EASY with FFD: A Progressive Data Augmentation Approach for Few Shot Oracle Character Recognition

Zhao Xinyi
19307110452
Fudan University

Liu Siyuan
19307130018
Fudan University

Abstract

Recognition of the oracle character, the earliest known hieroglyphs in China, has been a hot topic in recent years, featuring hardship in the number of existing unraveled oracle images. The task of oracle character recognition can be interpreted as a few-shot image learning problem. Few-shot learning aims at leveraging knowledge learned by one or more deep learning models, to obtain good classification performance on new problems, where only a few labeled samples per class are available. Recent years have seen a large number of works in the field, introducing methods with numerous ingredients. A frequent problem, though, is the use of efficient data augmentation methods to boost classification. In this work, we propose to implement the free form deformation method which is common in the registration region to help generate augmented deformed data to aid our training. We also use a new methodology called EASY which reaches or even beats state-of-the-art performance on multiple standardized benchmarks of the few-shot image classification field. Extensive experiments on few-shot learning demonstrate the effectiveness of our free form deformation augmentor in improving the performance in the few-shot oracle character recognition.

1 Introduction

Oracle characters are the earliest known pictographs in China, which were carved on animal bones or tortoise shells. The purpose of an oracle was to make fiery divinations about the weather, state power, war, and trade to ease uncertainty. Due to the scarcity of oracle bones and the long tail problem in the use of Chinese characters in the Shang Dynasty, oracle character recognition suffers from data limitations and imbalances. The sample of characters is very limited, so oracle character recognition is a natural few-shot learning problem for researchers. In addition, there is a high degree of intra-class variance in the shapes of oracle characters, since oracle bones were made from different ancient people in various regions thousands of years ago. Therefore, oracle character recognition/classification has been a challenging task troubling many researchers.

While the identification and deciphering of oracle bones have come a long way in the past few decades, there is still a long way to go to fully understand the entire writing system. There is still a large scale of unmarked/undeciphered oracle characters. Therefore, unlike the standard few-shot learning setting, this task does not assume the existence of large-scale labeled source oracle characters. Formally, you must learn in a more realistic setting where your model has access to large-scale unlabeled source oracle characters to efficiently identify new oracle characters that contain a small number of labeled training samples. Supervised methods obtain good performance but they require a large amount of labeled data. Contrary, unsupervised or learning-free methods can be applied when labeled data is not available, but they lead to lower performance. Thus, to maintain good performance while reducing the manual annotation, few-shot learning for oracle character recognition seems preferable, since it reaches a performance similar to supervised methods but requires few annotated text lines.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

In this paper, we integrate the ideas of self-supervised learning in data augmentation and propose a new data augmentation method, referring free form deformation method in the registration field. First, we utilize the online Free Form Deformation algorithm as our augmentor and generate n augmented data for each annotated training instance for each category. Then, we conduct a backbone training using random resized crops, random color jitters, and random horizontal flips. We propose to generate augmented feature vectors for each sample from the validation and novel dataset and use two transforms to concatenate them with random seeds. We train our model using a nearest class mean classifier.

The main contributions of our work are as follows:

- We introduce a few-shot learning model for recognizing oracle characters in low-resource scenarios with minimal human effort.
- To the best of our knowledge, we are the first to apply the non-rigid transformation, namely FFD, to the field of data augmentation based on the construction knowledge of Chinese characters.
- We demonstrate the effectiveness of our approach through comprehensive experiments on Oracele_fs datasets, reaching a significantly higher accuracy compared with other baseline approaches.

2 Related Work

2.1 Oracle Character Recognition

As technologies evolve, a handful of works have begun to concentrate on oracle character recognition from the perspective of computer vision. Earliest researches (5),(6),(7),(8) regarded oracle character as an undirected graph, and utilized its topological properties as features to perform classification. Guo et al. (10) collected an Oracle-20K dataset which composes of 20K handprinted oracle characters belonging to 261 categories and constructed a novel hierarchical representation.

Recently, CNNs have achieved great progress in some computer vision tasks and are introduced into oracle character recognition. Huang et al. (11) released a dataset of scanned oracle characters called OBC306 and presented the CNN-based evaluation for this dataset to serve as a benchmark. OracleNet (12) considered the radical-level composition of oracle characters and detected radicals using a Capsule network to recognize characters. Cross-modal oracle character recognition was first proposed by Zhang et al. (13). They utilized adversarial learning and cross-modal triplet loss to perform classification on handprinted and scanned data in a supervised manner. However, our method focuses on the background where most labels of scanned data are unavailable which is a more realistic but difficult scenario.

