More than Orc-BERT: Characters Recognition Transfer and Various Data Augmentation for Recognizing Few-Shot Oracle Characters

Yifan Li
School of Data science
Fudan University
Shanghai, China
19307110499@fudan.edu.cn

Xuetian Chen
School of Data Science
Fudan University
Shanghai, China
19307110189@fudan.edu.cn

Abstract

This paper studies the recognition problem of oracle characters based on few-shot learning. For oracle bone data, we restrict access to a large scale of unlabeled source ancient Chinese characters and a small number of labeled oracle characters. We propose transfer learning on this task, combining self-supervised learning and data augmentation. On the basis of transfer learning, we further explored the Orc-Bert Augmentor based on self-supervised learning pre-training and analyzed its effect. Specifically, we pre-trained the model using the HWDB dataset and transferred the parameters of feature extraction from the previous layers to our classification model. We experimentally demonstrate the effectiveness of our transfer learning method on this task, compare and analyze the effects of combining different augmentors. Extensive few-shot learning experiments demonstrate that our transfer learning method combined with data augmentation greatly improves the classification accuracy under all network settings in few-shot oracle character recognition. Our best top-1 accracy under three few-shot setting are 52.9%, 75.8%, 84.3%.

1 Introduction

Oracle bone inscriptions are the earliest, systematic, recognizable and meaningful characters known in my country. As one of the four ancient scripts in the world, oracle bone inscriptions are named for their inscriptions on tortoise shells and animal bones. It appeared in the Shang Dynasty and was a tool for people to communicate at that time. In the beginning, the ancients engraved the meaning to be expressed on the oracle bones of the prey in the form of symbols, mostly on the front; because humans feared the gods at that time, and their lives were mainly activities related to gods such as sacrifice and divination, so the oracle bone inscriptions The main content is divination, which is called "divination".

So far, more than 160,000 oracle bone fragments have been unearthed. It is a pity that these oracle bone fragments have been buried underground for many years, and have experienced the impact of underground activities and wear caused by excavation. Most of the oracle bones are severely damaged; and due to the migration of human history, they have spread to all corners of the world, and it is difficult to have complete information. . In the new century, more than 15,000 oracle bones with characters have been discovered in our country, but far less than half of them have been deciphered.

The results of the study show that deep learning can help historians to better interpret inscriptions to facilitate the demonstration and understanding of ancient history. For example, DeepMind's deep neural network Ithaca Assael et al. (2022) can decipher ancient Greek characters from damaged cultural relics with an accuracy rate of 62% and an accuracy rate of 71% in recognizing their original location. It can also lock the age of ancient characters within their real date range. within 30 years. This is the first deep neural network capable of recovering the missing text of a damaged inscription, recognizing its original location and helping to date its writing.

In recent years, with the development of artificial intelligence, the automatic identification and matching of oracle bones can be realized through image recognition technology, which can improve the quality and efficiency of oracle bone literature scholars. However, due to the scarcity of oracle bone inscriptions, the problem of long tails of characters using, and the large differences in writing style between the same characters, character recognition is a very difficult task for both archaeologists and deep learning.

We wanted to simulate the real background of archaeological work and identify oracle bone characters with little reference information. This corresponds to few-shot learning Wang et al. (2019) in deep learning. But in the past, few-shot learning cases based on Oracle were very few, and most of the techniques used traditional image enhancement. Han et al. (2020) proposes a few-shot learning scenario based on Oracle. The paper argues that under few-shot learning, the only possible approaches are data augmentation and self-supervised learning. Vector-format or sequential-format character images allow for more diverse enhancements than pixel-format data on this task. Therefore, a new data augmentation method, Orc-Bert Augmentor, is proposed for few-shot oracle character recognition. Randomly mask points in a vector format oracle with different mask probabilities, then restore the masked input using Orc-Bert Augmentor. This works by accessing large-scale unlabeled source oracle characters to efficiently identify new oracle characters that contain a small number of labeled training examples.

We believe that for most of the current deep learning networks, the amount of parameters is staggering, and it is difficult for learning to converge to a better local optimum on a small dataset. Therefore, we believe that increasing the amount of data input to the model is the best way to make the model converge. But in the context of few-shot learning, we cannot use the larger annotated oracle database. We need to introduce other datasets, so is the essential idea of Orc-Bert. We combined transfer learning to this problem, by pre-training the model on a handwritten Chinese database similar to Oracle, learning the feature extraction parameters, and then applying it to the Oracle classification task. In addition, we also implement the Orc-Bert enhancement method proposed in Han et al. (2020). Experiments show that the enhancement directions of these two optimization methods are similar, and our method is much better than using Orc-Bert only. Additionally, our method is highly adaptable and can be used in conjunction with other data augmentation, which can greatly refresh the current state-of-art record.

In conclusion, we propose a feasible transfer learning method for this task and implement the Orc-Bert model, which greatly refreshes the original record.

2 Related Work

Few-shot Learning.. 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 a large-scale labeled dataset, and then fine-tuning the parameters to generalize it to similar but disjoint target dataset with extremely limited training samples. The basic method for few-shot learning is transfer learning. Modern solutions to few-shot learning includes model-based, metric-based and optimization-based methods. In this project, we will explore the efficiency of transfer learning and data augmentation on few-shot learning.

Transfer Learning. When the target training set is too small to train a network from scratch, we can directly use networks that others have trained on general large-scale labeled dataset, and then fine-tune the parameters to generalize it to similar but disjoint target dataset with extremely limited training samples. Many CNNs of various architectures trained on ImageNet, such as VGGSimonyan and Zisserman (2015) and ResNetHe et al. (2016) are outstanding models that we can download directly.

