Dog Breed Identification

ECS171 Machine Learning
Group 38 Project Report

Team Leader
Ishan Singh

Team Members
Joseph Cho, Sahana Hiremath, Andy Lam

June 2023

Project Github Repository:
https://github.com/ISN-SINGH/dog-breed-identification

Introduction and background

This project hopes to build a machine-learning model that will identify dog breeds when an image in inputted. The problem of identifying dog breeds is a fine-grained classification problem since all breeds, although different in name, share similar body features and structures. For example, all dogs have four legs, a tail, ears, and a snout. Because of this, differentiating between breeds can be a difficult problem. Another problem with identifying dogs is that young dogs may look different than adult dogs even though they are the same breed. The project builds off of other classification model ideas such as facial recognition models or alphabetical classifiers.

Motivation

With the abundance of cameras used for surveillance in today's world, we decided to tackle the issue of reducing the number of stray dogs. Lost dog posters are frequently left on street poles even after a dog has been found. As a result, other civilians may continue to search for a dog that is no longer missing, wasting their time and energy. Another problem with lost dog posters is that they may not reach a wide audience. Some people may not see the posters that are plastered around the city so when they encounter a stray dog, they don't know whom to contact. Through the use of computer vision techniques and statistical analysis from a machine-learning model, we can create an accurate and dependable system to help dog owners find their dogs. We can then integrate these models into existing street cameras for quick functionality, making it easier for dog owners to be contacted when a dog that looks like their missing dog is found.

Preview of Report

In this report, we will discuss the following:

- Literature Review
- Dataset Description & Exploratory Data Analysis
- Proposed Methodology
- Experimental Results
- Conclusion

The literature review will provide information about similar research projects that inspired the various tools and ideas that were used in our project. In the Dataset Description & Exploratory Data Analysis section, we will discuss how we did our preprocessing for the data used in our model. Our report will focus on a Convolutional Neural Network model that uses Stanford's Dog Dataset which can be found on Kaggle. We will provide evidence through graphs and experiment results to justify which hyper-parameter values we decided on. In the Experimental Results section, we will include performance metrics such as F1-score and accuracy for individual breeds to show how our model performed. After reading our report, we will guide people through our process in building the model as well as discuss improvements/changes that would help increase the performance of our model.

Literature Review

Machine learning principles have played a major role in facial recognition. Deep learning has risen rapidly in recent years through research as a tool for recognition models, offering the advantages of great adaptability and powerful training abilities over existing machine learning methods such as support vector machines and k-nearest neighbors. Presented in this section are examples of related work done by researchers.

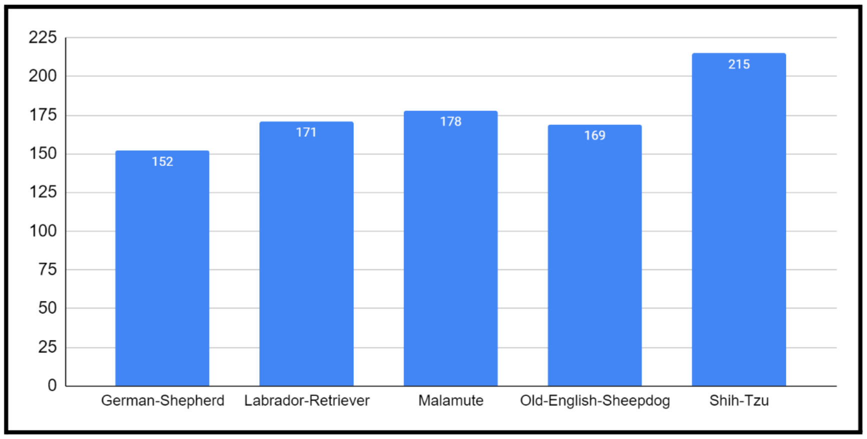
Ilyas et al. created a facial recognition system based on deep learning neural networks. The Viola-Jones face detection method was used to extract the face from the input image then passed to the Histogram Equalization Algorithm , which was used to make each image be grey level in order to reduce background color influence. After preprocessing the images, they utilized VGG16 and ResNet50-based convolutional neural networks to extract facial features and classify human faces. They achieved 96% accuracy with the VGG16 model and 97% with the ResNet50 model. Similarly, there were other works that resembled the model we chose for our project. Lam et al. used a transfer learning CNN model. The characteristics of each image were encoded into a unique code word for image representation in a codebook. A linear regression-based classifier was used for face recognition, and the method was tested on three datasets with the model performing with a 98% accuracy on each.

