Self-supervised Learning of Orc-Bert Augmentor for Recognizing Few-Shot Oracle Characters

Wenhui Han[0000−0001−5896−3692], Xinlin Ren[0000−0002−8175−7392], Hangyu Lin[0000−0002−0538−7692], Yanwei Fu⋆[0000−0002−6595−6893], and Xiangyang Xue[0000−0002−4897−9209]

School of Data Science, Computer Science, and MOE Frontiers Center for Brain Science, Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai, China
{19210980110, 20110240015, 18210980008, yanweifu, xyxue}@fudan.edu.cn

Abstract. This paper studies the recognition of oracle character, the earliest known hieroglyphs in China. Essentially, oracle character recognition suffers from the problem of data limitation and imbalance. Recognizing the oracle characters of extremely limited samples, naturally, should be taken as the few-shot learning task. Different from the standard few-shot learning setting, our model has only access to large-scale *unlabeled* source Chinese characters and few labeled oracle characters. In such a setting, meta-based or metric-based few-shot methods are failed to be efficiently trained on source unlabeled data; and thus the only possible methodologies are self-supervised learning and data augmentation. Unfortunately, the conventional geometric augmentation always performs the same global transformations to all samples in pixel format, without considering the diversity of each part within a sample. Moreover, to the best of our knowledge, there is no effective self-supervised learning method for few-shot learning. To this end, this paper integrates the idea of self-supervised learning in data augmentation. And we propose a novel data augmentation approach, named Orc-Bert Augmentor pre-trained by self-supervised learning, for few-shot oracle character recognition. Specifically, Orc-Bert Augmentor leverages a self-supervised BERT model pre-trained on large unlabeled Chinese characters datasets to generate sample-wise augmented samples. Given a masked input in vector format, Orc-Bert Augmentor can recover it and then output a pixel format image as augmented data. Different mask proportion brings diverse reconstructed output. Concatenated with Gaussian noise, the model further performs point-wise displacement to improve diversity. Experimentally, we collect two large-scale datasets of oracle characters and other Chinese ancient characters for few-shot oracle character recognition and Orc-Bert Augmentor pre-training. Extensive experiments on few-shot learning demonstrate the effectiveness of our Orc-Bert Augmentor on improving the performance of various networks in the few-shot oracle character recognition.

⋆ corresponding author.

1 Introduction

Oracle characters are the earliest known hieroglyphs in China, which were carved on animal bones or turtle plastrons in purpose of pyromantic divination in the Shang dynasty [15]. Due to the scarcity of oracle bones and the long-tail problem in the usage of characters as shown in Fig. 1, oracle character recognition suffers from the problem of data limitation and imbalance. Recognizing the oracle characters of extremely limited samples, naturally, should be taken as the few-shot learning task, which is topical in computer vision and machine learning communities, recently. Previous researches on oracle character recognition tend to discard characters of extremely limited samples or perform simple geometric augmentation to them [8, 21]. To the best of our knowledge, there is no research focused on few-shot oracle character recognition, which, however, is shown to be a real archaeological scene. Different from standard few-shot learning setting, our task does not assume the existence of large-scale labeled source oracle characters. Formally, we study under a more practical setting, where our model only has access to large-scale *unlabeled* source Chinese characters and few labeled target oracle characters.

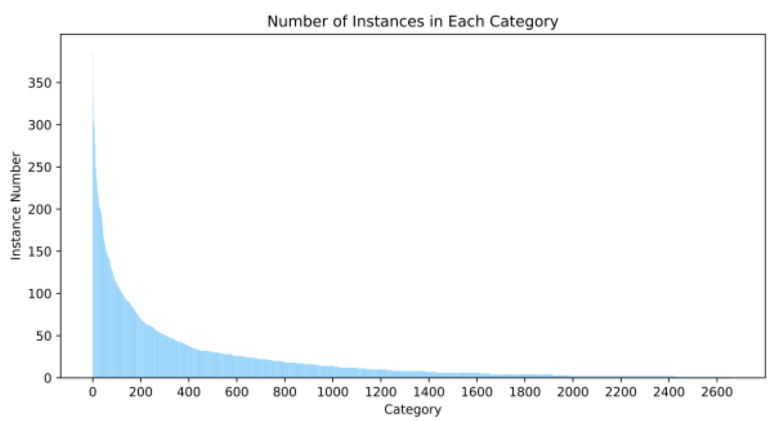

Fig. 1. The distribution of oracle character instances in Oracle-50K. Categories are ordered by number of instances.

In such a setting, meta-based or metric-based few-shot methods are failed to be efficiently trained on source unlabeled data, thus the only possible methodologies are data augmentation and self-supervised learning. Many previous works of few-shot learning [9, 13, 37, 1] utilized data augmentation and synthetic data to break the data limitation. One of the most popular strategies is geometric augmentation which includes scaling, rotation, and perspective transformation. Such methodologies, unfortunately, have to be carefully designed in order to efficiently train a robust classifier.

