Oracle Character Recognition

Haoxian Chen
School of Data Science
Fudan University
220 Handan Rd., Shanghai
18307110276@fudan.edu.cn

Xinghua Jia
School of Data Science
Fudan University
220 Handan Rd., Shanghai
18300290007@fudan.edu.cn

Abstract

In this project, we solve the problem of oracle character recognition on dataset Oracle-FS. Oracle character recognition is a challenging task in few-shot learning due to the data limitation and imbalance. In order to solve this problem, we propose a model using mixup, augmented feature vector, ensemble backbone, feature vector preprocessing and nearest class classifier, and train it using the training data only. The best model approach 61.3%, 89.79%, 95.87% top-1 accuracy on 1-shot, 3-shot and 5-shot Oracle-FS. And we also do some ablation experiments to show the effect of these methods and try to explain them.

1 Introduction

Few-shot learning aims at leveraging knowledge learned by one or more deep learning models, in order to obtain good classification performance on new problems, where only a few labeled samples per class are available. Few-shot learning is becoming more and more popular and important in deep learning, for the concepts revealed by few-shot learning have more generalization and robustness.

Methods in few-shot learning can be mainly divided into three: model based, metric based and optimization based methods. Model based method aims to update parameters quickly on a small number of samples through a well-designed model structure, and directly establish the mapping function from input to the prediction. Metric based method solves few-shot learning by measuring the distance between training samples and testing samples and classifying with the distance, which often uses the idea of nearest neighbor. Optimization based method thinks that general gradient descent method is difficult to optimize in the few-shot learning setting, so this task can be solved by adjusting the optimization methods.

In this paper, we focus on the problem of oracle character recognition. Oracle characters are the earliest known hieroglyphs in China, which were carved on animal bones or turtle plastrons in purpose of pyromantic divination of weather, state power, warfare and, trading to mitigate uncertainty in the Shang dynasty [1]. Due to the scarcity of oracle bones and the long-tail problem in the usage of characters, oracle character recognition suffers from the problem of data limitation and imbalance. Data limitation means extremely limited samples, making oracle character recognition a natural few-shot learning problem. Imbalance means a high degree of intra-class variance in the shapes of oracle characters, since oracle bones were carved by different ancient people in various regions over tens of hundreds of years. As a result, oracle character recognition is a challenging task in few-shot learning.

The oracle character recognition problem we want to solve aims at utilizing large-scale unlabeled source data, including unlabeled oracle or other ancient Chinese characters, to facilitate novel oracle characters recognition. We are able to use unlabeled source data and training data to train our model, and evaluate it in the dataset Oracle-FS. However, we shows that for this problem, we can simply use the training data and get a rather good performance through the model. We believe that this model can be expanded to solve the practical oracle recognition.

More precisely, in this paper, we:

- Introduce a model to solve the oracle recognition problem on the dataset Oracle-FS, with only training data used;
- Show the effect of the methods we used by doing abundant ablation experiments;
- Show the best performance of our model on Oracle-FS under 1, 3 and 5 shots.

2 Related Work

Many approaches have been proposed in the field of few-sample learning, and in this section we introduce the state-of-the-art works in a pipeline order of learning processes. Our work is related to these efforts in that our approach uses multiple components from these efforts, but differs in that we combine these components in a new way to achieve the best results on Oracle's few shot dataset.

2.1 Background

Oracle character recognition suffers from data limitations and imbalances, so it is necessary to study few shot learning of Oracle. Our work uses the Oracle's few shot dataset designed by Han et al. [2] and uses the experimental results from their work as baseline.

There is also some work in the oracle field that attempts to repair oracle fragments [3], which is also very interesting, but is not used in our work because the samples in the Oracle's few shot dataset are very complete.

In the few shot learning task, Bendou et al. show that the combination of some simple components can outperform complex models [4], and validate this conclusion on some classical datasets such as such as MiniImageNet and CIFAR-FS. Our work is inspired by their idea.

