Dog Classification into 120 Breeds

Hang Zhang Haonan Song Leyan Zhu Zan Deng

Abstract— **Dog classification is a challenging problem as certain dog breeds have near identical features or differ in color and age. Our project, based on the Stanford dog dataset, aims to classify the dogs into different breeds. We preprocess the dataset and use different models including Alexnet, VGG16, and Inception-v3 to train the data.**

I. INTRODUCTION

Dog classification is one of the fine-grained image classification problems. Fine-grained image classification aims to recognize the subcategories of a certain category. It is more difficult than the general object classification since the little difference between subcategories and the large difference within the subcategories. For the dog classification problem, dogs of different breeds could share almost all the same visible features, like the same color, same facial characteristics, but they are still classified as different breeds. Dogs of the same breeds, however, could differ significantly depending on the age, position, size and even color.

The solution of the problem is applicable to other fine-grained classification problems, such as deciding plant species from leaf and species of bird. It helps to provide the biodiversity information for biologists, replacing the traditional expensive human annotations. Face recognition is also an important application of fine-grained classification, which is widely used in access control and commercial identification.

The input to our model is an image. We then use a neural network to output a predicted breed.

II. RELATED WORK

A. Traditional image recognition methods

There is a fair amount of research that has been done in the fine-grained image classification problem. Traditional image recognition methods are used, one of which is scale-invariant feature transform(SIFT). The scale-invariant feature transform(SIFT) is a feature detection algorithm to detect and describe local features in images. It can be decomposed into four steps: feature point detection, feature point localization, orientation assignment and feature descriptor generation.

Aditya Khosla et. al[1] applys grayscale SIFT descriptors for classification on Stanford dogs and uses the result as the baseline for the dataset.

B. General CNN

Recently, with the advances of deep learning, deep convolutional neural networks have provided a new opportunity for fine-grained image recognition. Numerous deep convolutional feature-based algorithms have been proposed, which have advanced the development of fine-grained image research.

LeCun et al.[2] first introduced CNN into the area of recognition tasks and it is very popular ever since. Many state-of-the-art CNNs can be applied in fine-grained image classification problem, including Alexnet, VGGnet and GoogLeNet.

C. Part localization

Liu et. al[3] demonstrates that if the features used for classification are localized at object parts, the accuracy can be improved greatly. A sliding window detector is used to locate dog faces, and Then the eyes and nose are detected. With this small set of face, greyscale SIFT descriptors around the keypoints are used as features by an SVM classifier.

Some neural networks based on part localization are developed. Part-based R-CNN[4] extends R-CNN, learning the part detectors by leveraging deep convolutional features computed on bottom-up region proposals. Pose normalized nets[5], Part-stack CNN[6] are also widely used.

III. DATASET AND FEATURES

A. Dataset

The Stanford dogs dataset is introduced by Aditya Khosla et al.[] in 2011. It contains two part. The first part is images, which includes 20580 images of dogs belonging to 120 species, divided into 120 folders. The second part is annotation which contains the basic information of the corresponding image. Also, it has information about the bounding box and class label, which can help us preprocessing of the dataset. To train these models, we divide the dataset into three parts - training, validation and test, which contains 80%, 10% and 10% of the entire dataset respectively.

B. Pre-processing

To help us get a better result, we use several ways to preprocessing these images including localization, data augmentation and random flips. Also, to fit different models we used, we normalize these images to different sizes.

Firstly, to localize the dogs in these images, we crop these images according to the bounding box information in the annotation, so that dogs can stand out in the image and can be easier to capture.

