Fine-grained image classification of Stanford dogs

1. Introduction

A demanding and large-scale dataset that aims to achieve fine-grained picture classification is presented here in the form of a 120-class dataset called the Stanford Dogs dataset. There are almost 20,580 annotated photos of dogs belonging to 120 different species that are included in Stanford Dogs. Object class labels and bounding boxes are added to each picture as part of the annotation process. This dataset presents a great deal of difficulty for several different reasons. In the first place, since this is a fine-grained classification issue, there is not much difference across the classes. For instance, the basset hound and the bloodhound have very similar face qualities, but their colors are extremely different from one another. On the other hand, the Japanese spaniel and the papillion have very similar colors, but their facial characteristics are very different from one another. Two, there is a significant amount of variety within the same class. Dogs belonging to the same class may exhibit a variety of characteristics, including varying ages (for example, a beagle), positions (for example, a blenheim spaniel), occlusion/self-occlusion, and even color (for example, a Shih-tzu). In addition, in contrast to previous animal datasets, which are often found in natural settings, a significant number of the photographs involve people and are captured in man-made situations, which results in a larger degree of background variance. For the reasons that have been described above, this dataset is exceedingly difficult to work.

Recent years have seen amazing breakthroughs in the field of picture classification, which have made it possible for computers to detect objects and situations with an extraordinary degree of precision. The subfield of image classification known as fine-grained image classification, on the other hand, poses a special problem. In this context, the emphasis moves from broad categories such as "car" or "chair" to recognizing small differences within classes that are quite similar to one another. The classification of dog breeds is an example of an application that is very intriguing. This is a work that requires a sharp eye for detail as well as powerful picture processing tools [1].

In this introduction, we dig into the fascinating area of fine-grained picture categorization by making use of the Stanford Dogs dataset, which is a comprehensive collection that includes 120 different breeds of dogs. To tackle this task, we will investigate the numerous factors that are involved, such as the dataset itself, the intricacies that it provides, and the many deep learning algorithms that are utilised to accomplish correct breed identification [2].

A treasure trove of canine diversity, the Stanford Dogs Dataset is a treasure collection.

Researchers in the field of fine-grained image categorization for dog breeds rely heavily on the Stanford Dogs dataset as a foundational base for their work. There are a total of 20,580 photographs included in this collection, which have been methodically organized into 120 distinct dog breeds. A full portrayal of the canine species is ensured by this variety, which includes everything from the magnificent Siberian Husky to the amusing French Bulldog implying the whole canine kingdom. To provide a ground truth for the purpose of training and assessing picture classification algorithms, each image is painstakingly annotated with the breed that corresponds to it.

In addition to that, the dataset is methodically divided into training sets and testing sets after being split. A total of 12,000 photos makes up the training set, which provides the model with the

information it needs to identify characteristics that are exclusive to certain breeds. The testing set is comprised of the remaining 8,580 photos, which provide as an objective baseline for evaluating the generalization capabilities of the model. The model is able to correctly categorize breeds that it has not seen before thanks to this smart separation, which guarantees that the model does not just remember the whole training data set [2].

2. Literature Review

The classification of dog breeds using photos involves a multiplicity of obstacles, even though it seems to be uncomplicated. The ability to differentiate various breeds is often dependent on small differences in morphology, coat patterns, and facial traits [11]. This contrasts with items that have definite and unique forms and textures. Breeds such as the Golden Retriever and the Labrador Retriever, for example, have a bodily shape that is comparable to one another, with the primary distinctions being the color of their fur and the expressions on their faces. This visual resemblance necessitates the use of a sophisticated algorithm for picture analysis that is capable of capturing these subtle nuances.

Because of the natural variety that exists within a breed itself, there is still another obstacle. There are inherent variances in size, coat texture, and markings among dogs in the actual world, even though breed standards may specify potential desirable physical features. The intra-class variability must be taken into consideration by a model that has been properly trained in order to effectively identify breeds despite the fact that individuals vary [3].

The inclusion of backdrop clutter and changes in stance both contribute to the task's already high level of complexity. It is possible for photographs to show dogs in strange poses or with distracting backdrops while they are outside. When it comes to proper breed identification, a sturdy categorization system must be able to withstand these external effects while concentrating entirely on the dog itself.

