Fine-grained Image Classification with Semi-supervised Deep Learning Algorithm

Xinqi Shen **Chenyang Lu**

Abstract

Fine-grained image classification is one of the popular research directions in the computer vision field. The current approaches are mainly based on deep learning and researchers focus on investigating models and algorithms. There are some state-of-the-art models, such as ResNet, DenseNet and EfficientNet etc. In our project, we pay more attention to data extension by implementing multiple data augmentation techniques combined with the semi-supervised learning method to expand our dataset. Then, we compare the performance of different models using our extended dataset. Finally, in order to get better performance, we ensemble the results from different neural network models.

1 Introduction

The fine-grained image classification task focuses on differentiating between hard-to-distinguish object classes. These categories are both visually and semantically very similar, it is very difficult and even challenging for humans without careful training, and is critical for establishing a more detailed understanding of the visual world. In our project, we use the Standford Dogs Dataset[1] which contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotations from ImageNet for the task of fine-grained image categorization. There are some similar fine-grained visual categorization datasets, such as Caltech-UCSD 200 Birds dataset[2], Standford cars dataset[3] and Oxford 102 Flowers dataset[4]. This type of dataset is studied to test the performance of different models or algorithms of the fine-grained visual categorization problem.

Related Work The state-of-the-art model for image classification is EfficientNet[5], it is based on the neural architecture search and introduces a compound model scaling algorithm to scale the width, depth of model and resolution of the input image. EfficientNet tries to find the optimal amplified model from the baseline model. Another state-of-the-art model is Vision Transformer[6], it introduces the transformer model that is outperformed in the NLP area to image classification problem. The input image is split into pitches and then position embedding before feed into the transformer encoder that contains multi-head attention and MLP. Both EfficientNet and Vision Transformer outperformed ImageNet for the image classification task, however, they require huge computational cost.

2 Dataset

2.1 Statistics

The Standford Dogs Dataset as a fine-grained categorization dataset, it has some challenges due to a variety of reasons. First, there is litter inter-class variation, different classes may have a very similar colour or facial characteristics. Then, there exists a large intra-class variation, dogs from the same breed could have different ages, poses, occlusion/self-occlusion and even colour. Also, a large proportion of the images contain humans and are taken in man-made environments causes greater background variation. The total number of images is 20,580, however, due to the 120 classes, the average number of images per class is only around 170 even before splitting into train, validation

and test set. Also, the dataset exists unbalance, the maximum class contains 252 images while the minimum class only has 148 images. Thus, in order to expand our dataset, we will implement multiple image argumentation skills and the semi-supervised learning method.

Figure 1: Variation in the Standford Dogs Dataset. Upper two dogs from different breeds, bottom two dogs from the same breed.

2.2 Data augmentation and extension

2.2.1 Transformation

Image transformation is the basic but powerful augmentation technique that is popularly used, it increases the diversity of training data and helps reduce overfitting. In our project, we use a series of image transformation includes random horizontal flip, random rotation, random resized crop, translation, and gaussian noise. Although it improves the performance, the drawback of this approach is also obvious. It only augments the single image and does not model the relation across examples of different classes.

2.2.2 Mix up

The idea of Mix up[7] is very simple, it forms a new example through weighted linear interpolation of two existing examples.

$$\lambda \sim Beta(\alpha, \alpha), \text{ where } \alpha \text{ is a hyperparameter}$$

$$\widetilde{x} = \lambda x_i + (1-\lambda)x_j, \text{ where } x_i, x_j \text{ are raw input vectors}$$

$$\widetilde{y} = \lambda y_i + (1-\lambda)y_j, \text{ where } y_i, y_j \text{ are one hot label encodings}$$

Mix up makes decision boundaries transit linearly from class to class, providing a smoother estimate of uncertainty and reduces the memorization of labels.

2.2.3 Semi-supervised learning

In order to extend our dataset, we use another fine-grained dogs dataset, Tsinghua Dogs Dataset. It contains 70428 images and most of the categories are the same as the Standford Dogs Dataset, but we only use the images, not labels. In reality, the large labelled dataset is not always possible and it takes time and needs many labellers. Also, it is highly likely that the unlabelled data is much more than labelled data. Thus, we propose a semi-supervised workflow to extend our dataset, specifically the self-training method. We first train our model with the labelled training set, then fine-tune the model and select the best model which has the highest validation accuracy. After that, we predict every unlabelled image using our model and calculate the probability of the predicted class by softmax. If the probability is greater than the set threshold, we consider it as highly confident and assign a pseudo label. After we test all the unlabelled images, we will train our new model based on the images with a true label or a pseudo label. Depend on the usage, we use 0.9 as our threshold and it can be adjusted. According to this workflow, our new training data is almost three times more than the original training set.

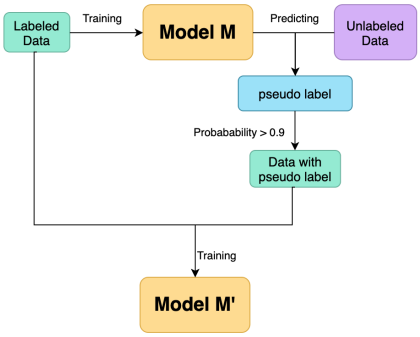

Figure 2: Proposed semi-supervised learning workflow

3 Models

3.1 ResNet

For our primary model, we used ResNet 50[8] to do the classification task. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150 more layers successfully. Prior to ResNet, training a very deep neural network was difficult due to the problem of vanishing gradients. We pick the ResNet 50 model because this model is one of the most popular classification models nowadays. When we train the ResNet model we find the training set has much higher accuracy than the validation set. So, we decrease the number of epochs for training and set the learning rate to a small 1e-4 to avoid the overfitting issue. We also used early stopping to stop training when the validation loss began to increase.

