



B.Sc. (Software Engineering) Degree Program

Stage 4 Project Group Thesis Receipt

Project Title:

Supervisor:

Team Members		
Student Name(s)	UCD Student Number	BJUT Student Number

Plagiarism: the unacknowledged inclusion of another person's writings or ideas or formally presented work (including essays, examinations, projects, laboratory presentations). The penalties associated with plagiarism designed to impose sanctions seriousness of University's commitment to academic integrity. Ensure that you have read the University's **Briefing for Students on Academic Integrity and Plagiarism** and the UCD **Statement, Plagiarism Policy and Procedures**, (<http://www.ucd.ie/registrar/>)

Declaration of Authorship

I/we declare that all of the following are true:

- 1) I/we fully understand the definition of plagiarism.
- 2) I/we have not plagiarised any part of this project and it is my original work.
- 3) All material in this report is my/our own work except where there is clear acknowledgement and appropriate reference to the work of others.

Signed..... Date

Office Use Only

Date and Time Received:
[School Stamp]

Received by:
Report Tracking

Ancient Discovery: AI for Hieroglyphics Recognition

Linghua Gong - 15205904

Zechen Feng - 15205885

Yikun Wu - 15205937

Wei Du - 15205901

Comp3032J Software Engineering Degree Project

Supervisor: Dr. Ruihai Dong



UCD School of Computer Science

Beijing-Dublin International College

University College Dublin

May 2019

Abstract

The oracle bone inscriptions is an ancient Chinese character which has 3600 years of history. Nowadays, experts have already translated 1000 oracle bone inscriptions characters from about 3500 excavated characters [1]. There are still a lot of characters are not translated by philologists. With the development of artificial intelligence (AI) algorithms used in the translation field, we want to do some discovery work about the AI application on the oracle bone inscriptions translation fields.

In this report, we mainly discovery the translation by AI in three directions, which include using models to simulate the evolution process of Chinese ancient characters, using models to translation characters within the context and decomposing oracle bone inscriptions characters into several strokes. In the first part, we present several neural networks generating oracle bone inscriptions and their corresponding ancient bronze inscriptions and small seal style characters. These networks based on generation models, which include Autoencoder, Variational Autoencoder (VAE), U-Net and generative adversarial network (GAN). We adjust them in our project in order to suit our requirements. In the second part, we analyzed articles of the oracle bone inscriptions and attempt to establish a dataset of the oracle bone inscriptions rubbing by the order of reading. When establishing the dataset with the correct reading order, we built an algorithm to split regions which contain single sentences in the whole rubbings and then sort characters appearing in the same region according to current oracle bone inscriptions research. In the third part, we attempt to decompose strokes off from oracle bone inscriptions characters, which can help us understand the structures of each character to improve the generation model. The whole project starts from collecting data, tackling data, step by step, towards establishing complex neural network and algorithms, with the guidance and help of our supervisors and current research done by predecessors in each academic area.

Contents

1	Introduction	3
2	The Preparatory Work	6
3	Generation and Evolution	7
3.1	Literature Review	7
3.2	Generative Models	10
3.2.1	U-Net	10
3.2.2	Autoencoder	12
3.2.3	Variational Autoencoder	13
3.2.4	DCGAN Model	15
3.2.5	VAE/GAN Model	16
3.2.6	CYCLE-GAN Model	19
3.3	Transfer Learning	21
3.3.1	Descending Dimension Algorithms	21
3.3.2	The Design of the Transfer Learning Model	23
3.3.3	The Experiments of Transfer Learning	25
3.4	The Classification	28
3.4.1	The Elastic Deformation Algorithms	28
3.4.2	The Design of classification	29
3.4.3	The Experiment of Classification	30
3.4.4	Using classifier to detect style conversion	31
3.5	Chapter Summary	33
4	Dataset Establishment and Image Inpainting	34
4.1	Literature Review	34
4.1.1	YOLO	34
4.1.2	Image Processing	35
4.2	Worklog	37
4.2.1	Collection of Rubbings	37
4.2.2	Preprocessing of Rubbings Images	37
4.2.3	Discerning Characters	39
4.2.4	Sorting Characters	40
4.2.5	Repairing Characters Intersected with Cracks	42
4.3	Evaluation	43
4.3.1	Establishing Rubbings Dataset	43
4.3.2	Repairing Damaged Characters	45
4.4	Analysis of Unsatisfactory Result	46
4.4.1	Holes Residues and Target Elimination	46
4.4.2	High Average Loss	46
4.4.3	Model Collapse	46
4.5	Chapter Summary	47

5 Decompose Oracle Bone Inscriptions	48
5.1 Literature Review	48
5.1.1 The Canny algorithm	48
5.1.2 The Harris Corner Detector algorithm	49
5.1.3 The K-means clustering algorithm	49
5.1.4 The Hierarchical clustering algorithm	49
5.2 Decompose the Oracle bone inscriptions	49
5.2.1 Outline the Oracle bone inscriptions	49
5.2.2 Detect the concave points	50
5.2.3 The relationship between concave points and junctions	50
5.2.4 Strip the characters	52
5.3 Cluster the strokes	52
5.4 Chapter Summary	53
6 Conclusion and Future Work	54
Reference	56

Chapter 1

Introduction

Oracle bone inscriptions, an ancient Chinese script, also called "Qi script", "oracle bone inscriptions", Yin ruins script or "tortoise shell and animal bone inscriptions", is the early form of Chinese characters. It is the most ancient one of the mature characters in extant Chinese dynasty period. Oracle bone inscriptions are symmetrical and stable. Preparation of the three elements of calligraphy, namely, the use of a pen, knot, chapter. From the number and structure of the font, oracle has developed to a more rigorous system of writing. The principle of "six books" [2] of Chinese characters is reflected in the oracle bone inscriptions, however, the traces of the features of the original painting characters are still quite obvious [3]. With the passage of time, the change of dynasties, the oracle bone inscriptions evolved into several different characters, such as the ancient bronze inscriptions and the small seal style characters, and finally evolved into the Chinese modern character (as shown in Figure 1.1).

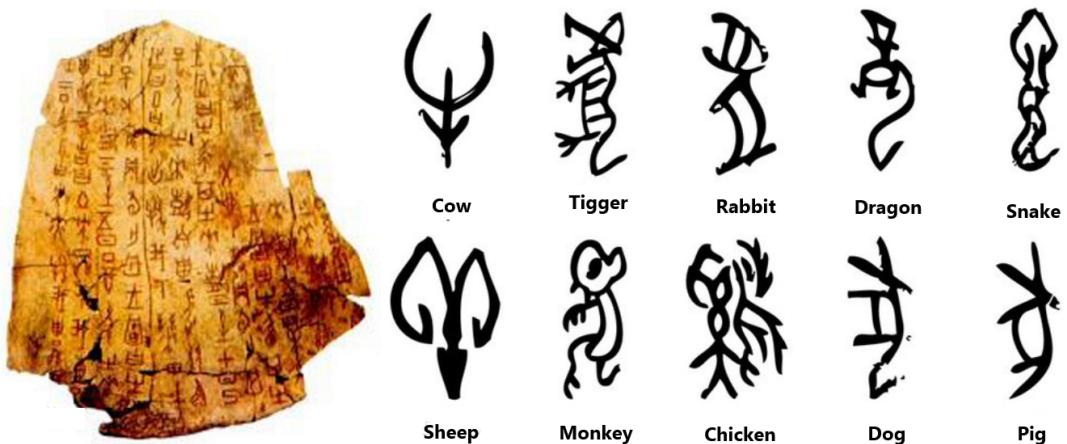


Figure 1.1: some examples of oracle bone inscriptions. Some features of paintings are still obvious in oracle bone inscriptions characters[3]

The research on the oracle bone inscriptions has a significant influence on archaeology and history. Although these oracle bone inscriptions have significant value, the research on this area exists several barriers. On the one hand, there is not enough data related to these ancient characters. The oracle bone inscription is the cultural product of the Shang Dynasty (about the 17th century BC to the 11th century BC). The only source of research on the oracle bone inscriptions is archaeological excavations, which lead to lack of related materials. On the other hand, most of the existing data, such as animal bones, tortoise shells and bronze wares, suffer from different extent damage. With these barriers, authorities in the oracle bone inscriptions struggle to discern some characters, which affects the research value of many cultural relics. What's more, caused by the fact that ancient characters are seldom used and the lack of research data, it is difficult for a novice to access this area.

The promise of our project is to discover a modern way of discerning unidentified oracle bone inscriptions characters, analyzing the meaning and structure of oracle and creating an oracle bone inscriptions rubbings dataset to help research on Chinese ancient characters.

So far, the research techniques on oracle still remain unadvanced. The original data of oracle is coming from archaeological excavation, the technique of discerning oracle is monopolized by a few authorities. In that case, our project provides an approach based on deep learning, with the oracle data that already identified by authorities, train a model which can predict unidentified oracles according to the identified oracles it learned. With the help of this model, once discover new unidentified oracles, users just input the image of these new oracles and then the model will generate a prediction of the corresponding ancient and modern characters. This model can also be applied to the research of the evolution of Chinese ancient characters. Our target characters are ancient bronze inscriptions (1300 B.C. to 219 B.C), the small seal style (221 B.C. to 8 A.D.) and Chineses modern characters (1950 A.D. to now).

We try to use some generative models to realize the evolution process of Chinese characters. The main idea is that the input is an oracle bones inscriptions character and trough the generative model to convert to its matched later style one. The main challenge is the shape conversion, the model needs to get the graphics features of a character and represent those features in the new character image. In there we build several generative models to do the shape conversion, which include U-net, Autoencoder, VAE and a serious of GAN. Each of them has its advantages and disadvantages in this project and some of them succeed in the project and some failed. We will give our analysis of the success and failure of each model. We even create a transfer learning model which combine with the DCGAN to do the shape conversion work. Finally, in order to evaluate the results of generated images, we design and train a classification, which can classify images data into a specific Chinese character.

On the other hand, we try to understand the specific meaning of oracle bone inscriptions character with context and help novices to access this area easier and faster. As oracle bone inscriptions are most carved on animal bones and turtle shell, our main data of oracle bone inscriptions articles are collected from oracle rubbings. We establish an oracle bone inscriptions articles dataset from the oracle rubbings we already have and crack. In this dataset, we provide our recommendation of the reading order of oracle bone inscriptions context (the distribution of oracle bone inscriptions content on rubbings is chaotic, really unfriendly to novice). The basis of this work is discerning oracle bone inscriptions characters appearing in rubbings using neural network model. The model has to know what oracle bone inscriptions and the hidden features of them is, then predict the coordinates of target oracle bone inscriptions characters. we use convolution layers to extract features and fully connected layers to identify the coordinates. The training set has been improved by image processing before inputting into the model. During the improving process, a model combining Autoencoder and the generative adversarial network is designed to understand the context of the damaged character and predict the reasonable content in damaged regions. We use a Python script to cut oracle bone inscriptions characters down from initial rubbings and sort them. We need to know the written laws and design algorithm to describes the laws.

Finally, we analyze the structure of oracle bone inscriptions characters, separate strokes in oracle bone inscriptions characters. The learning of strokes is another creative idea of discerning oracles. Both ancient and modern Chinese characters' meaning is based on the shape of the character, and the shape is based on the position strokes. Consequently, according to our investigation, strokes are a crucial factor in deciding the meaning of a character in Chinese, deeper knowledge of structure and strokes do contribute one possibility to make a prediction of unidentified oracle bone inscriptions characters and understand the changing process of Chinese ancient characters. So it is inevitable to decompose the strokes and extract the common strokes from the Chinese character to help further analyze.

In this report, we summarize the work we have done, the possible interpretation of some experimental result, challenges yet to be overcome and track down possible approaches. In Chapter 2, we briefly list the preparatory work, including data crawling and unification. The exploration to generation and evolution of the ancient characters presented in Chapter 3. Chapter 4 records the detailed process of establishing the oracle bone inscriptions rubbings dataset with sorted oracle bone inscriptions characters. The research on the structure of the oracle bone inscriptions characters is discussed in Chapter 5.

Chapter 2

The Preparatory Work

At the start stage of our project, we did some preparatory works which include data collecting, buying rubbing books, create the tfrecords format dataset and learning some online courses provided by Hongyi Li professor.

We used the crawler technology to collect oracle bone inscriptions characters, ancient bronze inscriptions characters and small seal style characters from a Chinese ancient character researcher's website [4]. After the collection process, we tidy our collected data and get a statistical overview of the data. The details of the characters are showed in Table 2.1. The categories mean how much types of characters and the total number means the number of all categories images. Each category of oracle bone inscriptions and ancient bronze inscriptions usually have multiple instances, which means there are different shapes of characters have the same meaning and can be translated to the same modern Chinese character. That situation is caused by that those two inscriptions had not been unified. However, the small seal style character is a unified character by Qin dynasty. As a result, each character only has one standard instance. We also did some picture processing work, which includes reshaping images into 96-pixel width and 96-pixel height and unify all images into 3 channels format. We also

21355 872 1480^b
甲 → 小篆 872
乙 → 金 518

Table 2.1: The details of collected data

	categories	the total number
21355 15344	The oracle bone inscriptions	918
	The ancient bronze inscriptions	1505
	The small seal style characters	5258

brought several books for creating our rubbing dataset. These books include *Selection of Oracle Bone Inscriptions Rubbing 1*[5], *Selection of Oracle Bone Inscriptions 2*[6] and *Classic Oracle Bone Inscriptions Rubbing 100 instances*[7]. Each of them provides lots of Rubbing instances which support a context of the oracle bone inscriptions. The rubbing can be seen as a printed paper which contains many oracle bone inscriptions sentences on it. There are some characters in sentences have been not translated by experts. We attempt to create a dataset within the characters order information.

As the preparation of training models, we created several tfrecords format datasets, which makes our model read data easily. To realize the shape conversion from the oracle bone inscriptions to the small seal style and ensure the output character correspond with the input character, we need using pair data to create dataset file and give each data in the dataset a type number. There are 872 pairs of the oracle bone inscriptions to the small seal style characters in the dataset. The type sequence number is from 1 to 872. Because that there are multiple instances of the same oracle bone inscriptions character, we code them with the same sequence number.

{ image inpainting
 { establish oracle rubbing dataset
 Chapter 3 { transfer learning
 { PCA distribution of dataset
 classification
 expand dataset } discomponent character
 Generation and Evolution { Unit style converting

In this chapter, we will discuss the shape conversion process from the oracle bone inscriptions to the small seal style inscriptions. We tried a variety of ways to reproduce this shape conversion process, using U-net, Autoencoder, Variational Autoencoder, DCGAN model, VAE/GAN model, CYCLE-GAN model, Transfer Learning models, and classifiers. From the oracle bone inscriptions to the small seal style inscriptions, the character is gradually abstracted, and gradually separated from the image attributes. The morphological changes of Chinese characters are regular, continuous and gradual.

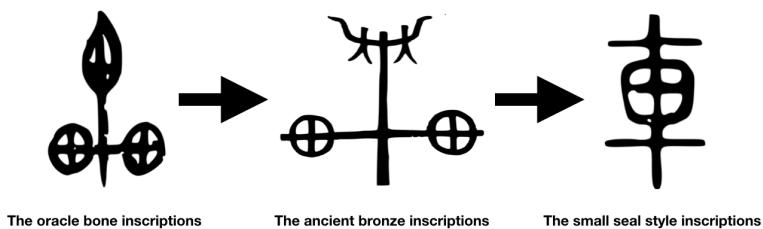


Figure 3.1: The evolution of character 'Vehicle' from oracle bone inscriptions to the ancient bronze inscriptions, and then to the small seal style

The Figure 3.1 is the shape conversion of the character "车" which means "Vehicle" in English. The oracle bone inscription "车" like a certain instrument has a wheel on each side. The ancient bronze inscription "车" also has the characteristics of the wheel. And the small seal style inscription "车" with one wheel instead of two wheels. we can get such a law through this shape conversion process, from the oracle bone inscriptions to the small seal style inscriptions, the character "车" has the characteristics of wheels. So, we hope to use the model find the law.

3.1 Literature Review

The oracle bone inscriptions are a type of hieroglyph, which based on the real object in the world. So, firstly, we would like to analyze the oracle bone inscriptions from the image direction, and the deep learning network is the best choice for us. We used a convolutional neural network to capture the features of the image. After that, we hoped to show the process of the evolution of Chinese characters. From Oracle to the ancient bronze inscriptions, then the small seal style inscriptions, and finally the modern simplified Chinese, we explored with the generative model to generate the goal characters, including U-Net, Autoencoder, Variational Autoencoder, Generative Adversarial Networks, and Transfer Learning. Finally we train the classifier to determine the correct rate of the generated results.

Convolutional Neural Network

In order to extract the feature of the character, and realize classification, we chose Convolutional

Neural Network to achieve this task. The Convolutional Neural Network (CNN model) is a type of deep learning network and be widely used in the region of visual imagery[8]. The CNN was inspired by biological experiments in the visual cortex in monkeys and cats held by Hubel et al [9]. Later, the CNN model was first proposed by Fukushima K.[10] The typical CNN model is composed of three layers: convolution layers, pooling layers, and fully connected layers [11]. The convolution layers are composed of several filters or kernels, they all have small respective fields respectively and produce a two-dimensional activation map. The max-pooling layers take charge of selecting the pixels that best characterizes the features. The fully connected layer is the same as the traditional multi-layer perceptron neural networks and in charge of classification[8].

U-Net

In order to realize the character shape's feature extraction and conversion, U-Net is a feasible technique to take an experiment because both input and output of the network is an image and the output is the input's feature. So it is feasible to give the experiment on the origin and modification version of this model to test its performance on this object. The U-Net model is a modified version of Fully Convolutional Network (FCN) which is for purposes and majored in the medical image region. The structure of U-Net is composed by several convolutional layers and several deconvolution layers which form the U-shape and the most important modification that let U-Net different from FCN is the large number of feature channel that transforms the convolution result in each layer in each deconvolution layers to make the model accurate up-sampling the image[12]. The U-Net can accept the input images of any size and use the deconvolution layer to upsample the feature map of the last volume base layer to restore it to the same size of the input image so that prediction can be generated for each pixel[13]. At the same time, the spatial information in the original input image is retained and finally, the parity is extracted in the upsampled feature map. It changes the fully connected layer of the CNN model by the convolution layer, and the adding deconvolution layer can solve the problem of the shrink image which was influenced by the convolution layer and max-pooling layer.

