

Critique of **group 25** presentation - **Dog Classification via Deep Learning**
Critiques by **group 84**.

The presentation elaboratively introduced the background, motivation, and the approach they employed to address the fine-grained image classification problem. They mainly compared CNN models such as AlexNet, GoogleNet, ResNet, VGG, and then conducted experiments accordingly to seek a better solution.

Possible improvements:

- Other approaches such as SIFT, Part-detection-and-alignment-based approach are mentioned in the background section. However, there is neither experiment on these methods, nor statistical results showing they're inferior to CNN models. It would be better if related information could be included to help better demonstrate the superiority of state-of-the-art CNN models.

Thanks for your advice, in this project, we mainly focus on the deep learning methods on this classification tasks. More experiments towards traditional methods will be done in the future.

- There're only plots of test accuracy shown in the presentation. However, to provide a better analysis that is convincing, plots of training and validation accuracy/loss should be also illustrated. These data could help detect problems of the model, such as overfitting, underfitting, etc.

Since there is page limitation in slides, we did not put loss curves on slides. But we do discuss loss curves in our final report. Thank you for reminding us.

- The code demo part merely shows the training process. The plotting part could also be demonstrated since they don't require much time to complete. And procedures such as visualization of the dataset, visualization of prediction(through a test example) could also be considered adding to the demo part since they're helpful with regards to an end-to-end framework.

Since the training process could be very long even for fine-tuning, we just showed training for one epoch. Thank you for the advice. We will take it into account next time.

Critique of **group 25** presentation - **Dog Classification via Deep Learning**

Critiques by **group 48**.

Overall we thought you did a good job. We liked that you showed sample images from your dataset and discussed the difference between fine-grained and generic image recognition. We liked that you made observations on the dog images, such as various poses, but are curious as to how your model will differentiate between different poses. Also, we would have liked to know how your model handles when humans are also in the images. We would have appreciated more specifics on your dataset, such as the number of images per class and what train/test ratio you used.

We feel like you could have discussed your features more. In your literature survey, you mentioned SIFT for feature detection, and later on, mentioned that the CNNs automatically do the feature extraction for you. What feature extraction is it doing? We could have used a lot more information on this because we are not sure what your model is being trained on. Is it colors? Shapes? Texture? We could have used more insight on that.

[For classification tasks, CNNs extract pixel-wise information.](#)

We liked that you discussed the preprocessing on the data, and appreciated that you mentioned the specific order in which you did things. We think that you could have explained your pre-processing more, explaining why you did each of these steps, and why in this specific order. We think you did a good job explaining your CNN models' layers and discussed each model in depth. It was very useful for us to see the models and layers. We also thought it was insightful that you mentioned why you used these models specifically. It gave us a deeper understanding of the problem you are trying to solve.

[Thank you for your praise. We really appreciate it.](#)

When you discussed your results, we thought it would have been more clear if you displayed your results in a table. Based on the graphs, we can't really tell what your accuracy is, for each model. The way your graphs are laid out made it hard for us to see which models worked better than others. While you explicitly wrote which model had the highest accuracy, and what that accuracy was, we are not sure why you got those results. We would have liked some deeper comparisons on the different models. You also mentioned that your results were for models trained from scratch. What does that mean? Did you play around with a pre-trained model as well? Later on, you showed results based on fine-tuning. What does that mean? We think that more explanation was needed here.

[Thank you for your advice on visualization. We will consider displaying it that way next time. As for the meaning of fine-tuning, the Professor has mentioned this in class. So I would suggest watching the course video if you want to know more about this.](#)

We thought you gave a good walkthrough of your code. We thought it was useful that you went through it cell by cell and gave specific lines of code throughout your explanation. We are curious as to what scalar you used, as you briefly mentioned that, and what else is in your utility function.