2.2 Data Augmentation

Many studies have focused on expanding the data with self-supervised learning, requiring the use of large amounts of unlabeled oracle image data [2], but some recent work has found that simple rotated samples do not perform poorly either [5], [6].

Besides, random cropping [7], mixup [8] and manifold-mixup [9] can be used to solve the challenge of lack data. Random cropping helps the model to focus better, and mixup and mainfold-mixup use linear interpolation to fuse samples in the input and intermediate layers, respectively, to improve the model's discrimination ability.

2.3 Backbones

Some classical networks can be used as backbone to extract feature vectors, including ResNet [10] and WideResNet [11]. In further, Huang et al. proposed an ensemble method capable of fusing multiple networks without additional training costs [12], which can improve the situation where the model is attracted to locally optimal solutions; Mangla et al. propose S2M2R, a method to change the network sturcture, to make single backbone performs better [9].

2.4 Feature Processing and Predicting

The process between feature vectors and prediction can be further optimized for few-sample learning. Hu et al. propose a method to make the feature vector closer to the Gaussian distribution [13], which helps improve the prediction accuracy. Furthermore, Wang et al. have shown that the method of classifying few shot datasets directly using nearest neighbor classifier can achieve very good classification results [14]. Therefore, in our work, the model always uses the nearest neighbor classifier, while the feature vector preprocessing requires further investigation of its effect

3 Methodology

Our methodology consists of 5 parts, which will be illustrated thereafter and also shown in Figure 1.

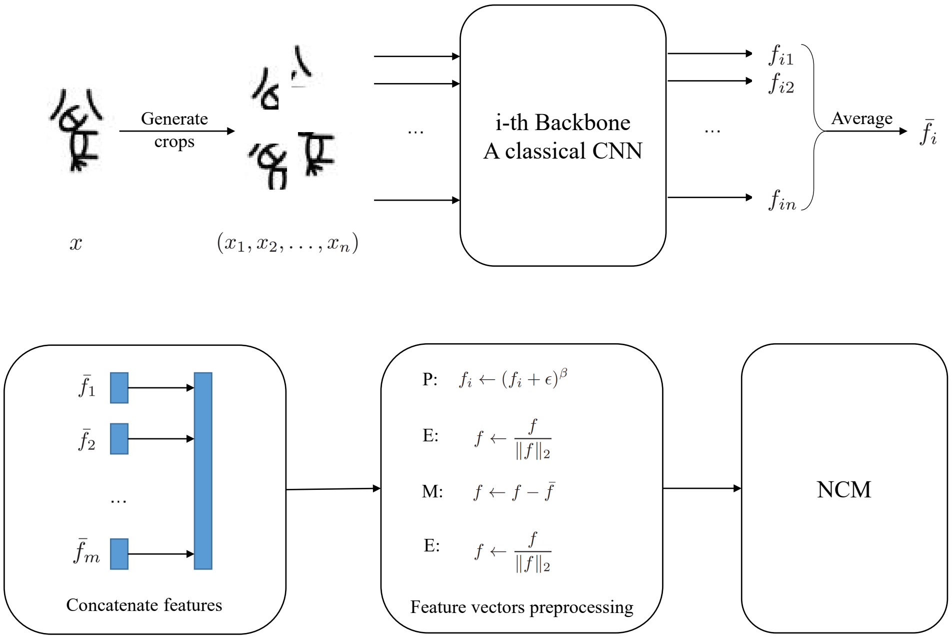

Figure 1: Methodology

3.1 Backbone

We use the pixel format data in our model. We firstly use data augmentation with random resized crops, and random horizontal flips to make the data have more diversity. This is because we found the three channels of every training and testing data are the same so this simple data augmentation on an image is enough for increasing the diversity. But we still treat the images as three channels in the model. We also normalize the data with mean 0 and standard error 1.

