Neural Network and Deep Learning Final Project: Unsupervised Few-Shot Oracle Character Recognition

Miaopeng Yu
School of Data Science
Fudan University
19307130203@fudan.edu.cn

Yi Shao
School of Data Science
Fudan University
19307130113@fudan.edu.cn

Abstract

Oracle character recognition is a challenging task due to the data limitation and imbalance, and the high degree of intra-class variance in the shapes of oracle characters. In this project, we try to solve k-shot learning tasks for three different settings of $k = 1, 3, 5$. We first utilize a large-scale of unlabeled source data to train an Orc-Bert data augmenter, and use it to augment the original dataset. After that, we try two classifiers to recognize the oracle characters. One is the classical classifier like ResNet. The other is our proposed classifier, which is a combination of Content-Based Image Retrieval and Siamese Network. Both of these two models show good results in recognizing the oracle characters. For the classical classifier ResNet-18, the top-1 accuracy for all 200 classes are 40.88%, 66.83% and 76.55% for $k = 1, 3, 5$, respectively. For our proposed model, the results also indicate that this model may lead to higher accuracy for recognizing the oracle characters.
The code's repo[1] and our trained model[2] are provided below in the footnotes.

1 Introduction

1.1 Background

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 [1] Although there is a long history of the Oracle characters, there is only a limit number of Oracle bones, which causes the long-tail problem in the usage of characters. Some characters suffer from the problem of data limitation and imbalance. Therefore, it is naturally to think of the Oracle character recognition problem as a few-shot learning problem, which is a topical task in computer vision and machine learning communities these days [2].

Besides, due to the high degree of intra-class variance in the shapes of Oracle characters, the Oracle character recognition is still a challenging task. And 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 [3].

The first possible way to solve this task is to find a better augmentor. Since 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. So, in standard few-shot learning setting, where one of the most popular strategies is geometric augmentation which including scaling, rotation and

[1] GitHub Repo: https://github.com/Tequila-Sunrise/FDU-NNDL-Final
[2] Best Model: https://github.com/Tequila-Sunrise/FDU-NNDL-Final/releases/tag/best-model

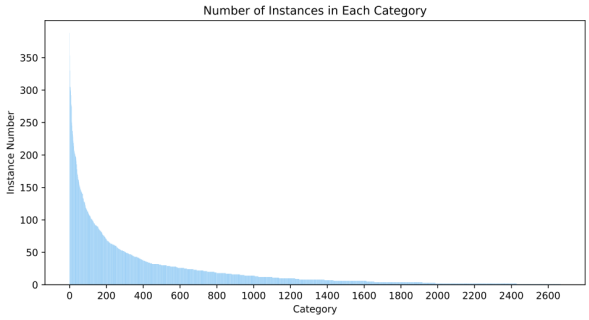


Figure 1: Distribution of Oracle Characters

perspective transformation, these methodologies are designed for pixel format images and only perform a global transformation, thus cannot be used to train a robust classifier efficiently.

Another possible way is to find a better network to solve this task. Instead of using just CNNs as our classifier, can we compare the labeled training data with the test data, and assign a label to the test data according to the training data that is the most similar? So inspired by the application of Content-Based Image Retrieval (CBIR), which finds the most similar images by calculating the distance of their features, we try to use a structure that also has the similar idea. And it is quite similar to Siamese Network [4].

1.2 Potential Applications

As we mentioned above, only about 2,000 out of 4,500 unique Oracle characters have been successfully deciphered, so there is still some obstacle to deciphering the notes written in Oracle characters. These notes may contain a lot of important historical information, which can help historians better understand the events that happened in that period. It can also help the development of the linguistic by understanding these Oracle characters. Therefore, these can be regarded as a treasure for the world.

It will be expensive to acquire a large dataset with labels, and when we want to train our own dataset, we can hardly get a large number of images. Therefore, sometimes it does not assume the existence of large-scale labeled source data, and we can only train the models with a small dataset. Therefore, sometimes it will be more practical to apply few-shot learning in the real world scene, and make the training more efficient.

