Hieroglyphics

Data Science 440
Project Report (to instructor)

Team Six

Juliana(Jiayin) Hu, Sydney Wehn, Kangdong Yuan, Xueqing Zhang, Yuqi Gao

Table of Contents:

Personal reflection 3

Week One 5

Week Two 7

Week Three 13

Week Four 16

Week Five 17

Week Six 19

Week Seven 22

Week Eight 29

Week Nine 32

Week Ten 39

Week Eleven 43

Week Twelve 49

Week Thirteen 50

Kangdong's reflection (Contributed by Kangdong Yuan)

What issues I have faced and personal weekly journal detailing:

In this semester, I encountered a lot of difficulties in our group plan. I am still trying to solve some of these difficulties until now.

First, the research direction and topic of our group made me feel very confused. Make sense of hieroglyphs, this topic can represent many things. First of all, there are many hieroglyphs in the history of the world, including ancient Egyptian hieroglyphs, Mesopotamian hieroglyphs, Asia Minor hieroglyphs and Eastern hieroglyphs. In our group's weekly reports 1, 2, we conducted a detailed analysis of these hieroglyphs, including how these hieroglyphs were deciphered. Then we determined that the direction of our research is ancient Egyptian hieroglyphs, because ancient Egyptian hieroglyphs are an ancient writing that has become extinct. The interest in ancient Egyptian scripts is much higher than that of other scripts in history.

Afterwards, Professor Redford had a meeting with us and he mentioned that he hoped that we could design a program that could translate the ancient Egyptian language to english. In the next two weeks, we hoped to find a way to try to translate ancient Egyptian hieroglyphs, and then we discovered that although ancient Egyptian hieroglyphs had been deciphered by linguists, but using natural language processing technology (NLP) to translate ancient Egyptian hieroglyphs is a impossible task for us. This requires that we all have a deep understanding of the vocabulary and grammar of ancient Egyptian hieroglyphs, but it is difficult for us to do these things in one semester.

Then I hope to perform image recognition on ancient Egyptian hieroglyphs. But a key difficulty is that it is difficult for me to find a mature Egyptian hieroglyphic picture database. Although I can search a large number of Egyptian hieroglyphic pictures, these pictures have different colors and resolutions. Then I found a database that I can use in a GitHub code repository. Later, our team decided to use the database of Egyptian hieroglyphic letters in this code warehouse for image recognition analysis.

In the process of writing image recognition code, the selection and setting of the programming environment took me a lot of time. First, I used the r language to make my k-means model, but the operating efficiency of the r language is very poor, and the scalability of the r language cannot satisfy me. Then I used python language to make my k-means model, although I successfully used the training model and got results. But python is difficult to write intuitive instructions to help my users use and understand this program. In addition, I want to visualize my k-means model, but it is difficult to do this in the python environment. Then I wrote my code on

jupyter-notebook, which took a lot of time to get familiar with the environment of jupyter-notebook. Then I found that running jupyter-notebook on the local machine was very slow, so I chose the Google colab environment to run my code. These changes in the development environment made me waste a lot of time and caused a lot of burden on me.

In the end, writing a lot of documentation is also very complicated for me. Because I need to write many explanations for my code and generated images, and I also need to explain the functions of different code blocks to my users. In my jupyter-notebook, I wrote a lot of tutorials and set my variables carefully so that users can easily customize their programs by changing some variables.

How much time did I spend:

Every week, our group meets once, each meeting takes about an hour, and I participate in each meeting. Then I basically participated in every meeting with the professor. In addition to this, I often communicate with group members in the group chat room.

At the beginning of the semester, I would spend a lot of time watching materials and documentaries about ancient Egyptian hieroglyphs, including some books on learning the grammar of ancient Egyptian hieroglyphs. I spend about 8 hours a week.

In the process of writing code, I spent a lot of time understanding the principles of machine learning and unsupervised learning algorithms. In addition, I spent a lot of time setting up my programming environment and learning the k-means model. During this period, I worked more than 10 hours a week to solve my problems.

In the final stage of writing documents, I spend more than 12 hours a week due to a lot of paperwork.

Week One:

Correspondence Updates:
This week, we met our team and were assigned our topic of Making Sense of Hieroglyphics. Kangdon and I spoke to Prof Mitrai and set up a meeting for 11:00AM on January 27 to discuss our project and goals. Sydney reached out to Yuqi to verify that this time also worked for her, however we believe that she may be in a different time zone which will make meeting times a little more complicated. We will be following up on this in our report next week.

Organizational Updates:
In addition to setting up a time to meet with our customer/professor, Sydney made a GroupMe to make quick, daily communication easier. She also created a Google Drive folder to make collaboration on weekly reports and meeting agendas as efficient as possible. Also, in order to prepare for our meeting with Prof Mitrai tomorrow, we made an agenda with questions surrounding overall questions about what hieroglyphics entails, what the key objectives and tasks will include for the project, and other items related to which weekly meeting times will be best for the group.

Idea related to main topic
Egyptian hieroglyphs could not be deciphered and read in ancient times. All the grammar and meanings of hieroglyphs were a secret to historians and linguists at that time. The deciphering of the hieroglyphics began when the French discovered the Rosetta Stone, a stele with Greek and hieroglyphs. This allows linguists to infer the specific meaning of each hieroglyph through exhaustive methods, and to permute and combine them. After more than ten years of hard work, linguists deciphered these hieroglyphs. But in modern times, encryption and decryption methods have become diverse and complicated. How to encrypt and decrypt text through the enlightenment of hieroglyphs has become very worthy of our exploration.

Week Two:

1.) Correspondence Updates:

    a.) Found a time what works for group members (weekly together and biweekly with Professor)

i.) 6:00-6:30pm Monday evenings
ii.) Will ask professor in our meeting on Thursday

b.) Breakdown of Work:

Yuqi	- 6.) What are the different types of hieroglyphics
Kangdong	- 5.) Sounding out Egyptian hieroglyphics - 7.) What has been deciphered - 8.) The Alphabet of Egyptian hieroglyphs - 9.) How are they deciphered
Sydney	- 2.) Project Goals - 3.) Steps to Achieve Goals - 4.) What hieroglyphics entail - 10.) Related Work/ Open problems - 11.) Helpful Terms

2.) Project Goals:
a.) Deciphering hieroglyphics to create meaning
b.) Classification of languages based on usage/resources
c.) "How to maintain the reading order of Hieroglyphics script and produce an accurate English meaning based on image processing techniques?"

3.) Steps to Achieve Goals:
a.) Image Acquisition
b.) Image Segmentation
i.) Find reference photos as our test set and find datasets that can be used for classification/ translation as train set
ii.) Decide if we should read left to right or right to left based on face orientation of the symbol
iii.) Possible Methods
(1) Mask-refined r-CNN with ROI Align adjustment
(2) Multi-layer convolution feature fusion and online hard example mining
(3) Compression techniques: Block Truncation Coding with greyscale images
c.) False Positive Reduction

i.) Define accuracy and testing metrics for the outcome to identify which level of certainty we should be achieving

(1) In this experiment, the accuracy, in this case *average percentage of intersection area over union area,* is **67.73%**

d.) ROI matching (region of interest)

i.) Euclidean distance for image mapping

ii.) HOG is a shape-extraction system that is utilized by ascertaining the Euclidean separation between the element vectors acquired from both the query and the data set pictures

e.) ROI sequencing

i.) Validation > adjectives as modifiers > pronouns > consecutive nouns > sentence recognition

f.) English mapping

g.) English sentence reconstruction

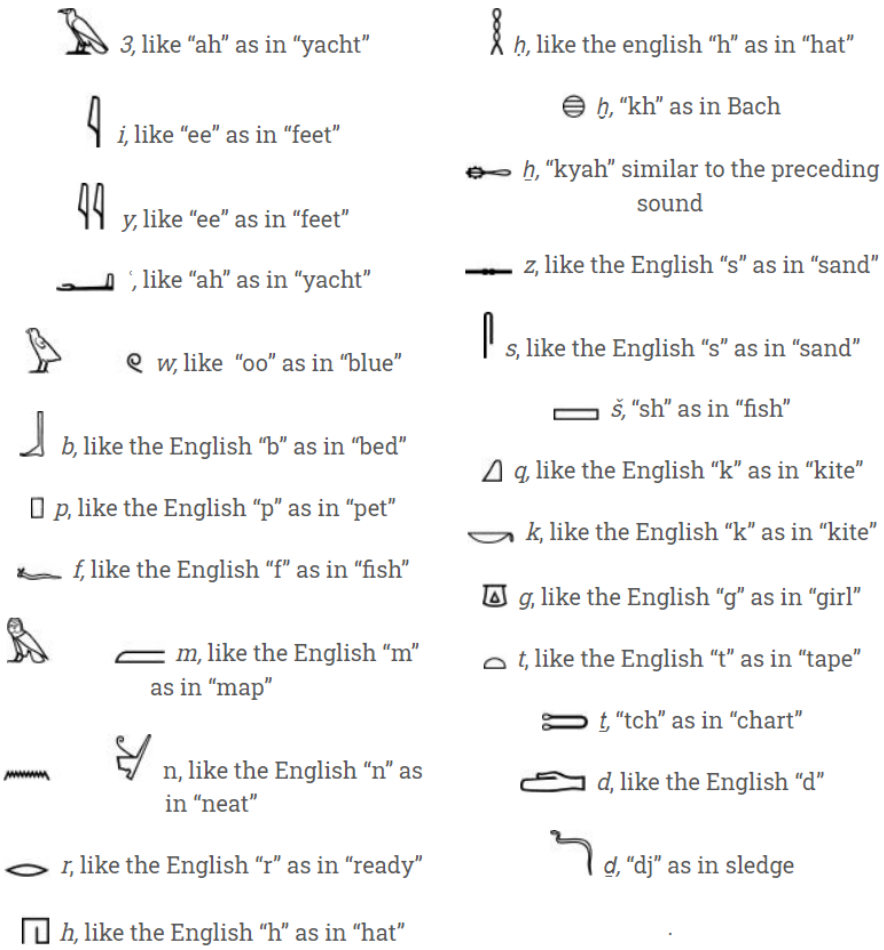
4.) What do hieroglyphics entail

a.) Images of symbols that will be mapped to english sentences and meanings

5.) Sounding out Egyptian hieroglyphics.

Find a figure with a head and start reading toward the front of its head.

Pronunciation



6.) What are the different types of hieroglyphics?

- The hieroglyphics were known by people as different cultures during different eras in history had developed and used them. Therefore, there are certain different types of hieroglyphics, the possibly oldest one found by human kind nowadays are the Egyptcian hieroglyphs which date back around 5,200 years ago.

- Types
    - Anatolian hieroglyphs
        - This type of hieroglyph can roughly date back to the 13th century BC. Used by people in Anatolia, which is roughly located in Western Asia, nowadays is Turkey. The Anatolian hieroglyphs represent the language used by people of that time in that area called Luwian. Typologically similar to the Egyptian hieroglyphs, but is not graphically derived from the Egyptian hieroglyphs, and does not play a sacred role like the ones in Egyptian hieroglyphs.
    - Aztec hieroglyphs
        - This type of hieroglyph can roughly date back to the 16th century according to the extant manuscript people found nowadays. It was used by Nahua people in the central Mexico area. It was been considered as it was adopted from other writing systems, e.g. Zapotec writing system.
    - Chinese hieroglyphs
        - Typically refers to the Oracle bone script, which was the form of Chinese characters used on animal bones or turtle plastrons used in pyromantic divination, which was around the 2nd millennium BC. It was considered as the earliest known form of Chinese writing.
    - Cretan hieroglyphs
        - This type of hieroglyph was used during the bronze age, in Crete, which was located in Greek, there was not lots of archive about this typical hierography, and it wasn't deciphered until this year.
    - Egyptian hieroglyphs
        - Probably the most known hieroglyphs in the world, as they were also shown in a lot of documentaries and movies. It is until now known as the most ancient writing system in the world, and it was used by ancient Egypt, combining around 1,000 distinct characters including logographic, syllabic, alphabetic elements.
    - Mayan hieroglyphs
        - Also known as Maya glyphs, as it was the writing system of the lost Maya civilization, it was created around the 3rd century BC. It was the only Mesoamerica writing system that has been substantially deciphered. Maya writing used logograms complemented with a set of syllabic glyphs, it was originally confused by the Europeans as they discovered it and address it as "hieroglyphs"(Egyptian hieroglyphs) since it made people recall the existence of the Egyptian hieroglyphs, although two writing systems have no relations at all.