2.2 Few-shot Learning Techniques (FSL)

A great number of works aim to improve the performance of the few-shot training process. Recent substantial progress is based on the meta-learning paradigm(also called learning-to-learn) with multi-auxiliary tasks and data augmentation. (14)(15)(16)(17). The key is how to robustly accelerate the learning progress of the network without suffering from over-fitting with limited training data. Finn et al. propose MAML (15) to search for the best initial weights through gradient descent for network training, making the fine-tuning easier. Chen (18) proposes a generalized embedding network with self-supervised learning (SSL) which can provide robust representation for downstream tasks by learning from data. Rusu et al. (19) propose a network called LEO to learn a low dimension latent embedding of the model to reduce the complexity. Then the samples can be classified by the nearest neighbor (NN) criterion using a pre-defined distance function.

A deeper embedding backbone has also been tried. Chen et al. (20) propose a data augmentation method to cope with the over-fitting issue with a deeper backbone embedding network. In (21), Hariharan et al. indicate a similar observation, as claimed in this paper, that with a deeper backbone(ResNet-50) embedding is not only costly but also not effective.

2.3 Data Augmentation Approaches

Data augmentation is an effective approach to enlarge training data and boost the overall ability of the models. Different variants of data augmentation bring more disturbance to the model while helping to better capture the invariant features hidden behind. For FSL, some efficient augmentation algorithms cannot be applied to our setting (without large-scale labeled data), like AutoAugment (29) or any semi-supervised learning work. This makes data augmentation a challenging topic in the field of FSL.

On data augmentation for image classification, current research mainly focuses on rigid transformation, such as translation (36), flipping (37), shearing (38), cropping (37) and rotation (39). Complex methods, such as occluding parts of an image (23) or blending images (24) are also proved to be effective but they are highly based on expert domain knowledge. Little work on non-rigid transformation has been studied, among which Søren et al. (40) learned the spatial transformation between images in the same class and applied the learned model to data augmentation under the assumption that the latent spatial transformation between images in the same class is a diffeomorphism. But their image-registration-based model requires a huge amount of training and cannot be applied to the complex and low-shot dataset.

Distinguished from the traditional image, the character image is regarded as an intermediate between text and image. As an image, it retains many textual features, such as the ability to be reconstructed into a sequence of sketches. Han et al. (31) captured the stroke orders information of Chinese characters and utilize the pretrained Sketch-Bert model to augment few-shot labeled oracle characters. However, none of the state-of-art data augmentation approaches addressed the problem of Chinese characters recognition from the overall structure. To be more specific, most Chinese characters are combining characters, i.e. they are made up of radicals. Therefore, simply introducing noise to strokes cannot enhance the model's ability to recognize radical components.

2.4 Non-Rigid Transformation

Non-Rigid Transformation is a critical technique in the field of image registration, which aims at finding the optimal transformation that best aligns structures in two input (2D) or volume data (3D) images (42). Particularly in the medical field, extensive researches have been conducted on non-Rigid transformation for its important role in the analysis of medical effects over time. Existing non-rigid transformation includes AIR, Diffeomorphic Demons, and FFD. (43)

Free Form Deformation (FFD) is a commonly used algorithm for elastic image registration (41). The registration from moving image to fixed image is modeled by the combined motion of a global affine transformation and a free-form deformation based on B-splines. Compared with rigid or affine registration techniques, free form deformation provides a high degree of flexibility to model the motion and can significantly reduce motion artifacts between images.

3 Methodology

We formally define the problem of oracle recognition under few-shot settings and then give an overview of our overall framework followed by detailed illustrations.

3.1 Problem Formulation

We intend to train a model capable of fully understanding the overall structure of certain oracle characters from one or a few samples.

Different from the problem formulation of Han et al. (31), we do not use external data for data augmentation or data classification, whether labeled or not. Our Augmentor and classifier would have only access to k annotated training instances for each category and then be tested on 20 instances per class, namely, Oracle-FS.

3.2 Overview of Framework

As is shown in Fig. 1, our framework consists of several parts.

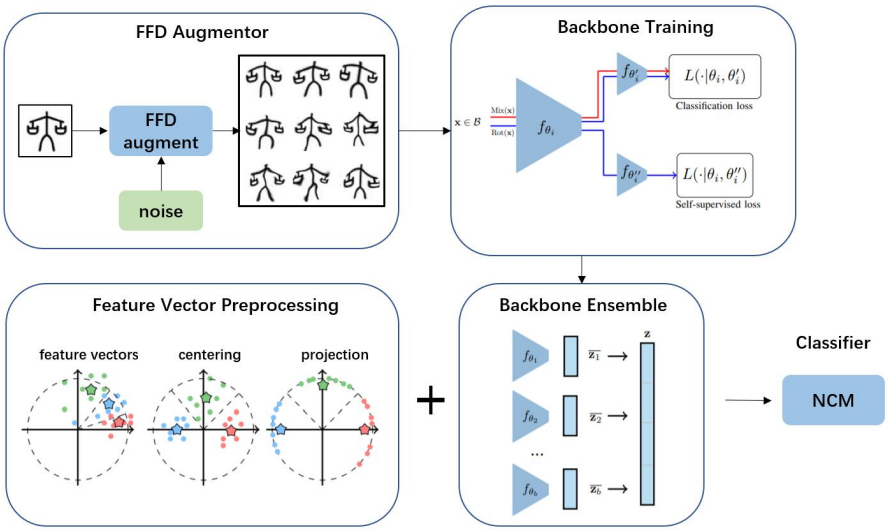

Figure 1: Illustration of our proposed method.