For the scalar, we added Top1, Top5, and losses for both training and validating process. In the utility function, we defined several classes and functions.

adjust_learning_rate controls how the learning rate is changing.

accuracy computes the accuracy over the k top predictions for the specified value of k.

save_checkpoint saves the checkpoint in the given location.

AverageMeter and ProgressMeter Class are used to make benefits for saving more tensor's information' in the training/validation process, such as tensor's overall sum and display.

Critique of **group 25** presentation – **Dog Classification via Deep Learning**

Critiques by **group 41**.

In this project, the authors propose a neural network model based from scratch to transfer learning to extract features and classify the breed of dogs. The authors sampled the dataset from Stanford Dog Dataset. Then, they trained the AlexNet, GoogLeNet, ResNet18, and VGG16 from scratch, and compared them with the pre-trained weights of ImageNet by various learning rates. They showed the best is 95.85% of ResNet18 in 100 epochs of scratch training models, and 94.25% of VGG16 in 30 epochs.

This project comprehensively approaches to the public dataset using SOTA NN models, from scratch to pre-trained models with various learning rate.

Questions/Possible improvements:

- What is the motivation of this project? I saw the background in the slide ('challenge') but it seems weak to appeal to the audience. The strong motivation would be more helpful to understand why you choose this and what the hurdle you overcome.

Different from the generic image recognition task which aims to distinguish between categories, dog breeds classification is difficult because there are hundreds of dog breeds while many of them appear to be highly similar to each other.

- In Models slides, there is a lack of explanation for the models. The comment on the video seems fine, but there is no word in the slide to see. Also, why did you draw VGG16 without the batch normalization? I saw the plot of both models in the result section.

Due to the slide limitation, the models' details were discussed in the video. More information has been delivered in the project paper.

- Is there any specific reason you chose the transfer learning method? How's the performance differ from the Vanilla CNN models?

We want to find out whether the large-scale single-label classification problem will help with the fine-grained classification problem. The result has been discussed in the project paper.

- In Result section, y-axis scales are not the same, and the plot with grid lines would be more helpful. Also, the font sizes are different from each other.

- In the last Result slide, the conclusion is 'cost-effectiveness', so you would be better to write down that keyword in the slide.

- In the literature survey slides, the numerical index or evaluation criteria will be more helpful to understand. If you don't do the feature extraction, and there is not much to explain, then it would be good to skip or move and merge to the previous slide.

- Except for the learning rate control, how about using the schedule learning method for future work? - Typo in Future Work slide, 'epoches' should be replaced by 'epochs.'

Thank you for reminding us. Next time we will pay more attention to these.

Dog Classification via Deep Learning

Yudian Li Shanshan Xiao
Computer Science and Engineering
UCSD
A53313287 A53302520

Pu Cheng
Electrical and Computer Engineering
UCSD
A53306940

Abstract

Dog breeding classification is a challenging task since dogs with different breeding may have little visual difference. In order to achieve better result comparing to baseline method on this task, we reimplemented several well-known CNN networks and use Stanford Dog Dataset which is a subset from ImageNet used for the task of fine-grained image classification to verify our approach. Furthermore, we studied the transfer learning ability from a broad, general classification problem in a fine-grained classification problem.

1. Introduction

Dog breeding classification is a challenging fine-grained classification task. Different from the generic image recognition task which aims to distinguish between categories, fine-grained image classification is used to tell which subcategory the object belongs to. Usually, fine-grained image classification is more complex than a generic-image recognition task. Because the former has small inter-class variance and large intra-class variance[9]. In our task, there are hundreds of dog breeds while many of them appear to have little visual difference. And dogs belonging to the same breed may have various colors and sizes(puppy or fully grown).

There are many real-world applications of fine-grained image classification problems. For example, this can be used in biodiversity protection and product recognition in online retailer. Illegal fishing practice are threatening the sea ecosystem. This technique might help detect the species in danger and protect them, and thus maintain the balance of the ecosystem. Another example is in the retailer business. Many items may share similar outlooks. How to automatically classify them is a very important challenge. If we are able to have a good result in this task, we can further gain a more concise workflow on how to solve other fine-grained classification problems.