- Mi’kmaq hieroglyphs
    - This type of hieroglyphs were developed by the indigenious of Canada, called the Mi’kmaq, belongs to the First Nations people, the system was invented around the 17th century and used until the 19th century, considered to have a relatively short history. This type of writing system referred to the sucker fish’s track at the bottom of the muddy river, so it was also called komqwejwi’kasikl in the Mi’kmaq language, translated to modern english as the “sucker-fish writings”.
- Muiska hieroglyphs
    - The Muiska are the indigenious people of the west of Columbia, as in their civilization, there was no script for text, the Muiska hieroglyphs are only used for numeral countings, and they were depicted on rocks, ceramics and textiles.
- Ojibwe hieroglyphs
    - Although Ojibwe, the language is considered one of the largest Native American languages north of Mexico in terms of number of speakers and is characterized by a series of dialects. But the discovery of its hieroglyphs are not much known, as it was similar to Mi'kmaq hieroglyphic writing, they are found as petroglyphs. There are said to be several Ojibwe elders who still know the meanings of many of the symbols, but as their content is considered sacred, very little information about them has been revealed.
- Olmec hieroglyphs
    - Olmec hieroglyphs designate a possible system of writing or proto-writing developed within the Olmec culture. The Olmecs were the earliest known major Mesoamerican civilization, flourishing during the formative period (1500 BC to 400 BC) in the tropical lowlands of the modern-day Mexican states of Veracruz and Tabasco. The subsequent Epi-Olmec culture (300 BC to AD 250), was a successor culture to the Olmec and featured a full-fledged writing system, the Isthmian (or Epi-Olmec) script. The existence of a writing system in Middle and Late Olmec periods has been a matter of long-standing debate. Direct evidence includes the appearance of individual (potential) glyphs as well as the singleton Cascajal Block that bears a potential text. Indirect evidence includes similarities in calendar and writing systems between subsequent Mesoamerican cultures, for which the Olmecs are sometimes considered a mother culture.

7.) What has been deciphered?

Egyptian hieroglyphs are divided into two writing forms: hieroglyphics and domestic script. With the continuous efforts of ancient Egyptian linguists for many years, a solid foundation has been established for the study of Egyptian characters, and most of the grammar and vocabulary of ancient Egyptian hieroglyphs have been deciphered.

8.) The Alphabet of Egyptian hieroglyphs.

Hieroglyph	Transliteration	Description
	3	Vulture
	i	Reed leaf
	y	Double Reed Leaf
	ꜥ	Arm
	w	Quail chick, rope curl
	b	Foot
	p	Stool
	f	Horned viper
	m	Owl
	n	Water, red crown
	r	Mouth
	h	House plan
	ḥ	Rope
	ḫ	Unknown
	ḫ	Animal belly and tail
	z	Door bolt
	s	Folded cloth
	š	Pool
	q	Hill
	k	Basket with handle
	g	Stand for vessel
	t	Bread loaf
	ṯ	Tethering rope
	d	Hand
	ḏ	Cobra

9.) How are they deciphered?

Medieval to Renaissance

After the Arabs conquered Egypt, the study of hieroglyphics was mainly conducted by Islamic scholars. However, during this period of time, the deciphering of Egyptian hieroglyphs has been slow, and some Islamic scholars have speculated that hieroglyphs may also be a kind of phonetic script.

European scholars began to notice Egyptian hieroglyphics during the Renaissance, and some of them tried to decipher these characters, but many deciphering efforts were slow. During the Renaissance, a large number of ancient Egyptian artifacts including obelisks,

stone steles, and some ancient Egyptian books were acquired by Western European scholars, so this provided the basis for the deciphering of ancient Egyptian hieroglyphs.

After eighteenth century

In the nineteenth century, the deciphering of ancient Egyptian hieroglyphs made a breakthrough, thanks to the discovery of the Rosetta Stone. The Rosetta Stone was discovered by the French army that invaded Egypt. The Rosetta Stone is a stele written in three languages that express the same content. The top one is Egyptian hieroglyphics, the middle one is Egyptian secular writing, and the bottom one is Greek. Champollion, a French linguist, made an epoch-making contribution to the deciphering of Egyptian hieroglyphs. Before he engaged in research, linguists generally believed that hieroglyphs were a kind of ideographic hieroglyphs, so each hieroglyphic symbol represented a word. But Champollion pointed out that on the Rosetta Stone, 1,500 hieroglyphic symbols correspond to 500 Greek words, and each symbol of the hieroglyph should be a part of a word, not a word. Through long-term research and comparison, Champollion has come to the conclusion that Egyptian hieroglyphs are phonetic scripts, and each symbol of a hieroglyph represents a letter. After a long period of work in linguistics, more and more Egyptian hieroglyphs were deciphered.

10.) Related Work/ Open problems

- a.) Egyptian hieroglyphic numbers & mathematics
  - i.) https://www.open.edu/openlearn/science-maths-technology/mathematics-and-statistics/mathematics/egyptian-mathematics/content-section-1.1.2
- b.) Image Processing
  - i.) https://patentimages.storage.googleapis.com/1f/f7/63/a151849d322958/US20050232486A1.pdf
  - ii.) Edge volume calculation unit that sets multiple sets of reference pixels that can be used to calculate the absolute value of the difference between pixel value of reference and of adjacent
  - iii.) Image distinguishing units which thresholds are set to reference images
- c.) Extending Gardiner's Code for Hieroglyphic Recognition and English Mapping
  - i.) https://link.springer.com/article/10.1007/s11042-020-09825-2
  - ii.) Recognition from images to translate into English meanings
    - (1) Meaning of images are found through classification using Gardiner's sign list as a standard to classify hieroglyphics symbols (grouped in similar categories)
    - (2) Optical Character Recognition (OCR) - electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo

iii.) Two types of hieroglyphics
(1) Ideograms
(a) Speak to what the item communicates
(2) Phonograms
(a) Pictures that portray the old Egyptian language sound

11.) Helpful Terms
- Antigrams: examples: united versus untied, “a way with words”
- Anagrams: word that means the opposite of the original word, also a phrase spelled by rearranging letters of another word or phrase
- Polysemy: the coexistence of many possible meanings for a word or phrase
- Monolingual: speaking only one language
- Parallel corpora: a collection of texts, each of which is translated into one or more other languages than the original

Week Three:

Correspondence Updates:

1.) Meeting One: On Thursday, Feb 4 all three group members met with Prof Mitra to discuss project goals and progress. We asked questions and aimed our focus for this week on data acquisition and potential steps on how we will be able to translate Egyptian hieroglyphics into meaningful ideas.
2.) Meeting Two: On Thursday, after the meeting with Prof Mitra, Sydney and Kangdong briefly met to discuss ideas about how we should go about data acquisition and processing.
3.) Meeting Three: On Monday, Feb 8 all three group members met with Prof Mitra, Donald Redford, Saptarshi Sengupta, and Asunta Redford to further discuss ideas that the Redfords were experiencing in their hieroglyphic translations. Sydney documented the conversation and questions asked and answered during the meeting.

Ideas Generated for Data Acquisition:

1. Link sent by Saptarshi: https://www.bbaw.de/
2. Book mentioned by Donald Redford http://onlinebooks.library.upenn.edu/webbin/book/browse?type=lcsubc&key=Egyptian%20literature%20--%20Translations%20into%20English&c=x
3. Book mentioned in email chain by Donald Redford
4. https://lionbridge.ai/datasets/25-best-parallel-text-datasets-for-machine-translation-training/

Problems Associated with Translation of Hieroglyphics:

1.) Separating ghost writing and other writers as more of a forensic challenge
2.) Translation for niche languages with less resources
3.) Differentiating writing styles and how they are different between different people
    a.) Fonts consideration
4.) Differentiating differences in the languages hieroglyphics associated with how the King/ leader spoke through time

Moving Forward:

1.) Identify which direction the clients want to go

    In the previous plan of our group, our hope was to design a program that could convert images of Egyptian hieroglyphs into universal codes (unicode). This program can help Egyptian hieroglyphics researchers convert many hieroglyphic pictures into a database that can be stored in a computer in a short time. But during the meeting, I heard that our customers wanted us to make a program that could translate Egyptian hieroglyphs into English. This translation software will require a lot of knowledge of natural language processing, and we need a certain understanding of the grammar and vocabulary of Egyptian hieroglyphs. So our main work during this time is to understand the grammar and translation of Egyptian hieroglyphs.

2.) Design System Architecture
    a.) Different database/ storage options
        i.) Hierarchical, network, relational, object-oriented, graph, ER model, Document, NoSQL
        ii.) I am thinking that we will be a nonrelational database that uses a simple key-value method to store data.
            (1) https://www.memoq.com/tools/what-is-a-cat-tool#:~:text=The%20term%20CAT%20stands%20for,are%20stored%20in%20a%20database.
            (2) https://www.digitalocean.com/community/tutorials/a-comparison-of-nosql-database-management-systems-and-models
            (3) NoSQL non relational database
                (a) Allows for greater flexibility than a rigid structure imposed by relational databases
                (b) Eventual consistency - no immediate updates
                (c) Options
                    (i) MongoDB
                    (ii) Couchbase
                    (iii) Apache CouchDB
            (4) Key-Value Databases

(a) Store and manage associative arrays (dictionary or hash table)
(b) Retrieve values like simple objects, strings, or JSON
(c) Highly performant, efficient, and scalable
(d) Options
    (i) Riak, Memcached, Redis

b.) Different processing techniques

There are many ways to implement language translation models, but the most important methods now are pytorch and TensorFlow based on python implementation. But these methods require a lot of calculation, which means that it will take a long time for computers to train these programs. An accurate and rich database is also a prerequisite for the success of the model.

3.) Data Acquisition and Processing

Customers gave us some data in the meeting, but most of these data are written in German about Egyptian hieroglyphs, so we need to find suitable databases and resources on the Internet.

4.) Translate their idea to text for training/ testing

Natural language processing is a novel technology for us, and we need some time to adapt and learn this technology. In particular, we need to determine the programming language and algorithm to implement translation, and training a translation model requires a lot of data and time.

5.) EvaluFor a language translation model that has not been trained and optimized for a long time, the accuracy and efficiency of translation are difficult to predict. Due to the limitation of the program production time and the size of the database, the translation program we made hopes to be able to accurately translate some simple words, such as numbers and some pharaoh names, but for logically complex sentences and paragraphs, we cannot guarantee the accuracy of the translation .ation with BLEU scores and levels of accuracy

6.) Deliverables for Future

At this point, teams should be having their data, build their system architecture, and started to think about starting their implementations. Certain teams are still in the needs stage where the project is not very well defined.

7.) Breakdown of Work

Yuqi	
Kangdong	- Find the possible path to perform machine learning - Try to learn ancient Egypt language
Sydney	- Design System Architecture - Translate their idea to text for training/ testing
Juliana	- Not in the group yet. Joined in Week Four. - Working on another mini project related to BMI level
Xueqing	- Not in the group yet. Joined in Week Four - Working on another mini project related to BMI level

Week Four:

This week our group decided to proceed and use the dataset from the following link.
https://github.com/morrisfranken/glyphreader

Jiayin-
I found this link that I believe we can use as labels for our modeling techniques.
https://en.wikipedia.org/wiki/List_of_Egyptian_hieroglyphs#Aa

Kangdong Yuan: I found information about ancient Egyptian hieroglyphs very difficult to find. I watched some documentaries and learning materials about ancient Egyptian hieroglyphs.
https://www.youtube.com/watch?v=I0C4BYy_EWQ
https://www.youtube.com/watch?v=Oens3_k3xmw
https://www.bibalex.org/learnhieroglyphs/lesson/Introduction_En.aspx

Xueqing Zhang -
I manually went through and explored the dataset to brainstorm what kind of models we could build with these hieroglyphics.