Raduly et al. introduced us to the method of data augmentation. They used data augmentation to reduce overfitting on training data. They also used the Inception-Resnet V2 and the NASNet-A mobile architecture as other preprocessing methods during the learning of the CNN model. The Inception-Resnet V2 trained model classifies10 different dog breeds with 100% accuracy. On the other hand, the NASNet-A trained model classifies only 1 breed with 100% accuracy, namely the sealyham terrier.

We drew inspiration from each of these research projects to create our dog breed identification CNN model.

Dataset Description and exploratory data analysis of the dataset

We found the dataset the "Stanford Dogs Dataset" for our project on Kaggle. The zip file contains 20,580 images of 120 different dog breeds which are in the form of JPG files from ImageNet for achieving fine-grained image categorization. The dataset includes annotations such as class labels and bounding boxes. We cropped the dataset to downsize it to 5 breeds for the purpose of our project. The dogs are encoded from 0 to 4 and the 5 breeds we chose are the German-Shepherd, Labrador-Retriever, Malamute, Old-English-Sheepdog, and Shih-Tzu respectively. The images are 224 by 224 pixels and thus the size of the pixel array is 50,176. The images in the original dataset are in color and not centered on the dogs. We split this dataset for training (80%), validation (10%), and testing (10%). Below is an image of the distribution of the cropped dataset containing the 5 breeds.



The Shih-Tzu is the most frequently occurring breed of dogs with 215 images whereas the German-Shepherd is the least with 152 images. The other breeds have around 170 to 180 images each. We chose this dataset as we found it to be the most suitable for training our model to correctly identify a range of different dog breeds. Since they are not in greyscale form, we need to do that manually during the preprocessing of the data. We also need to manually center the images to focus on the dogs.

Proposed methodology

We used a Convolutional Neural Network, one of the most popular models for dealing with images when trying to solve a multi-classification problem, using dogs' facial keypoints for our project. Our original dataset contained too many breeds and was resulting in low accuracy values when we trained our model. There were 120 breeds in total with around 170 images for each breed. This made the model very complex as there weren't enough samples to accurately make the necessary predictions. We also tried another model with colored images and data augmentation but we were not able to fit the training data in the model.

Thus, we cropped the dataset using bound boxing during image processing to include only 5 breeds. The bound box rectangle contains 2 coordinates to specify the location of the dog. For the rest of the image processing, we converted the images to grayscale form to simplify the computation involved and to help better distinguish the pictures. We also changed the focus to center the dogs.

We considered using a pretrained model such as ResNet-50 for image classification to handle the large complexity in our data, but did not proceed with it as we decided to train it ourselves. We then decided to develop a web application to access our trained model. Using flask, we created an interface to test our model. The model receives one image as input and outputs the percent chance that the image belongs to one of the breeds.

Experimental Results

Model Specifications

Our initial testing revealed that creating a model that could accurately distinguish between the full 120 breeds in our Stanford dataset would be an arduous endeavor unsuited for a group with a beginner's level of experience. Our hyperparameter tuning only got us as far as 7 to 8 percent accuracy this way. After aggregating our results from tuning and changing the number of breeds, we settled on classifying between 5 breeds.

The neural network consists of 4 modules each with one Conv2D layer followed by a MaxPooling2D layer. Each of the Conv2D layers has 32 filters, a kernel size of (3,3), and ReLU activation functions. After these modules we have a Flatten layer, a Dense layer, a Dropout Layer, and a final Dense output layer. The first Dense layer has 128 filters and ReLU activation function. The Dropout layer is set at a rate of 50%. The Dense output layer has 5 filters for our 5 classes and uses softmax activation function.

The Conv2D layers are the model's primary means of extracting and capturing patterns in our images. The number of filters in Conv2D layers are generally increased as a network gets deeper, in order to focus less on the noisy data at the front of the layer and calibrate to important features that remain in the deeper layers. Despite this, we found that keeping the number of filters at 32 did best to reduce overfitting and increase validation and testing accuracy. With only around 160 images per class, overfitting was a major concern for us. Size, angle, color, and body posture are just a few of the many attributes of the images of dogs that vary in our dataset. The MaxPooling2D layers deal with this by downsampling input data not only for computational efficiency but to allow the model to preserve important features for deeper layers. The Dense layers are meant to train the model on all the features the neural network has acquired so far, unlike the Conv2D layers which train the model on increasingly important features. Dense layers are commonly added as the last hidden layers and the output layer to force the model to adjust weights on all inputs.

