Orc-DeBERTa, Orc-MAE and others: Unsupervised Few-Shot Oracle Character Recognition

Yiqun Wang
School of Data Science
Fudan University
Shanghai, 200433
yiqunwang19@fudan.edu.cn

Zichen Cheng
Fudan University
School of Data Science
Shanghai, 200433
zichencheng19@fudan.edu.cn

Abstract

Oracle characters are the earliest known hieroglyphs in China, and are important for modern archaeology, history, Chinese etymology and calligraphy study. However, nowadays, oracle character recognition is still undeveloped due to the scarcity of oracle bones, the long-tail problem in the usage of characters and the high degree of intra-class variance in the shapes of oracle characters. Therefore, to deal with the task of oracle character recognition, data augmentation strategies are quite significant. In this paper, we introduce 2 strategies based on deep learning: Orc-DeBERTa, which combines the Orc-BERT and DeBERTa model, and Orc-MAE, which combines the Orc-BERT and MAE model; we also summarize 3 strategies based on image pre-processing: CutOut, MixUp and CutMix. Based on these data augmentation strategies, we use pre-trained ResNet-18 as the classifier to get the final outputs of oracle character recognition. We fine-tune and test CutOut, MixUp and CutMix on the Oracle-FS dataset under the self-supervised and few-shot settings, but not our Orc-DeBERTa and Orc-MAE due to the limitation of computational resources. Experiments show that all of the results exceeds the state of the art. Our code has been released on https://github.com/quniLcs/nndl.

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 of weather, state power, warfare and trading to mitigate uncertainty in the Shang dynasty [Keightley, 1997]. Oracle characters are important for modern archaeology, history, Chinese etymology and calligraphy study. [Guo et al., 2016, Zhang et al., 2019]

In the past decades, although identification and decipherment for oracle characters have made huge strides, there is still a long way to fully understand the whole writing system. So far, more than 150,000 animal bones and turtle shells had been excavated, including approximately 4,500 unique oracle characters, but only about 2,000 of them have been successfully deciphered [Huang et al., 2019]. 2 main reasons are as follows:

Due to the scarcity of oracle bones and the long-tail problem in the usage of characters as shown in Fig. 5, oracle character recognition suffers from the problem of data limitation and imbalance, thus is a natural few-shot learning problem, which is topical in computer vision and machine learning communities recently.

Besides, as is shown in Fig. 1, 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 vari-

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

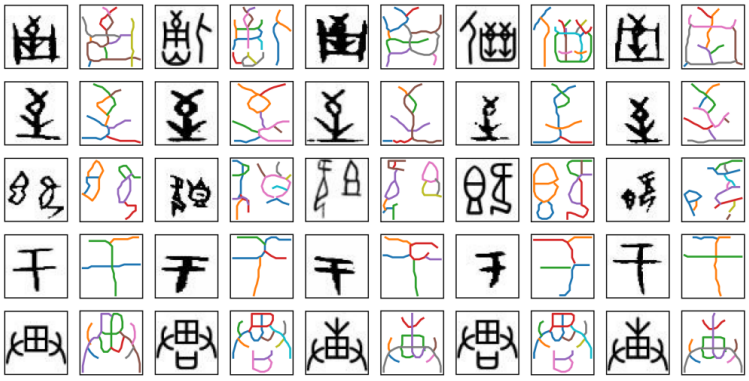


Figure 1: Examples of oracle character images and corresponding stroke data.

ous regions over tens of hundreds of years. As a result, oracle character recognition is a challenging task.

In this paper, we intend to address the problem of oracle character recognition under self-supervision and few-shot settings. More specifically, we will utilize a large-scale unlabeled source data as well as a few labeled training samples for each category to train our model by transferring knowledge.

2 Related Works

Sketch Data Processing Unlike MNIST handwritten digit database [LeCun et al., 2010], which is in pixel form, oracle data or sketch data are always processed in vector form [Lin et al., 2020], where we use a 5-dimensional vector to show each point in a sketch:

$$O = (\Delta x, \Delta y, p_1, p_2, p_3) \tag{1}$$

In this form, $\Delta x, \Delta y$ are continuous values, which stand for the position offset between two adjacent points, while p_1, p_2, p_3 are 0 or 1 and sums to 1, where $p_2 = 1$ indicates that the point is at the end of a stroke, and $p_3 = 1$ indicates that the point is at the end of the whole character. Based on this, quite a lot of works have sprung up to process sketch data using deep neural networks. For example, Sketch-a-Net implements 2 novel data augmentation strategies as well as network ensemble fusion strategies to deal with the task of sketch recognition [Yu et al., 2017]; Sketch-RNN learns a generative neural representation for sketches by Long Short Term Memory networks (LSTM) [Ha and Eck, 2017]; Sketch-R2CNN uses an RNN for stroke attention estimation in the vector space, followed by a CNN for 2D feature extraction in the pixel space, also to deal with the task of sketch recognition [Li et al., 2018]; TC-Net uses triplet Siamese network and auxiliary classification loss to deal with the task of sketch retrieval [Lin et al., 2019]. Among them, Sketch-BERT is the state-of-the-art, which adopts BERT as its backbone [Lin et al., 2020]. It gets sketch embeddings as the sum of point embeddings, position embeddings and stroke embeddings, and pre-trains on a novel self-supervised learning task, sketch Gestalt task, including mask position prediction and mask state prediction. It can be fine-tuned and tested on downstream tasks, such as sketch recognition, when it adds a `[CLS]` label to the beginning of the sequential data of each sketch, serves as a generic feature extractor of each sketch and adds a standard softmax classification layer at the end.

Oracle Data Processing As a sub-domain of sketch data processing, quite a lot of works also have sprung up, however pays less attention to deep neural networks. For example, Guo et al. [2016] propose a novel hierarchical representation that combines a Gabor-related low-level representation and a sparse-encoder-related mid-level representation; [Meng, 2017] uses the line feature to deal with the task of oracle character recognition; [Zhang et al., 2019] extract features by a convolutional neural network and perform classification by the Nearest Neighbor algorithm also to deal with the task of oracle character recognition; [Xing et al., 2019] present a unified implementation of the Faster R-CNN, SSD, YOLOv3, RFBnet and RefineDet to deal with the the task of oracle character detection; [Meng et al., 2019] extend the SSD model also to deal with the the task of oracle character

detection. Among them, Orc-BERT is the state-of-the-art model in oracle character recognition and is under similar settings as our work [Han et al., 2021]. First, Orc-BERT is pre-trained on a large-scale unlabeled dataset under self-supervision settings by predicting the masked from the visible. Then, a convolutional neural network based classifier is trained under few-shot learning settings with Orc-BERT as the data augmentor.

Language Representation Models Since BERT is introduced, quite a few works intend to improve it [Liu et al., 2019, He et al., 2020], which we can also use to improve the Orc-BERT model. DeBERTa is one of them, which implements Disentangled Attention and Enhanced Mask Decoder [He et al., 2020, Wolf et al., 2020]. On one hand, while BERTs input is just the sum of token embedding, segment embedding and position embedding, DeBERTa inputs content embedding and position embedding respectively, and the latter represents the relative position between tokens. On the other hand, since Disentangled Attention only captures relative positions instead of the absolute positions, which is also important, those absolute positions should be incorporated after all the Transformer layers and before the softmax layer for Masked Language Modeling (MLM), so that Transformer layers can make better use of those relative positions, while absolute positions can also play a part in the softmax layer.

Computer Vision Self-supervised Models Like language representation models, self-supervised models is also popular in computer vision. Among them, the most representative one is the masked autoencoders (MAE) [He et al., 2021], where random patches of the input image are masked and reconstructed. More specifically, MAE uses an asymmetric encoder-decoder architecture, with an encoder that operates only on the visible patches and a lightweight decoder that reconstructs the original image from the latent representation and mask tokens. Since the oracle character recognition task can be solved either from the perspective of natural language processing (NLP) or from the perspective of computer vision (CV), we can also try out MAE.

3 Approach

Our model contains 2 parts: augmentor and classifier. When we use Orc-DeBERTa as our augmentor, we can input a masked sketch data and get an augmented image; when we use Orc-CM as our augmentor, we can input a masked image data and also get an augmented image; we can also use CutOut, MixUp and CutMix to get augmented image likewise. Then we can input the augmented data into the classifier, which is based on a pre-trained ResNet-18. The whole structure of our model is shown in figure 2.