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

2. Related Work

2.1. Traditional Machine Learning Methods

Many traditional machine learning methods can be used to solve this problem. And basically it contains two steps: feature engineering and a prediction model. One important feature extraction method in computer vision is called scale invariant feature transform(SIFT). It is used to detect and describe local features in an image. This method has several steps: detecting points of interest, key point localization, orientation assignment, and generating key point descriptor. A gray scale descriptor was used as baseline for the Stanford Dog Dataset[4].

2.2. Part Detection And Alignment Based Approach

Semantic part localization can help fine-grained image classification by extracting the subtle visual difference associated with different objects. Liu et al.[7] built an exemplar-based geometric and appearance models of dog breeds and their face parts to find accurate dog parts such as eyes and faces. Gavves et al.[2] segment images and align the segments in a unsupervised way. These segments are then used to transfer part annotations from training images to test images.

The part detector widely used in many tasks improves its performance with deep convolutional features. Therefore, the Part-based R-CNN [19] learns the part detectors by leveraging deep convolutional features computed on bottom-up region proposals. It extends R-CNN to detect objects and localize their parts under a geometric prior. Deep LAC, Part-stacked CNN, and Pose-normalized nets[10] are also used in the fine grained image classification.

2.3. General CNN

CNN has been used in various computer vision tasks since LeCun et al.[6] first introduced it. Due to the advent of large scale training data such as ImageNet, CNN exhibits more advantages over other methods. Impressed by this, researchers adapt CNNs pre-trained on ImageNet to other domains or datasets, such as the fine-grained image classification dataset. Most of the current state-of-the-art CNN can be used in fine-grained image classification[10].

In order to have a first taste on this problem, we decide to use AlexNet[5], VGG-16[8], ResNet-18[3], ResNet-50[3] networks and use Stanford Dog Dataset[4] as our approach.

3. Dataset and Features

3.1. Dataset

We use Stanford Dogs dataset[4] in our project. The Stanford Dogs dataset contains image of 120 breeds of dogs from all the world. The dataset is built into two parts. The first part is images from ImageNet[1] for the task of fine-grained image categorization. The second part is annotation files that link images to the path of directory. In total, the number of categories is 120, the number of images is 20580 and the annotation is class labels and bounding boxes, which help us locate the target we want.

3.2. Preprocessing

We apply data augmentation and random flips to images to get better results. Also, due to the high variance of images pixels, we rescale images to [0, 1] before further applying normalization.

We apply different data augmentation methods for training dataset and validation dataset. For training dataset, we apply data augmentation methods in the following order: resize to [256, 256], randomly rotate 45 degree, randomly crop the image to [224, 224]; for the validation dataset, we apply data augmentation methods in the following order: resize to [256, 256], crop the image to [224, 224] from the center. Finally, we flip the image horizontally and randomly. For all neural network architectures we use, we follow the standard of ImageNet training.

4. Methods

In this section, we will have a short introduction on the four well-known neural network architectures we used in this project, which are AlexNet, GoogLeNet, VGG-16 w/o BN and ResNet18.

4.1. AlexNet

AlexNet is introduced by Krizhevsky et al. in 2012 and won the 2012 ImageNet LSVRC-2012 competition by a