Simple geometric transformations apply the same fixed augmentation process to all input samples. However, in few-shot scenario, original training samples are so limited that geometric transformations are inefficient to generate numerous and diverse samples specific to an individual sample. What's more, for handwritten text or character recognition, conventional augmentation methods usually perform transformation in image level and fail to take into account the diversity of components of character or text, such as strokes of an oracle character and characters of an English word. Consequently, they can hardly imitate various writing styles to cover diversity at different levels [22].

The stroke orders of Chinese characters contain a lot of information, for which people can usually recognize a character correctly even if it is unfinished or incomplete. In another way, vector format or sequential format character images allow more diverse augmentation than pixel format data, because we can manipulate strokes and points to realize more complicated, stroke-wise, and point-wise data augmentation. However, existing geometric transformation methods are designed for pixel format images and only perform a global transformation, thus it is powerless to conduct effective augmentation to vector format images. A more powerful and complicated augmentation method is needed.

Although the stroke orders of oracle characters have been lost in history, there are two fundamental facts: 1) oracle writing or Shang writing is ancestral to modern Chinese script; 2) the modern Chinese writing system is in a left-to-right then top-to-bottom writing order. We can simply assume oracle character writing is in the same order and generate pesudo online data. To utilize stroke orders of oracle characters, sequence modeling is necessary. BERT [5] or more general transformer structure [29] presents great performances in modeling language; Sketch-BERT [19] extends BERT with sketch embedding and learns sketch representation from vector format data by self-supervised learning of Sketch Gestalt. Moreover, self-supervised pre-training task of both allows utilization of large-scale *unlabeled* data, suitable for our hard and practical setting.

In this paper, we integrate the ideas of self-supervised learning in data augmentation and propose a new data augmentation method, named Orc-Bert Augmentor, for few-shot oracle character recognition and release two characters datasets: one for Orc-Bert Augmentor pre-training and the other for few-shot oracle recognition. Like online approximation in [24], we convert offline character images in pixel format to online data in 5-element vector format, with 2-dimension continuous value for the position, and 3-dimension one-hot value for the state. Then we pre-train BERT by self-supervised learning on vector format large-scale *unlabeled* character datasets and utilize it as augmentor for augmentation of few-shot labeled oracle characters. Typically, we randomly mask points in vector format oracle with different mask probability(the higher mask probability, the harder reconstruction) and then recover the masked input using our Orc-Bert Augmentor. The pre-trained reconstruction model tends to generate a similar specific sample to the original sample, while various mask probability brings diversity. In addition, we perform random point-wise displacement by adding recovered input with Gaussian noise, i.e., a random moving state, to show that Orc-Bert Augmentor is amenable to such a naive strategy. Finally we can easily re-convert it to pixel format image. Extensive experiments on few-shot learning demonstrate the effectiveness of our Orc-Bert Augmentor on improving the performance of various networks in the few-shot oracle character recognition.

To summarize, we boost the few-shot recognition performance of existing networks by proposing a novel data augmentation method. In particular, we make the following contributions:

1. We conceptually formulate the few-shot oracle recognition problem, similar to a real-world archaeology scenario in which we does not assume the existence of large-scale labeled source oracle characters.
2. We propose a novel data augmentation approach named Orc-Bert Augmentor pre-trained by self-supervised learning. Additionally, we would also highlight the novelty of vectorization for oracle characters.
3. We collect 59,081 oracle character images and 221,947 other ancient Chinese character images, based on which we create two datasets named Oracle-FS and Oracle-50K for oracle character recognition. To the best of our knowledge, Oracle-50K is the largest public oracle character dataset until now.

2 Related Work

Oracle Character Recognition. Oracle character, the oldest hieroglyphs in China, is important for modern archaeology, history, Chinese etymologies and calligraphy study [8, 38], which has attracted much research interest [14, 38, 8, 25, 32]. Different from general Chinese character recognition [39, 35, 36], oracle character recognition suffers from data insufficiency and data imbalance. To tackle this problem, [8] discards minority categories, [38] proposes the nearest neighbor rule with metric learning, and [21] conducts simple geometric augmentation to characters with too few samples. Unlike all the prior works, we aim to address few-shot oracle character recognition under a more practical setting.