The picture is the hare symbol I found from the github link. Below link describes the meaning of the hare symbol as goddess Wenet.

https://iseumsanctuary.com/2018/03/31/the-hare-goddess-wenet/

Sydney - This week I explored data storage and how we could read in the data from GitHub. I ran into a lof of errors as I had never processed image data like this. I realized that I may not have sufficient RAM to process these images on my local computer. I reached out to the PSU computing services and met with my Professor to inquire about how we should process this many images in a productive way.

Yuqi	
Kangdong	Try to learn the Egyptian hieroglyphs Try to learn the computer vision
Sydney	- Data storage and processing
Juliana	- Exploration of potential groupings of the data
Xueqing	- Manual data exploration

Week Five:

Jiayin(Juliana): After the last meeting, we realized we should look for more Egyptian Hieroglyphs datasets with the cartouches, which represents the author and possibly indicates the time period. Therefore, last week, our group denoted most of the time to find more Egyptian Hieroglyphs datasets. However, most of the resources online are images or drawings without any

cartouches. Jiayin found a page with detailed labels of Egyptian hieroglyphs alphabet; Xueqing found the meaning of the cartouches for our precious dataset, Sydney immediately contacted the customer to see if there were certain changes that could be made. Since we did not find more useful datasets, we decided the best solution for the current situation is to use the one dataset we found previously. Since the owner of the dataset had split the data based on their gardener label, we want to see if the images in one label have differences. If so, this could indicate differences between scribes. We wish to understand the differences by clustering. First, we will try to store the data and reproduce the labels, and then we could start investigating the necessary features for classifications.

Xueqing noticed that in the picture folder there are 10 pictures of Egyptian Hieroglyphs and each picture has been named with a number for example (3,5,7…) And under the raw and preprocessed folder, there are both 10 folders named corresponding to the Egyptian Hieroglyph picture's name(3,5,7..) In these 10 folders, there are about 500 pictures of individual hieroglyphic symbols. After checking these symbols, she thinks that each of these 500 pictures of individual hieroglyphic symbols were pieces cut from the original picture that has the same name with the folder.

Sydney worked on storing the data in the database and believes that a simple MySQL Server will be the best bet. She also looked into differentiating factors of scribes to help up continue to search and gain insight into how scribes differ from each other. She also met with Yuqi to work out details regarding the project.

Kangdong Yuan: I tried r language image recognition this week, and then I found that the data structure of r is very inappropriate for image recognition, and my computer does not have enough memory to read the image.

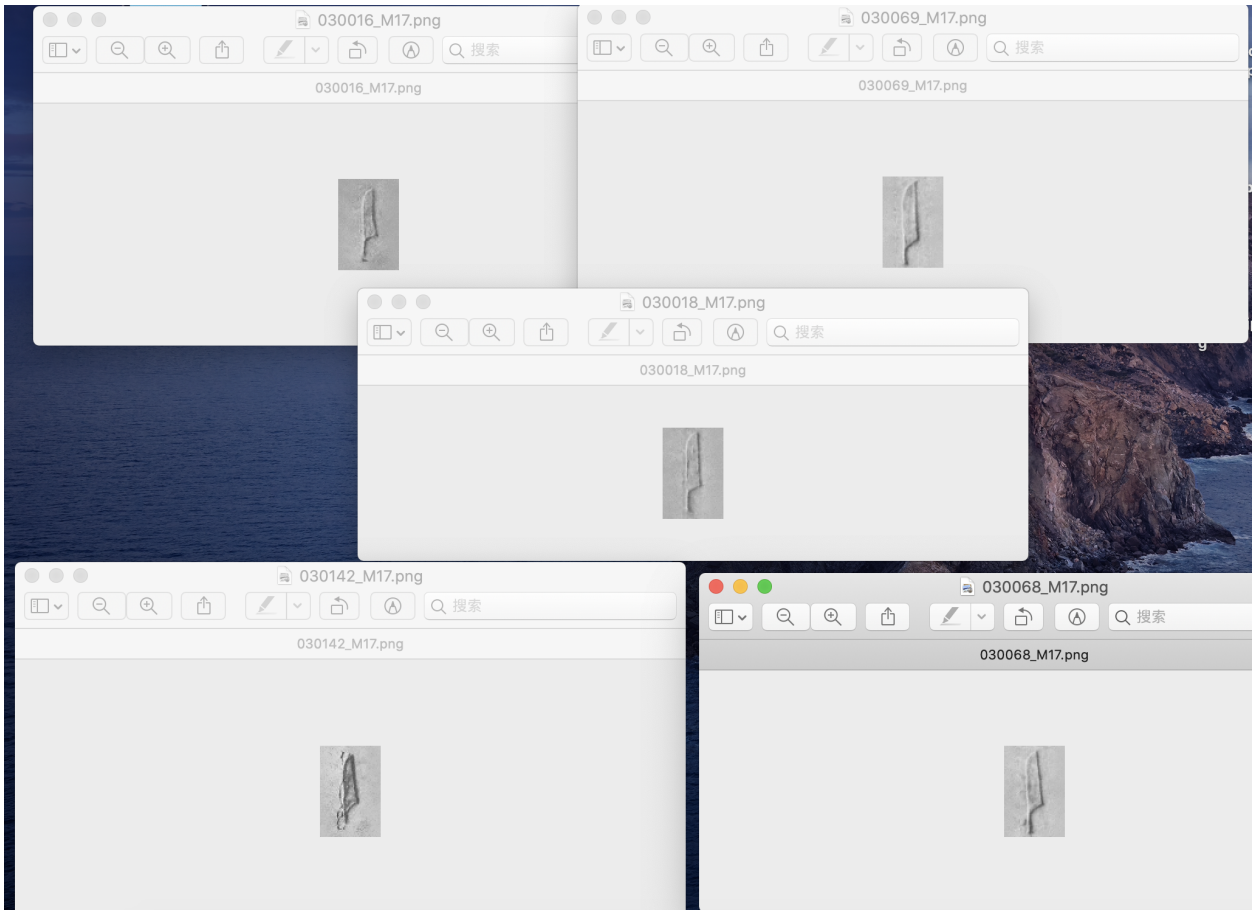
Yuqi	Try to experiment on the contour detect features to detect the cartouche
Kangdong	Try to perform machine learning in r language Find a another possible to do k mean
Sydney	- Contacted the client with questions and potential roadblocks - Corresponded with the group to find alignment in our work - Looked into scribe differentiation - Preprocessing of data
Juliana	- Cartouches research

Xueqing	- Discovering of file naming and classes of data

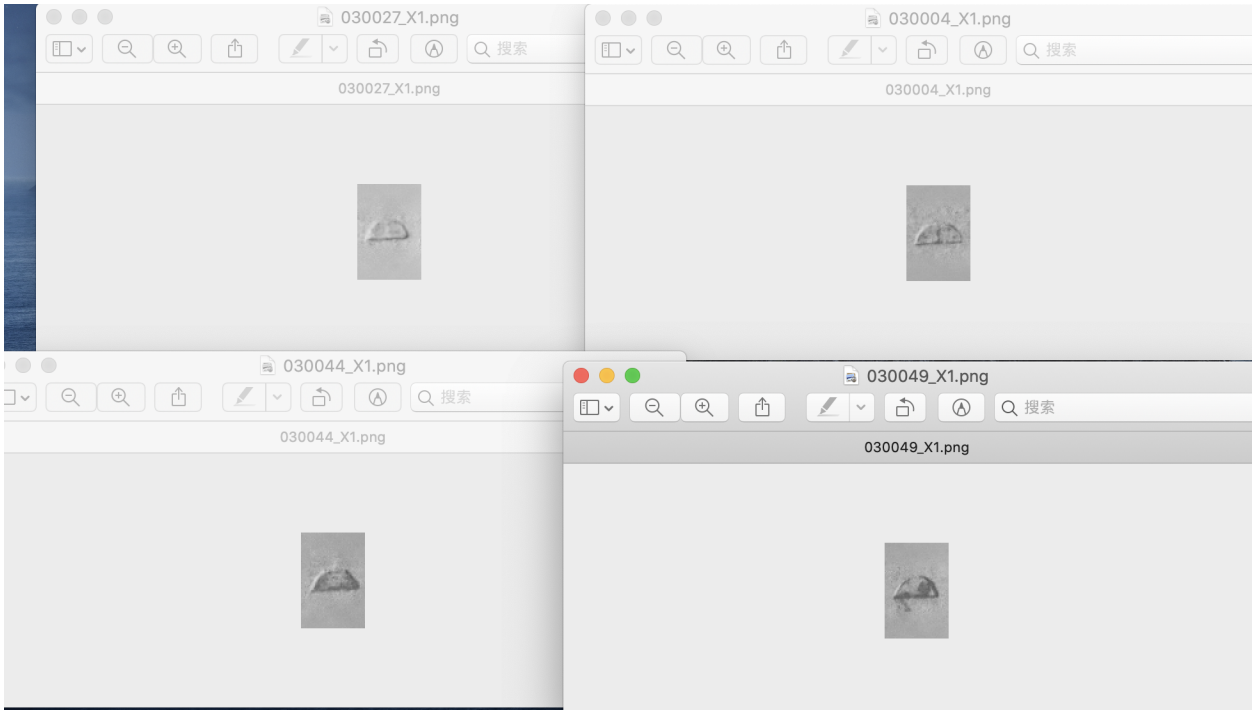
Week Six:

Xueqing:

In the github resources under the folder Manual. I noticed that in every ten folders under processed and raw. There are around 500 pictures, each picture is a drawing of a symbol. There are pictures that share the same label names and those symbols also look the same with just small handwritten differences. See an example of pictures with the same label as M17.They are all symbols of a knife.



Another example is label X1, the symbol looks like a hat. And this symbol was used as alphabetic 't'.Which is intorduced in https://www.ancient-egypt-online.com/hieroglyphics-alphabet.html



Bread Bun

Shaped as a semicircle, used as the alphabetic 't', feminine words or before other determinants or qualifying ideograms

I think we definitely can cluster the data manually, like cluster based on their label and appearances. By clustering these data by their names and appearance we could then add labels to these data with their meanings or interpretations.

Jiayin:
I did some research on the Hieroglyphics classifier accuracy and I found one article that presents a new algorithm to segment and recognize the ancient Egyptian Hieroglyphs from images and produce the corresponding English meaning. The article explained that the Gardiner signs, which closely relate to the data we currently have. Gardiner's sign list is a standard list used to classify the Hieroglyphics symbols such that similar shapes are grouped in the same category but it does not help to recognize the reading direction. Thus, they use an extension of the Gardiner list. The

following screenshot shows the accuracy they manage to receive. “the proposed algorithm segmentation precision is 80% nonetheless, the state-of-the-art of segmentation exactness is 65% The mean of the precision rates of the proposed algorithm is 66.64% and that of the state-of-the-art is 55.27%”.

Image	Hieroglyph Gardiner's code	Proposed work	Zhang, X. Y. et. al [33]
	F12	80%	65%
	G13	73%	59%
	F16	33%	10%
	H06	60%	67%
	F10	65%	49%
	F01	52%	63%
	M03	76%	69%
	G37	87%	61%
	N41	54%	59%

https://link-springer-com.ezaccess.libraries.psu.edu/article/10.1007%2Fs11042-020-09825-2#Sec10

Our group and this article both classify the Egyptian Hieroglyphics based on the Gardiner labels, although we might not translate them to English, we could reference the method and accuracy the article provided and see if we can make some improvements on the accuracy. The average of the classifier accuracy is around 66% so this could be the goal for our project, which is to gain a better classification accuracy.