Autoencoder

The Autoencoder model consists of an encoder and a decoder, which can use any network structures such as Convolutional Neural Network (CNN), Deep Neural Networks (DNN). It tries to approximate an identity function, making the output close to the input. In order to make this function meaningful, we need to add some restrictions such as limiting the number of hidden neurons; then the function can find some meaningful structure. Autoencoder can learn some compressed representation of data.

Variational Autoencoder

The Variational Autoencoder model consists of an encoder and a decoder. The encoder is a neural network. Its input is data, its output is a latent representation space, which has weight and biases. The encoder encodes the data into a latent representation space, which is much less than data dimensions because the encoder must learn an efficient compression of the data into this lower dimensional space. The decoder is another neural net. Its input is the latent representation space; it outputs the parameters to the probability distribution of the data and has weights and biases..

DCGAN Model

Similar to the VAE model, the generative adversarial network model also contains a pair of sub-models. GAN 's name contains a concept of confrontation. In order to reflect the concept of confrontation, in addition to generator model, there is another model to help generator model to learn the conditional distribution of the observed data better.

This model can be called the discriminator model, its input is an image in the dataset, and the

output is the probability value indicating the probability that the image belongs to real data. For the generator model, its input is a random variable, which obeys a specific distribution, and the output is an image. If the generated image passes the discriminator model and the probability values are high, which means that the generator model has better mastered the distribution pattern of the data, and can produce samples that meet the requirements; otherwise, it does not meet the requirements, and further training is needed.

VAE/GAN Model

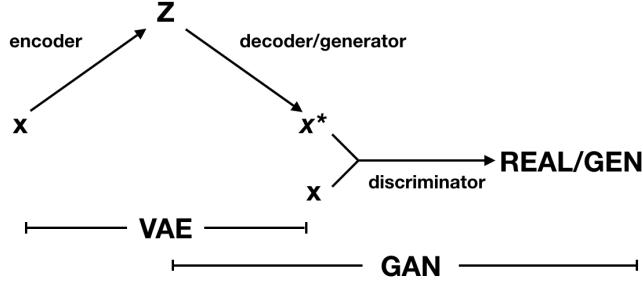


Figure 3.2: Overview the VAE/GAN network

The VAE with a GAN as shown in Figure 3.2. The VAE/GAN model collapse the VAE decoder and the GAN generator into one by letting them share parameters and training them jointly. For the VAE training objective, VAE/GAN model replaces the typical element-wise reconstruction metric with a featurewise metric expressed in the discriminator [14].

CYCLE-GAN Model

The CYCLE-GAN model presents an approach for learning to translate an image from a domain X to a target domain Y in the absence of paired examples. This model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and associated adversarial discriminators D_Y and D_X . D_Y encourages G to translate X into outputs indistinguishable from domain Y , and vice versa for D_X and F . The cycle consistency losses that capture the intuition that if the translation from one domain to the other and back again we should arrive at where we started [15]. The network of CYC-GAN model shown in Figure 3.3.

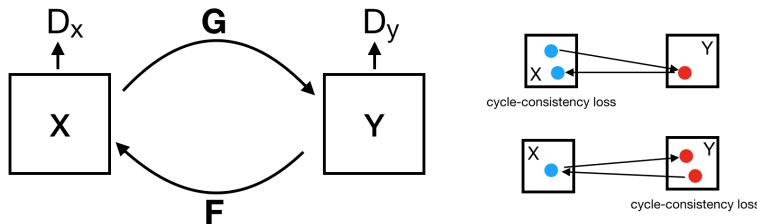


Figure 3.3: Overview the CYCLE-GAN network

Transfer Learning Model

We also used transfer learning in our project. We assumed that we can use a trained generation model as the generator and change the input data. For example, we can train a generator which can generate small seal style characters well and we change the input data to oracle bone inscriptions character. If we add some appropriate constraint conditions in the model, that new model will have the ability to do some shape conversion work. That idea is similar to transfer learning. So we design a transfer learning in our project.

The idea of transfer learning is developed in years ago, which supposed to solve the problem in

traditional machine learning. In traditional opinion, the trained model and future data must be in the same feature space and have the same distribution, while in the real world the model and data always cannot match each other very well [16]. In the idea of transfer learning, we can train a sparse coding model to transfer a high-level feature space into another one [17], which is an unsupervised learning process. In transfer learning, the original feature space is contained by new dataset and the target is the feature space of trained dataset. The transfer model can ensure that new dataset follows the same distribution in the same feature space as the trained model. After that process, the trained model could work well with a new kind of input data.

3.2 Generative Models

3.2.1 U-Net

The U-Net model is major in image segmentation; however, the input and the output of U-Net is relatively similarity to the objective. The inputs are Oracle bone inscription images and output is Ancient bronze inscription images. Although the U-Net is satisfied with image segmentation, it can also be experimented to apply in the image transfer area with some modification.

To construct the model which satisfies the current situation, the size of the convolution layer was modified. For each layer, a dropout method was adding to prevent the over-fitting problem. Also, the input size and output size are the same which is different from the original layer. The dataset is very ordinary, the unit of it is composed of one Oracle bone inscriptions image and another accordance character image. Within the training stage, the whole network converges in relativity quickly speed, however, after adjustment the parameter and training, the testing stage is not good, because the structure of the U-Net, the deconvolution step was based on the convolution result[13], consequently, the output keeps most of the input shape. So, the output is very similar to the input but seems not close to the expected Ancient bronze inscription result. As Figure 3.5 exhibit, the output image is more similar to the input rather than output.



Figure 3.4: Three input Oracle bone inscriptions (Origin U-Net model)



Figure 3.5: Three accordance output characters (Origin U-Net model)



Figure 3.6: Three accordance Ancient bronze inscription (Origin U-Net model)

According to the previous testing and the result reflected by the current model, the structure of the model needs to be modified to satisfied the image deformation issue. As the origin design of the model, in each convolution layers, the result is kept to make sure the accuracy of the deconvolution steps [6], that is the main point which affects the output of the model majorly.

So, there is one possible pattern for modifying the current model, which is adding multiple transform layers within the U-Net deconvolution layers. At each layer, the origin result from the convolution layer will be passed through the transform layer, after transformation, it will be given to the deconvolution layer. In the beginning, only two transformation layers were added to each deconvolution layer. In each transformation layer, the origin matrix will be flattened and multiplied by a bunch of weights variables and added bias variables, then pass through to the Rectified Linear Unit (ReLU) activation function, at last, reshape back to the original shape. After the adjustment the parameter and training for 10 epochs, the resulting picture does not reflect a strong shape correlation with the expected Ancient bronze inscription. Figure 3.7 exhibits the structure of the U-Net model with two transformation layers. What's more the output Figure 3.9 does not reflect strong correlation with the expect output 3.10

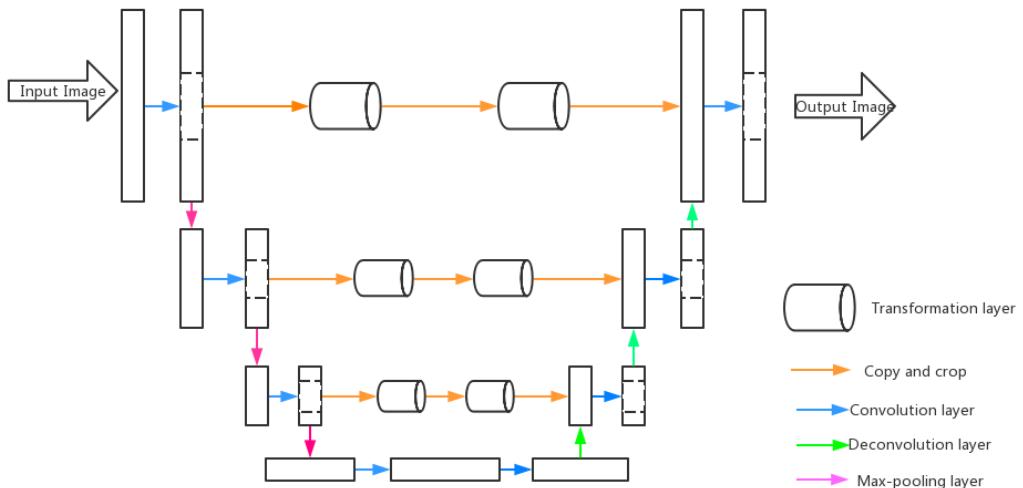


Figure 3.7: The U-Net structure with transformation layers



Figure 3.8: Five input Oracle bone inscriptions (U-Net model with two transformation layers)

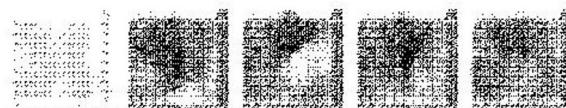


Figure 3.9: Five accordance output characters (U-Net model with two transformation layers)



Figure 3.10: Five accordance Ancient bronze inscription (U-Net model with two transformation layers)

After analyzing the modified model, the potential problem might be the lack of transformation steps cause the image shape can't change properly, which means the current two steps transform layers are not strong enough to transform the shape well. So, in the second version of

the model, four transformation layers were added to the original model, after adjustment the parameter and training, the result images fade seriously, but according to the shape reflected by the images, the transformation layer does some work but not strong enough to represent the shape deformation pattern between Oracle bone inscriptions and Ancient bronze inscriptions. The output of the model Figure 3.12 is fade and also does not reveal the strong correlation with the output 3.13



Figure 3.11: Three input Oracle bone inscriptions (U-Net model with four transformation layers)

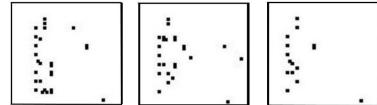


Figure 3.12: Three accordance output characters (U-Net model with four transformation layers)



Figure 3.13: Three accordance Ancient bronze inscription (U-Net model with four transformation layers)

In general, the U-Net model and it's altered version is not good as shape transformation. There might be two potential issues, first one is the transformation layer might not strong enough as GAN for shape deformation, the second one is there might not have a strong deformation pattern between the oracle bone inscriptions and the ancient bronze inscriptions.

3.2.2 Autoencoder

The Autoencoder model can learn some compressed representations for data. For instance, if the input data is 100-dimensional and the hidden layer is 20, then 100-dimensional output needs to be reconstructed from the 20-dimensional data, making this output close to the 100-dimensional input. Therefore, the 20-dimensional data of this hidden layer will inevitably contain some correlation of the input data. So, if we can encode the oracle bone inscriptions to some dimensional data that contain some correlation of the oracle bone inscriptions, we will get the output result that close to the input.

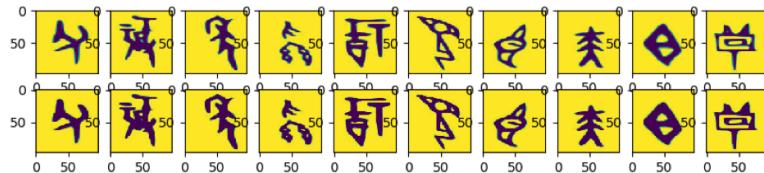


Figure 3.14: The first row is the real the oracle bone inscriptions and the second row is the data after the encoder and decoder. This is the result of 50 iterations.

In the Autoencoder model, we used the encoder to reduce the dimension of the character, which is from 96 x 96 x 3 to 200 dimensions, and the decoder to generate the image. The encoder and

decoder network are all a 2-layer convolution network. The structure of the Autoencoder model shown in Figure 3.15. After training the Autoencoder model, we got the output oracle bone inscription close to the input. The result shows in Figure 3.14. Therefore, the Autoencoder model can generate the oracle bone inscription. However, the Autoencoder model can only be said to extract image features and cannot generate new images by itself. The Autoencoder model helped us find the best dimensions that contain the correlation of the oracle bone inscriptions.

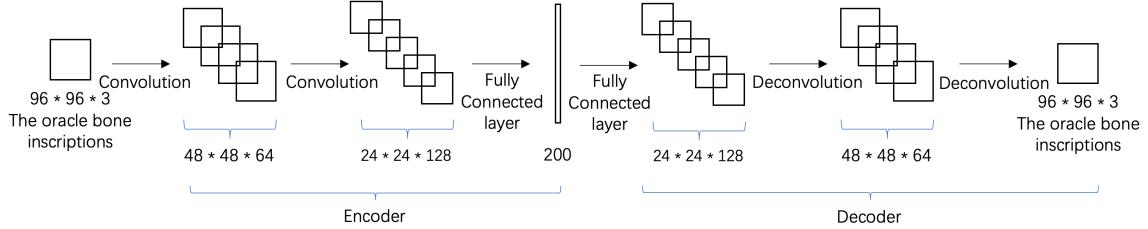


Figure 3.15: The structure of Autoencoder Model

3.2.3 Variational Autoencoder

The difference between Autoencoder and Variational Autoencoder(VAE) is that the output of the encoder is a latent representation space. There are two terms in the loss function. One is the reconstruction loss, the other is regularize loss.

The reconstruction loss. The expectation is taking concerning the encoder's distribution over the representation. The decoder uses this term to learn to reconstruct the data. If the decoder's output does not close to input the data well, statistically the decoder parameterizes an impossible distribution, and this probability distribution will not place much probability mass on the correct data.

两个分布间差异的度量

The regularize loss. This is the Kullback-Leibler divergence between the encoder's distribution $q_\theta(z | x_i)$ and $p(z)$. This divergence measures how much information is lost when using q to represent p . It is one measure of how close q is p . In the Variational Autoencoder, p is specified as a standard normal distribution with mean and variance. If the encoder outputs representations z that are different than those from a standard normal distribution, it will receive a penalty in the loss. This regularize term means 'keep the representations z of each digit sufficiently diverse'[18].

We did two models; one is using the oracle bone inscriptions to generate the oracle bone inscriptions, the other is using the oracle bone inscriptions to generate the ancient bronze inscriptions. The first model structure shown in the Figure 3.16. The second model structure shown in the Figure 3.17.

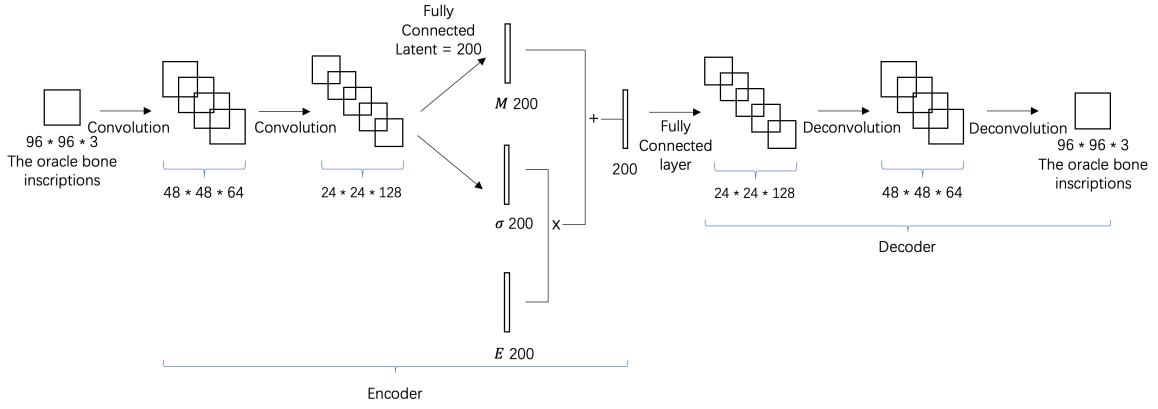


Figure 3.16: The structure of VAE Model 1, there are two convolution layers and one fully connected layer in encoder and two convolution layers and one fully connected layer in decoder.

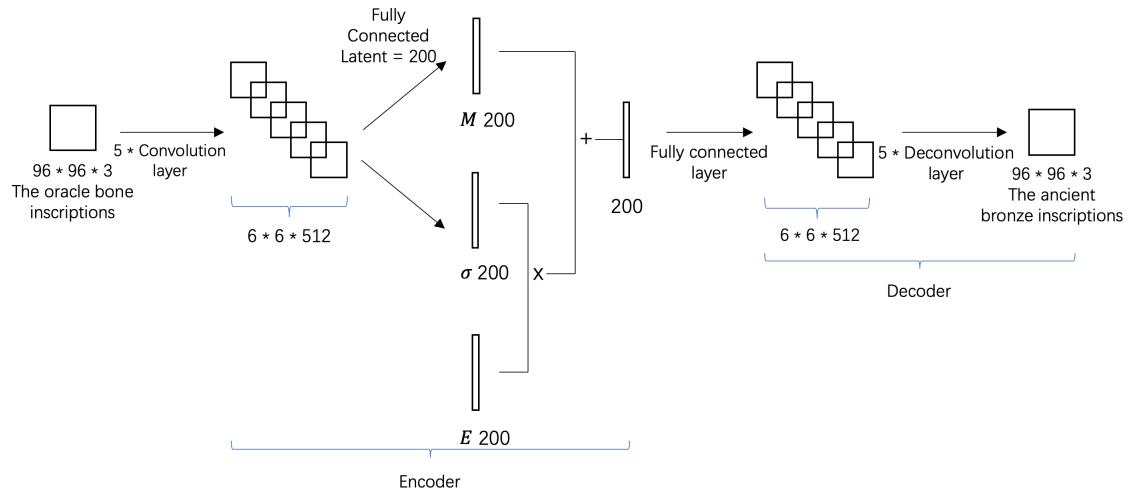


Figure 3.17: The structure of VAE Model 2, there are two convolution layers and one fully connected layer in encoder and two convolution layers and one fully connected layer in decoder.

The first model is successful. The result shows in Figure 3.18. So, we would like to use the oracle bone inscriptions to generate the ancient bronze inscriptions.

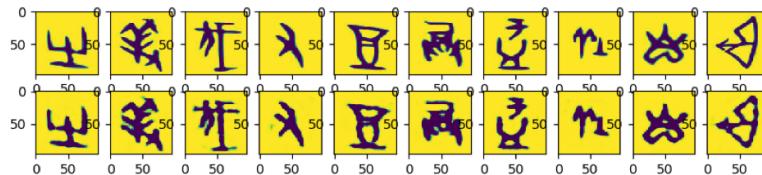


Figure 3.18: The first row is the real oracle bone inscriptions and the second row is the output oracle bone inscriptions after the VAE encoder and decoder. The latent number is 200 and the epoch number is 50.