Then we use data augmentation, including random rotation, random crops and random flips to increase the variability of training images. But it also introduced another problem that the size of these images become different. Thus, the images must be resized to get a uniformed shape. According to the ImageNet standards, we normalize these images into

Fig. 1. Sample image

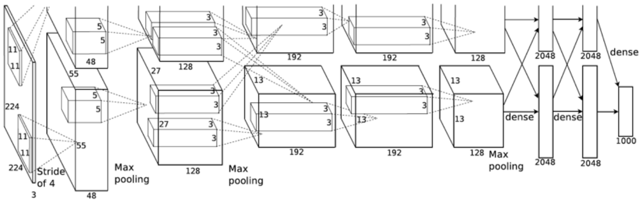
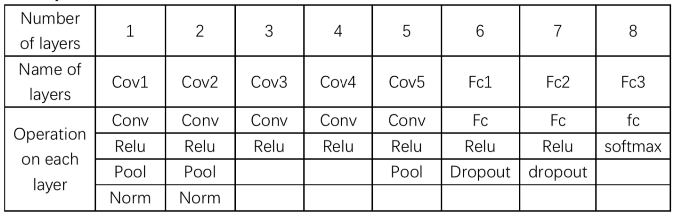


Fig. 3. Alexnet



Number of layers	1	2	3	4	5	6	7	8
Name of layers	Cov1	Cov2	Cov3	Cov4	Cov5	Fc1	Fc2	Fc3
Operation on each layer	Conv	Conv	Conv	Conv	Conv	Fc	Fc	fc
	Relu	Relu	Relu	Relu	Relu	Relu	Relu	softmax
	Pool	Pool			Pool	Dropout	dropout	
	Norm	Norm						

Fig. 4. Operation of each layer

different shape depending on different models. For Alexnet, the input image size is $227 \times 227 \times 3$. For VGG, the input image size is $224 \times 224 \times 3$. As shown in figure 2, we can see the difference between images with and without pre-processing.

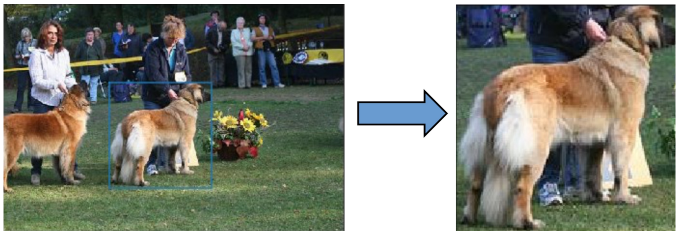


Fig. 2. Sample image

IV. Methods

We applied three different models, namely Alexnet, VGG16, and Inception-v3, in this dog breeds classification task. In this section, we will discuss the structure and learning algorithm for each model.

A. Alexnet

Alexnet[1] contains eight layers. The first five layers are convolution layers and then follows three fully connected layers. The whole structure of this model is shown in figure 3.

In each layer, this model does different operations to the input data. The operations used in each layer can be concluded in figure 4.

For the first seven layers, the model uses ReLu as activation function.

In this formula, x denotes the input value and y denotes the output value. Different from first seven layers, the last layer applies the softmax function. The standard softmax function is defined as follow.

$$\sigma(Z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

The softmax function takes K inputs and normalized these inputs. In other words, by applying softmax function to a K value vector, all element in the output vector will be in the interval (0,1) and the sum of all elements will be 1.

The syntax Norm in the form means local response normalization. The response-normalized activity $b^i_{x,y}$ can be computed from the formula as follow.

$$b^i_{x,y} = a^i_{x,y}/(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^i_{x,y})^2)^\beta$$

In this formula, $a^i_{x,y}$ denotes the activity of a neuron, which computed by applying kernel i at position (x,y). k, α , n, β are hyperparameters and they are constant. In Alexnet, k=2, α=10^{-4}, n=5, β=0.74.

B. VGG16

The structure of VGG16[2] is as follow. This model contains 13 convolution layers, 5 max pooling layers and 3 fully connected layers. ReLu function is act as activation function for each convolution layer and fully connected layer, except the last layer. The last layer in the model is a fully connected layer with softmax function. The detail explanation of ReLu function, softmax function can be found in Alexnet model mentioned above.