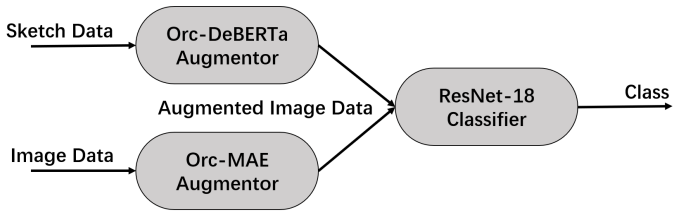


Figure 2: The whole structure of our model.

3.1 Augmentor

To increase the volume and diversity of training data, and to address the challenge of a high degree of intra-class variance in the shapes of oracle characters, especially under the few-shot settings, data augmentation strategies are quite significant.

3.1.1 Orc-DeBERTa

When a sketch is inputted into the Orc-DeBERTa, it is first embedded, then encoded and finally outputs an image.

Embedding In the DeBERTa [Wolf et al., 2020], the input includes the word ID, the token type ID, the position ID and the mask, while in our Orc-DeBERTa, the word ID is not included. Instead, we use a fully-connected network to create the embedding from the vector from sketch data.

Encoding As is mentioned in section 2, DeBERTa inputs content embedding and position embedding respectively. More specifically, we denote $H_i \in R^d$ as the content embedding for a token at position i, and $P_{i|j} \in R^d$ as the position embedding to represent the relative position for the token at position i with the token at position j. Then the cross attention score is:

$$\begin{aligned} A_{i,j} &= (H_i, P_{i|j}) \cdot (H_j, P_{j|i})^T \\ &= H_i H_j^T + H_i P_{j|i}^T + H_j P_{i|j}^T + P_{i|j} P_{j|i}^T \end{aligned}$$

where the 4 terms stand for content-to-content, content-to-position, position-to-content as well as position-to-position respectively, and the last term can be removed since it doesn't provide much additional information. To put all $P_{i|j}$ into a matrix P, we denote k as the maximum relative distance, and $\delta(i,j) \in [0, 2k)$ as the relative distance from the token at position i to the token at position j, where

$$\delta(i,j) = \begin{cases} 0 & i-j \leq k \\ 2k-1 & i-j \geq k \\ i-j+k & |i-j| < k \end{cases}$$

Then the cross attention score is:

$$\tilde{A_{i,j}} = Q_i^c K_j^{cT} + Q_i^c {K_{\delta(i,j)}^r}^T + K_j^c {Q_{\delta(j,i)}^r}^T$$

where

$$\begin{aligned} Q^c &= HW_{q,c} & K^c &= HW_{k,c} \\ Q^r &= PW_{q,r} & K^r &= PW_{k,r} \end{aligned}$$

are projected matrices and $W_{q,c}$ $W_{k,c}$ $W_{q,r}$ $W_{k,r}$ are projection matrices. Finally, the output of self-attention operation is $H_o = softmax(\frac{\tilde{A}}{\sqrt{3d}})V^c$ where $V^c = HW_{v,c}$.

Reconstruction After encoding, we use a fully-connected network to further reconstruct the information and output an image.

3.1.2 Orc-MAE

When an image is inputted into the Orc-MAE, it goes through a modified MAE as shown in figure 3 and outputs an image. More specifically, MAE uses an asymmetric encoder-decoder architecture, with an encoder that operates only on the visible patches and a lightweight decoder that reconstructs the original image from the latent representation and mask tokens.

3.1.3 Others

CutOut To implement CutOut, for every input image during training, choose a random point as the central point of the mask, and set the square area around it to 0 [DeVries and Taylor, 2017].

MixUp MixUp convex combines a pair of input images and labels during training, and the convex combination coefficient follows the Beta distribution [Zhang et al., 2017]. More specifically, for input images x_i, x_j, labels y_i, y_j, and the convex combination coefficient $\lambda \sim Beta(\alpha, \alpha)$, the new image and label are:

$$\begin{aligned} x &= \lambda x_i + (1-\lambda) x_j \\ y &= \lambda y_i + (1-\lambda) y_j \end{aligned}$$