In the second model, we did not get clear, and most of the results were vague. The result shows in the Figure 3.19.

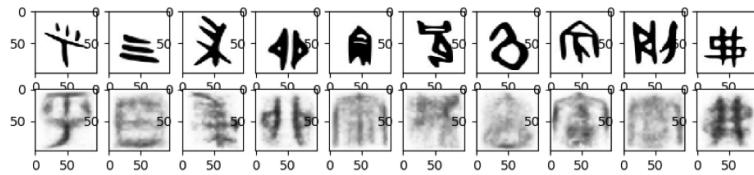


Figure 3.19: The first row is the real ancient bronze inscriptions and the second row is the out ancient bronze inscriptions data after the encoder and decoder. The latent number is 200 and the epoch number is 60.

It

The shortcomings of VAE are obvious. It directly the mean square error of the generated image, which makes the generated image a bit fuzzy.

3.2.4 DCGAN Model

Because of the shortcomings of VAE, we cannot use the oracle bone inscriptions to generate the ancient bronze inscriptions clearly. We tried another generative model which called Generative Adversarial Network (GAN).

uniform

At first, we used a random point to generate the oracle bone inscriptions. 200 joint distribution random points are input into the generator, and the oracle bone inscription picture enters the discriminator. The structure of DCGAN model shown in the Figure 3.20.

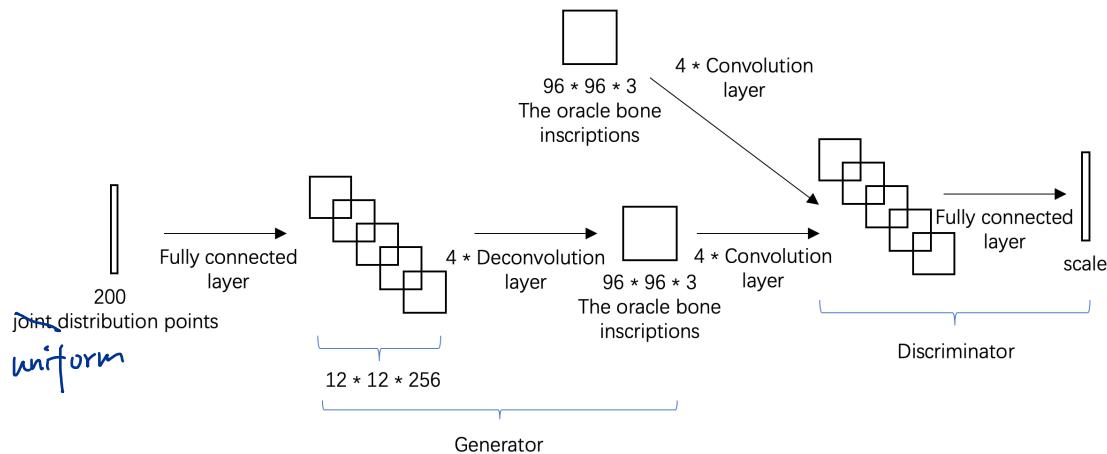


Figure 3.20: The structure of DCGAN model, four deconvolution layers and one fully connected layer in the generator, four convolution layers and one fully connected layer in the discriminator.

Then, we use the oracle bone inscriptions to generate the ancient bronze inscriptions. The initial idea is that the input source is the oracle bone inscriptions images into the generator and the ancient bronze inscriptions into a discriminator. This generator is different from the original GAN's generator. In this model's generator, the oracle bone inscriptions will convert into 200 points; this is because this kind of 200 points will use to replace 200 joint-distributed random points as the input source.

uniform

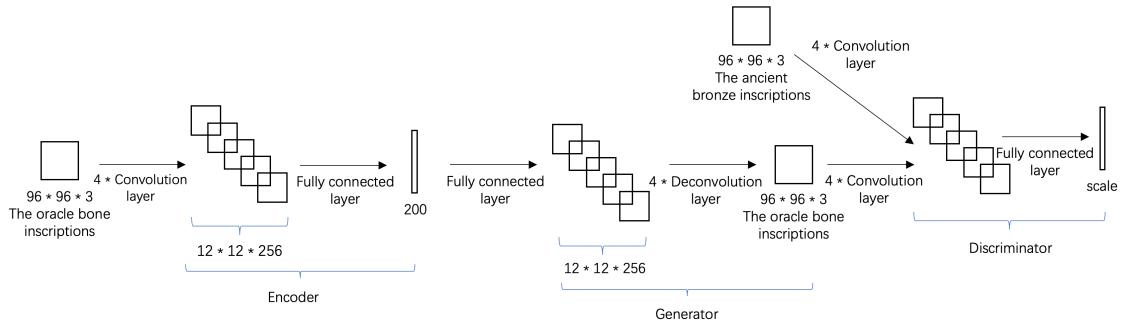


Figure 3.21: The structure of DCGAN Model

The structure of the encode is four convolution layers, four max-pooling layers and one fully connected layer to convert the oracle bone inscriptions which pixels are 96×96 and the channel is three to 200 points, which is shown in the Figure 3.21. The convolution kernel's number is 64, the size is 3×3 , the strides is 1, and padding mode is the SAME. The structure of the generator is one fully connected layer and four deconvolution layers to use 200 points to generate an image in which image channels is three and pixels are 96×96 . The deconvolution kernel's number is 32, the size is 3×3 , and strides is two which does not need up max-pooling layer, and padding mode also is the same. The discriminator's structure is four convolution layers, four max-pooling layers, and one fully connected layer. The structure of convolution is the same as the encoder's convolution.



Figure 3.22: DCGAN model the oracle bone inscriptions to the ancient bronze inscriptions

However, the DCGAN model that generated the ancient bronze inscriptions from the oracle bone inscriptions failed. The output became the same after the first epoch, and the discriminator loss function becomes to zero rapidly. The result shows in Figure 3.22. We analyze the results, the reason for the collapse of DCGAN model caused by the disappearance of the generator gradient: when the discriminator is very accurate, the discriminator's loss quickly converges to zero, which does not provide a reliable path to keep the generator's gradient updated, which cases the generator gradient to disappear. The encoder converts oracle bone inscriptions into a 200-dimensional point distribution, which is too far away from the real ancient bronze inscriptions. There is almost no overlap between the two distributions. At the same time, the discriminator can quickly distinguish the real data and the generated fake data to achieve the optimality of the discriminator, causing the generator's gradient cannot continue to update or even the gradient to disappear.

3.2.5 VAE/GAN Model

When we converted Oracle to small seal style inscriptions using the DCGAN model, we found that the model converges very slowly. The model collapsed after dozens of iterations. We analyzed that the learning ability of the generator was too weak; the result that generator

generated cannot fool the discriminator. If the generator knows the feature distribution of small seal style inscriptions at the beginning, it will not deviate from the features of small seal style inscriptions when generating the small seal style inscriptions. At first, we added the teacher forcing to the DCGAN model, but the results were not satisfactory. We then noticed the VAE/GAN model, which is the best solution for this problem.

GAN uses the generated confrontation network to be useful in the quality of picture generation, and the pictures have apparent features. However, it is prone to collapse during the training of GAN, and the disappearance of the gradient during training. The game theory that generates the confrontation network is to let the G generated image fool the D. This will make the G drill hole once fooled the D, regardless of the image's unreasonable output. The above reasons cause the image generated by GAN to look a bit abnormal at times.

VAE uses the real picture to encode the latent representation space under the encoder. This space retains the features of the original image in the case of obeying the Gaussian distribution. The picture obtained after decoder decoding is more reasonable and accurate. However, when the image trained, the picture obtained after decoder decoding is more reasonable and accurate. However, when the image trained, the loss function can only be measured by a rough error such as Mean squared error, which results in the resulting image not retaining the sharpness of the original image well, which makes the image look a little blurry. The advantage is that the picture generated by VAE is reasonable, and the learned latent representation space can restore the image well, and the training will not collapse.

The purpose of VAEGAN model is to combine the advantages of VAE and GAN to ensure the stability of the model and the quality of the image under reasonable conditions.

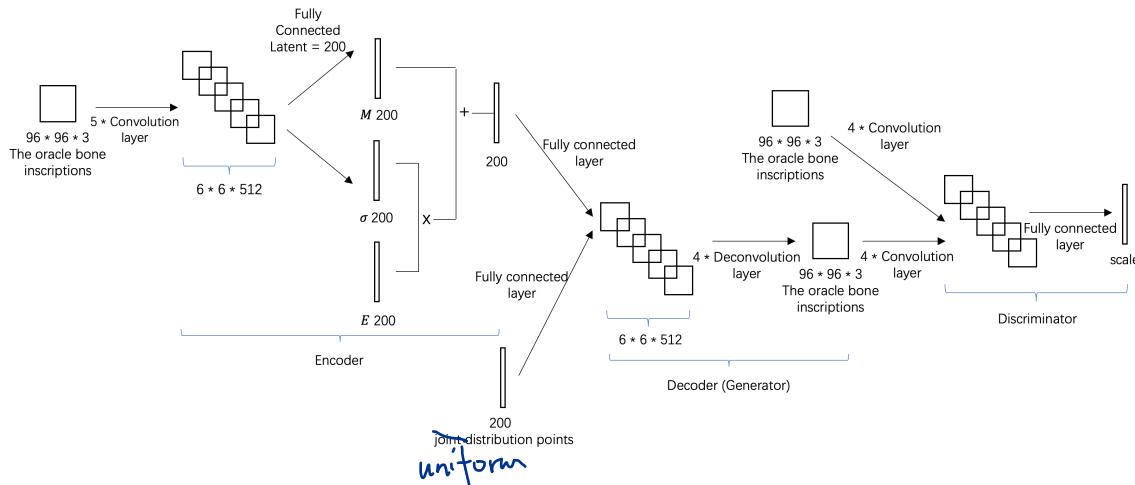


Figure 3.23: The structure of VAE/GAN Model, five convolution layers and one fully connected layer in the VAE encoder, one fully connected layer and four deconvolution layers in generator, and one fully connected layer and four convolution layers in discriminator

Before we used the VAE/GAN model to generate the small seal style inscriptions by using the oracle bone inscriptions as the input source, we used 200 random joint distribution points as input sources to generate the oracle bone inscriptions to test the VAE/GAN performance, which structure shown in Figure 3.23. The results met expectations.



Figure 3.24: The first image is the oracle bone inscriptions generated by DCGAN model through 200 joint distributions points, the second image is the generated oracle bone inscription “人” which means people in English, and the third image is the real oracle bone inscription “人”

Figure 3.24 shows the result of DCGAN model iterating 200 times through 200 joint distribution points. We can clearly get information that the generated oracle bone inscription is similar in shape to the real oracle bone inscriptions.

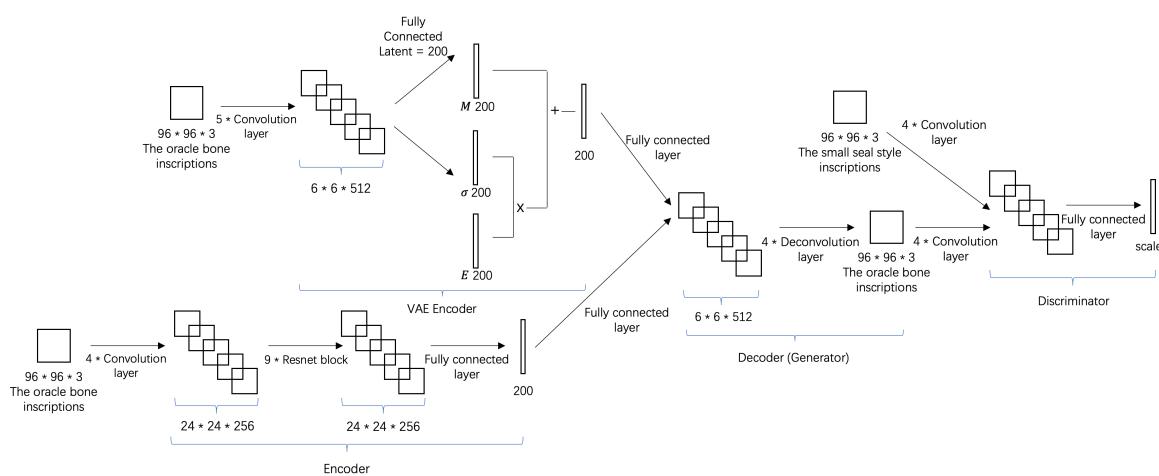


Figure 3.25: The structure of VAE/GAN Model

Then we used the oracle bone inscriptions to generate the small seal style inscriptions. The structure of the VAE/GAN model shown in Figure 3.25. The input source is the oracle bone inscriptions dataset and the small seal style inscriptions dataset, respectively. We put the oracle bone inscriptions into the encoder in order to convert it into 200 points to replace 200 joint distributed random points and put the oracle bone inscriptions into VAE encoder in order to let model learn the features of the small seal style inscriptions. Put small seal style inscriptions into discriminator in order to confirm whether the result generated by the generator is small seal style inscriptions.

CNN can extract low/mid/high-level features, the more layers of the network, the richer the features that can be extracted to different levels. Moreover, the deeper the network extracts, the more abstract the features, the more semantic information. So, we would like to use more layers to extract the features of the oracle bone inscriptions to find links with random joint points. For the CNN network, if we increase the depth, it will lead to gradient dispersion or gradient explosion. So, we use Resnet into encoding. The encoder is an 18-layer network. Two 3×3 convolution networks are concatenated together as a residual block.

We set two convolute layers in the VAE encoder, one fully connected layer and four deconvolute layers in the generator and four convolute layers and one fully connected layer in the discriminator. The number of oracle bone inscriptions in the dataset is 21354, and the number of the

small seal style inscriptions in the dataset is 783.



Figure 3.26: The first image is the input the oracle bone inscriptions, the second image is the target the small seal style inscriptions, and the third image is the actual output the small seal style inscriptions

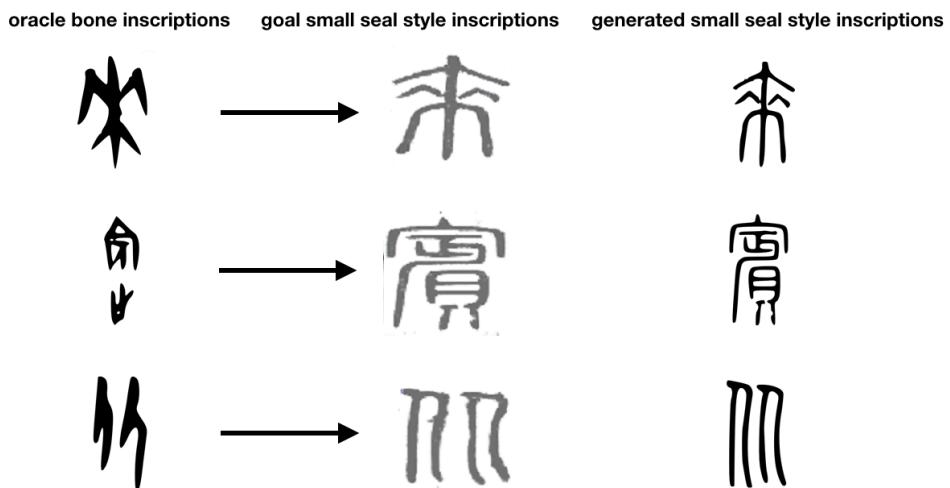


Figure 3.27: The sheet of VAEGAN model result

From Figure 3.26 and Figure 3.27, this VAE/GAN model can generate the right small seal style inscriptions, but there is another problem that the input image and the output image cannot correspond. We analyze the model and the result and draw two conclusions.

- We think that it is the limitation of the existence of the GAN model itself. The discriminator is a two-classifier model. It can only judge whether the result produced by the generator is or is not small seal style inscriptions, instead of classifying the small seal style inscriptions so this model can generate the correct small seal style inscriptions, but we cannot know which character it is.
- The datasets of the small seal style inscriptions and the oracle bone inscriptions. For instance, the character “宾” and “儿”. The number of different styles of the oracle bone inscription of “宾” is 251 and the number, and the number of different styles of the oracle bone inscription of “儿” is 49. As a result, each character has different training time, “宾” is trained more often than “儿”, which makes it easier to generate “宾” in the result.

3.2.6 CYCLE-GAN Model

We think that the oracle bone inscription has its style, and the small seal style inscriptions have its style too. Generating the small seal style inscriptions with the oracle bone inscriptions is a

conversion of different styles of a character mainly. The CYCLE-GAN model is the best choice for us.

The CYCLE-GAN model's goal is to learning mapping functions between two domains X and Y , so we use oracle bone inscriptions as X dataset and the small seal style inscriptions as Y dataset. We denote the data distribution as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. Cycle Gan model includes two mappings $G : X \rightarrow Y$ and $F : Y \rightarrow X$. Also, this model has two adversarial discriminators D_X and D_Y , where D_X aims to distinguish between images x and translated images $F(y)$; in the same way, D_Y aims to discriminate between $\{y\}$ and $G(x)$. The objective contains two types of terms: adversarial losses for matching the distribution of generated images to the data distribution in the target domain and cycle consistency losses to prevent the learned mappings G and F from contradicting each other.