3.2 VGGNet

Another model we considered to use is the VGG16 model[9] with batch normalization. With a small kernel size, the model is more non-linear and has fewer parameters to train when compared with the AlexNet. The deep depth of the model increases prediction accuracy. However, this model is still too parameter-heavy and takes a long time to train.

3.3 GoogLeNet

We also tried GoogLeNet[10]. This model includes 9 inception blocks. The original version of the inception block contains 1*1, 3*3, 5*5 convolution layer and 3*3 convolution layer. Then they concatenate the output together to get the output. However, this original block has too many parameters so it introduces 1*1 convolution layer to reduce dimension. GoogLenet has 22 layers but has only 1/36 parameters compared with VGG net.

Optimization To make our model get higher accuracy, we used transfer learning and bagging to help classification. For transfer learning, since we found that all the three models mentioned above were pre-trained on the imagenet which is a very large dataset of classification. We first get the pre-trained model and then we change the last hidden layer of each model to output 120 features which is our class size. Later, We fine-tuned the model with our dog breed dataset. During validation, we found that all three models were good choices to use for our classification, so we used bagging to help increase accuracy. For each image, we first get each model's prediction class and check if they have a majority vote class. If two or three models agreed with one class, we use this as our final prediction. If all of them get different labels, then we check the confidence of each model and pick the label produced by the model which has the highest confidence.

Experiment For our experiment, we used the 224×224, with 3 channel images cropped by the bounding box and their corresponding class as the input. For each model, we transform the original image with different augmentation including no augmentation, semi-supervised learning, normal augmentation and mixup augmentation. We conduct experiments for our CNN baseline, ResNet50 without transfer learning, ResNet50 with transfer learning, VGG model, GoogLeNet and the bagging

of VGG, GoogLeNet with ResNet50. During training, we train each model with 15 epochs and use Adam as an optimizer with a learning rate of 1e-4. These models are trained with a batch size of 32 on one GPU. For testing, we only evaluate the original images which have been assigned to the test set. For accuracy, we use top-1 accuracy rate as the metric.

4 Results

Table 1: Top-1 Accuracy across different data augmentation or extension methods

Method	Original dataset	Normal augmentation	Mixup	Semi-supervised
Top-1 accuracy	75.25%	76.45%	78.44%	85.72%

Table 1 shows the accuracy of various augmentation settings applied to the pre-trained ResNet. From the table, we see that semi-supervised learning can increase accuracy by around 10 percent which is really large. Mixup augmentation and normal augmentation both increased the accuracy but the improvement is not significant.

Table 2: Top-1 Accuracy across different models

Model	Baseline	ResNet	Pre-trained ResNet	Pre-trained VGGNet	Pre-trained GoogLeNet	Bagging
Top-1 accuracy	36.73%	40.13%	85.72%	80.07%	82.61%	86.53%

Table 2 shows the accuracy of different models using semi-supervised augmentation. From here, we can see that the baseline accuracy is really low. ResNet without pre-training outperforms the baseline. The reason might be ResNet has much more depth and we have 120 different classes so we might need a complex model to learn all features. When we compared ResNet with and without pre-training, we can see that transfer learning really helps to tune the weight of the network. Since even after semi-supervised learning, we still don't have enough data to train a very deep model. With transfer learning, the accuracy increased by around 45 percent. When we compared ResNet, VGGnet, and GoogLeNet which all pre-trained with ImageNet, we can see that they all achieve high accuracy and ResNet gets 5 percent more than VGGNet and 3 percent more than GoogLeNet. Also, when we use bagging to combine the models together, it gets a higher accuracy since it makes use of all the models.

5 Discussion and conclusion

We used different data augmentation methods and different models to find a way of increasing classification of the Stanford Dog breed dataset. when we use semi-supervised learning and using bagging of VGGNet, ResNet and GoogLeNet as our model, we get the best result 86.53%. It gets 50 percent more than our baseline. Since we only use the image itself as the feature for the model, we cannot tell which feature represents well and because our model is a deep neural network, we cannot interpret the parameters in the network. The reason our supposed data augmentation and model works is that our original dataset is too small to train a deep neural network. With the help of semi-augmentation, we can increase our data set size by a large amount. Also, our output has 120 different classes, it is hard to get an accurate result if we use a simple CNN model. We think the reason ResNet outperforms is that it has large depth to increase accuracy and uses the "residual block" to solve the degradation problem. So it can learn our data better than the others. We think the reason VGGNet has lower accuracy than the other two is because it only has 9 layers so it cannot learn very deep knowledge in the images.

Future Work If we have more time, we would try to modify the structure of ResNet and use this as the backbone and include some latest architecture like SE block to see if we can get a better model with higher accuracy.

References

[1] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. June 2011.

[2] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[3] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[4] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[5] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[7] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

Attributions

All group members contributed equally:
Xinqi Shen: mainly work on data cleaning, image pre-processing, image augmentation, and semi-supervised learning part.
Chenyang Lu: mainly work on model establishment, model training, and model comparison.
Other parts work together.

Code is available on github: `https://github.com/lcytoronto/2515project`