Hieroglyph Signs Decoding

Team Members

- Remon Samir Zaki Gad
- Omar Mohamed Othman Ahmed
- Michael Samir Youssef Fam
- Amr Ahmed Mohamed Mahmoud
- Ahmed Khalil Mohamed
- Mohammed Rabie Khames

- Supervised by: Dr. Shahira Mahmoud Habashy

Contribution

- Ahmed Khalil Mohamed

Phase one → Flutter

- Amr Ahmed Mohamed Mahmoud

Phase two → Integration

- Mohammed Rabie Khames

Phase two → Integration

- Omar Mohamed Othman Ahmed

Phase three → Character Detection

- Michael Samir Youssef Fam

Phase four → Glyph Classification Machine Learning

- Remon Samir Zaki Gad

Phase five → Glyph Classification Deep Learning, Translation and Augmented Reality

- Tourism is a very important topic for Egypt economy and is continuously growing nowadays. Many tourists come to Egypt to see the beautiful Pharaonic antiquities and read the Egyptian hieroglyphs to know more about the old Egyptian habits and how they lived.

Project Idea Explanation

- Hieroglyph Signs Decoding app comes to support tourism and researchers who interests in this field using machine learning. It is Flutter app that helps the tourist to understand Egyptian Hieroglyphs instead of just watch them by translation process to English. Now just by taking a photo from your phone to the part of text you want to translate you can read in English the Pharaonic words and understand their great civilization by just a shot.

Objectives

- This project aims to promote tourism using machine learning and help tourists to understand the old Egyptian Hieroglyphs in an easy way by taking a photo of the text to be translated and the program will process this photo and return the text in English.
- The program also help researchers in this field to fasten their work as the program acts as a tool to translate any Hieroglyphic sentences.
- Translation System can be used as a separate API to serve the needs of researchers for translating any confusing sentences to English in just seconds.

Used programs and algorithms

- Our project is a Flutter app so GUI is made by dart programming language. The main functionality of our app is to translate Hieroglyphic sentences to English language by using python programming language and Machine Learning algorithms that manipulate data of the input image and using Deep learning model we transform input image to Gardiner's code then by Natural Language Processing (NLP) we transform Gardiner's code to English sentences.

Used programs and algorithms

- We also use Augmented Reality (AR) to enhance the visuals of the application and give the user the right atmosphere by displaying 3D statues for kings beside their name in the Cartouches. Then we connect our mobile app with python code through Django server that run the python code

The main challenge

- Our main challenge is data entry time and data number limit. Hieroglyph language has more than 1000 letter and there is no dataset that contains all of these letters with variations in writing style and with high resolution size and quality. We also do not have any ready made library that translate Hieroglyph language so we have to make our library with ourselves.

Egyptian Hieroglyph Language Signs

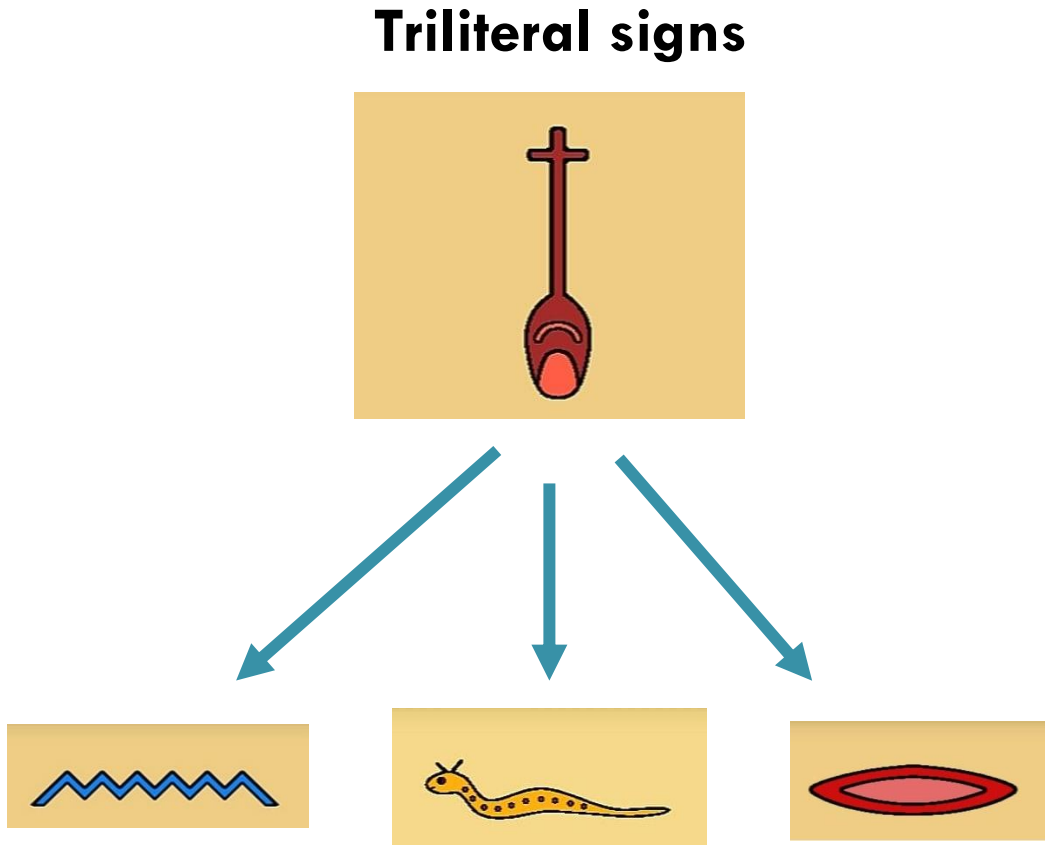
- Egyptian hieroglyphs were the formal writing system used in Ancient Egypt. Hieroglyphs combined logographic, syllabic and alphabetic elements.
- Hieroglyphs may have emerged from the preliterate artistic traditions of Egypt.
- Egyptian hieroglyphs have a total of 1,000 distinct characters.

Egyptian Hieroglyph Language Signs

- Subsets

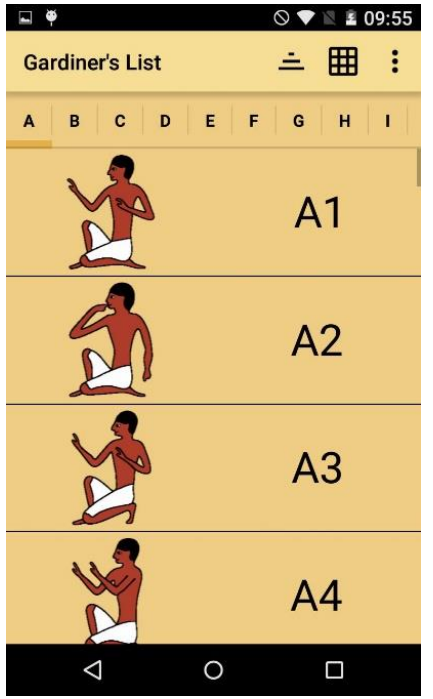
Notable subsets of hieroglyphs:

1. Deterministic
2. Uniliteral signs
3. Biliteral signs
4. Triliteral signs
5. Egyptian numerals

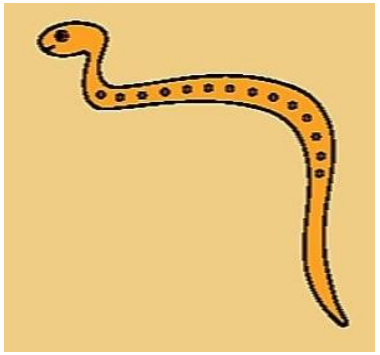


Gardiner's sign list (Gardiner's code)

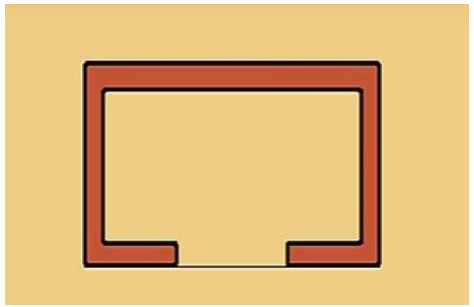
- In 1928/1929 Alan Gardiner published an overview of hieroglyphs, Gardiner's sign list
- Gardiner's Sign List is a list of common Egyptian hieroglyphs compiled by Sir Alan Gardiner. It is considered a standard reference in the study of ancient Egyptian hieroglyphs. It describes 763 signs in 26 categories (A–Z, roughly).
- Gardiner lists only the common forms of Egyptian hieroglyphs.



Gardiner's sign list (Gardiner's code)



Gardiner's code = J10

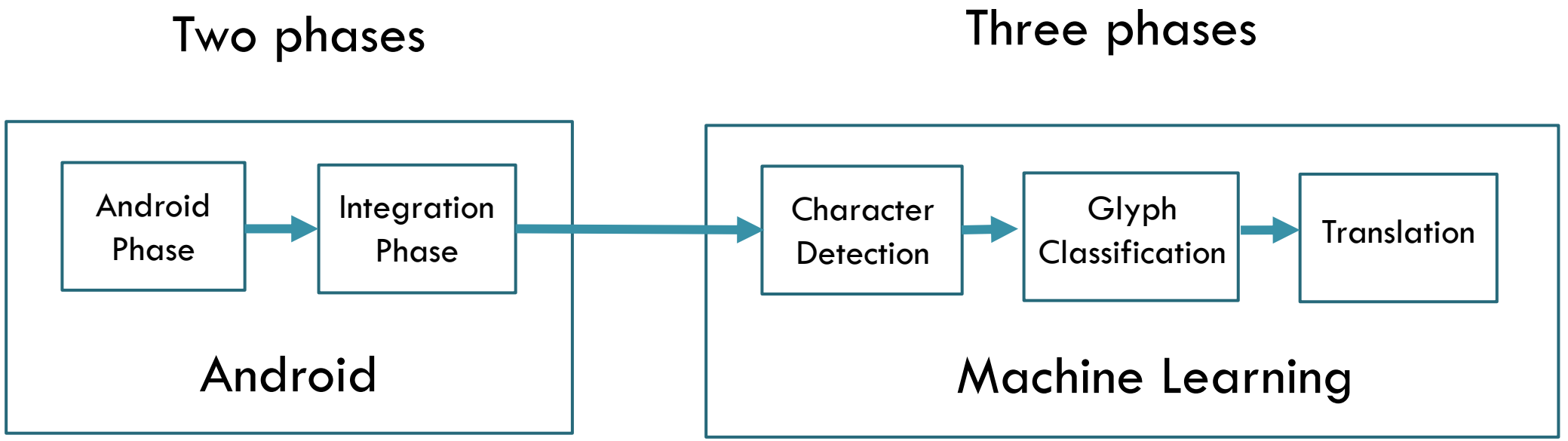


Gardiner's code = O1

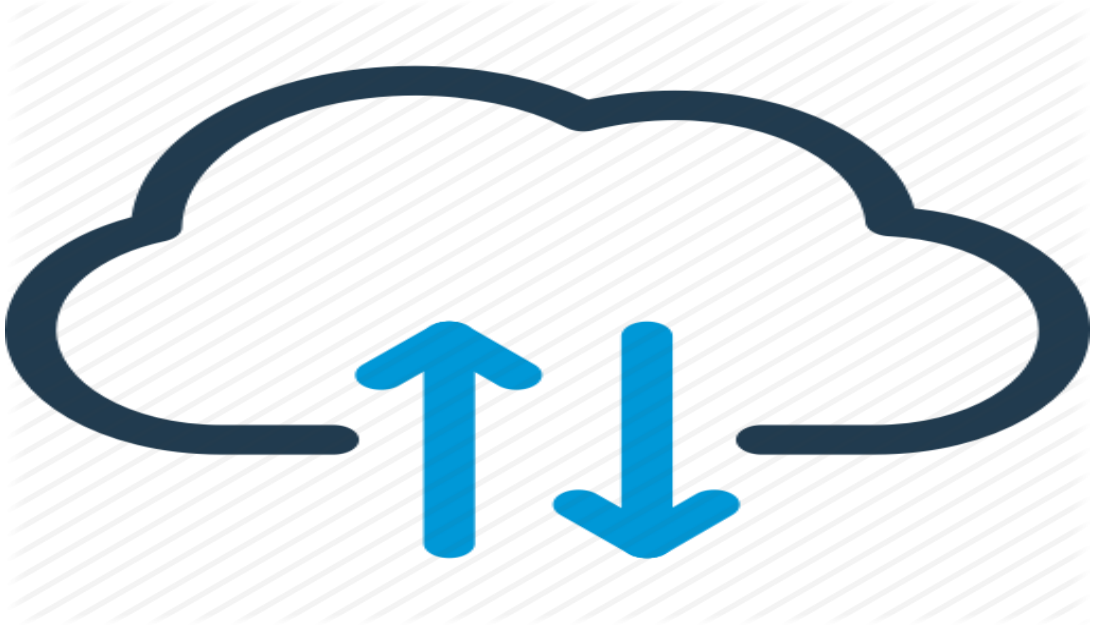
Gardiner's code = F35

Project Phases

- Block Diagram to describe project phases



Project Phases



Phase One
Flutter Phase

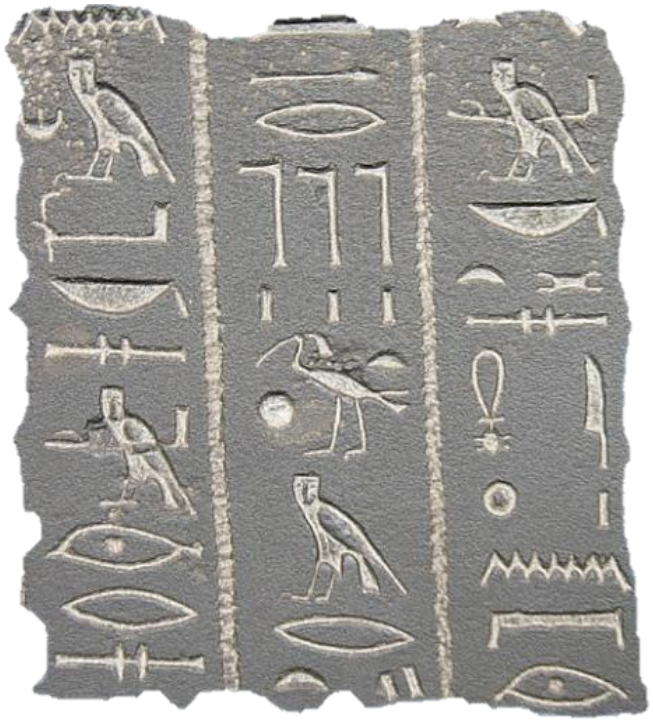
App design and implementation

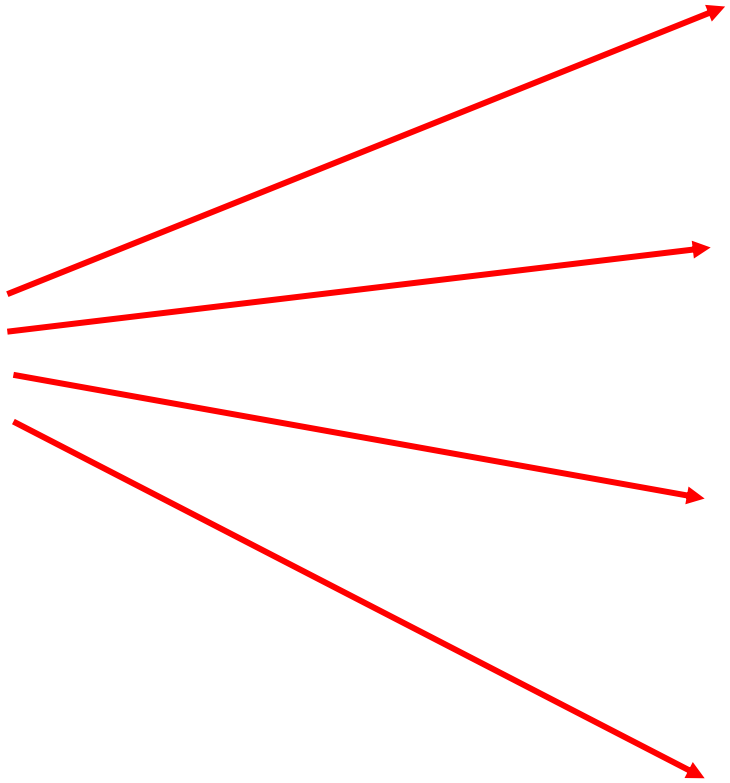
Phase Two
Integration Phase

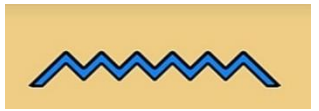
Send image to server database
and store it for processing

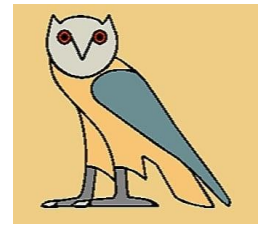
Project Phases

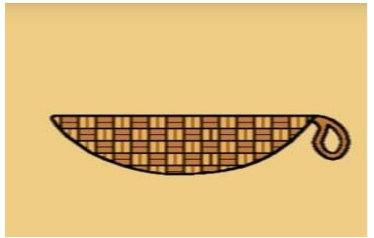
Phase Three







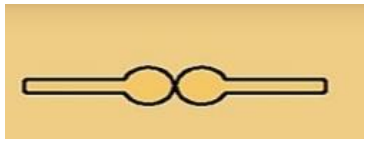


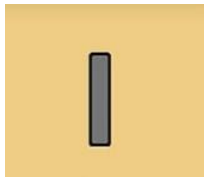


- Process the image and divide it into small letter images

Project Phases

Phase Four


O34 ,

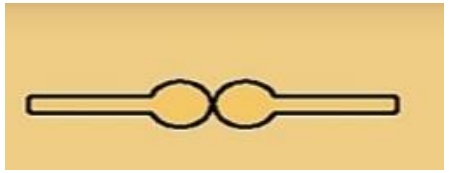
A1 ,

Z1 ,

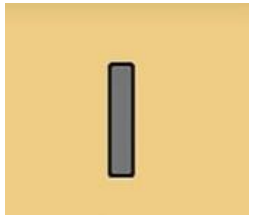
F35”

- Process the letters images and convert each of them to into its equivalent Gardiner code