Few-shot Learning(FSL). Few-shot learning refers to the practice of model training with extremely limited labeled samples for each category. Basically, a few-shot learning model is trained on source/base data with a large number of labeled training samples and then generalized, usually by fine-tuning [33], to relevant but disjoint target/novel data with extremely limited training samples. Recent substantial progress is based on the meta-learning paradigm [7, 18, 30, 26, 27] and data augmentation [3]. In our practical setting, meta-based or metric-based few-shot methods are failed to be efficiently trained on source unlabeled data, thus the only possible methodologies are data augmentation and self-supervised learning. However, [3] is designed for few-shot natural image classification, not suitable for characters. In this paper, we propose a novel and powerful augmentation method for vector format characters. Besides, different from few-shot learning with labeled source data or semi-supervised few-shot learning [31, 1], we use unlabeled data to pre-train the augmentor.

Data Augmentation. The state-of-art deep neural networks need numerous training data with an unambiguous label. However, compared to our real world, training data usually are limited in quantity and quality, so data augmentation is an effective approach to enlarge training data and boost the overall ability of models. Random geometric augmentation [4], such as rotation, scaling, and perspective transformation, is a popular way and commonly used in classification models trained on natural images [34, 16]. But for text or character images, it regards multiple characters in a word or multiple strokes in a character as one entity to perform a global augmentation [22], without considering the diversity of each character or stroke. Different from geometric augmentation, our augmentation method converts pixel format images to points sequence and achieve more complicated augmentation both at the global and local level, satisfying local diversity. Besides, instead of randomly transformation, some works propose to generate samples by using image interpolation [6, 20] or combination [17] and generative adversarial network(GAN)[28, 23, 2], which suffers from producing augmented images very different from original images. As for Orc-Bert Augmentor, the procedure of completing or reconstructing the masked part guarantees the similarity between augmented data and original data. Note that some efficient augmentation algorithms cannot be applied to our setting (without large scale labeled data), like AutoAugment[4] or any semi-supervised learning work.

3 Datasets

The oracle characters, from Bronze Age China, are carved on animal bones or turtle shells for pyromantic divination. So far, more than 150,000 bones and turtle shells had been excavated, including approximately 4,500 unique oracle characters. Only about 2,000 of them have been successfully deciphered[12].

Oracle-20k [8] and OBC306 [12] are two currently known datasets but unfortunately un-public. Oracle-20k consists of 20,039 character-level samples covering 261 glyph classes, in which the largest category contains 291 samples and the smallest contains 25. OBC306 is composed of 300,000 instances cropped from oracle-bone rubbings or images belonging to 306 categories, which is also imbalanced. Due to limited categories in the both above datasets, we collect and publish an oracle dataset, Oracle-50K, with 2,668 unique characters. Fig. 2 shows that there is a high degree of intra-class variance in the shapes of oracle characters, resulting from the fact that oracle bones were carved by different ancient people in various regions over tens of hundreds of years. As a result, oracle character recognition or classification is a challenging task.

Oracle-50K. [1] Oracle character instances are collected from three data sources using different strategies, shown in Table 1. Instances from Xiaoxuetang Oracle[2] is collected by our developed crawling tool, wherein there are 24,701 instances

[1] https://github.com/wenhui-han/Oracle-50K.git
[2] http://xiaoxue.iis.sinica.edu.tw/jiaguwen

Table 1. Statistics of Oracle-50K and other ancient Chinese character datasets, including data source, number of instances, and number of classes.

	Data Source	Num. of Instances	Num.of Classes
Oracle-50K	Xiaoxuetang	13255	1096
	Koukotsu	18671	1850
	Chinese Etymology	27155	1120
	Total	**59081**	**2668**
Other Ancient Chinese Characters	Font Rendering	221947	/

of 2,548 individual characters. However, some instances are not provided a corresponding label represented by one single modern Chinese character, thus we only remain the deciphered instances with 13,255 instances of 1096 categories in Oracle-50K. Koukotsu[3] is a digital oracle character and text database. We utilize the TrueType font file obtained from Koukotsu to generate 18,671 annotated oracle character images belonging to 1850 classes. Chinese Etymology[4] provides 27,155 instances of 1,120 unique characters. It contains not only oracle characters but also bronze, seal, and Liushutong characters, which are also collected for augmentor training.

As we can see from Fig. 1, there exists a long-tail distribution of oracle character instances in Oracle-50K, so recognition or classification of oracle characters, especially for the categories at the distributions' tail, is a natural few-shot learning problem.

Oracle-FS. Based on Oracle-50K, we create a few-shot oracle character recognition dataset under three different few-shot settings (see Table 2). Under the k-shot setting, there are k instances for each category in the training set and 20 instances in the test set. In this paper, we set k=1, 3, 5.

Table 2. Statistics of Oracle-FS, including number of instances and number of classes.

	k-shot	Num. of Instances per Class		Num. of Classes
		Train	Test	
Oracle-FS	1	1	20	200
	3	3	20	
	5	5	20	

[3] http://koukotsu.sakura.ne.jp/top.html
[4] https://hanziyuan.net/

The Datasets for Orc-Bert Augmentor Pre-training. To pre-train BERT as our augmentor, we collect a large unlabeled ancient Chinese character dataset, including undeciphered instances of Xiaoxuetang Oracle, bronze, seal, and Liushutong characters of Chinese Etymology, and images generated from various TrueType fonts file of ancient Chinese script.