Sydney and Kangdong:

This week, Sydney and Kangdong worked together to find the distribution of the dataset. We set up a GitHub repository and were pushing files back and forth all week. Kangdon first parsed the files and stripped the .png file names so that only the identifying characters were shown. Sydney then took those characters and created a table with counts for each character to get a better idea of the distribution of our data. This screenshot, to the right, is specifically a count of characters for table 20, which we will reproduce for the rest of the tables too.

	labels1	Freq
1	Aa	1
2	D	43
3	E	12
4	F	11
5	G	54
6	H	3
7	I	18
8	M	49
9	N	49
10	O	11
11	P	4
12	Q	9
13	R	6
14	S	23
15	U	7
16	UNKNOWN	26
17	V	20
18	W	8
19	X	23

Now we are looking at creating testing and training sets based on these distributions and folders. Then by next Thursday, we will have a clustering model for our current dataset.

Our team also received a response email from Dr. Redford, which I will forward to you as well. This email seems to pivot our research quite a bit. Our team would like to discuss this email with you when we meet next week to see what you think, but we are currently thinking that we must improve our knowledge, not only of what Dr. Redford is even saying, but about the following aspects of different Kings.

-lexical choices (and spellings peculiar to this writer)
-favored tenses and non-verbal patterns
-tense vs. aspect in verbal sentences.
presence/absence of group writing
-loan words
-any constructions and patterns which seem to break the rules!
-influence from neighboring languages.

Yuqi	Attempt to do more tweaking on the variables in contour detection Issues found
Kangdong	Complete the k mean cluster in python language Try to improve running time of k mean
Sydney	- Preprocessed the data into train and test classes - Preprocessed files to parse through classes in titles - Looked at distribution of the data
Juliana	- Researched similar article that works on Engyptian Hieroglyphics - Found article working on translating, accuracy around 66% - Request data from the article
Xueqing	-

Week Seven:

The goal of our project is to create clustering and classification models to predict labels through image detection. In this case, our labels will be Gardiner's sign list items.

After meeting with the client, it was clear that this task was very complex. Dr. Redford spoke about the differences in scribe language and verbiage but did not have any datasets to provide to our team to analyze. We then went along to find a GitHub that contained thousands of images all corresponding to a category in the Gardiner list. We decided to take those images and process them to create insights into how hieroglyphics can differ within the correct labels.

During this image gathering process, we also reached out to hieroglyphic researchers at Penn State, who are using similar image processing and computer vision techniques to gain insights into image data, but it turned out that we were using the exact same dataset as them. This was interesting and confirmed our idea that datasets on this topic are limited but valuable.

After exploring the data and discussing ideas as to how we could provide value to our clients, we looked at specific modeling techniques. This past week, Kangdong created a K means clustering model. The k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. In our plan, we have many pictures of Egyptian hieroglyphic letters, and we converted these pictures into a data matrix as seen below.

Because we already know how many letters there are in Egyptian hieroglyphs, we can set the value of k in advance, which perfectly meets the requirements of the kmean method. The implementation of the kmean method requires a lot of work to achieve, first of all, the construction of an unsupervised learning platform. As seen below, Kangdong imported many packages to make this type of model possible.

Kangdong:

```
from keras.preprocessing import image
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.applications.vgg16 import preprocess_input
from PIL import Image as pil_image
from keras.applications.vgg16 import VGG16
from keras.models import Model
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import os, shutil, glob, os.path
import numpy as np
import matplotlib.pyplot as plt
from random import randint
import pandas as pd
import pickle
```

Some of them are for data processing, while others are for matrix transformation and model building. The sklearn package contains the k mean class, which is the centre method to find the similarity between images. The PCA method is used for reducing the dimensions of the matrix, so the running time will be reduced.

```
imdir = r'C:\Users\mjx918\Desktop\hw\final\data'
targetdir = r"C:\Users\mjx918\Desktop\hw\final\res"
```

Above are the lines that work for the readfile into our python environment.

```
number_clusters = 27
```

Kangdon also defined the number of k in advance and assigned the image model to be VGG16.

```
model = VGG16(weights='imagenet', include_top=False)
```

```
# Loop over files and get features
filelist = glob.glob(os.path.join(imdir, '*.png'))
# filelist.sort()
featurelist = []
```

Kangdong then stores all of the images into a list.

```
for i, imagepath in enumerate(filelist):
    print("    Status: %s / %s" %(i, len(filelist)), end="\r")
    img = image.load_img(imagepath, target_size=(224, 224))
    img_data = image.img_to_array(img)
    img_data = np.expand_dims(img_data, axis=0)
    img_data = preprocess_input(img_data)
    features = np.array(model.predict(img_data))
    featurelist.append(features.flatten())
```

For each image in our list, we load into and transform the dimension of the matrix. We also append the feature of the list into a new list that store features.

```
# Clustering
kmeans = KMeans(n_clusters=number_clusters, random_state=0).fit(np.array(featurelist))
```

Then we apply the kmean to my dataset with defined k and random state.

```
try:
    os.makedirs(targetdir)
except OSError:
    pass
```

Then we work to detect whether the output path is correct or not.

```
for i, m in enumerate(kmeans.labels_):
    print("    Copy: %s / %s" %(i, len(kmeans.labels_)), end="\r")
    shutil.copy(filelist[i], targetdir +'\\'+ str(m) + "_" + str(i) + ".jpg")
```

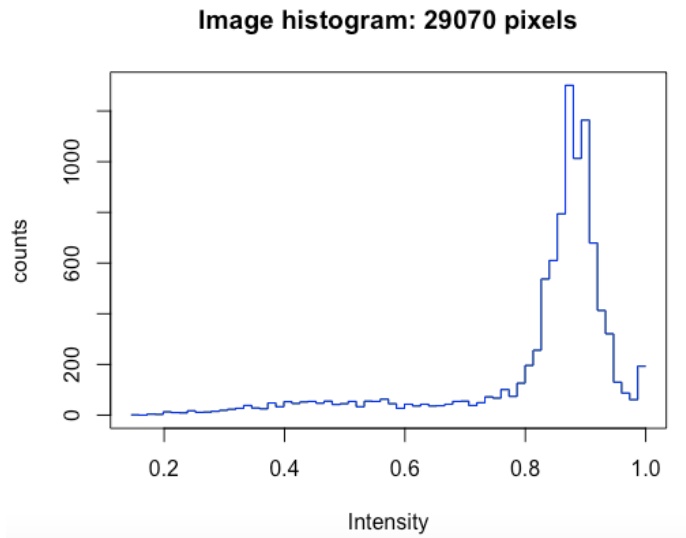
After saving the result into the output path, we will continue to work towards evaluating this model and improving it to be as accurate as possible.

In addition to the clustering techniques that Kangdong performed on that dataset, Sydney worked to classify these images. She began in R where she downloaded the following packages to make this model possible. Then she set her working directory to then be able to read the files that we found in the GitHub.

```
install.packages("BiocManager")
BiocManager::install("EBImage")
install.packages("EBImage")
install.packages("keras")
install.packages("OpenImageR")
install.packages("magick")
library(magick)


setwd('/Users/sydneywehn/ds440-r/DS440/Dataset/data')
```

After specifying columns and a file path, Sydney read in the files with the readImage function in R. She then explored the data and created visualizations to summarize and take a deeper look into what was possible with R image processing. The visualizations showed distributions of pixel data in terms of intensity versus counts.



Sydney then resized and reshaped the images to increase processing speed and accuracy of the model. Sydney then did a row bind with the photos and the train/ test data to be able to train and test the model. After this, she did one hot encoding to be able to fit the model.

```
pics <- c('070390_I10.png', '070241_G7.png', '210360_G43.png')
path = file.path('/Users/sydneywehn/ds440-r/DS440/Dataset/data')

mypic <- list()

for (i in 1:3) {mypic[[i]] <- readImage(pics[i])}

# explore
summary(mypic[[2]])
hist(mypic[[2]])

# Resize
for (i in 1:3) {mypic[[i]] <- shape(mypic[[i]], 28, 28)}

# Reshape
for (i in 1:3) {mypic[[i]] <- array_reshape(mypic[[i]], c(28, 28, 3)
```

Moving forward with this model, Sydney will be evaluating the model with train and test data to determine how accurate Keras was in image classification.

Additionally, Sydney was interested in using Pytorch as another classification technique to process the image data and predict the classification label. She began this technique by importing packages and processing the tensors to numpy arrays.

```
%matplotlib inline

# Python 2/3 compatibility
from __future__ import print_function, division

import itertools
import time

import numpy as np
import matplotlib.pyplot as plt

# Colors from Colorbrewer Paired_12
colors = [[31, 120, 180], [51, 160, 44]]
colors = [(r / 255, g / 255, b / 255) for (r, g, b) in colors]

# functions to show an image
def imshow(img):
    """
    :param img: (PyTorch Tensor)
    """
    # unnormalize
    img = img / 2 + 0.5
    # Convert tensor to numpy array
    npimg = img.numpy()
    # Color channel first -> color channel last
    plt.imshow(np.transpose(npimg, (1, 2, 0)))
```

She then worked to take training and testing samples of the data with subset random samplers to randomly crop, flip, and normalize the images. She is now working to implement a convolutional neural network, with ResNet and CIFAR10, to train the model by next week. She intends on comparing and contrasting her two classification techniques to give the client into which methods are most effective on this dataset.

This week, Jiayin and Xueqing also researched similar projects that utilized kmeans and Agglomerative clustering. They also researched feature extraction methods for image processing using the "HOG" feature descriptor and Canny Edge feature descriptor, which can be seen below.

```
def getHOG(imfn):
    img = cv2.imread(imfn)
    img = cv2.resize(img, (64,64))
    features = ft.hog(img, orientations=6, pixels_per_cell=[8, 8], cells_per_block=[2, 2], visualize=True)
    return features[1]
    plt.imshow(features[1], cmap=plt.cm.gray)
    plt.show()
```

They also worked with the Canny Edge feature descriptor

```
def getCanny(imfn):
    img = cv2.imread(imfn)
    im = cv2.resize(img, (64, 64))
    im = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    features = ft.canny(im)
    return features
    # plt.imshow(edges1, cmap=plt.cm.gray)
    # plt.show()
```

When Jiayin and Xueqing first planned to do the clustering, they researched how to process images with different potential features. One of the ways that they found was Hog, which extracts the information of the edges magnitude as well as the orientation of the edges. They worked to generate the hog with 1 image first and then wished to combine 3 different feature descriptors to do the clusters for all the images. Then they looked into how they could cluster

images based on symbols' features or similarities. After combining 3 different feature descriptors together, they plan to do two clustering methods, which include k means and agglomerativeClustering, using the data that they got from the feature descriptors.

Moving forward, Kangdong will be working to evaluate and improve the accuracy of their classification and clustering models. We would also like to have a strong comparison of all models and methods used in our research to provide the client with as much evidence as to how we can model hieroglyphics as possible. We would also like to give a strong comparison between clustering with and without feature extraction and classification. We intend on pushing all of our deliverables to our GitHub repository so that our client will be able to translate our compiled work into something of value for future research.

Just as a side note, Yuqi has not corresponded or worked with us to progress this project. We are unsure as to how we should provide support to her, but are working to be as collaborative as possible to overcome the lack of work from our team member.

Below are some resources that the group used this week as reference.