CutMix CutMix combines the above two strategies [Yun et al., 2019]. More specifically, for input images x_i, x_j, labels y_i, y_j, and the convex combination coefficient $\lambda \sim Beta(\alpha, \alpha)$, first choose a random point as the central point from the image x_i, whose size is $H \times W$, and set the square area around it, whose size is $H\sqrt{1-\lambda} \times W\sqrt{1-\lambda}$, to the value of image x_j, which means the ratio of this area is $1-\lambda$. Since the square area may be beyond the boundary of the image, finally set λ to the ratio of area which keeps the initial value of the image x_i, and the new label is:

$$y = \lambda y_i + (1-\lambda) y_j$$

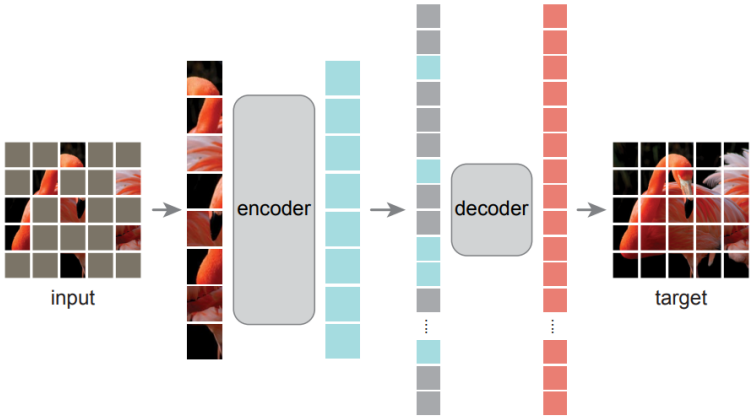


Figure 3: The structure of Orc-MAE.

3.2 Classifier

We use pre-trained ResNet-18 as the classifier [He et al., 2015], which mainly contains two kinds of unit structures as shown in figure 4 and whose output is modified to be the probability of the 200 classes.

4 Experiment

4.1 Dataset

Oracle-50K In this dataset, labeled oracle character samples are collected from three data sources using different strategies [Han et al., 2021]. There are 2668 unique characters and 59081 images in total. Besides, as is shown in Fig. 5, there exists a long-tail distribution of oracle character samples in Oracle-50K. Therefore, oracle character recognition is a natural few-shot learning problem.

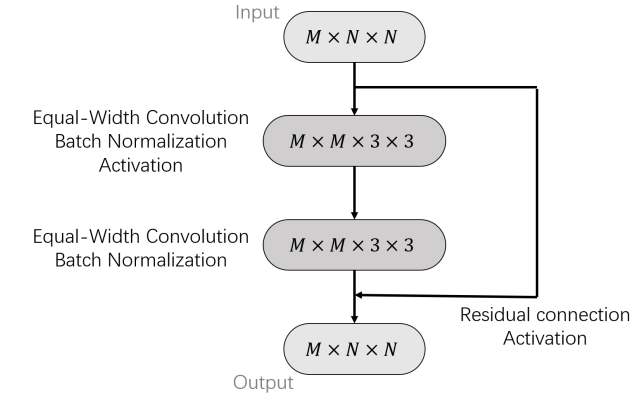
Oracle-FS Based on Oracle-50K and other collected ancient Chinese character images, Han et al. [2021] created a few-shot oracle character recognition dataset, Oracle-FS, including 276,031 images, under three different few-shot settings. Specifically, under the k-shot setting, there are 200 classes, with k training samples and 20 test ones per class, where k can be 1, 3 and 5. Besides, 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 incomplete, Oracle-FS includes both pixel and vector format data. Although the stroke orders of oracle characters have been lost in history, there are two fundamental facts: 1) oracle 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, so assuming oracle character writing is in the same order and using existing approximation algorithm [Mayr et al., 2020], character images in pixel format can be converted to data in vector format. Nevertheless, due to 3 failure cases during approximation algorithm, the number of source samples in vector format are 276,028.

4.2 Data Pre-process

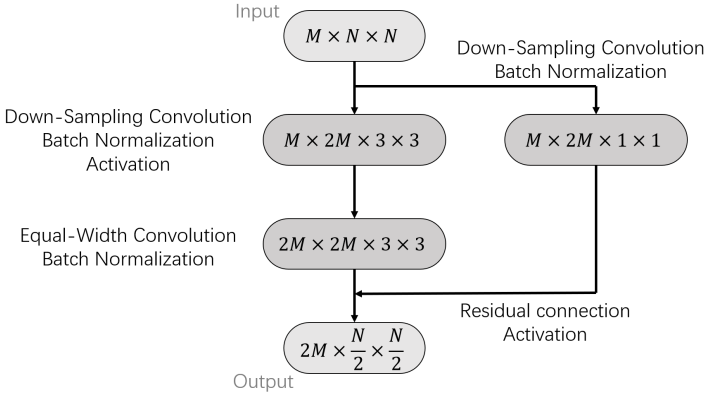
Pixel Form Image Data In Oracle-FS dataset, all images are gray-scale ones, whose pixel values ranging from 0 to 255. Since pixel value 255 means the white background, we get all pixel values be divided by 256 and minus by 1.