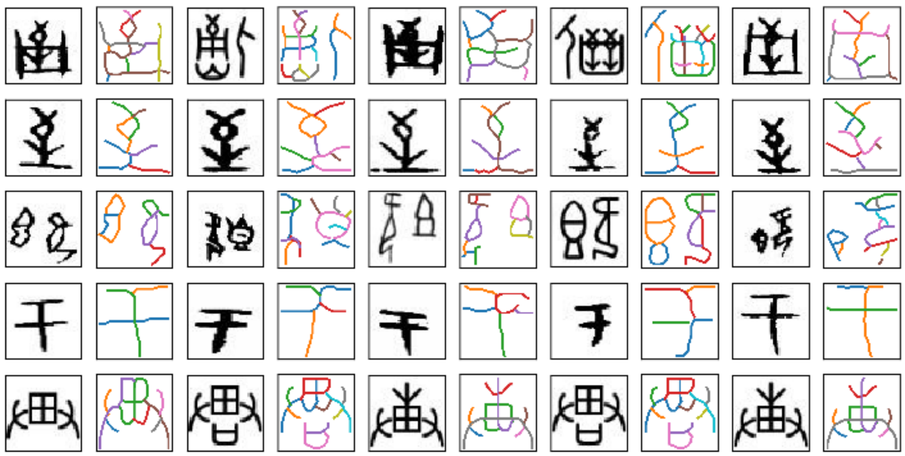

Fig. 2. Examples of oracle character images and corresponding stroke data.

4 Methodology

4.1 Problem Formulation

In this paper, we intend to address the problem of oracle character recognition under few-shot settings. More specifically, we intend to train a more human-like oracle character classifier, capable of learning from one or a few samples. Different from the conventional formulation of few-shot learning, we do not use labeled base/source data. Our 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, $\mathscr{D} = \{(\mathbf{O}_i, y_i), y_i \in \mathscr{C}\}$. In addition, we have a large amount of unlabeled data to pre-train Orc-Bert under self-supervision. Our proposed framework is illustrated in Fig. 3.

4.2 Overview of Framework

As shown in Fig. 3, our proposed framework consists of the following parts.

First, we utilize the online approximation algorithm in [24] to convert offline oracle character images with annotations and other Chinese ancient characters

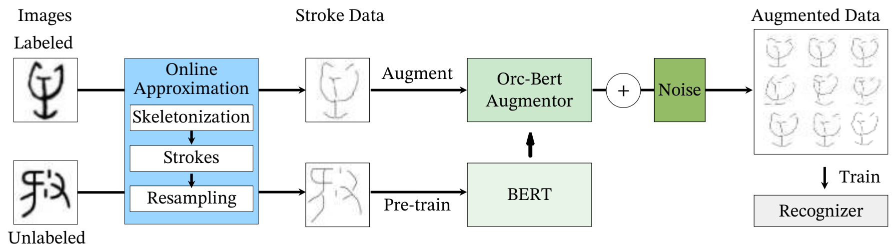

Fig. 3. Schematic illustration of our proposed framework.

images without annotations, both in pixel format, to online data in 5-element vector format (see Fig. 2), including 2-dimension continuous value for the position, and 3 dimensions one-hot value for the state. Thus, a character would be represented as a sequence of points, where each point consists of 5 attributes:

$$\mathbf{O}_i = (\Delta x, \Delta y, p_1, p_2, p_3)$$

where Δx, Δy is the position offsets between two adjacent points, and (p_1, p_2, p_3) is a 3-dimension one-hot vector ($\sum_{i=1}^{3} p_i = 1$). $p_2 = 1, p_3 = 1, p_1 = 1$ indicate the points at the ending of a stroke, the points at the ending of the whole character, and all the rest points, respectively.

Second, after getting stroke data of large-scale unlabeled data, we pre-train Orc-Bert in a self-supervised setting by predicting the masked from the visible.

Then, we utilize our pre-trained Orc-Bert as our augmentor. We randomly mask points at different mask probability and then recover masked input using our pre-trained Orc-Bert. The higher the mask probability, the harder reconstruction. To further improve the diversity of augmented data, we perform random point-wise displacement by adding completed masked input with Gaussian noise or a random moving state and re-convert it to pixel format image.

After augmentation, we train CNN-based classifiers over augmented data.