First, we utilize the online Free Form Deformation algorithm as our Augmentor and generate n augmented data for each annotated training instance for each category.

Then, we conduct a backbone training using random resized crops, random color jitters and random horizontal flips. We propose to generate augmented feature vectors for each sample from the validation and novel dataset and use two transforms to concatenate them with random seeds.

Finally, we train our model using nearest class mean classifier with the generated feature vectors.

3.3 FFD Augmentor

Oracle Character Chinese characters can be classified into two categories, single characters, and combined characters. Single characters have an independent structure, while combined characters are composed of radicals and single characters, by the left and right combination, top and bottom combination, and others. These radicals and structures determine the pronunciation and meaning of Chinese characters. In a famous antiquarian book called *Shuowen Jiezi*, Xu Shen of the Eastern Han Dynasty summarised the laws of Chinese character construction as the **Six books**: Pictographs, Referents, Metonymy, Morphology, Transcription, and Pseudo-pronunciation. Among them, the four principles of character formation, namely Pictographs, Referents, Metonymy, Morphology, are the *Method of Character formation*, while Transcription and Pseudo-rotation are the laws of character use, the *Method of Using Characters*. It is by grasping these construction methods that we humans learn new words.

As the origin of Chinese characters, the oracle characters should also follow these construction methods. Therefore, to help train the model better understand oracles' structured features, we perform data augmentation employing non-rigid transformation. Non-rigid transformations can greatly destroy partial information, like specific strokes, while retaining the overall structure. (See Figure 2)

Free Form Deformation We use the classical non-rigid transformation algorithm, namely a Free Form Deformation based on B-splines. The algorithm is a grid-based non-rigid deformation model, which distributes a series of grid points over the image at a certain spacing. The algorithm uses the position of the grid points to calculate the coordinate offset of each pixel, and finally re-samples the image by pixels according to the coordinate offset to achieve non-rigid deformation.

Augmentor Algorithm 1 illustrates the detailed pseudo-code for our data augmentor. The algorithm requires the training image img, block num n, max offset O, and the size of BPLINE_BOARD. The algorithm returns the augmented image after FFD transformation. The reasons for the selection of these hyperparameters will be discussed further in Section 4.3. The algorithm will first initialize all the grid points and randomly set their offset. Then, for each pixel within the image, the algorithm

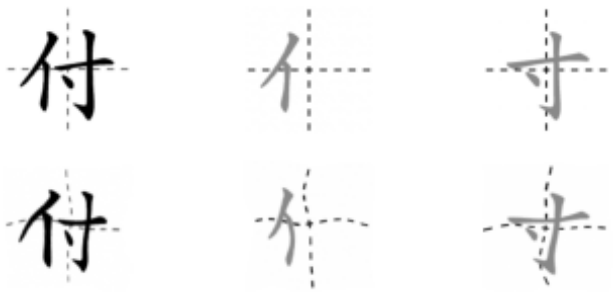

Figure 2: The Chinese character 'Fu' is made up of two parts, a 'People' radical and the single character 'Inch'. After the FFD transformation, though the strokes are distorted, the overall structure remains unchanged. We can still recognize what the two part means and combine them to re-recognize the character 'Fu'.

calculates its coordinate offset based on the 4 gird points in its vicinity. If the pixel transformed coordinates exceed the shape of the image, the gray scale value will be replaced by 255.

3.4 Backbone Training

Following the Ensemble Augmented-Shot Learning method (34) there are several steps in this stage.

First, We use data augmentation with random resized crops, random color jitters, and random horizontal flips, which is standard in the field. We use a cosine-annealing scheduler, where at each step the learning rate is updated. During a cosine cycle, the learning rate evolves between η_0 and 0. At the end of the cycle, we warm-restart the learning procedure and start over with a diminished η_0. We start with $\eta_0 = 0.05$ and reduce η_0 by 10% at each cycle.