- https://www.thepythoncode.com/article/hog-feature-extraction-in-python#:~:text=The%20Histogram%20of%20Oriented%20Gradients,image%20or%20region%20of%20interest.
- https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/
- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html
- https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123

Yuqi	Found couture detection is not ideal for the purpose of detect the cartouche Previous method abandoned and seeking for new methods
Kangdong	- K-means clustering
Sydney	- Began first neural network - Used R to visualize pixels and understand which features we would use - Explored ResNet and CIFAR10

Juliana & Xueqing	- Clustering techniques - Jiayin:Canny Edge feature - Xueqing: Hog Edge feature
All	- Reevaluated our goals - Redirected progress after our meeting

Week Eight:

This week Jiayin and Xueqing finished their work on clustering. First they used three feature descriptor methods (HOG, Canny, and inceptionv3) to extract the information of every image. They have stored the information of three feature descriptors for every picture into a txt file called pca. Below is a part of the file.

pca.txt


```
./20/200145_I9.png_{"hog": [-98.25000762939453, -21.06928825378418, 11.178654670715332,
-49.9773063659668, 2.951134443283081, 18.705062866210938, -18.5618953704834,
13.15726089477539, 49.45524978637695, 17.12569236755371, -34.2385139465332,
-23.50819206237793, 24.25905418395996, 38.20875549316406, -2.4119107723236084,
-4.384189128875732, 19.257606506347656, 9.57158374786377, 5.404262542724609,
-8.770486831665039, 24.053627014160156, -7.985709190368652, 26.301218032836914,
-4.396979808807373, -2.1471118927001953, -7.47017765045166, 5.7586445808410645,
6.716533184051514, 9.331729888916016, -5.094630718231201, -11.71792984008789,
-5.898746013641357, -3.064401626586914, 13.899150848388672, -4.702086925506592,
-10.73022174835205, 6.521754741668701, 12.500710487365723, -1.9840760231018066,
-12.20256233215332, 0.9172377586364746, -15.761320114135742, -5.256949424743652,
2.308288812637329, 9.031079292297363, -1.1264429092407227, 5.1713762283325195,
1.0423699617385864, 3.5054805278778076, -3.6583354473114014, -13.644416809082031,
3.1871654987335205, 0.9816334843635559, -2.476550579071045, 2.0313196182250977,
-4.662537097930908, 5.017992973327637, -5.611405372619629, -2.552428722381592,
4.0138349533081055, -2.3033785820007324, -10.293743133544922, -0.8031153082847595,
0.31321409344673157], "canny": [-2.937309741973877, -0.2804293632507324,
-1.109710454940796, -2.33583402633667, 1.1853609085083008, -2.387486696243286,
4.151598930358887, 0.45279794931411743, -2.1658530235290527, 0.30135756731033325,
0.7048139572143555, -0.35982874035835266, 1.540076494216919, -1.5619351863861084,
-2.33447265625, -1.045156717300415, 0.49639227986335754, 0.4186582565307617,
-1.8185040950775146, 0.7049290537834167, -0.040794339030981064, 0.9272582530975342,
0.47546130418777466, 0.5046623945236206, -0.6918630599975586, -0.8560479283332825,
0.985646665096283, 1.3884780406951904, -0.5987682938575745, -0.7094223499298096,
0.40367773175239563, 0.37510138750076294, -0.3601122498512268, 1.1522817611694336,
-0.012591770850121975, 0.1768188774585724, -0.2844025194644928, 0.15152105689048767,
0.3246033787727356, -0.17178738117218018, -0.43213334679603577, -0.4885208308696747,
0.31814712285995483, 0.3838008940219879, 0.7428610324859619, -0.1855955421924591,
0.8626838326454163, 1.1070047616958618, -0.5550492405891418, 0.17886953055858612,
0.25068116188049316, 0.0029023943934589624, -0.6772937774658203, -0.04445361718535423,
```

Later they did two clustering methods, Kmeans and AgglomerativeClustering by using the information in pca.txt.

```python
def cluster():
    f = open('pca.txt')
    fn_list = []
    emb_list = []
    for line in f:
        res = line.strip().split('_')
        fn = "_".join(res[:-1])
        data = json.loads(res[-1])
        emb_list.append(data['hog'] + data['canny'] + data['inception'])
        fn_list.append(fn)

    emb = np.array(emb_list, dtype=np.float)
    from sklearn.cluster import KMeans
    clf = KMeans(n_clusters=100)
    y = clf.fit_predict(emb)
    fw = open('kmeans.txt', 'w')
    for i in range(len(fn_list)):
        fw.write(fn_list[i] + ':' + str(y[i]) + '\n')
        # print(y[i], fn_list[i])
    fw.close()

    from sklearn.cluster import AgglomerativeClustering
    y = AgglomerativeClustering(linkage='ward', n_clusters=100).fit_predict(emb)
    fw = open('agg_cluster.txt', 'w')
    for i in range(len(fn_list)):
        fw.write(fn_list[i] + ':' + str(y[i]) + '\n')
        # print(y[i], fn_list[i])
    fw.close()
    # for i in range(len(fn_list)):
    #     print(y[i], fn_list[i])
```

Below is the result of Kmeans and AgglomerativeClustering.

kmeans.txt

```
./20/200145_I9.png:21
./20/200301_D21.png:18
./20/200105_M17.png:8
./20/200242_U1.png:32
./20/200077_G43.png:86
./20/200274_N35.png:49
./20/200315_UNKNOWN.png:67
./20/200373_D21.png:94
./20/200306_G17.png:24
./20/200123_Q3.png:68
./20/200230_R8.png:27
./20/200218_E34.png:52
./20/200234_G43.png:57
./20/200384_UNKNOWN.png:70
./20/200080_V28.png:29
./20/200083_G43.png:51
./20/200129_F34.png:96
./20/200081_UNKNOWN.png:67
./20/200305_N35.png:77
./20/200158_D54.png:59
./20/200228_Z1.png:17
./20/200360_M17.png:4
./20/200270_I9.png:98
./20/200063_N35.png:5
./20/200318_G43.png:57
./20/200304_Q3.png:89
./20/200283_P8.png:67
./20/200263_I10.png:22
./20/200275_W24.png:88
./20/200292_V31.png:64
```

agg_cluster.txt

```
./20/200145_I9.png:15
./20/200301_D21.png:19
./20/200105_M17.png:4
./20/200242_U1.png:57
./20/200077_G43.png:74
./20/200274_N35.png:9
./20/200315_UNKNOWN.png:26
./20/200373_D21.png:83
./20/200306_G17.png:6
./20/200123_Q3.png:7
./20/200230_R8.png:54
./20/200218_E34.png:49
./20/200234_G43.png:46
./20/200384_UNKNOWN.png:13
./20/200080_V28.png:13
./20/200083_G43.png:36
./20/200129_F34.png:11
./20/200081_UNKNOWN.png:0
./20/200305_N35.png:9
./20/200158_D54.png:87
./20/200228_Z1.png:53
./20/200360_M17.png:24
./20/200270_I9.png:90
./20/200063_N35.png:16
./20/200318_G43.png:46
./20/200304_Q3.png:22
./20/200283_P8.png:98
./20/200263_I10.png:68
./20/200275_W24.png:70
./20/200292_V31.png:76
```

During Tuesday's meeting, the professor has pointed out some advice for us to adjust. First is about the cluster number. Jiayin and Xueqing have randomly set the cluster number to 100, However, they need to try more times with different numbers of the cluster in order to give a meaningful reason for choosing a particular number. The professor has given a suggestion with 27 clusters since there are 27 letters in egyptian alphabet. Next week, they will try with different cluster numbers and finally they will use elbow method to find the best cluster numbers.

This past week Sydney worked with the GitHub code and explored parameters to tune the number of batches. She has been documenting the changes to the original code and is continuing to work with other models to see if she can improve the accuracy achieved in the Logistic Regression classifier using Grid Search cross validation.

Kangdong: Because I found that if my users use my python code to run on pycharm, they will be very confused. Because they didn't know which part of the code was performing which function, so I chose to transfer my program to jupyter notebook.

Set the model, but don't suggest user to change it.

```
[ ] image.LOAD_TRUNCATED_IMAGES = True
    model = VGG16(weights='imagenet', include_top=False)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/
58892288/58889256 [==============================] - 1s 0us/step
```

The input and output path, user can change it freely

```
[ ] imdir = '/content/data'
    targetdir = r"/content/result"
```

Extract the feature to the list

```
[ ] filelist = glob.glob(os.path.join(imdir, '*.png'))
    # filelist.sort()
    featurelist = []
    for i, imagepath in enumerate(filelist):
        print("    Status: %s / %s" %(i, len(filelist)), end="\r")
        img = image.load_img(imagepath, target_size=(224, 224))
        img_data = image.img_to_array(img)
        img_data = np.expand_dims(img_data, axis=0)
        img_data = preprocess_input(img_data)
        features = np.array(model.predict(img_data))
        featurelist.append(features.flatten())
```

Yuqi	Continue searching on new methods
Kangdong	Try to transfer code to jupyter notebook.
Sydney	- Worked with classification GitHub code - Researched cross validation techniques that could be of use here
Juliana & Xueqing	- Cluster the data with 100 as number of cluster and with feature descriptors - Jiayin: Agglomerative Clustering - Xueqing: Kmeans Clustering

Week Nine:

Sydney:

- This week I continued to run files from the GitHub repository. I ran into many errors with local machine versions of packages and tried to look at alternative routes to classification. I ran into the following errors with the two .py files.

Classify.py:

```
    prep_data = spawn.get_preparation_data(process_obj._name)
  File "/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/multipro
essing/spawn.py", line 154, in get_preparation_data
    _check_not_importing_main()
  File "/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/multipro
essing/spawn.py", line 134, in _check_not_importing_main
    raise RuntimeError('''
untimeError:
        An attempt has been made to start a new process before the
        current process has finished its bootstrapping phase.

        This probably means that you are not using fork to start your
        child processes and you have forgotten to use the proper idiom
        in the main module:

            if __name__ == '__main__':
                freeze_support()
                ...

        The "freeze_support()" line can be omitted if the program
        is not going to be frozen to produce an executable.
```

Train.py:

```
o enable them in other operations, rebuild tensorflow with the appropriate comp
iler flags.
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-pac
kages/keras/engine/base_layer.py", line 2719, in __setattr__
    super(tf.__internal__.tracking.AutoTrackable, self).__setattr__(name, value)
AttributeError: can't set attribute

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "train.py", line 49, in <module>
    featureExtractor = FeatureExtractor()
  File "/Users/sydneywehn/Capstone/glyphreader/src/featureExtractor.py", line 19
  in __init__
    self.model.layers[-1].outbound_nodes = []
  File "/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-pac
kages/keras/engine/base_layer.py", line 2721, in __setattr__
    raise AttributeError(
AttributeError: Can't set the attribute "outbound_nodes", likely because it conf
licts with an existing read-only @property of the object. Please choose a differ
ent name.
Sydneys-MacBook-Air:src sydneywehn$ python3 train.py
```

I believe that the best route is to create my own models instead of use the ones from GitHub. Not only will I then explore the data pipeline architecture, but I will then not be limited to logistic regression. I have also researched transfer learning and how I can implement that in our project.

Xueqing Zhang:

I have tried different cluster numbers for Kmeans. I have tried 27 and 50. Screenshots in below.

kmeans_27.txt

```
./20/200145_I9.png:9
./20/200301_D21.png:14
./20/200105_M17.png:6
./20/200242_U1.png:24
./20/200077_G43.png:3
./20/200274_N35.png:11
./20/200315_UNKNOWN.png:1
./20/200373_D21.png:14
./20/200306_G17.png:10
./20/200123_Q3.png:2
./20/200230_R8.png:5
./20/200218_E34.png:15
./20/200234_G43.png:25
./20/200384_UNKNOWN.png:1
./20/200080_V28.png:19
./20/200083_G43.png:25
./20/200129_F34.png:24
./20/200081_UNKNOWN.png:19
./20/200305_N35.png:11
./20/200158_D54.png:26
./20/200228_Z1.png:1
./20/200360_M17.png:5
./20/200270_I9.png:9
./20/200063_N35.png:11
./20/200318_G43.png:25
./20/200304_Q3.png:21
./20/200283_P8.png:6
./20/200263_I10.png:22
./20/200275_W24.png:24
./20/200292_V31.png:13
```

kmeans_50.txt

```
./20/200145_I9.png:19
./20/200301_D21.png:22
./20/200105_M17.png:14
./20/200242_U1.png:20
./20/200077_G43.png:12
./20/200274_N35.png:18
./20/200315_UNKNOWN.png:23
./20/200373_D21.png:41
./20/200306_G17.png:7
./20/200123_Q3.png:6
./20/200230_R8.png:45
./20/200218_E34.png:37
./20/200234_G43.png:17
./20/200384_UNKNOWN.png:10
./20/200080_V28.png:16
./20/200083_G43.png:17
./20/200129_F34.png:29
./20/200081_UNKNOWN.png:23
./20/200305_N35.png:18
./20/200158_D54.png:47
./20/200228_Z1.png:10
./20/200360_M17.png:9
./20/200270_I9.png:35
./20/200063_N35.png:42
./20/200318_G43.png:17
./20/200304_Q3.png:27
./20/200283_P8.png:14
./20/200263_I10.png:15
./20/200275_W24.png:26
./20/200292_V31.png:11
```