1.3 Organization

In this project, we solve k-shot learning tasks for three different settings of k. We first replicate the Orc-Bert to augment the original dataset. After that, we use both the original and augmented dataset to train two classifiers. The first one is the classical classifiers, which are CNNs like ResNet [21]. The second one is our proposed classifier, which is a combination of CBIR and Siamese Network. Both of these models achieve quite promising results.

The Organization of this report is as follows. In section 2, we introduce some related works, and review some state-of-the-arts. In section 3, we introduce the Oracle characters datasets we use in this project. In section 4, we overview the some frameworks, like the Orc-Bert, the workflow of CBIR and Siamese Network. After that, we give our proposed network and its pseudo-code in a nut shell. In section 5, we shows the implementation details of the training process for both classifiers. After that, in section 6, we show and analyze the results of our models, and discuss the feasibility of our proposed methodology. At last, in section 7, we give our conclusion, and discuss the possible improvement for the models in the future.

2 Related Work

We review methods for Oracle character recognition, highlight key architectures for the Few-Shot Learning, state-of-the-art data augmentations, and connect these insights to our focus on Few-Shot Learning across self-supervised generated augmentation.

Oracle Character Recognition. Oracle character is one kind of the earliest hieroglyphics, which can be dated back to Shang Dynasty in China. It is of significant impact to recognize such characters since they can provide important clues for modern archaeology, ancient text understanding, and historical chronology, etc. To overcome the data insufficiency and class imbalance of training data, Zhang et al. applied a convolutional neural network to map the character images to an Euclidean space where the distance between different samples can measure their similarities, such that classification can be performed by the Nearest Neighbor rule [6], and Li et al. proposed to design the mix-up strategy that leverages information from both majority and minority classes to augment samples of minority classes such that their boundaries can be pushed away towards majority classes [7], while recently Gao et al. employed a method of image translation from Oracle characters to modern Chinese characters based on generative adversarial network to capture the implicit relationship between them [8].

Few-Shot Learning. Few-Shot Learning is an example of meta-learning, where a learner is trained on several related tasks, during the meta-training phase, so that it can generalize well to unseen (but related) tasks with just few examples, during the meta-testing phase. An effective approach to the Few-Shot Learning problem is to learn a common representation for various tasks and train task specific classifiers on top of this representation. Following this way, Finn et al. proposed an algorithm for meta-learning that is model-agnostic, in the sense that it is compatible with any model trained with gradient descent and applicable to a variety of different learning problems, including classification, regression, and reinforcement learning [9]. While on the other way, Chen et al. combine a meta-learner with an image deformation sub-network that produces additional training examples, and optimize both models in an end-to-end manner, in which the deformation sub-network learns to deform images by fusing a probe image that keeps the visual content and a gallery image that diversifies the deformations [10].

Data Augmentation. Data augmentation involves techniques used for increasing the amount of data, based on different modifications, to expand the amount of examples in the original dataset. Data augmentation not only helps to grow the dataset but it also increases the diversity of the dataset. When training machine learning models, data augmentation acts as a regularizer and helps to avoid overfitting. Random geometric augmentation such as rotation, scaling, and perspective transformation, is a popular way and commonly used in classification models trained on natural images [11]. Besides, some works propose to generate samples by using image interpolation [12] or combination and generative adversarial network [13], which suffers from producing augmented images very different from original images.

Moreover, Sketch-BERT [15] extends BERT [14] with sketch embedding and learns sketch representation from vector format data by self-supervised learning of Sketch Gestalt. Also, self-supervised pre-training task of both allows utilization of large-scale unlabeled data, which is suitable for our hard and practical setting.

3 Datasets

Oracle-20k [16] and OBC306 [17] are two currently known datasets but unfortunately unpublic. Oracle-20k consists of 20,039 character-level samples covering 261 glyph classes, where 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 follow the Oracle datasets proposed by Han et al. [2], Oracle-50K, with 2,668 unique characters. 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.

Oracle-50K. Oracle character instances are collected from three data sources using different strategies, shown in Table 1. Instances from Xiaoxuetang Oracle is collected by our developed crawling tool, wherein there are 24,701 instances of 2,548 individual characters. However, some instances are not provided a corresponding label represented by one single modern Chinese character, thus the

author only remain the deciphered instances with 13,255 instances of 1096 categories in Oracle-50K. Koukotsu is a digital Oracle character and text database. Then they utilize the TrueType font file obtained from Koukotsu to generate 18,671 annotated Oracle character images belonging to 1850 classes. Chinese Etymology4 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.