Project Phases

Phase Five





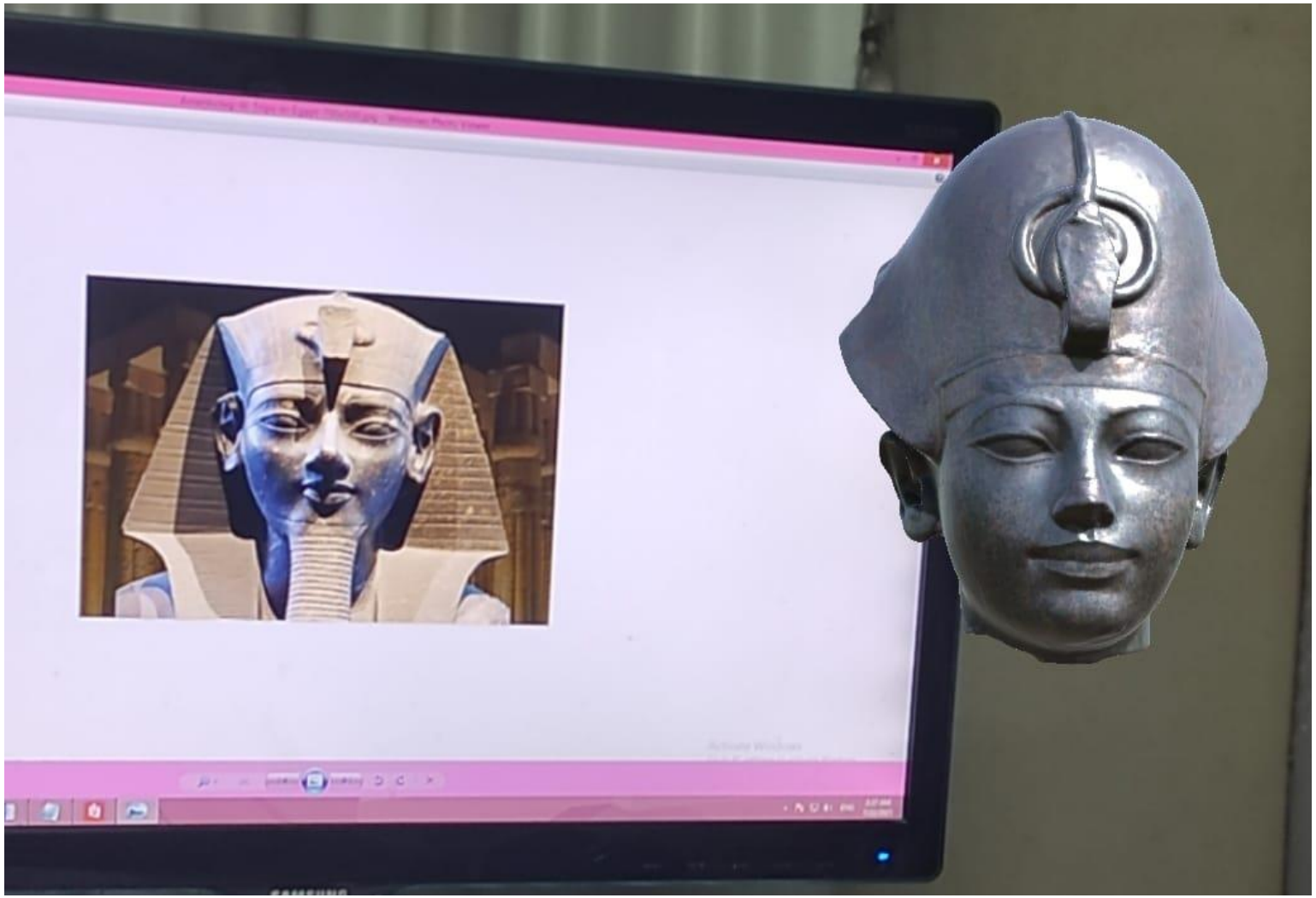
- Input text: "O34" , "A1" , "Z1" , "F35"
- Output text in Gardiner Code: "O34 , A1 , Z1" , "F35"
- Output text in English: Man is good (الرجل طيب)

- Process the Gardiner codes and divide them into useful words and sentences then translate them to English and apply grammar correction

Augmented Reality with Unity game engine and Flutter

- What if the line between your imagination and the real world didn't exist? With augmented reality, not only is that possible, it's here.
- Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by digital visual elements, sound or other sensory stimuli delivered via technology. AR can be defined as a system that incorporates three basic features: a combination of real and virtual worlds, real-time interaction, and accurate 3D registration of virtual and real objects.

Augmented Reality with Unity game engine and Flutter



Augmented Reality with Unity game engine and Flutter

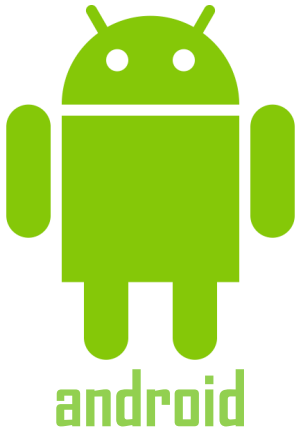
- With our application any tourist can open his camera and point it to any king cartouche and then he will listen to a voice that talks about the king that this cartouche related to with some music and he will see a 3D statue of the king appears beside it in real life.
- Augmented Reality enhance natural environments or situations and offer perceptually enriched experiences.

Phase One

Android and Flutter

MEMBERS TEAM

- **GUI Department**
  - Ahmed khalil Mohamed khalil
  - Amr ahmed Mohamed Mahmoud



- **Machine learning Department**
  - remon Samir
  - Omar Mustafa
  - Maikel Samir
  - Mohamed rabbie

ANDROID

Front End

- GUI is present the user interface that the user will use it to i

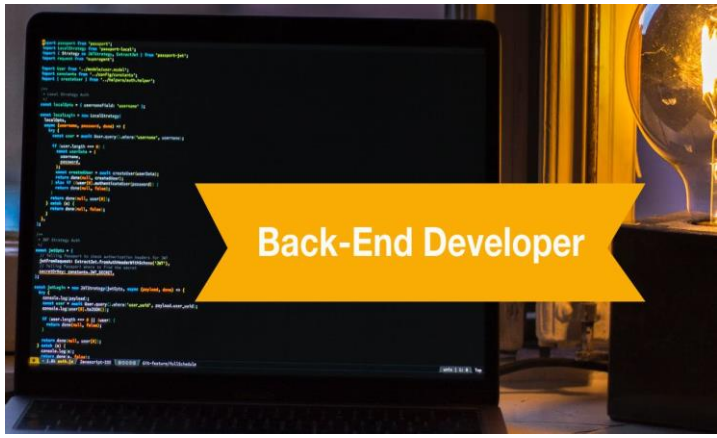
  With application to do

  1) Capture image
  2) Send to serve
  3) About
  4) Learn

Back End

- The main goal is connect to the server has :-

  1) Data base
  2) Python code (machine learning)



GUI PROTOTYPE

- The GUI design contents from :-

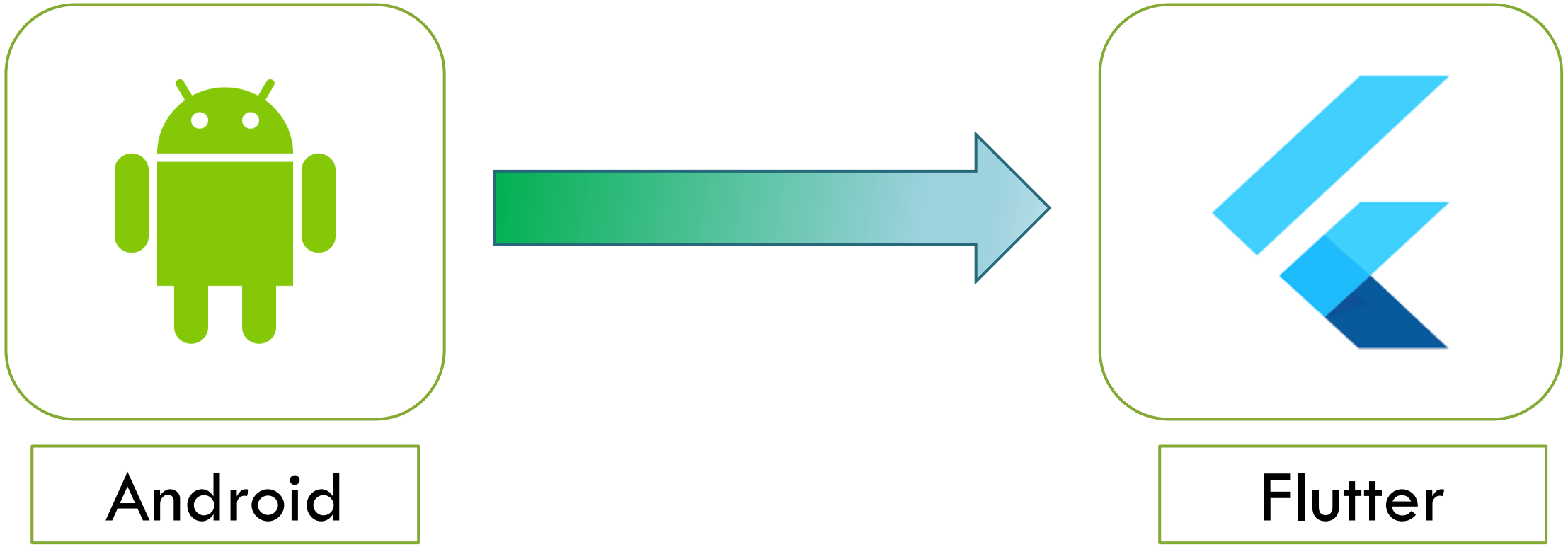
- Four buttons
  1) Capture image
  2) Send to serve
  3) About
  4) Learn

- Image view
  - for show the image before translation process

ANDROID PROBLEMS

- Android Studio Requires High Process And Modern Pc
- Android Is Platform Work Only On Mobile

UPDATING TO NEW PLATFORM



FLUTTER FEATURES?

- Simple Platform-Specific Logic Implementation.
- Same UI and Business Logic in All Platforms iOS and Android
- Reduced Code Development Time

WELCOME SCREEN

Welcome

Welcome to your Hieroglyph Signs Decoding app. Lets take a tour before beginning

Ancient Hieroglyph

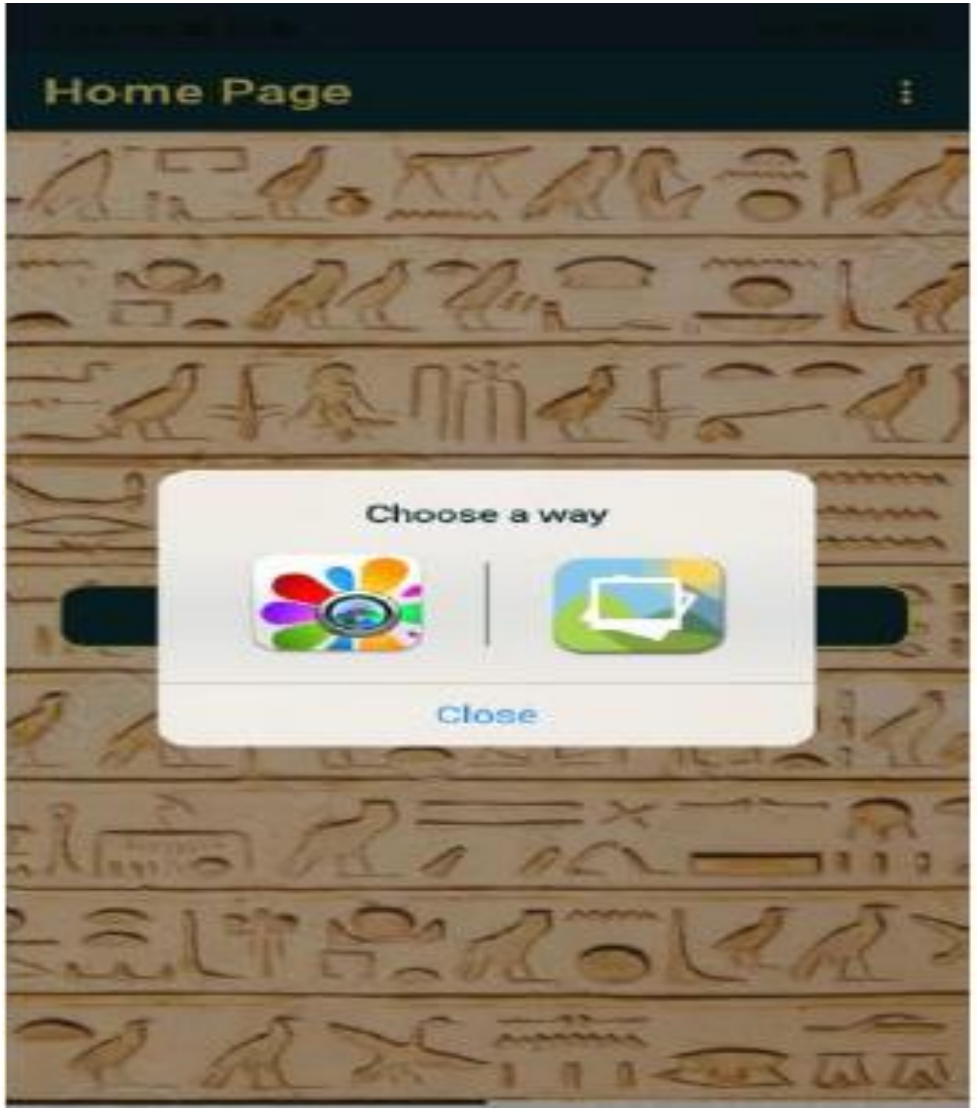
The Language of our ancient pharaohs is alive again. by simple clicks you can understand their language

Augmanted Reality

What about seeing pharaohs in real life after all this years! what about discovering our history and glory

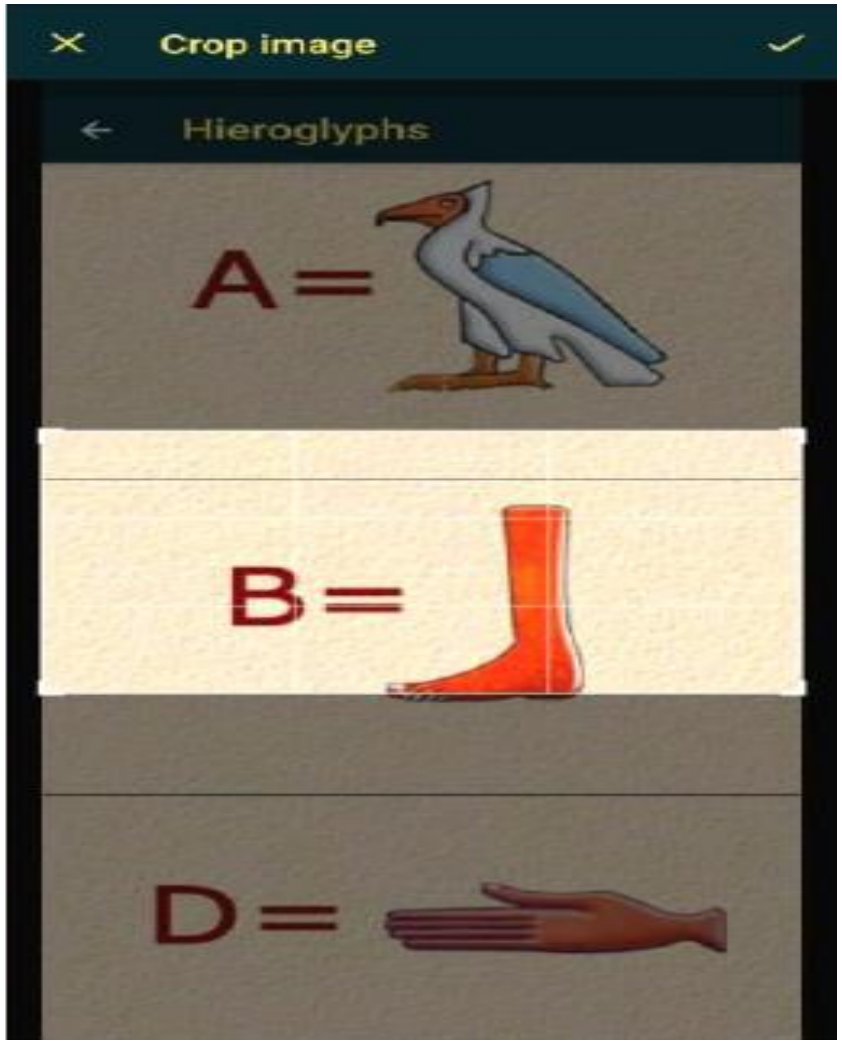
INSERT IMAGE

- Input image can be entered by two methods either
  - take a photo
  - choose one in gallery



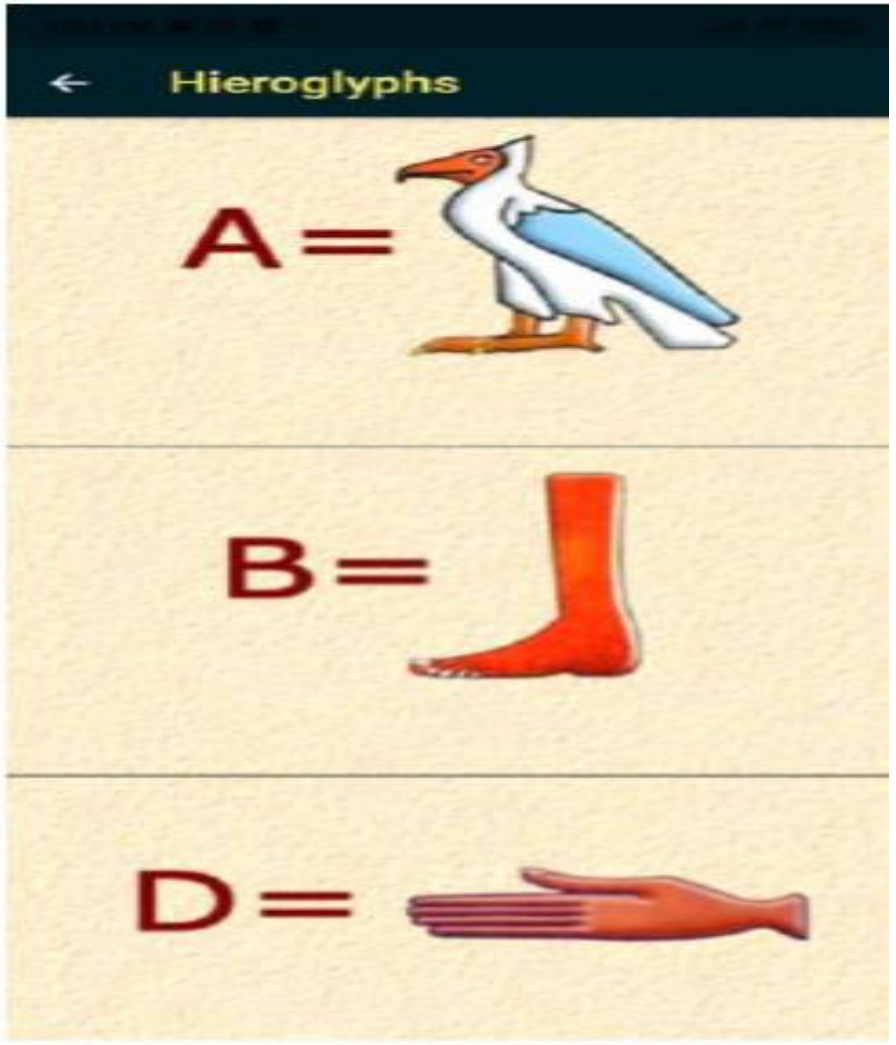
CROP TOOL

- This feature use to apply on image to crop symbol or set of symbols to define meaning



LEARN

- This is activity about learn about Hieroglyph symbols and language



THE RESULT SCREEN

- Sample for insert image and get the result

HISTORY WORK

- You can see the previous image translations used as we store them in our database

Thanks for all

Hieroglyph Signs Decoding

Phase Two

Integration

Integration

- We would like first to work on a PHP server that act as a prototype to test on then we will build the final project with Django server to enable the communication between the Flutter app and Machine Learning code.