4.3 Online Approximation

Following the online approximation algorithm in [24], there are 3 steps in this stage. The first step is skeletonization, in which we convert a character image to its corresponding skeleton. The next step is the conversion of the bitmap representation to strokes that is realized by converting the bitmap to graph and then removing cycles and intersections. Finally, we conduct temporal resampling and ordering to get a more sparse points sequence. Different from simple constant time resampling, maximum acceleration resampling is proposed to imitate the dynamic speed of real writing(see [24] for more details).

4.4 Orc-Bert Augmentor

Self-supervised Pre-training of BERT and SketchBERT. BERT[5], a language representation model, is designed to pre-train bidirectional representations from unlabeled data by exploiting the mask-language model and next sequence prediction as pre-training tasks. Expanding BERT to process stroke data in computer vision, SketchBERT[19] proposes a self-supervised learning process that aims at reconstructing the masked part in a sketch. It is common practice in NLP to fine-tune the pre-trained BERT with just one additional task-specific output layer for different downstream tasks. Similarly, pre-trained SktechBERT also aims to be fine-tuned for different downstream tasks, such as sketch recognition and sketch retrieval.

Contrast to SketchBERT. We creatively propose to utilize the reconstruction procedure to generate new samples. The general structure, output layers, and input embedding of Orc-Bert are all slightly different from SketchBERT. Specifically, We adopt a smaller network architecture suitable for our data volume(see implementation details in 5.1); we add a module after the output layer to convert point sequence to pixel format image; we corrupt input for diverse augmentation(see 4.4).

Algorithm 1 Orc-Bert Augmentor+PA

1. Given an oracle image I from training set $\mathscr{D}_{train}$ and convert I into stroke data $O = (\Delta x, \Delta y, p_1, p_2, p_3)$ using online approximation module;
2. Initialize range $[a, b]$ and step n;
3. Set initial mask probability $m = a$;

repeat

1) Generate mask M according to mask probability m with the same shape of O and randomly mask O to get $O_{mask} = O \odot M$;
2) Reconstruct O_{mask} by predicting the masked states p_1, p_2, p_3 and positions $\Delta x, \Delta y$ using pre-trained Orc-Bert Augmentor and get O_{comp};
3) Sample noise ϵ from Gaussian distribution $N(\mu, \sigma^2)$ in which μ, σ^2 are mean and variance of $(\Delta x, \Delta y)$ of all training samples;
4) $O_{comp+pa} = O_{comp} + 0.1 * \epsilon$;
5) Increasing mask probability $m = m + n$;
6) Convert O_{comp} or $O_{comp+pa}$ back to pixel image $I^{'}$ and save it.

until $m > b$

Pre-training. Pre-training tasks over unlabeled data significantly facilitate the performance of BERT[5] as well as SktechBERT[19]. This auxiliary task is generally as follows: under our oracle character recognition setting, given an input data O in vector format, we perform a mask transformation and get masked input $O_{mask} = O \odot M$, where M is the mask with the same shape of input. In pre-training, Orc-Bert aims to predict the masked positions and states in O_{mask}, and generate O_{comp} as more similar to O as possible. During pre-training, we set default mask probability as 15%.

Augmentation. In augmentation, we adopt dynamic mask probability respectively for each original example to generate numerous augmented data. We discretize the range of magnitudes $[0.1, 0.5]$ into 80 values (uniform spacing) so that we get 80 different mask probability to mask the oracle sequence, respectively. With various degrees of masked input, Orc-Bert Augmentor can generate diverse augmented data. Finally, point-wise displacement is accomplished by simply adding Gaussian noise to positions or offsets of each point.

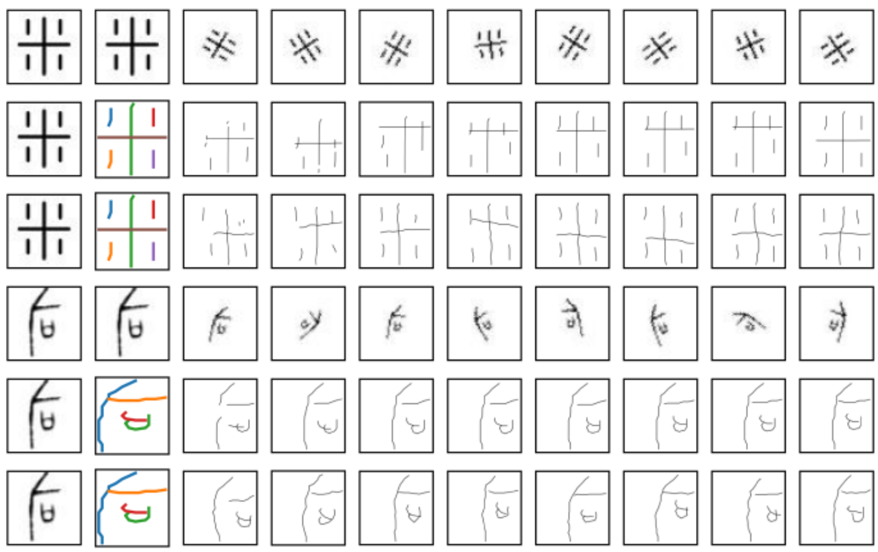

Fig. 4. Examples of augmented data generated by Conventional DA(row1&4), Orc-Bert Augmentor(row2&5), and Orc-Bert Augmentor+PA(row3&6).

