参赛队员姓名：陈筠富、张瀚之

中学：广东华南师范大学附属中学

省份：广东省

国家/地区：中国

指导教师姓名：张玉金教授、杨晓安老师

指导教师单位：广东省中国语言学会（会长）、广东华南师范大学附属中学

论文题目：Decoding the Past: Solving Challenging Oracle Bone Characters Recognition Problem by Integrating Vision Transformer and Generative Adversarial Image Restoration Techniques

# Decoding the Past: Solving Challenging Oracle Bone Characters Recognition Problem by Integrating Vision Transformer and Generative Adversarial Image Restoration Techniques

**Junfu Chen, Hanzhi Zhang**
Affiliated High School of South China Normal University
Email: {chenjf.victor, zhanghz.frank}2023@gdhfi.com

**Abstract**

Oracle Bone Characters (OBC), one of the earliest forms of writing in China, are crucial for understanding ancient religious practices and daily lives. Inscribed on animal bones and shells, these artifacts, known as Oracle Bone Scripts (OBS)[1], offer pivotal historical and cultural insights. Yet, the identification of OBC poses substantial challenges due to factors such as **Cracks**, **Weathering**, **Similarities** and the **Sheer Volume of Characters**, mainly caused by the long-timed burying, which resulted in serious damage of OBS. Our goal is to help archaeologists improve the efficiency of Oracle exploration.

Firstly, we comprehended on-site about the challenges and difficulties that OBC researcher encountered during archaeological excavation: researchers initially need to extract single OBC from the OBS, then scanning every section and further transcrbing these scanned OBC into Hand-Printed version, finally identifying whether the characters are deciphered or un-deciphered before matching them to modern Chinese characters. There are four obstacles lying in this process: (1) The present OBC recognition models are all based on the clear Hand-Printed version data; however, for the reasons like weathering and defacement, **not all the scanned OBC can be transcribe into Hand-Printed format** for the model; (2) **Ligature**, due to ancient writing format, under certain situation, multiple OBC are written together like one, which cannot be distinguished by the current models; (3) **overwhelming amount of categories** and **Long-tailed problem** Some categories have only a few samples; (4)**open-set recognition** Some un-deciphered or unpresented OBC. In order to deal with these problems, we built our own pipeline, including following three parts:

- Specific to **Ligature** situation, an **automatic OBS detection system (OBS-DET)** was developed, representing the ligature challenge we faced, helping archaeologist extracting every OBC. This model is trained on Oracle Bone Inscription Detection Dataset downloaded from YinQi-WenYuan (殷契文渊), and run on RTX3090Ti, achieving the AP of 47.3% with the FPS of 53.4. It can accurately and rapidly detect and extract OBC, dealing with **ligature and massive OBC processing task**.
- We built a set of **generative adversarial based image restoration system (OBC-Trans-GAN)**, transcribing scanned version OBC into clearer Hand-Printed version. Traditional computer vision method like **Edge Sharpening Filter** can only handle simple problem, but the more incomplete OBC cannot be correctly transcribed. OBC-Trans-GAN trained on Oracle-241 dataset can generalize to complex oracle recovery scenarios fine, further simplify the OBC recognition task.
- To solve **Long-Tailed Problem** and **Open-Set Recognition Problem**, we designed a **transformer based OBC recognition model (OBC-VIT)** which is the essential part of the whole pipeline, OBC-VIT is trained on the HUST-OBS dataset, can achieve the accuracy of **97.2%,**

[1]The abbreviations used for Oracle Bone Characters are OBC, denoting individual characters, and OBS, representing the animal bones or shells inscribed with OBC.

**which is the highest among the existing model**. Through designing a specialized data augmentation method aim at pictograph, while combining generative adversarial network to generate sample of tail categories, the model can perfectly solve the Long-Tailed problem. Meanwhile, applying **Open-Set Recognition (OSR)** based on measuring similarity can achieve the goal of recognizing un-deciphered OBC.

This project is made open-source on: https://github.com/zhz5687/Transformer-OBS-Recognition. We also develop an APP using the above AI models to assist archaeologists which was demonstrated and used in the Oracle Bone Research Center of South China Normal University.

***Keywords:*** *Oracle Bones Script Detection, Oracle Bones Character Recognition, Open-Set Recognition, Long-Tailed Problem, Large Language Models, Generative Adversarial Networks, Vision Transformer*

# 解码历史：结合 Vision Transformer 识别与生成对抗式图像复原技术解决复杂场景的甲骨文识别问题

陈筠富，张瀚之
广东华南师范大学附属中学
{chenjf.victor, zhanghz.frank}2023@gdhfi.com

**摘要**

甲骨文（OBC），中国最古老的书写形式之一，对于了解中国古代宗教习俗和日常生活至关重要。刻有甲骨文的动物骨头和龟壳被称为甲骨（OBS）。然而，由于这些文字埋藏时间过长从而导致**裂缝、风化、相似性和文字数量**庞大等因素，导致甲骨出土物损坏严重，甲骨文识别的鉴定面临巨大挑战，我们希望解决这类复杂的甲骨文识别场景。

首先我们实地了解了考古发掘中研究人员遇到的难点与问题，整体流程如下：研究人员首先需要从甲骨中提取每个甲骨文，然后对每个切片扫描，并进一步转化为字体清晰的手写版（Hand-Printed）版本甲骨文，最后判断其是否属于已被破译文字，并将已破译的文本翻译成现代汉语。这里面就有四个难点：（1）目前的甲骨文识别模型都是基于清晰的 Hand-Printed 版，但是实际中由于风化、损毁等影响，**并不是所有的扫描版甲骨文都可以转化为 Hand-Printed 格式**给模型识别；（2）**合文场景**，由于古代书写格式，两个甲骨文紧凑写在一起，但实际意义是两个字，目前的识别算法都无法解决这类问题；（3）**甲骨文识别种类多**及**长尾问题**：很多文字出现次数少；（4）**开集识别问题**：归类为未被破译的甲骨文（Un-deciphered）。为了解决这些问题，我们搭建了一套自己的流程，主要包括三个部分：

- 针对**合文**场景开发了一个**自动化 OBS 检测系统**（**OBS-DET**），帮助考古学家从甲骨中提取每个甲骨文。该模型使用殷契文渊甲骨文检测数据集进行训练，在 RTX3090Ti 上**以 53.4 FPS 的速度实现了 47.3%的 AP**，可以快速准确提取甲骨文，**解决合文和大批量甲骨文检测问题。**

- 搭建了一套**基于生成对抗式甲骨文图像复原技术**（**OBC-Trans-GAN**），将扫描的甲骨文转换为更清晰的手写版本 Hand-Printed。传统计算机视觉方法如**边缘锐化滤波器**只能解决简单的场景，但是针对损毁程度较大的 OBC 基本不能准确的转化，基于 Oracle-241 训练的 OBC-Trans-GAN 可以很好的泛化复杂甲骨文复原场景，把甲骨识别问题进一步简化。

- 设计了一种可以**解决极端长尾问题**（**Long-Tailed**）**与具备开集识别**（**Open-Set Recognition**）**能力的甲骨文识别模型**（**OBC-VIT**），识别甲骨文是整个任务的关键元素，该模型在 HUST-OBS 数据集上进行训练，**可以达到目前所有方法中最高的准确率 97.2%**。通过设计一种专门针**对象形文字的数据增强方法，同时结合生成对抗技术扩展少样本数据来**解决**长尾问题**，最后基于度量相似性的方式进行**开集识别 OSR**，从而可以识别 Un-deciphered 类型的甲骨文。

该项目代码发布在 https://github.com/zhz5687/Transformer-OBS-Recognition。我们还利用上述模型开发了一个辅助考古学家的 APP，并在华南师范大学甲骨文研究中心进行了演示和使用。

**关键词：检测及提取甲骨文，甲骨文识别，开集识别，长尾分布问题，大语言模型，Vision Transformer，Generative Adversarial Networks**。

# Contents

# 1 Introduction

## 1.1 Background

Oracle Bones Scripts (OBS), mainly popular in the Shang and Zhou dynasties, are the earliest mature writing system in China, shown in Fig.1. They are mainly carved on animals' bones or turtle shells. The Shang royals, while doing both state affairs and private lives, including sacrifice, climate, harvest, and expedition, all divined to the gods first in order to know whether good luck or bad fortune was on their sides, so they could decide how they do or whether they should do these things. The divine always needs to use animals' bones or shells, which greatly promotes the development of OBC. The OBS were firstly discovered by Wang Yirong in 1899[1], till today, people have discovered about 150000 OBS and 4500 OBC, including 2000 deciphered ones and 2500 un-deciphered ones. Studying OBC can help us know better about Chinese history since they are the origins of Chinese characters and the roots of Chinese excellent traditional culture.

When OBS are dug out, there are lots of different characters on shells or bones, as shown in Fig.2, the workflow for archaeologists excavating OBS as follows. Firstly, find turtle shells or bones inscribed with OBS and clean the dust and dirt. Secondly, separate each character from the bones or shells. Then, transform scanned image to Hand-Printed image, better recognizing each distinct OBC. Finally, identify whether there are characters that have been deciphered or un-deciphered, if deciphered, match them to known modern Chinese characters. These steps are currently performed manually by archaeologists using their specialized knowledge, resulting in relatively low efficiency.

***Figure 1:*** *An Example of Oracle Bone Character Script*

However, studying OBC faces lots of problems. Among them, the most significant problem is the lack of expertise to identify the meaning of OBC, since a few people study OBC today, and among them, only a few outstanding experts can identify the meaning of lots of these characters and further explore the meaning of the un-deciphered character. Therefore, nowadays, with the development of AI, people have begun using the computer, mainly using Convolutional Neural Network (CNN), to help them identify the meaning of the character. To achieve this purpose, people need to solve problems such as weathering and abrasion caused by long time burying, the lack of well sorted datasets, the scratching which may mislead us, the variability of word forms and so on. We try several public models and find out their accuracy are not really high, and they are also not really practical for the researchers to identify them. Therefore, we build our own pipeline, which is very practical that it can identify all the characters on OBS after researchers give it pictures of the OBS they just dig out, while its identified rate is also really high. Our pipeline mainly includes three parts: **OBS Detection**, **Scanned to Hand-Printed Transcript**, **OBC Recognition**. This project aims to provide an efficient solution by leveraging AI technology to address challenges encountered in real archaeological research. Additionally, we are exploring the capabilities of Large Language Models (LLMs) like ChatGPT-4o and Gemini 1.5 to test their ability to recognize Oracle Bone Characters.

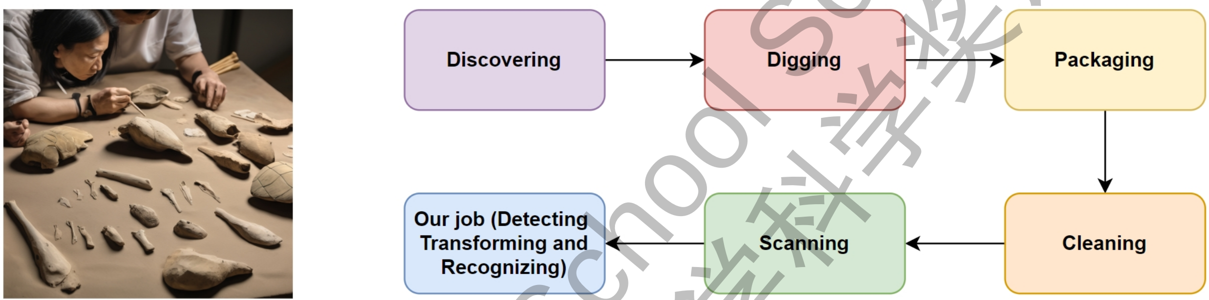



***Figure 2:*** *Steps for OBC digging*

Overall, our main contributions can be summarized as follows:

- A system named **automatic OBS detection (OBS-DET)** was built based on the **ligature** to assist archaeologists in extracting individual OBC from OBS. The model was trained on the Oracle Bone Inscription Detection Dataset downloaded from YinQiWenYuan (殷契文渊). OBS-DET achieves an impressive 47.3% AP at a speed of 53.4 FPS on an RTX 3090Ti, which means it can extract the OBC accurately and rapidly and address ligature and lots of OBC detection problems.
- A **generative adversarial based image restoration system (OBC-Trans-GAN)** was built to convert scanned OBC images into clearer, Hand-Printed versions. Traditional methods like **edge sharpening filter** are only helpful in easy and basic circumstances, but for OBC with large wear they can't transcribe effectively. Therefore, the OBC-Trans-GAN was built. It can simplify the identification problem by generalizing the complex OBC restoration scenario
- To recognize the OBC, a OBC recognizing model, **which can tackle Long-Tailed problem and do Open-Set Recognition**, was built. This model was trained on datasets of HUST-OBS and can achieve the accuracy of 97.2%, highest in this domain. It can solve Long-Tailed problem by **designing a data enhancement method for pictographs, while enlarging the number of data by utilizing generative countermeasure technique in the same time**. Since there are data of un-deciphered OBC in the datasets, it can also do the OSR.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 outlines the problems we address and introduces our pipeline. Section 4 details the methods and processes of our model. Section 5 presents the experimental results of our model. Finally, section 6 concludes the paper.

# 2 Related Work

In this section, we summarize the existing papers of using machine learning and deep learning methods to identify OBC. Most of the papers focus on eliminating the influence of noise such as cracks or erosion caused by weathering, in addition, we also compare the current the most well-known OBC/OBS-related datasets, such as datasets HWOBC [2], OBC306 [3], Oracle-MNIST [4] and HUST-OBS [5].