Android and Php processes

1- capturing an image

2- sending image parameters (name,date,time) to web service (php)

3- php file will have (image parameters , server content)

4- then sending PHP file to the server & applying it to our database

Capturing image

- First we define our permissions :-

Like (camera , internet , wifi , read external storage , write external storage)

After openning the camera and captures the image ,

We get the image file and put it in variable ;

Then we construct the image itself from that variable and put it in another variable to make processing on it

Capturing image

- To improve performance ; we will convert that image to string form

By using ((imagetostring)) function ; which allow us to specify and control the image quality

Then collecting some paramters like :-

1-image as string

2- name = android-id + current-date + current-time

((using android-id to make sure that we have a unique parameter))

3-Operation-name

4-current-date

Capturing image

- to send those parameters to the web service :

We using VolleySingelton Class as an object to send request

((with paramaeters)) To our web service

Web Services

- We using PHP as Web Service
- After sending the parameters of image PHP holds all information about image & about server .
- Informations like :-

  A) PHP recieves picture ((parameters)) as keys

  Image as string with key (img)

  Name of picture with key (in)

  Current-date with key (da)

  Operation-name with key (opn)

Web Services

- B) PHP holds server information :-

Like (server name , username , password , name of database)

& the connection method

Web services

- PHP processes :-

A) create a folder with operation date if it does not exist

B) and gives the sql query which refer to our ((stored procedures on server))

C) holds reaction responses after connecting to the server

Web Services

- we have two responses :-

A) On Response : means PHP successfully connected with server so we have two cases

a) image was sent to server

Response message ((image loaded successfully))

b) image not send to server

Response message ((please try later))

Web Services

- B) ErrorOnResponse :

means PHP can't reach to the server ;

so we have one response message

((can't reach server))

Server contents

- on the server we have our database with
  - username
  - password

- the database which has three table for storing the capture image :-
  - Operation name
  - Pic name
  - Result

we are using ((stored procedure feature)) to apply queries on our database

- We choose to work with Django server as most of our ML code is written in python so we want the server to directly communicate with our code and by connecting this server to our app in Flutter by sharing the same network we can open our app in android or IOS. Also using Django is useful as it is easy to extend and scale. We used this technique to minimize the overall processing time as any other server will complicate and increase communication time in our app.

DJango

- Django is a high-level Python web framework that enables rapid development of secure and maintainable websites
- Django helps you write software that is:

Complete : Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box"

many-sided : It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.)

Secure : Django helps developers avoid many common security mistakes by providing a framework that has been engineered to

"do the right things" to protect the website automatically.

For example, Django provides a secure way to manage user accounts and passwords

Django processes

- First we create our server and running it using python code
- Then generate our API which will communicate with Flutter
- API will be responsible for receive request and get response
- We will use Django REST framework to get image from Flutter and to return result to Flutter by using JSON
- REST is a loosely defined protocol for listing, creating, changing, and deleting data on your server over HTTP

- As we see Django is more simple than Php sql server and have many advantages :

1) Written in Python
2) Designed as a batteries-included web framework
3) Supports MVC programming paradigm
4) Compatible with major operating systems and databases
5) Provides robust security features
6) Easy to extend and scale

Phase Two

Integration

Processing of the image

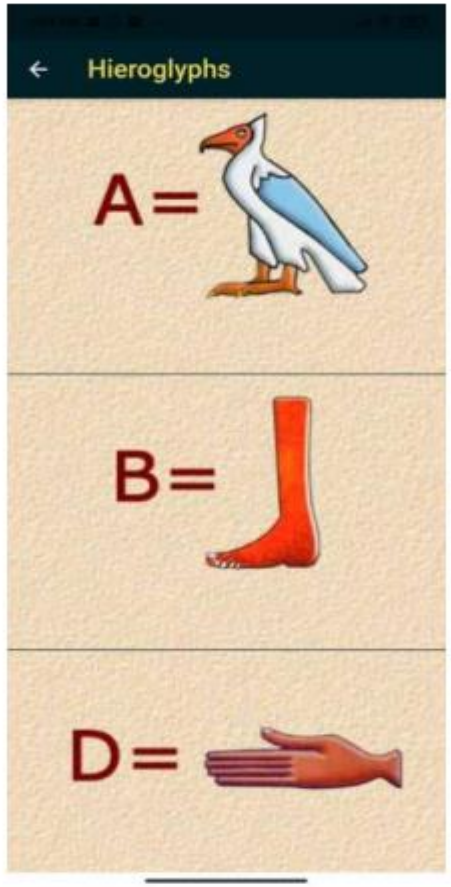
Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it.

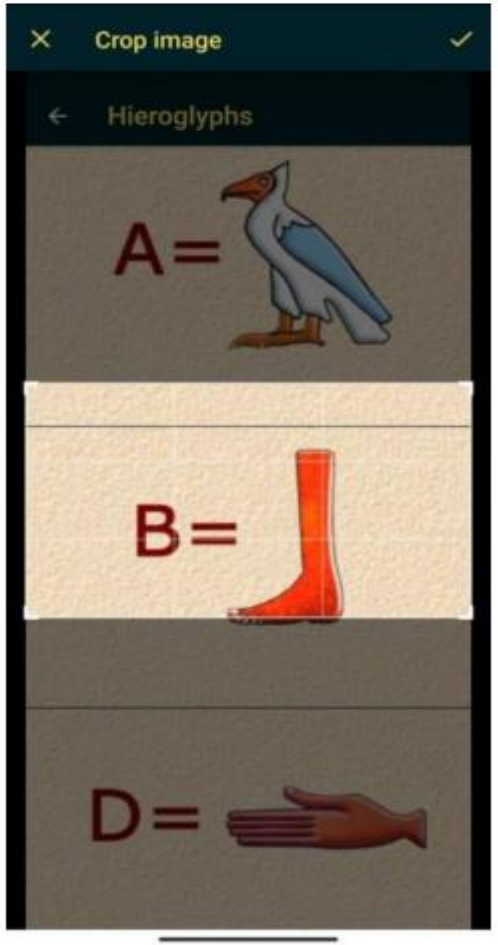
It consists of :

- Cropping of an image
- Image scaling or Resize
- Rotation of an image

Cropping of an image

CROPPING an image is the act of cutting away and discarding the unnecessary portions of the image.





Why should you use cropping?

- To emphasize the center of interest
- To eliminate an unwanted portion
- To adjust the shape to fit a given layout
- To enlarge small portions

Cropping an image can:

- change the direction and balance of a composition
- change the focus
- remove unnecessary information or parts of a ground that simply don’t work within the composition
- create greater emphasis
- help resolve background issues and can assist in placing the figure on the ground more effectively.

The difference between resizing and cropping:

Resizing changes the dimensions of the image, which usually affects the file size (and, thereby, image quality).

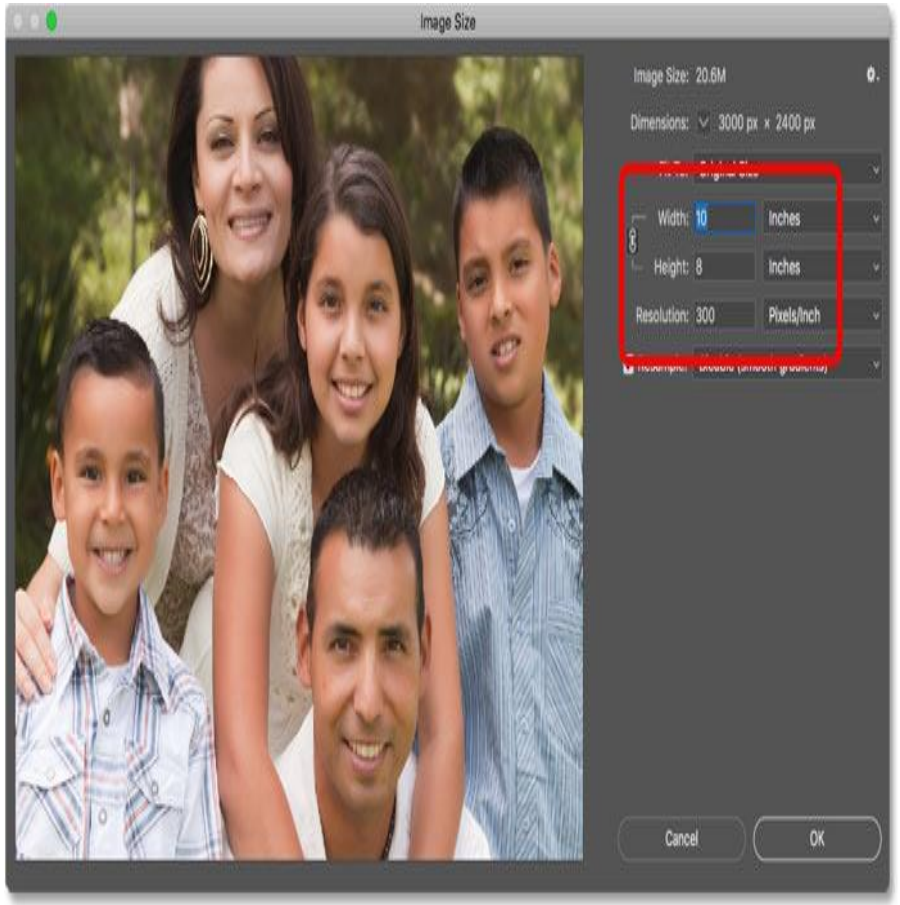
Cropping always involves cutting away part of the original image and results in some of the pixels being discarded.

image scaling or Resize :

When an image is resized, its pixel information is changed.
For example, an image is reduced in size, any unneeded pixel information will be discarded.

When an image is enlarged, the photo editor must create and add new pixel information -- based on its best guesses -- to achieve a larger size which typically results in either a very pixelated or very soft and blurry looking image.

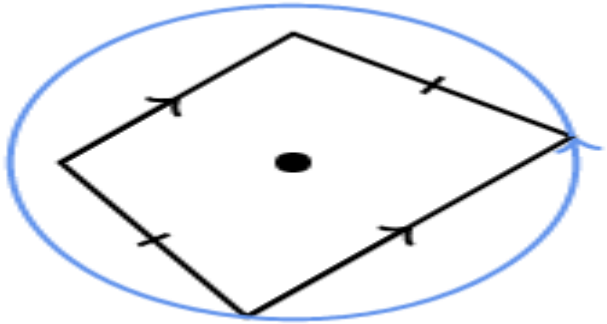
When we resize an image usually change the width, height and Resolution of the image.



Rotation of an image:

Image rotation is a common image processing routine with applications in matching, alignment, and other image-based algorithms.

Image rotation makes image turn in a cycle around a definite center point.



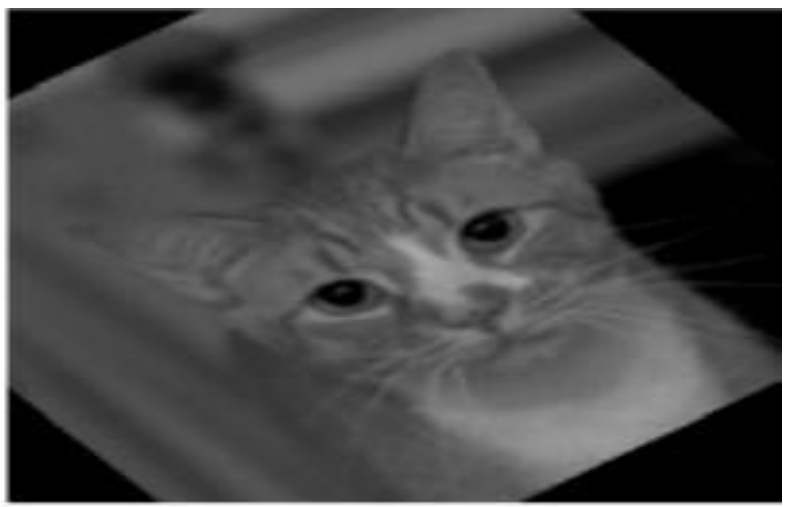
the distance of each rotated point from the center remains the same. Only the relative position changes.

The image consists of small parts called pixels, Each pixel has a coordinate pair *(x,y)* describing its position on two orthogonal axes from defined origin *O*.

The coordinates of a point (x_1, y_1) when rotated by an angle θ around the origin (0, 0), we get(x_2, y_2) Where

$$x_2 = \cos(\theta) * (x_1) + \sin(\theta) * (y_1)$$
$$y_2 = -\sin(\theta) * (x_1) + \cos(\theta) * (y_1)$$

Original image



After rotation of 45°

Phase Three

Characters Detection

Introduction

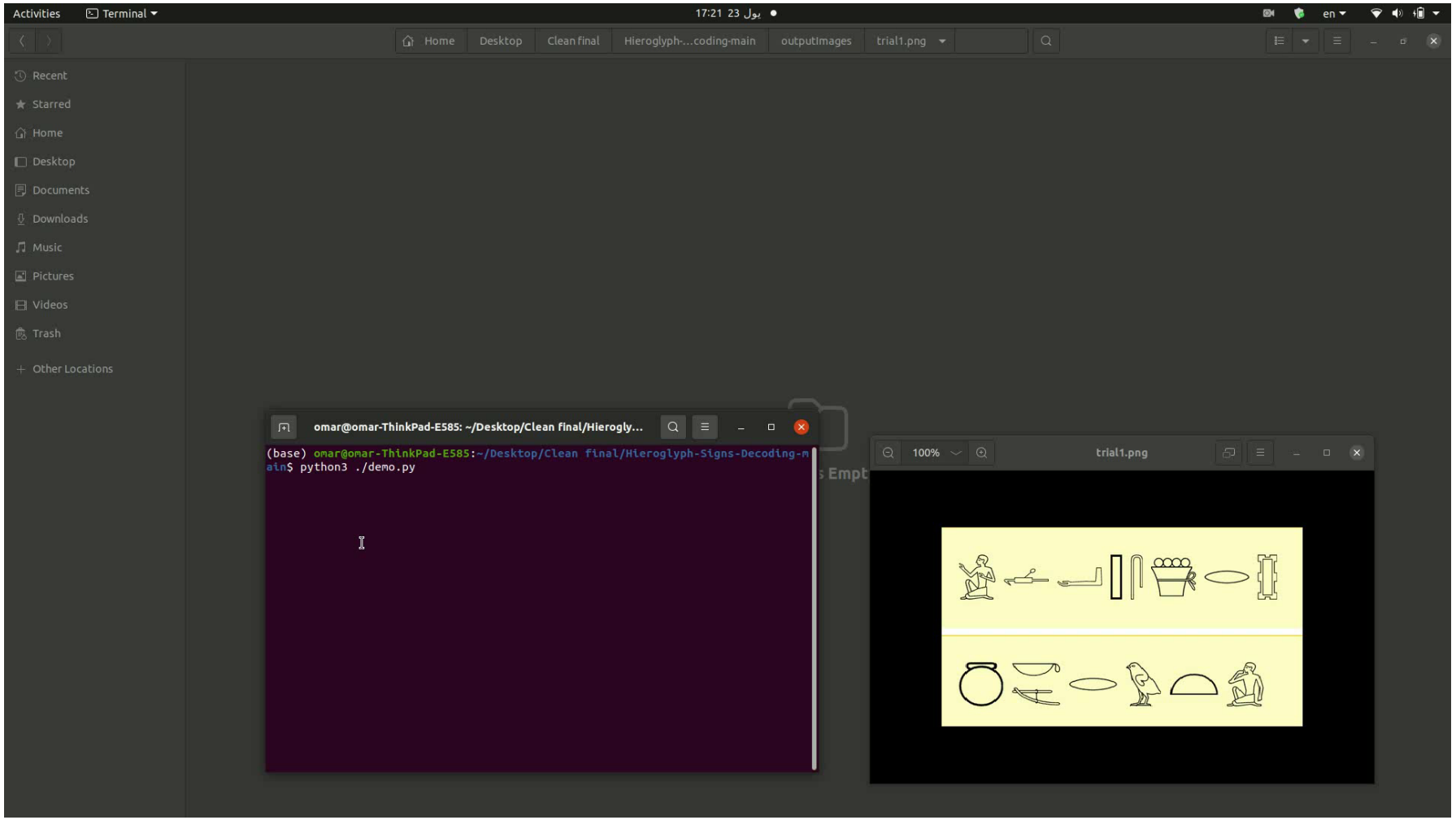
What is the goal of this part ?

- The goal of this part is the detection of the characters present in the image, then trying to find their optimal reading order.

