Final Project - Stanford Dogs Image Classification

Ryan King

Introduction

Over the last few years machine learning and image classification have seen incredible advancements due to deep learning techniques pushing them forward. In this ever-changing field of computer vision, one of the most fascinating and difficult aspects is classifying images into fine-grained categories, such as distinguishing between different dog breeds. This technical report focuses on the development of a machine learning model using the Stanford Dogs dataset, that properly classifies dog breeds. This dataset is derived from the ImageNet database and consists of a plethora JPEG images of unique dog breeds. It presents a unique challenge because of the tiny physical differences among the different breeds, making it an excellent opportunity to test the capabilities of image classification models.

This problem is important because there can be practical applications in numerous fields, such as veterinary medicine, animal care, and improving services involving pets. For example, the model can help veterinarians identify a dog's breed then make decisions specifically tailored to that breed. Also, most individuals have trouble differentiating between dog breeds because of the subtle differences and similarities. This model gives people a way to tell the difference without the necessary knowledge and expertise to do so.

This project explores and displays the effectiveness of convolutional neural networks (CNNs) in managing fine-grained image classification tasks. The report will provide a detailed overview of the data preparation, followed by the modeling approach including information about the model's architecture. Next, the report will discuss the results produced, and contain a reflection highlighting encountered problems and takeaways from this project.

Analysis

The Stanford Dogs dataset, from ImageNet, contains 20,580 JPEG images of 120 distinct dog breeds. Initially, the dataset was accessed and set up by unzipping the files stored on Google

Drive. Since the dataset is so large and the images take up a lot of space, this is the only method that works during the data loading process. An important part of the exploratory data analysis was understanding the distribution of images across all of the breeds. The bar chart below displays the distribution and the table shows the top ten breeds when it comes to count. Most of the breeds' counts ranged from 150 to 200 but there were some with more, displayed in the table.