## 2.1 Methods of OBS Detection and Segmentation

Recognizing Oracle Bone Characters (OBC) presents several challenges, primarily due to the noise introduced by prolonged burial, weathering, and cracking, shown in Fig.3, as well as the inherent difficulty in extracting individual characters. The latter difficulty is exacerbated by the absence of a punctuation system and the presence of meaningless scratches. X. Fu et al[6] use the non-local means (NLM) approach to solve the problem. This technique involves filling in the value of a pixel with the average value of all similar pixel regions, thereby eliminating Gaussian noise, which frequently occurs at the edges of characters. Another innovative method, time serialization model, was employed by C. Zhang et al.[7]. This model transforms fragmented borderlines into "time series" and compares the differences to identify similarities between fragmented and normal borderlines. This process helps rejoin the OBC and reduce the impact of cracks and scratches. For segmentation, Y. Fujikawa et al.[8] have developed a two-step model to manage this task. Initially, a YOLOv3 model is applied to the uploaded image to generate bounding boxes around different characters. However, this method may leave some characters undetected. To address this, MobileNet[8] is utilized to detect characters in the remaining areas. Despite its effectiveness, this approach is not fully automated and requires human intervention to select undetected areas when using MobileNet. Therefore, transforming the complete scanned text into separated characters remains a significant challenge in this field.

**Current limitation:** One problem that all the current models didn't take into account is the problem of ligature, which is rare but important for OBC research. Their detection model mainly just detect the strokes that grouped together, which resulted in detecting ligature as one character. Some models are too old, for example, YOLOV1, so we decided to try several more recent method with higher accuracy. Unlike recognition, there's only one dataset for the OBC detection task, which we think is not sufficient.

***Figure 3:*** *Example of Weathering and Cracking*

## 2.2 Methods of OBC Pre-Processing and Recognition

Identifying ancient characters, such as Oracle Bone Characters (OBC), presents several significant challenges, including a lack of data, class imbalance, glyph diversification, and open set recognition problems. To address these issues, X. Liu et al.[9] had introduced a Siamese similarity network based on a similarity learning method, which directly learns input similarity and applies the trained model for one-shot classification tasks. Additionally, X. Han[9] proposed a cumulative class prototype to tackle deviations from the average class prototype and establish a robust class representation. This approach enhances the model's ability to distinguish between known and novel categories, crucial given the continuous discovery of new ancient characters. Furthermore, J. Liu et al.[10] have developed a program to identify whether a character is an OBC, employing semi-supervised self-training learning to understand connections between oracle bones and other ancient Chinese characters. Utilizing a loss function and a novel database called CL-Oracle-100, this program has achieved an accuracy rate of 87.9% in identifying Oracle script fonts. These advancements collectively represent significant progress in the recognition and classification of ancient characters, although ongoing research and development are necessary to further refine these methods.

Recently, there has been a growing interest in utilizing deep Convolutional Networks to identify Oracle Bone Characters (OBC). However, the training of deep models typically requires a substantial quantity of labeled samples, posing a significant challenge. To address this issue, M. Wang et al.[11] have developed an Unsupervised Structure-Texture Separation Network (STSN). STSN separates characteristics into structure (Glyph) and texture (noise) elements using generative models, then aligns handprinted and scanned data in the structure feature space to mitigate the negative impacts of significant noise during adaptation. Moreover, the process involves exchanging the acquired textures between domains and training a classifier for final classification to predict labels of the transformed scanned characters. This approach ensures complete differentiation and enhances the discriminative capacity of the acquired features. In addition, there remains a need for an efficient and reliable system for symbolizing and serializing Oracle Bone Script (OBS) data. Currently, most OBS data is stored as unserialized images of bones and shells, which is not suitable for character recognition or text comprehension. To address this gap, X. Han et al.[12] have developed an information system known as IsOBS to represent and organize OBS data on a character-by-character basis, facilitating the potential application of machine learning techniques for OBS analysis in the future. They have established a database specifically for OBS characters, linking each character to its modern Chinese counterpart (if deciphered) along with multiple variations of its glyph. Additionally, he have introduced a character recognition component for OBS characters using few-shot learning models, given the limited number of examples available for each OBC. Leveraging this character recognition component, they have created an information retrieval system [11] for searching within character and document databases.

**Current limitation:** Many of the models didn't consider the Long-tailed problem, which is common due to the various categories in OBC and the imbalance of sample number between different classes. None of the articles mentioned the problem of open set recognition, which we think might be a big problem, since not all the input are OBC, many of them are noise or wrongly detected OBC segments. Therefore, it's a brand new field for us to try researching.

For now, effective computer-assisted methods for predicting meanings of un-deciphered OBC have been lacking, necessitating manual input for each OBC category's significance. Nonetheless, several datasets are available for categorizing OBC, encompassing various characters and their respective meanings, like IsOBS[11], HWOBC[2], OBC306 [3]. These datasets can be directly employed to evaluate and validate our model.

## 2.3 OBC Datasets

Scholars build various kinds of datasets with own features for different tasks, shown in Table 1. Traditional datasets contain scanned version OBC and handwritten version as label, aiming to train models solving classification tasks. These datasets do not include the modern meaning of each categories of OBC, therefore can't be used in interpretation tasks. Oracle-MNIST [4] is a 10-class classification datasets, but the small scale limits it to some simple tasks. This datasets do not suit the real world situation, which usually requires classifying OBC into hundreds of categories. Another similar datasets

***Table 1:*** *Summary of Different OBC Dataset Characteristics*

| Datasets | Images | Categories | Tail Classes | Noise | Whole Texts | meaning | un-deciphered |
|---|---|---|---|---|---|---|---|
| Oracle-241[11] | 80k | 241 | | ✓ | | | |
| Oracle-MNIST[4] | 30,222 | 10 | | | | | |
| OBC306[3] | 309,551 | 306 | ✓ | ✓ | | | |
| HWOBC[2] | 83,245 | 3,881 | | ✓ | | ✓ | ✓ |
| HUST-OBS[5] | 62,989 | 9,411 | | ✓ | | ✓ | ✓ |
| OBI-100[6] | 4,748 | 100 | ✓ | ✓ | | | |
| OBI-125[13] | 4,725 | 25 | | ✓ | ✓ | | |
| Oracle-20k [13] | 20,039 | 261 | ✓ | ✓ | | | |
| Oracle-AYNU[14] | 40,000 | 2583 | | | | | |

is Oracle-241 [10], containing 80,000 images belonging to 241 categories. The large scale makes it suitable for most classification model training. Besides classification, interpretation is also a widely discussed task. HWOBC [2] is a datasets not only includes the OBC, but also contains the meaning of each categories. This enables it to be used for training translating model by matching meaning to the categories. However, not all the OBC have clear meaning, some might be not deciphered, so scholars build another datasets called HUST-OBS [5]. It labels 9,411 kinds of characters out of the total 10,999 as un-deciphered characters. This also shows that most of the OBC have still not been deciphered, so predicting un-deciphered OBC's meaning is a field worth studying. Some tasks also require the whole OBS instead of OBC to train segmentation model, so here comes the OBI-125 [13], a special rubbing type datasets. This datasets is make up of scanned version of originally unearthed OBS. Just like other languages, not all the OBC are equally used, some frequently used characters occurs more in the datasets, while some categories only have few images, this might lead to the inaccuracy of the recognition of some categories, which is known as long-tailed problem. A datasets called Oracle-20k [13] takes this into account, it design some categories that include less than 50 images, while some include hundreds. This kind of datasets can test the models' ability of handling long-tailed problem, but some datasets themselves use data augmentation to create more image for the "tail" categories in order to solve this problem.

**Long-Tailed problem**, which will affect the accuracy of the recognition for some specific categories, occurs due to the uneven distribution of the data in all the categories. Some scholars use augmentation to either solve Long-tailed problem or increase the scale of the datasets. First, some simple transforming methods like rotation, stretching, dilation, and compression. These method [6] might be simple, but can be combined with other methods to further deform images. GAN model, Stable Diffusion model, and ControlNet model [14] are also effective ways of data augmentation. However, these methods have some disadvantages like the lack of controllability and the absence of the content or meaning either. Another method specifically effective to tackling Long-tailed problem is mix up generation [15], mixing the tail samples with other samples to create data that increases the weight of tail categories. This model needs to be followed by a GAN system to ensure that the data isn't out of real distribution.

# 3 Challenges and Significance

## 3.1 Problems

The study on OBC is always an important field since it can introduce us to lots of interesting things about the ancient China. However, it's now facing the problem of manpower shortage. There are a lack of experts in this domain. Therefore, people try to use AI to build models, which can identify OBC automatically, to solve the problems. Nevertheless, there are also several serious problems in this solution, mainly including four problems:

- Because the characters are being burried for too many years, some of them suffer from **serious damage**, causing lots of cracks and abrasions. This lead to the result that the traditional method can't transcribe all scanned version OBC into Hand-Printed version. After transcribing them, there are still lots of noise and the images are still obscure, **causing the low accuracy in recognition part**.
- **In the ancient time, people sometimes wrote two characters like one, which is known as ligature**. For lots of existing models, they can't solve this problem, since most of them do not have a detection part. Even though they have, it's also hard to solve this problem because we can't make sure whether it's one character or two, which needs to be decided based on the meaning, or sometimes if we regard these two characters as one, it's an unknown character, which means it probably does not exist. We can understand it better through the examples below.
- The **Long-Tailed problem** is quite serious. There are some kinds of OBC that just have a few data, causing the accuracy of identifying these kinds of OBC is lower than others. The tradition method of solving long-tailed problem is to generate more samples of these kinds of OBC manually by rotating it, mirroring, changing the font size and so on. However, this tradition method does not work well, since for the computer, the features do not change, and they just need to exchange the weight between different parts.
- **Open-Set Recognition** is also a troublesome problem. There are some characters that are not **deciphered** yet. However, for the model, they always regard these characters as the deciphered ones, and output a wrong answer. Besides, these characters are really scarce, so there are also a lack of samples of them.

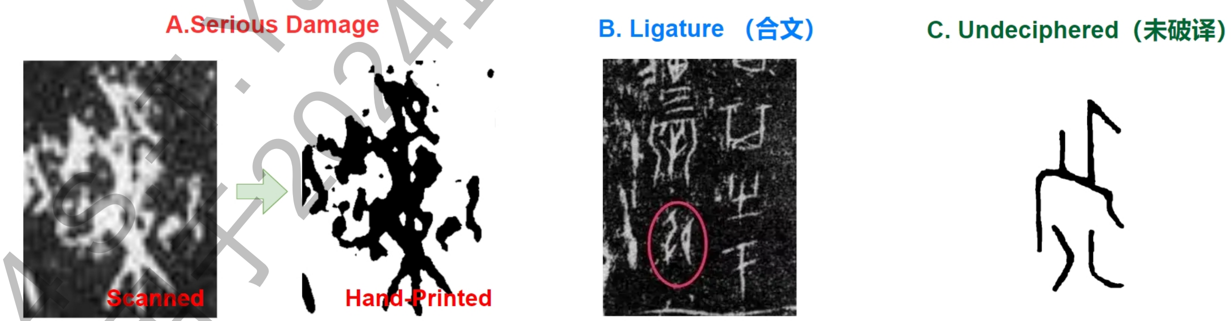



*Figure 4: The illustration consists of three parts. Part A illustrates the inefficiency of traditional method of trancipt. Part B gives an example of legiture, while part C is an un-deciphered OBC, illustrating open-set recognition. Its code in 新甲骨文编 is 1053 in appendix.*

These problems and challenges within the field of OBC research makes it necessary but also challenging to build up model that can effectively and accurately help recognizing OBC. To make the pipeline more practical, we build the detection part, transcript part, and recognition part, which can help identifying the OBC on an animal' shell or bone, so researchers can immediately know what the characters on the bones and shells after they unearth them and upload the images to our model, which can greatly improve the practicability of our model.

### 3.2 Explore the Capabilities of the LLM

Widely utilized by individuals for educational purposes, task completion, programming, and various other applications, Large language model (LLM) has gained significant intentions today. It offers substantial time-saving benefits and excels in accurately and swiftly addressing various challenges. This prompts an intriguing inquiry into whether they can effectively recognize OBC and whether they can use existing images to produce new ones aiding in tackling the low accuracy of identifying OBC with little images. To explore the answer, we select the OBC version of Chinese character 贝, denoting "shell", as a case study. We try it in both LLMs and our program, which answer is right. In contrast, for two of the most famous LLMs of google, Chatgpt4o and Gemini-1.5-Flash, the results were not satisfactory, as shown in the picture. Moreover, we proceed to task the LLMs with generating the OBC version of the Chinese character 贝. As shown in Fig. 5, the result is unpleasant either. Overall, for these LLMs, the challenge of accurately identifying OBC or producing the Oracle Bone Character version of a Chinese character remains formidable. This underscores the pivotal role of our model in effectively deciphering Oracle Bone Scripts.

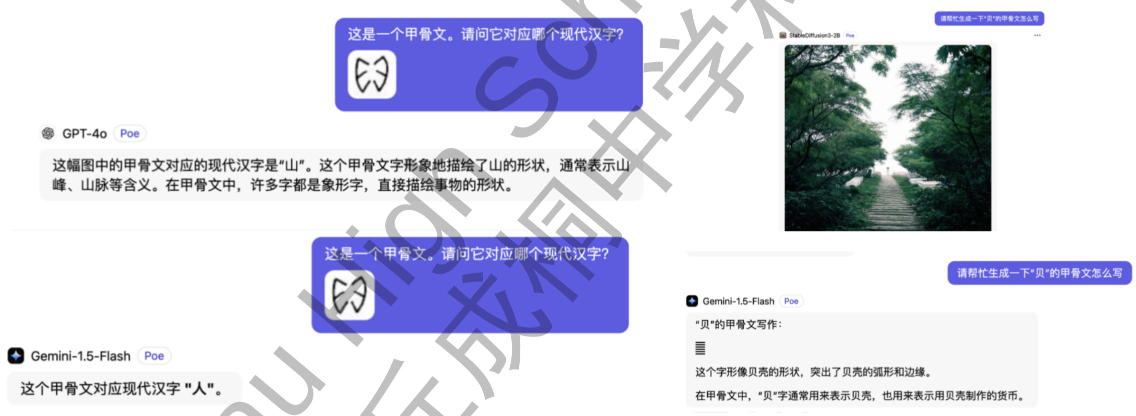



*Figure 5:* *Using LLM to recognize and generate OBC (LLM does not work well)*

## 4 Methods and Innovative Process Flow

As shown in Fig. 6, we propose an innovative AI based pipeline to solve archaeological problems. The next part will introduce the principles of each module.