Figure 1. A Sample Image from Stanford Dog Dataset

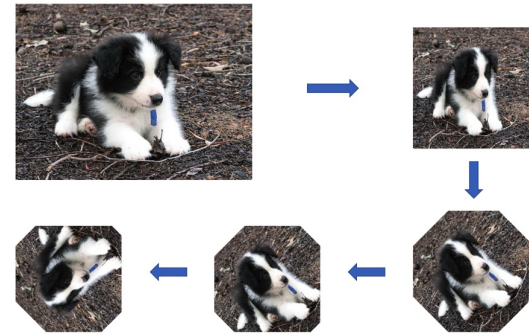


Figure 2. Data Augmentation for Training Dataset

large margin. It contains 5 convolutional layers and 3 fully-connected layers. ReLU is applied after every convolutional layers and fully-connected layers. Dropout is applied before the first and the second fully connected layer. The output of the last fully-connected layer is fed to a 120-way softmax function which produces a distribution over the 120 labels. The network maximizes the multinomial logistic regression objective, which is equivalent to maximizing the average loss training cases of the log-probability of the correct label under the prediction distribution. The network has 62.3 million parameters, and need nearly 1.1 billion computation units to proceed one forward pass.

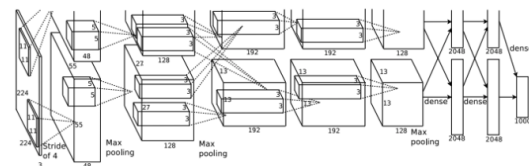


Figure 3. AlexNet Details

The standard softmax function σ is defined by the formula

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

where softmax performs a transform on n number $x_1 \dots x_n$.

The outputs of the softmax function transform are always in the range of $[0, 1]$ and add up to 1. Hence, they form a probability distribution.

4.2. GoogLeNet

GoogLeNet is introduced by Szegedy et al. in 2014 and won the 2014 ImageNet LSVRC-2014 competition. It achieved a top-5 error rate of 6.67%. It contains 1x1 convolution layer at the middle of the network. And global average pooling is used at the end of the network instead of using fully-connected layers. The most important technique in GoogLeNet is inception module, which has different sizes of convolution layers for the same input and stacking all the outputs as one output.

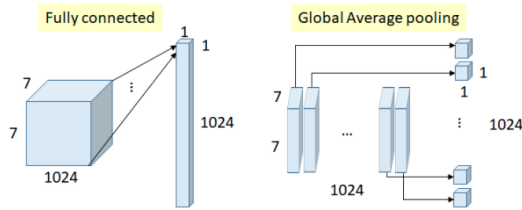


Figure 4. Global Pooling vs Average Pooling

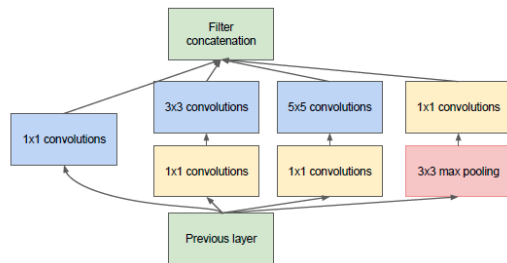


Figure 5. Inception Module Details

4.3. VGG16

VGG16 is a convolutional neural network model introduced by Simonyan et al. in 2014. The model achieved 7.3% top-5 error rate in ImageNet. It makes the improvement over AlexNet by replacing larger kernel-sized filter with multiple 3 x 3 kernel-sized filters one after another. There are 13 convolutional layers, 5 max pooling layers and 3 fully connected layer. ReLU is applied after every convolutional layers and fully connected layers exception the last one, which was followed by a softmax function. We chose to add batch normalization after each convolutional layer to speed-up the training process and fix the gradient vanishing problem in result getting a better result.

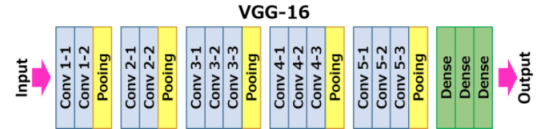


Figure 6. VGG16 without Batch Normalization Layers

4.4. ResNet-18

ResNet is introduced by He et al. in 2015 and won the ImageNet ISVRC-2015 competition. It achieved amazing result not only in single-label classification but also in generalization performance on other tasks including detection and segmentation. The key component for ResNet is residual block that skips one or more layers.

