Unsupervised Few-Shot Oracle Character Recognition

Yifan Hu **Junzhe Jiang**
Fudan University
{yfhu19, jzjiang19}@fudan.edu.cn

Abstract

The recognition of oracle character is widely considered as a hard task for the data limitation and imbalance. Because only having few labeled oracle characters and large-scale unlabeled source Chinese characters, we can't use a transfer learning model to deal with this few-shot learning task, and thus the only possible methodology is data augmentation. In this paper, we fully take advantage of the traditional data augmentation and the Orc-Bert method, and get better results (ACC@1: **53.17%**, **74.48%** and **83.65%** for **1-shot**, **3-shot** and **5-shot**, respectively) on the Oracle-FS dataset. The code has been opened on `https://github.com/selfspin/Few-Shot-Oracle-Character-Recognition`.

1 Introduction

Oracle characters - carved words on animal bones or turtle plastrons - are indispensable for archaeological work and historical research. By studying these characters, researchers can learn about the climate , customs, culture and glorious civilization of ancient China. However, the recognition of oracle bone inscriptions has been puzzling numerous researchers, due to the scarcity of oracle bones and the long-tail problem. Oracle characters classification, indeed, is a challenging few-shot classification problem. Luckily, previous works [4, 8] inspire us to solve the problem through data augmentation and improving network structure.

Traditionally, data augmentation transforms the input data while training, without enlarging the data set. We, however, discovered that enlarging the data set with the conventional data augmentation methods(rotation, linear transform, flip, crop, and color jitter) will yield better results. In fact, after enlarging the data set by 20 times with Conventional Data Augmentation, we receive an ACC@1 on one shot learning for 41.57% (30.3% improvement compared to the result in Orc-Bert [4]).

We further discover that parameters pretrained on ImageNet will help the network converges to a better result on Resnet and Efficient Net. Without data augmentation, only using the pretrained parameters will enhance the ACC@1 on one shot learning to 38% from 18%. We proposed that the information in pretrained parameters help recognize the Oracle characters.

What's more. The volatile nature of the ancient character in this task may invalidate popular loss functions like ArcFace loss and Focal Loss [2]. These two methods decrease the final accuracy by 8% in experiments. On the other hand, dropout layers will improve the classification accuracy, enhancing the generalization ability of our model.

To summarize, simply enlarging the data set with a combination of conventional data augmentation method and Orc-Bert, coupled with dropout Layer, L_2 penalization and parameters pre-trained on Imagenet, the resnet18 Network can produce an unbelievable high baseline (ACC@1: 53.17% for 1-shot, 74.48% for 3-shot, and 83.65% for 5-shot). The contribution of this papar is as follows:

1. We critically find that better convergence results can be achieved by enlarging the data set with conventional data augmentation methods. Then, we propose a new method combining Orc-Bert and conventional data augmentation.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2. We emphasize the role of parameter initialization and Network structures in this paper, which has long been neglected in previous research. We hope that these renaissance techniques can shed more light on follow-up research.
3. An easy and powerful structure for Oracle classification is proposed in this paper, which makes it convenient for follow-up research. Also, the simple techniques will be easy for application: only training on a GeForce RTX 2080Ti for 30 minutes will yields excellent results.

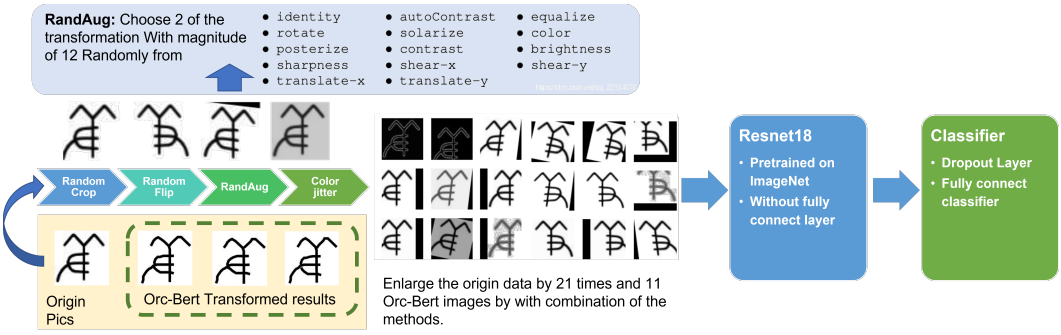

Figure 1: Data augmentation and network structures used in this paper.