Data Augmentation. The state-of-art deep neural networks need numerous labeled training data. 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. The basic random geometric transformations, such as rotation, scaling, and flipping are effective augmentation methods in image classification and other computer vision tasks. However, considering the natural left-to-right writing order, the basic augmentation methods is less suitable. Orc-BERTHan et al. (2020) is a transformer based neural augmentor, designed for sequence of strokes Oracle characters.

BERT, Sketch-BERT and Orc-BERT. BERTDevlin et al. (2018) is a bidirectional transformer-based language representation model, which is pre-trained on large-scale unlabeled corpus by exploiting the mask language model and next sequence prediction as pre-training tasks. Expanding BERT to process stroke data in computer vision, Sketch-BERTLin et al. (2020) proposes a self-supervised learning process that aims at reconstructing the masked part in a sketch. Orc-BERTHan et al. (2020) share the same architecture as Sketch-BERT, but its network is quite smaller, which is suitable for Oracle-50K volume.

3 Datasets

Oracle Dataset Description. The dataset we use includes oracle character examples and other collected ancient Chinese character images, which are collected by Han et al. (2020).

The data is mainly divided into two parts, the labeled Oracle-50K and Other Ancient Chinese CHaracters which are unlabeled. Oracle-50K is collected from three data sources including Xiaoxuetang, Koukotsu, Chinese Etymology according to different strategies. The label with some characters is not a single modern Chinese character, does not meet the requirements, and is also classified as unlabeled Other Ancient Chinese Characters. In addition to oracle bones, the dataset also includes bronze, seal, and Liushutong characters. Other Ancient Chinese Characters also contains images various True-Type fonts file of ancient Chinese script. The specific statistics are shown in the table below1.

	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	/

Table 1: Statistics of collected data

This dataset has obvious long-tailed features, so few-shot learning is very suitable for the character classification problem of this dataset.

Train/Test Split. Unlabeled data is used to learn stroke structure, while labeled data is used to train and test classification models. Based on Oracle-50K, pick out a part of the data under different shot settings as few-shot Oracle character recognition dataset. Specifically, 200 classes are picked out as objects for learning. In the K-shot setting, each class has k instances in the training set and 20 instances in the test set. In this project, set k=1,3,5.

Datasets for Orc-Bert Augmentor Pre-training. To pre-train Orc-Bert as augmentor, all the remaining data above is grouped into the dataset for training the Orc-Bert augmentor. In order to give the model a better perception of strokes, we used all the data.

HWDB Dataset. To pretrain our classification model, we used a database of Chinese Handwritten isolated characters from the Institute of Automation of Chinese Academy of Sciences[1]Liu et al. (2011). Handwriting samples were created on paper using Anoto pens by 1,020 writers, resulting in both online and offline data. The isolated character dataset contains about 3.9 million samples with 7,356 categories (7,185 Chinese characters and 171 symbols).

For handwriting data collection, a character set was first compiled according to the standard set GB2312-80 and the Common Set of Modern Chinese Characters (Common Set for short), and the union of the two sets was collected, including 7,170 characters. In addition, a set of 171 symbols has also been collected, including 52 English letters, 10 numbers, and some commonly used punctuation,

[1] http://www.nlpr.ia.ac.cn/databases/handwriting/Download.html

mathematical and physical symbols. Therefore, the total number of character classes is 7,356. The training set and test set are divided in a ratio of 4:1.

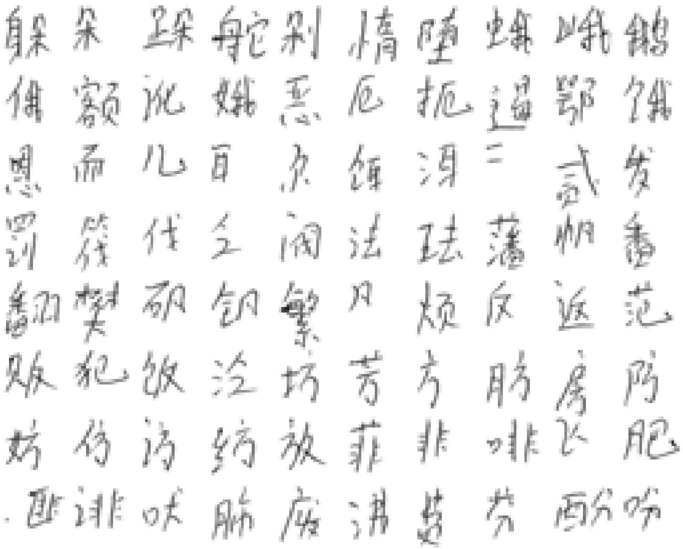

Figure 1: Isolated character samples

The dataset we selected this time is HWDB1.1, which contains a total of 3755 classes, the training set contains 897,758 pictures, and the test set contains 223,991 pictures. Due to the large amount of total data, we picked 500 classes of characters as our training and testing dataset.

4 Methodology

4.1 Problem Formulation

In this project, we intend to address the problem of oracle character recognition under few-shot settings. More specifically, we intend to train an oracle character recognizer, capable of learning from one or a few samples. Different from the conventional formulation of few-shot learning, we do not use large-scale labeled dataset. Our classifier would have only access to k labeled training instances for each category and then be tested on 20 instances per class, namely, Oracle-FS. In addition, we have a large amount of unlabeled data (Oracle-source) to pre-train Orc-Bert under self-supervision. Our proposed framework is illustrated in Fig. 2.

4.2 Overview of Framework

As shown in Fig. 2, our proposed framework consists of the following steps.