- Step1: To deal with **Ligature** and **massive OBC processing task problem**, we first build **OBS-DET** system to excel in segmenting and identifying individual OBC from OBS, generating a multitude of images for analysis. Archaeologists, upon unearthing OBS, face the initial task of isolating each Oracle Bones Character. We have employed the YOLO-X based detection model and train it on the YinQiWenYuan (殷契文渊) website to achieve this goal.
- Step2: Due to the reason of **weathering and defacement**, there are lots of noise and the images are still obscure, causing the low accuracy in recognition part. We develop **OBC-Trans-GAN** to

restore the image through precise transformation and generalization. After this operation, we can obtain clear OBC images for next stage.

- Step3: We implement the transformer based recognition model **OBC-VIT**. We compared different backbone like VGG, Resenet50 and VIT while training the model using the HUST-OBS dataset. In the end, we achieved an accuracy of about **97.2%**. we Integrate 4 complementary technologies: Through designing a specialized data augmentation method aim at pictograph, while combining generative adversarial network to generate sample of tail categories, then use Seesaw Loss and two stage training method to solve the **Long-tailed problem**.
- Step4: Considering **OBC Open-Set Recognition problem**, we added a layer adjusted from **Openmax** to create the open area for the un-deciphered OBC. This layer was **linked to multiple featuring layer, and each layer has multiple head with different attention from Transformer backbone that's responsible to different features of the object**. The layer boost the model's accuracy of distinguishing un-deciphered OBC, now reaching **85.43%**.

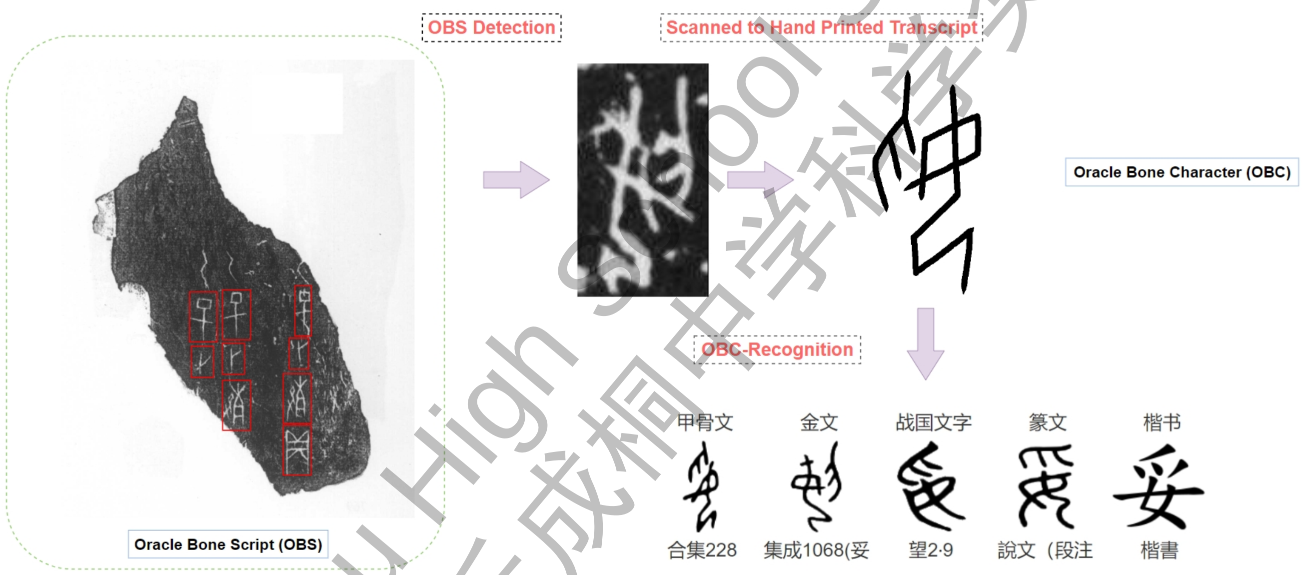



***Figure 6:*** *Innovative AI Models for Oracle Bone Character Identification Pipeline*

## 4.1 OBS Detection: Handling Ligature Problem and Batch Extraction of OBC

In some rare situations that multiple characters mixed together, known as Ligature (合文), people sometimes wrote two characters like one. We tried lots of existing recognition models, they can't solve this problem, as shown in Fig.7. We cleverly combined the object detection algorithm to train a highly accurate OBS Detection System (OBS-Det), which can easily separate the above-mentioned combined scenes and then identify them separately.

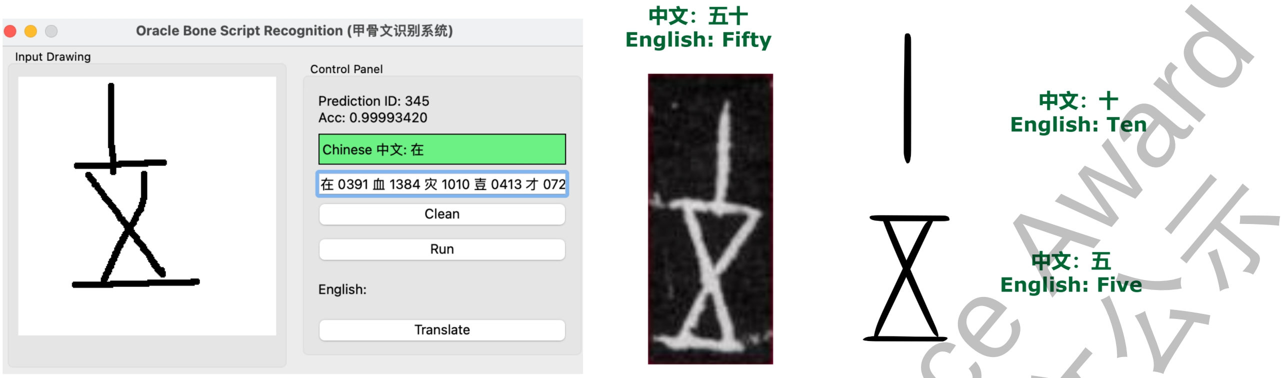



***Figure 7:*** *Challenge of ligature: the OBC 五 (Five) and OBC 十 (Ten) are sometimes written together to form the ligature 五十 (Fifty). However, the recognition model can;t figure out that this is a ligature formed by two characters, the recognition model think it is one character and will predict the wrong result.*

In addition to solving the problem of Ligature, our OBS-Det has another application, for most of the existing models that identify OBC, they are not such practical that researchers can't use them to identify OBC on OBS once they dig out some OBS. With this aim, we build the detection part. Detection part can help us segment the OBS and produce some images that each image only includes one OBC, as shown in Fig. 8. In our detection part, we firstly locate the bounding box of each OBC, and cut the character to produce a scanned version of this character. The format of the bounding box is [X1, Y1, X2, Y2], which means the coordinates of the top left and bottom right. Our detection part use YOLO-X framework.

YOLO-X [16] represents a significant advancement in the YOLO (You Only Look Once) series of object detection models. Since the introduction of the inaugural YOLO model in 2015, which rapidly gained prominence for its object detection capabilities, the series has undergone several iterations. Unlike its predecessors YOLOv3[17], YOLOv4[18], and YOLOv5[19], which relied on the Anchor-Based method for target box extraction, YOLO-X innovatively employs an Anchor-Free approach, offering distinct advantages. Additionally, the Anchor-Free methodology in YOLO-X eliminates the need for IoU (Intersection over Union) calculations, thereby significantly reducing computational overhead and minimizing the number of predictive frames. Moreover, this approach effectively addresses the challenge of positive and negative sample imbalances, where positive samples contain object instances and negative samples do not—while simultaneously avoiding the complexities associated with anchor tuning.

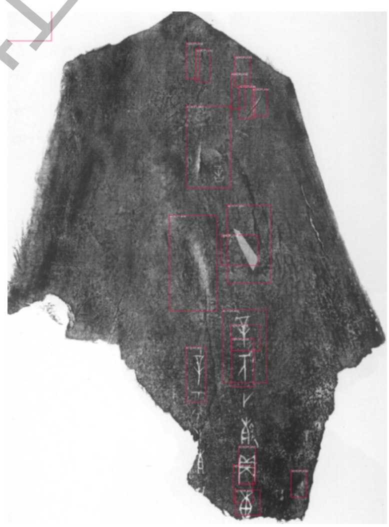


***Figure 8:*** *Bounding Box and the OBC Detected*

The architecture of YOLO-X comprises several key components. The backbone, a pre-trained CNN trained on extensive image datasets, is responsible for recognizing low-level features and patterns. YOLO-X employs Darknet53 for this feature extraction process. The neck module, utilizing **Path Aggregation Feature Pyramid Network (PAFPN)**[20], combines feature maps extracted from the backbone network to enhance detection performance and enable the model to learn at larger scales. The final important component, the head, is tasked with generating predictions based on the features provided by the backbone and neck. YOLO-X implements a **Decoupled Head**, **Anchor-Free mechanism**, and **Multi-positive** approach to accomplish this task effectively. As a concluding step, a post-processing technique, typically **Non-Maximum Suppression(NMS)** [21], is applied to filter out overlapping predictions and retain only the most reliable detections, as shown in Fig. 9.

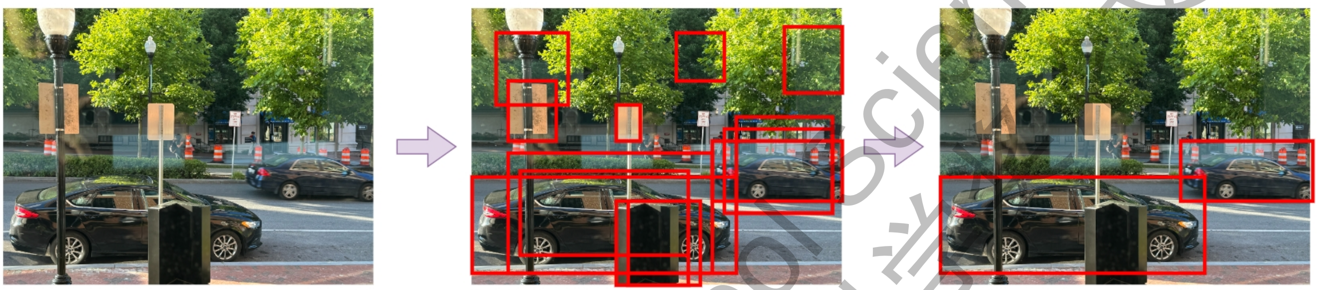


***Figure 9:*** *How YOLO-X Works: Anchor-Free, Decoupled Head, Multi-Positive*

The noise model will return a noise for the adjacment of parameter, the loss function expresses as:

$$L = \frac{1}{n}(w * L_r + L_o) \tag{1}$$

where $L_r$ representing regression loss, $L_o$ representing objective loss, $w$ serving as a weight, and $\frac{1}{n}$ for average loss. The method for calculating object loss is using Sum of the Square Error, and Cross Entropy Loss for regression loss. $w$ is for weighing the most important regression loss, and is set to 3 here.We didn't include the object loss for it's only a detection task. The classification was done in the recognition model, since it's so complicated and will reduce the accuracy to classify over a thousand types of OBC in the detection model.

This refined structure and innovative approach position YOLO-X as a state-of-the-art object detection model, building upon the strengths of its predecessors while addressing their limitations. **Our scenario is easier than traditional target detection because we only need to consider the detection of text, so there are no influencing factors such as occlusion and large target size differences**. Therefore, we only need to modify the head structure to train on the target detection dataset to get the model we need in the end.

## 4.2 Scanned to Hand-Printed Transcription

From the detection part, we can get a scanned version OBC. Then, we build the transcript part to the Hand-Printed version. In the meantime, make the images more distinct. We have two thoughts to achieve the aims, and here are details of them. Firstly, traditional computer vision methods are used to handle simple transcript problems, and Generative Adversarial Networks (GAN)[22] methods are used to restore severely missing or indistinguishable ones.

**Algorithm 1** Traditional Scanned to Hand-Printed Transcription Pipeline

```
Dependencies:
Import os, torch, math, numpy, cv2
Imports: Import modules and datasets
for each images do
    Load the image
    Define sharpen kernel, applying sharpen filter, returning sharpened images
    Convert to grayscale, ready for binarizing the image
    Eliminate the image noise that doesn't match the grayscale of the main body
    Invert colors to fit the Hand-Printed version
end for
Return processed images
```

### 4.2.1 Traditional Image Filter and Sharpening Method

Following the detection phase where scanned images are obtained, our primary objectives are to enhance image clarity and convert the images to have a white background with black text. This conversion is crucial as our recognition model is specifically trained to identify images with white backgrounds and black text. To achieve this, our code is structured into four main components. Initially, the input image is read and converted to a **Grayscale image**. Subsequently, a **Sharpening Filter**[23] is applied to enhance image clarity. Additionally, binary processing is utilized to isolate the white portion, representing the text, while eliminating noise and imperfections that could distort the final outcome. Finally, the colors are inverted to ensure a white background with black text, we used **MedianBlur Filter**[24] to handle this step and the processed image is saved and outputted. However, during the execution of the code, several issues have been identified. Firstly, the denoising process is not consistently effective, as some cracks are occasionally retained, significantly impacting the experimental results. Furthermore, some output images remain blurry, indicating a need for refinement in the sharpening filter to improve image quality. **These challenges highlight areas for improvement in the image processing pipeline to enhance the overall effectiveness and accuracy of the recognition process**.