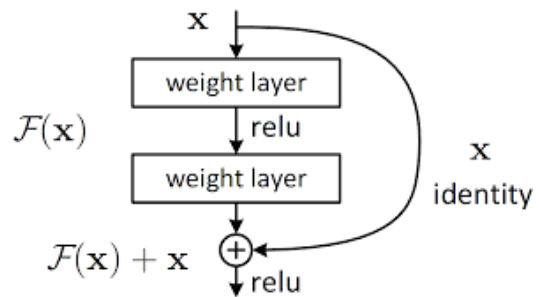


Figure 7. A Residual Block

5. Experiments

We trained 5 models from scratch for 100 epochs with initial learning rate is 0.1, 0.01 and 0.001 whose best accuracy shown in the left column of Fig.8. The performance of alexnet is the worst and accuracy of googlenet and vgg16.bn are better. Resnet18 has the best performance, whose accuracy is up to 95.85% when its learning rate is 0.1. Generally, the performance increases as the number of epoch increases. Here are some interesting observation. Alexnet, vgg16, vgg16.bn are more sensitive to initial learning rate, while googlenet isn't. As you can see, the Best Accuracy curves of googlenet with different initial learning rates have similar speed to increase as training more epochs. And finally they almost at the same point start leveling off. When learning rate is too large like 0.1, the performance of alexnet, vgg16, vgg16.bn is really bad. The reason is that the model cannot converge because of vibration.

The 5 models are also fine-tuned based on pre-trained models provided by ImageNet for 30 epochs whose best accuracy shown in the right column of Fig.9. Generally, their performance are better than the models we trained from scratch. All of best accuracy can reach to 80 percent. Especially for alexnet, using pre-trained model improves its performance significantly, which makes sense since pre-trained models have seen larger dataset, so they can generalize to

unseen data much better. The best result is from vgg16_bn, which is up to 94.25%. When the learning rate is too small like 1e-05, if we only train for 30 epochs, the curves do not start to level off, so the performance is bad. And when learning rate is too large, alexnet, vgg16 and vgg16_bn still cannot converge.

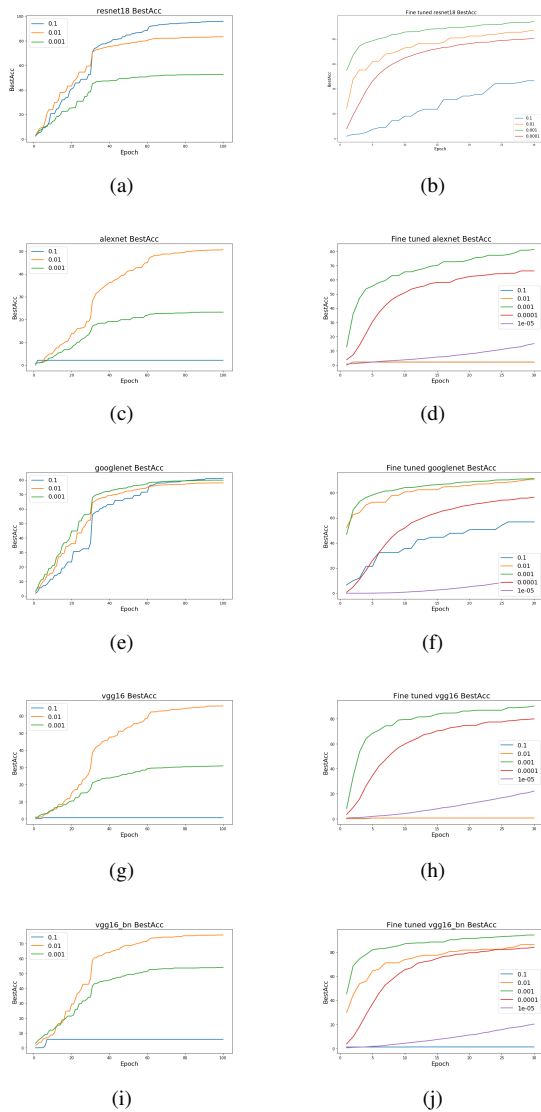


Figure 8. Best Accuracy

Training loss and validation loss are shown in Fig.9. The trend of loss curves is similar to best accuracy curves to some extent. Googlenet is the least sensitive to initial learning rates, while resnet18, alexnet, vgg16 and vgg16_bn is more sensitive. When the initial learning rate is too small like 0.1, losses of alexnet, vgg16 and vgg16_bn are extremely large or small so they are discarded in the figure. Generally, loss decreases as the number of step increase and start to level off at some point.