1. Convert the 3-element sequence data of Oracle-source to 5-element sequences, where the first 2-dimension are continuous values for the position shift and the other 3-dimensions are one-hot values for the state. Thus, a character would be represented as a sequence of points, where each point consists of 5 attributes:

   $$O_i = (\Delta x, \Delta y, p_1, p_2, p_3),$$

   where Δx, Δy is the position offsets between two adjacent points, and the one-hot values p_2, p_3, p_1 will be 1 if the point is the end of a stroke, the end of a character, and the rest normal point.
2. After getting stroke data sequence of large-scale unlabeled data, we pre-train our Orc-Bert augmentor in a self-supervised setting by predicting the masked point from the visible.
3. Then we utilize our pre-trained Orc-Bert augmentor to generate new training samples. We randomly mask points at different mask probability and then recover masked input using

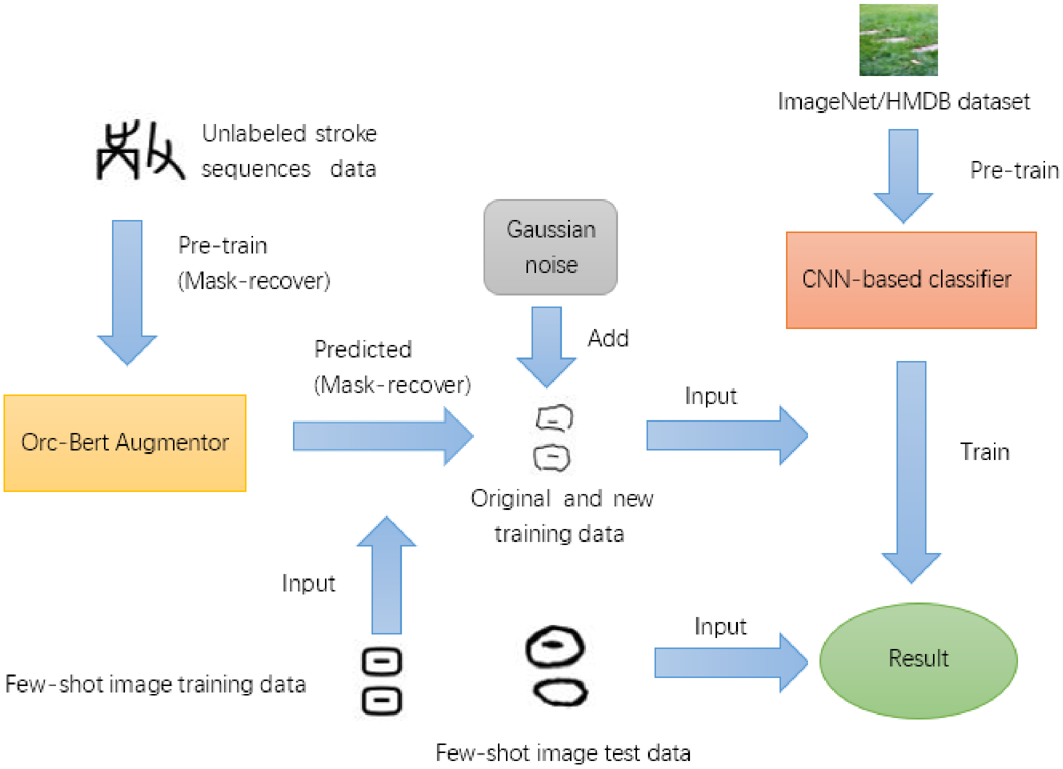

Figure 2: Our framework of this project

our pre-trained Orc-Bert. In other words, we input the masked sequences of few-show training examples, and the output of Orc-Bert is the completed ones, which will be used as new training samples of our recognizer. 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 and then reconvert it to pixel format image.

4. After augmentation, we train CNN-based classifier over augmented data (augmented Oracle-FS training set). In order to make use of transfer learning, we will pre-train the CNN-based classifier by large-scale image dataset or character dataset. In particular, we directly use the pre-trained weights from ImageNet (14,197,122 images, 21841 classes), and try to pre-train on a small part of HWDB Dataset (about 150,000 images, 500 classes) from scratch.

5. The final step is evaluate our classifier over the Oracle-FS test set. No data augmentation and extra are used here.

We will explain the details of step 2 and 3 in the following sections.

4.3 Orc-Bert Augmentor

BERT and its self-supervised learning BERTDevlin et al. (2018) is a bidirectional transformer-based language representation model, which is pre-trained on large-scale unlabeled corpus by exploiting the mask language model and next sequence prediction as pre-training tasks. In detail, the input sequence of words are randomly masked, and the goal is to predict the masked words by the bidirectional information from the context in the mask language model. By the process of self-mask and self-recover, the model can learn general semantic features from the large-scale corpus. After pre-training, it can easily recover the simple masked sentences with reasonable words.