We train our backbones by taking a standard classification architecture (e.g., ResNet18 (32)), and branching a new logistic regression classifier after the penultimate layer, in addition to the one used to identify the classes of input samples, thus forming a Y-shaped model 1. This new classifier is meant to retrieve which one of four possible rotations (quarters of 360° turns) has been applied to the input samples. We use a two-step forward-backward pass at each step, where the first batch of inputs is only fed to the first classifier, combined with manifold-mixup (33). The second batch of inputs is then applied to arbitrary rotations and fed to both classifiers.

We experiment using a standard ResNet18 as described in (32), where the feature vectors are of dimension 640. These feature vectors are obtained by computing a global average pooling over the output of the last convolution layer. Such a backbone contains 700k trainable parameters.

3.5 Classification

We generate augmented feature vectors for each sample from the test datasets. To this end, we use random resized crops from the corresponding image. We concatenate the feature vectors obtained from multiple backbones trained using the same previously described routine, but with different random seeds. To perform fair comparisons, when comparing a backbone with an ensemble of backbones, we reduce the number of parameters in the ensemble backbones to make the total number of parameters remain identical.

In the case of inductive few-shot learning, we use a simple Nearest Class Mean classifier (NCM). let $z = f(x)$ denote the feature vector associated with x. Predictions are obtained by first computing class barycenters from labeled samples:

$$\forall i : \overline{\mathbf{c}_i} = \frac{1}{|\mathcal{S}_i|} \sum_{\mathbf{z} \in \mathcal{S}_i} \mathbf{z} \tag{1}$$

Then find the closest barycenter for each query:

$$\forall \mathbf{z} \in \mathcal{Q} : C_{ind}\left(\mathbf{z}, [\overline{\mathbf{c}_1}, \ldots, \overline{\mathbf{c}_n}]\right) = \arg\min_i \left\|\mathbf{z} - \overline{\mathbf{c}_i}\right\|_2 \tag{2}$$

4 Experiments

We conduct extensive experiments and show that the model under our new augmented-shot settings outperforms other baseline methods employing different classification networks. Then we confirm this result on ablation studies about the volume of augmented and pre-training data and mask probability.

4.1 Experimental Settings

Dataset We used Oracle-FS training images to train the model and evaluate it on test data of Oracle-FS. Under the k-shot setting, there are 200 classes, with k training instances and 20 test ones per class. In this paper, we set k=1, 3, 5.

	k-shot	Num of instances per class		Num of classes
		Train	Test	
oracle-FS	1	1	20	200
	3	3	20	200
	5	5	20	200

Table 1: Statistics of Oracle-FS, including number of instances and number of classes

Classifiers In this paper, we adopt 4 representative CNN models with various depths and widths, including ResNet18, ResNet20, WideResNet, and ResNet12, to show the effectiveness of our Augmentor in boosting the classification performance of conventional networks. In the inductive setting, we use basic Nearest Class Mean (NCM) classifier.

Implementation Details We implement FFD Augmentor and backbone training methods using PyTorch. In detail, the number of training epochs is 200 with a batch size of 64. We adopt Adam as the optimizer with a cosine scheduled learning rate of 0.1 for classifier training and Augmentor pre-training, respectively. All images are resized to 50×50 before online approximation. Augmented images generated by FFD Augmentor are also 50×50. We initialize the number of crops as 10 in the augmented sampling section. We divided the num of the feature map by 2, with smaller feature vectors of dimensions.

4.2 EASY Backbone Training Evaluation

We first report the results comparing the EASY framework with no preprocessing baseline method. Preprocessing session includes ingredients like mixup and rotations to train the backbones. Meanwhile, EASY also provides preprocess sequences for few-shots feature vectors, like relu, sqrt, sphering, and centering. In the paper we use relu, sphering, and centering to project features on the hypersphere, the abbreviation being PEME. The experiment results are shown in Table 2.

As illustrated from the table, we can find that there is a significant improvement in accuracy with backbone preprocessing and feature vector preprocessing methods. The more parameters of the model, the more obvious the accuracy improved. However, for Resnet12, the accuracy dropped. We find that Resnet12 contains too few parameters for large-size images Therefore, when cut into many backbones and ensembled, it cannot reflect the original nature of images.

4.3 FFD Augmentation Analysis

In this part, we conduct more experiments to evaluate our FFD augmentor, including the min and max of offset, number of blocks, and augmented samples. Note that, in these experiments, we employ ResNet18 as the classifier and perform experiments only under the 1-shot scenario with 100 epochs and a 0.1 learning rate. To make our results more accurate, all experiments on hyperparameters are conducted twice and we average the results of the two experiments as the final accuracy.

Max Offset Value In our 2D FFD augmentor, the random offset value was generated through a uniform distribution in the interval. The maximum values limit the movement range of the offset. The closer the value is to 0, the smaller the deformation of the image. In our experiment, we tested the maximum value from 0 to 15. For max offset value equals 0, we mean that no free form transformation is performed on the original image.