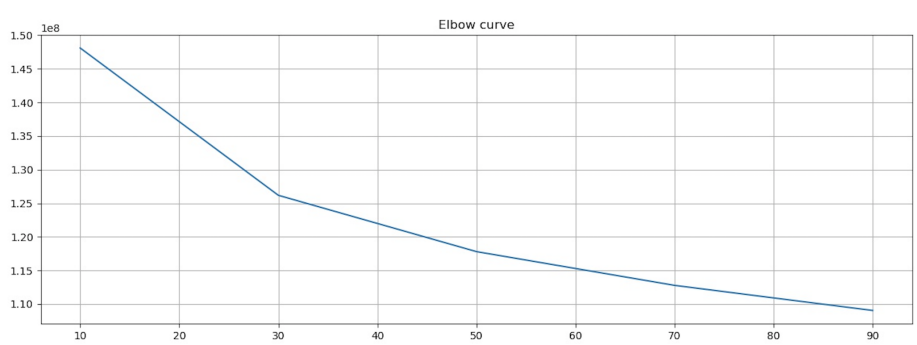
I have also used the elbow method to find the best cluster number which is 30. And the result of cluster 30 I have listed below.

kmeans_30.txt

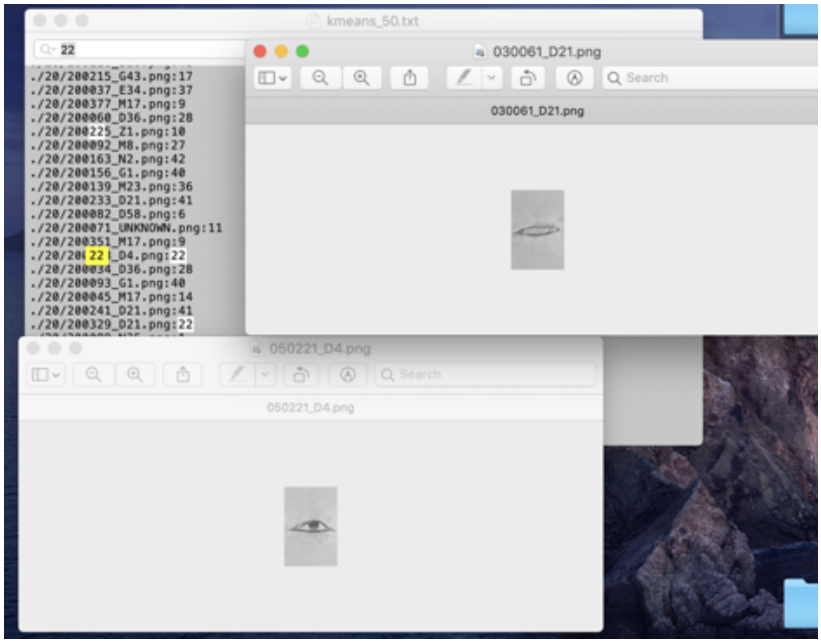
```
./20/200145_I9.png:26
./20/200301_D21.png:3
./20/200105_M17.png:16
./20/200242_U1.png:6
./20/200077_G43.png:13
./20/200274_N35.png:28
./20/200315_UNKNOWN.png:17
./20/200373_D21.png:3
./20/200306_G17.png:2
./20/200123_Q3.png:23
./20/200230_R8.png:11
./20/200218_E34.png:7
./20/200234_G43.png:13
./20/200384_UNKNOWN.png:17
./20/200080_V28.png:1
./20/200083_G43.png:13
./20/200129_F34.png:6
./20/200081_UNKNOWN.png:17
./20/200305_N35.png:28
./20/200158_D54.png:5
./20/200228_Z1.png:17
./20/200360_M17.png:21
./20/200270_I9.png:26
./20/200063_N35.png:28
./20/200318_G43.png:13
./20/200304_Q3.png:5
./20/200283_P8.png:16
./20/200263_I10.png:14
./20/200275_W24.png:6
./20/200292_V31.png:25
```

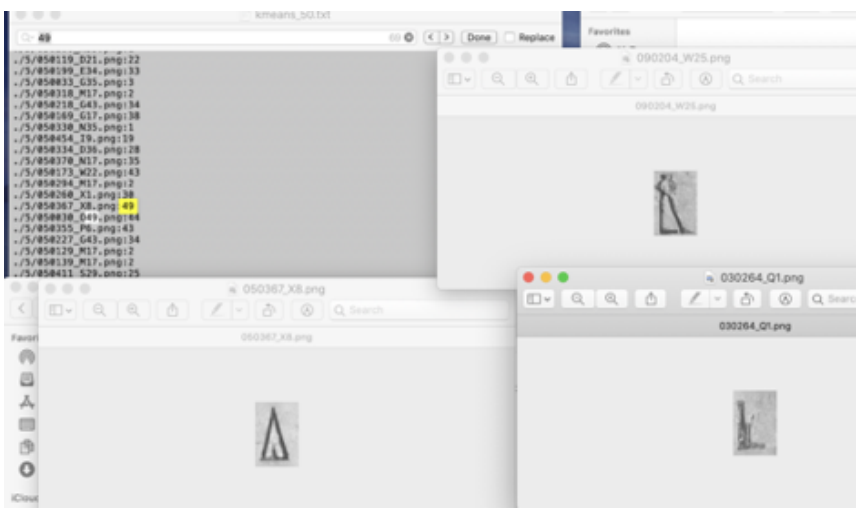
```python
distorsions = []
# for n_clusters in [10, 30, 50, 70, 90]:
for n_clusters in [10, 30, 50, 70, 90]:
    clf = KMeans(n_clusters=n_clusters)
    clf.fit_predict(emb)
    distorsions.append(clf.inertia_)
fig = plt.figure(figsize=(15, 5))
plt.plot([10, 30, 50, 70, 90], distorsions)
plt.grid(True)
plt.title('Elbow curve')
plt.savefig("elbow_kmeans.jpg")
```

WechatIMG77.jpeg



I have encountered an issue that no matter what the cluster number is, 27,50,100 or 30. There are different symbols that fall into the same cluster. See two examples I got. The D21 and D4 fall into the same cluster 22 when there are 50 clusters. D21 and D4 actually look pretty similar. They both look like an eye, though D21 looks like an eye without an eyebrow.





Jiayin(Juliana) Hu:
I have tested on different cluster numbers for Agglomerative Clustering. I have tried 27 and 50. Below are the screenshots.

agg_cluster_27.txt

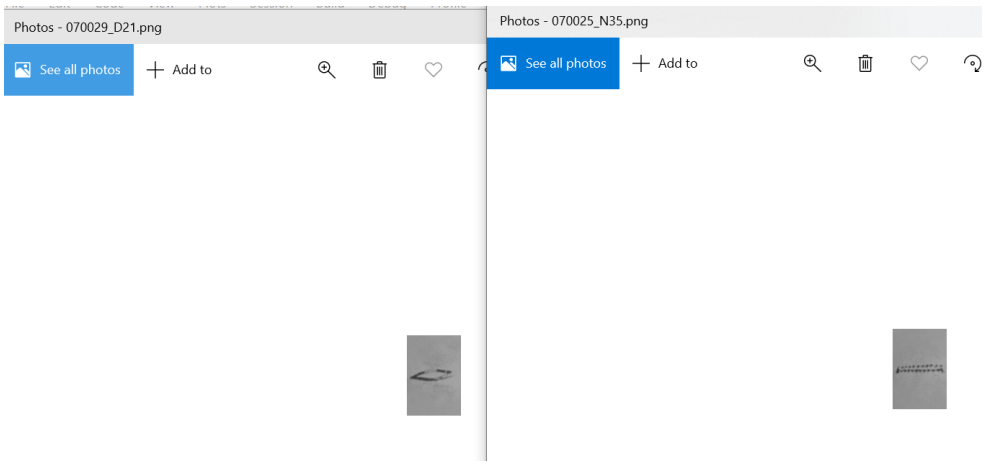
```
./20/200145_I9.png:17
./20/200301_D21.png:19
./20/200105_M17.png:9
./20/200242_U1.png:1
./20/200077_G43.png:13
./20/200274_N35.png:0
./20/200315_UNKNOWN.png:12
./20/200373_D21.png:19
./20/200306_G17.png:3
./20/200123_Q3.png:6
./20/200230_R8.png:20
./20/200218_E34.png:15
./20/200234_G43.png:13
./20/200384_UNKNOWN.png:4
./20/200080_V28.png:4
./20/200083_G43.png:13
./20/200129_F34.png:6
./20/200081_UNKNOWN.png:12
./20/200305_N35.png:0
./20/200158_D54.png:5
./20/200228_Z1.png:26
./20/200360_M17.png:20
./20/200270_I9.png:17
./20/200063_N35.png:0
./20/200318_G43.png:13
./20/200304_Q3.png:10
./20/200283_P8.png:9
./20/200263_I10.png:16
./20/200275_W24.png:1
./20/200292_V31.png:22
```

agg_cluster_50.txt

```
./20/200145_I9.png:17
./20/200301_D21.png:40
./20/200105_M17.png:18
./20/200242_U1.png:28
./20/200077_G43.png:46
./20/200274_N35.png:14
./20/200315_UNKNOWN.png:9
./20/200373_D21.png:20
./20/200306_G17.png:29
./20/200123_Q3.png:2
./20/200230_R8.png:4
./20/200218_E34.png:32
./20/200234_G43.png:8
./20/200384_UNKNOWN.png:13
./20/200080_V28.png:13
./20/200083_G43.png:46
./20/200129_F34.png:48
./20/200081_UNKNOWN.png:0
./20/200305_N35.png:14
./20/200158_D54.png:42
./20/200228_Z1.png:26
./20/200360_M17.png:4
./20/200270_I9.png:17
./20/200063_N35.png:21
./20/200318_G43.png:8
./20/200304_Q3.png:1
./20/200283_P8.png:18
./20/200263_I10.png:33
./20/200275_W24.png:16
./20/200292_V31.png:23
```

As Xueqing suggested earlier, the same thing happened to Agglomerative Clustering. Although we tested with the number of Egyptian alphabet, it seems that still pictures from other different alphabets would be clustering into the same group. From the elbow method, we discovered that 30 is the best number of clusters, so I think 27 is a good place to take more investigation, since it is also the number of Egyptian alphabets. As the following screenshot suggests, N35 and D36 are both in the cluster 0, so I opened the images and discovered that they have similar shape but different amounts of white space in between the two lines. Since our images focused on image processing instead of gardiner labeling, I think such things happen because the similarity of images.