We try a self-supervised methodology to train the network, which is inspired by S2M2 in [9]. We choose a standard classification network, such as ResNet18, WideResNet, and add a new branch for logistic regression classifier after the penultimate layer. While the original branch is still be used to predict the class of the input samples, the new branch is to retrieve which one of four possible rotations (quarters of 360° turns) has been applied to the input samples. The training is divided into two steps. The first step is to feed the feature to original classifier and the second is to rotate the input data arbitrary and feed to the two classifier. Although this can enhance the robustness of the model, we found this not work in this problem, for the reason that the images are almost upright and the rotation makes no sense. We will show this in the experiment.

We use MixUp and Manifold Mixup respectively to do the data augmentation. MixUp assigned a weighted linear interpolation of random image pairs from the training data and enhance the robustness to adversarial samples. Given two images with their ground truth labels $(x_i, y_i), (x_j, y_j)$, a synthetic training example:

$$(\hat{x}, \hat{y}) = \lambda(x_i, y_i) + (1 - \lambda)(x_j, y_j)$$

where $\lambda \sim \text{Beta}(\alpha, \alpha) \in [0,\ 1]$, and $\alpha \in [0,\ +\infty)$ controls the intensity of interpolation. But these type mixup images are overlays and tend to be unnatural.

Instead of linear interpolating the input data, Manifold Mixup combined the data in the hidden layers. Since the convolution layers act as a feature extractor, the data through the hidden layer can be seen as the feature of the original data, and therefore the linear interpolation of them can combine their features better. Manifold Mixup can learn robust features and the representations learned by it are more discriminative and compact.

We use this two data augmentations instead of other augmentations, such as CutOut, CutMix, for the special characteristics of the images. Although MixUp images are overlays and tend to be unnatural in many cases, in the oracle character the interpolation is more explicable: it can be seen as an image with the two labels for human. Considering this, we also do the experiment of setting $\lambda = 0.5$ to fully exploit the images and find it really works.

3.2 Augmented Feature Vectors

We use random resized crops on the images and feed it to the backbone to generate multiple feature vectors. We average them to obtain the augmented feature vectors. This will give more features for it considers different regions of an image. In practice, we generate 30 crops for each image.

3.3 Ensemble Backbones

In order to boost the performance, we propose an ensemble of backbones on the basis of augmented feature vectors. We concatenate the feature vectors using the same network but with different crops. In practice, we use different random seeds to perform it. This is a enhancement of augmented feature vectors and we test the performance of different numbers of backbones in the experiments.

3.4 Feature Vectors Preprocessing

After obtaining the feature vectors f, we preprocess them to make them more expressive. We do the following three operations.

3.4.1 Power Transform

Power transform operation simply consists of taking the power of each feature vector coordinate. The formula is given by

$$f_i \leftarrow (f_i + \epsilon)^{\beta}$$

where $\epsilon = 1e - 6$ is used to make sure f_i is strictly positive and β is a hyper-parameter and we set it 0.5 in our experiment.

3.4.2 Euclidean Normalization

Euclidean Normalization is used to get a vector with Euclidean norm 1. This step scales the features, making large variance feature vectors do not predominate the others. It can be seen as a projection on the unit hypersphere. The formula is given by

$$f \leftarrow \frac{f}{\|f\|_2}$$

3.4.3 Mean Subtraction

Mean subtraction is used to make the vectors get a zeros mean. Denote $\bar{f}$ the average vector of all the feature vectors, the formula is given by

$$f \leftarrow f - \bar{f}$$

In our model, we deploy these three operations in order. We do a power transform (P) firstly, continued with an Euclidean normalization (E). Then we do the mean subtraction (M), followed by an Euclidean normalization (E) also. This sequential operation can be abbreviate PEME and it can make the data better align with a Gaussian distribution. Therefore, this preprocessing is able to make the feature vectors more separable and thus more expressive.