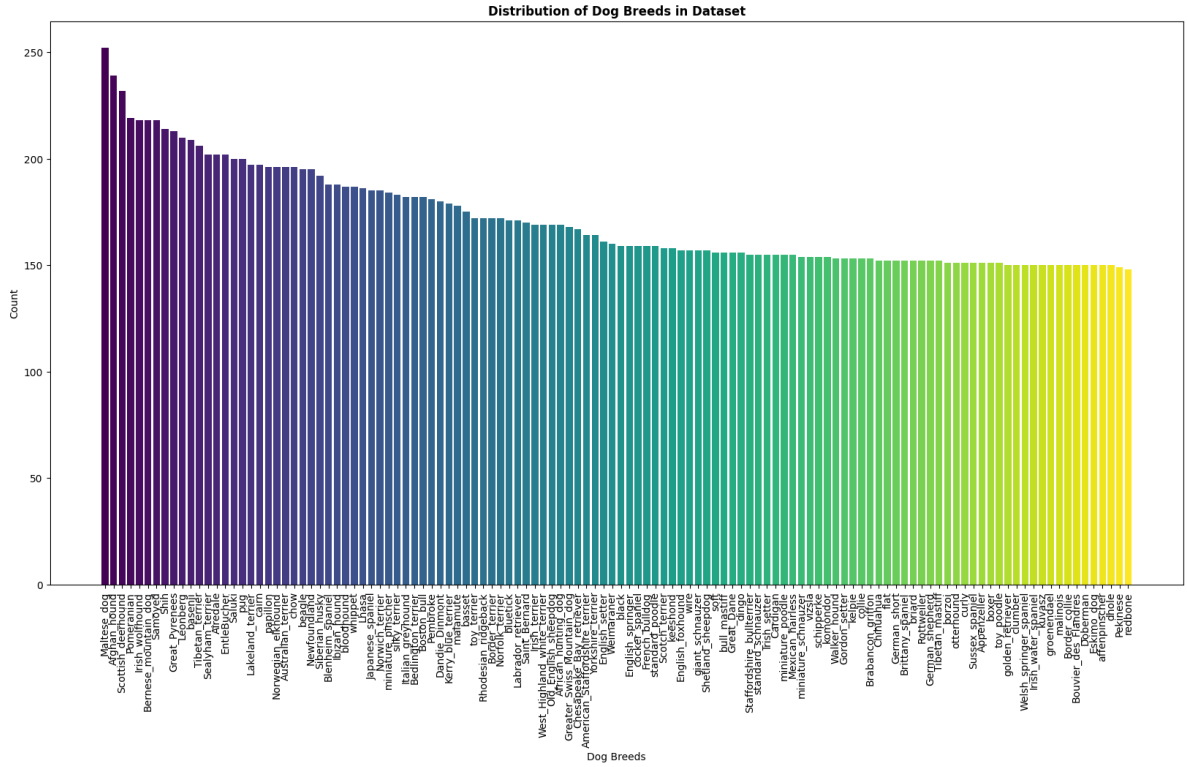

Figure 1: Distribution of Dog Breeds

Table 1: Top Breed Counts

Breed	Count
Maltese Dog	252
Afghan Hound	239
Scottish Deerhound	232
Pomeranian	219
Irish Wolfhound	218
Bernese Mountain Dog	218
Samoyed	218
Shih	214

Breed	Count
Great Pyrenees	213
Leonberg	210

Additionally, random images from the different breeds were displayed to gain a better understanding of the quality of the images, the various backgrounds and dog poses, and the angles at which the photos were taken. The images included close ups of the dogs faces, pictures with humans, full body shots, and indoor and outdoor shots. Pictured below are some examples of the different types of images the dataset contains.

This led into the next step which was preprocessing all of the images. Given the variety of images in this dataset, the preprocessing was crucial to success. All the images were given a size of 224 by 224 pixels to help maintain compatibility and consistency within the model. Also, data augmentation techniques were performed using Keras ImageDataGenerator to improve the datasets diversity. These augmentations included random horizontal flips, width and height changes, shear transformations, and rotations to the training images. They help reduce the model's chance of overfitting and improves the model's ability to generalize to unseen data. The dataset was also split into training and validation sets with an 80/20 split. The data generators were set up to train batches of 24 images to ensure efficient use of memory during the training process. The data generator also utilized one hot encoding which is essential for training the model on multi-class classification using categorical cross-entropy. The insights from the exploratory analysis and data preparation directly influenced the model building approach and training strategies seen in the next section

Methods

This section discusses the methodology in the development process of two models for the task of classifying different dog breeds. The first model is a custom designed Convolutional Neural Network, and the second is uses a pre-trained InceptionV3 architecture.

To start out, we will discuss some of the models that were attempted but ultimately did not work. Various general CNN models were explored, experimenting with different architectures and parameters. These variations mostly included changing convolutional layer amounts, filter sizes and network depths. Adjusted learning rates, batch sizes and neuron amounts in dense layers were also being altered during this process. Despite numerous attempts, most of these early models yielded very poor loss and accuracy metrics. A common issue seen was overfitting, where the models were performing well on training data but not good on the validation data. A pre-trained VGG16 model was then attempted, experimenting by implementing L2 regularization and dropout, but the results showed a training accuracy of around 77% while the validation accuracy stayed stagnent around 53%. This model showed indications of overfitting so it was not used.