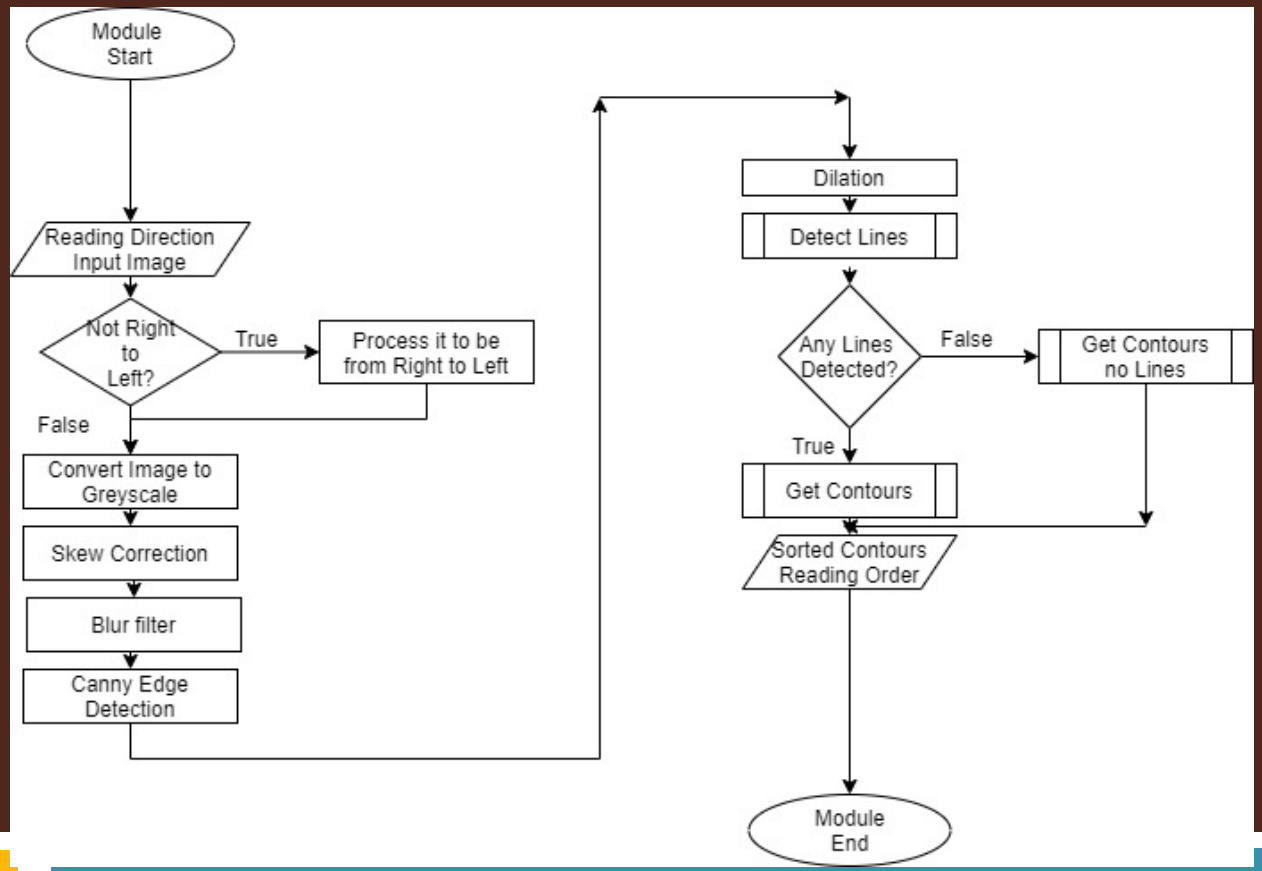
Why we choose this approach :

- Since Hieroglyph is considered as a dead language there is no available library to detect Hieroglyphics, while for other popular languages there is a lot of libraries like Tesseract and EasyOCR
- Also, Hieroglyph have more than 1000 distinct characters and a few available texts, and as mentioned before detecting reading order is a tricky task, which make the approach we use the best approach

Demo Video

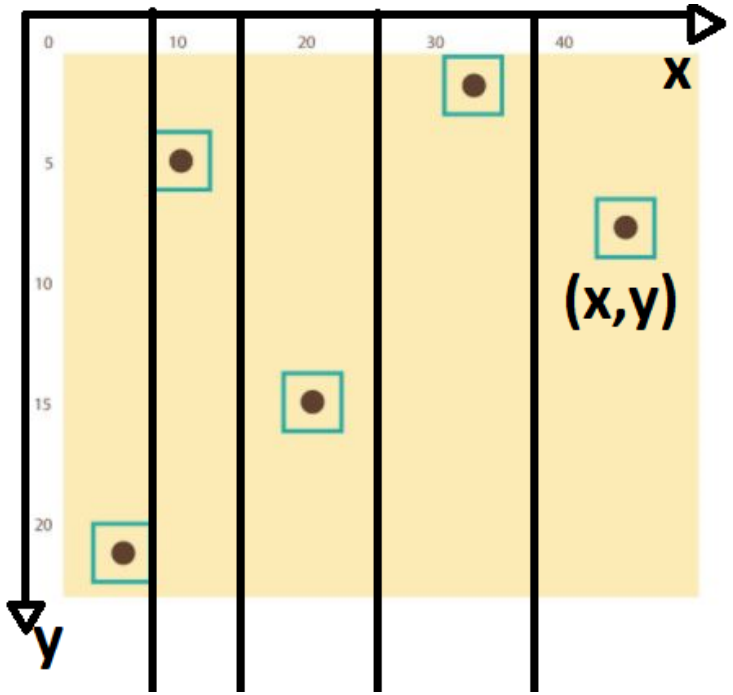

Activities
Terminal
17:21 23 يول
en
Home
Desktop
Clean final
Hieroglyph-...coding-main
outputImages
trial1.png
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
omar@omar-ThinkPad-E585: ~/Desktop/Clean final/Hieroglyph-Signs-Decoding-main$ python3 ./demo.py
100%
trial1.png

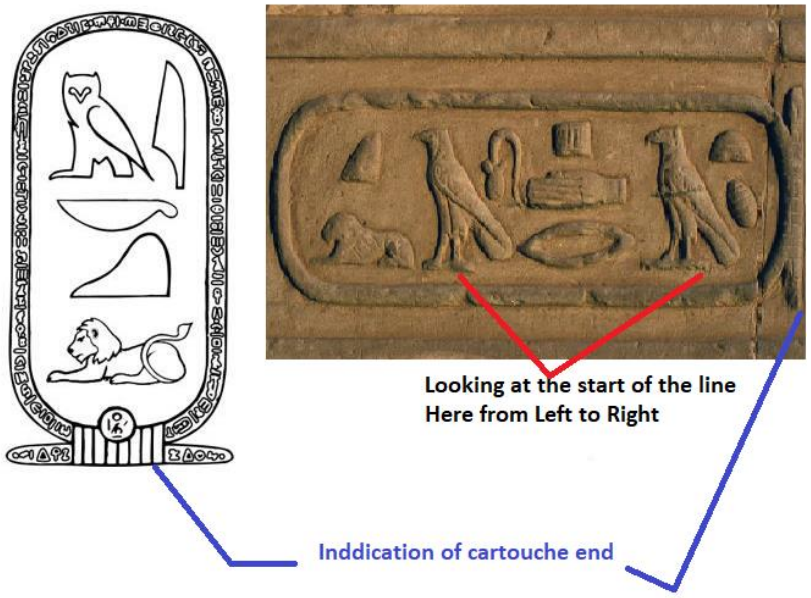
Flow Chart for the main class

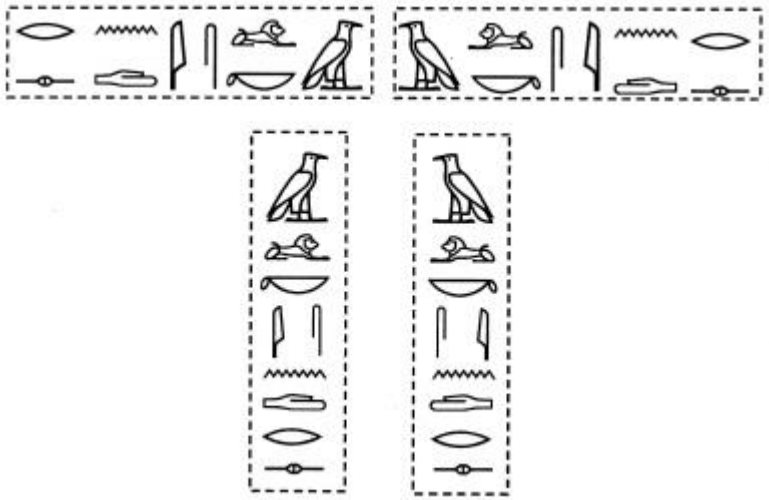


Hieroglyphics reading order

- According to Wikipedia 96% of hieroglyphics are horizontal and from right so by default the reading direction is set to right to left

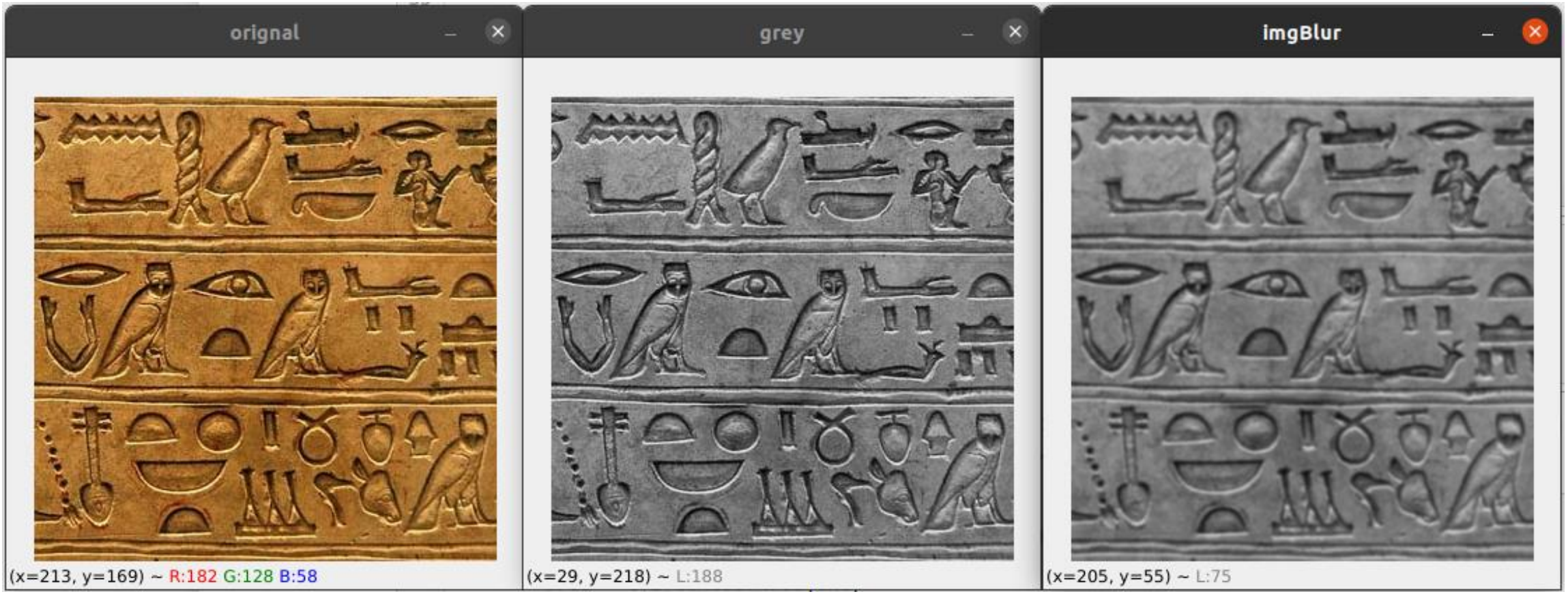






- Hieroglyphs are written in rows or columns and can be read from left to right or from right to left. You can distinguish the direction in which the text is to be read because the human or animal figures always face towards the beginning of the line. Also, the upper symbols are read before the lower.

Grey and blur



- convert the image to grayscale since grayscale images are entirely sufficient for our tasks and so there is no need to use more complicated and harder-to-process color image.
- Then we apply a blur filter to smooth edges and removes noise from an image (Gaussian Noise Reduction)

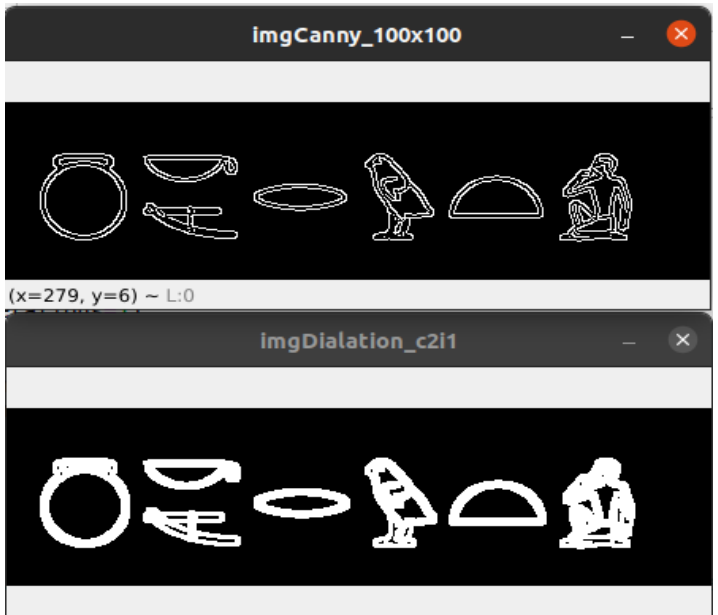
Skew Correction

- The process of straightening an image that has been scanned or photographed crookedly, that is an image that is slanting too far in one direction, or one that is misaligned. Also called deskew.

- Unfortunately, as we mentioned before that Hieroglyphics are written in rows or columns, detecting skew will be a confusing task to do, since vertical writing will be detected as skewed by approximately 90 degrees so we will add to the detected skew angle if it lies between +-45 and +-135 we will add to it 90 degrees, but deskewing will still be a confusing task.

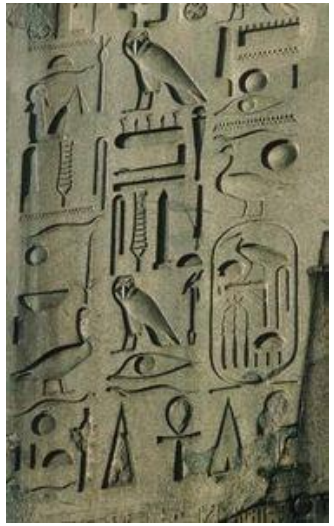
Dilation and Canny edge detection

- We use the canny edge detection algorithm as it is most effective edge detection algorithm
- There are other edge detection techniques like Sobel, Prewitt and Laplacian, but canny give the best results here by trial so we go with it.
- Kernel size is set to be 3 (the size of the Sobel kernel to be used internally) and we set the minThreshold & maxThreshold to 100 and 200 respectively
- Dilation used to increase the thickness since Canny will produce a thin line that may make a problem when passed to the cv2.findContours function leading to not all the contours got detected:
  - **We will do an initial dilation to start with using a 3x3 kernel of ones and 1 iteration**
  - **In some cases, this kernel can lead to detecting multiple characters as one character (will be fixed later)**



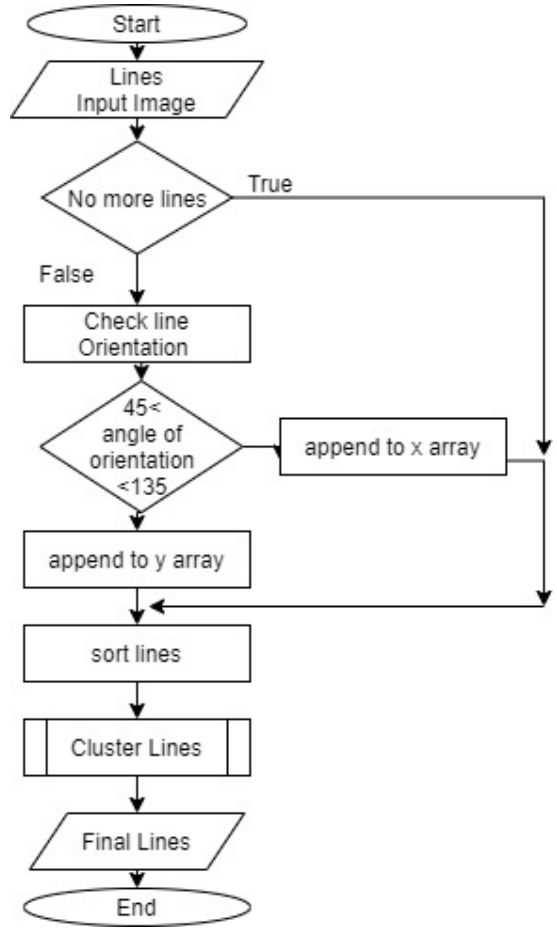
Lines detection

- As mentioned Hieroglyphs can be written in rows or columns , and it can be written with or without separating lines.
- In most case the Writings will have Lines especially vertical writings (lines will be absent in some digital designs and in a super rare cases).
- Detecting these lines will help us to get the optimal reading order.

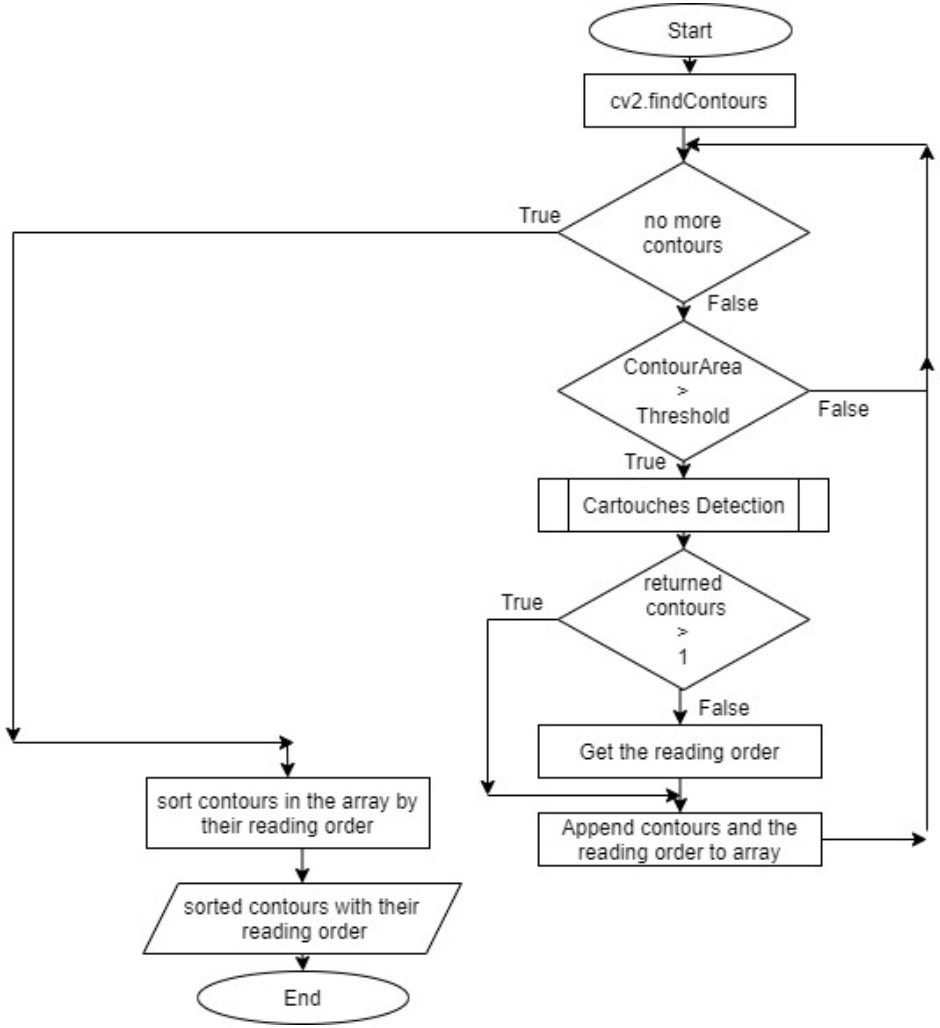



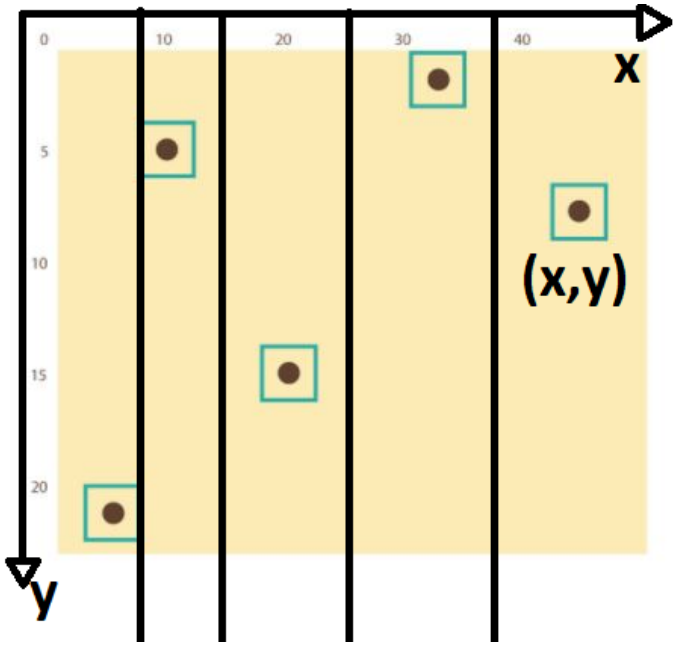
Lines detection

- Since the function “cv2.HoughLinesP” will produce so many lines, we need to process this lines and cluster each group of lines into a single line.
- We will loop all the lines and check the angle of orientation of each line, if its between 45 and 135 we will append this line to the vertical “y” array, else we will append it to the horizontal “x” array.
- Then we will initially sort all the lines, then we will cluster every group of lines that have same orientation angle +- a threshold and the same distance +- a threshold into a single final line.