3.5 Classification

To do the classification with the feature vectors, we use the Nearest Class Mean Classifier (NCM). Denote S_i the sets of feature vectors from the i-th class, then the barycenter of the class is:

$$c_i = \frac{1}{|S_i|} \sum_{f \in S_i} f$$

Then for a new sample, the predicted class is decided by the closest barycenter from the feature vector $\hat{f}$. The formula is given by

$$C(\hat{f}) = \arg\min_i \|\hat{f} - c_i\|_2$$

4 Experiments

In this section, we experimented our model on the Oracle-FS dataset with the same hyperparameters for each set of experiments, shown as table 1.

Table 1: Hyperparameter Setting

Dataset	Epoches	Loss function	Optimizer	Intial lr	Momentum	Weight decay
Oracle-FS	200	CrossEntropy	SGD	0.1*	0.9	5e-4

* lr will be 0.01 after 100 epochs

The Oracle-FS dataset is divided into three versions, 1-shot, 3-shot and 5-shot. The k-shot version means that each Oracle is provided with only k training samples, but each Oracle is provided with the same number of test samples of 20.

4.1 Preliminary Experiment

We first enabled all the methods mentioned in Section 3 unchanged and used WideResNet as Backbone. The WideResNet we used has depth 28 and widen factor 10. The test results of the model on three versions of the Oracle dataset are shown in Table 2.

Table 2: k-shot accuarcy of the model with all potential methods on Oracle-FS dataset

Model	Rotations	Mixup	PEME	k-shot	Top-1 Acc
WideResNet(×16)*	✓	✓	✓	1	58.97
				3	86.91
				5	94.68

* (×16) means that the number of ensembled backbones is 16

In the paper [2], the best top-1 accuracies of their model on three versions of the Oracle dataset are 31.9%, 58.3% and 69.3%, respectively. Thus, the preliminary experiments of our model alone can improve about 27% compared to baseline, and we further adjusted the method combination and details in the method through ablation experiments to obtain higher testing accuracy.

4.2 Changing the Backbones structrue

We controlled the other factors in the model constant, only changed the CNN network used by backbone or change the number of ensembled backbones for comparison test, the top-1 accuracy on 1-shot Oracle-FS dataset of each group is shown in the table 3.

Table 3: 1-shot accuarcy of the model with different backbone structrue

k-shot	Mixup	Rotations	PEME	Model	Top-1 Acc
1	✓	✓	✓	ResNet20(×4)	51.32
				ResNet20(×8)	54.28
				ResNet20(×16)	**58.75**
				ResNet20(×20)	57.05
				ResNet20(×16)	58.75
				ResNet18(×16)	54.35
				WideResNet(×16)	**58.97**

It is easy to see that as the number of ensembled backbones increases, the accuracy of the model shows a trend of first increasing and then decreasing, which is in line with the theory. On the one hand, a single backbone may converge to a local optimum in training, while by ensembling can give the model a wider view and thus break away from the local optimum more easily; on the other hand, when the number of ensembles is too large, the model may also overfit and lead to a decrease in accuracy. Therefore, setting the number of ensembled backbones to 16 is the best for Oracle-FS dataset.

The experiments of selecting different CNN as backbone shows that as the complexity of the model goes up, the accuracy of the final model on the test set improves. This can be seen in a way as an advantage of few shot learning, i.e., less likely to overfit compared to traditional machine learning due to fewer training samples, especially for more complex models. Therefore, the most complex WideResNet network is the best for the Oracle-FS dataset.

4.3 Changing the preprocessing Methodology

Immediately afterwards, we fix the best backbone structure and change the preprocessed data and feature vectors of the method groups for comparison test. The top-1 accuracy of each group on the 3 versions of Oracle-FS dataset is shown in Table 4.

