Project Report : CS 4803/7643 Spring 2020 - Dog Breed Image Classification

Robert Firstman
Georgia Institute of Technology
rfirstman@gatech.edu

Cameron Pepe
Georgia Institute of Technology
cpepe3@gatech.edu

Abstract

Each year, approximately 3.3 million dogs enter U.S. animal shelters [7]. The primary and secondary breeds of these dogs are identified successfully only 67% of the time [2]. When asked to identify both the primary AND secondary breed of a dog, the accuracy drops to nearly 10%. The misidentification of a dog's breed can yield negative consequences on their adoptability. For instance, a dog labeled as a pit-bull mix is less likely to be adopted versus a pure-bred of a different breed. As a result, the proper identification of a shelter dog's breed is an important task to ensure that adopters are well informed and shelter animals are adopted more rapidly. We propose the use of a Deep Neural Network built on top of a pre-trained ResNeXt model to classify dog breeds from images. The use of this model achieved 84.2% classification accuracy, better than human accuracy.

1. Introduction/Background/Motivation

In this paper, we will demonstrate our attempt at constructing an image classifier to accurately identify dog breeds. Such a classifier could be used to reduce the number of inaccurate dog breed identifications in shelters across the world. The current rate of dog breed misidentification is extremely high. This can damage the adoptability of dogs if they are misidentified as an unsavory breed to the average individual looking to adopt a dog.

Nowadays, shelter operators make their judgements based on the appearance of the dog in question. This introduces significant variability in the classifications of the breeds of these dogs. A more accurate means of identifying a dog's breed comes in the form of genetic testing, but this method is costly and time-consuming. With many shelters operating on a tight budget, lab testing is not a feasible option.

Shelter operators are often invested in the well-being of the animals that come through their doors. In order to ensure that their animals are adopted, significant effort is put into making these animals appear more desirable to potential adopters. Social media campaigns and professional photo shoots are some examples of steps these shelters take. However, the inaccuracy of current dog breed identification methods can hinder their efforts. If there were a cheap and accurate means of identifying dog breeds, shelter operators and animal rights advocates would be pleased to see the outcomes of these animals improve. Adoption rates could increase considerably, and statistics regarding breed outcomes could be improved.

For this problem, we utilized the Stanford Dogs Dataset. This dataset was built by Stanford researchers using images and annotations from the ImageNet database. It includes 20,580 images of 120 different dog breeds. There are between 1 and 100 images for each breed. This dataset was created for the task of fine-grained visual classification (FGVC). This task involves the identification of hard-to-distinguish object classes, such as dog breeds, bird species, and car models. In such datasets, there are high amounts of intra-class variation and inter-class similarity. Many images of the same class appear extremely different, as the dogs of the same breed have different poses, hair lengths, and different ages. Many dog breeds look similar, as they may have similar build and coat. This poses a difficult classification task, even for humans.

2. Approach

2.1. Data Manipulation

In order to increase the diversity of the data that we had available, multiple data augmentation techniques were employed. First, training images were randomly resized to be 1 to 1.3 times their original size. Then, these images were randomly flipped across the horizontal axis. The images then underwent a random perspective transformation with a distortion scale of up to 0.2. Finally, the a random rotation of up to 20 degrees was applied to the images. By doing this, the model is forced to extract robust features from the dog images. Given the nature of FGVC and the relatively small set of data instances, it is important that the model does not overfit on the training data.

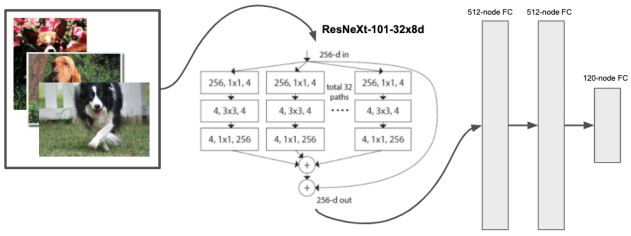
2.2. Model

We first attempted to train a simple Convolutional Neural Network (CNN) on the dataset. This CNN had 2 convolutional layers followed by an affine linear layer. Given the complexity of this fine-grained visual classification problem, and the relatively small size of the dataset, we did not expect good results from this basic approach. Unsurprisingly, this basic CNN model was only able to obtain a test accuracy that was barely better than chance (.932% accuracy).