Table 2: EASY With preprocessing: mixup+rotation+PEME

k-shot	Model	No Preprocess	With Preprocess
1	ResNet18	47.00	**55.17**
	ResNet20	49.26	**54.67**
	WideResnet	49.91	**53.77**
	Resnet12	**62.1**	58.46
3	ResNet18	78.71	**79.45**
	ResNet20	78.29	**82.53**
	WideResnet	81.79	**84.81**
	Resnet12	**88.34**	84.91
5	ResNet18	89.36	**90.34**
	ResNet20	90.58	**91.55**
	WideResnet	92.30	**92.43**
	Resnet12	**92.42**	91.30

Num of Blocks Num of blocks influences the control points of the grid. The more blocks in the FFD transformation, the more transformation control points there are, and the more complex the deformation can be. In our realization, we test the number of blocks from 3, 5 to 7. The time of generating the augmented dataset is proportional to the square of block nums. When the number of blocks increases to a large amount, the enhancement effect of the picture is not obvious but it will take considerable time which is much longer than the training cost.

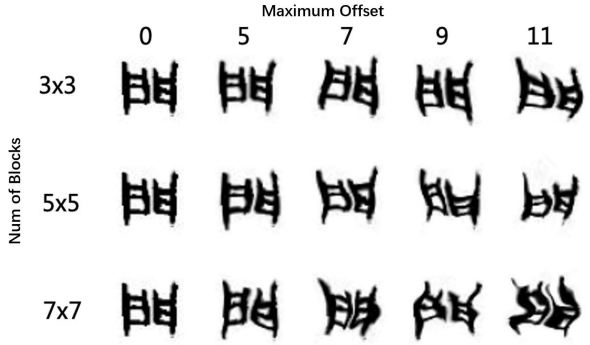

Figure 3: Illustration of num of blocks and maximum offset difference. When the num of blocks is small, the deformation is closer to rigid transformation and the overall shape of the text remains unchanged. But when the num of blocks and maximum offset value becomes too large, the deformation is so complex that the original strokes are distorted and hard to identify.

We trade off both max offset value and num of Blocks. The results are shown in Figure 4. With the rise of the max offset value, the accuracy of all block nums increases. But when the offset exceeds a certain threshold, their accuracy all begins to decrease. As can be seen, the top three accuracy combinations of block Num and max offset is $(3, 11)$, $(5, 15)$ and $(7, 11)$.

Num of Augmented Sample We then conducted an experiment on whether the larger num of augmented sample contributes to better performance of our model. By the num of augmented sample , we mean generate num of augmented images for each training image. Intuitively, with more augmented samples, the accuracy will be higher. However, the results of experiments (See Figure 4) shows that due to the limited number of samples, too many augmented images will lead to overfitting, i.e. the test accuracy will become lower with the decrease of training loss. Except block num of 3, all FFD combinations show an increasing trend followed by a decreasing trend.

Besides, the computation time of data augmentation is also a crucial factor to be considered. Our FFD algorithm takes about 0.4 to 0.5s to generate each image. Due to the expensive time cost for data augmentation, we trade off both performance and computation time. Combining all the results above, we find that with 5 block num, 11 max offset and 30 augment samples helps our model achieve the best performance of 78.9% in 1-shot. Our subsequent experiments were also based on this hyperparameter setting.

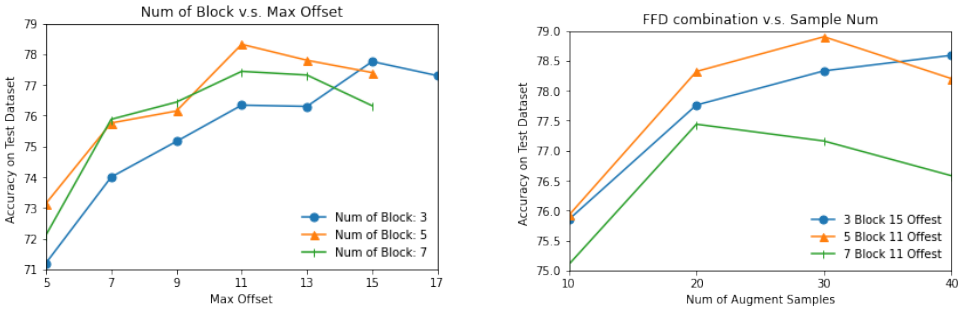

Figure 4: **left:** Different combination of Block numbers and max offsets varies in accuracy. For block num of 3 and 7, the threshold of max offset is 11. For block num of 5, the threshold of max offset is 15. **right:** Num of Augmented Sample influences the model accuracy. When the number of augmented sample equals to 30, the FFD combination of 5 block num and 11 max offset reaches the maximum accuracy.