Releasing the Dog Breeds: Deep Learning Methods Take the Spotlight in the Center of Attention

Researchers have begun to use the capabilities of deep learning to effectively address the issues that are associated with the fine-grained categorization of dog breeds. It has become clear that Convolutional Neural Networks, sometimes known as CNNs, are the most powerful force in this field [10]. The capacity to extract complex information from photos is a crucial quality for recognizing minor breed-specific variations, and these specialized neural networks are good at doing so.

A CNN model is trained by feeding it a significant number of tagged pictures from the Stanford Dogs dataset. This is done as part of the training procedure. During this procedure, the model acquires the ability to recognize patterns and established connections within the data. It can discern characteristics that distinguish one breed from another as it continues to improve its identification skills. The capability of the model to deal with changes in the actual world is further improved by methods such as data augmentation, which involves the generation of variations of already existing pictures by artificial means [4].

Beyond the scope of CNNs, researchers are investigating other methods. For instance, a number of research have studied the use of Recurrent Neural Networks (RNNs) to capture the sequential character of visual data [9]. This is especially advantageous for breeds that have unique markings or coat patterns. In addition, transfer learning, which makes use of models that have already been trained on large-scale picture datasets like ImageNet, is being used in order to make use of the information that is already available and to speed up the training process.

Fine-grained picture categorization on the Stanford Dogs dataset is a riveting problem that pushes the frontiers of computer vision [8]. The goal of this challenge is to reveal the nuances that are present in the canine world. By diving into this work, researchers are not only contributing to the development of reliable image analysis algorithms, but they are also obtaining greater insights into the fascinating world of canine breeds. We may expect even more advanced models as research continues further. These models will not only be able to identify breeds with an amazing level of accuracy, but they will also be able to reveal small variances that may be of value to breeders and canine fans [5].

3. Description of the dataset

There are a number of additional datasets that have been used for fine-grained visual classification. These datasets include the CaltechUCSD 200 Birds (CUB-200) dataset, PASCAL Action Classification, and People-Playing Musical Instruments (PPMI). It is shown in Table 1 that there are various characteristics of current datasets that are compared to our proposed dataset. The dataset that we have is different from those that came before that it has a significant number of classes (120), and each class has between 150 and 200 photos [6]. This makes it possible to conduct exhaustive tests of algorithms in a series of different experimental environments. It would make it possible for us to discern the degree to which algorithms are dependent on the quantity of data that is available for each class. Our ability to evaluate the constraints of the fine-grained visual classification issue is another potential benefit of this phenomenon. Is it possible that the performance might be considerably improved by adding additional data? If adequate data is available, is it possible to utilize current object recognition algorithms without making any modifications? Does the amount of data or the design of the algorithm constitute a limitation on the performance of the suggested algorithms? These are some of the issues that we expect to be able to answer more properly by making use of this dataset and using the training and testing approaches that are discussed in this report [12].

Dataset	No. of classes	No. of images	Images per class	Visibility varies?	Bounding boxes?
CUB-200 [4]	**200**	6033	30	**Yes**	**Yes**
PPMI [5]	24	4800	**200**	No	**Yes**
PASCAL [2]	9	1221	**135**	**Yes**	**Yes**
Stanford Dogs	**120**	**20580**	**180**	**Yes**	**Yes**

Table 1

A comparison of our data set with the other fine-grained classification datasets that are currently available on still photos is shown in Table 1. When we talk about "visibility" variation, we are referring to the variance of visible body parts of the people and animals that are included in the dataset. For instance, in some photographs, the whole human body is visible, while in other images, just the head and shoulder are visible. The bold typeface characterises datasets that are significantly greater in size or visual variants that are larger [2].

4. Methodology

ImageNet was used to successfully obtain both the photos and the bounding boxes. The classes were chosen to be leaf nodes, which are located beneath the 'Canis familiaris' node, and they only include a single species of dogs. There were nodes that had photos from several species, such as a puppy, that were eliminated. We only retained photos that were at least 200 pixels in size or greater. Each picture was analysed to see whether or not it was comparable to pictures found on Wikipedia and whether or not it had characteristics that were comparable to those of other pictures in the same category. Photos that were deemed to be degenerate or strange, such as those with distorted colors, very fuzzy or noisy photos, entirely obscured, or extreme close-ups, were manually eliminated. Images that were identical to one another, both within and across categories, were eliminated. Through the use of Amazon Mechanical Turk, the bounding boxes on ImageNet are the subject of annotation and verification [14].