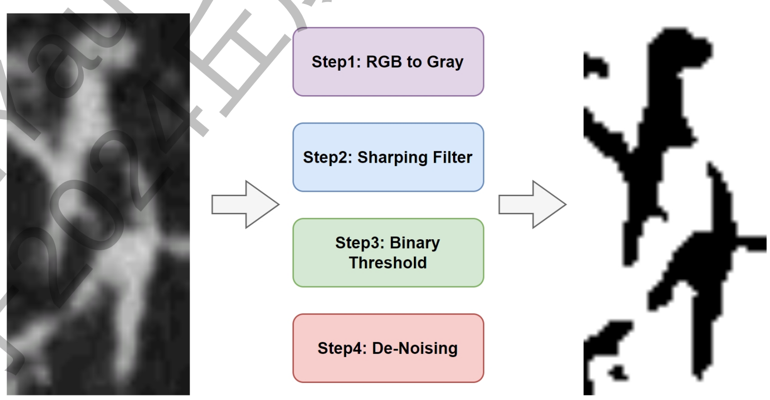



***Figure 10:*** *Process diagram for converting Scan to Hand-Printed*

### 4.2.2 OBC-Trans-GAN for Style Transfer

For some challenging transcript scenarios, traditional image processing methods may not work well. Here, we draw on the ideas of CycleGAN[25] and propose an innovative model **OBC-Trans-GAN**, which is a framework designed for image-to-image translation tasks. CycleGAN is the abbreviation of

Cycle-Consistent Generative Adversarial Network. It can help us to transfer the style of one picture to another style. For example, if people give some pictures of zebras and horses, the model can identify the style of the two different animals. Then, once we input a picture of zebra, it can give us a picture of the horse, or the other way around, without changing setting and other objects in the picture, just like the example below. In our scenario, given a blurred and incomplete **Scanned** image, convert it into **Hand-Printed** image, as shown in Fig. 11.

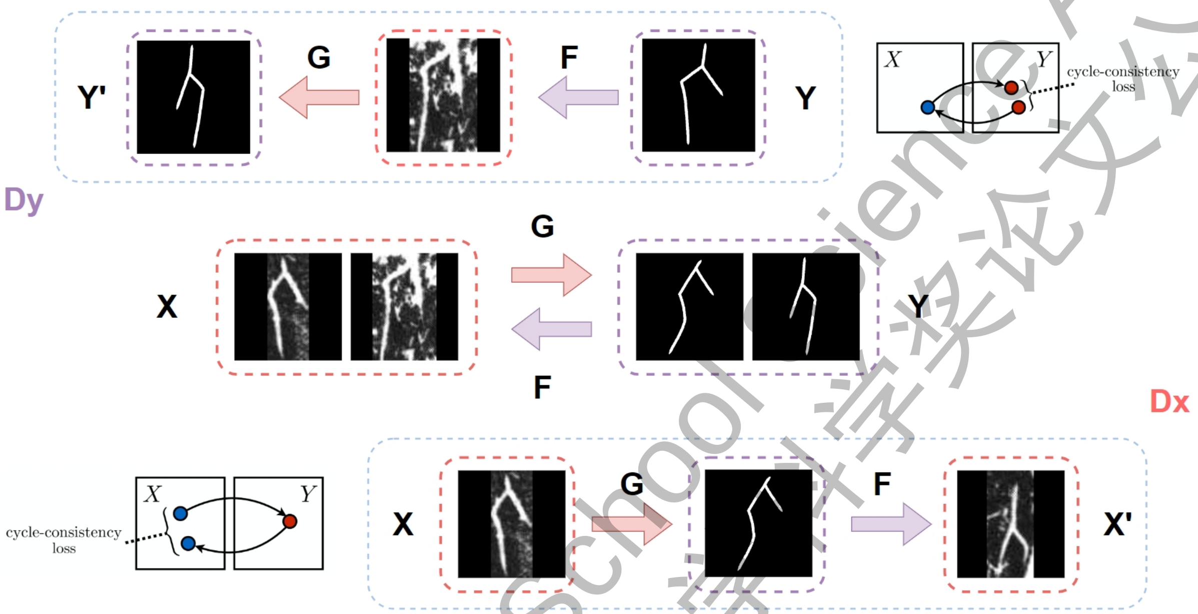



***Figure 11:*** *How OBC-Trans-GAN works*

Our model contains two image generation mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, where $X$ means **Scanned** image, $Y$ means **Hand-Printed** image, which draw on the idea of CycleGAN **Cycle**. On the one hand, we hope that Scanned image can directly generate Hand-Printed image, and on the other hand, we also hope that Hand-Printed image can generate Scanned image in reverse. In order to allow training to proceed well, there are two associated adversarial discriminators $D_y$ and $D_x$. The discriminators used in this framework are based on PatchGAN architecture[26] , which evaluates parts of the image rather than the whole, contributing to a more detailed and local realism. The generators take example by U-Net[27] and DCGGAN[28] architecture. $D_y$ encourages G to translate X(Scanned) into outputs indistinguishable from domain Y(Hand-Printed), and vice versa for $D_x$ and F. To further regularize the mappings, we also learn from two ***cycle consistency losses*** that capture the intuition.

Cycle consistency loss is to try to minimize the change of the background, the size of the picture. For loss part, we use **Least-Square Loss**, a way to minimize the sum of squares of residuals. The function of this is to judge whether the picture the model generates is A to the B picture. For example, we give some pictures of Scanned and some pictures of Hand-Printed. Then, we give another picture of Scanned, trying to let the model generates a picture with Hand-Printed using this picture. The function of the Least-Square Loss is to calculate the similarities between the picture of Hand-Printed that the model generates and the Hand-Printed pictures giving by us to the model. When we translate from one domain to the other and back again we should arrive at where we started: forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$ [25]. If the Least-Square Loss is very small, it means the model has converged, and the generated picture is useful. In fact, not only we hope that the discriminator can judge that the generated picture is a Hand-Printed picture, which means the generator is intelligent and advanced, but we also hope that the discriminator can judge that the generated picture is a fake picture, since this means the discriminator is very advanced and in return it can help us have a more realistic and advanced generated picture. Finally, Image-to-image translation is very important task, through above methods, OBC-Trans-GAN could learn the mapping between Scanned image and Hand-Printed image, then we can apply to our OBS pipeline.

### 4.3 OBC Recognition

After finishing all the steps above, the input is already clear and are able to recognize. However, it's still not a simple task to match these OBC to their modern meanings. When we are searching for related works, we found a program on github that's focusing on recognizing OBC, it's the code of the article[8]. We download the code and run the program that can draw OBC on it to recognize which character it is. The UI is well-designed, having clear direction and is easy to understand, which we've decided to use as our basic program framework. However the model it contain has low accuracy, which can only recognize few character that's frequently seen in the datasets. When it comes to the OBC that's rarely seen in the dataset-which is the most part-it can't predict accurately. We think that there are two factors that disabled the model's efficiency and accuracy:

- **Long-tailed & Over-fitting**: Firstly, the datasets used is not big enough, missing a lot of characters. There are many categories that contain only one image, and they are included in both training set and testing set. As, a result, the model is over-fitting and has low generalization performance as well as accuracy that seems high but actually low. To solve this problem, we searched for many different datasets and found that HUST-OBS[5] is the best one for this task for its variety in category and sufficiency in images. We separate the datasets into two parts, 80 percents is in the training set and 20 percents is in the testing set. For some categories with only one image, we put them into the training set to make sure that their won't be unknown OBC in the testing set to affect the accuracy. Under these precondition, the images are randomly distributed into two sets. After setting up training and testing set, we numbered each categories so that the output of the program could be the character itself instead of the ID of it, which make our program easy to use.
- **Model Capability**: Secondly, the inaccuracy may also caused by the deficiency of the Shallow Neural Network. We are inspired by the code of article[5] which used ResNet-50 instead to manage the task. To improve the model, we tried to used LeNet[29], AlexNet[30], VGG[31], ResNet[32], and VIT[33] as backbone in our model and select a best candidate for our pipeline.

The next chapter will introduce the accuracy of each model. Overall, the classic ResNet and the epoch-making VIT are the best backbone models, and we will introduce the principles of these two models and how we use them in OBC.

ResNet is a very classic model that aim to eliminate the drawbacks of neural networks with extremely deep layers, which can't avoid the problem of vanishing gradient. Theoretically, deeper network has better performance, but the deeper layer will "forget" the feature input after all the layer about it. ResNet solve this problem by adding direct connect between the layer on the top and the deep layer between two layer not adjacent. As a result, the input of every layer is not just form the layer just above it anymore, data from different layer that's processed in different degree are all contained. ResNet solved the problem by adding shortcut and highway networks, which increase the complexity as well as the efficiency. Basically, model with greater complexity like ResNet-50 is better than those with less layer like ResNet-34, but we still want to test whether the smaller network like ResNet-34 can have better performance since it is more easy to run and train. In our work, we first build the normal convolution and activate layers. Then we added residual blocks to make skip-connection, or shortcut. At last, we put layers and blocks in sequence to create the forward spreading model. The next step is to use the loss to modify our models by using the method of back-propagation.

Vision Transformer (VIT)[33] is a very popular architecture that performs well in classification tasks.. Transformer model as well as other attention models are widely used in NLP field, but since a image is not like a piece of sentence,it coudln't be applied to this model. The author of VIT came up with a new idea to make it possible to use Transformer in computer vision field. They separated the image to many parts and input to the Transformer model in a sequence so that the place of each small part are recorded.VIT successfully give the task of computer vision a new way other than CNN (ResNet is still a improvement of CNN), so we want to apply this to our model to test which side is more effective in recognizing OBC, therefore, we proposed the **OBC Vision-Transformer (OBC-VIT)** architecture, OBC-VIT is displayed in Fig.12. We used the softmax function as the activation layer in the mdoel:

$$P(y|x) = \frac{e^{h(x,y_i)}}{\sum_{j=1}^{n} e^{h(x,y_j)}} \tag{2}$$

where $h(x,y)$ representing a function that indicates the likeliness of one specific prediction, and the denominator representing all $n = 1781$ categories, the whole function, known as softmax, serves as the activation function, returning a probability between 0 and 1. Finally, we use seesaw loss instead of cross entropy loss to solve Long-Tailed problem.

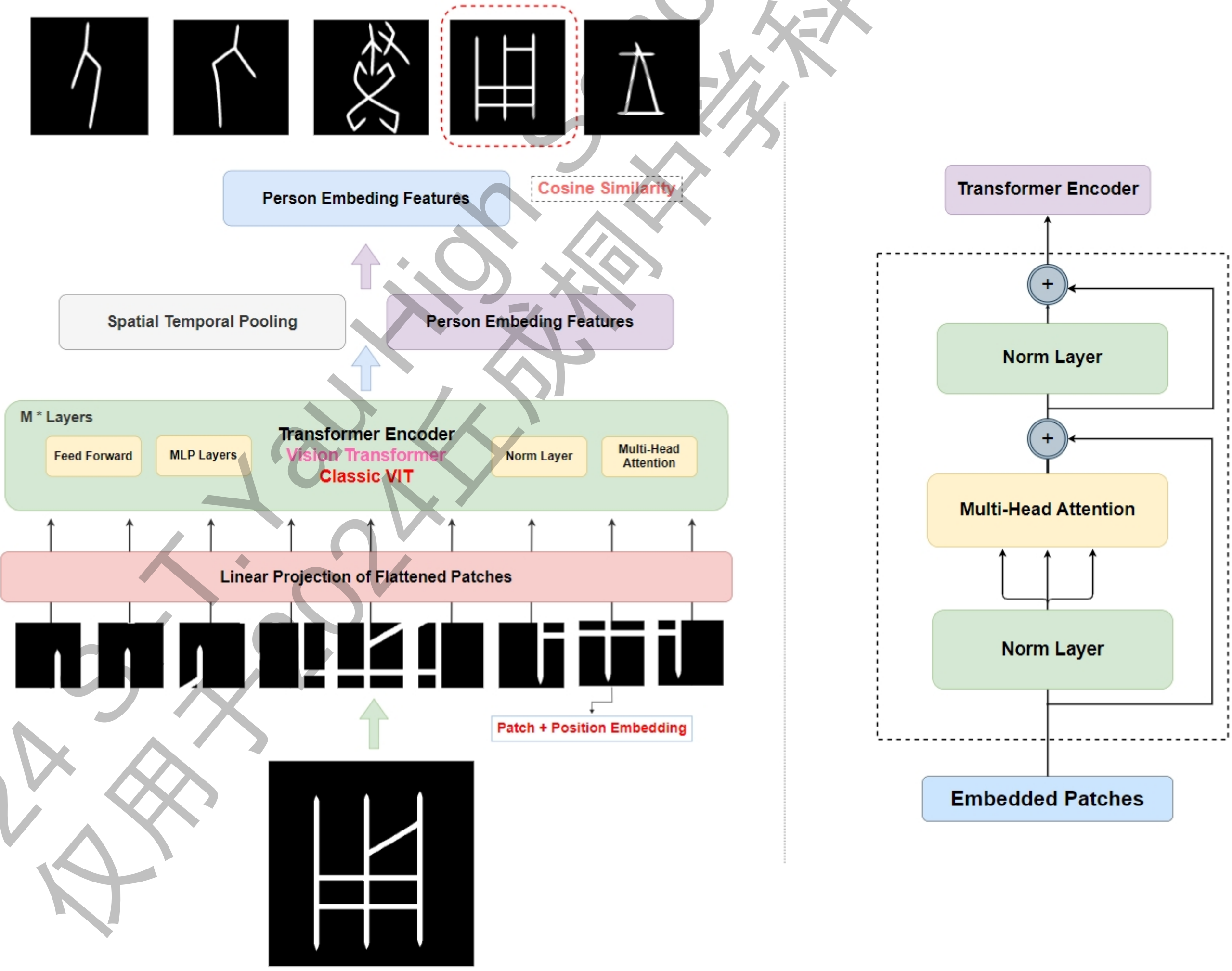



***Figure 12:*** *The Framework for OBC-VIT*

## 4.4 Solving OBC Long-Tailed Problems

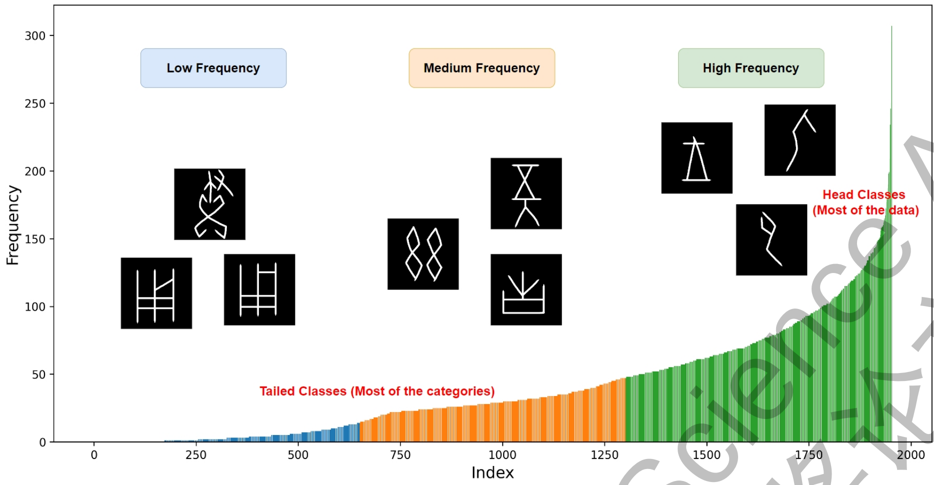



***Figure 13:*** *Long-tailed Problem: The number of samples in each class sorted in an ascending order*

The extreme massive number of categories and the limited data of OBC highlighted the long-tailed problem in OBC recognition. From analyzing data from HUST dataset shown in Fig.13, we concluded that some data augmentation measure must be applied. we Integrate 4 complementary technologies, this part will explain the role of each method, and the experimental section will analyze the impact of each method through ablation studies.