Vector Form Sketch Data First, the vector form sketch data in equation 1 is simplified into:

$$O = (\Delta x, \Delta y, p_2 + p_3)$$



(a) The 1st unit structure.



(b) The 2nd unit structure.

Figure 4: Two kinds of unit structures in ResNet-18.

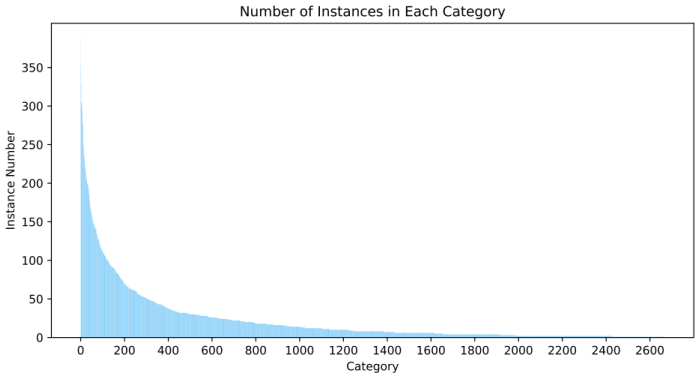


Figure 5: The distribution of oracle character samples in dataset Oracle-50K.

since $p_1 = 1$ at most times. In Oracle-FS dataset, it satisfies:

$$\begin{aligned} \Delta x &\in [-49, 49] \\ \Delta y &\in [-49, 49] \\ p_2 + p_3 &\in \{0, 1\} \end{aligned}$$

To normalize the data, we get:

$$\tilde{O} = (\frac{\Delta x}{49}, \frac{\Delta y}{49}, p_2 + p_3)$$

4.3 Hyper-parameters

During training, the number of epochs is 200, the batch size is 8, the optimizer is Adam, and the learning rate is 0.0001 and 0.001 for augmentor pre-training and classifier training respectively.

As for the structure of the Orc-DeBERTa augmentor, as what Orc-BERT did, the max input length of stroke is 300, the hidden size is 128 and the number of Transformer layers is 8. The embedding and reconstruction networks are fully-connected with structure of 64-128-128 and 128-128-64-5 respectively. The size of augmented images is 50×50, and becomes 224×224 for the classifier.

As for the structure of the Orc-MAE augmentor, similar to Orc-DeBERTa, the hidden size is 128 and the number of Transformer layers is 8. Since the size of input image is 50×50, we set the patch size to be 5×5.

4.4 Results

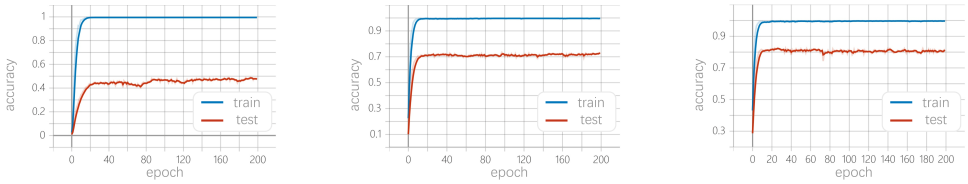
Due to the limitation of computational resources, we only fine-tune and test the baseline model, tradition data augmentation model, as well as CutOut, MixUp and CutMix. The result is shown in table 1.

Table 1: The results of our experiments

Setting	No DA	Tradition	CutOut	MixUp	CutMix
1-shot	0.40100	**0.49075**	0.38600	0.36750	0.36275
3-shot	0.65675	**0.73375**	0.64305	0.64850	0.63250
5-shot	0.76375	**0.82650**	0.75325	0.75875	0.72875

It seems that those fancy data augmentation strategies like CutOut, MixUp and CutMix don't work well on oracle character recognition. One possible reason is that different parts of one Chinese character usually have their own relatively independent meanings, so CutOut, MixUp and CutMix can sometimes output another existing character, instead of just outputting part of the initial character or the combination of two irrelative characters, which confuses the model.

After all, while all of the results exceeds the state of the art, the tradition data augmentation strategies works best, which includes random padding, cropping and horizontal flipping and whose accuracy curve is shown in figure 6.