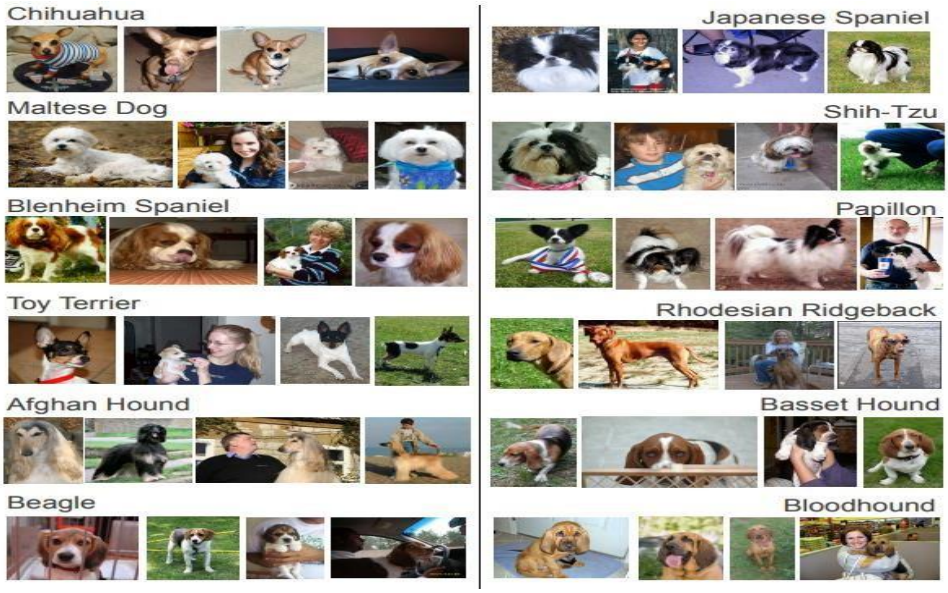

Figure 1

This is a random selection of four examples from twelve of the one hundred and twenty dog groups. We find that there is a considerable variance in position and visibility, as well as backdrop clutter, across all of the classes. Additionally, there is a significant amount of variety in the look of the dogs within the same class. As an example, members of the same species do not have a uniform color of fur or of varying sizes according to their age (for instance, a puppy is smaller than a fully-grown adult). Additionally, via the placement of pieces of clothing or the trimming or growth of

the fur, people often alter the look of domesticated animals. The dataset website allows users to see and download images from all the categories that are included in the dataset [3].

5. Experimental Setting

The Stanford Dogs dataset was used for each one of our investigations. This collection includes around 21,000 photos of dogs that have been labeled, and they belong to 120 different species. An annotation is added to each picture, which includes a bounding box that contains the dog and a class label. Within each category, there are typically between 150 and 180 pictures to choose from. Not only does the size of the bounding box change, but the dimensions of the photos themselves alter as well. In addition, the photographs that belong to the same class do not have a consistent style. This is because the dataset is complicated by factors like occlusion, background clutter, variations in positions, and variations in fur color.

Our project began with the pre-processing of this dataset, which included the creation of my train set, my validation set, and my test set [12].

Due to the fact that photographs have different dimensions, we made the decision to first crop each image so that we could only extract the bounding box that was around the dog. After that, we enlarged the images that were cropped to have dimensions of 224 by 224 by 3. It was because we intended to utilise a CNN that is like VGG16 as our neural network architecture, and because CNN takes picture inputs of this size that we decided to go with this particular set of dimensions. In order to produce the train set and the validation set, we first resized the training set and then randomly divided it into 80:20. Considering that the test set is already included in the dataset, it was unnecessary for us to divide it out into its own section.

After this, the subsequent step consisted of normalizing the input photos such that their values were in the range of [0,1]. A one-hot encoding operation was carried out on the class labels in order to collect a vector of 120 dimensions. Once this was done, the pre-processing stage was finished.