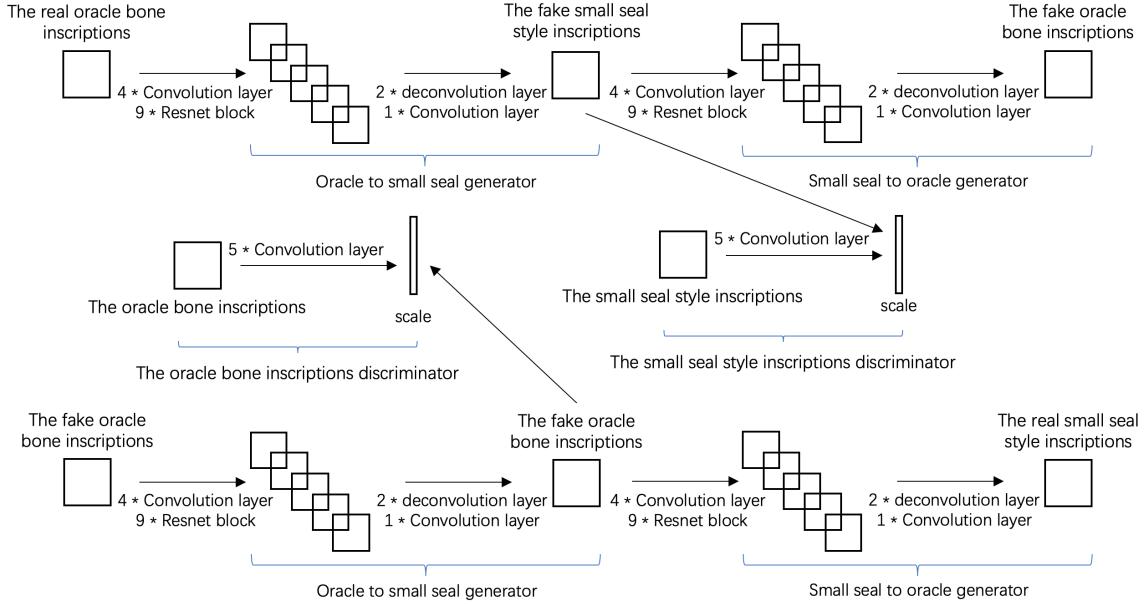


Figure 3.28: The structure of CYCLE-GAN Model

Figure 3.28 shows the CYCLE-GAN model. There are two discriminators and two generators, which are the generator of the oracle bone inscriptions generated the small seal style inscriptions, the generator of the small seal style inscriptions generated the oracle bone inscriptions, the discriminator that whether the generated character belongs to the oracle bone inscriptions and the discriminator that whether the generated character belongs to the small seal style inscriptions. In this model, we used the small seal style inscriptions expansion data, which number is 14800.

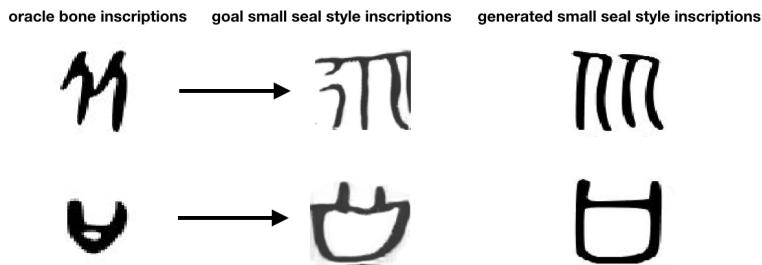


Figure 3.29: The first column is the input data, the second column is the output data and the final column is the goal of the output.

The result shows in the Figure 3.29. The resulting glyph is similar to the target, but there are many differences in details. The CYCLE-GAN model was initially designed for style migration,

such as changing horse to zebra, but from the oracle bone inscriptions to the small seal style inscriptions not only involve style migration but also involve glyph changes. Maybe both the oracle bone inscriptions and the small seal style inscriptions have their style of fonts, but the transformation of glyphs is not possible with the CYCLE-GAN model, because each character has its transformation, such as some characters are simple to complex changes, and some characters are complex to simple changes.

3.3 Transfer Learning

In the DCGAN part, we found that DCGAN model can generate some characters, which have the style of oracle bone inscriptions, by input the 200-dimensional noise data. However, if we want to use DCGAN model to do some shape transfer generation, for example, input oracle bone inscriptions data and setting the small seal style characters as the target, the model can not work well. In our analysis, the failure reason should be the model is too simple to finish the shape conversion work. As a result, we do some experiment about VAE/GAN and Cycle—GAN. We also considered that the transfer learning model can be the solution to do some shape conversion work.

Before we build and train the model, we still need a method that can help me to analyze the distribution of the data and judge if the distribution of new data is transferred into the target feature space. In there we used descending dimension algorithms to show the high dimensional information into a two-dimensional plane.

3.3.1 Descending Dimension Algorithms

Mainly we used Principal Component Analysis (PCA) algorithm as the descending dimension algorithm in our project. The PCA algorithm “is a linear algorithm that was originally formulated as a minimization of the sum of squared residual errors between projected data points and the original data” [19]. In other words, it uses a serious projection to transfer high dimensional data in a 2-dimensional plane, which can maintain much more space information of the original data. There are also some weaknesses to the PCA algorithm. Two datasets in the high-dimensional space do not entangle to each other, they split off from each other. Using the PCA algorithm, two datasets would entangle and cover each other. That is caused because the PCA algorithm only maintains the spatial information, but the neighbourhood information of each point. In that case, we only use it to analyze the distribution features between datasets in the high-dimensional space.

The PCA can show that in what extent two datasets’ distribution close to each other. If two datasets have the same shape of coverage on the same plane, generally we can judge that two datasets follow the same distribution. (However, they may in the different regions in the high-dimensional space.) If the range, shape or concentration has the obvious difference, we can judge that they are not following the same distribution in the high dimensional space. In that situation, if we want to use them as the source and target data in the transfer learning model, we need to build a model to convert data from one feature space to the other.

There are two experiments by using descending dimension algorithms. The first one compares the features of encoded oracle bone inscriptions, ancient bronze inscriptions, and small seal style characters. (The encoded data is the original distribution of image data because the parameter in the encoder is just initialized without any iteration. The structure of the encoder shown in Figure 3.30) This experiment is a supplement instruction. We attempt to understand the feature of the spatial distribution of three kind of Chinese ancient characters. Using the results, we can evaluate that to what extent they are similar to each other, and if it is difficult to do some shape conversion work between them. The second experiment to compares the distribution between encoded oracle bone inscriptions image data and 200-dimensional noise data.

This experiment tries to find the relationship between the source domain and target domain in the transfer learning. If they follow the same distribution or not. The difference between distribution will influence the design of conversion part. The more distinct differences there are, the more complex structure converter should be.

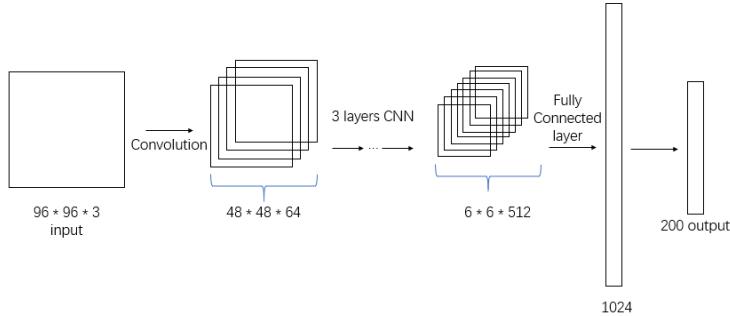


Figure 3.30: The structure of encoder used in the transfer learning model

The PCA results in Figure 3.31 show that the relationship between the three styles of ancient Chinese characters. Each dataset occupies the center area in the picture, so we should focus on the boundary of each dataset. The ranges and shapes of the oracle bone inscriptions and ancient bronze inscriptions are quite similar. The oracle bone inscriptions cover the slight upper area when compared with the ancient bronze inscriptions. However, the small seal style dataset has obvious differences with the oracle bone inscriptions dataset. The small seal style dataset is more centralized and the coverage is smaller than the oracle bone inscriptions. We can explain that by the order of inscriptions' evolution, the small seal style characters born 500 years later than two others.

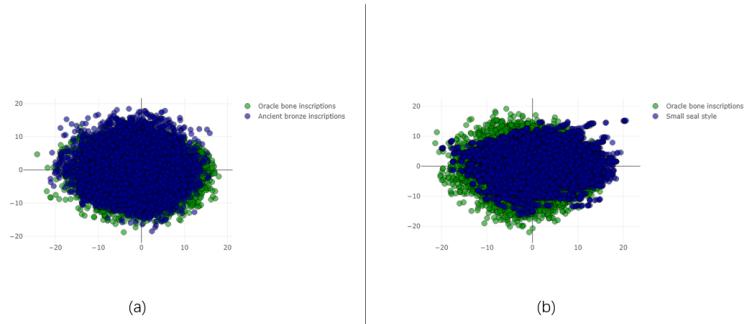


Figure 3.31: The PCA results about ancient Chinese character datasets.

(a) The blue points refer to the oracle bone inscriptions. The green points refer to the ancient bronze inscriptions

(b) The blue points refer to the oracle bone inscriptions. The green points refer to the small seal style characters.

In here, we need to review the target choice about the oracle bone inscriptions shape conversion. Obviously, using the ancient bronze inscriptions as the target is easier than the small seal style characters. The reason is that the ancient bronze inscriptions maintain much more similar graphic features about the characters, so the conversion difficulty is lower than the small seal style characters. While the real situation is that both oracle bone inscriptions and ancient bronze inscriptions are multiple characters match the same meaning. There are even some characters own several totally different shapes of characters have the same meaning. If we use them as the source and target, we must build a model with more restrictions to avoid the collapse of the model. As a result, we chose the small seal style characters as the target. Although that increase the difficulty of shape conversion, it solves the matching problem. The small seal style

characters are a unified character, the training model can become a multiple-to-single model.

The second serious PCA results are shown in Figure 3.32. (We did not find an algorithm can project the encoded value to the range from -1 to 1 as the noise. So we expand the range of random uniform distribution noise. The expanded range is from -14 to 15. Choosing that range is because, the maximum and minimum value in the encoded value is 14.53 and -13.29.) From that graph, we can find that encoded value has a different distribution with random uniform points. The noise data own a broad distribution, while the encoded value is more centralized. In general, the source data and the target data in transfer learning follow the different distribution in the high-dimensional space. We need train a convetor for data conversion.

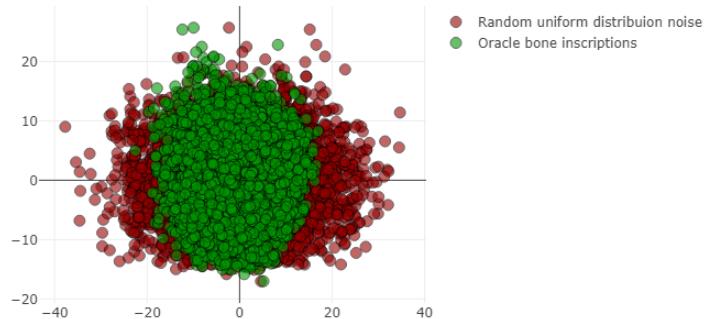


Figure 3.32: The PCA results about the oracle bone inscriptions and random uniform distribution noise. The green points refer to the oracle bone inscriptions. The red points refer to random uniform distribution noise

That result can also explain why in DCGAN model part, we use encoded value as the input of the model, the generated image eager to become the same one after several epochs. The dataset is so centralized that lead to parameters in generator convergence to a state, no matter what value input into the generator the results would be similar to others.

3.3.2 The Design of the Transfer Learning Model

The transfer learning model that we built has two main part. The first part based on DCGAN model mainly focuses on generating the small seal style characters. The second part is an encoder to encode the oracle bone inscriptions into a 200-dimensional value. Besides, it should also transfer the value into the same distribution space of the DCGAN model.

In the first part, we trained a DCGAN model which can generate the characters with the style of the ancient small seal style characters by input 200-dimensional noise. The noise follows the uniform distribution from -1 to 1, which can guarantee the result of DCGAN model is the best. The DCGAN model will be fixed after training and become the target model in transfer learning the part, which new data need to adapt to its feature space and distribution.

In the second part, we trained a convertor which has the structure shown in Figure 3.33 and Figure 3.34. This convertor not only needs to get the feature of the oracle bone inscriptions but also should transfer data into a uniform distribution feature space.

In the convertor, the input is the oracle bone inscriptions. The first stage target is that converting the encoded value into uniform distribution space. The second stage target of the model is

that generating the matched small seal style characters. The input characters have a label, we use it to find the matched small seal style characters. The whole process of convertor training has several steps. The first step is that a batch of images are inputted into the convertor and we get a batch of 200-dimensional encoded value. The model calculates the loss between encoded value and random uniform distribution noise. Then we use the encoded value to generate an image by the trained DCGAN model and calculate the distance between the generated image and matched image as the second loss. Finally, the model uses two losses to update the CNN and DNN. Updating the CNN part to ensure the output value still have the features of the original character and the generated images are similar to the matched small seal style one. Updating the DNN part to ensure the output value matches the uniform distribution space.

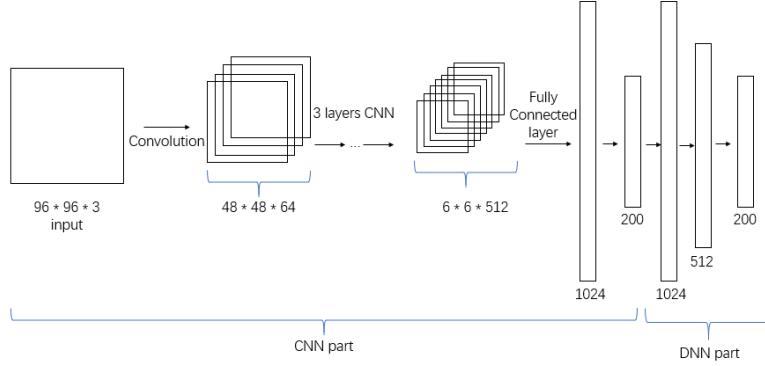


Figure 3.33: The structure of encoder part used in the transfer learning model

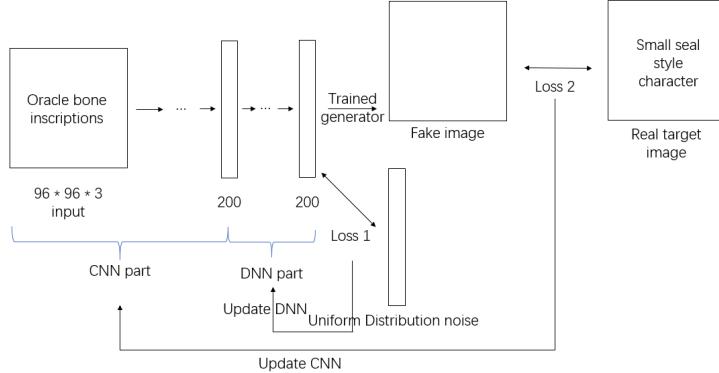


Figure 3.34: The structure of convertor used in the transfer learning model

The structure of the CNN layer is shown in Figure 3.35. It encodes an image value in 200-dimensional value. In each layer, the value from the former layer will pass convolution calculation, batch normalization, Rectified Linear Unit (ReLU) activation function and max pooling. The convolution and max-pooling are used to extract features from the image and reduce the number of values. The batch normalization can promote the convergence of the model. The ReLu activation function can avoid the overfitting problem of the model. After the convolution part, there is still two fully connected layers. Those two layers ensure the shape of output value with 200 dimensions. The loss of CNN is calculated by using the distance between the generated image between the matched small seal style characters. Using that as the loss can make the convertor generate the matched characters, which control the trained model to do some translation work.

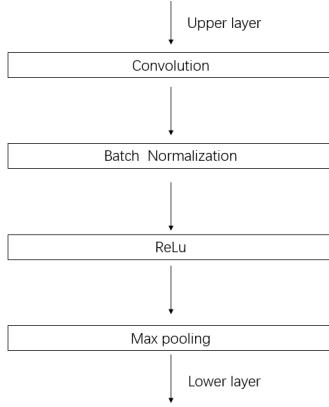


Figure 3.35: The structure of CNN layer used in the transfer learning model

The structure of the DNN layer is shown in Figure 3.36. There are three fully connected layers. It connects the input value to 2048 cells at first and uses two layers to reduce the cell to 200 later. There are still using batch normalization and Rectified Linear Unit (ReLu) activation function to avoid the model collapse during the training process. When calculating the loss of DNN, the first step is using Softmax function to map data into the range from 0 to 1. Using that can solve the problem that the range of encoded value and random uniform value is different from each other. The loss function is built by using cross entropy, which calculates the difference between logits distribution and target distribution. Using it can pull the result of encoder close to the target distribution, which is the key work of transfer learning data.

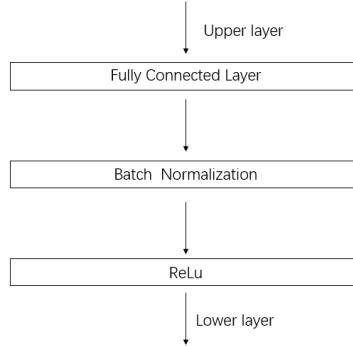


Figure 3.36: The structure of DNN layer used in the transfer learning model

In the assumption, if the convertor trained well, the output value will have the feature information of character and follow the uniform distribution. Then that value input into the DCGAN model, a matched small seal character of the input oracle one can be got, which realize the shape transfer of a character.

3.3.3 The Experiments of Transfer Learning

Figure 3.37 shows the result of DCGAN model, which input data is random uniform distributed 200-dimensional noise. The result is quite good, we can find those generated characters contain the features of the small seal style characters. As a result, this model can be used in transfer learning as the target.

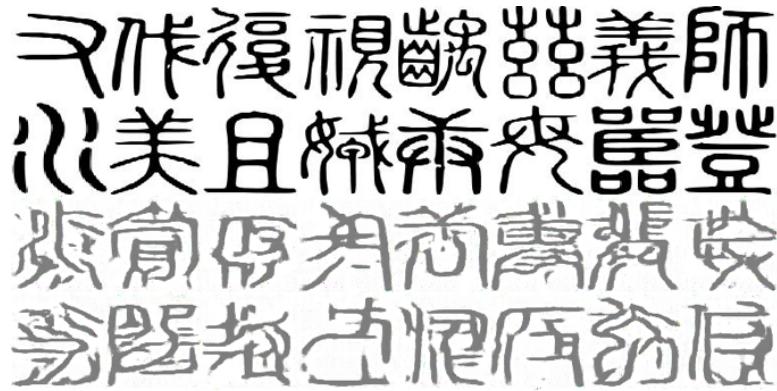


Figure 3.37: The result of DCGAN in transfer learning model. The black characters are the real small seal style characters. The grey characters are the generated characters.