Data Preprocessing and Feature Selection

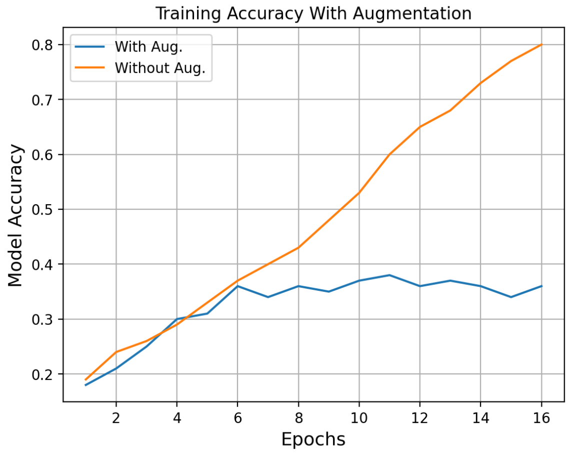


Figure 1. Models with versus without data augmentation and colored images

Before delving into hyperparameter tuning and training, we had to decide how our data should be modified. In Figure 1, the blue curve represents the training accuracy achieved with the highest performing model preprocessed with colored images and data augmentation. When attempting data augmentation, images were rotated, zoomed, and shifted in order to artificially create a larger sample size. The orange color represents the training accuracy with grayscale images and no data augmentation. We opted to train our model on grayscale images without augmentation for the simplicity it permitted. The color of a dog is hardly a predictor of breed, so it was not necessary for us to use colored images. Furthermore, training accuracy had stalled with augmentation and colored images. It is likely that the model ran into the dying ReLU problem, where neuron inputs are negative and outputs are continuously 0, never updating because of the gradient of 0 for negative values.

Performance Evaluation

```
Classification Report for Testing
              precision    recall  f1-score   support

           0       0.29      0.35      0.32        17
           1       0.42      0.48      0.45        27
           2       0.60      0.43      0.50        35
           3       0.45      0.45      0.45        22
           4       0.48      0.52      0.50        27

    accuracy                           0.45       128
   macro avg       0.45      0.45      0.44       128
weighted avg       0.47      0.45      0.46       128
```

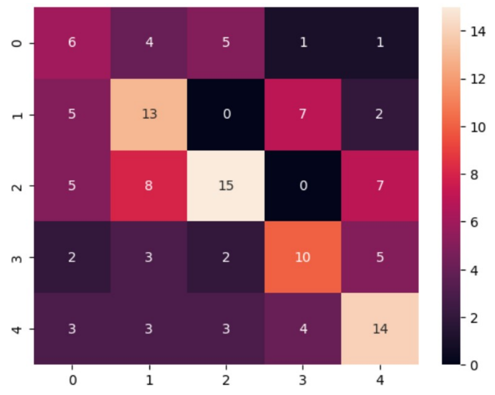


Figure 2. Testing set evaluation of best model

The 5 classes are represented alphabetically with one-hot encoding (i.e. German Shepherd is 0, Labrador Retriever is 1, etc). Our testing accuracy was 45%. The F1 scores reveal that German Shepherds (0) were correctly predicted worst out of all breeds. This is partly due to the unequal number of images per breed, and we can consequently see that German Shepherds had the lowest Support. German Shepherds only had 150 images whereas other breeds had close to 170 or more. We had thought about trimming the dataset to evenly represent each breed but this would take away from accuracy in identifying other breeds which we concluded would not be worthwhile.

Interestingly, Shih Tzus (4) tied for the best F1 score despite having an average Support. This can be attributed to the relatively large homogeneity in image features for this breed. Many images have this breed directly facing the camera with minimal variation in dog size and color. Additionally, Shih Tzus have a distinct two-layered fur pattern that the model may have easily recognized.