C. Inception-v3

The most important feature for Inception[3] model is inception module. It takes the same input and applies filters of different size in parallel and then concatenate all the outputs into one output. The main difference between Inception-v3 and Inception-v2 is that v3 divides single convolution kernels into several kernels. For instance, a 33 convolution kernel has

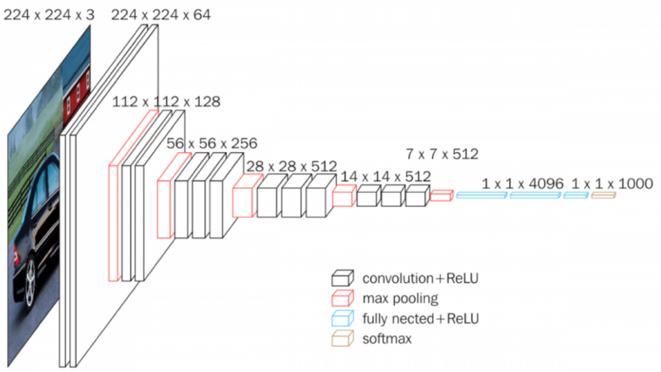


Fig. 5. VGG16

been divided into two kernels with the size of 31 and 1 3. By applying 1 1 convolution for dimensional reduction, this model can have a deeper structure but fewer parameters at the same time.

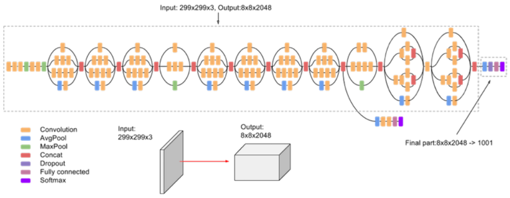


Fig. 6. Inception-v3

V. Experiments and Results

In the experiment, we construct three different models and use the training data to train them. Due to the fact that it will take a lot of time to train models with whole training dataset, we first adopt the data for thirty kinds of dog breeds to find the best model for this classification task. We use loss and accuracy to evaluate the performance of models. We record these evaluating results at each epoch and plot the result for each model. The plots are shown as follow.

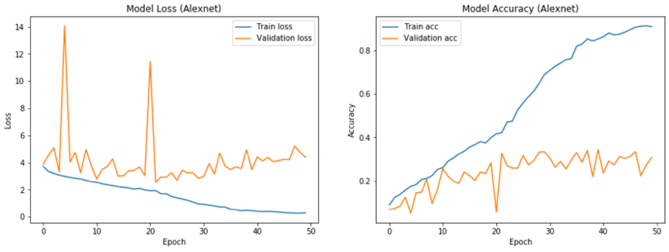


Fig. 7. Loss and Accuracy for Alexnet

From the plots we can find that after 50 training epochs, the predicting result from Alexnet model has reached 90 percent on training data while has only 30 percent on validation data. For VGG model, the predicting accuracy is over 50 percent on training data and 60 percent on validation data. For Inception-v3 model, the predicting accuracy is over 85 percent on training data and almost at 90 percent on validation data. The accuracy on validation data is higher than the accuracy on training data, which indicates that the dropout technique we adopted in VGG16 and Inception-v3 models have a positive influence on these models.

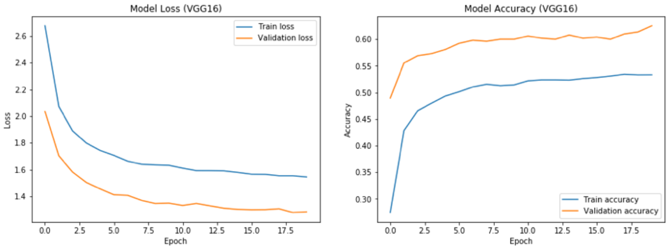


Fig. 8. Loss and Accuracy for VGG16

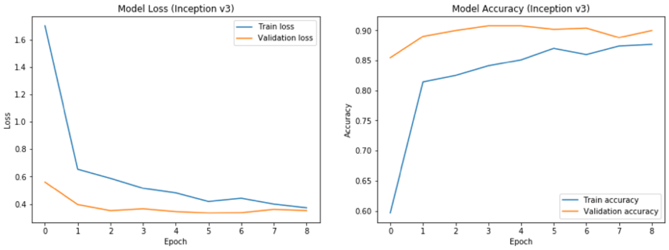


Fig. 9. Loss and Accuracy for Inception-v3

In the next step, we applied the trained models to testing data. The results for the three models are shown in the following form.