Generally, the experiment results are not following the assumption. Figure 3.38 shows result in the encoder model before training and trained after 10 epochs. Obviously, all generated image became the same one after 10 epochs. We can assume that after training several epochs, the encoder output the same value no matter what input is. The reason could be the CNN part or DNN part collapse during the training. On the other hand, there are some generated images resemble its target at the beginning of the training. The comparison of similar images is shown in Figure. However, the parameters in the model were only initialized without any updating. In our opinion, that kind of similar results is a coincidence and cannot pass verification.

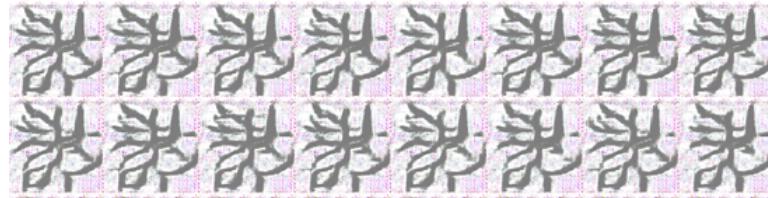


Figure 3.38: The result of transfer learning model after 10 epochs.

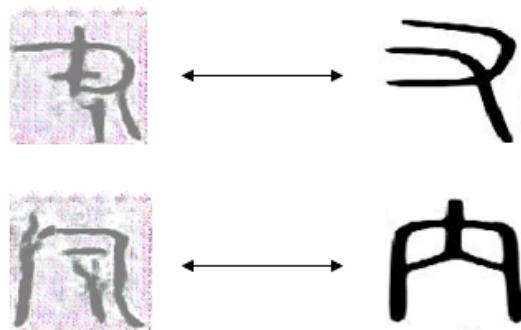


Figure 3.39: The result of transfer learning model at the beginning. The grey characters are the generated. The black characters are the target

In order to find which one causes the collapse problem, we did two tests. The first one is close to the optimizer of DNN part, make CNN part parameters. The second one is close to the CNN optimizer and keep DNN part still work.

The test one results are shown in Figure 3.40. The generated images became the same one after one epoch. The test two results are shown in Figure 3.41. The generated images are different

however these images only have the style of the small seal style characters. These images are not real small seal style characters, which close to the results of the original DCGAN model.



Figure 3.40: The result of transfer learning model which only updates CNN part.

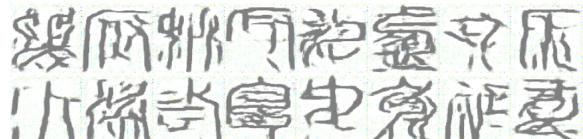


Figure 3.41: The result of transfer learning model which only updates DNN part.

From those two tests, we can find that the CNN leads to the collapse of the model. When we just update the DNN, the model can realize the target of the transfer learning. However, the real target is using transfer learning to realize the shape conversion from one oracle bone inscriptions character to matched small seal style characters character. The updating the CNN is a reasonable way to make the result can close to what it should be. However, no matter how to adjust parameters in CNN or optimizer or even the structure of the CNN, the problem still exists.

In our assumption, the problem may cause by the size of batch input into the model. In our experiments, the batch size is set as 64, which is quite usual batch size in CNN. When the optimizer wants to lower the loss between a batch of generated images and target images, there is a smarter way for model that making all generated images combine all target graphic features also become that shape. In this way, the loss could be lower to a point, even is the local minimum point. If the loss reaches the local minimal through that way, the loss will be stuck, the training model will be a collapse, and the generated images hardly be various even be able to iterative.

There is a solution can solve problems caused by batch size, which is set batch size as smaller as it can be. When the batch size becomes 1, the fluctuation between batch and batch is obvious enough, which can prevent the loss stuck in a local minimum point. There is a test has been done which batch sized is one and four. The results are shown in Figure3.42. The model is also collapsing, but not caused by CNN part. The DNN part causes the model to collapse. When the batch size becomes smaller, the random noises cannot contain the distribution features. There is a confliction, the adjust toward batch size cannot guarantee two parts. There should be more constraint conditions in the model so that both parts in the encoder can work well. However, we still do not find a reliable solution. That part will be our future work, to explore a method to ensure the transfer learning and shape conversion.

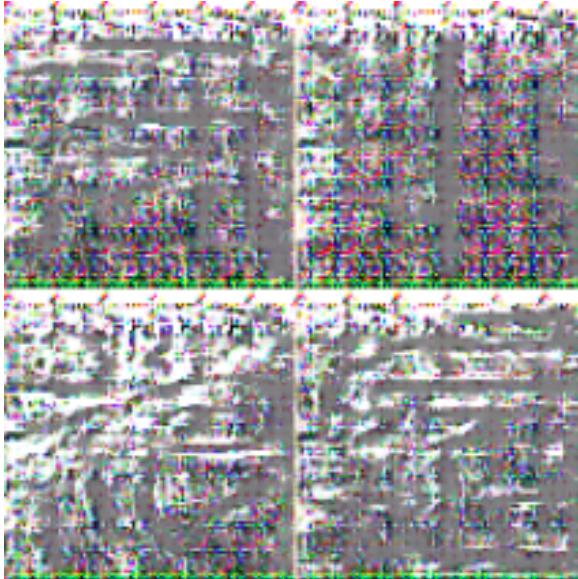


Figure 3.42: The result of transfer learning model which batch size is 1.

3.4 The Classification

In order to evaluate the accuracy of our generated images. We build the classification to classify image data into types, each type refers to a small seal style characters character. The classification used CNN as the main part, which is verified useful to extract the features of image data and can be used in the classification.

The target categories in the classification are 872. Classifying into 872 categories because there are 827 different kinds of oracle bone inscriptions characters have the matched small seal style characters. Because the small seal style characters are unified character, so each character only has a standard image. In other words, the 872 categories mean we only have 872 training data. In that case, the classification cannot be trained well, we even have no more data to split off a testing dataset. In there, we used the Elastic deformation algorithm to achieve the expansion of the number of image data.

3.4.1 The Elastic Deformation Algorithms

The expansion of the number of image data is usually used in classification training. That algorithm was proposed in 2003 and firstly used to attempt the expansion of the MNIST dataset. Using that algorithm to expand the number of data and using CNN to classify the dataset. The accuracy of classification rose up from 98.4% to 99.6% [20], which means that using the elastic deformation algorithm to expand dataset can improve the classification work of CNN. In P. Y. Simard 's opinion, although the improvement range is not large enough, the basement of MNIST classification is high enough [21]. If applying that algorithm in another situation, the accuracy can get a breakthrough result.

There are two steps to realize the elastic distortion algorithm. The first step, generating two random numbers from -1 to 1 for each pixel in the image. That two numbers refer to a 2-dimensional vector, the original pixel should add to that vector to get a new pixel in the new image. The second step, generating an $n * n$ sized convolution kernel which follows the normal distribution and standard deviation present as σ . The parameter n and σ can be set by the user, which may influence the effect of the algorithm directly. The parameter n controls the distortion scope and the σ controls the similarity with the original image. The testing results

are shown in Figure 3.43. We need to control the generated image close to the original one, so after tests, we choose the value of n and σ is 20 and 4.



Figure 3.43: The result of elastic distortion algorithm. The top image is the original image. In Each row, the n from left to right is 20, 40 and 60. In each column, the σ from top to bottom is 4, 8 ans 12.

We also test how much times should we expand the original dataset. The results showed in Table 3.1. From that, we can know when the times from 5 to 20, the accuracy of classification keep increasing and after 20 the increase is stopped. Considering an increase the number of data has no more obvious benefits, we expand 20 times of our dataset.

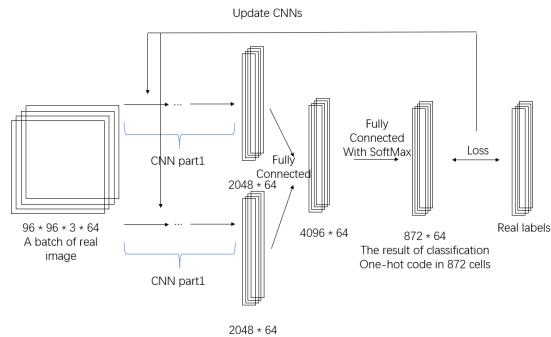
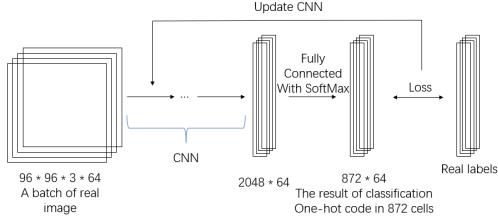
Table 3.1: The classification accuracy about dataset expansion

Times	The Accuracy
5	37.6%
10	57.3%
20	73.4%
30	72.1%

3.4.2 The Design of classification

Solving the training data problem, we focus on build training and testing dataset. The testing data set was chosen from the whole dataset randomly, which occupy one-seventh of the whole dataset. The other parts were written into the training dataset.

The structures of the classification model are shown in Figure 3.44 and Figure 3.45. There are two versions structure we designed for classification, both based on CNN. The first one is classic CNN classification and the second one imitates the AlexNet structure. The AlexNet has an upside part and a downside part, both of them is a CNN to capture the feature of images. Ideally, using multiple CNN can capture the different kinds of graphics features of images, because each CNN contains different



parameters[22]. Then combine these features together in a fully connected layer, which process can improve the performance of the classification model. In Google's ImageNet competition, AlexNet had a quite good performance in image classification field. It decreased the error rate from 25% to 16%. In our project, the characters image dataset is not similar to the ImageNet dataset, so we attempt to use two classification networks for getting a better result. Besides, we can verify that if the AlexNet can provide a better result in the character classification field than the traditional CNN.

3.4.3 The Experiment of Classification

The challenges in the small seal style characters classification are that the number of categories and the variety of the data. The number is too large to ensure accuracy. The Image V5 classification image database has 2.8 million object instances in 350 categories [23], which provide by Google to discover the classification of a complex image. Comparing with the Image V5, we have about 15 thousand training image instances in 827 categories. The accuracy of Google's classification competition keeps around 77% in recent year.

The accuracies of two versions of the model are shown in Table 3.2. The accuracy is calculated by the right rate in testing data classification. It is calculated when the accuracy has no obvious fluctuation and average the recent three times accuracy. The performance of AlexNet is

Table 3.2: The accuracy of each classification model

Model	The accuracy
The CNN classification	73.2%
The AlexNet classification	76.7%

better than the traditional one but the increase of accuracy does not reach the level what we

expect before. The main reason should be that there is overfitting phenomenon in the AlexNet. Although there are 9216 pixels in each character images, most of them are pure white pixel. There are only 50% or fewer pixels contain the pivotal information which can help classification to classify the image data. When we use two parts CNN to capture the features of character images, some useless features or wrong features may influence the classification. Besides, the number of data is so less that can not avoid overfitting in such a huge classification.

The reason why we reach the same accuracy level of Google's competition with a worse condition should also be that the small seal style characters are much simpler than other real images. The small seal style characters image data only have black and white pixels and the difference between character and character is the graphic structures. Considering the pivotal pixel number of a small seal style characters image is also smaller than a real image, the accuracy of our classification without the overfitting problem should be even better. In the future work, we can still keep improving the classification model to get better results.

After we got a reliable classifications, we used it to classify generated character images by CycleGAN. We chose images generated by Cycle-GAN because those images are most similar to the target images intuitively. The accuracies showed in the Table 3.3. Considering that there is about 24% error is caused by classification, we add a error range on the accuracy. Generally, our generator accuracy reaches 25%, which is higher enough in this project even better than what we expect. There are still some problems in the classification result. Our trained classification

Table 3.3: The accuracy of Cycle-GAN model

Model	The accuracy
The CNN classification	24.7% \pm 6.18%
The AlexNet classification	% 25.8% \pm 6.42%

can classify data into 872 categories while 872 is the number of pairs oracle bone inscriptions and small seal style characters in our datasets. If we want to translate those unknown oracle bone inscriptions characters, the classification need has the ability classify data into more than 872 categories. Ideally, it can classify data into about 3000 categories and keep an acceptable accuracy, which is the number of oracle bone inscriptions experts have already found. That should be our future work.

3.4.4 Using classifier to detect style conversion

Traditionally, in order to achieve the transformation or find the potential relationship from one type of shape to another type of shape, it always needs some techniques to encode the input shape and then decode to output shape. In this chapter later described the GAN model and Auto encoder model are typical examples, however, the modified version of CNN can achieve the goal above in a different route. Typically, the CNN model only takes one image and gives the classification output. To achieve this goal, the typical CNN model can be modified to have two input, one is Oracle bone inscriptions and another one is Ancient bronze inscription, and a Boolean label to indicate whether they belong to the same character. If this model can predict the result precisely, later, when a new unrecognized Oracle is unearthed, this model can judge the oracle and the known verses, and obtain the correct corresponding essays to help the expert identify.

After repeated scrutiny, two feasible ways to modify the CNN model was chosen. One is merged two input images as a two-channel matrix before the convolution layers. Another one has two separate convolution and max-pooling layers. This architecture merging the two input before the fully connected layer. The structure of the two models is described in Figure 3.46 and 3.47

respectively .

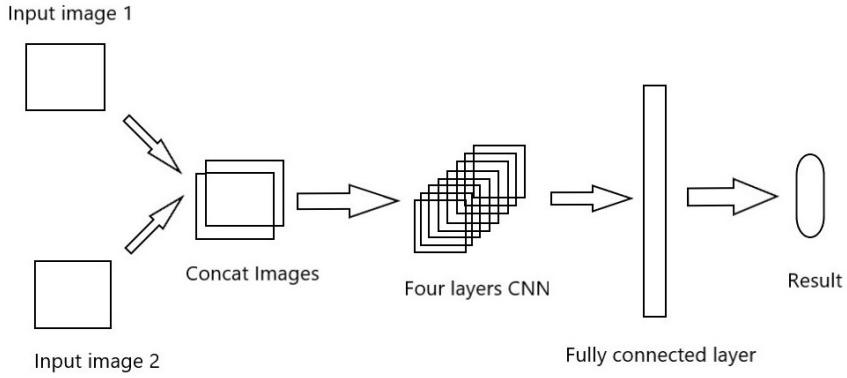


Figure 3.46: The modified CNN model with single convolution aisle

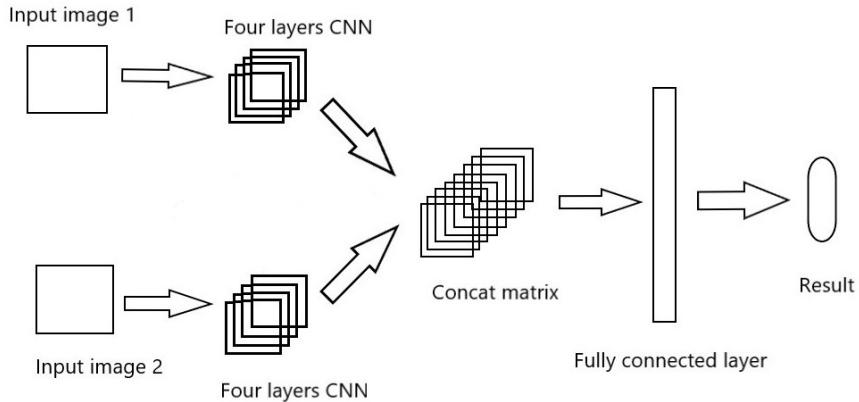


Figure 3.47: The modified CNN model with double convolution aisle

The first type of altered CNN model (single convolution aisle) was selected to be an experiment first. At first, two given images are merged into two channel matrices, then pass through four convolution layers and max-pooling layers and they pass to the fully connected layer. For the convolution layer, the kernel size is 3×3 for the initial value was chosen because the input pictures have been adjusted to 96×96 size is a relatively small size. The dataset also needed to be modified to satisfy this model. Based on images that have been binarized and normalized before, for balance the quantity of 0 and 1 label, for each Oracle bone inscriptions, three accordance Ancient bronze inscription with different writing was chosen and three other Ancient bronze inscriptions which are different with the Oracle bone inscriptions was chosen. For instance, for an Oracle bone inscription 安, three Ancient bronze inscription with different 安 was chosen, and other three random Ancient bronze inscription like 宝, 巴, and 匕 are chosen. The 12 images and the labels combine together to become a unit if the dataset, later the unit need to be shuffled to increase the randomness of the whole dataset.

At the training stage, only 90% of the whole dataset was taken to train the model. The rest 10% is used to test the accuracy of the whole model. The characters in the rest 10% dataset never appeared in the training dataset. After 20 epoch of training, the whole network was convergence base on the loss value reflect by the softmax_cross_entropy_with_logits function. Later, the testing reflects an average 59% percent of accuracy, the precision and recall are 55% and 54% respectively.

The second type of altered CNN model (double convolution aisle) was trained later. In this model, two images are passed through to convolution layers separately, after being convoluted,

the two matrices are merged together then pass to the fully connected layer. Because the image is merged after the convolution step, the quantity of the variables (weights and bias) is twice the first, consequence the training needs to takes more steps. The number of epochs was added to 30 to adjust this model. After the whole training complete, the testing reflects an average 58% percent of accuracy, the precision and recall are 54% and 52% respectively. The table below illustrates the performance of the two models in detail.

Table 3.4: The details of the performance of the two models

	First type of model(single aisle)	Second type of model(double aisle)
Accuracy	$59\% \pm 2\%$	$58\% \pm 2\%$
Precision	$55\% \pm 2\%$	$54\% \pm 2\%$
Recall	$54\% \pm 2\%$	$52\% \pm 2\%$

Other methods are also used to improve the accurate reading of the two models like increasing the window size to $5*5$ to the two models respectively. However, this method does not optimize the performance, both two models converge in a very dramatic speed and stop converge with a relatively high loss value. The reason for such a phenomenon might be enlarging the respective field decrease the accuracy of the model to scan the detail of the image.

In general, the feedback of these two models are not as good as expected, the first model is a little bit better than another one because it takes a few variables and easier to training. Other possible ways to increase recognition accuracy in the future might increase the size of the quantity, more convolution layers and optimize the method of construct the dataset.