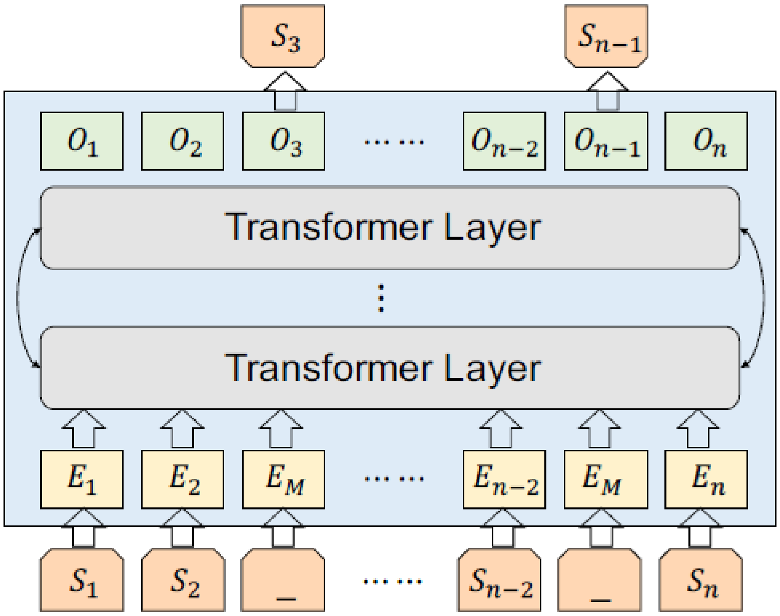

Figure 3: Sketch-BERT model structure

Sketch-BERT and Orc-BERT Expanding BERT to process stroke data in computer vision, Sketch-BERTLin et al. (2020) proposes a self-supervised learning process that aims at reconstructing the masked part in a sketch. Considering a sketch as a sequence of 5-element points (illustrated in section 4.2), its model setting is quite similar to BERT, but it use ALBERTLan et al. (2019) structure to share the parameters in the transformer layers to accelerating convergence and prevent overfitting.

Using similar architecture as BERT, Sketch-BERT network starts from three embedding layers, which represents for point embedding, positional embedding and stroke embedding, corresponding to three levels of word embedding in BERT. Then the multi-head bidirectional transformer layers extract the features of the sketch. Then the output embedding pass through the mask-recover head to predict the masked stroke of the sequence. Fig. 3 shows the structure of Sketch-BERT model, whose basic conponets are embedding layer, transformer encoder layers, and the reconstruction head.

Our Orc-BERT model share the same architecture as Sketch-BERT model, but owns a smaller network, which is suitable for our data volume. See 5.1 for detailed settings.

Pre-training The pre-training task is completed over unlabeled stroke sequence data (Oracle-50K and other fonts). For the i^{th} training character O_i:

1. First we generate a binary mask tensor M_i, which shares the same shape as O_i, with probability 0.15. In other words, 15% elements of M_i are 0, and the rest are 1. Note that the mask matrix of position and states are generated jointly and separately. It means that if a pair $(\Delta x, \Delta y)$ is chosen to be masked, they will both be zero. And if a pair (p_1, p_2, p_3) is chosen, they will all be zero, too. But the mask of $(\Delta x, \Delta y)$ and (p_1, p_2, p_3) are generated independently.
2. Perform element-wise product to get the masked input: $O_i^{mask} = O_i \cdot M_i$.
3. The input O_i^{mask} passes through the encoder and constructor, and the output is viewed as the predicted of the original sequence O_i. Cross entropy loss is chosen to help the model learn to recover the masked positions and states of the masked input.

Augmentation After pre-training, characters are input as sequence of strokes. Orc-BERT will perform deep learning based augmentation and point-wise augmentation (PA) as follows:

1. Confirm an interval of mask probabilities $[a, b]$ and the number of new generated samples per training sample m. Sample n mask probabilities $p_1, p_2, \ldots, p_m$ equidistantly.

2. For each one p_j of the m mask probabilities, mask the i^{th} training sequence O_i with probability p_j. Then reconstruct m samples $O_{ij}^{tmp}, j = 1, 2, \ldots, m$.
3. Add a Gaussian noise $\epsilon_j \sim N(\mu, \sigma^2)$ to O_{ij}^{tmp} to get the new training sample: $O_{ij}^{new} = O_{ij}^{tmp} + 0.1\epsilon_j,\ j = 1, 2, \ldots, m$, where μ, σ^2 are the mean and variance of $(\Delta x, \Delta y)$ of all original training samples.
4. Convert the sequence of strokes O_{ij} to a pixel image I_{ij} and save it.

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.

The generating process is described in detail in the following algorithm 1.

```
Algorithm 1: Orc-Bert Augmentor+PA
Input: $O_i$, the $i^{th}$ training oracle character, a sequence of points $(\Delta x, \Delta y, p_1, p_2, p_3)$, the mask
       probability interval $[a, b]$, and the number of new samples per instance $m$.
Output: $I_{ij}, j = 1, 2, \ldots m$, images of the new training characters predicted by Orc-BERT.
Initialize the mask probability $p = a, j = 0$;
while $j < \frac{b-a}{m}$ do
    1) Generate mask matrix $M_j$ according to mask probability $p$ with the same shape of $O_i$ and
       randomly mask $O_i$ to get $O_{ij}^{mask} = O_i \cdot M_j$;
    2) Reconstruct $O_{ij}^{mask}$ 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 $O_{ij}^{tmp}$;
    3) Sample noise $\epsilon_{ij}$ from Gaussian distribution $N(\mu, \sigma^2)$ in which $\mu, \sigma^2$are mean and
       variance of $\Delta x, \Delta y$ of all training samples;
    4) Add Gaussian noise to the generated sequences $O_{ij}^{new} = O_{ij}^{tmp} + 0.1 * \epsilon_{ij}$;
    5) Convert the stroke sequence $O_{ij}^{new}$ into pixel format image $I_{ij}$;
    6) $j \leftarrow j + 1$;
end
```

4.4 Classifier

After generating new training samples, we use the original and the new training images to train a CNN-based classifier by a simple task: image classification. The baseline of CNN we use is ResNet-18. CNN will deeply extract the features of images and perform efficient classification. Inspired by transfer learning, we use large-scale labeled data (images with classes) to pre-train the network, re-initialize the last fully connected layer, and train on our Oracle-FS dataset. We directly download the parameters learned in ImageNet as the feature extractor. For contrast, we pre-train the network on HMDB from scratch.

5 Experiments

We conduct extensive experiments showing that using transfer learning achieves substantial performance gains in three few-shot settings. Then we implement Orc-Bert augmentor, compare the effects of three different augmentor settings, and revised the experimental settings based on the original paper.