2 Related work

The task we studies comes from the setting of [4], as an few-shot two hundred ways problem. The computational complexity makes it impossible to use methods based on similarity comparison, such as Siamese network [10] and too limited data also exclude meta-learning based method.

Transformers and Orc-Bert Data Augmentor

Recently, researchers believe that "attention is all your need" [13], and the models based on Transformer are dominating the performance on almost all NLP tasks. Particularly, BERT [3] exploited the mask language model as pre-training task.

Han and Liu proposed a way of solving this task - data augmentation. They innovatively introduce the BERT method to the field of data augmentation (named Orc-Bert) and achieve great promotion with respect to the traditional methods. However, our study finds out several limitation in the experiments of the literature.

First of all, the comparison between Orc-Bert Method and Conventional Data Augmentation is not fair. The Orc-Bert will enlarge the data set by 80 times in the literature, while the Conventional Data Augmentation have only the original number of pictures in data set for training.

Secondly, the volatile nature of the ancient character also leads to difficulties. Because ancient China was governed by different leaders, or there was misinformation in the spread of words, the same oracle bone inscriptions may have different shapes, as shown in Figure 2. So it still remains to be skeptical about whether using Orc-Bert to recover the shape is efficient. The Orc-Bert Method will only recover the strokes to be really similar to the original picture, or to be really different.

On the other hand, we notice that horizontal flipping may work well, since many characters are mirror symmetric. What's more, crop and Random linear transformation may cut out the features at boundaries of a character, where is the most heavily diverse part of a Oracle character. As a result, conventional data augmentation method may work well.

Traditional Data Augmentation, Dropout and Effective Losses

We further notice the strong augmentation RandAugment[1]. This method choose randomly N data augmentation from Identity, AutoContrast, Equalize, Rotate, Solarize, Color, Posterize, Contrast, Brightness, Sharpness, ShearX, ShearY, TranslateX, TranslateY. Then, change the picture with the chosen transformation with strength(magnitude) M. The simple method achieve SOTA results on CIFA10, and is easy to implement and have very small parameter search space. So in this experiment, we implement the method to augment the data.

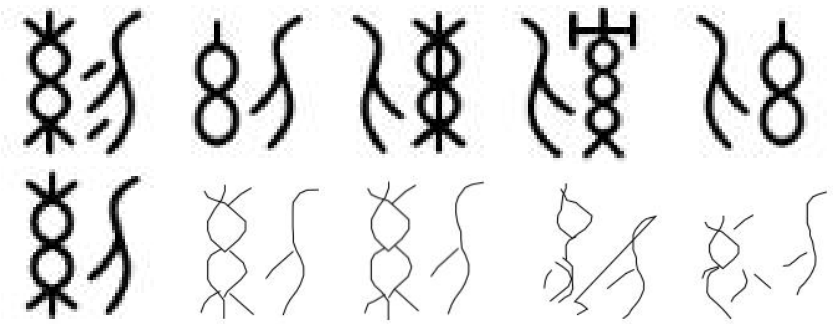

Figure 2: Above five oracles which all correspond to the same Chinese character 刺, with really different shape. While on the bottom line, the first character is the picture from the one shot training set. The following four pictures are the results produced by Bert. As we can see, none of the produced pictures can characterize the diverse distribution of Oracle character.

Furthermore, the Literature neglects the power of Network structures, loss functions and parameter initialization. These three parts plays important roles in optimizing.

Dropout[8], have been proved to be useful in few-shot learning. They argue that most of features in the high dimension embedding of deep network are not useful for classification of novel class data, which may contain domain-specific information that may degenerate few-shot learning. Additive angular margin loss (arcface) is proposed to obtain high discriminant features for face recognition. Because Arcface is precisely corresponding to the geodesic distance on the hypersphere, it has a clear geometric interpretation. A large number of experiments show that Arcface is always better than SOTA, and is easy to implement, with negligible computing overhead. The loss for specifically:

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}.$$

where N is the sample size, m is the parameter that control the additive angular margin, and s is the parameter that improves backward gradient calculation. θ_j the the angle between W_j and the embedding.

Focal loss [6] on the other hand, have the form of:

$$\mathrm{L}_{fl} = \begin{cases} -\alpha\left(1-y'\right)^{\gamma}\log y', & y=1 \\ -(1-\alpha)y'^{\gamma}\log\left(1-y'\right), & y=0 \end{cases}$$

It neglect already "True" classification with high confidencey'. After all, we care nothing about these already true cases. We should focus on the samples that are really hard to classify, which means, the wrong classification with high y', or low y' even if we correctly classify the samples. This may happens in Oracle character classification, since some samples are really hard to classify.