4.4 FFD Augmented Training Evaluation

We conducted experiments under the best hyperparameters settings of FFD previously analyzed. For 1-shot and 3-shot, we used a learning rate of 0.1 while we used a learning rate of 0.01 for 5-shot. It can be clearly noticed that our FFD-based data augmentation can effectively improve the accuracy of all the models. To be more precise, our 1-shot accuracy reaches 76.5% for all the models, which outperforms the previous models by 20%. Our 3-shot accuracies achieved 93.42%, exceeding the accuracy of all the 5-shot models without FFD augmentation. Finally, on WideResNet, our model reaches an astonishing accuracy of **97.59%** for 5-shot.

Table 3: EASY With preprocessing and FFD

k-shot	model	EASY	FFD Augmentor+EASY
1	ResNet18	55.17	**78.90**
	ResNet20	54.67	**77.09**
	WideResNet	53.77	**77.75**
	ResNet12	58.46	**76.79**
3	ResNet18	79.45	**93.36**
	ResNet20	82.53	**92.57**
	WideResNet	84.81	**93.42**
	ResNet12	84.91	**92.89**
5	ResNet18	90.34	**96.47**
	ResNet20	91.55	**95.26**
	WideResNet	92.43	**97.59**
	ResNet12	91.30	**95.38**

5 Conclusion

In this paper, we intend to address the novel task for oracle character recognition, with a few labeled training samples. We propose a new data augmentation tool for few-shot oracle recognition problems referring to the free form deformation method commonly used in registration experiments. It generates a series of augmented images by random FFD on the original images for the classifier for training. FFD augmentor is simple yet effective, readers can easily reproduce it by referring to the pseudo-code. Extensive experiments under three few-shot settings confirm the effectiveness of our FFD Augmentor to improve the performance of various networks on few-shot oracle character recognition. Meanwhile, we extended the application of EASY (34), fine-tuned the parameters and proved that EASY also has the potential of high accuracy in oracle character recognition through experiments. Our model has broad prospects in the field of written character recognition field.

References

[1] Yalniz, I. Z., Jégou, H., Chen, K., Paluri, M., Mahajan, D. (2019). Billion-scale semi-supervised learning for image classification. CoRR, abs/1905.00546. http://arxiv.org/abs/1905.00546.

[2] Souibgui, M. A., Fornés, A., Kessentini, Y., Megyesi, B. (2021). Few Shots Is All You Need: A Progressive Few Shot Learning Approach for Low Resource Handwriting Recognition. CoRR, abs/2107.10064. https://arxiv.org/abs/2107.10064.

[3] Lowe D G. Distinctive image features from scale-invariant keypoints, International journal of computer vision, 2004, 60(2): 91-110.

[4] Philbin J, Chum O, Isard M, et al. Lost in quantization: Improving particular object retrieval in large scale image databases, Computer Vision and Pattern Recognition, 2008. CVPR 2008, IEEE Conference on, IEEE, 2008: 1-8.

[5] X.-L. Zhou, X.-C. Hua, and F. Li, A method of jia gu wen recognition based on a two-level classification. in Proceedings of Internationa Conference on Document Analysis and Recognition, 1995, pp. 833–836.

[6] F. Li and P.-y. Woo, The coding principle and method for automatic recognition of jia gu wen characters, International Journal of HumanComputer Studies, vol. 53, pp. 289–299, 2000.

[7] Q. Li, Y. Yang, and A. Wang, Recognition of inscriptions on bones or tortoise shells based on graph isomorphism, Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications), vol. 47, pp. 112– 114, 2011.

[8] S. Gu, "Identification of oracle-bone script fonts based on topological registration," Computer and Digital Engineering, vol. 44, pp. 2001–2006, 2016.

[9] Wang, M., Deng, W., Liu, C. L. (2022). Unsupervised Structure-Texture Separation Network for Oracle Character Recognition. IEEE Transactions on Image Processing, 31, 3137-3150.

[10] J. Guo, C. Wang, E. Roman-Rangel, H. Chao, and Y. Rui, Building hierarchical representations for oracle character and sketch recognition, IEEE Transactions on Image Processing, vol. 25, pp. 104–118, 2015.

[11] S. Huang, H. Wang, Y. Liu, X. Shi, and L. Jin, "OBC306: A largescale oracle bone character recognition dataset," in Proceedings of International Conference on Document Analysis and Recognition, 2014, pp. 1–14.

[12] X. Lu, H. Cai, and L. Lin, Recognition of oracle radical based on the capsule network, CAAI transactions on intelligent systems, vol. 15, pp. 243–254, 2020.

[13] Y.-K. Zhang, H. Zhang, Y.-G. Liu, and C.-L. Liu, Oracle character recognition based on cross-modal deep metric learning, Acta Automatica Sinica, vol. 47, pp. 791–800, 2021.