After training a simple CNN, we trained a much deeper CNN with 4 convolutional segments, each containing a convolutional layer, batch norm, ReLU, max pool, and dropout followed by two affine layers. This network also had a low test accuracy of 14%, although this was considerably better than the simple CNN. This larger, more expressive model was much better suited for this complex task than the simple model, but the training set lacked enough data to appropriately train a more complex model.

Once we experimented with making CNNs from scratch, we shifted our attention towards transfer learning. In cases where there is insufficient training data available, transfer learning has been shown to be an effective method for training [6]. Using transfer learning, the model is pretrained on ImageNet1000, which contains over 14 million images of 1000 different classifications, allowing for the model's feature maps to be sufficiently developed. We selected four models for our approach: DenseNet, ResNeXt, Wide ResNet, and Inception v3 [4, 8, 9, 5]. For each model, we froze the weights on each convolutional layer and replaced the final layers with random-weighted fully-connected linear layers. We then trained each modified network, only adjusting the final linear layers' parameters using Stochastic Gradient Descent (SGD). Hyperparameter tuning was used on each network to find the optimal learning rates, momentum values, and training batch size.

Figure 1. ResNeXt-101-32x8d



The ultimate model used is the ResNeXt-101-32x8d model followed by 3 affine linear layers, two with 512 nodes followed by one with a 120-node layer as the classifier layer. This model can be seen in Figure 1. The optimizer used for training was Stochastic Gradient Descent using Cross Entropy loss. The optimal hyperparameters for training the model can be found in Table 1.

Hyperparameter	Value
batch size	32
learning rate	0.01
momentum	0.9
epochs	20
learning rate decay step size	7 epochs
learning rate decay gamma	0.1
weight decay	0.0

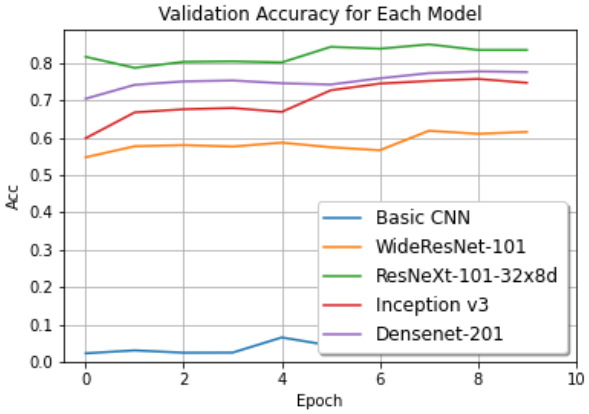
Table 1. Optimal Training Hyperparameters

3. Experiments and Results

The primary success metric was classification accuracy. This ultimately determines the effectiveness of this classification model. Various experiments were run in order to determine the best pre-trained model to use for transfer learning, the best method of transfer learning, and the optimal training hyperparameters.

In order to determine the best model to use for this task between the four pre-trained models, we ran various experiments using optimal hyperparameters for each model (found via hyperparameter tuning) and compared the loss and validation accuracy between each of the models. Each pre-trained model was followed by 3 fully-connected layers (same architecture as described above). The validation accuracy of each model over 10 training epochs can be found in Figure 2. The accuracy of each of the models generally improved with each epoch as the fully connected layers were trained. After about 5 epochs, the marginal improvements of the validation accuracy of each of the models became small. The resulting test accuracy of the five different fully trained networks with optimal hyperparameters can be found in Table 2.