Recognition Network Pre-training

From the view of Optimization, a good initialization starting point will determine whether the algorithm converges well. This should be emphasized on few-shot learning - since the loss function will be ugly, and there is high chance that the network converges into local minimum. Previous works design the pretrian method in a really complex way, including using RNN and LSTM[9]. We here simply use the imagenet pretrained parameters. Many tasks uses the parameters pre-trained on Imagenet, and achieve SOTA results. What's more, the infomation in object classification may well help recognize the Oracle characters, since this ancient writing itself comes from the abstract depiction of real things.

3 Method

3.1 Problem setup

The problem have three tasks, which aim to deal with data set with only one picture for learning, three and five the other data set. We will choose from 200 possible Oracle characters from the test set, which have all appeared in the train data set.

3.2 Orc-Bert data augmentation

We use the framework as Orc-Bert. First we pre-train the model over a large amount of unlabeled data by self-supervised learning. Then use this model as a augmentor to generate figures by recovering the masked training data under different mask probability.

We set the the structure of refinement network as 128-256-512-768 and embedding reconstruction network as 768-512-256-128-5. And we use a 8 layer transformer with multi-head 12 and MLP 768-3072-768. We train it on one NVIDIA Tesla T4 for 6 hours.

We first discretize the range of magnitudes [0.1,0.5] into 80 values as mask probabilities and use the Gaussian noise just like in Orc-Bert. But it's performance don't meet our expectations, as the recovered masked characters with high mask probability are very different from the origin ones. Then we try to magnitude [0.07,0.12] into 11 values and don't use the Gaussian noise, and in this way we get better results. Some results are shown in Figure 3. These pictures will also be transformed by our conventional data augmentation before training the Recognition Network.

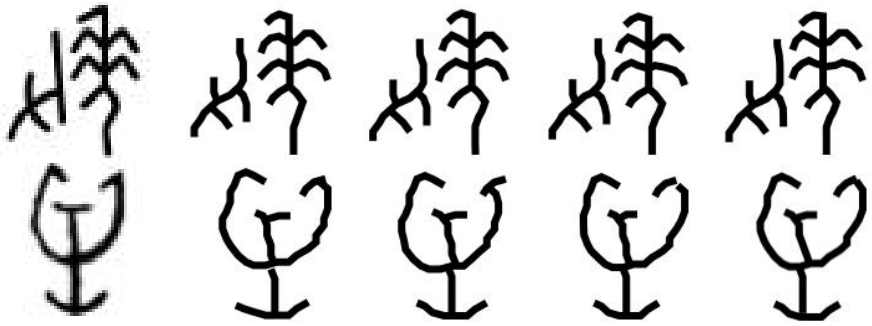

Figure 3: Origin oracle character and corresponding four recovered character with different mask probability

3.3 Conventional data augmentation

Our data augmentation process is described in the following pseudocode. We carefully preserved the original Oracle data, and then expanded the total amount of data to N times with our designed data augmentation method. Since the information in original pictures is really precious, so to preserve the data might be a wise choice.

And the transformation is described as following. At the first stage, we random enlarge the original picture without centering. Since the Resnet have the image size of 244, we then enlarge the picture to this size. Then, we random horizontally flip the pictures with possibility of 0.5. After that, we use RandAugment to transform the data. Finally, we use the Color jitter to recolor the pictures. The parameters are in Table 1.

Table 1: Traditional augmentation we use

Augmentation	Parameters
Random Resize Crop	size:224, scale:(0.8, 1.0), ratio:(1.0, 1.0)
Random Horizontal Flip	probability:0.5
RandAugment	options:2, magnitude:12
Color Jitter	brightness:0.2, contrast:0.2, saturation:0.2

Algorithm 1: Enlarging data set with Data Augmentation

Input: The image in the original data set `IMG`, Orc-bert transformed data set `ORC-IMG`, with each picture in `IMG` have m copies of generations, Enlarging parameter N, and transform function `T`.
Output: Data set `IMGN` with `IMG` enlarged for N times after data augmentation.

for Pictures in IMG **do**
    Append Pictures into IMGN
    for i in [2, 3, . . . , N] **do**
        Append T(Pictures) into IMGN
for Pictures in ORC-IMG **do**
    for i in [1, 2, 3, . . . , $\lceil N/m \rceil$] **do**
        Append T(Pictures) into IMGN
return IMGN

Figure 4: The first three lines describe the results of our data augmentation on ’’, while the last line is the actual data in test set. As we can see, the flip technique and the Orc-Bert can somehow intimate the true Oracle data.