3.5 Chapter Summary

In this chapter, we explored the evolution of the Chinese characters using a generative model and validate the results with a classification. The results of the DCGAN model and the CYCLE-GAN model by inputting random noise are significantly better than those produced by the encoder reducing the image to 200 dimensions. The model generated by CYCLE-GAN model gradually changed from the Oracle bone inscriptions style to the small seal inscriptions style, but there are still details that cannot be revealed. The reason for these results may be 1. The spatial similarity between the two datasets is too low. 2. The number of data sets is not uniform, and training will continue after balancing the data sets in future work. 3. The limitations of the model itself, such as the randomness generated by the adversarial network.

We also explored the classification about small seal style characters as the evaluation of generators. The performance of classifications are quite good while it can only classify data into 872 categories with a acceptable accuracy. That is the elementary exploration in the classification field. We still need to do more research and experiments.

Wei Du, Linghua Gong, and Zechen Feng are all contributed to this chapter. Wei Du contributed to U-Net Model and Classification. Zechen Feng contributed to Autoencoder Model, Variational Autoencoder Model, DCGAN Model, VAE/GAN Model, and CYCLE-GAN Model. Linghua Gong contributed to DCGAN Model, Transfer Learning, and Classification.

Chapter 4

Dataset Establishment and Image Inpainting

The oracle bone inscriptions rubbings are precious research materials in the area of oracle bone inscriptions. The characters do not appear solely in rubbings, instead, they are associated, and constitute meaningful sentences. For most of the collection book of oracle bone inscriptions rubbings, although the contents of rubbings are commonly recorded, hardly any collection books have recorded the order of characters in each sentence. In our project, we attempt to establish a rubbings dataset which records the order of each character in the sentences. In addition, as oracle bone inscriptions characters have the similar features to painting, it is possible to capture the features of characters by convolution neural network and repair them by generative model.

This section mainly presents the process of establishing our oracle bone inscriptions rubbings dataset and repairing damaged oracle bone inscriptions characters images. We improved the quality of collected oracle bone inscriptions rubbings by image processing. The treated images were then inputted into a trained neural network model called You Only Look Once (YOLO). This framework will label characters in the images so that we can get the content of the rubbings. The labeled characters are then sorted according to reading order. We also tried to repair some damaged characters appearing in the rubbings with a neural network based on Autoencoder and generative adversarial network.

4.1 Literature Review

We will introduce the supporting technology to establish dataset and repairing images. The object detection model YOLO and image porcessing will discussedd in Section 4.1.1 and 4.1.2. This section will not cover introduction to GAN model and Autoencoder model as they have been discussed in the former chapter.

4.1.1 YOLO

You Only Look Once (YOLO) is a powerful unified real-time object detection open source framework. Differ from normal Region-Convolution Neural Network (R-CNN) or fast R-CNN, which repurpose classifiers to perform detection, YOLO frame object detection as a regression problem aiming at bounding boxes and class probabilities. The model is mainly a convolutional neural network. The initial convolutional layers of the network are responsible for extracting features from the images while the fully connected layers predict the output probabilities and coordinates[24]. In YOLOv1 (first version), the number of convolutional layers is 24 followed by 2 fully connected layers. The number of convolutional layers raises to 53 in YOLOv3 with multi-scale training tackling the problems of only perform well on pictures that has the same size with the training pictures, and cannot recognize small objects, which remained unsolved in YOLO1[25].

As shown in Figure 4.1, in YOLO system, it divides the inputted image into an $S \times S$ grid and catches the center of an object. Each grid will be responsible for detecting the object that falls in its cell. It predicts several bounding boxes for that object and the system will calculate the confidence for each bounding box and conditional class probabilities [24]. The loss comes from localization error and classification error. It optimizes for sum-squared error in the output. To remedy the problem that localization and classification enjoy the same weights and grid cells containing no object affecting gradient from cells that do contain objects, the loss from localization is enlarged while loss from boxes contain no objects is reduced by adding two parameters $\lambda_{\text{coord}} = 5$ and $\lambda_{\text{noobj}} = .5$. In addition, the width and height of bounding box are replaced by square root of them, addressing the problem that the loss function should be less sensitive for large box compared to small box[24].

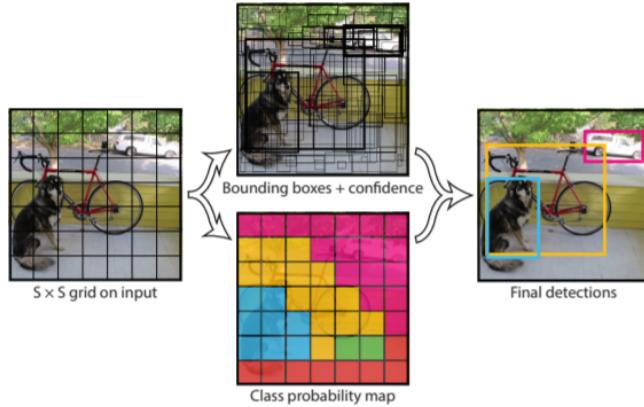


Figure 4.1: YOLO system divides the sample picture into 7 grids. There are 3 main objects—dog, bicycle and car. The system first will predict tons of bounding boxes, class probability and confidence then decides the final detections.[24]

4.1.2 Image Processing

Image processing, also known as digital image processing, is the technology in which images are analyzed by a computer to achieve the desired results. Image processing general exponential character image processing. Digital image refers to a large two-dimensional array captured by industrial cameras, cameras, scanners, and other equipment. The elements of the array are called pixels, and their values are called gray values [26]. In this project, we apply threshold process and noise reduction.

Threshold Processing

Threshold processing is a very common operation in image processing. In threshold processing, we choose a threshold value and for every pixel, if the gray value in this pixel is bigger than the threshold value, then increase the pixel's gray value into 255, otherwise decrease to 0. This operation helps to wipe off the influence of different color in the picture and can enhance and sharp margin of oracle character, which makes the inner structure of characters clearer.

Noise Reduction

In our project, median filter and morphological process are the main ways of noise reduction. Both these tools are aiming at small holes. In addition, morphological process also used to tackles little cracks and damaging images (later used in producing damaged images).

Median filter is a non-linear smoothing technique, widely used to reduce noise. The concept of median filter is for each pixel, set the gray value in this pixel as the median gray value of its adjacent assigned-size pixel area(for example, 3×3 square area shown in Figure 4.2).

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Neighbourhood values:
**115, 119, 120, 123, 124,
125, 126, 127, 150**

Median value: 124

Figure 4.2: X,Y are the coordinate of the pixel, the figure above is a 3×3 area, its median value is 124, so the center pixel's gray value is assigned to 124[27]

We mainly used erosion and dilation these two morphological processes. Totally speaking, erosion and dilation are the processes of convolving the image with the kernel, where the kernel is a small solid square or disk with a reference point in the center. Dilation is the operation of finding a local maximum value. When convolving the image with the kernel, it will find the local maximum value in the kernel area and assign the maximum value to the center reference point, which can enlarge the area with high gray value. This operation has outstanding performance in dealing with tiny cracks and holes in a relatively big area with high gray value (see Figure 4.3).

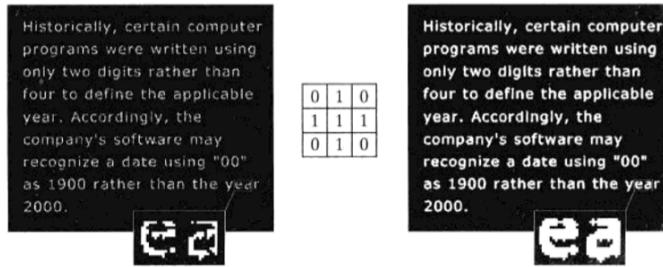


Figure 4.3: The effect of dilation operation. After dilation operation, the white areas are enlarged. Some ruptures are connected[27]

Conversely, erosion is the operation of finding a local minimum value. During the process of convolution, the minimum value in the kernel area is assigned to the center reference point, which dwindles the area with high gray value. Erosion leads to the boundary shrinkage of the area with high gray value. To some extent, it can be used to eliminate some small and meaningless target object (see Figure 4.4).



Figure 4.4: Erosion operation with different size of kernels. The bigger kernel can eliminate the thicker line. The center square area has been shrunken at the same time[27]

Combining these two operations with different order can achieve different effects and reduce damage caused by these two operations. For an image, first executing erosion operation then

dilation operation, which is called open operation, has the ability to eliminate small area outside the target area. On the contrary, when dilation operation is followed by erosion operation, this process is named as close operation. Close operation is used to clear the small area inside the target area (shown in Figure 4.5).



Figure 4.5: The left image is the effect of the open operation. The right image is the effect of the close operation

4.2 Worklog

The process of establishing oracle bone inscriptions rubbings dataset can be divided into four stages. The first step is the collection of oracle bone inscriptions rubbings. The next step is preprocessing of rubbings images. Then we discern the labeled characters in rubbings images in step 3, and finally, sorting labeled characters following reading order. In addition, we attempt to repair the damaged image of single characters based on deep learning.

4.2.1 Collection of Rubbings

We searched on the internet, although there is a large amount of oracle bone inscriptions rubbings images, the most majority of them are repetitive and without translation. In that case, we give up crawling oracle bone inscriptions rubbings on the internet, instead, we bought some oracle bone inscriptions rubbings collection books, in which the content on rubbings is already identified. We scanned the rubbings images into the computer as our initial data. The total number of our rubbings images is 100.

4.2.2 Preprocessing of Rubbings Images

Oracle bone inscriptions were carving on bones or turtle shells thousands of years ago. These bones and shells had been buried underground until the archaeologists dug them up. During such a long time, there are always all kinds of breakage and depression on bones and shells, which has influence on recognizing the carved oracle bone inscriptions characters on them. The oracle bone inscriptions rubbings dataset we own is suffering this problem. There are several images of oracle bone inscriptions rubbings in our dataset are damaged so seriously that we even cannot tell whether there exists a character in some parts of the picture. What's more, all the pictures of rubbings have more or fewer noises that may impede the output of YOLO system. In that case, we have to do some Image Processing, mostly are noise reduction and threshold processing, before we input them into YOLO system.

Unify Color

First of all, we converted the initial images to gray images by threshold processing. The purpose of this step is to unify the color of all the oracle bone inscriptions rubbings. The main problem in threshold value is how large the threshold value should be, as the threshold processing also has an impact on the boundary of the target object region. We focused on the pixels around the oracle bone inscriptions, then we chose the gray value of some pixels near the boundary as the threshold value and contrast the effect. After comparing, 200 is proved to be the most suitable threshold value, as when the threshold value is 200, the target object areas (oracle bone inscriptions) remain almost unchanged. The effect of this process is shown in Figure 4.6.

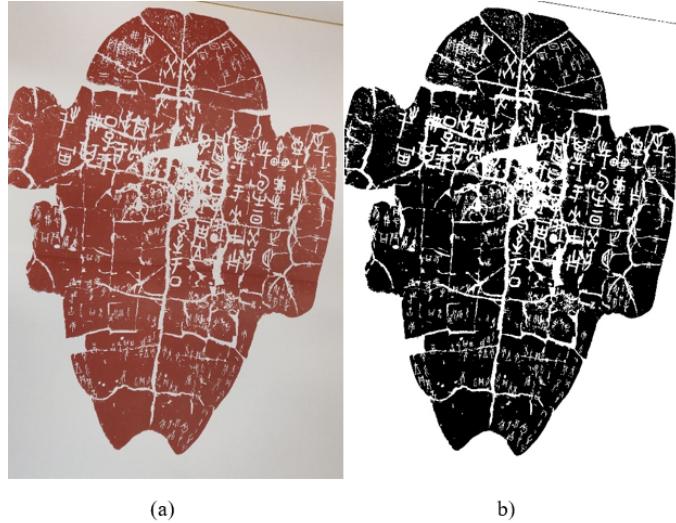


Figure 4.6: (a) initial rubbing image and (b) image after threshold processing

Tackling Noise

As we mentioned above, most of the oracle bone inscriptions rubbings are outworn. we view these small holes and cracks as the latent factor that may lead to an unsatisfactory result or even failure in the later stage of discerning the single characters. In order to minimize the impact of these flaws, we have tried several ways.

In book Digital Image Processing written by Rafael C. Gonzalez[27], we found that small holes in rubbings have a similar feature as Salt-and-pepper noise. Salt-and-pepper noise, also known as impulse noise, is a type of noise often seen in images. It is a randomly occurring white or black dot, which may be a black pixel in a bright area or a white pixel in a dark area (or both), just like small holes, which are white pixels in the dark area. According to the book, the median filter is widely applied in tackling this kind of noise. In that case, the median filter is our first choice.

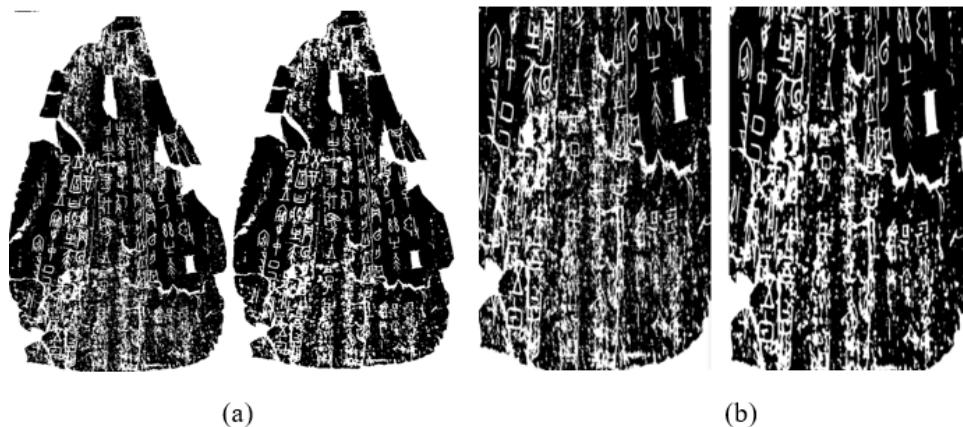


Figure 4.7: (a) is a comparison between initial image (left) and image processed by median filter with a 4*4 kernel (right), (b) is partial enlarged detail

We also attempt to use the morphological process. Compared to the median filter, using open operation (first erosion and then dilation) has almost the same result as the median filter. We finally chose the median filter to reduce noise for all images collected.



Figure 4.8: (a) is a comparison between initial image (left) and image processed by open operation with a 5×5 kernel (right), (b) is partial enlarged detail

Cracks are much intractable than holes. the majority of cracks are intersected with target oracle bone inscription characters. To some extent, these cracks have become a part of these characters so that image processing is not powerful enough to tackle them. In order to tackle cracks, we try to train model based on Autoencoder and GAN. This part will be discussed alone in Section 4.2.5.

4.2.3 Discerning Characters

The total number of treated images is 100. we pick up 40 of the them as the training set for YOLO and another half as the test set.

YOLO requires several complex steps to learn from the training set. Each image in the training set should be accompanied by a txt file recording the target objects' coordinates and class information. Fortunately, there is a software called LabelImg specifically for YOLO's requirement. It can record label information added by the user in the image as an XML file. We just used a simple python script to convert it to txt file.

The biggest problem is labeling images by LabelImg. On the one hand, every member of our team has never learned oracle bone inscriptions before, even our supervisor has no experience in discerning oracle bone inscriptions. On the other hand, oracle bone inscriptions were carved randomly, although the sequence of oracle bone inscriptions sentences obeys some hidden rules, the position of each sentence and the size of characters in each sentence is irregular. In addition, as we mentioned above, rubbings images are suffered from small holes and cracks intersected with target characters. All this makes labeling images in training set more difficult.

In order to make a perfect training set, we ask our friend who major in archaeology for help. Our supervisor also provides us a useful website which collects almost all the single identified oracle bone inscriptions character and their corresponding modern Chinese language. With the help of them, finally, the training set is worked out. Two samples of the labeled rubbings are shown in Figure 4.9. After parameters configuration, training was executing. The following work is getting the coordinates of the characters in each rubbing and sorts them with right reading order.



Figure 4.9: labeling images by LabelImg

4.2.4 Sorting Characters

During the discerning process, each target object's coordinate (top, right, left and bottom) is already calculated by YOLO model, so we added a recording function to YOLO source code and wrote all the target objects' coordinates into a txt file. This file is read by the python script and with tool OpenCV, each character was cut down from the initial image. Each character cut from the initial image is square.

Making sure the right reading order is not easy. There are basically 3 types of rubbings and the reading order is different in different type rubbings as shown in Figure 4.10.

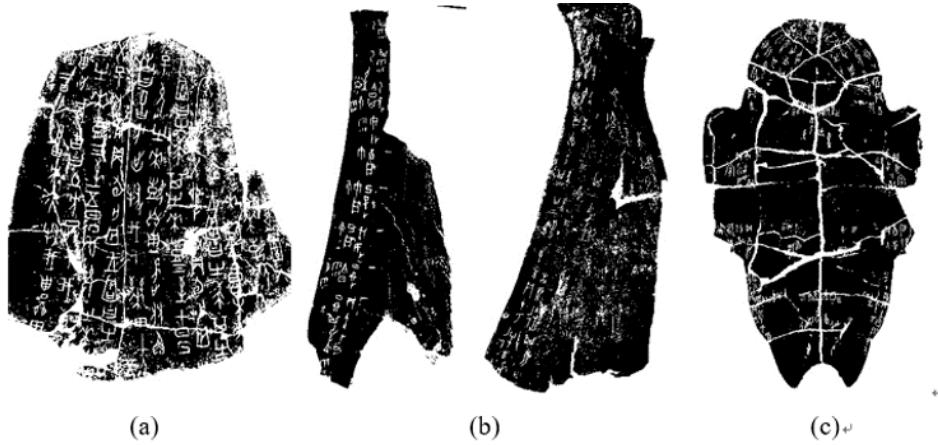


Figure 4.10: Three types of oracle bone inscription rubbings

Figure 4.10 (a) is the simplest type. All the oracle bone inscriptions characters were craved from right to left, top to down. Figure 4.10 (b) has two conditions, one is from top to down and the sentence curves following the shape of the rubbing, another is that the rubbings are divided into several parts from top to down and in each part, the reading order is from right to left. Figure 4.10 (c) is the most complicated. It is bilateral symmetry—sentences away from margin should be read from right to left, top to down in the left side and from left to right, top to down in the right side. In the margin area, sentences are arranged following the curve of the

margin, from top to down and from margin to the center of the rubbing. There is a example of the reading order of one sentence near the margin of a type c rubbings shown in Figure 4.11.