5.1 Experimental Settings

Datasets. As illustrated in Section 3, we use Oracle-source (Oracle-50K removing Oracle-FS) as pre-training dataset (for Orc-Bert Augmentor), HWDB1.1 as classifier pre-training dataset, and employed Oracle-FS in our evaluations.

Among them, we use all the data of Oracle-source and Oracle-50K, and uses random sampling on HWDB dataset to select 500 character classes as training and testing dataset, of which there are 119,536 images in the training set and 29,825 images in the test set.

Classifier. Our proposed method is general across all networks. Considering the performance of the network and the time complexity of training, we mainly use ResNet-18He et al. (2016) as the model shown in the experiments. In addition, VGG-19-BNSimonyan and Zisserman (2015), wide ResNet 101 and other models are also used to find the best model on the task.

Augmentor. In this experiment, we mainly experimented with four different augmentor settings. The first is without any data augmentation (No DA for short). The second is conventional data augmentation of images (DA for short), augmenting training samples by random horizontal flips and random rotations. The third is to augment the stroke sequence, using Orc-Bert and point-by-point enhancement to augment the strokes of the training samples (Orc-Bert with PA for short). The fourth is to combine the two augmentors. First, the strokes are augmented by Orc-Bert with PA, and then the images generated by the augmented strokes are augmented by DA. The specific hyperparameter settings can be found in the subsequent experiments.

Implementation Details. The deep learning framework we use is Pytorch.

- **Orc-BERT Augmentor** The number of training epochs is 100 with a batch size of 10. We adopt Adam as the optimizer with a learning rate of 1×10^{-4} for pre-training. Augmented images generated by Orc-Bert Augmentor are 50 x 50. 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.
- **CNN classifier** The default number of rounds during training is 30 epochs, and the batch size is 10. We adopt Adam as the optimizer, and the learning rates for ResNet-18 classifier training is 5×10^{-4}. All images are resized to 224×224 during classifier training and normalized with a mean and variance of 0.5. Use cross-entropy as the loss function and cosine subsidence function as the learning rate scheduler.

5.2 Transfer Learning

In this section, we evaluate the performance of the models under No DA and DA on Oracle-FS. The specific setting of DA is to randomly flip horizontally with probability 0.5 and rotate randomly within the range of $[15°, 15°]$. The reason for the slight discrepancy with the original paper Han et al. (2020) is because we believe that on this data, due to the relatively small number of data samples, excessive data augmentation will confuse the model's learning of the data and reduce the performance of the model, so we adjusted the magnitude of data augmentation. The classification accuracy of the two augmentor settings under different shot settings can be seen in the table2. It can be seen that the results of the model with DA are indeed higher than the original paper Han et al. (2020), which confirms our conjecture.

In fact, in addition to traditional image transformation, there are many methods for data augmentation suitable for classification tasks, such as cutout Devries and Taylor (2017), mixup Zhang et al. (2018), cutmix Yun et al. (2019), etc. But we are not using these methods here. On the one hand, the reason is mentioned above, small dataset are not suitable for too large disturbance, otherwise it will cause detriment to model training. On the other hand, Chinese characters are very sensitive to structure, unlike images for other classification tasks. For example, using cutout to splicing two words together is likely to create completely different thirdly character, but not the weighted sum of the original two character labels. Therefore, the above data augmentation methods are not suitable for this classification task. In fact, we also uphold a rigorous attitude, and experiments have shown that these three data augmentors perform worse even than No DA.

From the results, we have the following findings:

1. It can be seen that the classification accuracy of the model with DA is significantly better than that of No DA in each shot setting.

2. Comparing 1-shot and 5-shot, the classification accuracy of 3-shot increases the most under DA, reaching 15.6%. When the shot setting is relatively small, extremely the 1-shot. DA will alleviate the overfitting of the model, but also due to the small amount of data, data augmentation may also distort the data information. As the shot increases and the information input to the model increases, the misleading effect caused by DA decreases, so the increase in the model classification accuracy will be larger. However, when the data set is gradually diversified, the gain brought by DA is gradually saturated, and the increase in classification accuracy brought by DA will decrease. And in here, 3-shot is the turning point.

setting	No DA	DA
1-shot	16.7	26.2
3-shot	41.9	57.5
5-shot	62.5	70.4

Table 2: Recognition accuracy(%)) on Oracle-FS under all three few-shot settings for ResNet-18 equipped with No DA and Conventional DA. Here, DA denotes Data Augmentation. Specifically, DA augments each input sample by random horizontal flipping and random rotation with a degree in $[15^\circ, 15^\circ]$.

Despite the presence of DA, the model still performs poorly due to the small amount of data. Inspired by transfer learning, we use pretrained model next. The pre-training of the model is implemented as follows.

We use a total of 500 classes of characters. Convert the HWDB data to image data in PNG format, and also resize the input image to 224×224 and normalize it. The pre-training batch size is 10, the learning rate is 1e-3, and the maximum epochs of training is 30. Use cross-entropy as loss function. The training log is as figure 4.