### 4.4.1 Dynamically Re-balance Gradients: Applying the Seesaw loss

Different from other long-tailed problem, OBC Long-Tailed problem is significantly hard for its extreme imbalance distribution. Therefore, the traditional loss function is no longer suitable: The loss is mainly decided by the model's performance on the major classes, while the tail classes are somehow "neglected", as a result. The overall performance of the model seems quite well, but what hide behind the illusion is that the model can hardly recognize OBC from tail class since they are almost not involved in the training and adjustment according to the loss. To raise the influence by the tail classes on the loss function, we applied a innovative loss called seesaw loss [34]. Just as its name suggested, the seesaw loss is the art of balancing: the difference in weight according to the sample number can, to a certain extend, balance the tail classes with the major classes, resulting in a more comprehensive loss that reflect the performance of model in every classes. Specifically, this is hoe the seesaw loss function:

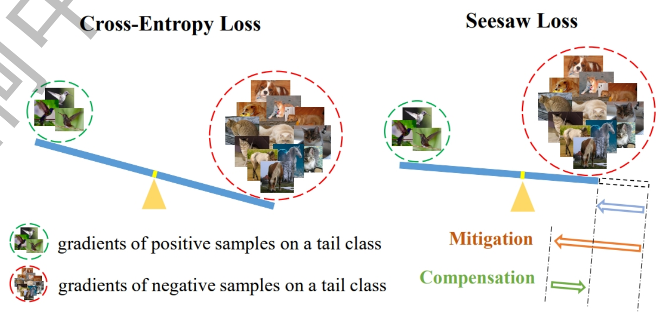



***Figure 14:*** *Seesaw Loss Principle [34]*

$$L = \sum_{i}^{C} y_i \log\left(\frac{e^{z_i}}{\sum_{j}^{C} S_{ij} e^{z_j} + e^{z_i}}\right) \tag{3}$$

The most important part in the loss function is the $S_{ij}$, the special weight designed for compensation of the tail classes, it's related to the frequency of the class and is greater when the class is really rare. However, there's one drawback for interference, the tail class may have some unexpected false due to the decreased penalty. There are mainly two ways to deal with it: For the model itself, we can add an additional factor to focus on the wornly classified situation, instead of the whole category. The other way is to avoid the over interference by beforehand fill the tail class to some degree by augmentation practices.

### 4.4.2 Designed for Pictographic based Data Augmentation Method

We also thought about generating samples for the tail classes to fill the gap. At first, we thought about using some basic image processing method like rotating, stretching, dilation, and compression15. The figure below is an illustration used in the OBI-100[6] model, but we added some extra methods like **blurring, shaking, and adjusting luminance**. These methods are simple but effective in creating new images that's different from the original one but can still identified as a same character. In this part, we mainly perform data enhancement on the long-tail category. The major class part is the traditional computer vision enhancement method, such as crop, hue, contrast, etc.

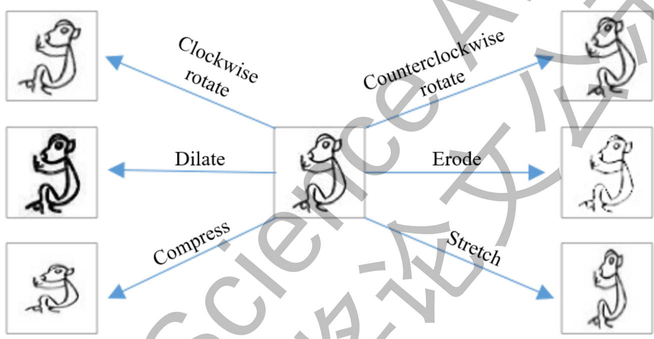



***Figure 15:*** *Pictographic Augmentation Methods[6]*

### 4.4.3 Leveraging GANs for Synthetic Data Generation in OBC Long-Tail Problem

Generative Adversarial Networks (GANs) have become a pivotal method for generating synthetic data, offering a solution to the long-tail distribution challenge prevalent in many datasets. This section delves into the theoretical principles, benefits, and practical steps involved in using GANs to mitigate the OBC Long-Tailed problem. GAN is a model that's widely used to manage data augmentation tasks. It contains generator and discriminator, each help develop others by modifying their own model to discriminate false data or to generate data that seems true. The two networks are trained in a continuous game, where the generator aims to produce increasingly realistic data that can deceive the discriminator.

The long-Tailed problem refers to the imbalance where a few classes have abundant samples, while many others are underrepresented. This imbalance skews model training towards majority classes. We use GAN to generate more augmented images in order to fill the gap with more authentic samples, and increase the diversity of the data. The optimization objective is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p\text{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{4}$$

where $G$ and $D$ represents generator and discriminator, respectively. $\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$ shows the probability that the discriminator can tell the fake data generated buy the generator. That's why the generator want to minimize the value, while discriminator want to maximize the value. With the traditional method and GAN, the differences between number of sample in major class and tail class can be lowered, while still keeping the reliability and quality of the data, there are three benefits to using this method:

1. Data Augmentation: GANs can create diverse and realistic samples for underrepresented classes, enhancing the training set.
2. Feature Learning: By exposing models to a wider variety of samples, GANs improve the learning of feature representations for rare classes.
3. Reduced Overfitting: Synthetic data helps prevent overfitting to the majority classes by providing more examples of minority classes.

**However, using synthetic data for training will reduce the accuracy of the major class. Therefore, we use the following two-stage training method to ensure the accuracy of the major class while also having good tail class recognition capabilities.**

### 4.4.4 Two-Stage Training Pipeline on Real and Synthetic Data

Directly using generated dataset also has its own limit. If the superfluous fake, or the generated, data were used in the training, the model may lost it's realness, for fake data can never totally replace the real data, and the massive generated data according to only a few samples must be inferior. Therefore, we adopt this two-stage training method [35]. This approach aims to optimize model performance across both major and minor classes:

- **Stage1 Combined Data Training**: Initially, we combine real data with synthetic OBC samples generated by GANs. This mixed dataset helps in balancing the class distributions, ensuring that the model is exposed to a wide variety of examples, including those from underrepresented classes. By training on this enriched dataset, the model learns robust feature representations that account for the diversity in both major and minor classes.
- **Stage2 Fine-Tuning with Real Data**: In the second phase, we fine-tune the model using only real data. This step refines the model's ability to generalize to authentic data distributions, mitigating any potential biases introduced by synthetic samples. Fine-tuning ensures that the model's performance is optimized for the actual deployment scenarios, enhancing its accuracy and reliability on real-world data.

This two-phase approach effectively addresses the challenges posed by long-tail distributions, leading to a model that excels across a spectrum of data scenarios, the details is shown in Fig. 16.

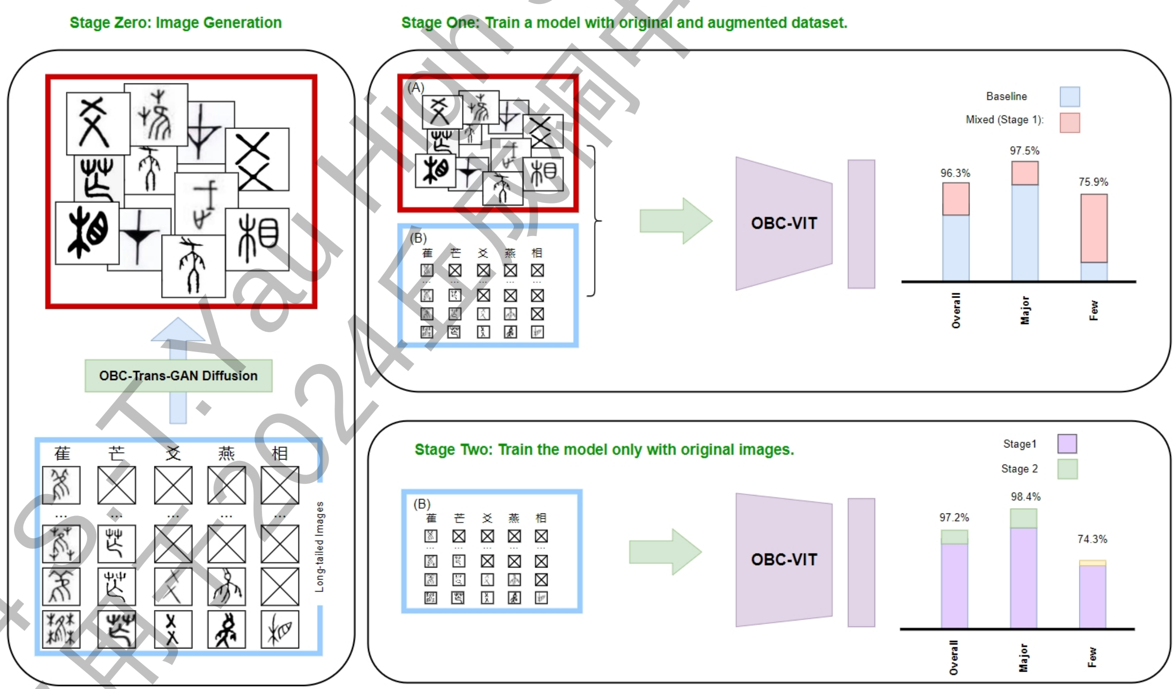



*Figure 16: Overview of Our 2-Stage (Training & Fine-tuning) Long-Tailed OBC Recognition Pipeline[35]*

## 4.5 OBC Open-Set Recognition

In many real-world applications, Deep Learning models encounter scenarios where they must classify data from both known and unknown categories. Traditional closed set recognition assumes all test samples belong to known classes seen during training. However, this assumption is often unrealistic, leading to the development of Open-Set Recognition (OSR) algorithms [36][37][38]. **OSR aims to correctly classify known categories while identifying and handling previously unseen classes as "unknown"**, Fig. 17 shows the principle of OSR. OSR builds upon traditional classification by incorporating mechanisms to detect unknown samples. This involves:

- **Thresholding**: Setting a confidence threshold below which inputs are considered unknown.
- **Distance-Based Methods**: Utilizing distance metrics (e.g., Euclidean distance) to determine how far a sample is from known class distributions.
- **Probability Estimation**: Modifying output layers, such as using OpenMax instead of SoftMax, to estimate the likelihood of being an unknown class.

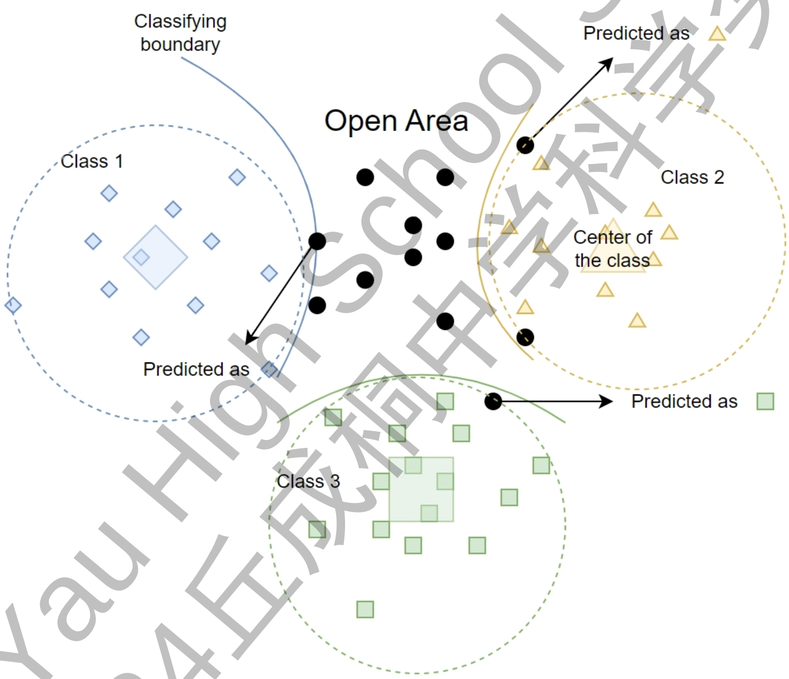



***Figure 17:*** *Explain the Principle of Open-Set Recognition*

Besides the deciphered OBC (已被翻译的甲骨文), there are other un-deciphered OBC (还未被翻译的甲骨文) or OBC that's not included in the datasets, which is OSR in the Deep Learning field. After solving all these problems we discovered a last question that our model don't have any open detection ability at all, which is shown in Fig.18. In other words, it can't tell that an input images is not seen before since it will always try it best to match the image to a closest category and return that as the prediction. However, the input is not always a character really exist, some may be part of the background wrongly detected in the detection step. Although we can improve the ability of the detection model, there will always be inevitable wrong target, so we can define this problem in the field of OSR.