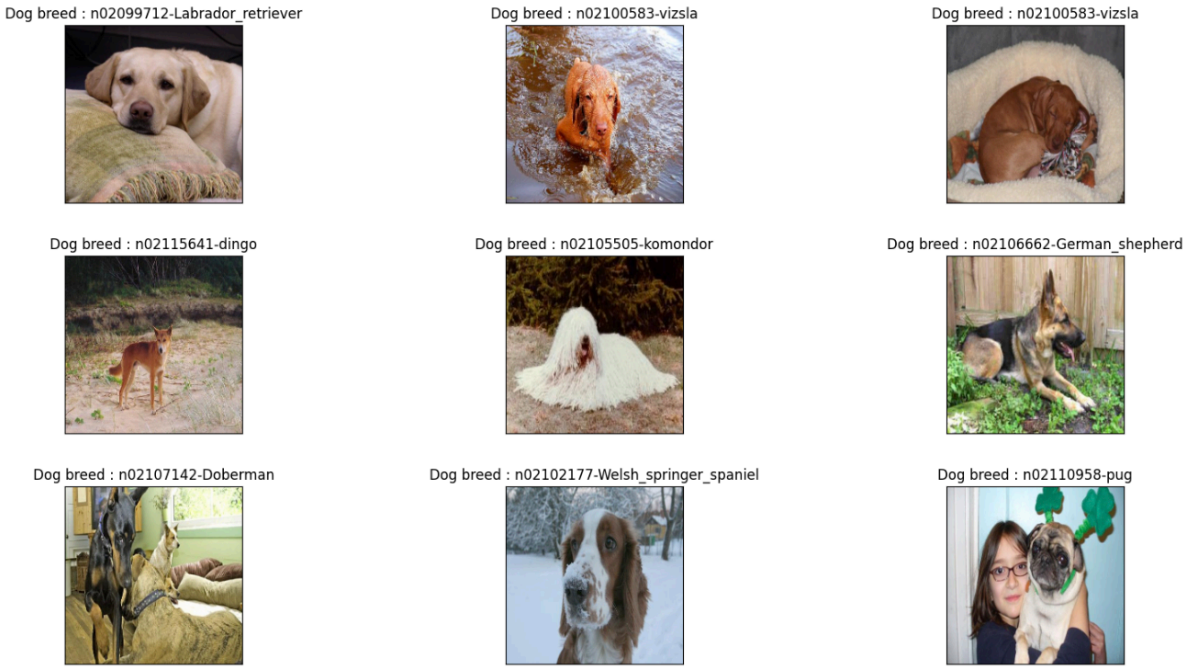

Figure 2: Random Images from Dataset

Moving onto the custom CNN, this model was designed to capture the fine-grained features that help distinguish the different dog breeds. This model started with a sequential convolution 2D layer of 16 increasing the filter sizes each layer (32, 64 and 128). Each layer included Batch Normalization and MaxPooling2D. The goal of the different layers or blocks was extract more complex features within the dog images as we got deeper into the model. The output of the convolutional layers was flattened and then fed through a 512 neuron Dense layer with a ReLU activation. This layer also included Batch Normalization like the others. Dropout was also utilized in the model with a rate of 0.5 to lower the chance of overfitting. The output layer consisted of a dense layer with a neuron amount that was equal to the total number of dog breeds and also had a softmax activation for classification. The model used the Adam optimizer, with categorical crossentropy to represent our loss and accuracy as the model performance metric. Finally, the model was then trained, using the data generators created earlier, for 25 epochs.

The other model created was an InceptionV3 model, which is pre-train on ImageNet. It contained a Global Average Pooling layer to help reduce the dimensionality of the different features. A Dense layer with 1024 neurons and a ReLU activation then followed to identify important features that the InceptionV3 had extracted. The output layer consisted of a Dense layer with a softmax activation for multi-class classification. To keep the pre-train extracted features, all layers within the base of InceptionV3 were frozen. After that, the model used an Adam optimizer to compile with categorical cross entropy as our loss function and accuracy

as the performance metric. This model was then trained for 22 epochs.

Since both of the models are trained using the same data augmentation and data split methods, we can easily compare their performance with this dataset against each other.

Results

The results of the two models are discussed thoroughly below, with an emphasis on their performance during the training and validation phases

Model 1: Custom Convolutional Neural Network Results

This model shows a consistent improvement in accuracy and loss over the 25 epochs. Initially, the model had a very high training loss of 5.3868 and a very low accuracy of 2.01%. As training progressed, the model was steadily increasing in both metrics showing a loss of 3.4765 and accuracy of 16.71% in the 10th epoch. The model display its peak training performance during the 25th epoch, where the model produced a loss of 2.7587 and an accuracy of 29.37%. The validation also had an upward trend, with the accuracy value increasing from 3.03% in the 1st epoch to 27.29% in the 25th epoch. The validation loss also went from 4.71 to 2.9126. In the figure below, we can see how the training and validation metrics were changing over the epochs.