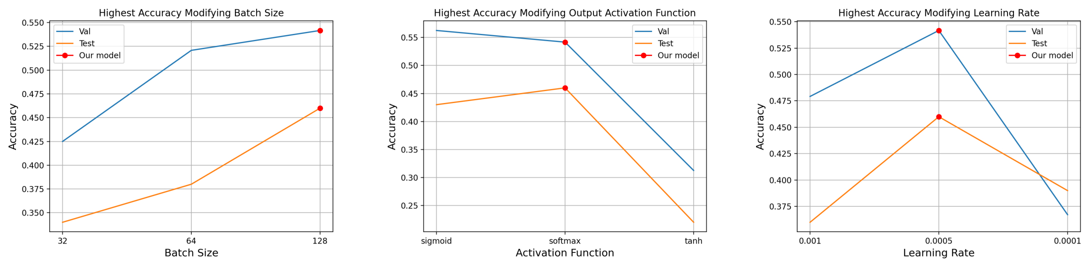


Figure 3. Graphs of three tuned hyperparameters

Three hyperparameters that were finely tuned are Batch Size, Output Layer Activation Funtion, and Learning Rate. Figure 3 shows the validation and testing accuracy as each hyperparameter is tuned. These metrics are logged from our most accurate CNN model described above. In order to keep the recorded values consistent and insightful, all hyperparameters for every graph are kept constant except for the parameter being tuned. As mentioned earlier, our group prioritized minimizing overfitting as our model had no trouble learning, comfortably reaching 95% training accuracy. Our group was using Google Colab, a cloud based python environment for machine learning, and the amount of time free users could utilize a GPU was strictly limited, so we could not perform Grid Search.

For the same reason, our group preferred larger batch sizes because this leads to less computationally intensive training. Fortunately, the largest batch size we tested (128) performed the best, and was optimal with 55 epochs. A batch size of 128 introduced the right amount of noise to prevent overfitting but maintain accuracy.

Regarding Output Layer Activation Function, sigmoid and softmax both performed well and we could have chosen either. Sigmoid was successful because it accurately maps inputs to the range (0,1), which is ideal when representing probabilities. Softmax was successful because it also outputs to the range (0,1). Furthermore, softmax calculates relative probabilities to ensure that the sum of chances of every prediction is always 1, unlike sigmoid. Evidently, softmax is most proficient for multi-label classification problems so we selected softmax as the Activation Function. Hyperbolic tangent is commonly regarded as an alternative to sigmoid, but our group saw poor results using this function. It maps inputs to the range (-1,1) and has a much steeper gradient than sigmoid which could explain its lackluster performance.

Concerning Learning Rate, out of the three values that were most tested, 0.0005 was optimum. By default the Learning Rate was set to 0.001 and until this value was changed, it was the main contributor to overfitting. A Learning Rate of 0.0001 led to underfitting such that neither training nor testing accuracy were satisfactory.

Conclusion and discussion

The 'Stanford Dogs Dataset' allowed us to create a Convolutional Neural Network with 45% accuracy in predicting the breed of a dog among five breeds. Manual cropping using bound boxes aided in reducing random noise in our images by filtering out the background. Transforming images into grayscale increased efficiency and retained the most important features.

To combat the imbalance of images per class, instead of implementing geometric data augmentation with rotations and shifts (invariably leading to underfitting) we could have tried using kernel filters to adjust sharpness and blur, or tried contrast and brightness transformations. We also could have developed a more sophisticated model capable of classifying more breeds but due to computational limitations we couldn't proceed.

With more time and computing resources we would have sought to classify all 120 breeds. An organization with greater resources and influence could certainly collaborate with local and state governments to employ new image recognition models on CCTV cameras to detect dogs. We hope that our work sparks conversation and makes an impact on the issue our society faces with losing man's best friend.

References

Dataset:

https://www.kaggle.com/datasets/jessicali9530/stanford-dogs-dataset

Literature Review:

Raduly, Zalan & Sulyok, Csaba & Vadaszi, Zsolt & Zolde, Attila. (2018). Dog Breed Identification Using Deep Learning. 000271-000276. 10.1109/SISY.2018.8524715.

Singh A, Bhatt S, Nayak V, Shah M.(2023, January). Automation of surveillance systems using deep learning and facial recognition. Int J Syst Assur Eng Manag. 2023;14(Suppl 1):236–45. doi: 10.1007/s13198-022-01844-6. Epub 2023 Jan 6. PMCID: PMC9821360.