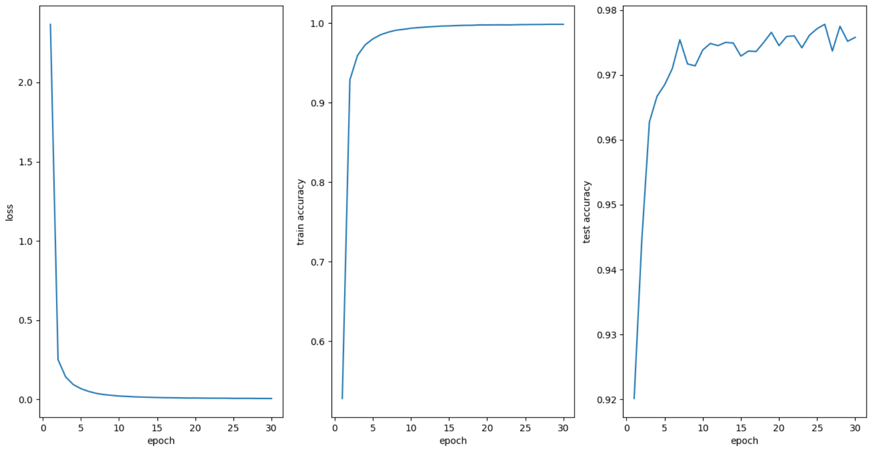

Figure 4: Loss, train accuracy and test accuracy of classifier pre-training

Using the HWDB pre-trained model, first reset the last layer parameters of the model, then train the model under the same settings as above, and the classification accuracy is shown in the table3.

As can be seen from the above two tables, the model using transfer learning has a large gain in any data augmentation setting and shot setting than the model without transfer learning. The largest increase in classification accuracy is 27.8%, under No DA and 1-shot setting. Such an increase is staggering, but also reasonable. This is because the pre-training model has already learned the hidden layer for extracting the primary features of the image, and fine-tuning based on the pre-training model only needs to learn the final hidden layers parameters which combined features to classify. This

setting	No DA	DA
1-shot	44.5	48.6
3-shot	65.2	70.2
5-shot	74.6	80.3

Table 3: Recognition accuracy(%)) on Oracle-FS under all three few-shot settings for pretrained ResNet-18 using HWDB dataset. The augmentor settings are shown in the table 2

greatly shortens the distance from which the model converges to the optimal point, enabling better convergence even if the training dataset has only a small amount of data.

From the above experimental data, we also found that

1. No DA always increases more than DA regardless of the shot setting. The diversity of input training data of No DA is less, so the overfitting problem is more serious. Therefore, the gain from using a pretrained model is greater. Transfer learning allows the model to use limited data to focus more on the underlying task rather than feature extraction. The end result is to narrow the distance between using No DA and using DA.
2. Under the two data augmentation settings, with the increase of the shot setting, the whole frame brought by the pre-trained model decreases. As the dataset grows, the model gets closer to the optimal point, and the effect of the starting position decreases gradually. Therefore, the effect of transfer learning gradually decreases.

5.3 Analysis For Pre-training

We experimented with the different amounts of pre-training data. The 100-class data contains 47,799 training images and 11,936 testing images. The classification accuracy of different pre-trained models is shown in the following table 4. The more data used for pre-training, the better the classification performance after the model is fine-tuned.

shot setting	No pretrain	100 classes	500 classes
1-shot	26.2	46.5	48.6
3-shot	57.5	69.5	70.2
5-shot	70.4	79.3	80.3

Table 4: Recognition accuracy(%) on Oracle-FS under all three few-shot settings for different pretrained ResNet-18. Using DA as augmentor. No pretrain here means not use any HWDB data to pretrain. 100 class means random sampling 100 class of HWDB dataset to pretrain ResNet-18. 500 class means random sampling 500 class of HWDB dataset to pretrain ResNet-18.

5.4 Data Augmentor

Continuing, we complete some experimental comparison of different data augmentors in this section. We implemented the Orc-Bert augmentor to augment the sequence.The learning rate of Orc-Bert pre-training is 1e-4 The augmented images generated by Orc-Bert Augmentor are 50 × 50, which will be resize to 224 $\times$ 224 before CNN input.

Different from SketchBERT, the weights in Orc-Bert Augmentor share the number of Transformer layers, the hidden size, and the number of self-attention heads to 8, 128, and 8, respectively. The embedded network is a fully connected network with a 64-128-128 structure, and the corresponding reconstructed network is four fully connected networks with a 128-128-64-5 structure. The maximum length of the input oracle sequence data is set to 300. The total number of training rounds of pre-training is 100 rounds.

In addition, in this experimental section, the dataset used by our pre-trained model is replaced by ImageNet, for the specific reasons stated in the next section.

The model classification accuracies for the four data augmentation settings are shown in the table 5.

From the experimental results:

			Orc-Bert	Orc-Bert
Setting	**No DA**	**DA**	**Augmentor**	**DA + Augmentor**
			with PA	**with PA**
1 shot	40.2	47.2	44.5	50.7
3 shot	68.0	72.0	69.9	74.9
5 shot	78.8	81.5	80.0	83.4

Table 5: Recognition accuracy (%) on Oracle-FS under all three few-shot settings for pre-trained ResNet-18 equipped with No DA, Conventional DA and and Orc-Bert Augmentor with 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 and random rotation with a degree in $[15^\circ, 15^\circ]$. As for Orc-Bert augmentor, we leverage our largest pre-training dataset, systematically sample mask probability in a range of [0.05,0.15] with a sampling interval of 0.01, and generate 10 augmented instances for each sample. PA indicates point-wise displacement based on Gaussian distribution.

1. On the pre-trained model, the effect of using Orc-Bert+PA alone is not as good as that of traditional data augmentation. This actually shows that the effect of Orc-Bert is in conflict with pre-training.

   Orc-Bert creates new samples from a sequence perspective by training a model to learn about strokes. These samples can help the model learn parameters for extracting primary features of character images. But transfer learning can achieve the same effect. Therefore the effect of Orc-Bert is greatly reduced. But in contrast, traditional data enhancement still maintains a good effect. We believe that this is because the data augmentation we use only flips or rotates the image, and does not change the relative calligraphy and strokes structure, resulting in most of the primary features are the same. So there is less conflict between DA and the effect of transfer learning (but there are also partial conflicts, in The experimental results in the previous section 5.2 confirm that the gap between No DA and DA is reduced after transfer learning).