5 Experiments

We conduct extensive experiments and show that Orc-Bert Augmentor achieves better performance under three few-shot settings employing different classification networks. Then we confirm this result on ablation studies about the volume of augmented and pre-training data and mask probability.

5.1 Experimental Settings

Datasets. As illustrated in Section 3, we mainly employed Oracle-FS in our evaluations. Table 2 presents statistics and train-test split about the dataset. Besides, we employed Oracle-50K(removing Oracle-FS) and all other ancient Chinese characters shown in Table 1 for self-supervised learning of Orc-Bert Augmentor.

Classifiers. Our augmentor is generic, and we can employ different networks as classifier. In this paper, we adopt 4 representative CNN models with various depth and width, including ResNet18, ResNet50, ResNet152 [10], and DenseNet161 [11], to show the effectiveness of our augmentor in boosting the classification performance of conventional networks.

Competitors. From now on, DA denotes Data Augmentation and PA denotes Point-wise Augmentation or Point-wise Displacement in this paper. For comparison, we train classifiers without augmentation (No DA) and with conventional augmentation (Conventional DA) that augment the training samples by random horizontal flipping, cropping with a ratio in $[0.8, 1]$, and rotation with a degree in $[-60°, 60°]$. Besides, we compose PA and Orc-Bert Augmentor for further exploration.

Implementation Details. We implement Orc-Bert Augmentor using PyTorch. In detail, the number of training epochs is 200 with a batch size of 10. We adopt Adam as the optimizer with a learning rate of 0.001 and 0.0001 for classifier training and augmentor pre-training, respectively. All images are resized to 50×50 before online approximation. Augmented images generated by Orc-Bert Augmentor are also 50×50, which would be resized to 224×224 for CNN training. Different from SketchBERT, the number of weight-sharing Transformer layers, hidden size, and the number of self-attention heads in Orc-Bert Augmentor are respectively 8, 128, and 8. The embedding network is a fully-connected network with a structure of 64-128-128 and the corresponding reconstruction network is 4 fully-connected networks with a structure of 128-128-64-5. The max lengths of input oracle stroke data are set as 300.

5.2 Orc-Bert Augmentor Evaluation

In this part, we evaluate our Orc-Bert Augmentor on Oracle-FS using classifiers listed in section 5.1. For Orc-Bert Augmentor, we, by default, leverage our largest pre-training dataset, set mask probability in a range of [0.1,0.5] with a sampling interval of 0.005, and generate 80 augmented instances for each sample.

Table 3 summarizes the classification accuracy (%) of each neural network classifier employing different augmentation strategies under 3 different few-shot settings. We can find that: 1) Compared with Conventional DA, almost all classifiers employing Orc-Bert Augmentor have achieved higher classification accuracy. Particularly, under the 3-shot setting, the classification accuracy achieved by ResNet50+Orc-Bert Augmentor is 52.9%, about 17% higher than the counterpart of No DA. 2) Compared with 1- and 5-shot settings, classifiers with Orc-Bert Augmentor under 3-shot scenario have improved most significantly. Extremely limited data makes data augmentation less effective because augmentation only based on one original sample is hardly practical. When data volume reaches a specific threshold, marginal effects from data augmentation is diminishing, that's what happens under the 5-shot scenario. 3) Orc-Bert Augmentor brings more significant improvement to classifiers with poorer original performance. 4) Orc-Bert Augmentor+PA achieves the best classification accuracy for almost all networks under all scenarios, showing that simple point-wise displacement is efficacious for stroke data augmentation. Both Orc-Bert Augmentor and PA are generic, which could be composed and applied to various classification networks.

Table 3. Recognition accuracy (%) on Oracle-FS under all three few-shot settings for various classifiers equipped with No DA, Conventional DA, our Orc-Bert Augmentor, and Orc-Bert Augmentor+PA. Here, DA denotes Data Augmentation and PA denotes Point-wise Augmentation or Point-wise Displacement. Specifically, DA augments each input sample by random horizontal flipping, random cropping with a ratio in [0.8,1], and random rotation with a degree in $[-60°, 60°]$. As for Orc-Bert augmentor, we leverage our largest pre-training dataset, systematically sample mask probability in a range of [0.1,0.5] with a sampling interval of 0.005, and generate 80 augmented instances for each sample. PA indicates point-wise displacement based on Gaussian distribution.