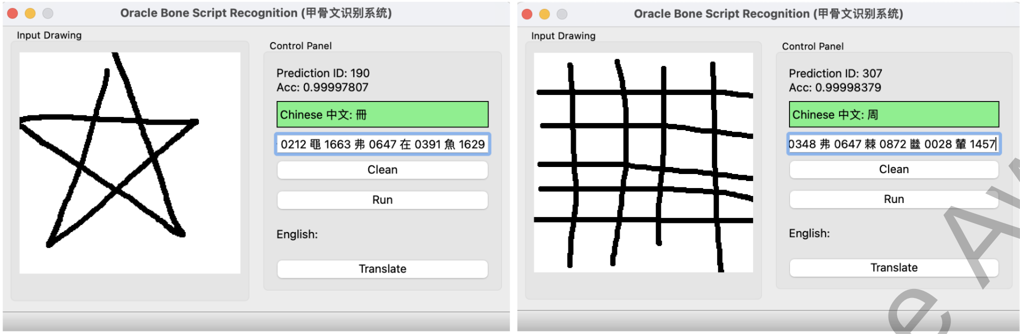



***Figure 18:*** *Problem with Open-Set Recognition in OBC Area*

There's roughly two ways for solving OSR problem. Some generated un-deciphered data can be fed to the machine to "create" un-deciphered classes so that the un-deciphered data will fit into these classes instead of the existing classes. However, the un-deciphered data varies, so it's hard to cover all of them will generated classes. What's more, using generated data is always not a save way. Therefore, inspired by the method of applying an Openmax layer[39] we chose another way: **by limiting the decision board of the existing classes to create an open space, so that the un-deciphered will automatically go to this class**. The traditional classifying method will calculate the distance of the object to all the other classes' center on the feature map, and choose the shortest one to be the result. This new method, in contrast, will draw border line for all the classes, and judge whether the object is close enough that it is inside some classes' border line. The method to decide the border lines are not simply just enclose the area with all the sample in the class. There's a specialized layer before the activating layer to decide the borderlines. This is how this layer, similar with the Openmax layer, works:

**Algorithm 2** The Algorithm of Transformer Based Multi-Heads and Multi-Layers Openmax OSR[39]

**Require:** Feature vector for $\mathbf{v}(\mathbf{x}) = v_1(x), \ldots, v_N(x)$
**Require:** $\alpha$, the number of classes to revise
1: Reserve only the right prediction and calculate the mean vector for each classes: $\frac{1}{n}\sum_{n=1}((v_1 n(x))$
2: through analysing the distribution, which fit in the Weibull distribution, calculate decision border lines
3: **for** $i = 1, \ldots, \alpha$ input data **do**
4: calculate fittingscore$\omega(\mathbf{x})$ according to the border line and the distance
5: **end for**
6: Revise feature vector into activation vector $\hat{v}(x) = \mathbf{v}(\mathbf{x}) \circ \boldsymbol{\omega}(\mathbf{x})$
7: Define $\hat{v}_0(x) = \sum_i v_i(x)(1 - \omega_i(x))$.
8: Pass activation vector to the next layer
9: Reject input if the vector fit into none of the existing classes

Openmax method added a layer before between the last feature layer and the activating layer, so it can draw the boundary based on the feature map of the last convolutional layer. We made some changes and proposed **transformer based multi-heads and multi-layers Openmax OSR algorithm**, since our layer is connected to the Transformer model's multi-heads on the last layer. **The multi-heads are responsible for different feature of the object, with different attention focus, so it will be more detailed. Besides, we also linked multiple final layers instead of just the last one**. Similar to the DenseNet idea[40], we directly connect these layer to that second-to-last layer for deciding boundary.

# 5 Experiments

## 5.1 Experiment Environment

The experimentation is conducted utilizing a school laboratory GPU server with 128 cores, coupled with 8 NVIDIA RTX 3090Ti GPUs and 256G memory capacity. The server's operating environment is Ubuntu 18.04 and the code running envs is Python 3.8.19, PyTorch 2.0.1 and Cuda 11.4. The APP use Qt to create the GUI with the VIT model and it deploy on Windows laptop. We use this server to train and evaluate YOLO-X-OBS, OBC-Trans-GAN and OBC-VIT.

## 5.2 OBS Detection: OBS-Det

**Datasets:** OBS dataset is really hard to find, unlike OBC, the original images of OBS are difficult to collect and annotate. Datasets was downloaded from YinQiWenYuan (殷契文渊)[2], is used to train this model. This dataset (甲骨文字检测数据集) is the only one that we found suitable for training OBS detection. PascalVOC Dataset format is the one we use, for it is a frequently used standard format in object detection tasks. The data dividing is quite traditional, separating 20% data to the testing set, 80% to the training set. However, instead of using the popular method, using one fourth of the training set to validate, we choose to have one eighth of the training set to be the validation set for adjusting parameter, in order to ensure that the model is fully trained. We also applied the straightforward k-fold method in our training set, specifically, setting k to 5 to minimize the impact of randomness brought by dividing dataset.

**Model:** Object detection is a long standing field with dozens of impressive models. We tried to applied several famous models, the result is listed in Table2. DETR model is quite unique, among the three models, it has the highest AP(average precision), but also need significant more detection time. The high accuracy is important, but the slow training and detecting speed, illustrated by the FPS (Frames Per Second), is a serious problem. We finally give up using DETR, a inventive application of Transformer on detection task, with its great convenience(eg. no need of NMS, end-to-end). Instead, we decided to use the YOLOX model for our detection task. The batch size is 64, efficient enough, while not that large so it won't require too much memory. The number of iterations is set to 120,000 for the sufficient training. Learning rate is 0.001 and will adjust to 0.1 times after 80,000 and 100,000 iterations for the high-speed converging in the begining and fine-tune training in the end. The momentum value is 0.9, helping accelerate convergence and reduce oscillation. Weight-decay value is 0.0005, preventing overfitting problem. Grad clip. It turns out that since the detection on OBS is not that complicated, with only 2 colors and no need to classify, which we separate in to the later model, and YOLOX do have really well performance after our modulation. The training speed: FPS(53.4) as well as AP(47.3%) are impeccable.

***Table 2:*** *Inference Speed and Accuracy of Different Detection Frameworks*

| Model | Inference Speed (FPS) | Average Precision (AP) |
|---|---|---|
| FasterRCNN Ren et al. [2015] [41] | 52.2 | 32.30% |
| YOLO-7 Wang et al. [2023] [42] | 43.2 | 36.50% |
| DETR Carion et al. [2020] [43] | 25.4 | 42.52% |
| **OBS-Det** (Ours) | **53.4** | **47.30%** |

[2] https://jgw.aynu.edu.cn/home/down/index.html

*Figure 19: OBS-Det Detection Visualization: Using Bounding Box to Determine OBC in OBS*

Our OBS-Det detection results as shown in Fig.19. It's quite obvious that the model can detect all the OBC on the image, and almost no wrong detection of noises. The performance of our OBS-Det model is good enough for the next step: **Transcription**.

**Challenge of Ligature:** The next problem is Ligature, which is the hardest problem we met in this task. Though it's hard to detect and separate them, it's possible through adjusting the parameter to avoid too frequently merging two bounding boxes in order to keep the ligature separated. **Below in the Fig. 20, it turns out that our model successfully detected the Ligature**. After doing a lot of research about ligature, we also find an interesting pattern: the majority of the ligature are consist of a number an da character meaning "month" "year" or "hundred", which can be seen as an example of using our model to analyze OBC in order to boost the research efficiency for the scholars.

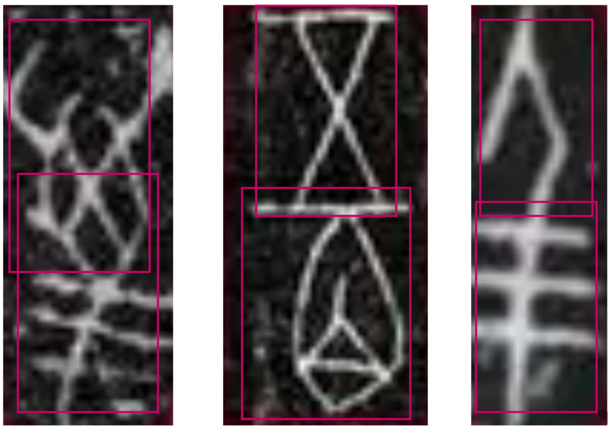


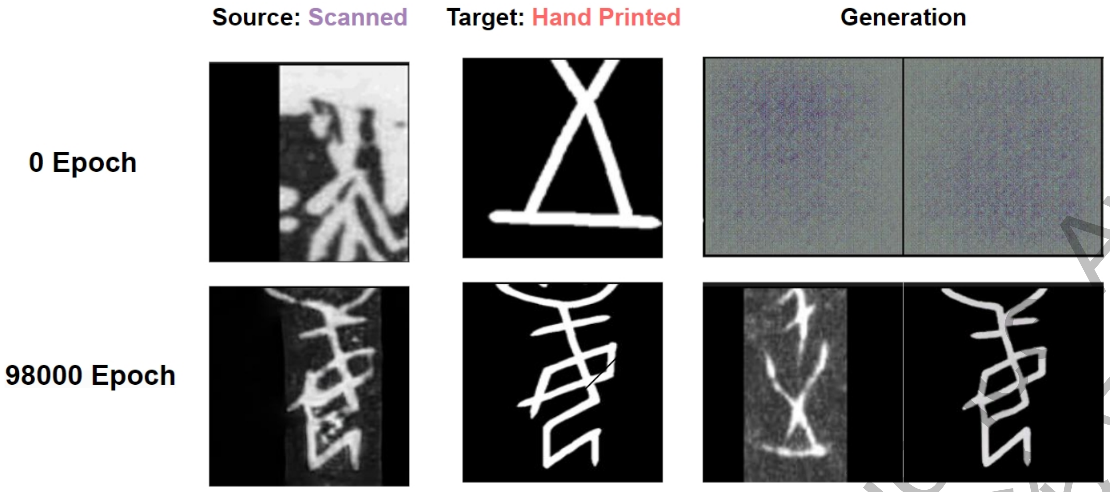
***Figure 20:*** *The example of the output of our OBS-DET model, about the ligatures that are each separated into two distinct OBC.*

One thing worth mentioning is that we do think about doing transcription before detection to eliminate noise first and make it easier to handle detection task. However, the detection models are surprisingly accurate, so we put transcription after it. The advantage of ordering models this way is that the transcription of single OBC, which thanks to the detection model, is much easier than the transcription of OBS. In addition, there's more OBC datasets to use, providing a higher chance to make our model transcription model efficient and further benefit the most important task, recognition task.

## 5.3 Transcript: OBC-Trans-GAN

**Dataset:** The model OBC-Trans-GAN was built by us in this part to transform the scanned version images into Hand-Printed version, while generating some other images of the same word in the same time. In order to better train or transforming model, we need to get a dataset with paired scanned and Hand-Printed version, which the dataset from Oracle-241 [11] (The dataset collected by Beijing University of Post and Telecommunication) perfectly fit.

**Model:** We trained the model, and finally it worked satisfactory. The model includes two parts. If the image is not hard to identify, the model will use the traditional method to identify it. including gray-scale conversion, edge sharpening filter, binary processing, and denoising. However, some images, shown in Fig.21, are hard to transform to the Hand-Printed version, due to the excessive noise point that overlap with the OBC. **It's clear that it's necessary to do the transcription by using OBC-Trans-GAN model, for these hand-printed version can be easily figured out which character it is. If we just use the traditionally unprocessed image, shown in the above image, it's basically just a bunch of noise. That's why we insisted to develop OBC-Trans-GAN to handle the small amount of extreme hard case.**



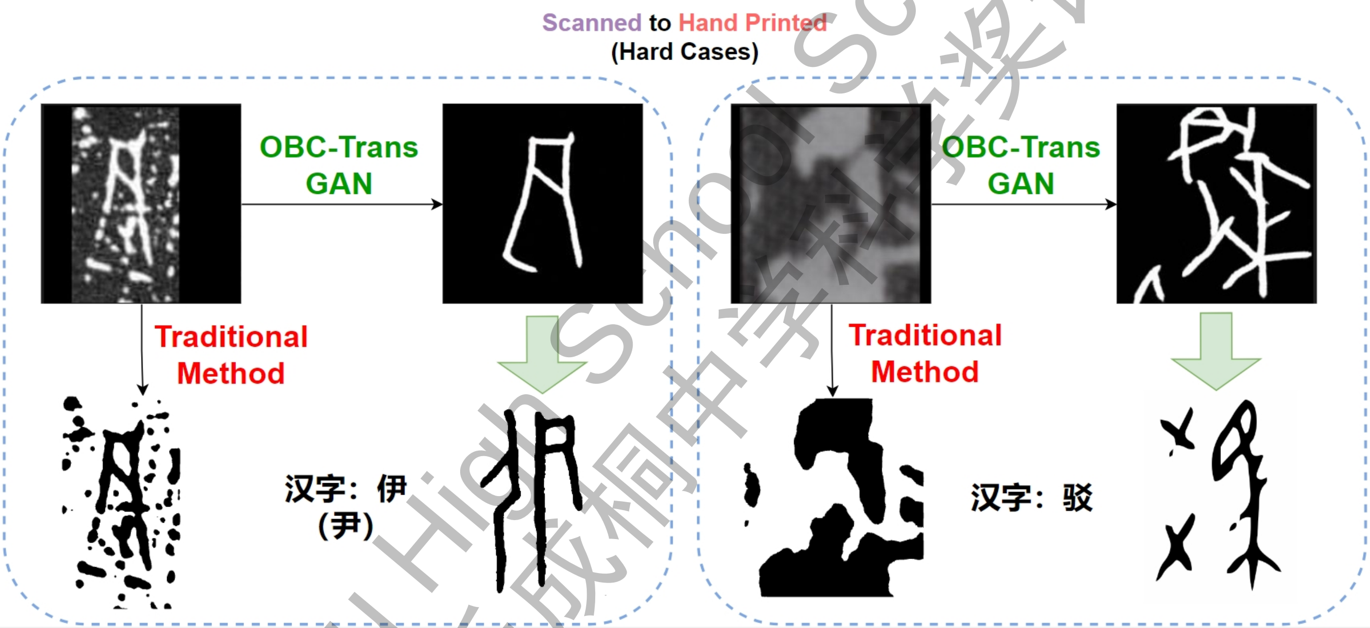




*Figure 21:* *OBC-Trans-GAN Results for Solving Hard Cases. The character on the left bottom is the character 尹, and the one on the left bottom is the character 驳, both are surrounded by serious noise*

For these situation, a better model is required, so here comes the second part, the CycleGAN model. Instead of directly eliminating the noise on the original image, which is how we usually deal with the easy cases, it generates new hand print version of the same OBC. After training for 45,300 steps, it works really well, supplementing to data samples while dealing with the hard case. The OBC-Trans-GAN models can simply and quickly deal with the easy cases that only has little noise on it, while using CycleGAN to handle the hard cases, resulted in the high efficiency and also generality.