Figure 4.11: Correct reading order for a type c rubbing sample

The first thing we do is to divide content into different areas. Each area should only contain one complete sentence. The core idea of the dividing algorithm is spreading. Store all coordinates (containing top, left, right and bottom 4 values) in a list and sort the list by top value, find the character on the top (having the smallest top coordinate) and then calculate the diagonal length of the selected square, for all the elements in the list (labeled characters), if the distance between the left-top point of selected character and element 's left-top point is smaller than $2.5 \times$ diagonal length, then this element belong to the same area with the selected character. Remove elements that already divided into an area and change the selected character to the last removed element, when the list is empty, the dividing task is finished.

The following step is sorting characters in an area. We learnt the correct reading order from the research we have done. Each sentence is divided into several columns. Aiming at different written laws in different regions, we designed three algorithms to sort sentences written in different regions. The main idea in these three similar algorithms is recording the coordinates of each character in a sentence, getting the first character 's coordinates (usual character having the smallest top coordinates in a sentence, choosing the most left or right one according to the position of the sentence) and then selecting the characters with coordinates located in specific area of a sentence into one column until there is no character does not belong to any column.

This task is easy in rubbing type a and b, as each type a or type b rubbing only needs one algorithm. However, for type c rubbings, three kinds of algorithms are needed and each algorithm has to be adjusted when the areas it tackles are in the left side or right side of the rubbings. When tackling type c rubbings, the whole image has to be divided into the left part and right part. Areas near the margin of the top side, the margin of the bottom side and other positions have to use different algorithms. When sort sentence located in the margin of the bottom side or top side, the algorithm has to consider allowable deviation for curved columns.

4.2.5 Repairing Characters Intersected with Cracks

We built a neural network to address this problem. The model is trained by collected complete images and damaged images forged based on the complete images.

Concept and Structure of Model

The concept and structure of our GAN model are inspired by Deepak Pathak[28], he provides a model called context encoder. Similar to Autoencoder, context encoder has an encoder extracting features from images and a decoder which reestablish image according to feature space [6]. Differ from context encoder, in our model, we replace decoder by a GAN to generate images based on features extracted by the encoder.

In GAN, generator in this model consists of a fully connected layer and 4 deconvolution layers. Before damaged images of single characters are inputted into the generator, they will first go through the encoder network with one convolution layer, three batch normalization layers, and three fully connected layers. Discriminator has one convolution layer, three batch normalization layers, and one fully connected layer. Discriminator takes undamaged images of single characters as its input. The detail of this model is shown in Figure 4.12.

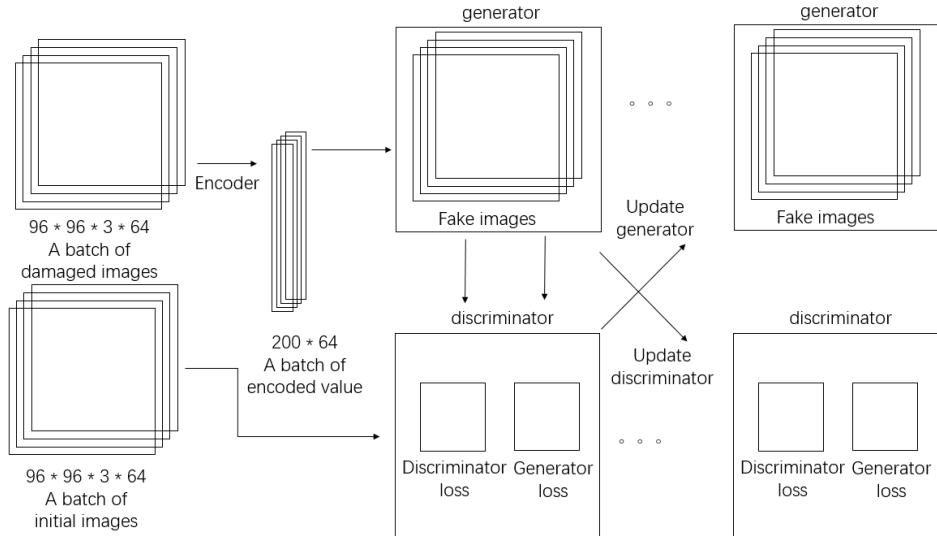


Figure 4.12: The structure of our image inpainting model

Produce Damaged Images

Images of single characters are already crawled, after a single picture inversion process, they can be inputted into the discriminator network. The input for the encoder is more intractable to collect. As the limitation of the number of our rubbing dataset, there are not enough oracle bone inscriptions characters intersected with cracks. In order to provide abundant input for the encoder, we tried to damage the images of single characters we already have.

The initial way is selecting two values in the range of the size of the undamaged image and changing the gray value of all pixels whose y coordinate located between the two values selected to 255. Compared to damaged images in real, the margins of cracks in these imitated damaged images are too smooth. In order to make the imitated image more realistic, we improved the first plan: select several pixels located in a quadrate area in the undamaged image randomly, change the gray value of the selected pixels and their adjacent area to 255, which can create small holes in the initial image. Using dilation operation to enlarge the small holes created before so that they connect to each other and compose a curved line. Finally, erode the dilated

image to recover the other parts of the single character. The result of the improved way is much better than the initial way (see Figure 4.13).

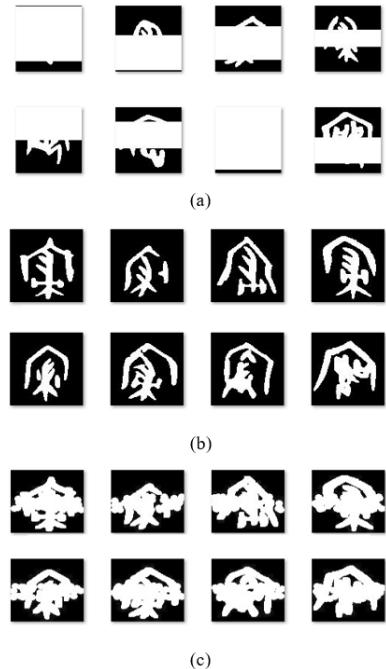


Figure 4.13: Initial samples (b), result samples of the initial way of damaging images (a) and improved way (c)

The training result will be discussed in Chapter 4.3.

4.3 Evaluation

We evaluated the effect and result of our work. In Section 4.3.1, we summarize the results of all the experiment in the process of establishing rubbings dataset. In Section 4.3.2, we make a comparison between the image generated by our model and target characters.

4.3.1 Establishing Rubbings Dataset

Compared to collection books published by authorities in archaeology, in which only the rubbings of animal bones or shells and the meaning of its content are collected, without telling readers which oracle bone inscriptions sentence matches which specific meaning in modern Chinese language, our attempting is pioneering.

Median Filter and Morphological Process

The effect of the median filter is not so satisfactory. The problem is how big the kernel should be. For median filter, when the kernel area is small, it can only eliminate extremely small holes, bigger holes are still remained, although shrunken. On the contrast, if the kernel area is bigger enough to fill bigger holes, it will also eliminate some small characters and affect the structure of large characters.

As to open operation, when the kernel is small, there are still some residual holes with almost the same initial size with the oracle bone inscription character, while with the bigger kernel, some target characters are affected as well. Although both the median filter and open operation cannot reach perfect output, the current images processed by any of these two operations is much clearer than before.

Discerning Target Characters

The YOLO model has been trained 11000 epochs. The initial loss for this model is 2300 and after 4000 epochs, the average loss reduces to 15. However, this level of average loss is far away from the average loss of a successful model (lower than 0.01). The following 5000 epochs only reduce the average loss from 15 to 11 and when it comes to 11000 epochs, the average loss even picks up to 12. With this YOLO model, the accuracy for test images is around 75 percent recall and 60 percent precision and cannot be improved. Figure 4.14 shows the result of a type (c) rubbings produced by YOLO.

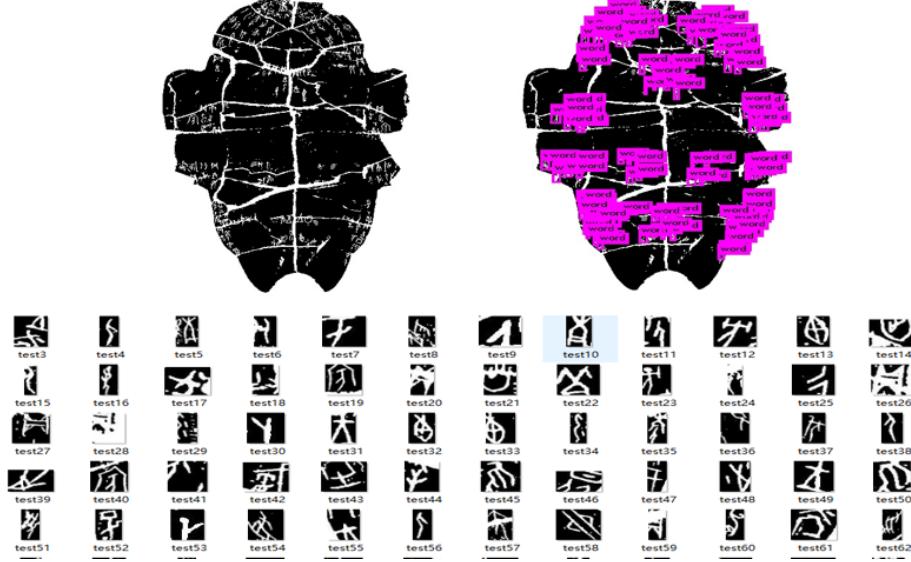


Figure 4.14: Effect of YOLO and characters cutting from initial rubbing

Sorting Labeled Characters by the Reading Order

In several test images, our algorithms have high accuracy. The accuracy of the algorithm dividing area is 80 percent, result from that some areas are so close and for characters labeled by YOLO, the accuracy for sorting characters following the reading order is 80 percent.

There is a sample of sorted characters in Figure 4.15. The drawback of our algorithms is that it struggles to do the batch process. As there are several different kinds of rubbings and each type needs its specific algorithm to sort the character by the reading order, especially when tackling bilateral symmetry rubbings, we need to manually judge the corresponding algorithm for different regions. In addition, our algorithms work well on the output of my YOLO model, in which the identified sentences are not complete. The accuracy of complete sentences is not available. These drawbacks prevent us from quickly expanding my dataset. A larger number of oracle bone inscriptions rubbings are needed to overcome these drawbacks.



Figure 4.15: (a) is a sample of sentence. Red boxes are the area bounding by Yolo, the green number is the right order of these characters in this sentence. (b) is the order sorted by an algorithm. Oracle2 means it is the second test rubbing, 1 means the first sentence and the number in the last means the position of this character in this sentence.

4.3.2 Repairing Damaged Characters

As shown in Figure 4.16, the result of our model is not excellent. The input of this model are 64 different undamaged images of single oracle bone inscriptions characters and their corresponding damaged images, however, the output only contains one kind of characters. In addition, the model collapsed during the training. After 47 epochs, the shape of the character disappears totally. The images generated by this model before collapse have a rough sketch of the target character, the majority of the damage parts produced by us has been repaired, but the whole character is not clear enough.

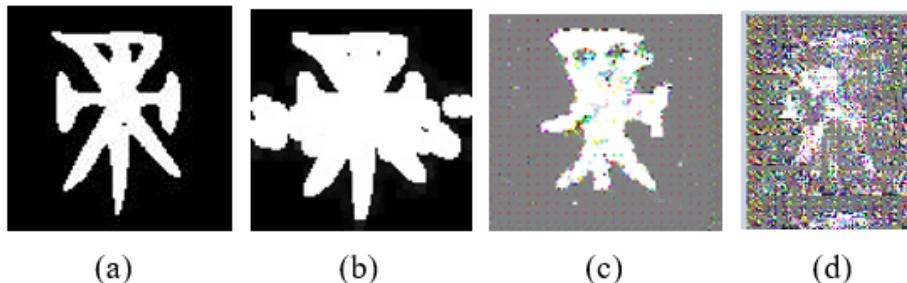


Figure 4.16: Comparison between target character (a), input damaged image(b), the result generated by GAN in epoch 30 (c) and epoch 48 (d)

4.4 Analysis of Unsatisfactory Result

Aiming at the unsatisfactory result, including holes residues or target elimination in noise reduction, high average loss in the process of training YOLO model and model collapse in image inpainting, we did research and find the possible explanations.

4.4.1 Holes Residues and Target Elimination

The reason why the median filter eliminated some target characters and affected the structure of oracle bone characters is that the target areas actually have the same gray value as the salt-and-pepper noise. The median filter is good at eliminating noise having different gray value with the background. When tackling with rubbings images, both the small holes and target oracle bone inscriptions characters are viewed as noise, which contributes to the unsatisfactory effect.

As to open operation, the main problem is that the oracle bone inscriptions characters in one rubbing are in different sizes. Some characters are in a similar size with small holes, so that when eliminating small holes, the characters of a similar size with these small holes will be eliminated as well.

4.4.2 High Average Loss

Compared to the successful training process on the internet, the size of the training set is the main factor that may cause higher average loss. The models having average loss lower than 0.01 at least have ten thousand images as its training set, while there are only 40 images in our training set. Unfortunately, there are no more oracle bone inscriptions rubbings available, enlarging the size of our training set is unfulfillable in a limited period.

4.4.3 Model Collapse

The explanation for the model only generating one kind of oracle bone inscriptions character is that the learning rate of the generator and discriminator does not reach a balance. The generator gets feedback from discriminator and adjusts itself based on the feedback to fake discriminator out. When the discriminator learns slower than the generator, once discriminator gives better feedback in generating a single oracle bone inscription character, the generator will follow this direction and only generating that oracle character in order to minimize the total loss. Besides, there are also some reasons lead to such a results, which we discussed in the the former chapter.

There are so many factors which can lead to this result. In limited time, the reason why the shape of characters disappears is still undiscovered.

4.5 Chapter Summary

In this chapter, we claimed the challenges we met and the approaches we used to overcome them during the process of establishing our oracle bone inscriptions rubbings dataset and image inpainting. We have designed some algorithms to sort the characters appearing in the rubbings which work well in the test set. We also developed a structure of neural network aiming at repairing damaged images. There are still some shortcomings in our project, such as the unperfect result from YOLO caused by the lack of data and the collapse of our model after 47 epochs. **This chapter is contributed by Yikun Wu, with the help of Linghua Gong.**

In the next chapter, we will introduce the research we have done in the studying of oracle bone inscriptions structure.

Chapter 5

Decompose Oracle Bone Inscriptions

5.1 Literature Review

As is described in chapter 1, each ancient and modern Chinese characters are composed of different strokes and the strokes of the character are very important for understating the deep information and meaning of each character. Currently, there are about 28 commonly used strokes of Chinese characters[29]. Meanwhile, Oracle bone inscriptions also have commonly used strokes too, however, there is not a clear set about the commonly used Oracle bone inscription strokes and currently, all the existing strokes are stripped by a human being. This chapter will introduce an algorithm to stripping all the strokes in Oracle bone inscriptions and how Oracle commonly used stroke set is constructed and being clustered. With this stroke set constructed, and cluster commonly used strokes it can auxiliary to many problems. Previously, the Oracle bone inscriptions are recorded by different images, however, multiple images take a lot of memory space and handling those images will consume many computing resources. However, if every Oracle bone inscription are labeled by the code and position of different strokes it can save a lot of memory space and computing resource. Another example is about to extract the feature of Oracle bone inscriptions. Previously, to extract the feature of the Oracle bone inscriptions, CNN and other convolutional neural network are used to extract the feature by convolution, however, after labeled the Oracle bone inscription by stroked number and positions, it is possible to convert that information to the matrix for deep learning to analyze the potential relationship. For instance, previously the Chinese character like '猫' was labeled as 1488, and '狗' was labeled as 436, there is no correlation between these characters by number, however, according to the strokes these two characters have the same stroke '犭' which means they are all animals. The Chinese character is based on the shape which is different from many other languages, if the character was numbered as strokes, the correlation between them can be detected by machine in a relatively obvious way. This idea has been realized in the cw2vec model and has a good result in modern Chinese characters.[30]. What ' s more, the same strokes in different Oracle bone inscriptions can also help the expert to find the potential relationship between Oracle bone inscriptions and modern Chinese characters, then classification them to pave the way for future analysis work.

5.1.1 The Canny algorithm

For each Chinese character, the meaning of the character is majorly based on the shape. Consequently, extract the outline of the character whiling able to represent the shape of the character precisely and reduce the number of pixels to reduce the calculation. The canny algorithm is a good method to achieve this object. The object about the canny algorithm is detecting the outline of the whole image by using the Sobel operator. The Sobel operator is commonly used the operator for edge detection. It is a discrete difference operator to calculate the image approximate value of the gradient of the image brightness. After calculating the gradient of the pixels in the image, this algorithm will scan each pixel to get the points that have the max gradient value to compare to the surrounding points which have the same gradient direction.

Finally, this algorithm will choose the edge in the selected pixels by two thresholds given by the user.[31]

5.1.2 The Harris Corner Detector algorithm

The corner point can best represent the characteristic of the image information. Meanwhile, all the junctions between the strokes. So it is crucial to utilize related technology to detect the corner point. Harris Corner Detector algorithm is a good example. The object of the Harris Corner Detector algorithm is detecting the corner points of the whole image. The principle of this algorithm is using a sliding window to detect the gradient change in each direction. If the window detects the gradient change in multiple directions, there will be a large probability of corner points in the window.[32]

5.1.3 The K-means clustering algorithm

In order to extract the commonly used strokes in stroke sets, the clustering step is inevitable. The k-means algorithm is a commonly used unsupervised machine learning algorithm for clustering. So this algorithm was chosen for cluster the strokes first. The main idea of this algorithm is random sample K points in the whole data, then it calculates the class of each piece of data, after that, it calculates the center of gravity for each class. After several iterations, this algorithm will finish the cluster.[33]

5.1.4 The Hierarchical clustering algorithm

The hierarchical clustering achieves the clustering step via a different technique, so this algorithm was selected to compare performance with the K-means algorithm. The hierarchical clustering algorithm realizes clustering the data by constructing a tree base on the similarity between different classes. The key idea of this algorithm is at the initial stage, set each piece of data as a tree, and for each piece of data, calculate the similarity between different classes, then cluster merge the class which has the smallest similarity. These steps will be iterated several rounds until all the data is clustered in one class.[34]

5.2 Decompose the Oracle bone inscriptions

5.2.1 Outline the Oracle bone inscriptions

To decompose a single Oracle bone inscription, the first step is to retrieve the profile of a character to reduce the data quantity which can auxiliary the later calculation [35].