Table 1: Statistics of Oracle-50K and other ancient Chinese character datasets

	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 Characters	Font Rendering	221947	/

Oracle-FS. Based on Oracle-50K, the author also create a few-shot Oracle character recognition dataset under three different few-shot settings (see Table 1). 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 project, we set k=1, 3, 5.

Table 2: Statistics of Oracle-FS

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

4 Overview of Framework

4.1 Orc-Bert Replicate

4.1.1 Overview of Orc-Bert

As shown in Figure 2, the Orc-Bert framework consists of the following parts.

First, Orc-Bert utilize the online approximation algorithm in [18] to convert offline Oracle character images with annotations and other Chinese ancient characters images without annotations, both in pixel format, to online data in 5-element vector format, 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 $\Delta x, \Delta y$ is the position offsets between two adjacent points, and (p_1, p_2, p_3) is a 3-dimension one-hot vector $\left(\sum_{i=1}^{3} p_i = 1\right) \cdot p_2 = 1, p_3 = 1, p_1 = 1$ 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 have to pre-train Orc-Bert in a self-supervised setting by predicting the masked from the visible. Then, we utilize the pre-trained Orc-Bert as the 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.

Different from the original paper, in order 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, and then apply conventional augmentations like random rotation or normalization methods, notice that horizontal flip is also performed because we find some flipped sample pairs in the original dataset.

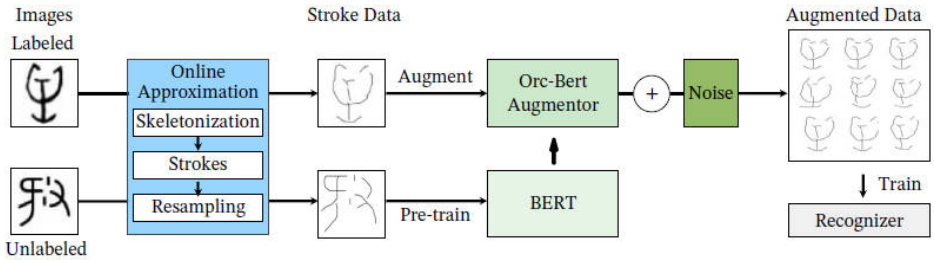


Figure 2: Schematic illustration of Orc-Bert

The augmented dataset we used to train our model is available at `https://github.com/Tequila-Sunrise/FDU-NNDL-Final/releases/tag/data-augmentation`.

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

4.1.2 Orc-Bert Augmentor

Self-supervised Pre-training of BERT and SketchBERT. BERT [14], 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 [15] 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 replicate 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; we add a module after the output layer to convert point sequence to pixel format image; we corrupt input for diverse augmentation.

Pre-training. Pre-training tasks over unlabeled data significantly facilitate the performance of BERT [14] as well as SktechBERT [15]. 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 $O_{\text{mask}} = O \odot M$, 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. The samples of the results is shown in Figure 3.

4.2 Proposed Classifier

4.2.1 Overview of Content-Based Image Retrieval (CBIR)

Content-Based Image Retrieval (CBIR) is the application of computer vision techniques to the image retrieval problem. It analyzes the contents of the image, like colors, shapes, textures, and any other information that can be derived from the image itself.

For all images in the database, a feature extractor will be applied to them, and get feature vectors to represent these images. When a user want to search for one object using an image, the system will extract image features for this query, and compare these features with that of other images in a database according to their distance, then rank the distance and output the top-k results. The process is shown below in Figure 4.