   In addition, Orc-Bert may not be as effective as DA on the hidden layers learned by downstream tasks. Because DA makes little changes to the image, the accuracy of most of the data can be guaranteed. But Orc-Bert cannot guarantee to create perfect new samples for all characters due to the limited learning ability of the model. We discovered this problem when we observed the data generated by Orc-Bert.

   For simple character 5 mask samples, Orc-Bert can generate new samples that are close to the real character even at high mask probability. For moderately complex words 6, the generated results are still good with small mask probability, but when the mask probability increases, the generated results are gradually distorted. For complex words 7, even if the mask probability is small, it is difficult to recover all the character details. When the mask probability is large, the generated result is difficult to identify even for humans. Therefore, for complex samples, these wrong samples created by Orc-Bert will interfere with the learning of the classifier and reduce the effect of the model. The classification accuracy of these complex samples makes the zonal classification accuracy bring a negative gain.

2. The best results are obtained by using both Orc-Bert + PA and DA. This shows that the two augmentors are complementary.

3. Comparing the table 3, you can see that the pre-trained model using ImageNet dataset and the pre-training model using HWDB dataset are slightly different. While the classification accuracy of 1-shot decreases, the classification accuracy of 3-shot and 5-shot both increased.

   We believe that this is because the ImageNet dataset is larger, and the images are all RGB three-channel data, which bring out more information, so the pre-trained model has a stronger ability to extract features from images. However, the classification task of ImageNet is completely unrelated to Oracle character classification, so they cannot help in parameter learning of downstream task classification. The HWDB dataset is a classification of handwritten Chinese characters which similar to Oracle, so the effect of using transfer learning will be better. Therefore, on 1-shot, the HWDB dataset performs better.

   However, for 3-shot and 5-shot, the training data set is enough to learn the parameters to achieve classification on character features, the marginal effect of the pre-training data set gradually decreases, and the robuster and more general ability of feature extraction of the ImageNet pre-training model can play a more important role. Therefore the ImageNet dataset performs better.

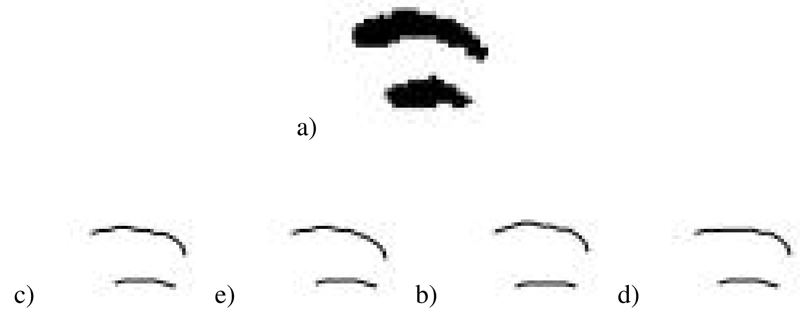

Figure 5: Original images of simple characters and restored images after masking with different probabilities. a) The original image a) The mask probability is 0.05. b) The mask probability is 0.08. c) The mask probability is 0.12. d) The mask probability is 0.15

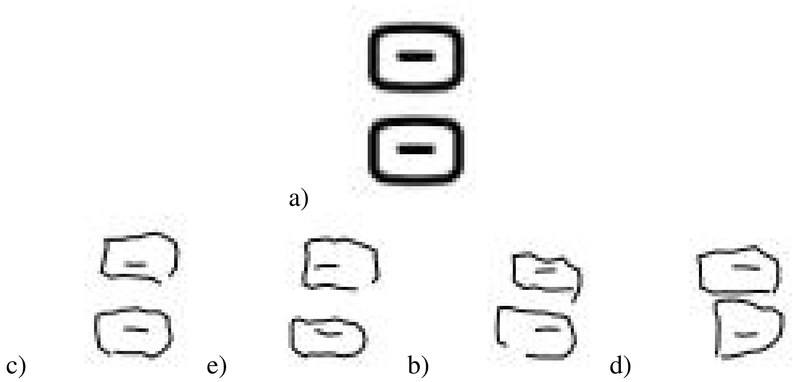

Figure 6: Original images of medium complexity characters and restored images after masking with different probabilities. a) The original image a) The mask probability is 0.05. b) The mask probability is 0.08. a) The mask probability is 0.12. a) The mask probability is 0.15

5.5 Other Augmentor

In the previous experimental method, Orc-Bert enlarges each picture by 10 times, while DA only modify on original data, not to expand the data set. This leads to a variable between the two data augmentors that is not reasonably controlled. Therefore, we explored the impact of different augmentors, excluding the difference in data set size.

The specific methods are as follows: for no DA, repeat each image for 10 times. For DA, a variety of random image argumentation are used to generate 10 different instances for each input sample.

The classification accuracy of the model after correcting the augmentors is shown in the table 6. It can be seen that the classification accuracy of the model is further improved regardless of the shot setting and data augmentation setting.

For no DA, scaling up the same dataset on the other hand means increase the number of training epochs which resulting in better convergence. For DA, expanding the data set using data augmentation not only means increasing the number of training epochs, but also improving the diversity of training samples, further alleviating the problem of overfitting.