[14] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al., Matching networks for one shot learning, in NeurIPS, 2016, pp. 3630–3638.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine, Modelagnostic meta-learning for fast adaptation of deep networks, in ICML. JMLR, 2017, pp. 1126–1135.

[16] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales, Learning to compare: Relation network for few-shot learning, in CVPR, 2018.

[17] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille, Fewshot image recognition by predicting parameters from activations, in CVPR, 2018, pp. 7229–7238.

[18] Chen, D., Chen, Y., Li, Y., Mao, F., He, Y., Xue, H. (2019). Self-Supervised Learning For Few-Shot Image Classification. CoRR, abs/1911.06045. http://arxiv.org/abs/1911.06045.

[19] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell, Meta-learning with latent embedding optimization, in ICLR,2019.

[20] Zitian Chen, Yanwei Fu, Kaiyu Chen, and Yu-Gang Jiang, "Image block augmentation for one-shot learning.," in AAAI, 2019.

[21] Bharath Hariharan and Ross Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in ICCV, 2017, pp. 3018–3027.

[22] Luo, C., Zhu, Y., Jin, L., Wang, Y. (2020). Learn to augment: Joint data augmentation and network optimization for text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13746-13755).

[23] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

[24] J. Lemley, S. Bazrafkan, and P. Corcoran. Smart augmentation learning an optimal data augmentation strategy. IEEE Access, 5:5858–5869, 2017.

[25] M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

[26] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In International Conference on Learning Representations (ICLR), 2016.

[27] A. Antoniou, A. Storkey, and H. Edwards. Augmenting image classifiers using data augmentation generative adversarial networks. In International Conference on Artificial Neural Networks, 2018.

[28] L. Chongxuan, T. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. In Advances in Neural Information Processing Systems (NeurIPS), 2017

[29] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 113–123, 2019.

[30] Martin Mayr, Martin Stumpf, Anguelos Nikolaou, Mathias Seuret, Andreas Maier, and Vincent Christlein. Spatio-temporal handwriting imitation. arXiv preprint arXiv:2003.10593, 2020.

[31] 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 ACCV, 2020.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-Decem, 2016, pp. 770–778.

[33] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. 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, 2020, pp. 2218–2227.

[34] Bendou, Y., Hu, Y., Lafargue, R., Lioi, G., Pasdeloup, B., Pateux, S., Gripon, V. (2022). EASY: Ensemble Augmented-Shot Y-shaped Learning: State-Of-The-Art Few-Shot Classification with Simple Ingredients. CoRR, abs/2201.09699. https://arxiv.org/abs/2201.09699

[35] Lin, H., Fu, Y., Xue, X., Jiang, Y. G. (2020). Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6758-6767).

[36] Benaim, S., Wolf, L. (2018). One-shot unsupervised cross domain translation. advances in neural information processing systems, 31.

[37] Qi, H., Brown, M., Lowe, D. G. (2018). Low-shot learning with imprinted weights. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5822-5830).

[38] Shyam, P., Gupta, S., Dukkipati, A. (2017, July). Attentive recurrent comparators. In International Conference on Machine Learning (pp. 3173-3181). PMLR.

[39] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T. (2016, June). Meta-learning with memory-augmented neural networks. In International conference on machine learning (pp. 1842-1850). PMLR.

[40] Hauberg, S., Freifeld, O., Larsen, A. B. L., Fisher, J., Hansen, L. (2016, May). Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In Artificial Intelligence and Statistics (pp. 342-350). PMLR.

[41] Rueckert, D., Sonoda, L. I., Hayes, C., Hill, D. L., Leach, M. O., Hawkes, D. J. (1999). Nonrigid registration using free-form deformations: application to breast MR images. IEEE transactions on medical imaging, 18(8), 712-721.

[42] Keszei, A. P., Berkels, B., Deserno, T. M. (2017). Survey of non-rigid registration tools in medicine. Journal of digital imaging, 30(1), 102-116.

[43] Oliveira, F. P., Tavares, J. M. R. (2014). Medical image registration: a review. Computer methods in biomechanics and biomedical engineering, 17(2), 73-93.

Appendices

A Experiments with More Settings

Num of Learning Rate. Different learning rates affect the convergence speed and accuracy of the model. A low learning rate may cause the model to overfit the training dataset or converge too slowly. A high learning rate may prevent the model from convergence. We made experiments on the influence of different learning rate under different k-Shot setting. As shown in Table 5, for k=5, learning rate of 0.01 reaches the highest accuracy.

Num of Samples Using Random Crop In our experiments, we also test the number of image samples processed by random crop and flip before backbone training. The results show that the accuracy rate is highest when the size is 10, and there is a risk of overfitting when the size is larger.