Figure 2. Validation Accuracy of Different Pre-Trained Models



Model	Test Accuracy %
Basic CNN	11.1
WideResNet-101	62.9
Inception v3	76.4
DenseNet-201	78.1
ResNeXt-101-32x8d	**84.2**

Table 2. Resulting Test Accuracy of Different Models Tested

Transfer learning method	Test accuracy %
Fine-tuning the CNN	75.0
CNN as a fixed feature extractor	84.2

Table 3. Finetuning the CNN vs. Using CNN as Fixed Feature Extractor Accuracy.

	Precision	Recall	F1 Score
Chihuahua	0.71	0.87	0.78
Japanese Spaniel	0.88	0.82	0.85
Maltese Dog	0.96	0.85	0.90
Pekinese	0.88	0.86	0.87
Shih-Tzu	0.79	0.77	0.78
Saint Bernard	0.97	0.91	0.94
Weighted Average	0.85	0.84	0.85

Table 4. Sample classification report

Two methods of transfer learning are widely used: fine-tuning the CNN and using the CNN as a fixed feature extractor [6]. When fine-tuning the network, a small learning rate is used to make incremental changes to the overall network. When using the network as a fixed feature extractor, the convolutional layers of the network are frozen and utilized to produce a feature vector, which is then fed into a classifier. Both methods were tested using the ResNeXt-101-32x8d. The test accuracy of each method is shown in Table 3.

The optimal trained model had a test accuracy of 84.2% with a training accuracy of 87%. This model showed strong validation accuracy after only the first training epoch, showing the power of transfer learning. The validation accuracy during training can be seen in Figure 3. The Stanford Dogs Dataset is a subset of ImageNet. Using models pre-trained on ImageNet as a fixed feature extractor allowed for strong results immediately. The training accuracy is barely better than the validation accuracy, meaning the model did not significantly overfit the training data.

Figure 4 shows the model's output for different test images. The confidence scores were obtained from using the Softmax function on the output and taking the Softmax value of the predicted class. In this example, the confidence of all the predictions is higher than some other examples. In cases, where the prediction is incorrect, the Softmax value ranges between $<$10% and 80%.

In Table 4, precision, recall, and F1 score values are demonstrated for some sample dog breeds. The chihuahua

Figure 3. Train and Validation Accuracy of ResNeXt Model

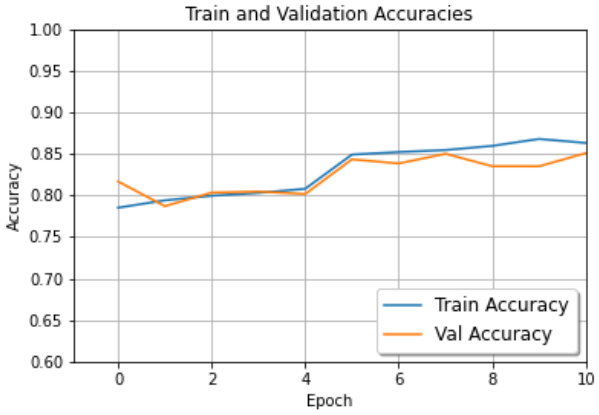
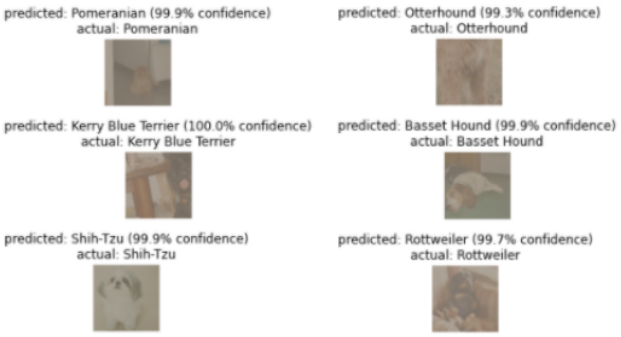


Figure 4. Model Output - Predictions and Confidence Score



class received a lower than average precision score of 0.71, which means that the model has a higher false positive rate for this breed. This could likely be due to chihuahuas sharing certain features with other small dog breeds, such as a large eye size to head size ratio. Meanwhile, Maltese dogs have a very high precision rate of 0.96. Very few images are misclassified as a Maltese, and this is likely due to their distinctive, long white coat. The model obtained both high precision and recall values for the Saint Bernard class. This is likely due to this breed's distinctive appearance with their unique and consistent fur pattern and large size.