Figure 3: Samples of Orc-Bert Augmentor

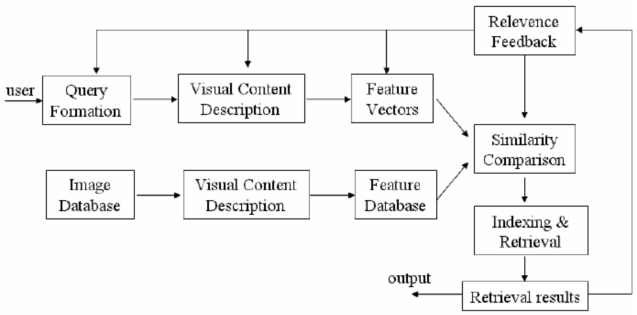


Figure 4: Workflow of CBIR

4.2.2 Overview of Siamese Network

The idea of Siamese Network is quite similar to the one of CBIR's. It also extract the features of two images, and compare their features' distance. A choice for the extractor is use CNNs without the last fully connected layer. And the illustration for the Siamese Network is shown below in Figure 5.

Siamese is more robust to class imbalance. Giving a few images per class is sufficient for Siamese Networks to recognize those images in the future with the aid of few-shot learning. But it also takes a longer time to train than normal networks.

4.2.3 Our Network

Inspired by the workflow of CBIR and Siamese Network, we combine the two frameworks together to train our model. First, we train the model like the original Siamese Network, using Triplet loss. After training, we use the feature extractor to extract the training samples' features. For each test sample, also extract the feature, use it as a query to find the closest training sample's feature, and assign the label of this training sample to this test sample.

The following pseudo-code shows the workflow of our model.

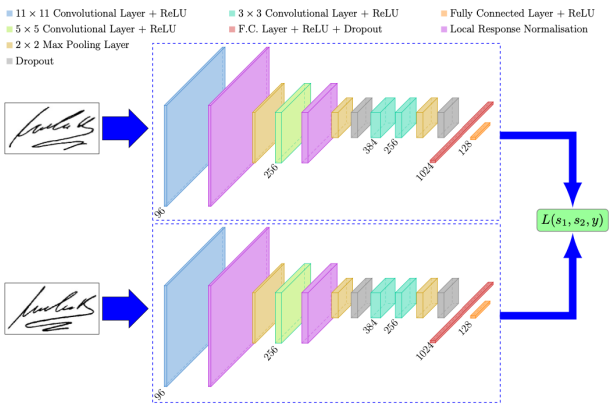


Figure 5: Workflow of Siamese Network [5]

Algorithm 1 Combining CBIR and Siamese Network

1. Train Siamese Network using triplet loss.
Choose the triples A (Anchor), P (Positive), N (Negative), where A and P are the same character, A and N are different characters.
Loss: $J = \sum_{i=1}^{M} \mathcal{L}(A, P, N) = \sum_{i=1}^{M} \max(0,\ \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha)$

2. Take the feature extractor of Siamese network as $f(\cdot)$.

3. Get prediction.
$\mathcal{D}_{train+feature} = \{X_{train}, f(X_{train}), y_{train}\}$
for X_{test} in $\mathcal{D}_{test} \doteq \{X_{test}\}$ **do**
$fea = f(X_{test})$
$i_{nearest} = \arg\min_i \|fea - f(X_{train,i})\|$
$y_{pred} = y_{train,i_{nearest}}$
end for

5 Implementation Details

To augment the data, we will train the Orc-Bert Augmentor in unlabled Oracle-50K through self-supervised methods.And for the classifier, we train two kinds of classifiers. One is the classical classifier, like ResNet [21], DenseNet [22] or other CNNs networks. Another one is our proposed classifier, which is a combination of Siamese Network and CBIR.

5.1 Augmentor and Classical Classifier

We follow the settings in the original paper of Orc-Bert [2], 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. In addition, cosine scheduler is employed to update learning rate for optimizer.

5.2 Proposed Classifier

For the training of Siamese Network, we choose the A (Anchor), P (Positive), N (Negative) randomly, where A and P are the same character, A and N are different characters. Then, we need to minimize

the triplet loss J.

$$\mathcal{L}(A, P, N) = \max(0,\ \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha),$$

$$J = \sum_{i=1}^{M} \mathcal{L}(A, P, N),$$

where M is the total number of triplets.

For different k, we train the k-shot task using SGD with a momentum of 0.9, and weight decay of 5×10^{-4}. We start with a initial learning rate of 0.05, and reducing the learning rate using cosine annealing, the minimum learning rate is 0.01 of the initial learning rate. We train the model for 200 epochs, and a batch size of 32. The feature extractor we choose is the VGG-16 [19] pre-trained on ImageNet [20].