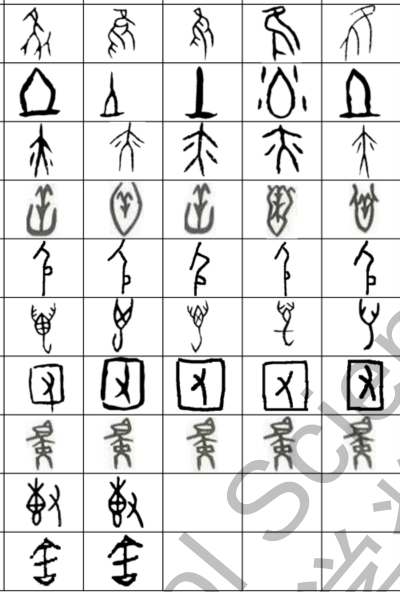
## 5.4 OBC Recognition: OBC-VIT

**Dataset:** HUST is the basic dataset we used for training recognition model. The division of the dataset is generally the same as the way we do in the OBS-Det model. The only adjustment we do is the measure of preventing the data from some tail classes, which only have a few images, are all divided into the testing set, therefore can't be recognized since they're missing or not enough in the training set. We first select out all the categories with 1 or 2 images, putting them into the training set. Although the k-fold method and other data augmentation measures we take to handle the long-tail problem can avoid complete unfamiliar OBC, we still think it's necessary to do this step.

**Model:** Plenty well-known models were tested and compared as listed in the Table3. Though RNN models are more classic, it turns out that the OBC-VIT model we used, adjusted from the VIT model, has a better performance. We first resized the images to unified scale. Then we horizontally flipped half of the images; rotate the image within clockwise and counterclockwise 15 degree around center(64,64); apply color jittering by the chance of 0.8; convert to grey scale by a chance of 0.2, as some simple data augmentation attempts. We pad the corner of the image with white to help preserve the object position. The last step of the data preprocessing is normalizing the image with mean deviation value of [0.85233593, 0.85246795, 0.8517555], and the standard deviation value of [0.31232414, 0.3122127, 0.31273854], in order to ensures consistent scaling and centering of pixel values across different images. However, unlike the detection task, recognition is much challenging and important, so instead of simply using VIT model, we combined some adjustment inspired by others. To be more specific, we applied the method of VIT distillation and pruning, it turns out that they work really well. We also download the application from github and replace the original model, which is not that accurate, with our OBC-VIT model to build a independent application that can show and test the capacity of out recognition part. User can draw OBC on the screen, simulating the Hand-Printed version OBC after the transcription, and the output will be the prediction made by the model about which OBC character it is as well as the corresponding Chinese character. The chart in Fig.22 are some OBC examples we randomly choose, showing the performance of our model by using the gui we build for testing.

***Table 3:*** *Accuracy for Different OBC Models, OBC-VIT V1 uses Pictographic + GAN Augmentation, then OBC-VIT V1 includes seesaw loss to improve the performance. OBC-VIT V3 uses a two-stage training strategy to improve the accuracy of real and synthetic data*

| Model | Epochs | Validation Accuracy |
|---|---|---|
| LeNet Lecun et al. [1998] [29] | 85 | 72.47% |
| AlexNet Krizhevsky et al. [2012] [30] | 47 | 81.52% |
| VGG16 Simonyan et al. [2014] [31] | 36 | 87.25% |
| VGG19 Simonyan et al. [2014] [31] | 33 | 91.42% |
| ResNet Kaiming et al. [2016] [32] | 30 | 94.30% |
| **OBC-VIT V1**: Augmentation (Ours) | **30** | **95.20%** |
| **OBC-VIT V2**: Augmentation & Seesaw Loss (Ours) | **30** | **96.30%** |
| **OBC-VIT V3**: Augmentation & Seesaw Loss &2 Stage (Ours) | **30** | **97.20%** |



| Ground Truth Category (HUST ID) | # Samples | Examples 1 | Examples 2 | Examples 3 | Examples 4 | Examples 5 | Accuracy |
|---|---|---|---|---|---|---|---|
| 隹 1544 | 122 | | | | | | 100.00% |
| 土 390 | 105 | | | | | | 98.10% |
| 亦 130 | 84 | | | | | | 97.62% |
| 岳 1772 | 66 | | | | | | 98.48% |
| 尻 585 | 64 | | | | | | 96.88% |
| 万 1354 | 63 | | | | | | 95.24% |
| 図 10 | 27 | | | | | | 85.19% |
| 瞚 1153 | 6 | | | | | | 66.67% |
| 专 91 | 2 | | | | | | 50.00% |
| 舍 1323 | 2 | | | | | | 100.00% |

***Figure 22:*** *Validation Accuracy of our OBS-VIT on Partial Categories of the HUST-OBS Datasets*

## 5.5 The Performance of Open-Set Recognition

Our recognition range is quite wide, including more than 1500 classes. After going through the OBC-VIT model, all the data were categorized into these 1500+ classes, grouping with the same category's sample on the area. Each group has a class center decided by the average vector distance of all the data in the same group. The whole area are dived into 1500+ small region, and the open area that filled between these region. The final layer before the activation that we added was connected to the multiple OBS-VIT layers or the second-to last layer of the ResNet. We also adjust the threshold for rejecting input from **0.15 to 0.35**. Table 4 below showed several accessories we tried. It turns out that VIT's multi-head characteristics perfectly fit the method of adding boundary. By calculating a average feature vector based on the multiple head of multiple layers on OBC-VIT and setting the **Threshold to 0.25**, **the accuracy of distinguishing un-deciphered OBC reached 84.76%**. Using multi head and layers has an unexpected outcome: under the thrshold with highest known type accuracy, it will slightly decrease the accuracy of known type. We consider this as a trade-off of using this method, increasing the ability of dealing with OSR but slightly decrease the ordinary performance.

***Table 4:*** *Experiments of Transformer Based Multi-Heads and Multi-Layers Openmax OSR*

| Different Backbone Based Openmax | Threshold | Unknown Acc | Known Acc |
|---|---|---|---|
| ResNet Second to Last Feature + Openmax | 0.15 | 64.38% | 91.43% |
| ResNet Second to Last Feature + Openmax | 0.25 | 74.43% | 90.25% |
| ResNet Second to Last Feature + Openmax | 0.35 | 71.28% | 89.73% |
| OBC-VIT Second to Last Feature + Openmax | 0.25 | 78.33% | 92.24% |
| OBC-VIT Multi Heads & Layers feature + Openmax | 0.15 | 79.37% | 94.45% |
| **OBC-VIT Multi Heads & Layers feature + Openmax** | **0.25** | **84.76%** | 94.38% |
| OBC-VIT Multi Heads & Layers feature + Openmax | 0.35 | 81.52% | 93.43% |

From the dataset of HUST-OBC, which totally included 62989 images of un-deciphered OBC, we randomly choose 5000 images to check whether our model works well. The result shows our model can identify most of them as "un-deciphered" so it's practical and efficient. After adding that border line drawing layer, the ability of OSR significantly raised. Here are several samples that we chose from the un-deciphered categories and the prediction of the recognition model, as shown in Fig.23

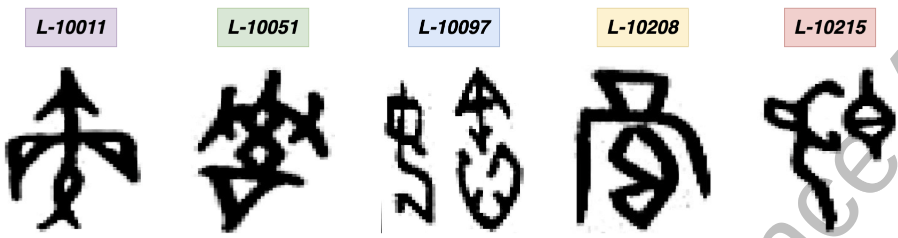



***Figure 23:*** *There are totally 62,989 images of un-deciphered OBC in HUST-OBC dataset. Some images are randomly choosen from them and are identified by our model, while these five images are examples. The number above the images are the codes of them in the dataset. The model can correctly output "un-deciphered" for nearly all of them, showing the efficiency of the model.*

# 6 Conclusion

Overall, this project proposed an innovation pipeline to address the challenges and difficulties encountered by archaeologists in excavating and identifying oracle bones.

The first challenge is extracting OBC from OBS affected by **Cracks**, **Weathering** and **Similarities**. We developed **OBS-DET** to assist archaeologists in extracting individual OBC from bones or shells and it can solve the ligature problem perfectly. It can achieve an impressive 47.3% AP at a speed of 53.4 FPS on an RTX 3090Ti. No matter in terms of detection accuracy or inference speed, it is the best model currently, and it can quickly help archaeologists segment effective OBCs.

The second challenge is that we need to convert the OBC from the **Scanned** version to **Hand-Printed Transcript** which can make the OBC easier to identify. Due to the long burrying, there are some OBC that are worn out seriously. Therefore, despite using the traditional image processing techniques to transcribe the ordinary scanned version OBC, we also use **OBC-Trans-GAN** to restore the image through precise transformation and generalization. Combining the two techniques, the model can handle most situations.

The final challenge is to accurately identify which modern Chinese character corresponds to OBC, while solving the Long-Tailed problem and Open-Set recognition problem simultaneously. We developed **OBC-VIT** which can achieve a remarkable accuracy of **97.2%**. To address the **Long-Tailed distribution problem**, we have developed a specialized augmentation technique focused on hieroglyphics, while combining generative adversarial network to generate sample of tail categories to improve the recognition performance of rarely represented characters. We also solved the OBC-OSR problem in this paper.

We have also tried to use cutting-edge AGI models such as **GPT-4o**, but their ability to recognize and generate oracle bones is relatively average. Therefore, our pipeline is basically the best solution at present, both in terms of professionalism and practicality. We have helped some archaeologists to detect and identify oracle bones, improving their work efficiency.

# References


[1] H.-h. Chou, "Chinese oracle bones," *Scientific American*, vol. 240, no. 4, pp. 134–149, 1979.

[2] B. Li, Q. Dai, F. Gao, W. Zhu, Q. Li, and Y. Liu, "Hwobc-a handwriting oracle bone character recognition database," in *Journal of Physics: Conference Series*, vol. 1651, p. 012050, IOP Publishing, 2020.

[3] S. Huang, H. Wang, Y. Liu, X. Shi, and L. Jin, "Obc306: A large-scale oracle bone character recognition dataset," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 681–688, IEEE, 2019.

[4] M. Wang and W. Deng, "A dataset of oracle characters for benchmarking machine learning algorithms," *Scientific Data*, vol. 11, no. 1, p. 87, 2024.

[5] P. Wang, K. Zhang, Y. Liu, J. Wan, H. Guan, Z. Kuang, X. Wang, L. Jin, and X. Bai, "An open dataset for oracle bone script recognition and decipherment," *arXiv preprint arXiv:2401.15365*, 2024.

[6] X. Fu, Z. Yang, Z. Zeng, Y. Zhang, and Q. Zhou, "Improvement of oracle bone inscription recognition accuracy: A deep learning perspective," *ISPRS International Journal of Geo-Information*, vol. 11, no. 1, p. 45, 2022.

[7] C. Zhang, R. Zong, S. Cao, Y. Men, and B. Mo, "Ai-powered oracle bone inscriptions recognition and fragments rejoining," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 5309–5311, 2021.

[8] Y. Fujikawa, H. Li, X. Yue, C. Aravinda, G. A. Prabhu, and L. Meng, "Recognition of oracle bone inscriptions by using two deep learning models," *International Journal of Digital Humanities*, vol. 5, no. 2, pp. 65–79, 2023.

[9] X. Liu, W. Gao, R. Li, Y. Xiong, X. Tang, and S. Chen, "One shot ancient character recognition with siamese similarity network," *Scientific Reports*, vol. 12, no. 1, p. 14820, 2022.

[10] J. Liu, R. Li, T. Fu, Z. Yin, and Z. Yang, "Interpretation study of unknown oracle bones based on self-training learning," in *Proceedings of the 2022 4th International Conference on Video, Signal and Image Processing*, pp. 7–11, 2022.

[11] M. Wang, W. Deng, and C.-L. Liu, "Unsupervised structure-texture separation network for oracle character recognition," *IEEE Transactions on Image Processing*, vol. 31, pp. 3137–3150, 2022.

[12] X. Han, Y. Bai, K. Qiu, Z. Liu, and M. Sun, "Isobs: An information system for oracle bone script," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 227–233, 2020.

[13] X. Yue, H. Li, Y. Fujikawa, and L. Meng, "Dynamic dataset augmentation for deep learning-based oracle bone inscriptions recognition," *ACM Journal on Computing and Cultural Heritage*, vol. 15, no. 4, pp. 1–20, 2022.

[14] J. Li, Q.-F. Wang, K. Huang, R. Zhang, and S. Wang, "Diff-oracle: Diffusion model for oracle character generation with controllable styles and contents," *arXiv preprint arXiv:2312.13631*, 2023.

[15] J. Li, Q.-F. Wang, K. Huang, X. Yang, R. Zhang, and J. Y. Goulermas, "Towards better long-tailed oracle character recognition with adversarial data augmentation," *Pattern Recognition*, vol. 140, p. 109534, 2023.

[16] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[17] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[19] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, K. Michael, J. Fang, C. Wong, Z. Yifu, D. Montes, *et al.*, "ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations," *Zenodo*, 2022.