Get contours

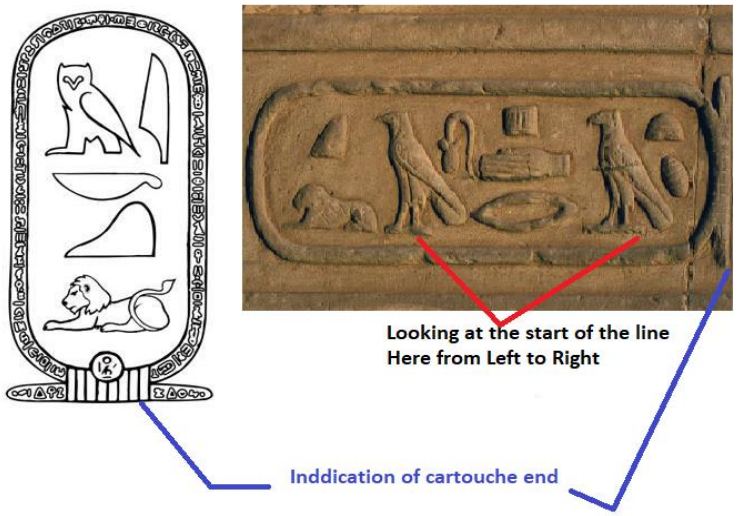




Cartouches Detection

- a function that work with the same manner as Get Contours function but it will crop the frame of the contour then apply the Find Contours function again

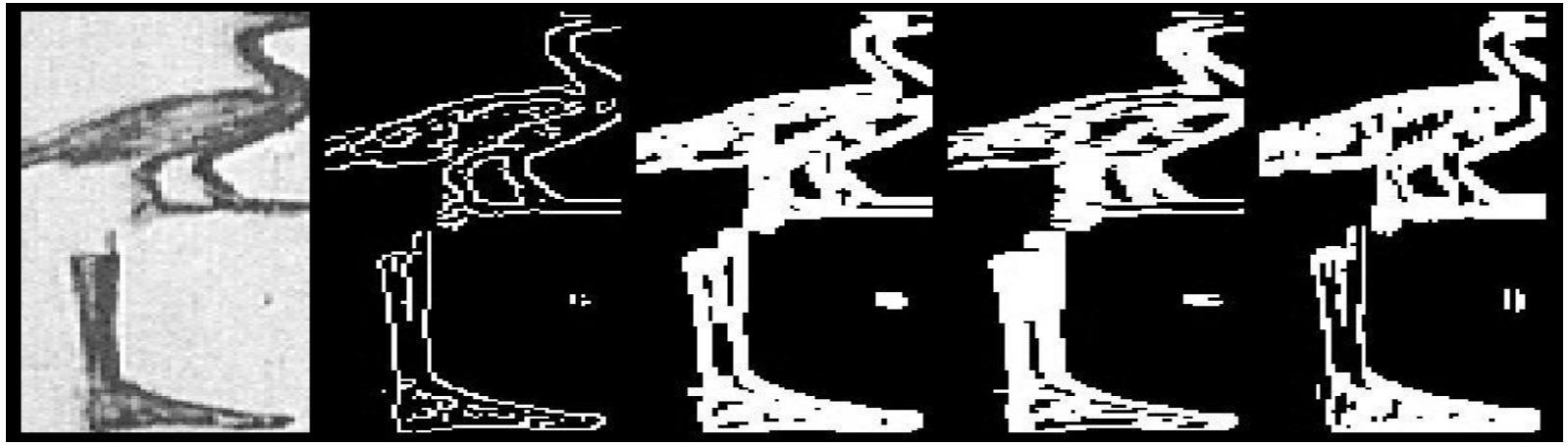
- if we detect more than a contour, we will get their reading order according to the cartouche and the whole image.



Overcoming dilation problems

- While rerunning cartouches detection we will also try to fix the problem that is caused by dilation, which is detecting very near characters as one character.

- we will use a 5x1 kernel of ones (vertical kernel) to apply dilation and use a 1x5 kernel of ones (horizontal kernel) to apply dilation, then try to rerun the cv2.findContours each time and we will compare which one produced a bigger number of contour and we will go with it



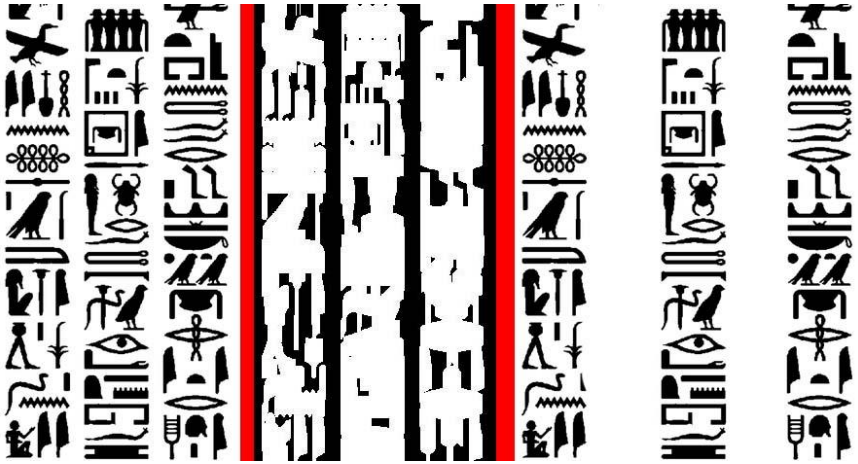
(greyscale, canny edges, dilated canny edges by 3x3 kernel, dilated canny edges by Horizontal kernel(1x5), dilated canny edges by Vertical kernel(5x1)) Images

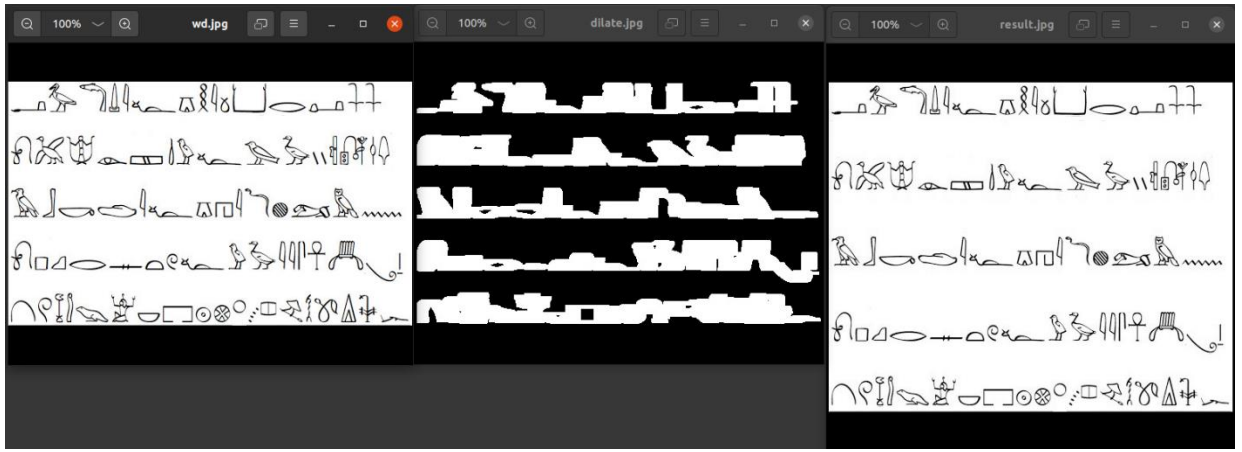
Get contours no lines

- It works with the same manner as get contours function but in this function the image passed has no lines so it's either an image that contains only a single line of text, or maybe it has several lines of text but without a separating line; which makes the detection of the optimal reading order harder than if there is separating lines.
- Although its not common issue; because most of hieroglyphics have lines, but it still exists (e.g., Rosetta stone ,some digital designs).

Detect Spaces

- We can use Detect Spaces function to detect and stretch spaces to try to get our optimal reading order, but we will not implement it in the application since it have good results for digital and high quality images, but it can cause confusion when low quality Images are used
  - First, we create a horizontal (1x10) and vertical(10x1) shaped kernel and dilate to connect the words of each line into a single contour.
  - Then, we extract each line contour and we sort them, and extract each line ROI (Return on Investment)
  - After that, we can append white space between each line contour, to increase the spaces between each line to increase the accuracy and ease the detection afterwards.



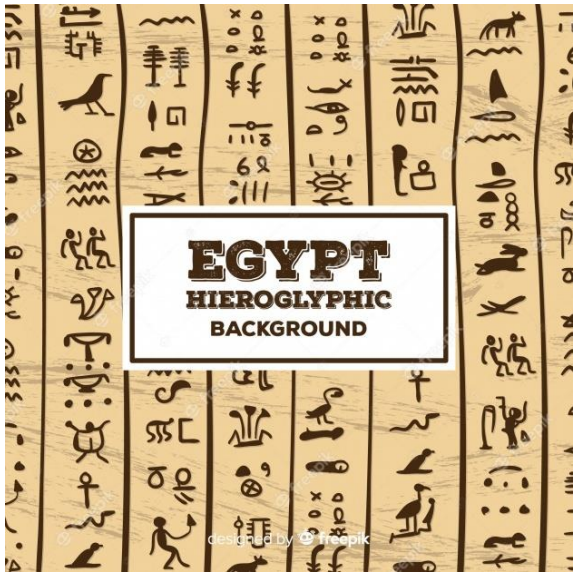


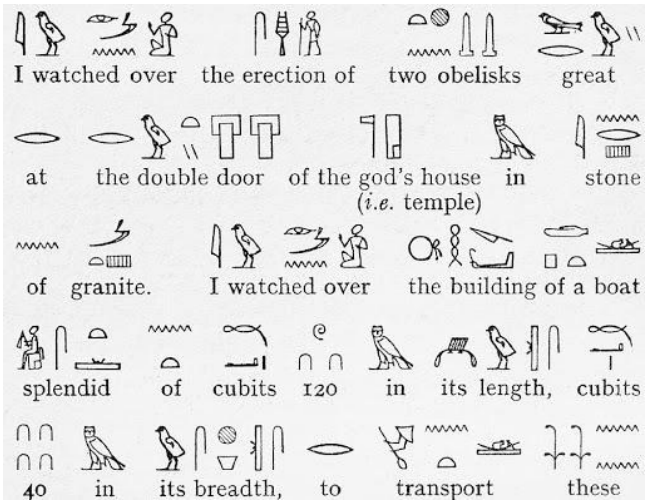
Future Work

Detect other non hieroglyphic languages

- Sometimes the image may contain more than a language and the user may not crop the hieroglyphic part only which may cause confusion since it will be detected as a contour by this module and passed to the other modules as if it is a hieroglyphic character.

- So, we can overcome this problem by using the libraries for other languages like Tesseract and EasyOCR to check if the image contains other languages to exclude this part from the image before starting the detection.

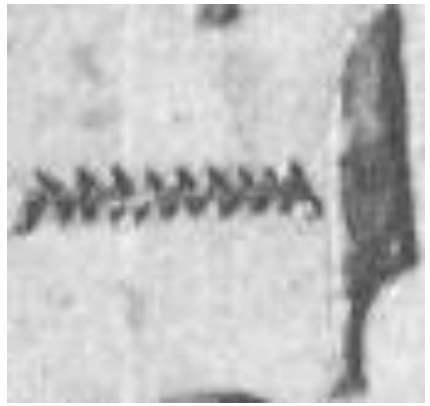


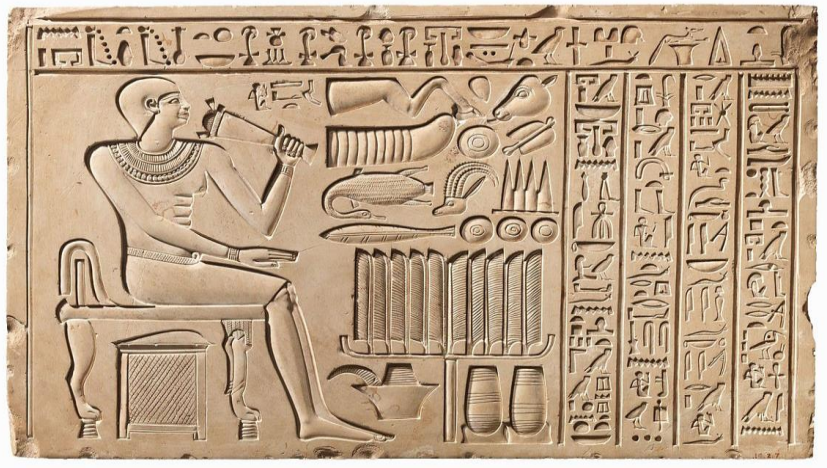


Future Work

Use The User Help

- In some cases It will be super hard and in other cases it may be confusing to get the best results without human feedback.
- So the user can help in the detection process to achieve the best results





Future Work

Use Location Information

- Knowing the temple or the museum the user trying to translate the Image from will be a great help , especially when have a good data about this place like which family this temple belongs too ,and using pre translated texts to get the best results.

Rerun and Change the Parameters

- We can rerun the module several times and every time we run it we can change the used parameters (e.g. canny edge upper and lower threshold, minimum accepted area, ...) to get different results and use the highest voted results.

Phase Four

Glyphs Classification

Table

- Introduction
- Software Components
  - Image
  - Dataset
  - Model
- Results
  - Model evaluation
  - examples

Introduction

- Morris Franklen's "Automatic Egyptian hieroglyph recognition by retrieving images as texts" from university of Amsterdam
  - Dataset
  - Comparison (Classifier+Descriptor)
- Classifiers : Ransac, BOW (bag of words).
- Descriptors : self similarity, Shape Context, HOG, HOOSS, HOOSC.

Introduction

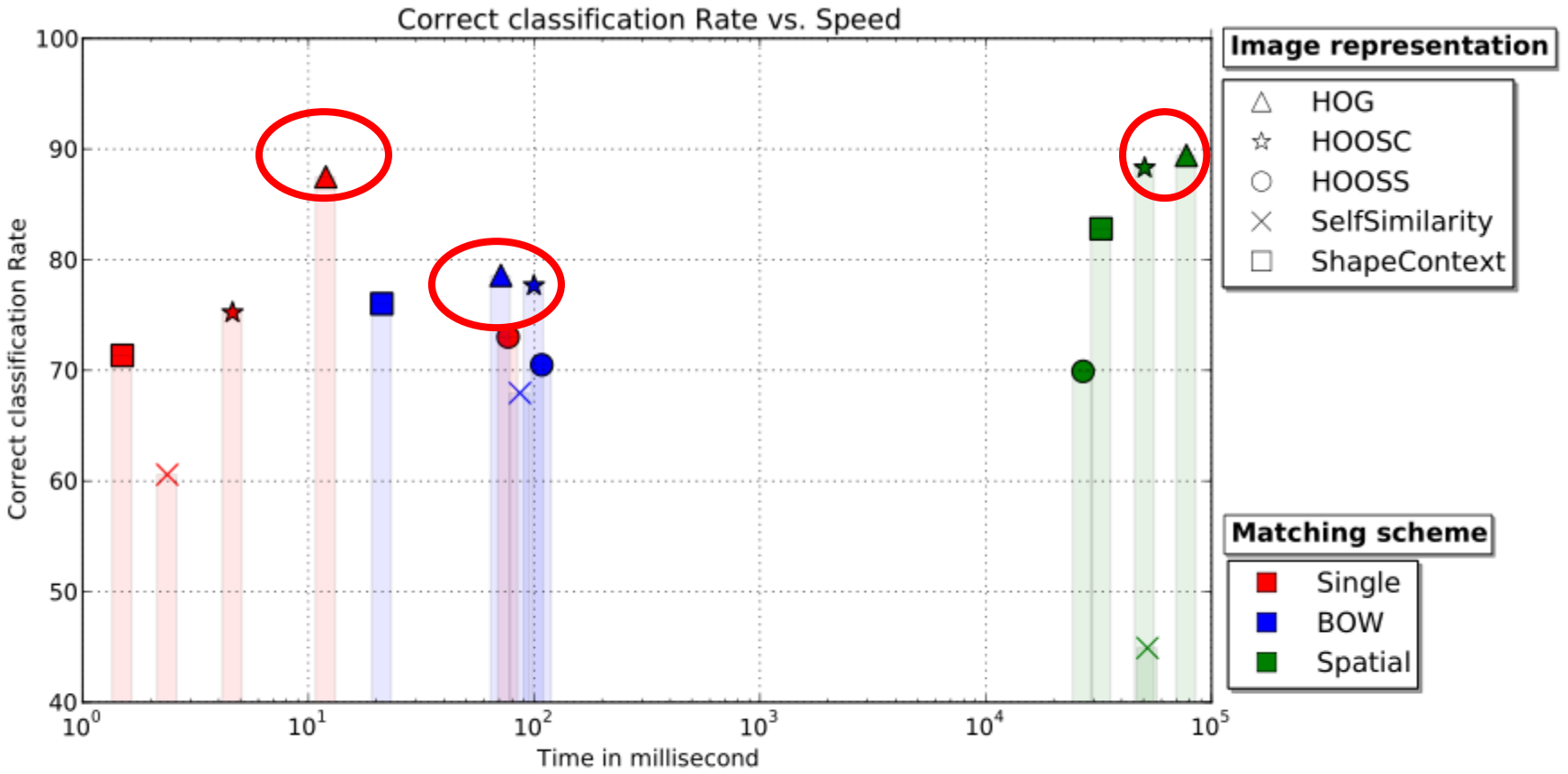
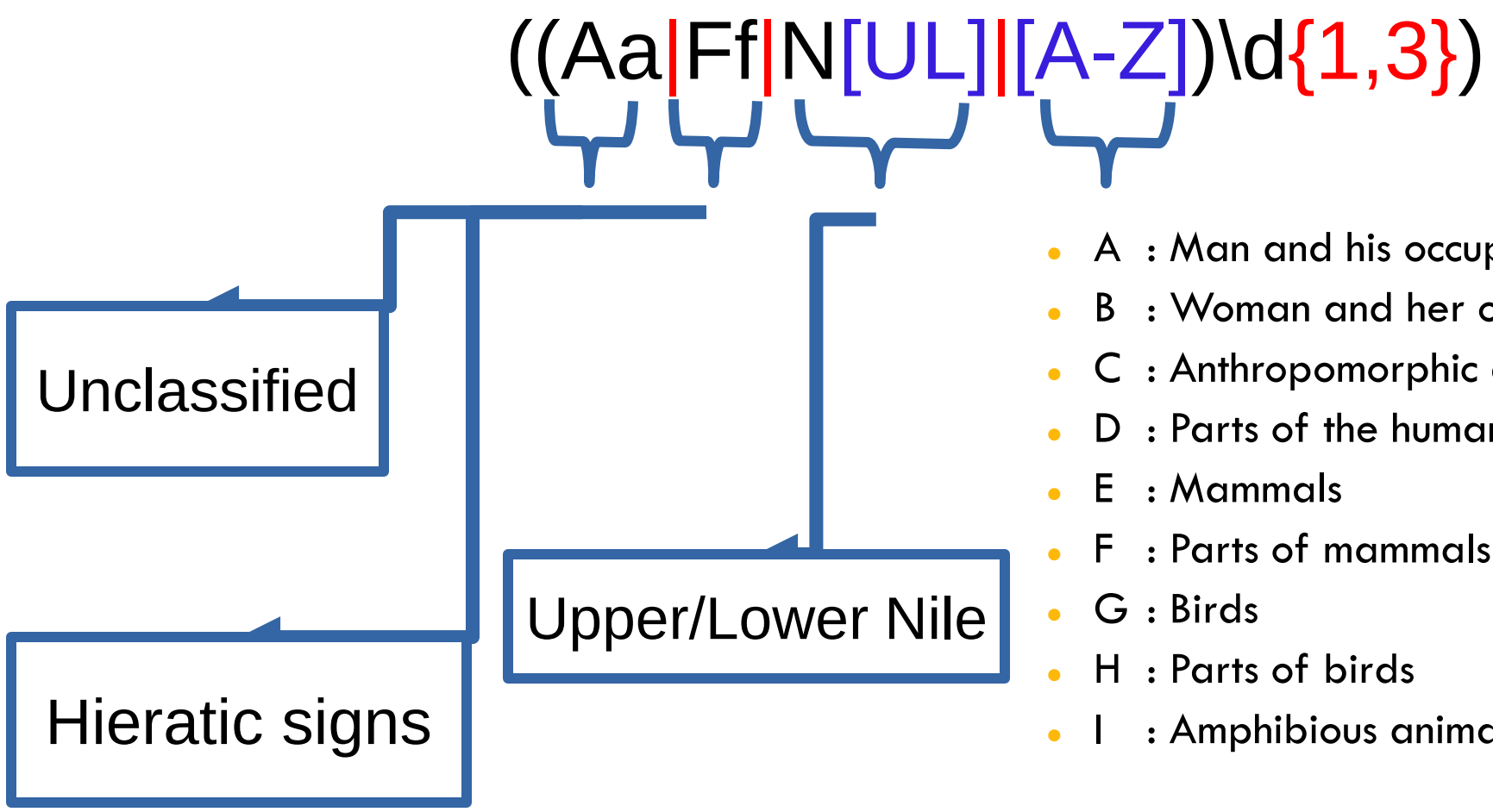


Figure 7.4: Correct classification rate (%) plotted against the average processing time in milliseconds.