Models	Alexnet	VGG16	Inception-v3
Accuracy on training dataset	93.25%	052.39%	93.22%
Accuracy on testing dataset	29.65%	62.09%	90.69%

TABLE I
Results

It is obviously that Inception-v3 performs much better than the other two models according to the accuracy on testing dataset. Thus, we then use Inception-v3 model to classify 120 classes of dogs.

The loss and accuracy on training and validation data during the training process is shown as follow.

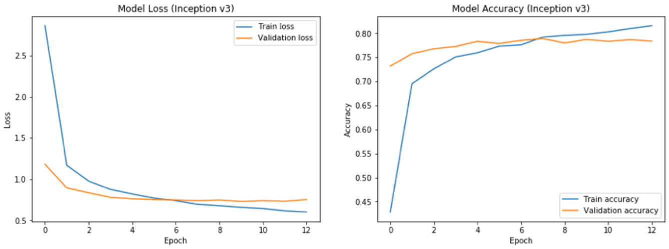


Fig. 10. Loss and Accuracy for Inception-v3 on 120 classes

The prediction accuracy on training data is over 80 percent. As the accuracy on validation data is almost stop increasing, thus we stop the training process. Then the trained model has been applied to testing data. The model has achieved an accuracy of 79.25 percent on testing data.

VI. Conclusion and Future Works

Having conducted the experiment on classifying a subset of the original dataset, which is 30 classes in 120 classes, we come to the conclusion that compared to Alexnet and VGG16 model, Inception-v3 model has a much better performance in this classification task. Then we use the training data to train the inception-v3 model and apply the trained model to classify 120 classes of dogs. The prediction accuracy is 79.25 percent. It is no doubt that this result is excellent and the model works successfully on this task.

However, there is still room to improve performance. Different models are suitable for a different task, we only tried three models. From the image below, there are many other advanced models worth trying, although some of them might not be suitable for this task.

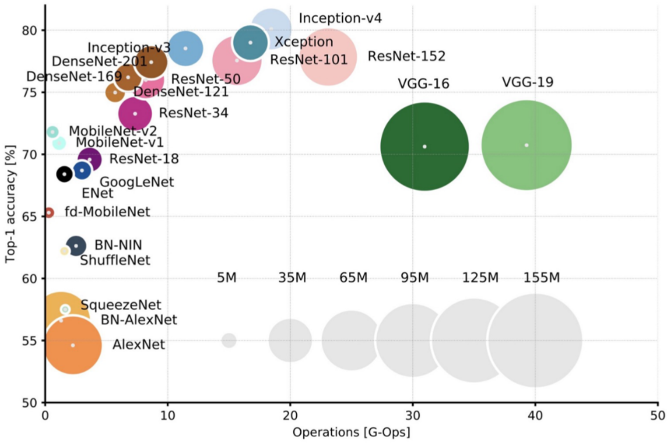


Fig. 11. Deep learning models

Contribution

Hang Zhang preprocessed the data. Zan Deng worked on Alexnet. Haonan Song worked on the VGG16. Leyan Zhu worked on Inception-v3. The whole write the final report together.

References

[1] Khosla, A., Jayadevaprakash, N., Yao, B., Li, F. F. (2011, June). Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)* (Vol. 2, No. 1).

[2] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.

[3] Liu, J., Kanazawa, A., Jacobs, D., Belhumeur, P. (2012, October). Dog breed classification using part localization. In European conference on computer vision (pp. 172-185). Springer, Berlin, Heidelberg.

[4] Zhang, N., Donahue, J., Girshick, R., Darrell, T. (2014, September). Part-based R-CNNs for fine-grained category detection. In European conference on computer vision (pp. 834-849). Springer, Cham.

[5] Branson, S., Van Horn, G., Belongie, S., Perona, P. (2014). Bird species categorization using pose normalized deep convolutional nets. arXiv preprint arXiv:1406.2952.

[6] Huang, S., Xu, Z., Tao, D., Zhang, Y. (2016). Part-stacked cnn for fine-grained visual categorization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1173-1182).

[7] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[8] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[11] https://www.kaggle.com/twhitehurst3/stanford-dogs-keras-vgg16

[12] https://www.kaggle.com/gabrielloye/dogs-inception-pytorch-implementation

[13] https://www.kaggle.com/msripooja/dog-images-classification-using-keras-alexnet