The outline algorithm is the canny algorithm within the two threshold values. After outlining, the profile of the character, the corner point of the image will also be labeled by using the Harris Corner Detector algorithm. The pictures below exhibit the origin Oracle bone inscriptions, the outline of the character and all the detected corner of the characters.



Figure 5.1: The origin characters, the outline of the characters and all the detected corner of the characters

5.2.2 Detect the concave points

When all the corner points are detected, the next step is detecting all the concave points of the character because according to observation, all the junction between strokes is the concave point [35]. Figure 5.2 exhibits that all the junctions are concave points.



Figure 5.2: The blue lines indicate some example of strokes in the characters, the red points indicates the example of conjunctions.

After this logical relationship is determined, the next object is to identify all the concave points within the corner point list.[35] The concave point can be judged by the signs of the dot product by the previous corner point and the next corner point. To combine the Oracle bone inscriptions, decompose situation, in order to find the previous and the next corner point for a detected point, a new algorithm is created. The method of this algorithm is for each corner point, it creates two separate point-set. Initially, these two sets start searching opposite directions. Each set can automatically search through the outline of the Oracle bone inscriptions by continuously adding the adjacent point for each point in the current set. Each set will automatically stop when it reaches another corner point. Finally, it can judge whether a corner point is concave or not base on the signs of the dot product of the previous and next corner point 's coordinate. The picture below gives two examples of the path about how two point-sets can detect the previous and next point by following the outline of Oracle. Figure 5.3 illustrates this process.

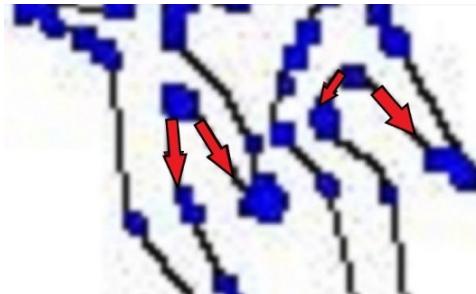


Figure 5.3: The method to find the previous and next corner point.

5.2.3 The relationship between concave points and junctions

When all the concave corner points are identified, one issue occurs that not all the concave point is the junction of the strokes. The picture below exhibits all the concave corner point in the Oracle bone inscriptions. Figure 5.4 exhibits all the concave corner point.



Figure 5.4: All the concave corner points in the character

In order to solve this problem, it is necessary to carry out some exploration on the relationship between the concave points and the junctions to find the correspondence between them. What's more, it is also very important to find the point pairs amount the junctions to realize stripping the strokes in a character, it is mandatory to find the pair of points to draw the cutting line and base that line to stripping the strokes. After observation, one possible relationship between them was detected. The relationship is all the point pairs are relatively close in straight line distance but need to go through a relatively long distance through the outline of the Oracle bone inscriptions. Figure 5.5 illustrates the relationship between the straight line distance and the travel distance between two points.

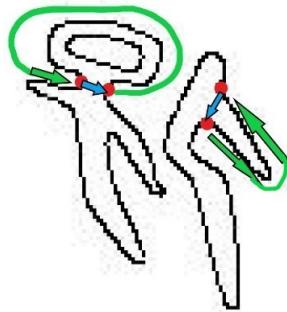


Figure 5.5: The green line indicates the straight line distance, and the blue lines indicate the travel distance between the two points.

With that relationship, the rest of the work was set the threshold values of these algorithms, the first value is the straight-line distance between two points, and another value is the travel distance on the outline base on the number of points it goes through. After adjusting the threshold values, all the point pairs were founded. Figure 5.6 illustrates all the junctions pair.

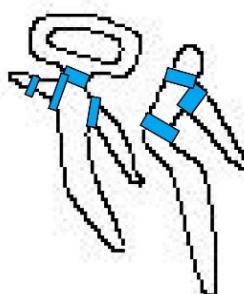


Figure 5.6: The rectangle indicates the funded point pairs connect by the rectangle and the strokes after stripping base on the point pairs.

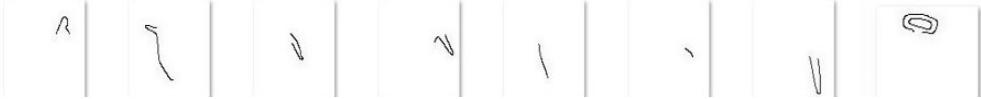


Figure 5.7: All the strokes stripped from the original character.

5.2.4 Strip the characters

The final step is stripping all the strokes on the characters by the founded point pairs. For the complete Oracle dataset, there are many very similar writings but have the same strokes, so for the concise of the stroke set, up to ten writing style was selected from each Oracle bone inscriptions. After processing, the algorithm stripping about 26,000 strokes from about 5000 Oracle bone inscriptions. What's more, there are also 8000 strokes that have an inclusion structure which was classified separately. Although the information provides by the inclusion structure is not strong enough, most of them cannot be called a complete stroke. However, some of them might also provide useful information. Consequently, the strokes which have inclusion architecture are classified alone.

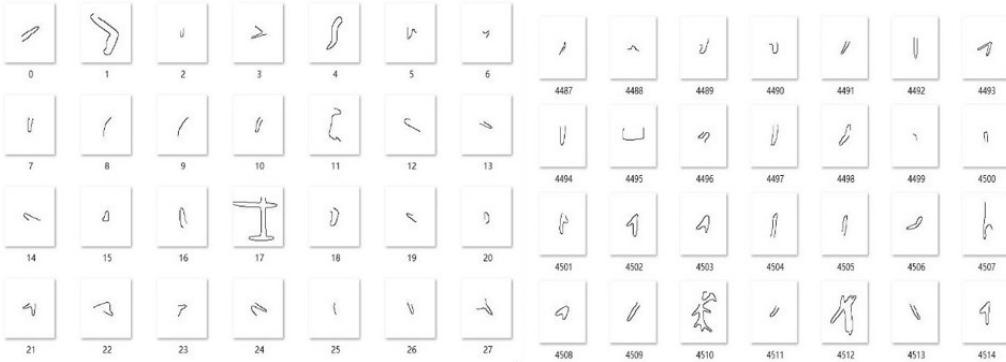


Figure 5.8: The silhouette of the stroke set.



Figure 5.9: The silhouette of the strokes which have inclusion architecture set

5.3 Cluster the strokes

After the strokes are stripped, it is not difficult to find that a lot of strokes are very similar and can be classified as one class. So, a clustering algorithm is needed to cluster similar Oracle strokes. There are two typical clustering algorithm which is the K-means algorithm and hierarchical clustering algorithm. The k-means algorithm is the first one to be examined. Base on the modern Chinese characters have 28 commonly used strokes, number 30 was selected to be the number of clusters. However, after clustering, the result did not reach the expected.

Although some similar images appear in the same set, many images that did not seem very similar were clustered to one set, and some image seems similar but appears in different sets. After this test, 10, 50, and 100 were selected to be the cluster sets number respectively to test which number is better for the total cluster categories. However, after testing, the result did not seem better than the 30-cluster categories edition. Later, the hierarchical clustering algorithm was picked to examined, the major difference between the hierarchical clustering algorithm and the K-means algorithm is hierarchical can automatically decide the total cluster categories, but the drawback of this algorithm does not perform well when the amount of data is huge. So only 1500 sampled image passed to that algorithm, after testing, the hierarchical algorithm only returns 6 categories. And the images in each category did not show a strong similarity between each other.

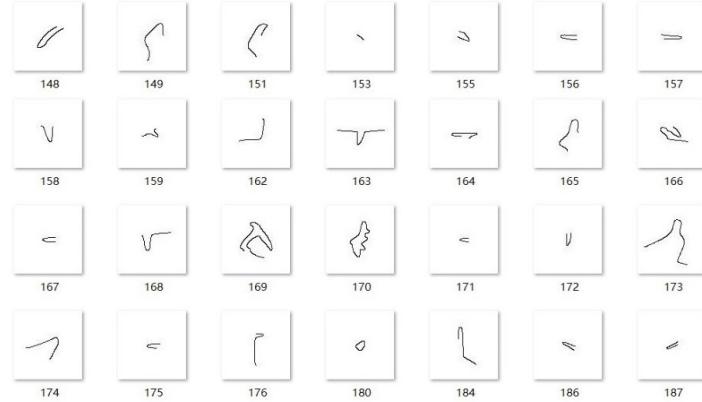


Figure 5.10: The silhouette of one cluster by k -means clustering.

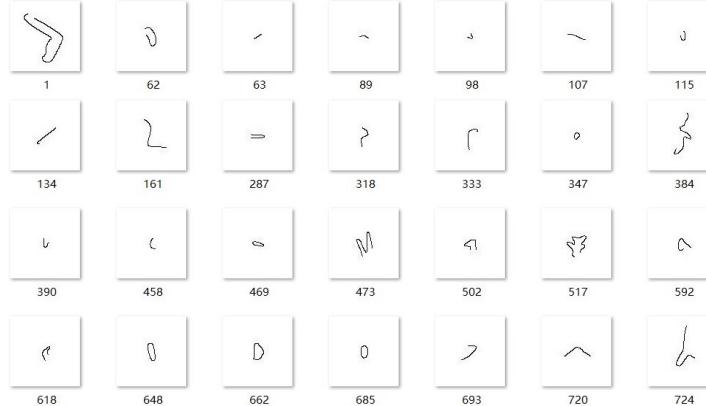


Figure 5.11: The silhouette of one cluster by hierarchical clustering

5.4 Chapter Summary

In general, the performance of decomposing Oracle bone inscriptions is relatively good, it separates the expected strokes out of the whole characters. By achieving this step, the expert no longer need to stripping the strokes by human beings. What's more, the separation of strokes which have inclusion architecture can auxiliary the analytical work in a more detailed way. However, the performance of the K-means clustering algorithm and the hierarchical clustering algorithm does not reach the expected. The potential issue might be too much data quantity or the gap between the image of the stroke is not strong enough for these two algorithms. In the future, more experiment should focus on the clustering step by try more potential clustering algorithm select the best performance one. **This chapter is contributed by Wei Du.**

Chapter 6

Conclusion and Future Work

This report elaborates set forth our progress of exploring several different types of ancient Chinese characters, results, and reflections. In the beginning, we majorly focus on the preparatory work and construct the complete ancient Chinese characters dataset classified by the type (typically the period) of characters to support our exploration. Later, we also put a lot of effort into the style conversion and generation of different types of ancient Chinese characters. The latest and the most advanced neural network technology was adopted by us for characters generation and style transformation like the GAN model and transfer learning model. Some models are successful and some of them do not perform as well as what we expected. There are many innovative ideas being proposed by us in a variety of ways to find potential correspondence between the ancient Chinese characters. What's more, we have done many innovation and experiments on existing neural network architectures. Although there are not very exciting results in this study, however, we have had many breakthroughs and attempts and these results can also be used in many other region's studies.

Some models are successful and some of them do not perform as well as what we expected. The Cycle-GAN is the most successful one, which can translate oracle bone inscriptions characters to the matched small seal style characters within 25.8% accuracy. (There is about 25% error of the accuracy caused by the classification.) That accuracy is much better than what we expect for this project. In that case, our project is successful.

In the future, we will keep improving the accuracy of the generation model and the accuracy of classification. We also want to explore what the limitations of our used generation models are, what potential of the models is still not excavated. If these generation models can accurately translate a character to the matched one or not. If their design theory contains some limitations lead to that it is hard to use them to do translation or shape conversion. If there are some ideas we did not consider can help us solve the problems. There are many questions for us to answer in future works.

We also take effort to utilize the current exist Oracle bone inscriptions rubbings to find alternative ways to have deep exploration. we have figured out algorithms to split the content of rubbings into complete sentences and identify the reading order of these sentences. This achievement is pioneering and outstanding. As oracle bone inscriptions characters in rubbings are interrelated rather than appear solely in rubbings, the algorithm can help learners access the content of oracle bone inscriptions more easily. In addition, we attempted to repair damaged rubbings with the help of neural networks. Although the final output of our model is not excellent, it proves the model which based on encoder and GAN has the ability to repair the damaged regions in rubbings.

Following our achievements, the future work should revolve around discerning the meaning of oracle bone inscriptions characters and trying to discern unknown characters according to the context. The parameters of our model need to be adjusted in order to optimize the effect of the

existing model.

We also try to find a new way to analyze the characters that have been experiment successfully on modern Chinese characters to Oracle bone inscriptions. Previously, the strokes of the Oracle bone inscriptions can only be decomposed by human beings, but after achieving this algorithm, the computer can automatically strip all the strokes in Oracle bone inscriptions. Although the experiment on clustering the strokes does not reaching the expectation, however, the direction of that thought is very meaningful to finally labeled the Oracle bone inscriptions based on strokes number, and to let the computer understand the correlation between the shape correlation between different characters.

In the future, more work should be put on further expansion of the dataset to feed the training more properly, the potential method might combine different data sources and merge them together and unify the data. Then in the Generation and Evolution part, more work should focus on further optimize the network structure. What's more, after the rubbings dataset which described in Chapter 4 are constructed, it is possible to have a model, not only based on the shape of the character but based on the context relationship of the character. The character decomposition to strokes in the future can also auxiliary this model by labeled the Oracle bone inscriptions by the numbering of strokes.

Reference

- [1] M. Jiang, B. Deng, P. Liao, B. Zhang, J. Yan, and H. Ding, “Oracle bone inscriptions database and building of intelligent knowledge base,” 2004.
- [2] S. Xu, *Shuo Wen Jie Zi*, 121.
- [3] B. Baike, “Oracle bone inscriptions baidu baike,” 2019. [Online]. Available: [https://baike.baidu.com/item/Oracle/16914reference-\[1\]-6337786-wrap](https://baike.baidu.com/item/Oracle/16914reference-[1]-6337786-wrap)
- [4] R. Sears, “Chinese etymology,” 2017. [Online]. Available: <https://hanziyuan.net>
- [5] *Selection of OracleBone Inscriptions Rubbing 1*. Henan art publisher, June 2018.
- [6] *Selection of OracleBone Inscriptions Rubbing 1*. Henan art publisher, June 2018.
- [7] *Classic OracleBone Inscriptions Rubbing 100 instances*. Shanghai calligraphy and paintings publisher, 2015/8.
- [8] Wikipedia, “Convolution neural network,” 2013.
- [9] W. T. N. Hubel D H, “Receptive fields and functional architecture of monkey striate cortex,” 1968.
- [10] F. K., “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” 1980.
- [11] UDFDL, “Convolutional neural network,” 2014. [Online]. Available: <http://ufdl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>
- [12] Wikipedia, “U-net,” 2015. [Online]. Available: <https://en.wikipedia.org/wiki/U-Net>
- [13] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.
- [14] A. B. L. Larsen, S. K. Sønderby, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” *CoRR*, vol. abs/1512.09300, 2015. [Online]. Available: <http://arxiv.org/abs/1512.09300>
- [15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [16] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [17] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, “Self-taught learning: transfer learning from unlabeled data,” in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 759–766.
- [18] J. ALTOSAAR, “Tuorial-what is a variational autoencoder?” [Online]. Available: <https://jaan.io/what-is-variational-autoencoder-vae-tutorial/>

- [19] J. P. Cunningham and Z. Ghahramani, “Linear dimensionality reduction: Survey, insights, and generalizations,” *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2859–2900, 2015.
- [20] P. Y. Simard, D. Steinkraus, J. C. Platt *et al.*, “Best practices for convolutional neural networks applied to visual document analysis.” in *Icdar*, vol. 3, no. 2003, 2003.
- [21] Y. H. Tay, P.-M. Lallican, M. Khalid, S. Knerr, and C. Viard-Gaudin, “An analytical handwritten word recognition system with word-level discriminant training,” in *Proceedings of Sixth International Conference on Document Analysis and Recognition*. IEEE, 2001, pp. 726–730.
- [22] P. Ballester and R. M. Araujo, “On the performance of googlenet and alexnet applied to sketches,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [23] Google, “Announcing open images v5 and the iccv 2019 open images challenge,” 2019. [Online]. Available: <https://ai.googleblog.com/2019/05/announcing-open-images-v5-and-iccv-2019.html>
- [24] D. S. K. G. R. B. . F. A. Redmon, J., “You only look once: Unified, real-time object detection,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 779788, 2016.
- [25] A. F. Joseph Redmon, “Yolov3: An incremental improvement,” 2018.
- [26] Baidu, “Image processing baidu baike,” 2019. [Online]. Available: [https://baike.baidu.com/item/Digital Image Process/294902?fr=aladdin](https://baike.baidu.com/item/Digital%20Image%20Process/294902?fr=aladdin)
- [27] R. E. W. Rafael C. Gonzalez, *Digital Image Processing, Third Edition*, 2011.
- [28] e. a. Pathak, Deepak, “Context encoders: Feature learning by inpainting.” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [29] Baidu, “common use character in chinese characters,” 2013. [Online]. Available: <https://baike.baidu.com/item/strokes/3040863?fr=aladdin>
- [30] D. Honglin, “Let the machine knows the strokes in chinese characters,” 2018. [Online]. Available: <http://www.neurta.com/index.php/node/417>
- [31] J. F. Canny, “A computational approach to edge detection,” 1987.
- [32] C. J. Harris and S. M, “A combined corner and edge detector,” 1988.
- [33] Wikipedia, “K-means clustering,” 2006. [Online]. Available: <https://en.wikipedia.org/wiki/K-means-clustering>
- [34] WikiPedia, “Hierarchical clustering,” 2010. [Online]. Available: <https://en.wikipedia.org/wiki/Hierarchical-clustering>
- [35] L. b. T. w. Z. Z. W. h. L. s. Cheng Li, Wang Jiangqing, “Algorithm on strokes separation for chinese characters base on edge,” 2013.