The transformation results are visualized and compared to the results in test sets in Figure 4. Although the Orc-Bert cannot fully intimate the writings of the original ones, it somehow still diverse the shape of the input.

3.4 Combination of two methods

Since we have already enlarge the in Orc-Bert for m times, we should also enlarge the original data set by the same times as the Orc-Bert ones - just a consideration of the data balance. In experiments, we choose to enlarge the original data 10 times, with 11 Orc-bert data by 1 time.

3.5 Network structure and Loss function

First of all, as we have conclude in the related work, we use the parameters pretrained on Imagenet. Also, we use Dropout layer before the fully connect layer that used to calculate the distribution.

When it comes to the Loss function, here we try two different framework. The first is using common Cross Entropy Loss. The second is to use the ArcFace or Focal Loss. Since the ArcFace need the weight to calculate the similarity, we so, change the fully connect layer to the ArcFace. When testing, we just return the $\cos(\theta_j)$ as a description of similarity, and choose the one with the highest value. Focal loss is just a substitute of Cross Entropy Loss.

We want to see if the change of loss functions can improve the final result.

4 Experiments

We train the Resnet-18 network with Adam with L_2 decay 0.001 for 200 epochs under the learning rate:

$$Lr(x) = \begin{cases} x/80000 & x \leq 80 \\ 0.001 - 19(x-80)/16000000 & 80 \leq x \leq 160 \\ 0.001/20 - (x-80)/800000 & 160 \leq x \leq 200 \end{cases} \tag{1}$$

Complex may it seems at first sight, it is indeed a triangle like learning rate, which follows from the training process in CONV-Mixer [12]. Also, since gradient explosion may happens in the process of training, we utilize gradient clip to prevent the disaster from happening.

We train our network on GeForce RTX 2080Ti. The training process end in about 30 minutes. We will first list our results in tabel **??**, then conduct some ablation experiments on oure model, which can be found in 4 and 3.

4.1 Training Results

Our best training results is in Table 2, comparing to the results Orc-bert reports.

Table 2: Recognition accuracy (%) on Oracle-FS under all three few-shot settings. Some of the results use the data Orc-Bert [4] reported.

Setting	Model	No DA	Orc-Bert	Ours
1-shot	ResNet-18	18.6	31.9	**53.2**
	ResNet-50	16.8	29.9	**50.2**
	ResNet-152	14.0	27.3	**47.0**
	DenseNet	22.4	28.2	**47.6**
3-shot	ResNet-18	45.2	57.2	**75.0**
	ResNet-50	35.8	57.7	**71.7**
	ResNet-152	38.9	57.1	**70.1**
	DenseNet	48.6	58.3	**64.6**
5-shot	ResNet-18	60.8	68.2	**83.7**
	ResNet-50	55.6	67.9	**81.7**
	ResNet-152	58.6	67.8	**80.6**
	DenseNet	**69.3**	69.0	67.6

We can see our method is accurate, achieving large improvements over prior work on Oracle-FS.

4.2 Power of Enlarging Size

To compare different Data Augmentation method fairly, we design the Ablation experiment on one shot data, with dropout layers and L_2 decay 0.001 on Resnet-18. The results are listed in table 3. It deserves to notice that, **just enlarging the data set with data augmentation improves training.** We are pleased to see that the combination of two methods yields even better results.

Table 3: Baseline just doesn't add any data augmentation. As we can see, enlarging the data set with data augmentation method improves a lot

baseline	+ Conventional	+ Orc-Bert in [4]	+ Conventional×20	+ Copy×20	+ Combined
18.6	20.2	31.9	41.2	12.1	45.15

Compared to the traditional way of transform the data when training on "big data", here, for extremely small amount of cases, the distribution of data will be greatly influenced by the data augmentation.

At last, we may only learn the information in data augmentation, not in the data itself. Just fix the data augmentation will **ensure the influence of data augmentation on the distribution is under control**, and that the network will learn enough information from data itself.

Table 4: The ablation experiments results. Pretained parameters and dropout layers are the most important techniques in this task. What's more, the Combined data augmentation achieve the best results. It is strange that ArcFace loss have poor performance.

baseline	Only Bert	Only Conventional	-Pretrian	-Dropout	+ ArcFace	Enlarge 40
53.17	49.98	51.46	45.15	47.63	45.95	52.65

4.3 Comparison Between Different Augmentation Method

Three Ablation experiments on Network with pretrained parameters shed more light on the nature of the methods. We find out that under pretrained parameter, the three different methods will yields similar results, after controlling number of the samples. The combination of two method will increase the performance by 3%. As we may see in table 4.