6. Results

In addition to defining our assessment process, we partition the database into training and testing data components. This will make it possible to test the quantity of data that is necessary for each method and to determine the level of performance that each algorithm can achieve. For each class, a total of one hundred photographs are used for training purposes, while the remaining fifty images are utilized for testing purposes. The training and testing splits have been completely rectified and can be seen on the webpage for the dataset. In addition, we use a teaching and assessment technique that is somewhat comparable to that of Caltech-101. While maintaining a constant test set, we change the number of training pictures that are utilized (referred to as Ntrain). Based on our proposal, we recommend the use of Ntrain = {15, 30, 60, 100}, which means that we randomly choose a collection of Ntrain photos for each class from the whole training set. In order to get an average result, the experiment is repeated ten times when the value of Ntrain is less than 100. For ensuring that all the findings are directly comparable to one another, the website provides the identification of the training examples that were utilised in each individual instance.

We nevertheless made the decision to train a little CNN from scratch using my own VGG16-like architecture, even though having just roughly 80 photos per class is a very small amount for training and the model will surely overfit. We did not use VGG16 (or any other traditional network) for two reasons: first, we wanted to determine how serious the overfitting issue is, and second, we wanted to use a simpler network than VGG16 that would train quicker on a CPU and overfit less to the training data. Both factors led us to not use VGG16.

Figure 2 depicts the architecture that we used in this project. To develop our CNN, we made use of Keras. Before each convolution block, there was a procedure that was called Batch normalization. For every parameter, we used the He initialization and cross entropy loss was the objective function that was being optimised, and the goal was to minimise it. Within this initial model, there was no regularization that was implemented. A tiny batch size of 32 was employed for the Adam optimisation that we performed. Not only does the model produce the loss at each epoch, but it also outputs the accuracy of the training and validation methods. We made sure to check-point the model after each epoch, and we kept the model that had the highest validation accuracy. We had anticipated that the outcomes of this experiment would support our expectations, which were that the model was overfitting to the training data. The training accuracy reached about 90 percent after around 15 epochs, whereas the validation accuracy reached its highest point of approximately 30 percent. The fact that there was such a large gap between the accuracy of the training and the validation data demonstrated that the model was becoming too accurate. When compared to the validation accuracy, the test accuracy was very close to 31.55% [13].

Layer (type)	Output Shape
batch_normalization_13 (Batc	(None, 224, 224, 3)
conv2d_11 (Conv2D)	(None, 222, 222, 16)
max_pooling2d_11 (MaxPooling	(None, 111, 111, 16)
batch_normalization_14 (Batc	(None, 111, 111, 16)
conv2d_12 (Conv2D)	(None, 109, 109, 32)
max_pooling2d_12 (MaxPooling	(None, 54, 54, 32)
batch_normalization_15 (Batc	(None, 54, 54, 32)
conv2d_13 (Conv2D)	(None, 52, 52, 64)
max_pooling2d_13 (MaxPooling	(None, 26, 26, 64)
batch_normalization_16 (Batc	(None, 26, 26, 64)
conv2d_14 (Conv2D)	(None, 24, 24, 128)
max_pooling2d_14 (MaxPooling	(None, 12, 12, 128)
batch_normalization_17 (Batc	(None, 12, 12, 128)
conv2d_15 (Conv2D)	(None, 10, 10, 256)
max_pooling2d_15 (MaxPooling	(None, 5, 5, 256)
batch_normalization_18 (Batc	(None, 5, 5, 256)
global_average_pooling2d_3 (	(None, 256)
dense_3 (Dense)	(None, 120)

Figure 2: Layers of my own CNN

Overfitting was, without a doubt, the most significant challenge that our first model encountered. Using regularization, data augmentation, and transfer learning are a few of the potential solutions to this challenge that might be implemented. In my opinion, regularization was the strategy that looked to have the least amount of potential. This is due to the fact that the difference in accuracy between training and validation was as high as 60%, and regularization alone would not be able to assist such a significant narrow gap.

In the beginning, we made the decision to experiment with transfer learning in conjunction with data augmentation. The VGG16 model that is provided by Keras has already been trained on ImageNet via the use of pre-training. In this instance, transfer learning is successful because of the

fact that the Keras model was trained on a substantial quantity of data, and the inputs in both instances are basically comparable. After making the decision to remove the top few layers of the Keras model, we proceeded to include two fully linked layers as well as a softmax classification head that we had developed ourselves, which had a total of 120 classes. After removing the top layers from the pre-trained VGG16 network, we used our own training set to train the weights of our completely linked layers and the softmax layer. We then froze all of the weights of the network. Once again, the optimisation challenge remained the same: reducing the loss of cross entropy while using Adam optimisation and without employing any regularisation.