10	./7/070030_N35.png	0
11	./7/070148_N35.png	0
12	./7/070334_N35.png	0
13	./7/070205_D36.png	0
14	./7/070444_N35.png	0
15	./7/070026_V13.png	0
16	./7/070133_N35.png	0
17	./7/070123_N35.png	0
18	./7/070245_N35.png	0
19	./7/070080_N35.png	0
20	./7/070237_N35.png	0
21	./7/070074_N35.png	0
22	./7/070306_D36.png	0
23	./7/070226_N35.png	0



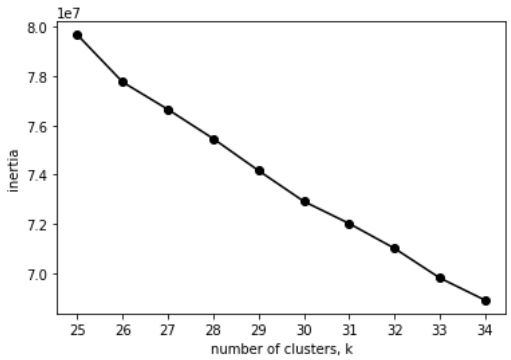
So far is what I found using the Egyptian data and since we need to use the model in another data eventually, I also started looking for the taiji images database, here is the data link I found and I wish it will be useful for our future works. I have not found a time to discuss with the teams yet, but I will provide the link and continue to look if there is more useful data.
https://github.com/numediart/UMONS-TAICHI

Kangdong: I am trying to realize the visualization of my k-means model on my jupyternotebook, and try to draw three-dimensional visualization pictures.

generate a graph. User can change the range freely.

```
[ ] ks = range(25, 35)
    inertias = []
    for k in ks:
        # Create a KMeans instance with k clusters: model
        kmeans = KMeans(n_clusters=k, random_state=0)
        kmeans.fit(pca)
        # Append the inertia to the list of inertias
        inertias.append(kmeans.inertia_)

    plt.plot(ks, inertias, '-o', color='black')
    plt.xlabel('number of clusters, k')
    plt.ylabel('inertia')
    plt.xticks(ks)
    plt.show()
```



Yuqi	Found possible method called template matching that can be used to achieve detection of cartouche
Kangdong	Jupyternotebook and visualization.
Sydney	- GitHub code errors and debugging - Research on new data pipelines and classification transfer learning models
Juliana & Xueqing	- Test two more different number of clustering, which are 27 and 50 - Used elbow method to find that 30 is the best number of clustering - Jiayin: Agglomerative Clustering - Xueqing: K-means Clustering

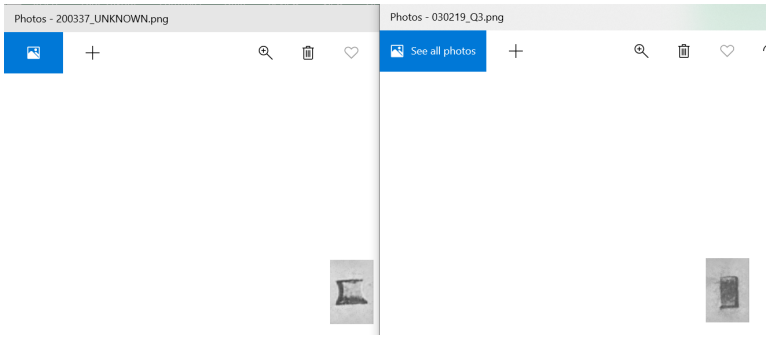
Week Ten:

Jiayin Hu:
After doing the elbow method, we realized that the best parameter for the number of clusters is 30, and we did cluster on 27,50, and 100. Thus, I choose to investigate the 27 clusters result deeply since I think it could be more meaningful without too much cluster error itself. After manually calculating the percent of the top 2 labels in each cluster, I examined two main different patterns. Below are the screenshots of the result.

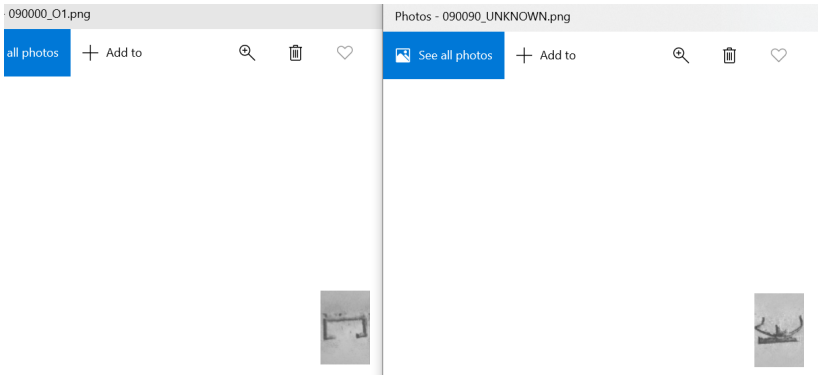
Cluster	Total Image	Top1	Percent	Top2	Percent
0	864	N35	51.85	Unknown	5.32
1	204	O50	48.04	W24	14.71
2	200	S29	100		
3	230	G17	53.91	G1	14.35
4	55	V28	65.45	S34	14.55
5	54	D54	22.22	G7/Unknown	18.52
6	206	Q3	33.98	Unknown	17.96
7	208	X1	85.1	E34	12.98
8	91	G17	76.92	G35	8.79
9	212	M17	81.13	U33/M18	4.25
10	172	Unknown	14.53	O1	11.63
11	63	D58	52.38	Q1	26.98
12	288	M17	28.13	S29	22.57
13	142	G43	100		
14	33	M23	100		
15	89	E34	100		
16	47	I10	87.23	Unknown	6.38
17	154	I9	92.21	F18	3.25
18	55	D35	100		
19	251	D21	61.35	D4	14.34
20	110	R8	59.09	M17	14.55
21	97	M17	96.91	Unknown	2.06
22	150	V31	87.33	F13/V30	5.33
23	65	V13	80	N37/Z7	6.15
24	51	G43	100		
25	69	G25	39.13	G39	23.19
26	50	Z1	90	W24	4

The first pattern is where the top 1 label takes the majority of the cluster and the top 2 only a small percent. There are 15 clusters that fall into this pattern and I think it means those clusters are clustered nicely and the top 1 in those clusters does not repeat each other. The second pattern is where the top 1 label and top 2 label does not show a great difference such as cluster 6 shows.

So I investigated the images manually and found out all of them share similar characteristics.



The screenshot above shows the top 2 images in cluster 6, we can see that two images do have a similar shape as one seems to be a rectangle and the other seems close to being a square, and they both seem to be solid(colored in the middle). Also in cluster 10, the top 1 label looks like the upside-down version of the top 2 labels. It is kind of understandable that the cluster made these mistakes because it is based on the feature of the images. In addition, each unknown could be different from the others, even if they are in the same folder.



Overall, I think the cluster had done its job in properly clustering the images and could be applied to the later dataset.

YOGA:
https://www.kaggle.com/niharika41298/yoga-poses-dataset
Using cluster method to cluster the pose into the given five categories. Same cluster algorithm, see the performance and limitation. In this case, since images are colored, the result could be really different. The best number of clusters might not continue to be 27.

A lot of research is all based on the yoga-82 dataset. Requested and downloaded the dataset, convert it to a zip file, and upload the data to the group folder. However, I found each image is indicated as links instead of images themselves. If we decide to use this dataset, we have to go through every link and download the images ourselves. I also downloaded the dataset from the first link and upload that to the folder as well.

Xueqing Zhang

From last week, I have tested with different cluster number for 50,100, 27 and 30. Using the elbow method, the best cluster number was 30. However, after talking with professor, I decided to use cluster number 27 for statistics summary. I have summed up the total picture number in each clusters. And also the two most frequent pictures with their percentage in the cluster. Below is the screenshot of the kmeans result.

	A	B	C	D	E	F	G	H
1	CLUSTER	NUMBER OF PICTURES	TOP1 NAME	PERCENT	TOP2 NAME	PERCENT		
2	0	123	D35	0.46	D36	0.293		
3	1	170	Z1	0.288	S29	0.129		
4	2	77	D58	0.455	Q1	0.064		
5	3	81	G43	0.975	U7	0.024		
6	4	176	S29	0.994	P8	0.006		
7	5	189	M17	0.423	R8	0.296		
8	6	201	M17	0.721	T22	0.299		
9	7	122	G25	0.221	G39	0.164		
10	8	133	G17	0.624	G35	0.12		
11	9	150	I9	0.927	F18	0.033		
12	10	174	G17	0.626	G35	0.086		
13	11	419	N35	0.807	UNKNOWN	0.041		
14	12	34	M23	0.97	M26	0.03		
15	13	136	V31	0.88	V30	0.04		
16	14	186	D21	0.77	D4	0.15		
17	15	108	E34	0.99	UNKNOWN	0.01		
18	16	104	V13	0.62	UNKNOWN	0.07		
19	17	434	N35	0.25	D21	0.09		
20	18	133	M17	0.87	M19	0.03		
21	19	139	S29	0.46	V28	0.22		
22	20	225	X1	0.92	D54	0.02		
23	21	90	O1	0.22	D28	0.16		
24	22	46	I10	0.89	UNKNOWN	0.07		
25	23	42	W25	0.26	S34	0.17		
26	24	203	O50	0.48	W24	0.16		
27	25	113	G43	0.96	G17,G7	0.02		
28	26	202	Q3	0.34	UNKNOWN	0.16		
29								

Viewing the dataset of yoga pictures, I think if we are doing clustering with yoga pictures could be different from our Egyptian symbols. Instead of using feature descriptor as we did previously, we might use some code that can detect on the human's head or feet. When the head is on the ground it may be some particular pose. I will keep doing research next week to find ways to do clustering with yoga pose.

Kangdong Yuan:

In practical applications, I need to find a more efficient way to train my model. Through my exploration of the matrix, I found that the matrix of each image has 224 numbers. If I use such a large matrix to train my k-means model, the cpu will face tremendous pressure and even cause the temperature to be too high and unstable. So I used the PCA method to reduce the size of the matrix. PCA is a mathematical method to convert data points in high dimensionality into data

points in low dimensionality. I use the PCA library of python to call the PCA function. When I reduced the matrix dimension of the training data to 10, the training speed of my k-means model dropped from the half hour to 20 seconds.

```
[ ] n_digits=10

    pca = PCA(n_components=n_digits).fit_transform(featurelist)
```

Yuqi	Doing experiment on using the template matching method
Kangdong	- K-means clustering - Visualization - Improved efficiency of model training
Sydney	- Began first neural network - Used R to visualize pixels and understand which features we would use - Explored ResNet and CIFAR10
Juliana & Xueqing	- Analyze each cluster and calculate the percentage of top 1 and top 2 appeared images in each cluster - Jiayin: Agglomerative Clustering - Xueqing: K-means Clustering - Researched about yoga poses - Jiayin: Researched for dataset and downloaded the most fitting one for later use - Xueqing: Clustering method
All	- Reevaluated our goals - Redirected progress after our meeting

Week Eleven:

This week our group met with Prof Mitra on Thursday and discussed how we were going to wrap up our project in our final three weeks. Sydney and Prof Mitra discussed what her

classes in her resnet34 and alexnet models represented. She took a further look and found that the classes were combined classes from the Gardiner Sign sheet. She then revised her classes to represent the more specific classes (alphabet letters in the hieroglyphics dataset) and then continued to work with parameters and mobilenet, shufflenet, and vgg16 to see how these models classified differently than the two previous classification models. She also documented normalization and optimization techniques could be used in addition to these models to improve accuracy and model predictability.

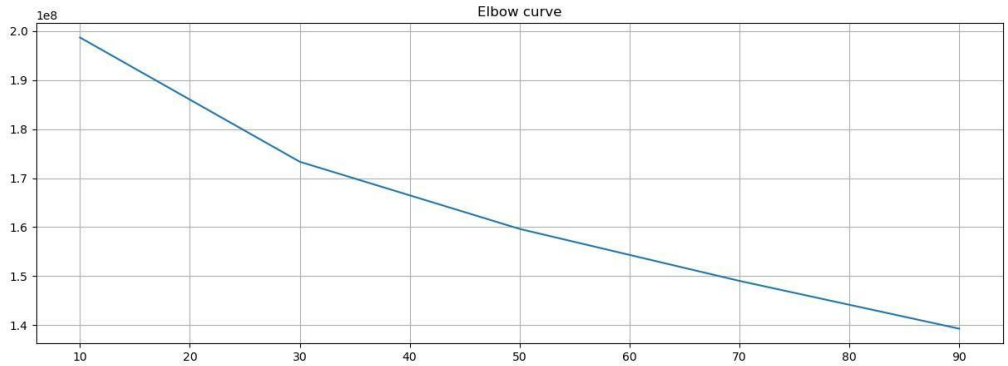
Xueqing and Jiayin(Juliana) evaluated their clusters based on the number of labels in each cluster, and found the top 2 labels and calculated their percentages. As a result, Jiayin found out that the algormative clustering she worked on had 15 out 27 clusters containing more than 80% of the top 1 label, which means the cluster does include 1 label instead of mixing up the labels. The rest clusters that have similar percentages for top 1 and top 2 actually have similar pictures. Prof Mitra and Jiayin discussed the similarities of the pictures and decided that the current cluster algorithm is good for the current data set and the number of clusters of 27 is also a good choice for the current data. The further analysis will be necessary to see if the clustering method will work good on other data like the yoga ones.