(a) 1-shot accuracy. (b) 3-shot accuracy. (c) 5-shot accuracy.

Figure 6: The accuracy curves of the the tradition data augmentation strategies.

5 Conclusion

In this paper, we try to deal with the task of oracle character recognition under natural settings, or self-supervised and few-shot settings. Therefore, we introduce 2 strategies based on deep learning: Orc-DeBERTa and Orc-MAE; we also summarize 3 strategies based on image pre-processing: CutOut, MixUp and CutMix. Based on these data augmentation strategies, we use pre-trained ResNet-18 as the classifier to get the final outputs.

Due to the limitation of computational resources, we only fine-tune and test the baseline model, tradition data augmentation model, as well as CutOut, MixUp and CutMix, on the Oracle-FS dataset. Experiments show that all of the results exceeds the state of the art. While those fancy data augmentation strategies like CutOut, MixUp and CutMix don't work well possibly because of the characteristic of Chinese characters, the tradition data augmentation strategies works best, which includes random padding, cropping and horizontal flipping. After all, we look forward to the better performance of our untested Orc-DeBERTa and Orc-MAE once given enough computational resource.

References

David 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. URL `https://www.jstor.org/stable/605249`.

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, 2016. doi: 10.1109/TIP.2015.2500019.

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, 2019. doi: 10.1109/ICDAR.2019.00057.

Shuangping Huang, Haobin Wang, Yongge Liu, Xiaosong Shi, and Lianwen Jin. Obc306: A large-scale oracle bone character recognition dataset. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 681–688, 2019. doi: 10.1109/ICDAR.2019.00114.

Yann LeCun, Corinna Cortes, and Chris Burges. Mnist handwritten digit database, 2010.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv, June 2020. doi: 10.48550/ARXIV.2005.09159. URL `https://arxiv.org/abs/2005.09159`.

Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-net: A deep neural network that beats humans. *International journal of computer vision*, 122(3): 411–425, 2017.

David Ha and Douglas Eck. A neural representation of sketch drawings, 2017. URL `https://arxiv.org/abs/1704.03477`.

Lei Li, Changqing Zou, Youyi Zheng, Qingkun Su, Hongbo Fu, and Chiew-Lan Tai. Sketch-r2cnn: An attentive network for vector sketch recognition, 2018. URL `https://arxiv.org/abs/1811.08170`.

Hangyu Lin, Yanwei Fu, Peng Lu, Shaogang Gong, Xiangyang Xue, and Yu-Gang Jiang. Tc-net for isbir: Triplet classification network for instance-level sketch based image retrieval. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 16761684, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368896. doi: 10.1145/3343031.3350900. URL `https://doi.org/10.1145/3343031.3350900`.

Lin Meng. Recognition of oracle bone inscriptions by extracting line features on image processing. In *ICPRAM*, pages 606–611, 2017.

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*, AIIPCC '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450376334. doi: 10.1145/3371425.3371434. URL https://doi.org/10.1145/3371425.3371434.

Lin Meng, Bing Lyu, Zhiyu Zhang, C. V. Aravinda, Naoto Kamitoku, and Katsuhiro Yamazaki. Oracle bone inscription detector based on ssd. In Marco Cristani, Andrea Prati, Oswald Lanz, Stefano Messelodi, and Nicu Sebe, editors, *New Trends in Image Analysis and Processing – ICIAP 2019*, pages 126–136, Cham, 2019. Springer International Publishing. ISBN 978-3-030-30754-7.

Wenhui Han, Xinlin Ren, Hangyu Lin, Yanwei Fu, and Xiangyang Xue. Self-supervised learning of orc-bert augmentor for recognizing few-shot oracle characters, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2020. URL https://arxiv.org/abs/2006.03654.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017. URL https://arxiv.org/abs/1708.04552.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017. URL https://arxiv.org/abs/1710.09412.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019. URL https://arxiv.org/abs/1905.04899.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Martin Mayr, Martin Stumpf, Anguelos Nicolaou, Mathias Seuret, Andreas Maier, and Vincent Christlein. Spatio-temporal handwriting imitation. In *Computer Vision – ECCV 2020 Workshops*, pages 528–543. Springer International Publishing, 2020. doi: 10.1007/978-3-030-68238-5_38. URL https://doi.org/10.1007%2F978-3-030-68238-5_38.