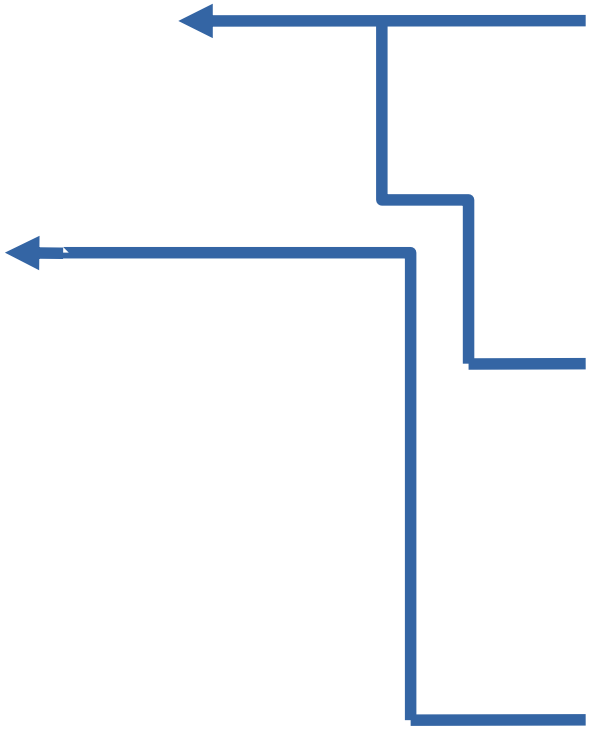
Intro (Gardiner's Codes)



- A : Man and his occupations
- B : Woman and her occupations
- C : Anthropomorphic deities
- D : Parts of the human body
- E : Mammals
- F : Parts of mammals
- G : Birds
- H : Parts of birds
- I : Amphibious animals, reptiles, etc.

Overview (software components)

- Glyph (Image)
- Dataset
- Learning Model



1. Input limiting
    1. Gray scale
    2. size
2. Features extraction (spatial,edges)
    1. Edge detection
    2. HOG
3. Classification
    1. SVM

Why software components ?

- Reading Dataset takes :
  - long time
  - memory
- Deploying/sharing/reusing machine learning model

Image component

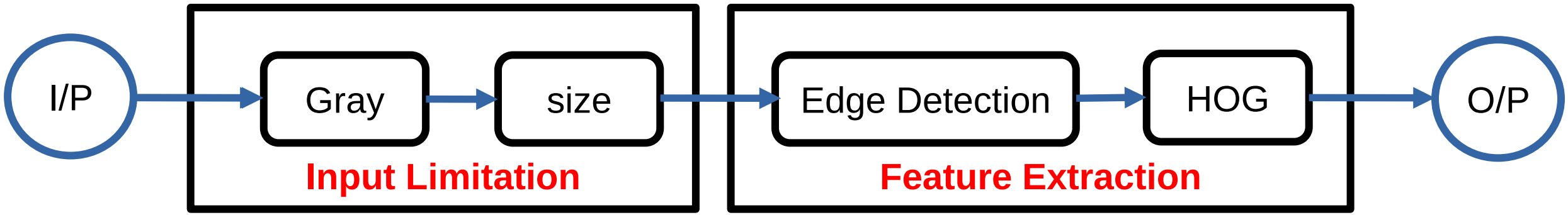
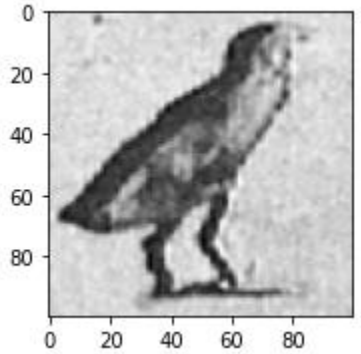
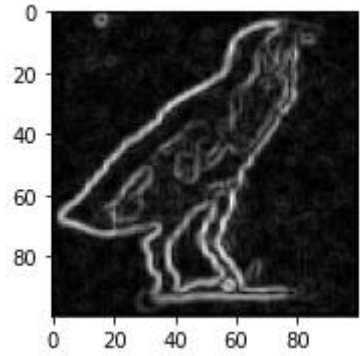
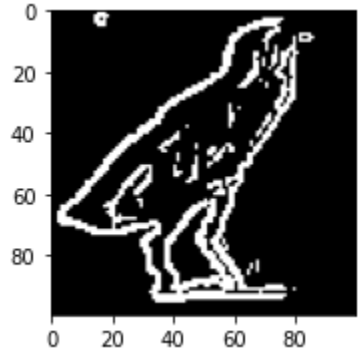
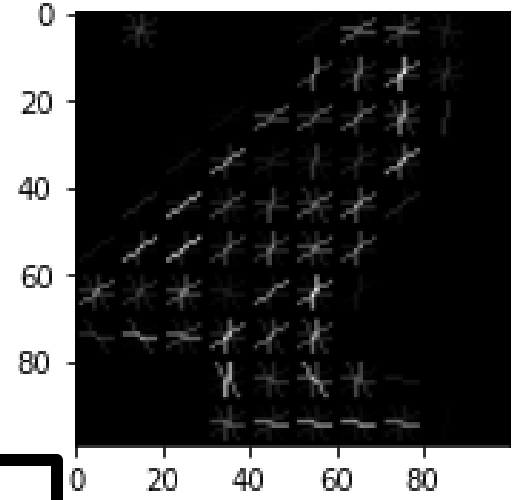


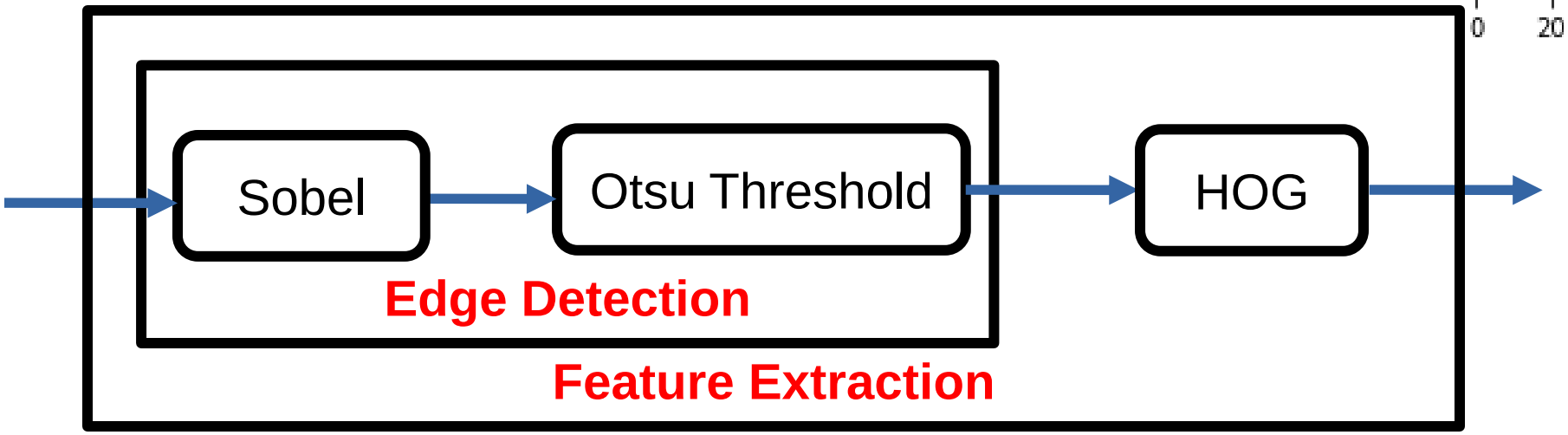
Image component (Feature extraction)



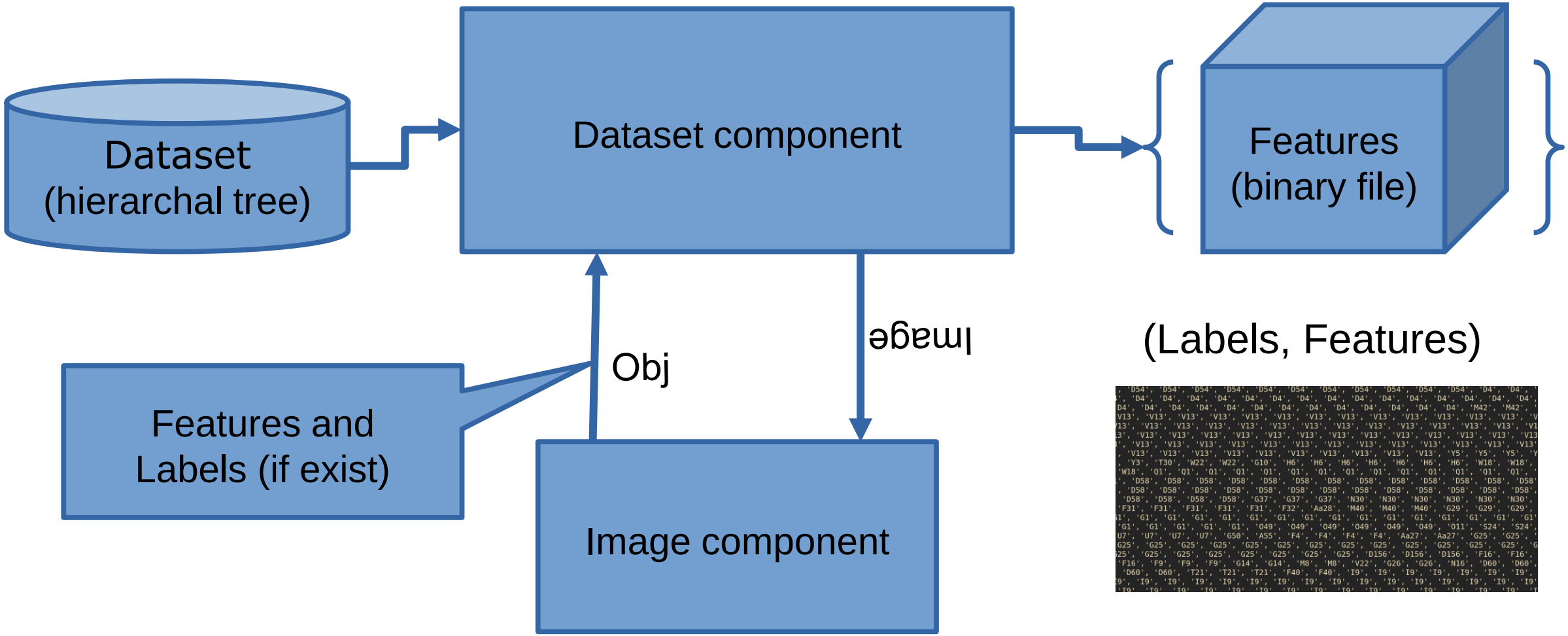




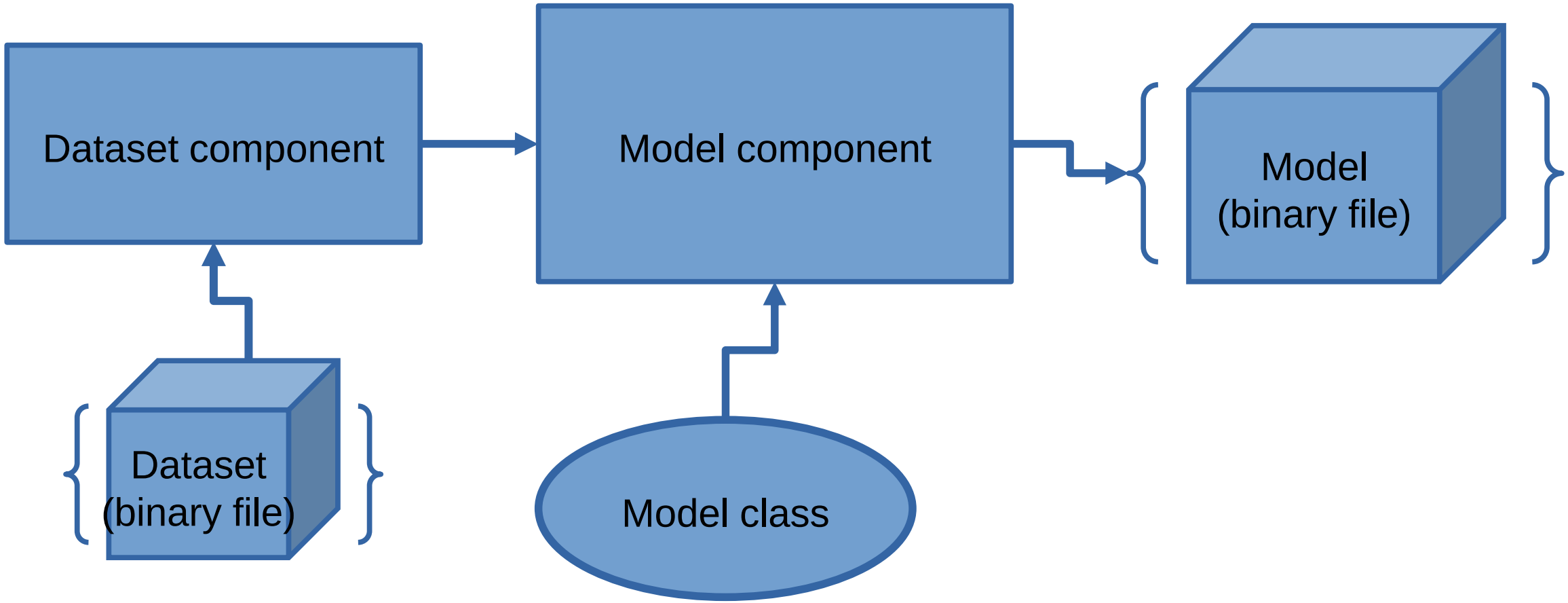




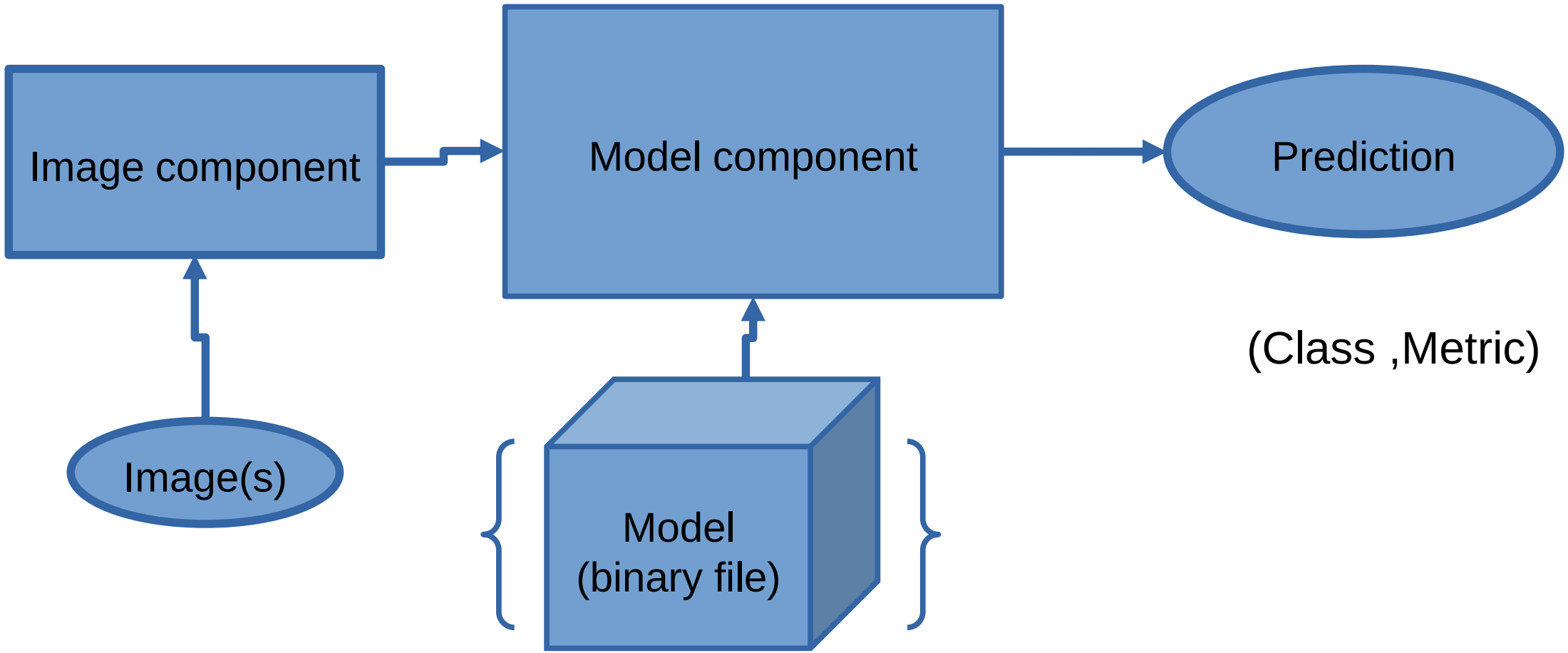
Dataset component (Compiling Features)