Table 4: Top-1 Accuracy under different learning rate.(k-shot=5)

Learning rate	0.1	0.05	0.01	0.005	0.001
Accuracy	95.82	96.31	**96.47**	89.08	87.05

Table 5: Top-1 Accuracy under different support samples with no FFD augmentation.(k-shot=1)

Aug-sample	1	5	10	20	40
Accuracy	52.94	53.39	**53.52**	53.40	53.36

Raw Data of Figure 4

Table 6: Top-1 Accuracy for different combination of Block Numbers and Max Offset

Max Offset/Block_Num	**3**	**5**	**7**
7	74.0	75.76	75.88
9	75.16	76.15	76.44
11	76.34	78.32	77.44
13	76.3	77.8	77.32
15	77.76	77.4	76.32
17	77.3	-	-
19	77.21	-	-

Table 7: Top-1 Accuracy for different combinations and Num of Augmented samples

Combination/ Augmented Num	**10**	**20**	**30**	**40**
3 Block Nums - 15 Max Offset	75.85	77.76	78.33	78.59
5 Block Nums - 11 Max Offset	75.92	78.32	78.90	78.2
7 Block Nums - 11 Max Offset	75.10	77.44	77.16	76.58

B Thesis

Algorithm 1 FFD Augmentor

Require: Training Image img, block Num n, Max Offset O, BPLINE_BOARD_SIZE.
Ensure: Augmented Image.

$GridPoints$ = InitParam(n,n,$-O$,O)
$out \leftarrow$ BsplineFfdKernel(img,n,n,$GridPoints$)
return out
function INITPARAM(n,n,$minO$,$maxO$)
    $GridRows \leftarrow n+$ BPLINE_BOARD_SIZE
    $GridCols \leftarrow n+$ BPLINE_BOARD_SIZE
    $GridSize \leftarrow GridRows * GridCols$
    $GridPoints \leftarrow (maxO - minO) *$ np.random.random_sample($2 * GridSize$) +$minO$
    return $GridPoints$
function BSPLINEFFDKERNEL(img,Col_n,Row_n,$GridPoints$)
    row, col = img.shape
    Δ_x, $\Delta_y \leftarrow col/Col\_n$, row/Row_n
    $GridRows \leftarrow Row\_n$ + BPLINE_BOARD_SIZE
    $GridCols \leftarrow Col\_n$ + BPLINE_BOARD_SIZE
    $GridSize \leftarrow GridRows * GridCols$
    $out \leftarrow$ np.zeros_like(img)
    for y in range(row) **do**
        for x in range(col) **do**
            x_block,$y\_block \leftarrow x/\Delta_x$, y/Δ_y
            i, $j \leftarrow$ np.floor(y_block), np.floor(x_block)
            u, $v \leftarrow x\_block - j$, $y\_block - i$
            $pX, pY \leftarrow$ np.zeros((4)), np.zeros((4))
            $pX[0] \leftarrow (1 - u^3 + 3 * u^2 - 3 * u)/6$
            $pX[1] \leftarrow (4 + 3 * u^3 - 6 * u^2)/6$
            $pX[2] \leftarrow (1 - 3 * u^3 + 3 * u^2 + 3 * u)/6$
            $pX[3] \leftarrow u^3/6$
            $pY[0] \leftarrow (1 - v^3 + 3 * v^2 - 3 * v)/6$
            $pY[1] \leftarrow (4 + 3 * v^3 - 6 * v^2)/6$
            $pY[2] \leftarrow (1 - 3 * v^3 + 3 * v^2 + 3 * v)/6$
            $pY[3] \leftarrow v^3/6$
            $T_x, T_y \leftarrow x, y$
            for m in range(4) **do**
                for n in range(4) **do**
                    $CP_x, CP_y \leftarrow j + n, i + m$
                    $temp \leftarrow pY[m] * pX[n]$
                    $T_x \leftarrow T_x + temp * GridPoints[np.int64(CP_y * GridCols + CP_x)]$
                    $T_y \leftarrow T_y + temp * GridPoints[np.int64(CP_y * GridCols + CP_x + GridSize)]$
                    x_1, $y_1 \leftarrow$ np.int64(T_x), np.int64(T_y)
            if ($x_1 < 1$ or $x_1 \leq col - 1$ or $y_1 < 1$ or $y1 \geq row - 1$) **then**
                $out[y, x] \leftarrow 255$
            else
                $x_2, y_2 \leftarrow x_1 + 1, y_1 + 1$
                $out[y, x] \leftarrow (x_2 - T_x) * (y_2 - T_y) * img[y_1, x_1] - (x_1 - T_x) * (y_2 - T_y) * img[y_1, x_2] - (x_2 - T_x) * (y_1 - T_y) * img[y_2, x_1] + (x_1 - T_x) * (y_1 - T_y) * img[y_2, x_2]$
    return out