Setting	Model	No DA	Conventional DA	Orc-Bert Augmentor	Orc-Bert Augmentor +PA
1 shot	ResNet-18	18.6	20.9	29.5	**31.9**
	ResNet-50	16.8	23.3	26.2	**29.9**
	ResNet-152	14.0	18.2	26.7	**27.3**
	DenseNet	22.4	24.6	26.4	**28.2**
3 shot	ResNet-18	45.2	46.6	56.2	**57.2**
	ResNet-50	35.8	45.6	52.9	**57.7**
	ResNet-152	38.9	40.9	54.3	**57.1**
	DenseNet	48.6	52.3	56.4	**58.3**
5 shot	ResNet-18	60.8	62.7	65.1	**68.2**
	ResNet-50	55.6	60.8	62.8	**67.9**
	ResNet-152	58.6	61.4	66.1	**67.8**
	DenseNet	**69.3**	65.8	66.6	69.0

5.3 Orc-Bert Augmentor Analysis

In this part, we conduct more experiments to evaluate various aspects of Orc-Bert Augmentor, including pre-training dataset volume, size of augmented data, and mask probability. Note that, in these experiments, we employ ResNet18 as the classifier and perform experiments only under the 1-shot scenario.

Pre-trainging Dataset Volume. We construct 3 *unlabeled* datasets of different volumes for Orc-Bert Augmentor pre-training. Here, Oracle denotes Oracle-50K from which removes Oracle-FS; we add bronze and seal character images collected from Chinese Etymology[5] to Oracle and get Oracle+Bronze+Seal, all characters of which are similar to oracle in shape; Oracle+Bronze+Seal+All

[5] https://hanziyuan.net/

Other Ancient Characters is the biggest one, containing some Chinese characters different from oracle. As shown in Table 4, we can see that Orc-Bert Augmentor pre-trained on our largest pre-training dataset boosts the performance of ResNet18 the most. It indicates that Orc-Bert, like BERT[5], is also beneficial from large-scale pre-training and can be transferred to another domain after being trained on one specific domain of data.

Table 4. Classification accuracy (%) under 1-shot setting for ResNet18 equipped with Orc-Bert Augmentor pre-trained on 3 datasets with various volumes, respectively. From top to bottom, pre-training data volume increases. Other hyperparameters remain the same as Table 3.

Pre-training Dataset	Num. of Instances	Test Acc.
Oracle	64,743	27.2
Oracle+Bronze+Seal	132,788	27.8
Oracle+Bronze+Seal+ All Other Ancient Characters	276,028	**29.5**
No DA	/	18.6

Quantity, Quality, and Diversity of Augmented Data. We discretize the range of magnitudes [0.1,0.5] into 40/80/400 values as mask probabilities. Each mask probability corresponds to an augmented image. Note that, we employ ResNet18 as the classifier and perform experiments only under the 1-shot setting. From Table 6, it is easy to see that the classifier performance is most improved when the original data is augmented by 80 times, and when we generate too many augmented samples based on extremely limited data, the classifier's performance may be poorer. In addition to the impact of the quantity of augmented data, we can visualize the augmented samples to measure quality and diversity. As shown in Fig. 4, it is obvious that geometric augmentation only performs the image-level transformation, and our Orc-Bert Augmentor can generate more diverse samples. PA further improved the quality and diversity of augmented data.

Table 5. Classification accuracy (%) under 1-shot setting for ResNet18 trained with augmented data of different quantity generated by Orc-Bert Augmentor. Other hyperparameters remain the same as Table 3.

Num. of Augmented Samples	Test Acc.
40	29.2
80	**29.5**
400	28.4
No DA	18.6

Mask Probability. Mask probability is essential to similarity and diversity trade-off. A larger mask probability enhances reconstruction difficulty, generating augmented images very different from original images, while a smaller one brings the opposite results. We conduct a series of experiments with different range of mask probabilities and the results are shown in Table 6. We find that the set of mask probabilities in intermediate-range, $[0.1, 0.5]$, helps augmentor generate the highest-quality images.

Table 6. Classification accuracy (%) under 1-shot setting for ResNet18 trained with augmented data generated by Orc-Bert Augmentor with different range of mask probabilities. Other hyperparameters remain the same as Table 3.