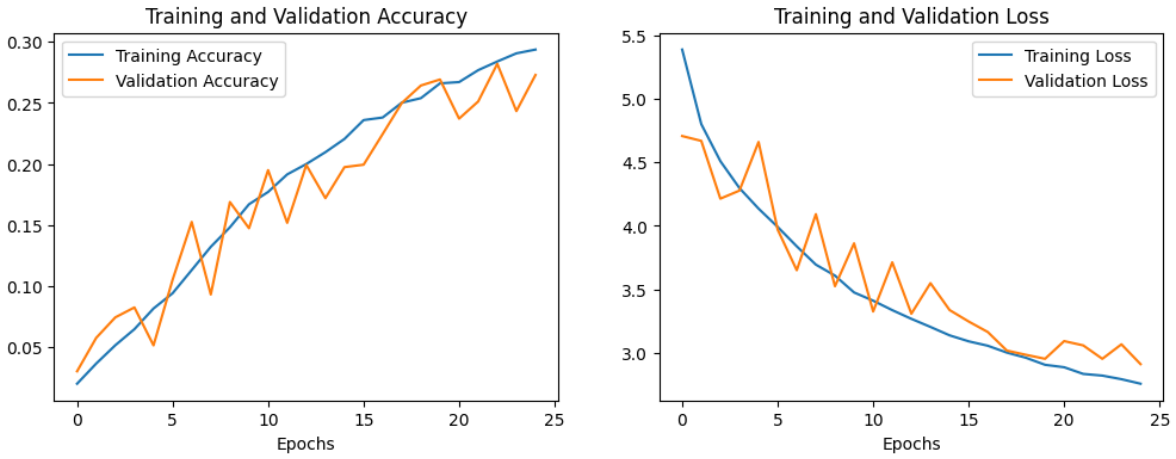

Figure 3: Model 1 Metrics

The results shown in the graphs indicate that the model is learning the dataset and continuing to improve at classifying dog breeds. However, the accuracy always stays below 30% and the loss is still pretty high. This model has significant room for improvement. Techniques like tuning the model architecture and hyperparameters were employed but they did not improve the

model's performance. Further experimentation, like changing the data augmentation method, more hyperparameter changes, and the use of regularization methods like L1 or L2 could help this model perform better in the future.

Model 2: InceptionV3 Based Model Results

This model was able to utilize the pre-trained InceptionV3 architecture, which allowed it to perform much better than the custom CNN. The model started with a training loss of 1.77 and a high accuracy of 57.19% compared to Model 1. These higher initial metrics are likely due to the pre-trained layers of InceptionV3. Like Model 1, this model consistently improved, producing an accuracy of 76.02% and a loss of 0.7887 by the 10th epoch. The model ended with a training accuracy of 80.72% and loss of 0.61. The validation performance started off very strong producing an accuracy of 77.17% and a loss of 0.78. These numbers fluctuated throughout, eventually producing a stable accuracy of 79.91% and a loss of 0.72. In the figure below, we can see the how training and validation metrics were changing over the 22 epochs.

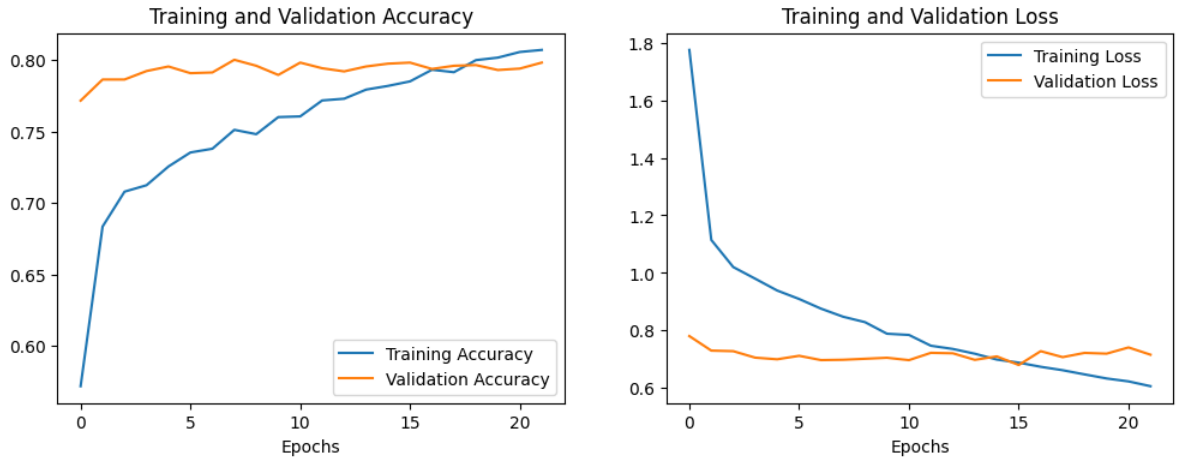

Figure 4: Model 2 Metric

The graph shows a narrow gap between the training and validation accuracy which indicates that this model generalizes well to unseen data without overfitting. It is also interesting to a see the sharp decline in training loss occur after the first epoch. This means that the model is learning a significant amount during those first few epochs. The model is quickly understanding the training data.

Overall, the InceptionV3 model showed much better performance than the custom model. It produced a mcuh higher accuracy and a lower loss. This model is better fit to classify the different dog breeds in the dataset and is more capable at differentiating between the subtle physical differences between the breeds.