Conventional augmentation with RandAugment will change the distribution of origin data too dramatically, while Orc-Bert, from our sight, will mildly change the shape of characters. Combination of the two method will force the network to focus both on overall character properties, as well as local stroke writing. So there is no doubt that the combination of two method will increase the accuracy.

However, the experiments also shows that, if there is limit of resources(both time and GPU resources), only enlarging data sets with Conventional augmentation will also leads to fair results. This convenient method show great power in the task. Maybe, other tasks will also be improved by this method.

4.4 Pretrained Parameters and Dropout Layer

When first experiment, we accidentally use the pretrained parameter of ResNet-18, unbelievably find out that, the accuracy improves to 38% form 17%. Further experiments shows that the method will always improves the accuracy: equipped with pretrained data will improve the final result to 53.17%, an enhancement of 8.02% in total.

Obviously, the pretrained parameter is a good initialization starting. But why? One reason is that the Oracle characters comes from abstraction of daily objects. Good classification parameters on 2000 objects contains enough information on Oracle character classification.

In this task, the Dropout Layer exhibit powerful strength on improving the accuracy. Without dropout, the accuracy drops to 47.63%, a 5% decrease comparing to the best one. This can be said to be an excellent proof of [8]

4.5 Why ArcFace and Focal Loss Fails

In the experiment, we find out that networks with ArcFace and Focal loss is really hard to train. For the first ten step, test accuracy will blow up to 50.5%, then suddenly drop to 40, and will never come back to 50 again. Testing accuracy proceeds to decreasing while training accuracy continuous to increasing. Over-fitting problem happens when using loss.

As far as we can know, the oracle characters differ from shape to shape even for the same concept. However, loss like Arcface tends to classify the same thing as far as possible away from the different ones. The low tolerance for differences may leads to difficulties in classifying the same character with really different shape. This task indeed is not like the Face recognition. A person's face may not change for years, but an Oracle character varies dramatically.

5 Conclusion

In this study, based on careful review of works on few shot learning, we propose that enlarging the data with augmentation will prevent network from only learning the distribution of data augmentation.

This method may also help other data augmentation method. As far as we have known, the Gan network are often rejected in the field of data augmentation since they may force the training network to learn the Gan's distribution. Just enlarge for a little copy may improve these methods a lot.

We further propose a combination of Orc-Bert and conventional data augmentation method, which both take the overall shape and local writing information into consideration. The method achieve SOTA results. Due to the lack of resources and time, the size of enlarging data, the M and N parameters in Rand Augmentation may be not the best ones. Follow-up researches may improve this method.

Also, we emphasize the importance of dropout layers and parameter initialization in this task. These two simple techniques will help network overcome the over-parameterized problem and converge to the minimum point more easily.

We then argue that ArcFace and Focal loss may not have the generalization ability in face of the changing shapes of Oracle characters. These two methods are really hard to train in this task. Follow-up research can study this problem further.

References

[1] Rowel Atienza. Data augmentation for scene text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1561–1570, October 2021.

[2] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. 2018.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. Number: arXiv:1810.04805.

[4] Wenhui Han, Xinlin Ren, Hangyu Lin, Yanwei Fu, and Xiangyang Xue. Self-supervised learning of orc-bert augmentor for recognizing few-shot oracle characters. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomas Pajdla, and Jianbo Shi, editors, *Computer Vision – ACCV 2020*, volume 12627, pages 652–668. Springer International Publishing. Series Title: Lecture Notes in Computer Science.

[5] Hangyu Lin, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Sketch-BERT: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. Number: arXiv:2005.09159.

[6] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, PP(99):2999–3007, 2017.

[7] Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. Learning a few-shot embedding model with contrastive learning. page 9.

[8] Chen Liu, Chengming Xu, Yikai Wang, Li Zhang, and Yanwei Fu. An embarrassingly simple baseline to one-shot learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4005–4009. IEEE.

[9] Tiago Ramalho and Marta Garnelo. Adaptive posterior learning: few-shot learning with a surprise-based memory module. Number: arXiv:1902.02527.

[10] ROOPAK, SHAH, EDUARD, SCKINGER, JAMES, W., BENTZ, ISABELLE, GUYON, and CLIFF. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 07(4):669–669, 1993.

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

[12] Asher Trockman and J Zico Kolter. Patches are all you need?, 2022.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[14] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. 53(3):1–34.

[15] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. Number: arXiv:1905.06549.