[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018.

[21] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th international conference on pattern recognition (ICPR'06)*, vol. 3, pp. 850–855, IEEE, 2006.

[22] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[23] J. G. Schavemaker, M. J. Reinders, J. J. Gerbrands, and E. Backer, "Image sharpening by morphological filtering," *Pattern Recognition*, vol. 33, no. 6, pp. 997–1012, 2000.

[24] M. Sukassini and T. Velmurugan, "Noise removal using morphology and median filter methods in mammogram images," in *The 3rd International Conference on Small and Medium Business*, pp. 413–419, 2016.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," pp. 2223–2232, 2017.

[26] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," *arXiv preprint arXiv:1803.07422*, 2018.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015.

[28] K. Du, C. Liu, L. Cao, Y. Guo, F. Zhang, and T. Wang, "Double-channel guided generative adversarial network for image colorization," *IEEE Access*, vol. 9, pp. 21604–21617, 2021.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[31] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," pp. 770–778, 2016.

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[34] J. Wang, W. Zhang, Y. Zang, Y. Cao, J. Pang, T. Gong, K. Chen, Z. Liu, C. C. Loy, and D. Lin, "Seesaw loss for long-tailed instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9695–9704, 2021.

[35] J. Shin, M. Kang, and J. Park, "Fill-up: Balancing long-tailed data with generative models," *arXiv preprint arXiv:2306.07200*, 2023.

[36] C. Geng, S.-j. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3614–3631, 2020.

[37] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4016–4025, 2019.

[38] P. Perera, V. I. Morariu, R. Jain, V. Manjunatha, C. Wigington, V. Ordonez, and V. M. Patel, "Generative-discriminative feature representations for open-set recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11814–11823, 2020.

[39] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1563–1572, 2016.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[42] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7464–7475, 2023.

[43] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

# Acknowledgments

This research project would not have been possible without the support of the following people.

Firstly, we want to express our sincere gratitude to Professor Yujin Zhang [3] at South China Normal University, Director of the Research Center for the Language of Unearthed Documents. He graduated from the Department of Chinese Literature at Peking University in 1988 and received a doctoral degree. He is mainly engaged in the language research of unearthed documents, the study of ancient characters and Chinese characters, and has published eight academic books and more than 120 academic papers in renowned journals such as "Chinese Language," "Language Studies," "Ancient Chinese Studies," and "Literature and History." We knew Professor Zhang from the Internet and he was willing to help us generously and freely. He patiently answered our questions and introduced us to numerous websites where we could learn extensively about OBC. We gratefully appreciate him for all the advice he offered throughout our project.

Moreover, during the project of developing a model for identifying OBC, our teachers and parents also helped us significantly. Mr. Yang, teacher of our current school, provided substantial support for this long-term project and guided us through the research. He also gave us access to the school's server. Our parents assisted us in finding experts in this field and provided us with the necessary facilities to build the model and conduct experiments.

We are also thankful to everyone who has conducted research in this field for publishing their findings and making their models open-source. We read nearly all of their papers, which gave us our initial ideas on how to build the model. We got new ideas from them throughout the process. This saved us a significant amount of time since they had already developed numerous datasets in this field, which requires considerable effort. Because of these datasets, we are able to train our model effectively and achieve high accuracy. In addition, we are thankful for Professor Enda Wu for his post of the clear introduction of the Machine Learning, Deep learning, Convolutional Neural Network and so on on bilibili, which helps us a lot in learning the basic knowledge of this field.

After reading several papers on the Internet and discussing for several times, we chose our subject, which was using AI to recognize OBC. We chose it because we both thought ancient characters were very interesting, and it was still within our capabilities to solve the problem. The dataset we used to train our OBS detection model was sourced from the website Yinqi Wenyuan; we asked a professor from Beijing University of Post and Telecommunication and trained the transcript part; Finally, we used the dataset of HUST-OBS to train our OBC recognizion part. We trained our model with these datasets on the school's server, using the knowledge we learn at school and on the Internet.

Junfu Chen and Hanzhi Zhang worked as a team to complete this research together. Both two students were responsible for the introduction, literature review, conclusion, and drawing the illustrations in the paper. Junfu Chen was responsible for writing the abstract, challenges and significances, and the first half of the methods section. Hanzhi Zhang was responsible for writing the latter part of the methods section and experiment part. During the process, when students had some problems with coding, the teachers would give some advice and helped solve them. They also gave us some interesting and novel ideas throughout the research to make the model more thorough. Again, we are very grateful to everyone who helped us directly or indirectly in our research.


[3]http://wxy.scnu.edu.cn/a/20191021/2091.html

# Appendix

## A. OBC Visualization

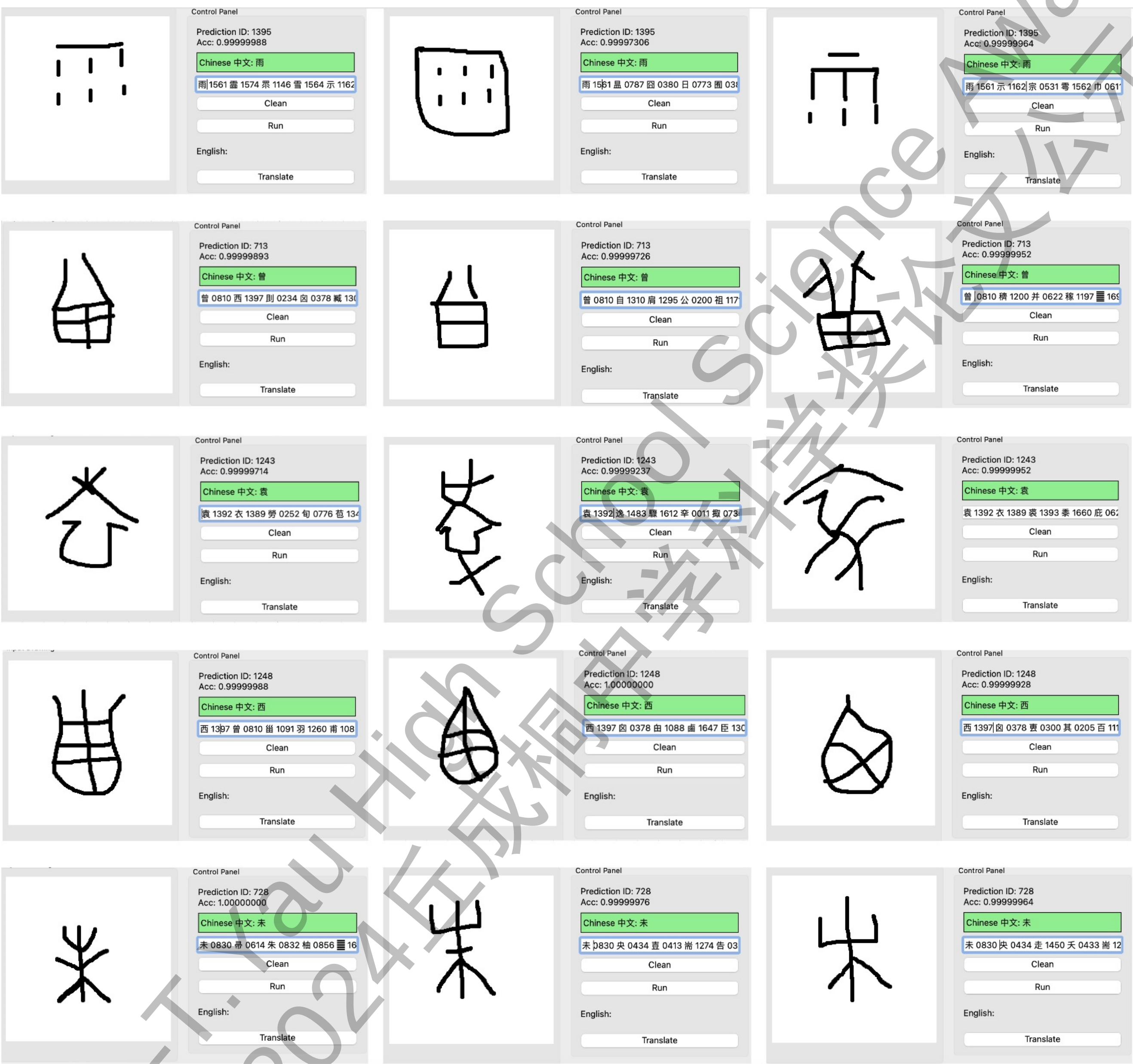



***Figure 24:*** *The OBC in first to five line represent modern Chinese "雨", "曾", "袁", "西", "未" respectively. These Chinese all have different forms of OBC. We can see that our model can detect all of their forms accurately.*

## B. AIGC Capability Exploration


***Figure 25:*** *We let Gemini-1.5-Flash and GPT-4o, two famous AIGC, to identify these OBC, but they can't do it*

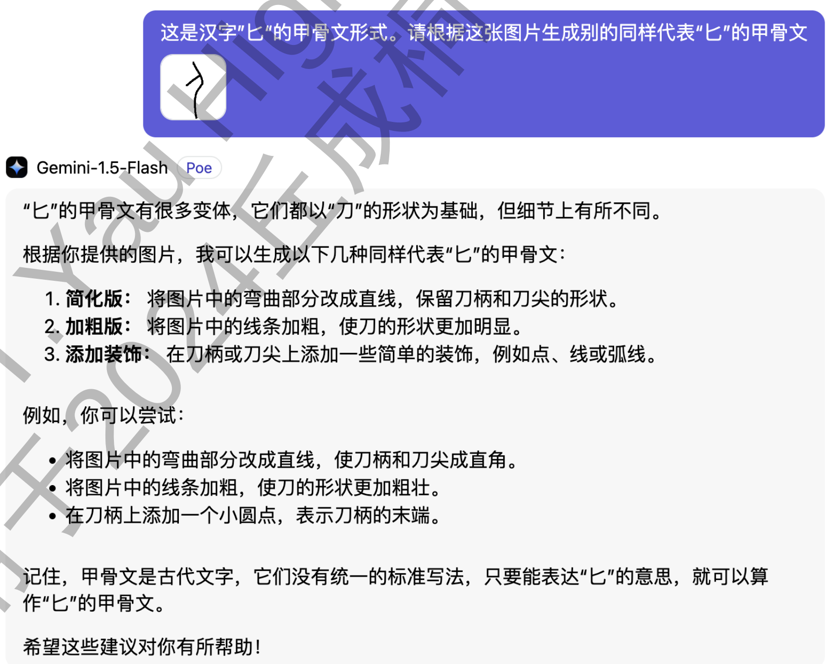



***Figure 26:*** *We provide Gemini-1.5-Flash with a image including an OBC representing Chinese "匕", letting it generate a similar image including an OBC with same meaning, and find out that it can't achieve that*


# 陈筠富 Junfu Chen

*The Affiliated High School of South China Normal University*

*+86 18320136260    chenjf.victor2023@gdhfi.com*


## Introduction

- During my formative years, I cultivated a keen interest in a variety of subjects, with a particular fondness for Biology and Mathematics. As I progressed into junior school, I was captivated by the allure of Computer Science, prompting me to delve into the realm of programming languages such as C++.

- In my leisure hours, I find great pleasure in playing the piano and delving into the enigmatic realms of oddities, such as the intricate world of fossils. This pursuit not only serves as a source of fascination but also as a means of expanding my knowledge and understanding of the natural world.

## Educational Background

- Academic Excellence Scholarship Y10 (Top 10%) 2024
- Y10 GPA: 4.90/5.00 2023-2024
- Attending Summer section of Johns Hopkins University. Taking *Introduction to Neuroscience* and *Maths for Sustainability*, while getting A and $A^+$ respectively. 2024
- TOEFL iBT score: 108/120 2024
- Gobi Trekking Challenge Camp (walk for 88 km in 3 days), $1^{st}$ in the group 2024
- The 11th Steinway National Youth Piano Competition Guangzhou final, Excellence Award 2023

## Academic Honors

- AMC10 Honor Roll, Top 5% 2023
- Euclid Math Contest Top 25% 2024

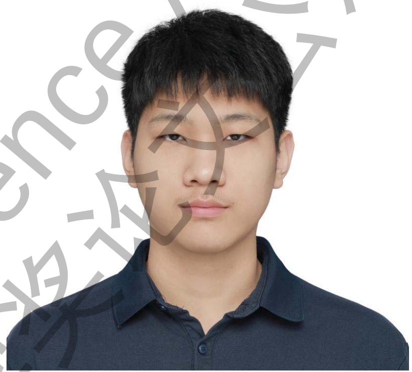
## Interests

- Learned piano and Guzheng for about 10 years, passed ABRSM $8^{th}$ grade.
- Vice president and core member of several clubs in the school.

# 张瀚之 Frank

*The Affiliated High School of South China Normal University*

*+86 14739575687*      zhanghz*.frank2023@gdhfi.com*

*After being introduced to natural science by some experiment, I indulged in physics and other natural science subjects. The fun of analyzing things around us through formula and the desire of exploring natural world. That's why we choose to* ***decode*** *the past.*

*Also, since I was young, I learned a lot of knowledge about ancient China. I love poetry, history, drawing and everything related to Chinese culture. That's also one reason we choose to applying our knowledge on the oracle researching field. My interest in the ancient culture is the reason why we choose to decode the* ***past****.*

## HONORS & ACADEMICS

| | |
|---|---|
| SAT score: 1540 / 1600 | May 2024 |
| TOFEL score: 109/120 | Oct 2023 |
| Academic Excellence Scholarship G10 (Top 10%) | 2023-2024 |
| Y10 GPA: 4.96/5.00 | 2023-2024 |
| AP Calculus BC: 5 AP Physics C Mechanism: 5 | May 2024 |
| BPhO round 1 gold | Nov 2023 |
| Euclid Mathematics Contest top 25% | April 2024 |

## ACTIVITIES & PROJECTS

| | |
|---|---|
| Summer course at John Hopkins University, Exploring the universe with space telescopes | Jun-July 2024 |
| European literature summer camp- Spain&France topic 2023; Northern Europe topic 2024 | Jul-Aug |
| Vice President of Physics club | 2023-24 |