Model Component : Training



Model Component : Deploying/Use

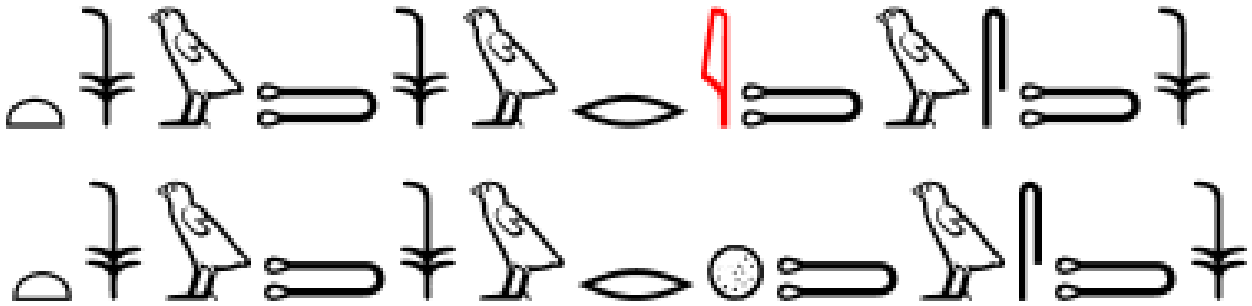


Results : Model Evaluation

- Accuracy (70/30) : 96%

Performance	
Input size (images)	Time (ms)
20	54.75
25	87.40
50	128.11
100	366.57

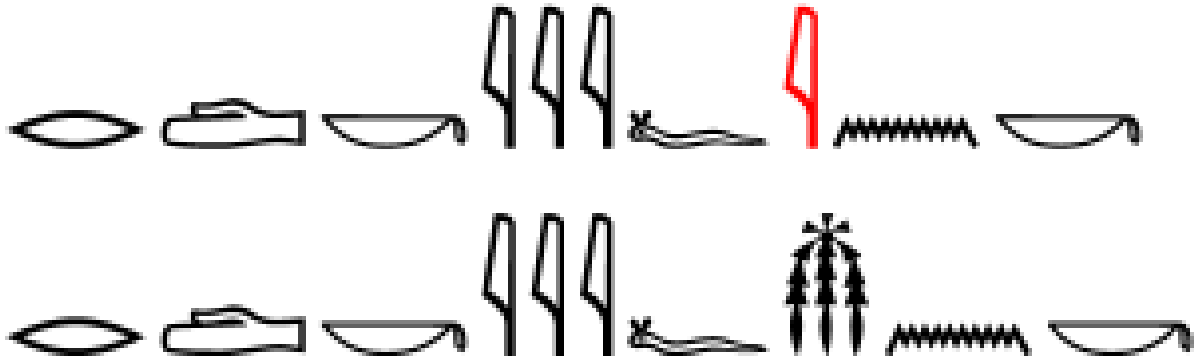
Examples (wall writing from dataset)



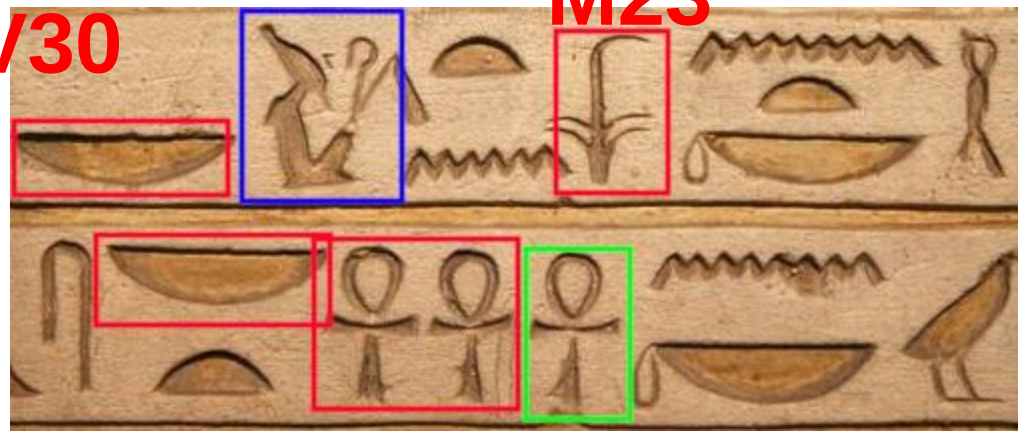
O50

Examples (wall writing from dataset)

F3
1

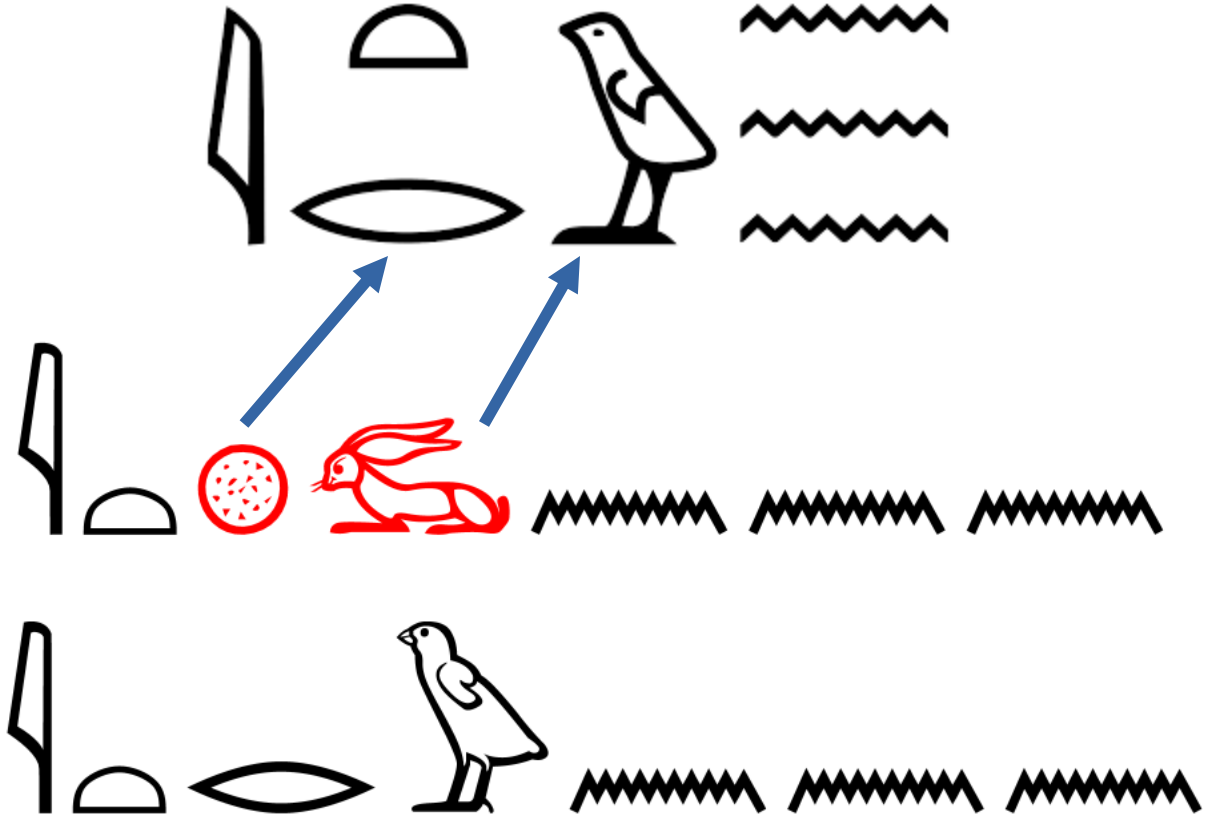


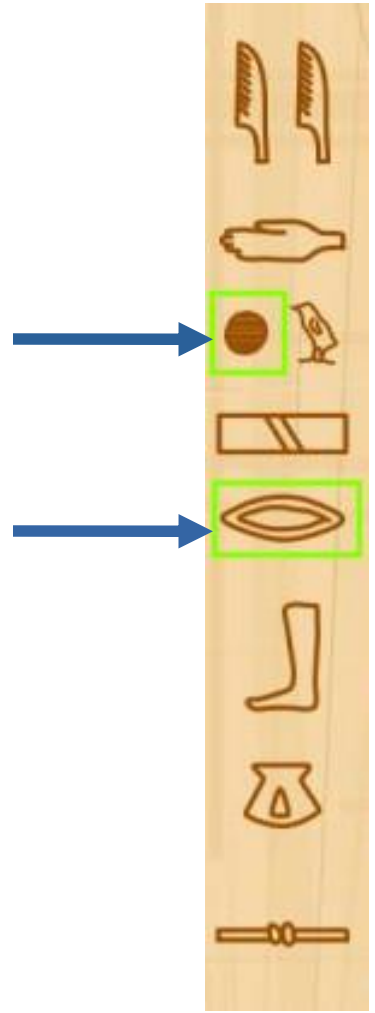
Examples (wall writing outside dataset)



- Misclassified
- Not in the Dataset
- Wrongly detected by CV

Examples (digital hieroglyphs)





Phase Five

Translation

Lets start with project Results

```
['Y3,A1', 'M17,N29,D21']
['NN', 'JJ']
['scribe excellent']
['JJ', 'NN']
['excellent scribe']

In [20]:
```

History log | IPython console

```
['F35,I9,D21', 'X1,Z7', 'V28,N35,D36', 'A1']
['JJ', 'PRP', 'IN', 'PRP']
['good you with me']
['PRP', 'VBP', 'JJ', 'IN', 'PRP']
['you are good with me']

In [21]:
```

History log | IPython console

Results

```
['O34,A1,Z1', 'F35']
['NN', 'NN']
['man good']
['NN', 'VBP', 'NN']
['man is good']

In [13]:
```

History log | IPython console

```
['O34,X1,B1', 'G17', 'O1,Z1']
['NN', 'IN', 'NN']
['woman in house']
['NN', 'VBP', 'IN', 'NN']
['woman is in house']

In [14]:
```

History log | IPython console

Results

```
['N35,Z2', 'G17', 'O1,Z1']
['PRP', 'IN', 'NN']
['we in house']
['PRP', 'VBP', 'IN', 'NN']
['we are in house']

In [16]:
```

History log | IPython console

```
['G47,X1,Z1', 'G17', 'O49,X1,Z1']
['NN', 'IN', 'NN']
['vizier in city']
['NN', 'VBP', 'IN', 'NN']
['vizier is in city']

In [17]:
```

History log | IPython console

Phase Five

- There are many steps to be done before we can translate the Hieroglyphic language to English language and these steps manipulate the sentence to be ready to translation and there are many differences in the structure of the sentence in the two languages.

1) As in many ancient writing systems, words are not separated by blanks or punctuation marks, there is no full-stop (.) between sentences or even a space
2) Probably in all languages, a word does not necessarily have one meaning but in Hieroglyph the word may have 15 meaning or more as it is a language with symbolic letters that describe a natural elements in life so multiple meaning of the same word is widespread.

Phase Five

- Tour guides translation is also not the same and every guide translate on his own explanation style, however there are base words in the sentence that determine most of the translation.
- Also we must take care of grammar differences between the two languages.

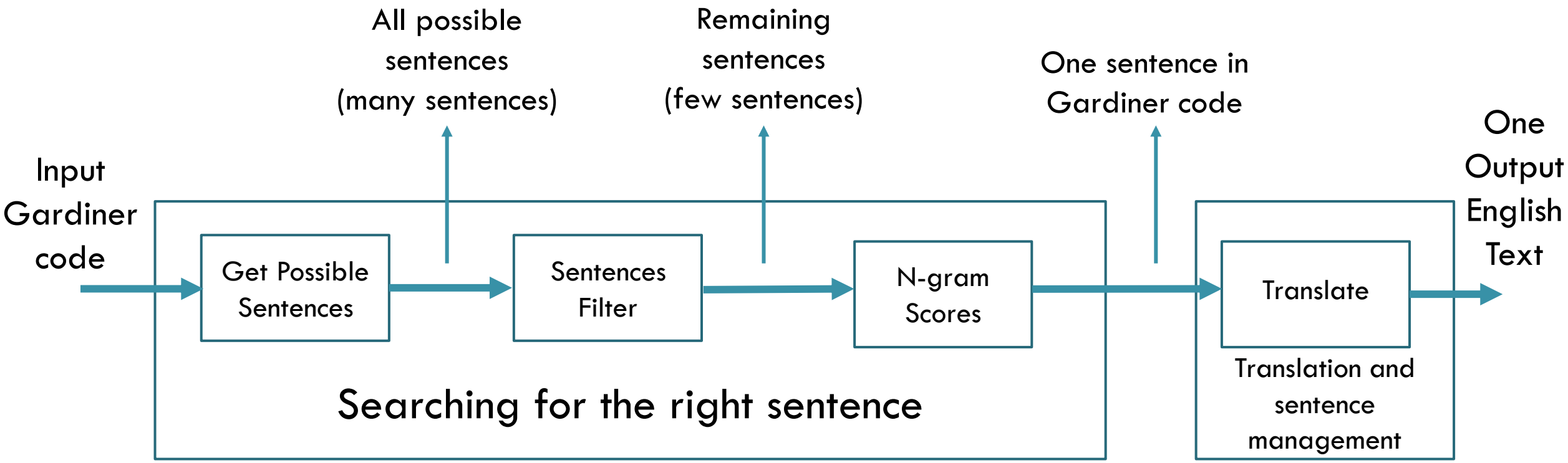
Phase Five

- Phase Five is classified into 5 classes:

1) Get Possible Sentences
2) Sentences Filter
3) N-gram Scores
4) Translate
5) Grammar Correction