This model was largely a success for this task. With all the inter-class similarity and intra-class variation that fine-grained visual classification problems present, 84.2% accuracy is great. This accuracy is also better than typical human classification accuracy of 67%. Upon analyzing the misclassified images, many of the dogs' breeds in the image were difficult to determine even to the human eye [2].

4. Future Work

Future work on this task could include implementing more complex models used specifically for fine-grained visual classification, such as using Batch Confusion Norm with Attention-based Gated Atrous Spatial Pyramid Pooling [3]. The purpose of batch confusion norm is to infuse slight classification confusions into the FGVC training procedure and drive the learning to work harder for making as many correct predictions in each training bag as possible. The Gated Atrous Spatial Pyramid Pooling expanded the base model to allow for simultaneous feature extraction and attention heatmap learning. This model showed state-of-the-art classification accuracy for various fine-grained visual classification tasks (CUB-200-2011, Stanford Cars, FGVC-Aircraft).

When a human is asked to identify a dog breed, they will often compare the dog with another of known breed to come to a conclusion. Leveraging this method in a dog breed classification model could yield improved results. Related literature utilizes pairwise learning methods to achieve high accuracy on fine-grained classification datasets. [1, 10]. Such "Siamese" networks compare image pairs to inform classification. By utilizing semantic differences between class instances, an image could be classified by comparing it to similar yet distinct examples.

In addition, the model could be expanded to include classification of mixed breeds. Many dogs at shelters are not purebred, so the expansion would prove extremely useful for shelters and dog adoption centers. This expansion would require either a data set that included mixed breeds, or altering the final layer to classify a dog as multiple breeds if two (or more) Softmax probabilities are over a certain threshold.

Beyond future work to this model, this model could be used with a mobile application, allowing the employees of the animal shelter to classify the dog breeds on the fly.

5. Work Division

See Table 5 on the following page.

References

[1] Abhimanyu Dubey, Otkrist Gupta, Pei Guo, Ramesh Raskar, Ryan Farrell, and Nikhil Naik. Training with confusion for fine-grained visual classification. *CoRR*, abs/1705.08016, 2017. 4

[2] Lisa M. Gunter, Rebecca T. Barber, and Clive D. L. Wynne. A canine identity crisis: Genetic breed heritage testing of shelter dogs. *PLOS ONE*, 13(8):1–16, 08 2018. 1, 3

[3] Yen-Chi Hsu, Cheng-Yao Hong, Ding-Jie Chen, Ming-Sui Lee, Davi Geiger, and Tyng-Luh Liu. Fine-grained visual recognition with batch confusion norm. *CoRR*, abs/1910.12423, 2019. 4

[4] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. 2

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. 2

[6] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. *CoRR*, abs/1808.01974, 2018. 2, 3

[7] The American Society for the Prevention of Cruelty to Animals. Shelter intake and surrender, pet statistics, 2016. https://www.aspca.org/animal-homelessness/shelter-intake-and-surrender/pet-statistics, Accessed: 2020-04-29. 1

[8] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016. 2

[9] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. 2

[10] Peiqin Zhuang, Yali Wang, and Yu Qiao. Learning attentive pairwise interaction for fine-grained classification, 2020. 4

Student Name	Contributed Aspects	Details
Rob Firstman	Initial Setup Scratch CNN models Transfer Learning Setup Contributed to Paper	Created the base setup for the data loading and model.
Cameron Pepe	Hyperparameter Tuning Ran Experiments Contributed to Paper	Set up method for hyperparameter tuning. Ran various experiments on the models and created corresponding graphs

Table 5. Contributions of team members.