Table 4: k-shot accuarcy of the model with different preprocessing methodology

Model	k-shot	Rotations	Mixup	PEME	Top-1 Acc
WideResNet(×16)	1				51.77
		✓			52.60
			✓		53.09
				✓	55.04 ± 4.55*
		✓		✓	57.58
			$\lambda = 0.5$	✓	59.35
			manifold	✓	60.07
		✓	✓	✓	58.97
		✓	$\lambda = 0.5$	✓	**61.30**
		✓	manifold	✓	55.40
	3	✓	✓	✓	86.9
		✓	manifold	✓	**89.79**
	5	✓	✓	✓	94.27
		✓	manifold	✓	**95.87**

* The reult of this group is not very stable, so we give give a relative error to show

First, it can be observed that preprocessing with any of rotations, PEME and mixup performs better than no preprocessing, which proves that all three methods are effective for model enhancement on Oracle-FS dataset. Among them, PEME is the most enhanced and at the same time the most unstable, because it is a processing approach for the feature vectors, which are obviously more sensitive to changes than the sample inputs, and therefore leads to large variability of the results. Nevertheless, we believe that the PEME approach is necessary because it brings far more enhancement expectations than the other two.

Then, the following rows of the table record the results of the experiments combining the three preprocessing methods. Moreover, we adjusted the mixup in some groups by setting the mixup parameter λ to 0.5 rather than randomly selected or by using the manifold-mixup.

We found that using all three preprocessing methods together and setting the mixup parameter to 0.5 achieved the best results on the 1-shot dataset, while using the manifold-mixup did not perform significantly better than the classical mixup as suggested in the paper [9]. This may be due to the fact that the sample data of few shot learning is too small, so once it is used together with rotations method, it confuses the model, and the advantage of manifold-mixup will be gradually reflected when the shots of training set increases.

Finally, we validated the above elaboration on the 3-shot dataset and on the 5-shot dataset, and observe a significant improvement in accuracy from using manifold-mixup, as shown in the last rows of the table.

5 Conclusion

To summarize, our approach achieves 61.3%, 89.79% and 95.87% top-1 accuracy on 1-shot, 3-shot and 5-shot Oracle-FS datasets, respectively. Compared to the baseline provided in the paper [2], each shot version improves by 29.4%, 31.49% and 26.57% respectively.

The paper [4] experimentally demonstrates that the combination of some simple methods in few shot learning is more effective compared to complex models, and now we draw the same conclusions by designing models and experiments on the Oracle-FS dataset.

References

[1] David N Keightley. Graphs, words, and meanings: Three reference works for shang oracle-bone studies, with an excursus on the religious role of the day or sun, 1997.

[2] Wenhui Han, Xinlin Ren, Hangyu Lin, Yanwei Fu, and Xiangyang Xue. Self-supervised learning of orc-bert augmentator for recognizing few-shot oracle characters. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[3] Yaolin Tian, Weize Gao, Xuxing Liu, Shanxiong Chen, and Bofeng Mo. The research on rejoining of the oracle bone rubbings based on curve matching. *Transactions on Asian and Low-Resource Language Information Processing*, 20(6):1–17, 2021.

[4] Yassir Bendou, Yuqing Hu, Raphael Lafargue, Giulia Lioi, Bastien Pasdeloup, Stéphane Pateux, and Vincent Gripon. Easy: Ensemble augmented-shot y-shaped learning: State-of-the-art few-shot classification with simple ingredients. *arXiv preprint arXiv:2201.09699*, 2022.

[5] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[6] Jialin Liu, Fei Chao, and Chih-Min Lin. Task augmentation by rotating for meta-learning. *arXiv preprint arXiv:2003.00804*, 2020.

[7] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12203–12213, 2020.

[8] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[9] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2218–2227, 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[12] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

[13] Yuqing Hu, Stéphane Pateux, and Vincent Gripon. Squeezing backbone feature distributions to the max for efficient few-shot learning. *Algorithms*, 15(5):147, 2022.

[14] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.