Xueqing calculated the Gini Impurity for k means that is 4.9754. And Xueqing has finished organizing the final excel with pictures included. Screenshot in below.

	A	B	C	D	E	F	G	H	I	J
1	CLUSTER	NUMBER OF PICTURES	TOP1 NAME	PERCENT	Symbol Display	TOP2 NAME	PERCENT	Symbol Display		
2	0	123	D35	0.46		D36	0.293			
3	1	170	Z1	0.288		S29	0.129			
4	2	77	D58	0.455		Q1	0.064			
5	3	81	G43	0.975		U7	0.024			
6	4	176	S29	0.994		P8	0.006			
7	5	189	M17	0.423		R8	0.296			
8	6	201	M17	0.721		T22	0.299			
9	7	122	G25	0.221		G39	0.164			

12	10	174	G17	0.626	G35	0.086	
13	11	419	N35	0.807	UNKNOWN	0.041	
14	12	34	M23	0.97	M26	0.03	
15	13	136	V31	0.88	V30	0.04	
16	14	186	D21	0.77	D4	0.15	
17	15	108	E34	0.99	UNKNOWN	0.01	
18	16	104	V13	0.62	UNKNOWN	0.07	
19	17	434	N35	0.25	D21	0.09	
20	18	133	M17	0.87	M18	0.03	
21	19	139	S29	0.46	V28	0.22	
22	20	225	X1	0.92	D54	0.02	
23	21	90	O1	0.22	D28	0.16	
24	22	46	I10	0.89	UNKNOWN	0.07	
25	23	42	W25	0.26	S34	0.17	
	24	203	O50	0.48	W24	0.16	
	24	203	O50	0.48	W24	0.16	
	25	113	G43	0.96	G17,G7	0.02	
	26	202	Q3	0.34	UNKNOWN	0.16	

The elbow method graph for yoga dataset sates that the best cluster number is 30.



Using the same feature descriptor method, I have did the kmeans with 30 cluster. Results in below.

kmeans_30.txt

```
./DATASET/TEST/goddess/00000091.jpg:17
./DATASET/TEST/goddess/00000022.jpg:16
./DATASET/TEST/goddess/00000001.jpg:16
./DATASET/TEST/goddess/00000055.jpg:4
./DATASET/TEST/goddess/00000012.jpg:23
./DATASET/TEST/goddess/00000058.jpg:5
./DATASET/TEST/goddess/00000027.jpg:5
./DATASET/TEST/goddess/00000034.jpg:7
./DATASET/TEST/goddess/00000040.jpg:3
./DATASET/TEST/goddess/00000005.jpg:3
./DATASET/TEST/goddess/00000080.jpg:12
./DATASET/TEST/goddess/00000029.jpg:7
./DATASET/TEST/goddess/00000045.jpg:3
./DATASET/TEST/goddess/00000084.jpg:3
./DATASET/TEST/goddess/00000028.jpg:16
./DATASET/TEST/goddess/00000046.jpg:23
./DATASET/TEST/goddess/00000085.jpg:4
./DATASET/TEST/goddess/00000004.jpg:3
./DATASET/TEST/goddess/00000087.jpg:17
./DATASET/TEST/goddess/00000060.jpg:28
./DATASET/TEST/goddess/00000068.jpg:4
./DATASET/TEST/goddess/00000041.jpg:17
./DATASET/TEST/goddess/00000003.jpg:13
./DATASET/TEST/goddess/00000069.jpg:14
./DATASET/TEST/goddess/00000014.jpg:7
./DATASET/TEST/goddess/00000073.jpg:27
./DATASET/TEST/goddess/00000088.jpg:3
./DATASET/TEST/goddess/00000031.jpg:5
./DATASET/TEST/goddess/00000077.jpg:16
./DATASET/TEST/goddess/00000023.jpg:19
```

Jiayin:

	A	B	C	D	E	F	G	H
1	Cluster	Total Imag	Top 1	Percent	Image Display	Top 2	Percent	Image Display
2	0	864	N35	51.85		Unknown	5.32	
3	1	204	O50	48.04		W24	14.71	
4	2	200	S29	100				
5	3	230	G17	53.91		G1	14.35	
6	4	55	V28	65.45		S34	14.55	
7	5	54	D54	22.22		G7/Unknown	18.52	
8	6	206	Q3	33.98		Unknown	17.96	

	A	B	C	D	E	F	G	H
9	7	208	X1	85.1		E34	12.98	
10	8	91	G17	76.92		G35	8.76	
11	9	212	M17	81.13		U33/M18	4.25	
12	10	172	Unknown	14.53		O1	11.63	
13	11	63	D58	52.38		Q1	26.98	
14	12	288	M17	28.13		S29	22.57	
15	13	142	G43	100				
16	14	33	M23	100				

	A	B	C	D	E	F	G	H
19	17	154	I9	92.21		F18	3.25	
20	18	55	D35	100				
21	19	251	D21	61.35		D4	14.34	
22	20	110	R8	59.09		M17		
23	21	97	M17	96.91		Unknown	2.06	
24	22	150	V31	87.33		F13/V30	5.33	
25	23	65	V13	80		N37/Z7	6.15	
26	24	51	G43	100				
27	25	69	G25	39.13		G39	23.19	
28	26	50	Z1	90		W24	4	

agg_cluster_30 - Notepad

File Edit Format View Help

```
./DATASET/TEST/goddess/00000091.jpg:12
./DATASET/TEST/goddess/00000022.jpg:11
./DATASET/TEST/goddess/00000001.jpg:11
./DATASET/TEST/goddess/00000055.jpg:7
./DATASET/TEST/goddess/00000012.jpg:23
./DATASET/TEST/goddess/00000058.jpg:9
./DATASET/TEST/goddess/00000027.jpg:9
./DATASET/TEST/goddess/00000034.jpg:15
./DATASET/TEST/goddess/00000040.jpg:6
./DATASET/TEST/goddess/00000005.jpg:11
./DATASET/TEST/goddess/00000080.jpg:2
./DATASET/TEST/goddess/00000029.jpg:15
./DATASET/TEST/goddess/00000045.jpg:11
./DATASET/TEST/goddess/00000084.jpg:11
./DATASET/TEST/goddess/00000028.jpg:11
./DATASET/TEST/goddess/00000046.jpg:3
./DATASET/TEST/goddess/00000085.jpg:26
./DATASET/TEST/goddess/00000004.jpg:6
./DATASET/TEST/goddess/00000087.jpg:5
./DATASET/TEST/goddess/00000060.jpg:11
./DATASET/TEST/goddess/00000068.jpg:1
./DATASET/TEST/goddess/00000041.jpg:5
./DATASET/TEST/goddess/00000003.jpg:1
```

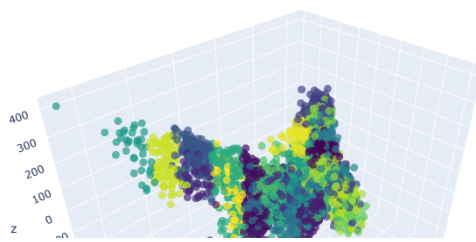
Kangdong:I spent a very long time making interactive 3D pictures. In addition, I optimized my PCA method, which allows users to do PCA only by modifying variables.

```
import plotly.graph_objects as go


# Helix equation
t = np.linspace(0, 20, 100)
x, y, z = d3[:,0], d3[:,1], d3[:,2]

fig = go.Figure(data=[go.Scatter3d(
    x=x,
    y=y,
    z=z,
    mode='markers',
    marker=dict(
        size=5,
        color=list1,                # set color to an array/list of desired values
        colorscale='Viridis',   # choose a colorscale
        opacity=0.7
    )
)])

# tight layout
fig.update_layout(margin=dict(l=0, r=0, b=0, t=0))
fig.show()
```



Yuqi	Found that template matching is also not ideal for the purpose of detect cartouche Starting planning on build a deep learning model to detect the cartouche pattern Doing research on how to build a custom object detecting model
Kangdong	- K-means clustering - Visualization - Improved efficiency of model training
Sydney	- Researched resnet34 and alexnet models - Explored classes for these classification techniques - Looked into python documentation of parameters for these models
Juliana & Xueqing	- Analyze the Egyptian Hieroglyphics with images attached

	- Did clustering for the downloaded yoga dataset - Used elbow method to find that 30 is again the best number of clusters - Jiayin: Agglomerative Clustering - Xueqing: K-means Clustering
All	- Statements of work - Team meetings x 2

Week Twelve:

This week Sydney worked to implement ResNet and AlexNet models. She spent a lot of time reading in the data and ran into processing time errors in Google Colab. She then implemented transformations to the data to find the features that represent pixels in the images. She also resplit the data to work with validation and training sets rather than just training and test sets. Additionally, she processed the data with dummy variables and dropped columns with drop out rates.

Xueqing and Jiayin(Juliana) continue to work on the yoga dataset. They use their clustering algorithm to fit the data with 100 clusters and 60 clusters. Although the results of clustering show that 30 is the best number of clusters, they think it is worth testing different clusters. Even if they see from the elbow method that 30 is a good number to cluster, they should not be 100% sure, it is good to check on other possibilities and then come to a conclusion. So they tested more, then decided to conclude to analyze and compare to 30 clusters for yoga and Egyptian dataset.

Kangdong：I made detailed settings for these steps and added notes at each step. I set each step as a code block, and each parameter is saved as a variable. The user can easily change the variable value to adjust the k-means model and PCA function. If the user wants to see the error value of the k-means model, when the number of clusters is within an interval. I set up a code block to automatically execute this process. The user only needs to specify a range of the number of clusters, and the code can automatically generate a line graph.

Yuqi	Starts to train the model and pick the best model among the outputs Use random images from the dataset to test if detection of cartouches can be succeed or not
Kangdong	- K-means clustering - Visualization - Improved efficiency of model training
Sydney	- ResNet and AlexNet background research continued - Added validation set for evaluation - Transformed the data and defined features of pixels in the images - Dummy variables implementation for labeling
Juliana & Xueqing	- Did cluster for yoga dataset with clustering number of 60 and 100 - Jiayin: Agglomerative clustering - Xueqing: K-means clustering
All	- Team meeting - Meeting with Professor

Week Thirteen:

This week, Sydney worked on documenting hyperparameters for the ResNet and AlexNet models. She also did a lot more research on transfer learning and how drop out rates, shuffling, and larger training sets can improve accuracy on models. She also worked with another dataset to see if the model would work well on other images but ran into the errors associated with processing black and white images versus RGB images. She also finalized her data pipeline and plans on continuing to improve accuracy of her models in the last two weeks of the project.

This week, Jiayin and Xueqing analyzed their yoga data clustering. They both found the similar problem that the clustering does not do too well with the same feature descriptor and same clustering number as the Egyptian Hieroglyphic data. They guessed that it could because the

yoga data only have five different poses. As for the conclusion of the project. The best model might be the classification for it has the best fitting for all data.

Kandgong: I spent a lot of time writing documents, and I also undertook the task of managing group documents, so I did a lot of documenting work.

Yuqi	Piece everything together and start to write the scratch of the report
Kangdong	- K-means clustering - Visualization - Improved efficiency of model training - Document writing
Sydney	- Worked with another dataset to see model evaluation - Worked with hyperparameters and studied documentation to try to implement improvement in the following weeks - Finalized data pipeline and design - Compiled all of our weekly reports for project report to instructor - Wrote the background and model comparison sections for project report to customer
Juliana & Xueqing	- Analyze the clustering of yoga data - Compared with the Hieroglyphic data
All	- Added to user manual - Created individual work documents - Contributed to project reports

Week Fourteen

Yuqi	Document Writing
Kangdong	-

Sydney	- Implemented ResNet34 and AlexNet models on another dataset and documented accuracies
Juliana & Xueqing	-
All	-

Week Fifteen

Yuqi	Document writing
Kangdong	-
Sydney	- Finalized model - Wrote final report sections
Juliana & Xueqing	-
All	- Team meeting to discuss goals - Team meeting to create presentation - Made slides for personal sections - Presented final slides to professor and other students