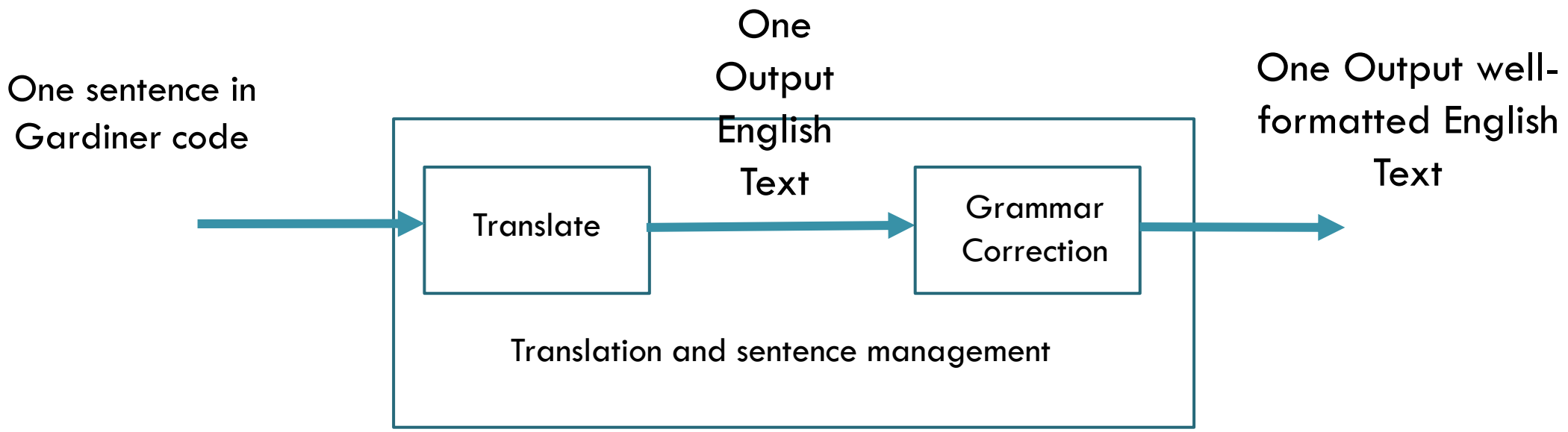
Phase Five

- Block Diagram to describe this phase

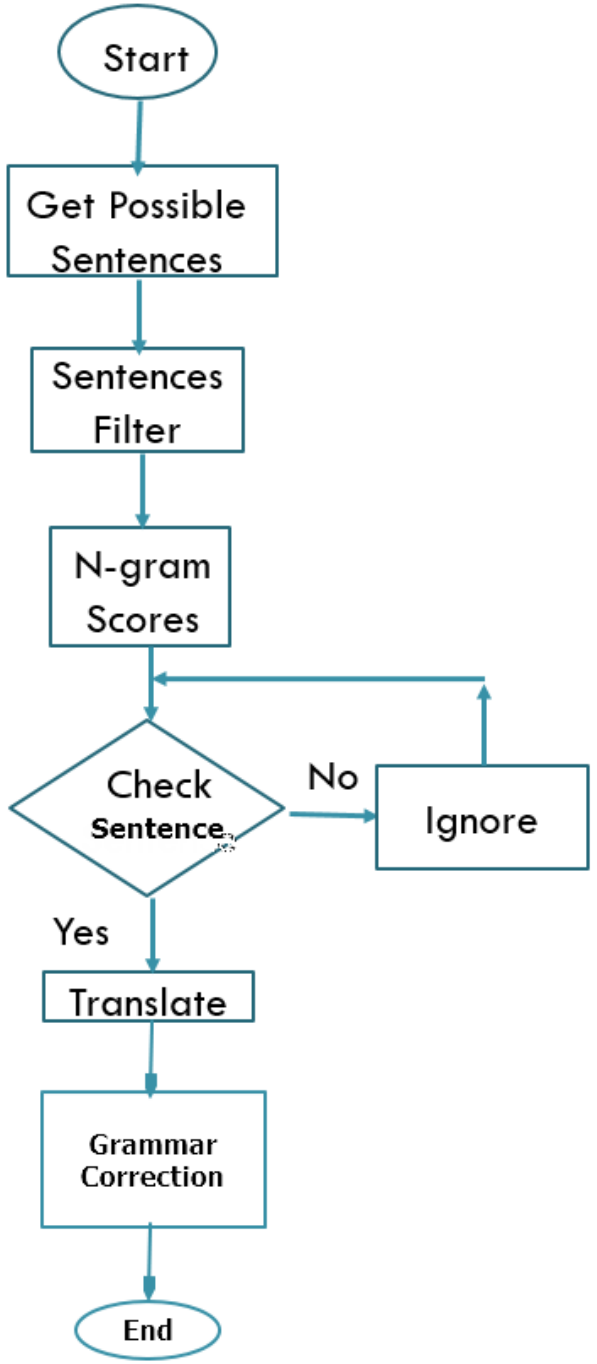


Phase Five

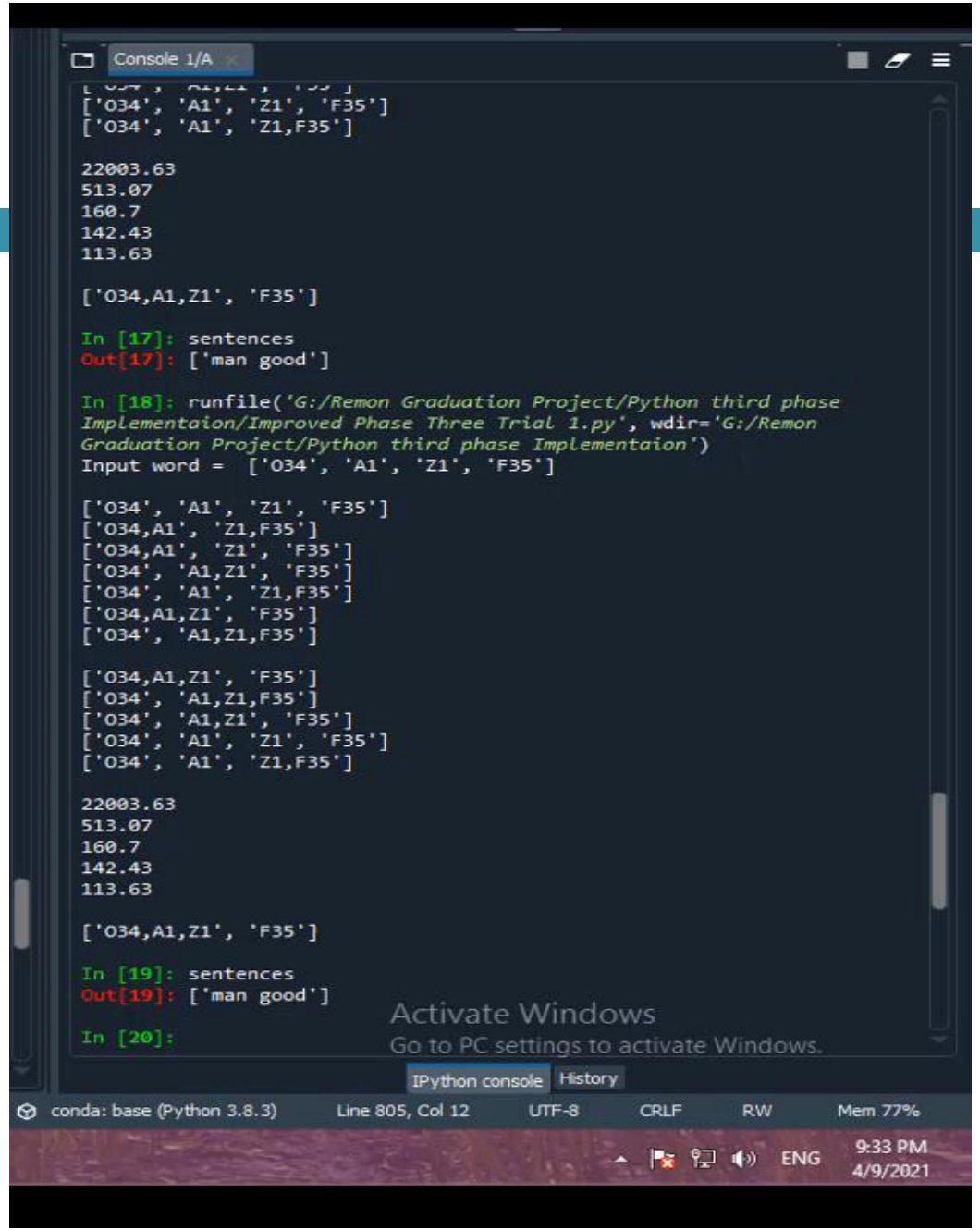
- Block Diagram to describe this phase



Phase Five Flow Graph



Prototype Video



Suggested Sentences Class

- There are no spaces in Hieroglyphic language so every word is directly attached to the next word. The idea behind this phase came from the following: Imagine English had no spaces so we test each combination of words to see which combination is valid.
- For example, the sentence “Thegladiatorisstrong”. Trying to take “t” or “th” would fail, but taking “the” would succeed.
- Moving to the next word, “glad” and “gladiator” would succeed. But if “glad” was picked, it would ruin the rest of the sentence

Suggested Sentences Class

1) Reading any number of letters using two recursive functions one call the other

    1. get_Text_Bigram()
    2. get_Text_Trigram()

Get Possible Sentences Class

Work:

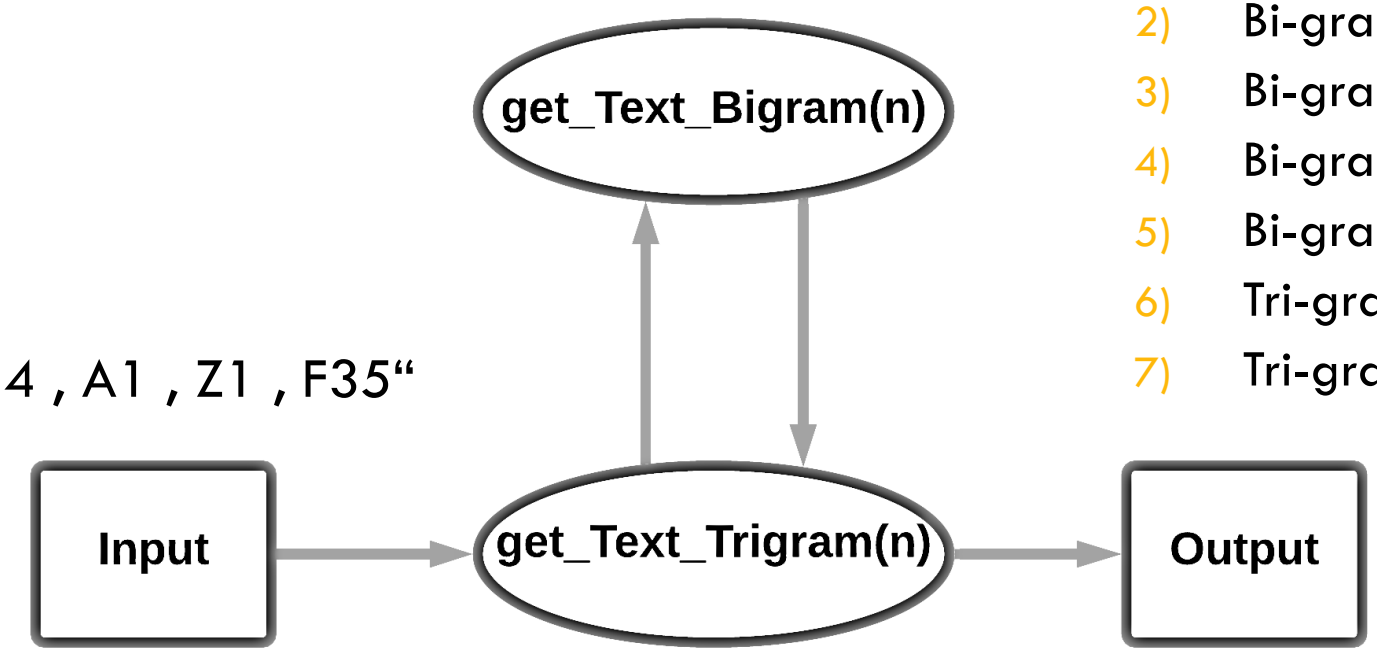
Possible sentences:

- Uni-gram: O34 – A1 – Z1 – F35 — 4 Words
- Bi-grams: (O34,A1) – Z1 – F35 — 3 Words
- Tri-grams: (O34,A1,Z1) – F35 — 2 Words

Get Possible Sentences Class

Work:

- Input text: "O34 , A1 , Z1 , F35“

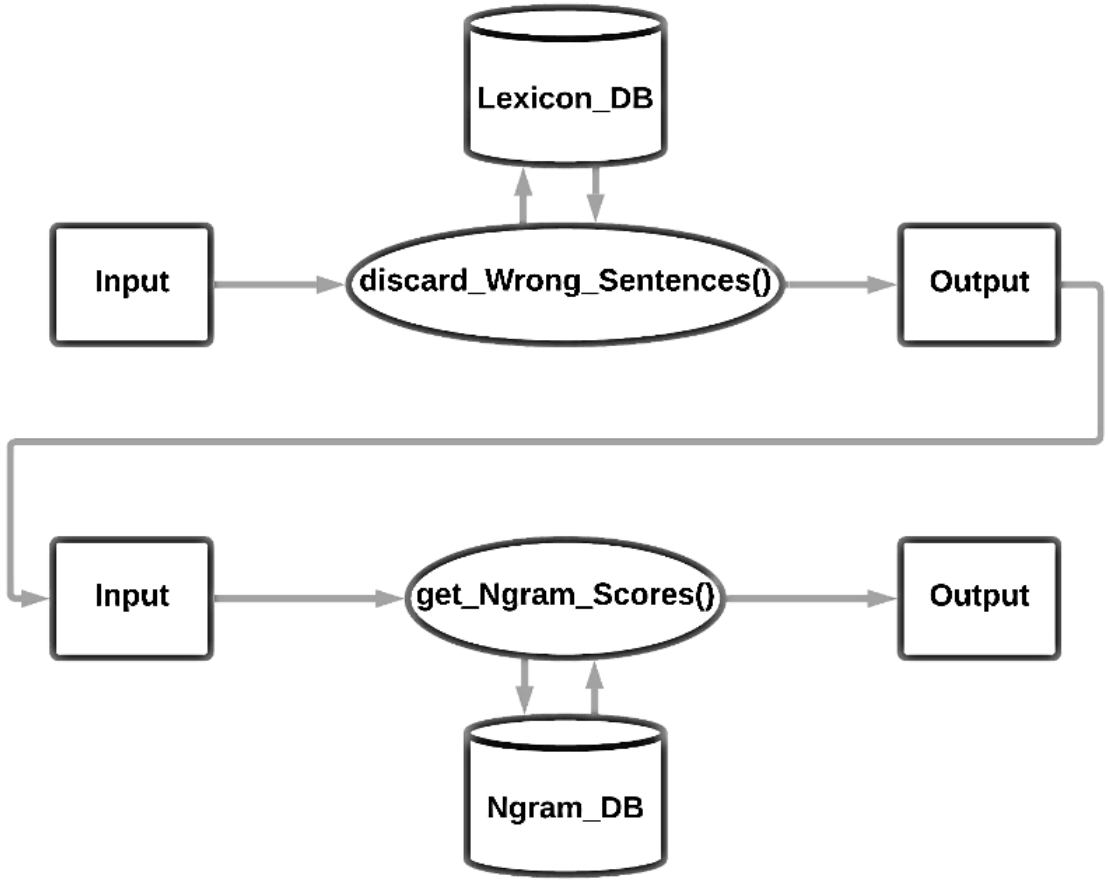


Possible sentences:

1) Uni-gram: O34 – A1 – Z1 – F35
2) Bi-grams: (O34,A1) – Z1 – F35
3) Bi-grams: O34 – (A1,Z1) – F35
4) Bi-grams: O34 – A1 – (Z1,F35)
5) Bi-grams: (O34,A1) – (Z1,F35)
6) Tri-grams: (O34,A1,Z1) – F35
7) Tri-grams: O34 – (A1,Z1,F35)

Heuristic Search and N-gram Scores

- Block Diagram to describe the process



Heuristic Search - Optimization

- A Heuristic is a technique to solve a problem faster than classic methods, or to find an approximate solution when classic methods cannot. This is a kind of a shortcut as we often trade one of optimality, completeness, accuracy, or precision for speed.
- We can go and search for all suggested sentences scores but we will take few minutes to do that task so we make use of Heuristic search concept to discard sentences that are symmetric. This symmetry is in the words as we have many sentences starts and contains the same word and if this word has no meaning in the lexicon we can discard all of these wrong sentences which greatly fasten and ease the next step, n-gram scores, computation and that means that this step is for optimization.

(O34,A1) – Z1 – F35

(O34,A1) – (Z1,F35)

The first word is the same in both sentences

N-gram Probabilistic Language Model

- An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a $(n-1)$–order Markov model. In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs but in our application items are letters or glyphs. The n-grams typically are collected from a text or speech corpus.

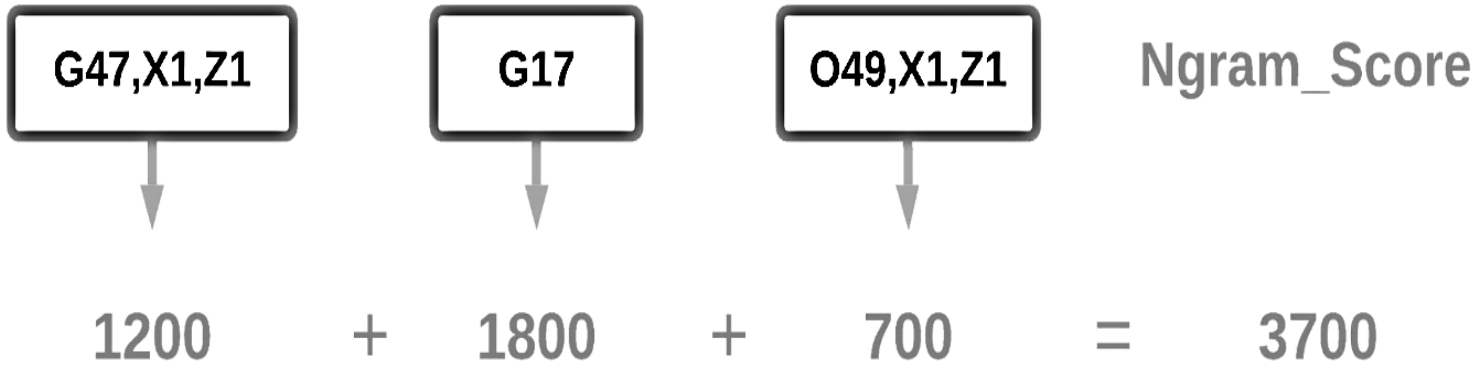
N-gram Probabilistic Language Model

- The main use of N-gram language model in our project is to add spaces between glyphs according to the number of words in the sentence.
- The scoring system is really efficient and give very high and accurate results. Its idea is that each sentence of the suggested sentences consists of some words so each word in each sentence will get a probability score then we will add all the word scores in the same sentence so each sentence will get a score. Sentence with highest score will be the right candidate.

N-gram Probabilistic Language Model

Example : human

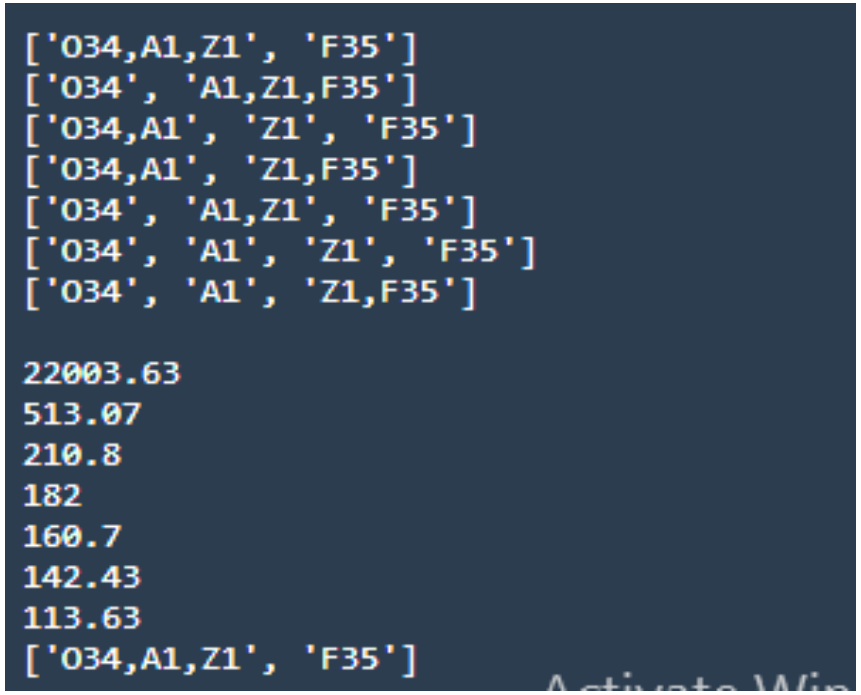
- In uni-gram: h-u-m-a-n
- In bi-grams: hu-um-ma-an
- In tri-grams: hum-uma-man



- Scores will be given to all possible sentences.

N-gram Probabilistic Language Model

- As we can see there are a huge differences in scores as the first score is very high related to the second or the other score and that ensures that the scoring system is successful and has a very high accuracy



Translate Class

- The sentences will be ordered in descending order and the first sentence in the list or the highest score sentence is the right candidate.
- After that we will take each word in the result sentence and check for its meaning in the lexicon to get the right meaning.
- the code will ignore any sentence that has null word meaning

Grammar Correction

- After translation of the Hieroglyphic sentence to English sentence we must take care of some grammar rules that is different between the two languages and some restructure processes required to organize the English sentence in a good format.

Rule 1

- In phase 1, which is “Adjectives as modifiers”. It looks for adjectives modifying a noun. In English, when adjectives come as modifiers to nouns, they precede the noun, for example “the excellent plan”.
- On the other hand, in Hieroglyphic, the adjective follows the noun, literally “the plan excellent”.
- Then their order is switched so that the noun follows the adjective(s).

Rule 2

- In this phase, “Pronouns” phase, a very common use of pronouns is with nouns. In English, when used with nouns, pronouns always precede nouns, for example, “my book”, “this man” and “their house”. In Hieroglyphic, there is no single rule for all pronouns. Furthermore in case of a pronoun that follows a noun, the order is switched.

Rule 3

- “Consecutive nouns” phase, the main aim is to look for any consecutive nouns and try to guess what kind of relationship exists between the nouns. When exactly two nouns come together, it is hard to tell their relationship so to solve this problem, if there is no verb in the sentence this relationship could result in verb to be as in Hieroglyph verb to be is sometimes removed or it could result in “and/or/of” that is added between the nouns.

Singular and Plural

- The translation system also detect the singular and plural nouns to add right verb to be when needed

```
['N35,Z2', 'G17', 'O1,Z1']
['PRP', 'IN', 'NN']
['we in house']
['PRP', 'VBP', 'IN', 'NN']
['we are in house']

In [16]:
```

History log | IPython console

```
['G47,X1,Z1', 'G17', 'O49,X1,Z1']
['NN', 'IN', 'NN']
['vizier in city']
['NN', 'VBP', 'IN', 'NN']
['vizier is in city']

In [17]:
```

History log | IPython console

Augmented Reality

- We used AR that is presented by Unity game engine. The built project is then presented on Flutter app so we managed to open Unity AR as real-time application running on a widget of Flutter application.
- In Unity we managed to use Image Tracking to detect and track Cartouches of kings in ancient Egypt and after detection a 3d statue will appear and a voice talking about the king mentioned in the Cartouche with an addition of some sound effects. Image.

Augmented Reality

- A great achievement is to connect two complicated programs as a game engine that uses C# as its programming language and new framework like Flutter that uses dart programing language. This connection requires complicated communication between the two programs so that they can work smoothly with each other and this communication is managed by the two programs.