6 Results

6.1 Orc-Bert Replicate

In this part, we evaluate our trained Orc-Bert Augmentor on Oracle-FS task. 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.

After Orc-Bert augmentation, we perform conventional data augmentation (CDA) including random resize scaling, random horizontal flip, random rotation and normalization, the comparison is listed in Table 3.

Table 3: Comparison between different augmentation strategies

Setting	Model	No DA	CDA	Orc-Bert	Orc-Bert+PA	Orc-Bert+CDA(Ours)
	ResNet-18	18.6	20.9	29.5	31.9	**40.88**
	ResNet-50	16.8	23.3	26.2	29.9	
1 shot	ResNet-152	14.0	18.2	26.7	27.3	
	DenseNet	22.4	24.6	26.4	28.2	
	ResNet-18	45.2	46.6	56.2	57.2	**66.83**
	ResNet-50	35.8	45.6	52.9	57.7	
3 shot	ResNet-152	38.9	40.9	54.3	57.1	
	DenseNet	48.6	52.3	56.4	58.3	
	ResNet-18	60.8	62.7	65.1	68.2	**76.55**
	ResNet-50	55.6	60.8	62.8	67.9	
5 shot	ResNet-152	58.6	61.4	66.1	67.8	
	DenseNet	69.3	65.8	66.6	69.0	

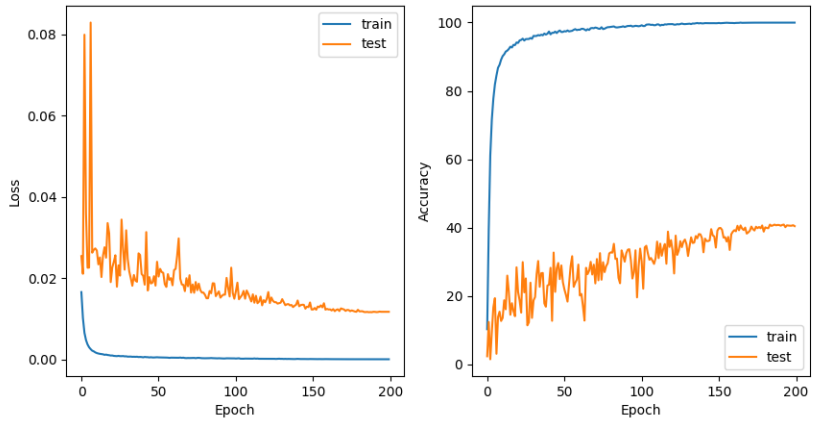
The plot of the training process is shown below in Figure 6, from which we can observe the curves on test set all fluctuate sharply in the preliminary stage and smooth out gradually in the end.

6.2 Our Proposed Model

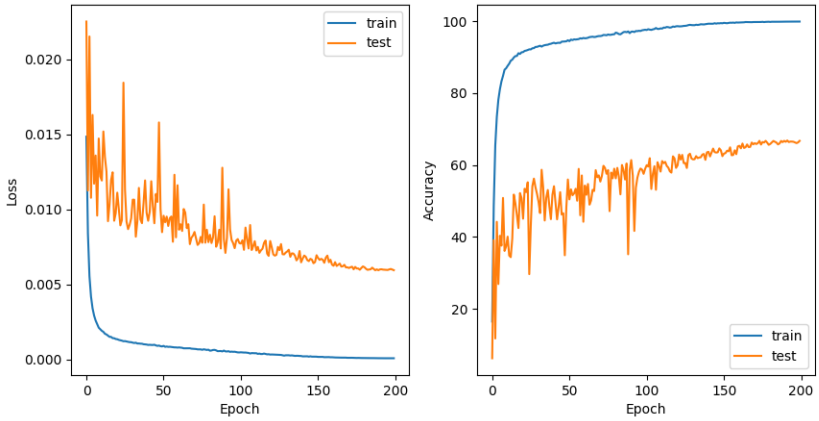
Due to time-limited, after training the Siamese Network, we only test the accuracy of the Siamese Network. In other words, we test the model's ability to figure out if two images are the same characters. So, the accuracy of our proposed classifier is a little bit different from the classification accuracy. But since we think the result is quite promising, which indicates that this model may lead to a higher accuracy of the Oracle character few-shot problems, we report our half-accomplished results here too[3].