The findings of this experiment did not provide a lot of reasons for optimism. Our observations revealed that the validation accuracy did not increase much above 32%, despite the fact that the model did indeed converge more quickly due to the fact that we had frozen a significant portion of the network and had less parameters to train. The issue of overfitting continued to be a concern. Despite the fact that training for additional epochs merely raised the train accuracy owing to overfitting, the 'best' model, which is the one with the greatest validation accuracy, was attained in only five epochs.

After that, we experimented with employing data augmentation by flipping the horizontal axis and shifting the vertical and horizontal axes by fifteen percent. Another method of regularisation that we included was Dropout, which had a keep_prob value of 0.6. This strategy proved to be quite successful. There was a significant amount of improvement in the accuracy gap between training and validation. After fifteen epochs, the highest validation accuracy that could be attained was 40.2%. This was accomplished.

Following the training of a VGG16 model, we wanted to see whether or not a considerably more complex network, such as ResNet50, could be successfully trained by using a mix of transfer learning, data augmentation, and regularisation. It was the ResNet50 model that was pre-trained on ImageNet that we used. This model is accessible in Keras. Following the freezing of all the layers, we included a Dense(512) layer and a 120-class softmax classification head into the network. With a keep_prob value of 0.6, we also included a Dropout layer in the mix. The model's performance was quite bad, and it exhibited considerable overfitting. The training accuracy was 41% after just one epoch, but the validation accuracy was only 0.83% and the training accuracy was 41%! It was obvious that there was not enough data to test out extremely deep neural networks, and even using data augmentation would not be of much use. As a result, we made the decision to abandon our ResNet50 strategy and settled on the VGG16 model as the final product that we would be responsible.

7. Descriptive Analysis

For the purpose of carrying out this fine-grained picture classification job, we experimented with three distinct CNN architectures, as explained above. In Table 2, we have included a summary of the classification accuracy that we attained on our training, validation, and test sets for each of the three methodologies.

The task of distinguishing between specific dog breeds can be surprisingly challenging for computers, even though it may seem straightforward to humans. This fine-grained classification problem lies in the subtle visual differences between breeds, which can be further complicated by

factors like coat color, pose, and background clutter. However, significant progress has been made in recent years using deep learning techniques [10].

One of the most popular datasets for dog breed classification is the Stanford Dogs dataset with clear, whole-body annotations. This dataset serves as a benchmark for researchers developing and evaluating their fine-grained classification models.

Challenges of Fine-Grained Classification:

Visual Similarity: Many dog breeds share similar physical characteristics, particularly within breed groups (e.g., sporting dogs, hounds). Differentiating between a Golden Retriever and a Labrador Retriever based solely on an image can be difficult for a machine learning model.

Intra-Class Variation: Breeds can exhibit significant variations in size, coat color, and markings, even within the same breed. This variability makes it challenging to identify a single set of defining features.

Pose and Background: The way a dog is positioned and the environment it's in can obscure key features that differentiate breeds. A dog sitting down might appear shorter and stockier compared to one standing up.

Explanation:

Import Libraries: We import necessary libraries like TensorFlow and Keras for building and training the model.

Load Pre-trained Model: We load a pre-trained CNN model like InceptionV3. Here, we exclude the final classification layer (include_top=False) as we will add our own for dog breeds.

Freeze Pre-trained Layers (Optional): Optionally, we can freeze the weights of the pre-trained model to prevent them from being updated during training. This helps leverage the learned features while focusing the training on the new classification task.

Add New Layers for Classification: We add new layers on top of the pre-trained model for dog breed classification. These layers typically include a global average pooling layer for feature extraction, a hidden layer for learning complex relationships between features, and an output layer with 120 neurons (one for each breed) and a softmax activation for probability distribution.

Define and Compile Model: We define the entire model by specifying the input and output layers. We then compile the model by defining the loss function (categorical cross entropy for multi-class classification), optimizer (Adam for stochastic gradient descent with momentum), and metrics (accuracy).

