, the CNN should be able to output the breed of the dog or the breed that matches most closely the human likeness. In order to measure the performance of the solution, it will make the most sense to provide the model with a series of dog images and measure its accuracy. An accuracy of greater than 60% should be acceptable for the purposes of this project.

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section: - *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?* - *Are the techniques to be used thoroughly discussed and justified?* - *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

Benchmark

The pre-trained VGG-16 model to identify dog breeds can be used as a benchmark model. It currently classifies dog breeds from a test set with an accuracy of roughly 40%, which is below the success criteria defined for this project. The goal of this project is to create a CNN that out-performs this model and delivers an accuracy of greater than 60%.

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section: - *Has some result or value been provided that acts as a benchmark for measuring performance?* - *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

III. Methodology

(approx. 3-5 pages)

Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section: - *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?* - *Based on the **Data Exploration** section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?* - *If no preprocessing is needed, has it been made clear why?*

Implementation

The end-to-end functionality of this project will employ a multitude of separate components. Any input would first be run through the `face_detector` and `dog_detector` functions to determine if the image is of a person or dog. Afterwards, the image would be fed into a CNN to determine the appropriate breed to which the image should belong.

The machine learning pipeline will include:

1. Importing the datasets
2. Binary classification
    - a. Detecting humans
    - b. Detecting dogs
3. Convolutional Neural Network (CNN)
    - a. Classifying dog breeds from scratch
    - b. Classifying dog breeds using transfer learning

4. Algorithm implementation for images provided as input
5. Model performance testing

The `face_detector` function will leverage OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images. The `dog_detector` function will utilize a pre-trained ResNet-50 model to detect dogs in images. The CCN will use transfer learning and extract bottleneck features from one of the following different pre-trained models available in Keras:

- VGG-19 bottleneck features
- ResNet-50 bottleneck features
- Inception bottleneck features
- Xception bottleneck features

To better describe the function of the CNN in this project, the illustration below shows an example of how CNNs handle image classification. The convolutional and max-pooling layers extract features from a provided input image. Those features are then used to perform non-linear transformations in the fully-connected layer and produce a classification result.

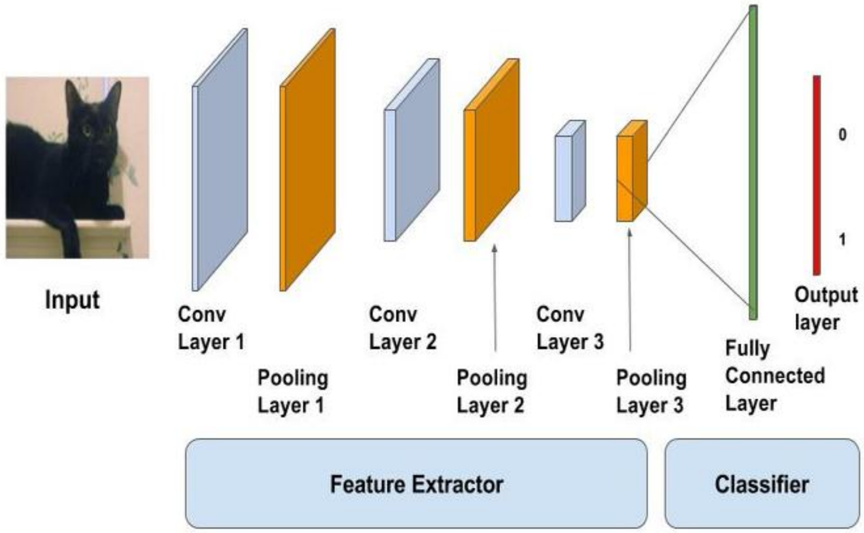

Figure 7: CNN Schema

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section: - *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data? - Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution? - Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section: - *Has an initial solution been found and clearly reported? - Is the process of improvement clearly documented, such as what techniques were used? - Are intermediate and final solutions clearly reported as the process is improved?*

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section: - *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate? - Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data? - Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results? - Can results found from the model be trusted?*

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section: - *Are the final results found stronger than the benchmark result reported earlier? - Have you thoroughly analyzed and discussed the final solution? - Is the final solution significant enough to have solved the problem?*

V. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section: - *Have you visualized a relevant or important quality about the problem, dataset, input data, or results? - Is the visualization thoroughly analyzed and discussed? - If a plot is provided, are the axes, title, and datum clearly defined?*

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section: - *Have you thoroughly summarized the entire process you used for this project? - Were there any interesting aspects of the project? - Were there any difficult aspects of the project? - Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section: - *Are there further improvements that could be made on the algorithms or techniques you used in this project? - Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*

- If you used your final solution as the new benchmark, do you think an even better solution exists?

Before submitting, ask yourself. . .

- Does the project report you've written follow a well-organized structure similar to that of the project template?
- Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
- Would the intended audience of your project be able to understand your analysis, methods, and results?
- Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?
- Is the code that implements your solution easily readable and properly commented?
- Does the code execute without error and produce results similar to those reported?