Range of Mask Prob.	Test Acc.
$[0.1, 0.4] \cup [0.6, 0.7]$	26.3
$[0.1, 0.4] \cup [0.5, 0.6]$	29.4
$[0.1, 0.5]$	**29.5**
No DA	18.6

6 Conclusion

In this paper, we intend to address a novel few-shot setting: training few-shot model by only large-scale unlabeled source data, and few labeled target training examples. We propose a novel data augmentation method, named Orc-Bert Augmentor, for few-shot oracle character recognition. It may be the first augmentation method that converts pixel format character images into vector format stroke data and then manipulates strokes and points to generate augmented images. It leverages Orc-Bert pre-trained on large-scale *unlabeled* Chinese characters to recover masked images as augmented data for CNN classifiers training. Besides, we incorporate point-wise displacement with Orc-Bert Augmentor, which presents better performance. Orc-Bert Augmentor is simple yet effective. Extensive experiments under three few-shot settings confirm the effectiveness of our Orc-Bert Augmentor to improve the performance of various networks on few-shot oracle character recognition. Moreover, we collect and publish two datasets: Oracle-50K and Oracle-FS. In the future, we will explore to generate more diverse and higher-quality augmented samples by modifying the loss function or jointly optimizing the augmentor and the classifier. We dedicated to applying our method to more general sketch and handwritten character recognition.

Acknowledgement

This work was supported in part by NSFC Projects (U62076067), Science and Technology Commission of Shanghai Municipality Projects (19511120700,19ZR1471800).

References

1. Antreas Antoniou and Amos Storkey. Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation. *arXiv preprint arXiv:1902.09884*, 2019.
2. Ayan Kumar Bhunia, Abhirup Das, Ankan Kumar Bhunia, Perla Sai Raj Kishore, and Partha Pratim Roy. Handwriting recognition in low-resource scripts using adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
3. Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8680–8689, 2019.
4. 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.
5. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
6. Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017.
7. Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ArXiv*, abs/1703.03400, 2017.
8. Jun Guo, Changhu Wang, Edgar Roman-Rangel, Hongyang Chao, and Yong Rui. Building hierarchical representations for oracle character and sketch recognition. *IEEE Transactions on Image Processing*, 25(1):104–118, 2015.
9. Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016.
10. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
11. Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.
12. S. Huang, H. Wang, Y. Liu, X. Shi, and L. Jin. Obc306: A large-scale oracle bone character recognition dataset. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 681–688, 2019.
13. Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International journal of computer vision*, 116(1):1–20, 2016.
14. Naoto Kamitoku and Katsuhiro Yamazaki. Oracle bone inscription detector based on ssd. In *New Trends in Image Analysis and Processing–ICIAP 2019: ICIAP International Workshops, BioFor, PatReCH, e-BADLE, DeepRetail, and Industrial Session, Trento, Italy, September 9–10, 2019, Revised Selected Papers*, volume 11808, page 126. Springer Nature, 2019.
15. 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.
16. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

17. Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869, 2017.
18. Zhenguo Li, Fengwei Zhou, Fei Chen, and Huang Li. Meta-sgd: Learning to learn quickly for few shot learning. *ArXiv*, abs/1707.09835, 2017.
19. Hangyu Lin, Yanwei Fu, Xiangyang Xue, and Yu-Gang Jiang. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
20. Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9090–9098, 2018.
21. Guoying Liu and Feng Gao. Oracle-bone inscription recognition based on deep convolutional neural network. *JCP*, 13(12):1442–1450, 2018.
22. Canjie Luo, Yuanzhi Zhu, Lianwen Jin, and Yongpan Wang. 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*, pages 13746–13755, 2020.
23. Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018.
24. Martin Mayr, Martin Stumpf, Anguelos Nikolaou, Mathias Seuret, Andreas Maier, and Vincent Christlein. Spatio-temporal handwriting imitation. *arXiv preprint arXiv:2003.10593*, 2020.
25. Lin Meng. Recognition of oracle bone inscriptions by extracting line features on image processing. In *ICPRAM*, pages 606–611, 2017.
26. Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.
27. Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.
28. Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.
29. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
30. Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.
31. Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12836–12845, 2020.
32. Jici Xing, Guoying Liu, and Jing Xiong. Oracle bone inscription detection: a survey of oracle bone inscription detection based on deep learning algorithm. In *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, pages 1–8, 2019.
33. Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

34. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
35. Xu-Yao Zhang, Yoshua Bengio, and Cheng-Lin Liu. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition*, 61:348–360, 2017.
36. Yaping Zhang, Shan Liang, Shuai Nie, Wenju Liu, and Shouye Peng. Robust offline handwritten character recognition through exploring writer-independent features under the guidance of printed data. *Pattern Recognition Letters*, 106:20–26, 2018.
37. Yaping Zhang, Shuai Nie, Wenju Liu, Xing Xu, Dongxiang Zhang, and Heng Tao Shen. Sequence-to-sequence domain adaptation network for robust text image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2740–2749, 2019.
38. Yi-Kang Zhang, Heng Zhang, Yong-Ge Liu, Qing Yang, and Cheng-Lin Liu. Oracle character recognition by nearest neighbor classification with deep metric learning. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 309–314. IEEE, 2019.
39. Zhao Zhong, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Handwritten chinese character recognition with spatial transformer and deep residual networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3440–3445. IEEE, 2016.