The plot of the training loss and the test loss is shown below in Figure 7.

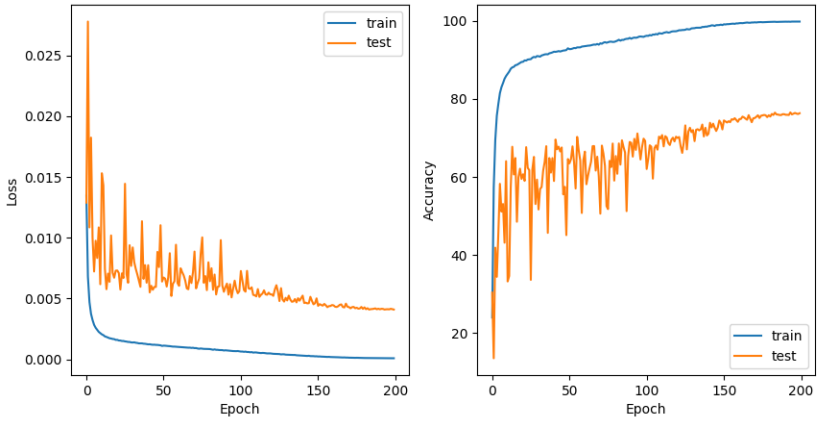
[3]To be noticed, the accuracy we show here (our proposed model) is the accuracy of judging whether two images belong to the same Oracle character. The accuracy we shown before (in last subsection) is the real top-1 classification accuracy.



(a) 1-shot



(b) 3-shot

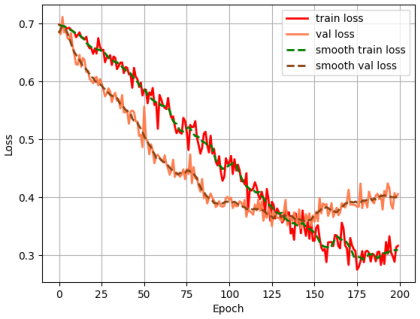


(c) 5-shot

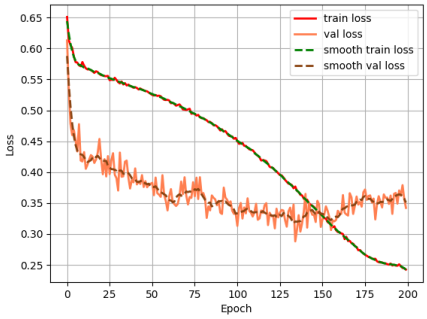
Figure 6: Training Process of ResNet-18 Model with Orc-Bert+CDA

Table 4: Accuracy of Our Proposed Model

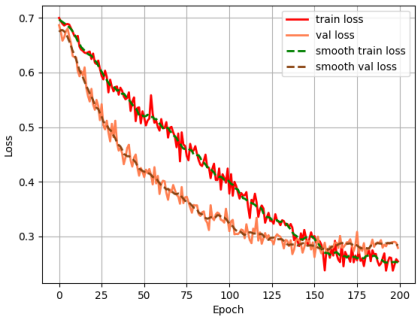
task	augmentation	accuracy
3-shot	CDA	86.0
	CDA + Orc-Bert	88.2
5-shot	CDA	89.9
	CDA + Orc-Bert	90.3 (150 Epochs)



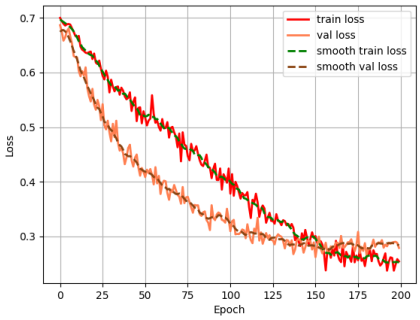
(a) 3-shot with CDA



(b) 3-shot with CDA and Orc-Bert



(c) 5-shot with CDA



(d) 5-shot with CDA and Orc-Bert (150 Epochs)

Figure 7: Loss of Our CBIR Model

We can see from the result that the accuracy of Siamese Network is pretty good considering that there are 200 classes in the dataset, and each class has only a few training samples. So we think our proposed model will have a promising outcome. And by applying the Orc-Bert, the test accuracy improves further. This is probably because by applying the Orc-Bert, the training become more robust, and the loss will not oscillate too much, which we can clearly see in the plot above.