Above experiments are conducted on Datahub machine with 1 GPU, 32 RAM and 8 CPUs.

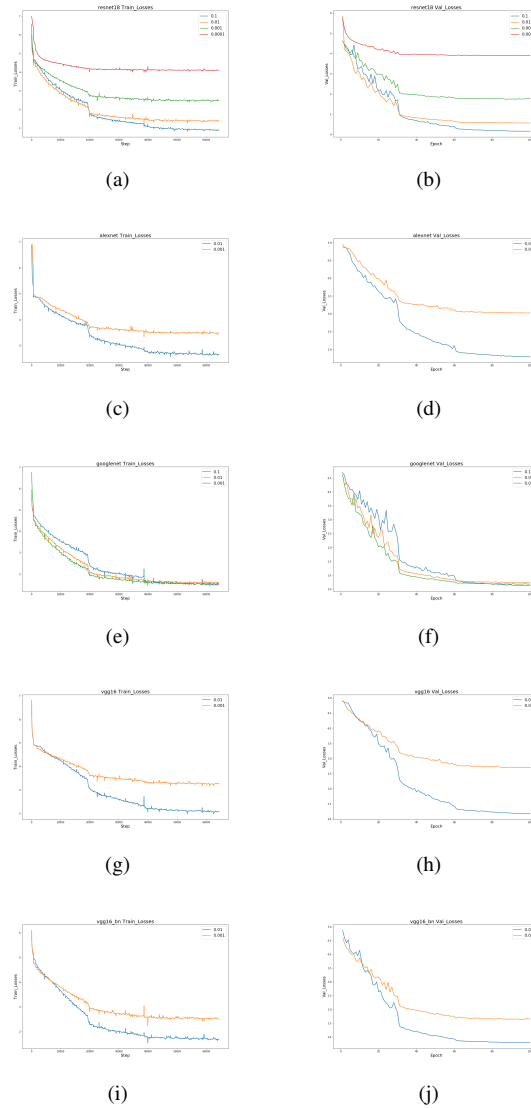


Figure 9. Training and Validation Loss

6. Conclusion

In the project, we reimplemented 5 CNN models—resnet, alexnet, googlenet, vgg and vgg_bn to do dog breed classification based on Stanford Dog Dataset. Then we trained and fine-tuned 5 models and provided experimental results and empirical observations. The best accuracy can reach to 95.85%, which meets our expectation. Furthermore, we studied the transfer learning ability from a broad, general classification problem in a fine-grained classification problem.

7. Contributions

Yudian Li: Pre-process dataset, train models and fine-tune hyperparameter on ResNet18.

Shanshan Xiao: Train models and fine-tune hyperparameter on VGG16 and VGG16.bn.

Pu Cheng: Train models and fine-tune hyperparameter on AlexNet and GoogLeNet.

References

- [1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 2
- [2] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *Proceedings of the IEEE international conference on computer vision*, pages 1713–1720, 2013. 1
- [3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2
- [4] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011. 1, 2
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2
- [6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 2
- [7] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur. Dog breed classification using part localization. In *European conference on computer vision*, pages 172–185. Springer, 2012. 1
- [8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2
- [9] X.-S. Wei, J. Wu, and Q. Cui. Deep learning for fine-grained image analysis: A survey. *arXiv preprint arXiv:1907.03069*, 2019. 1
- [10] B. Zhao, J. Feng, X. Wu, and S. Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017. 1, 2