Grad CAM Visualization

In addition to the metrics produced for Model 2, Gradient weighted Class Activation Mapping (Grad-CAM) visualizations were create to see what was having the biggest impact on the model's decision making. This technique highlights important features in an image that contribute most to what the model outputs.

At first, images were randomly selected from the dataset to get various examples of the breeds. The images were displayed with heatmaps highlighting important features to distinguishing between dog breeds. Some images did not work well with the Grad-CAM possibly due to the image quality or distractions within the image, but most produced insightful results. The top 5 predicted classes were also displayed so that we can see what other breeds the model thought the given dog was. Below are some examples of the Grad-CAM with explanations.

Figure 5: Grad-CAM: Boston Bull

Figure 5 shows the warmer colors around the dogs head, mainly the ears and facial area. The model was focusing on these areas as the most informative feature for classifying the Boston Bull dog breed. It was around 97% sure that it was a Boston Bull and 2.6% sure it was a toy terrier, which makes sense because both have similar ears.

Figure 6 shows a dog that is a Collie breed. The warm areas are focused around the dog's unique face and neck which are most important for determining this breed. The model showed

Figure 6: Grad-CAM: Collie

a prediction of 82.54% that it was a Collie and 17.36% that it was a Border Collie. This is interesting because the model got the correct dog breed but was also able to pick up the features of the other breed of Collie.

After using Grad-CAM on the breeds within the dataset, a study was done using images of mixed-breed dogs to see if the model could correctly predict both breeds of the dog. the heatmaps generated for mixed-breed dogs were not as clear as the purebred dogs. For most cases, the model correctly predicted one of the mixed dog's breeds but was never able to get both. In the images below, it is evident that the heatmaps struggled with mixed-breed dogs.

Figure 7: Grad-CAM: Bernedoodle

For this case in Figure 7, the heatmap is showing the warmer colors for the majority of the image. It is not exactly sure what features are important to classifying this dog. However, the model predicted at 63.77% that this is a Bernese Mountain Dog, which it is. It was able to correctly get one of the breeds but unfortunately not the other. The model predicted it to be an Afghan Hound, and EntleBucher after the Bernese Mountain Dog.

Figure 8: Grad-CAM: German Shepherd Lab

Similar to Figure 7, Figure 8 also shows a heatmap that is all over the place for this specific mixed-breed dog. The model was able to classify it as a German Shepherd at a 93.4% prediction rate but the other dogs it predicted it to be were Rottweiler and Terrier.

To improve the models performance on classifying mixed-breed dogs in the future, it is essential to incorporate more images of them in the training data. This will allow our model to learn the features necessary for classifying them. This may require the model to be trained on a dataset with multiple labels that have all the relevant breed mixes. Adding more data augmentation techniques could also help improve the model performance on mixed-breeds.

Reflection

During this project on the Stanford Dogs dataset, I was able to learn some valuable lessons and increase my knowledge of fine-grained image classification. With most of my custom models not producing strong results, I learned the power of transfer learning. These advanced architectures like InceptionV3, create a strong foundation when dealing with image classification. Deciding to implement this in my project gave me a working model that can correctly classify dog breeds at a high percentage. The project also gave me more experience with Grad-CAM. This is an aspect of image classification that I find very fascinating and I was able to use it to interpret my models results. I gained more experience working with it and explaining my findings from the generated heatmap images. I also gained a deeper understanding of CNN architectures. I began with a very simple one but slowly progressed into making it more complex, learning what changes I can make to improve the model's performance. The importance of data quality, augmentation and preprocessing became very evident to me during this project. These elements are essential to producing a successful model especially when dealing with image classification. In the beginning of the project, I was having trouble processing the data for proper use and produced models with accuracies close to zero. This may have been due to the way I was loading in the data but eventually I decided to implement data augmentation techniques and that helped improve my model's performance a lot. A limitation of the model's produced was generalizing to new data, specifically mixed breed dogs. If I had more time I would find images of mixed dogs and incorporate it into the current dataset. Some future improvements I would make are include more data involving mixed breed dogs and experiment with more advanced model architectures. I would also want to find a way to produce better heatmaps that are more refined. This would help showcase those tiny details better instead of the larger areas it was highlighting.

Overall, this project was very beneficial because I was able to apply machine learning to a practical problem. The experience has enhanced my model development abilities and my evaluation and interpretation skills of those models. As I continue to explore the field of machine learning, I will carry these takeaways with me into future projects.