7 Conclusion and Further Improvements

In this project, we finish the k-shot learning task for different k settings. We first replicate the Orc-Bert, and then use two kinds of classifiers - classical classifiers like CNNs and our proposed classifier. And we reach a quite promising result using these models.

For further improvement, we can first finish implementing our proposed model - a combination of CBIR and Siamese Network. Since this model is quite accurate at figuring out if two images are the same Oracle character, it should also be good at classification given the training set as the database, too.

Also, since we use VGG-16 as our feature extractor, the feature we can get may be a little bit shallow. So, we can try to change the extractor to some deeper CNNs like ResNet-101, DenseNet, Deep Layer Aggregation (DLA) [23], etc.. For DLA, this model can better fuse semantic and spatial information for recognition and localization [23], which maybe useful for the classification of the Oracle characters.

References

[1] Keightley, D. N. (1997). Graphs, words, and meanings: Three reference works for Shang Oracle-bone studies, with an excursus on the religious role of the day or sun.

[2] Han, W., Ren, X., Lin, H., Fu, Y., & Xue, X. (2020). Self-supervised Learning of Orc-Bert Augmentator for Recognizing Few-Shot Oracle Characters. In Proceedings of the Asian Conference on Computer Vision.

[3] Huang, S., Wang, H., Liu, Y., Shi, X., & Jin, L. (2019, September). Obc306: a large-scale Oracle bone character recognition dataset. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 681-688). IEEE.

[4] Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).

[5] Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Lladós, J., & Pal, U. (2017). Signet: Convolutional siamese network for writer independent offline signature verification. arXiv preprint arXiv:1707.02131.

[6] Zhang, Y. K., Zhang, H., Liu, Y. G., Yang, Q., & Liu, C. L. (2019, September). Oracle character recognition by nearest neighbor classification with deep metric learning. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 309-314). IEEE.

[7] Li, J., Wang, Q. F., Zhang, R., & Huang, K. (2021, September). Mix-Up Augmentation for Oracle Character Recognition with Imbalanced Data Distribution. In International Conference on Document Analysis and Recognition (pp. 237-251). Springer, Cham.

[8] Gao, F., Zhang, J., Liu, Y., & Han, Y. (2022). Image Translation for Oracle Bone Character Interpretation. Symmetry, 14(4), 743.

[9] Finn, C., Abbeel, P., & Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning (pp. 1126-1135). PMLR.

[10] Chen, Z., Fu, Y., Wang, Y. X., Ma, L., Liu, W., & Hebert, M. (2019). Image deformation meta-networks for one-shot learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8680-8689).

[11] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 113-123).

[12] DeVries, T., & Taylor, G. W. (2017). Dataset augmentation in feature space. arXiv preprint arXiv:1702.05538.

[13] Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., & Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2107-2116).

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

[15] 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).

[16] Guo, J., Wang, C., Roman-Rangel, E., Chao, H., & Rui, Y. (2015). Building hierarchical representations for Oracle character and sketch recognition. IEEE Transactions on Image Processing, 25(1), 104-118.

[17] Huang, S., Wang, H., Liu, Y., Shi, X., & Jin, L. (2019, September). Obc306: a large-scale Oracle bone character recognition dataset. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 681-688). IEEE.

[18] Mayr, M., Stumpf, M., Nicolaou, A., Seuret, M., Maier, A., & Christlein, V. (2020, August). Spatio-temporal handwriting imitation. In European Conference on Computer Vision (pp. 528-543). Springer, Cham.

[19] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[20] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[21] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[22] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[23] Yu, F., Wang, D., Shelhamer, E., & Darrell, T. (2018). Deep layer aggregation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2403-2412).