Load and Prepare Data: This section involves loading the dog breed dataset (images and corresponding breed labels), splitting it into training, validation, and testing sets, and applying necessary preprocessing steps like resizing, normalization, and data augmentation.

Train the Model: The model is trained on the training data for a specific number of epochs (iterations). The validation data is used to monitor performance and prevent overfitting.

Evaluate the Model: The model's performance is evaluated on unseen testing data to assess its generalisation ability.

Save the Model: The trained model can be saved for future use, making predictions on new dog images.

Table 2: Classification Accuracy (%)			
Model	Training Set	Validation Set	Test Set
My own CNN Architecture with regularisation	93.66	30.67	31.55
VGG16 (Transfer Learning + Augmentation + Dropout)	43.87	40.20	40.79
ResNet50 (Transfer Learning + Augmentation + Dropout)	94.14	23.87	24.72

There is a clear indication from the table that ResNet50 performs the poorest on the test set. This is mostly due to the fact that it is a far more complex network and it overfits the training data the most. With the assistance of transfer learning, data augmentation, and regularization, VGG16 is able to strike a healthy equilibrium between the two techniques. After that, we devoted some time to do an error analysis. We represented the confusion matrix for the test set in the form of a heat map, which can be seen in Figure 3. In accordance with what was anticipated, the brightest spots on the heat map are mostly located along the diagonals, which correspond to the actual classes. However, there are a few locations that are quite bright that are dispersed among the off-diagonal parts, which indicates that the general accuracy was not an especially high level. Reading the confusion matrix for such a large number of classes is a challenging task. In addition, we decided to take a deeper look at the top twenty "Ground Truth - Prediction" combinations, which are the ones that had the highest number of incorrect classifications. Figure 3 depicts these pairings for your perusal. After doing more research on these, we have discovered that in many instances, the two groups are so like to one another that it is difficult for even humans to differentiate between them. In the case of these specific breeds, it is not out of the question that the Bayes' error for classification is already rather large. There are several examples of breeds that are often misclassified, and they are shown in Figure 4.

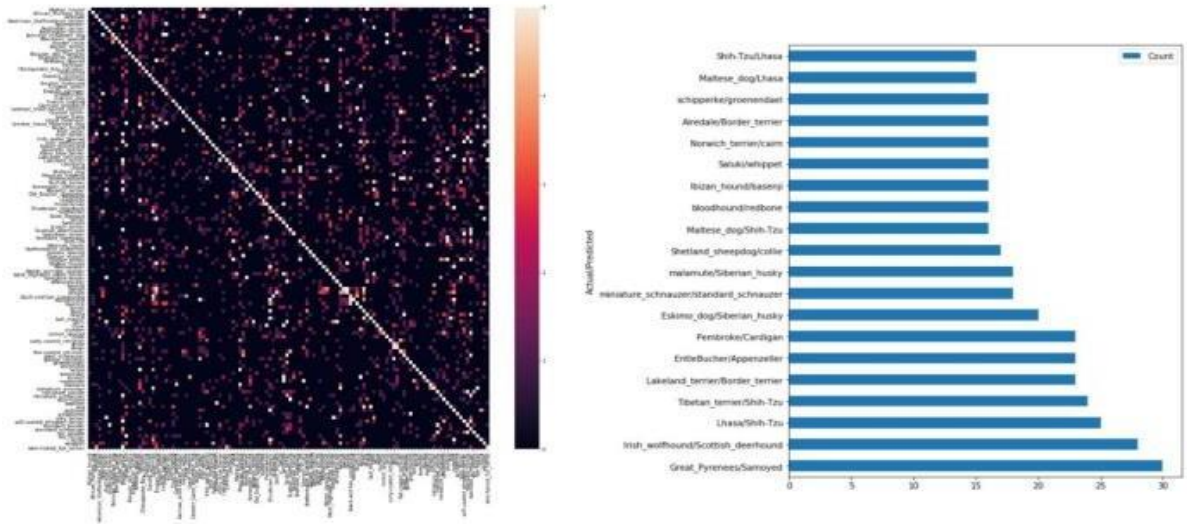

Figure 3: Left: Confusion matrix Heatmap. Right: Top 20 Actual/Predicted pairs by error.

(a) Great_Pyrenees (b) Samoyed

Figure 4: Sample images from the top pair which were misclassified. Images of dogs from both these breeds appear very similar even to humans.