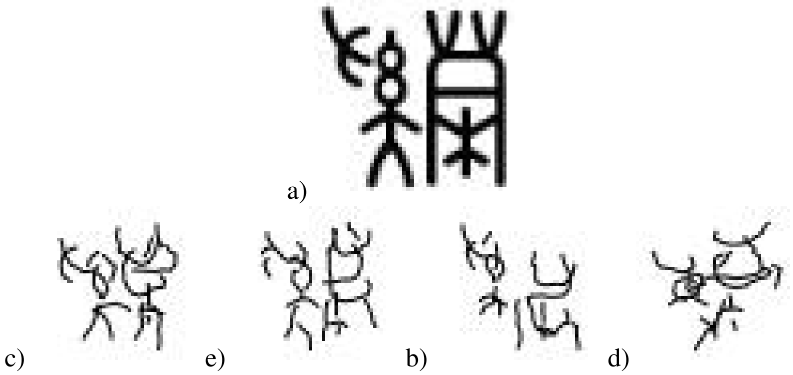

Figure 7: Original images of complex characters and restored images after masking with different probabilities. a) The original image a) The mask probability is 0.05. b) The mask probability is 0.08. a) The mask probability is 0.12. a) The mask probability is 0.15

setting	No DA	DA	Orc-Bert Augmentor +PA
1-shot	41.6	49.1	44.5
3-shot	68.4	72.5	69.9
5-shot	79.1	82.2	80.0

Table 6: Recognition accuracy (%) on Oracle-FS under all one-shot settings for ResNet-18 equipped with corrected No DA, Conventional DA and and Orc-Bert Augmentor+PA. Here, No DA repeat each data for 80 times. DA augment each input sample by random horizontal flipping and random rotation with a degree in $[15^\circ, 15^\circ]$], and generate 10 augmented instances for each samples.

5.6 Classifier Network

Now we explore the effect of the network structure of CNN-based classifier. In this part, we control other variables are equal and test different networks' classification accuracy. For data augmentation, we combine traditional DA with Orc-BERT + PA. For training hyparameters, the number of epochs is 200, learning rate is 1×10^{-4}, batch size is 10. We choose the cosine learning rate scheduler, and the parameters are pre-trained from ImageNet.

There are three different CNNs in our experiments: VGG-19, ResNet-18, and Wide ResNet-101.

setting	VGG-19	ResNet-18	Wide ResNet-101
1-shot	44.1	50.7	**52.9**
3-shot	66.7	74.9	**75.8**
5-shot	76.5	83.4	**84.3**

Table 7: Recognition accuracy(%)) on Oracle-FS under all three few-shot settings for VGG-19, ResNet-18, and Wide ResNet-101. All models are trained in the same hyper-parameter setting. CNN parameters are pre-trained from ImageNet. Traditional DA and Orc-BERT + PA are both applied.

The results indicate that the stronger the network's feature extracting ability is, the higher classification accuracy is. Though having much less learnable parameters than VGG-19, ResNet-18 has stronger feature extracting ability and is easier to optimize, which results in better performance. Wide ResNet-101's feature extracting ability beats ResNet-18, and 200 training epochs are enough for Wide ResNet-101 to reach higher accuracy than ResNet-18.

Note that in Han et al. (2020), ResNet-18 beats deeper residual nets, such as ResNet-50 and ResNet-152. This reason maybe there are not enough gap between ResNet-18 and its deeper variants. In this case, deeper layers and more parameters are burden for training. However, Wide ResNet-101's stronger ability outweighs its disadvantage of more parameters.

We also see that there are just small gaps between the results of ResNet-18 and those of Wide ResNet-101. Furthermore, the gaps are smaller in 3-shot and 5-shot cases than 1-shot case. This is because in 3-shot and 5-shot cases, the marginal effect of model architecture's advantage is weakened, and our model is approaching its bottleneck.

6 Conclusion

In this paper, we limit the training of few-shot models to only use large-scale unlabeled source data and a small number of labeled target training samples. We use transfer learning based on the HWDB dataset to greatly improve the baseline of the model's classification accuracy.

On the basis of transfer learning, we conduct experiments on traditional image argumentation and novel Orc-Bert stroke Augmentor, and obtain completely different experimental results. We analyzed the results according to the experiments. We propose that image augmentation and stroke augmentation can be combined, and also demonstrate that our method is broadly effective under various network settings.

Our experiments have made great progress, substantially refreshing the current state-of-art classification accuracy. However, we believe that this method still has can be improved. On the one hand, Orc-Bert's ability to learn strokes still needs to be improved, especially for complex characters, which it seems unable to handle. On the other hand, due to the limitation of computing resources, we only select a few data for pre-training on the HWDB dataset. While this already appears to have brought an exceptionally large gain, we believe that our model could be further improved by using more or even all of the data.

References

Y. Assael, T. Sommerschield, B. Shillingford, M. Bordbar, J. Pavlopoulos, M. Chatzipanagiotou, I. Androutsopoulos, J. Prag, N. de Freitas, Restoring and attributing ancient texts using deep neural networks, Nature 603 (2022) 280 – 283.

Y. Wang, Q. Yao, J. T.-Y. Kwok, L. M. Ni, Generalizing from a few examples: A survey on few-shot learning, arXiv: Learning (2019).

W. Han, X. Ren, H. Lin, Y. Fu, X. Xue, Self-supervised learning of orc-bert augmentor for recognizing few-shot oracle characters, in: ACCV, 2020.

K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556 (2015).

K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 770–778.

J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

H. Lin, Y. Fu, X. Xue, Y.-G. Jiang, 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, 2020, pp. 6758–6767.

C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, Casia online and offline chinese handwriting databases, 2011 International Conference on Document Analysis and Recognition (2011) 37–41.

Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942 (2019).

T. Devries, G. W. Taylor, Improved regularization of convolutional neural networks with cutout, ArXiv abs/1708.04552 (2017).

H. Zhang, M. Cissé, Y. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, ArXiv abs/1710.09412 (2018).

S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. J. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 6022–6031.