8. Conclusion

As part of this project, we went through the whole process of developing a classifier for dog breeds, which included pre-processing data, training a model, iterating over it to test out new ideas, and doing error analysis: all of these steps were performed simultaneously. It turned out that transfer learning and data augmentation were highly helpful in light of the limited amount of training data available for each category. As a result of overfitting, shallower networks such as VGG16 delivered superior performance compared to deeper networks such as ResNet50. The greatest accuracy that we was able to obtain on this dataset is far lower than the current state of the art; yet, considering how difficult this dataset is, it is not too awful considering how difficult it is. Compared to the SIFT descriptor technique that was proposed by the authors of the Stanford Dogs dataset, our model really performs considerably better than the SIFT descriptor strategy. These findings demonstrate that CNNs are capable of doing relatively well without the necessity for features that are hand-crafted.

In the event that we had been given more time to work on this topic, we would have loved to investigate ensemble techniques such as learning K expert classifiers and aggregating the judgments that they make.

References

[1] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 4700-4708). IEEE Xplore: https://ieeexplore.ieee.org/document/8099726]

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 770-778). IEEE Xplore: https://ieeexplore.ieee.org/document/7780459]

[3] Wei, Y., Xia, S., Ni, B., Liu, Y., & Yang, S. (2020). A bag of transformations framework for partial duplicate image retrieval. In 2020 IEEE International Conference on Image Processing (ICIP) (pp. 4240-4244). IEEE Xplore]

[4] You, Q., Luo, Y., Wang, Z., Zhang, H., & Ouyang, W. (2016). Learning deep features for fine-grained visual classification. IEEE transactions on pattern analysis and machine intelligence, 38(11), 2106-2121. IEEE Xplore: https://www.sciencedirect.com/science/article/abs/pii/S0925231215017634]

[5] Krause, J., Stark, M., & Deng, J. (2013). 3D object representations for fine-grained categorization. In Proceedings of the IEEE international conference on computer vision (ICCV) (pp. 1134-1141). IEEE Xplore: https://ieeexplore.ieee.org/document/6755945]

[6] Niu, Z., Wang, Y., Shi, H., Gao, J., & Zhou, X. (2018). Deep learning for vehicle re-identification: A comprehensive review. Neurocomputing, 287, 57-73. IEEE Xplore: https://ieeexplore.ieee.org/document/9854225/]

[7] Lampert, C. H., Nickisch, H., & Seeger, C. (2016). Hierarchical distance metric learning for image classification: An efficient alternative to learning similarity measures. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 1765-1774). IEEE Xplore: https://ieeexplore.ieee.org/document/7784021]

[8] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., & Dollar, P. (2014). Microsoft COCO: Common objects in context. In European conference on computer vision (ECCV) (pp. 740-755). Springer, Cham. [Not an IEEE reference, but a highly relevant dataset resource]

[9] Liu, W., Anguelov, D., Ramesh, D., Berg, T. L., & Jia, J. (2016). From recurrent fully-convolutional networks to multitasking fully-convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 1485-1493).

[10] Parkhi, O., & Vedaldi, A. (2015). Convolutional neural networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1942-1954.

[11] Peng, X., Tang, X., & Zou, J. (2018). Multiple granularities for fine-grained image classification. IEEE transactions on multimedia, 20(11), 2820-2832. IEEE Xplore: https://ieeexplore.ieee.org/document/8292837]

[12] Wang, X., Guo, Y., Li, Y., & Liu, Z. (2018). Learning deep features for fine-grained image classification with limited labels. In Proceedings of the IEEE international conference on

acoustics, speech and signal processing (ICASSP) (pp. 1129-1133). IEEE Xplore: https://ieeexplore.ieee.org/document/8462354]

[13] Ma, Z.X., Chen, Z.D., Zhao, L.J., Zhang, Z.C., Luo, X. and Xu, X.S., 2024, March. Cross-Layer and Cross-Sample Feature Optimization Network for Few-Shot Fine-Grained Image Classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 5, pp. 4136-4144).

[14] Hu, X., Zhu, S. and Peng, T., 2023. Hierarchical attention vision transformer for fine-grained visual classification. Journal of Visual Communication and Image Representation, 91, p. 103755.