Software Requirement Specification Document for Aegyptos Hieroglyphs Translation Mobile Application

Shehab ElDin Mohsen, Reem Mansour, Ahmed Bassem, Basma Dessouky
Supervised by: Dr. Taraggy, Dr. Samira Refaat

May 13, 2023

Table 1: Document version history

Version	**Date**	**Reason for Change**
1.0	6-Dec-2022	SRS First version's specifications are defined.
2.0	10-Dec-2022	New sections have been made
3.0	17-Dec-2022	Last edit points made
4.0	30-Apr-2023	Last updates on the SRS

GitHub Mobile App: https://github.com/ahmedbassemaly/Aegyptos

GitHub Backend Code: https://github.com/ShehabEMohsen/Aegyptos.git

Contents

Abstract

Hieroglyphs have always been fascinating with their stories and the ability to be read in several ways rather than one, which is a challenge in itself to be translated to modern languages. However, the full experience of getting familiar with these captivating historical stories will not be fulfilled if this translation was not on spot, and hence the tough challenge is met. The idea of Aegyptos was brought up to help learn how to read Hieroglyphs with respect to how the same symbol can have two different meanings according to the context, and also help to pronounce them by using a tool at the palm of users' hands with just their phones' live cameras. Performing segmentation on the symbols using different segmentation techniques, while a light-weight CNN is used for classification with the help of an API to translate the script into a language understood by the user.

1 Introduction

1.1 Purpose of this document

The purpose of this document is to analyze and break down the software requirements for developing a mobile application for translating hieroglyphs. It will state and describe the functional requirements alongside the expected behavior that the application should take. The document also outlines the needs and constraints of the mobile application. The target audience for this document are the stakeholders for the project and any developer wanting to continue to work on it.

1.2 Scope of this document

The scope of this document is to show the requirements and specifications of this mobile application. Explaining how the project will provide comfort with the help of showing the similar systems and applications. It also represents the limitations, data design and the starting class diagram as it is bound to change if needed.

1.3 Business Context

One of the most important income sources in Egypt is tourism; as Egypt has a large number of tourist sites for its ancient history going back thousands of years. From 2017 to 2020, the average number of tourists visiting Egypt was 10.1 Billion tourists[8] as it increased year after year except for 2020 due to the COVID crisis. Therefore having an application that helps tourists and even local Egyptians that are interested in visiting the ancient Egyptian sites would make them go to more sites to understand more and learn more stories and meanings and it will make their visit to these sites much more interesting and comfortable not having to rely on a tour guide.

2 Similar Systems

2.1 Academic

Translation of Hieroglyphic characters has always been quite a challenge. It requires a lot of studies with the help of Egyptologists to be translated according to the meaning of the semantics and not just the characters as characters. Several researchers have reached quite some great results regarding the process of automatic translation of Hieroglyphs. [10]

Plecher et al. [4] discussed teaching Hieroglyphs through a serious game with the AR usage. They have used a dataset of more than 6,800 manually interpreted labels for 35 different hieroglyphs and while the model has been trained on around 10,000 glyphs, by using Data Augmentation, gaining some amazing results. However, the number of classes was relatively small. They used Faster Region-Based Convolutional Network (F-RCN) which turned out to be slower than Single Shot Detection (SSD) and You Only Look Once (YOLO) techniques. All of their work resulted in a mobile application called ARsinoë.

Barucci et al. [5] tackle automatic hieroglyphs translation through merging two datasets, one used in [**franken2013automatic**] of 4,310 gray-scale images, and the other was manually collected, resulting in 1,310 colored images, forming 26 classes to give combinations of words and sounds [**gardiner1957egyptian**]. The images were cut from scanned photos, with an irregular contour, and the final images were obtained by filling their rectangular bounding box with a white background. However, if they used a stronger segmentation technique, it may have been better. They used Glyphnet as a classification technique resulting in astounding classification .

El-nabawy et al. [3] concentrated on translated hieroglyphs and mapping them to English. The authors used 700 images with 197 different hieroglyphic characters. However, if they have used larger dataset, they would have reached better results. They used bounding box technique as segmentation. They focus on how the sentences will be read; whether left-to-right, right-to-left, or top-down. They used Gardiner's code as a dictionary for translation. [6]. However, there is still no available software.

2.2 Business Applications

1. **Dua-Khety:** is a mobile application for translating Hieroglyphs by either taking a picture with your phone camera or uploading a picture from your gallery. It also has a feature for searching for symbols using their Gardiner code and get information about the result. The translating part of the application isn't working at all as it either crashes or sends you back to the home screen. [7]

2. **Scriba:** is a mobile application dedicated to translating Hieroglyphs in the shortest processing time possible. The application was published on the app store and the play store for some time but it is now taken down without a sight other than the website promoting it.[9]

3 System Description

3.1 Problem Statement

Translating Hieroglyphs has always been a challenge as it isn't an easy language to understand it can be read from left to right, from right to left, and from top-to-bottom so the application should know in which way should be read. Also when translated, the application needs to show the translation on the user's phone so we can't run a heavy deep learning algorithm as it won't run with efficient time on the user's phones therefore the application has to use a light-weight CNN so it can work efficiently on mobile phones. Most of the time translators are a picture you upload and it takes a long time to process and give you the translation which could be solved by the CNN and using live working cameras like the way QR codes are scanned live. Also most translators require access to the internet so they can work but some known places like some temples and pyramids have no access to internet so an offline part should help with this problem. These are the main challenges that we are facing making them the main focus of our application.

3.2 System Overview

As shown in figure 1 The user will position the camera at the hieroglyphic writing in order to obtain a picture of what he wants to understand from it. After the photo is taken, pre-processing techniques are used to remove any noise from the image, and then the image is segmented to separate the symbols from their backgrounds. This creates a clear image that can be easily translated later after using the Otsu thresholding techniques for segmentation and lightweight CNN (convolutional neural network) as a classification model. After the hieroglyphs have been identified, the text can be translated into either English or Arabic, then the user can have the option to listen to how it is pronounced using text-to-speech and the user also has the choice to explore the application even if there is no network nearby.

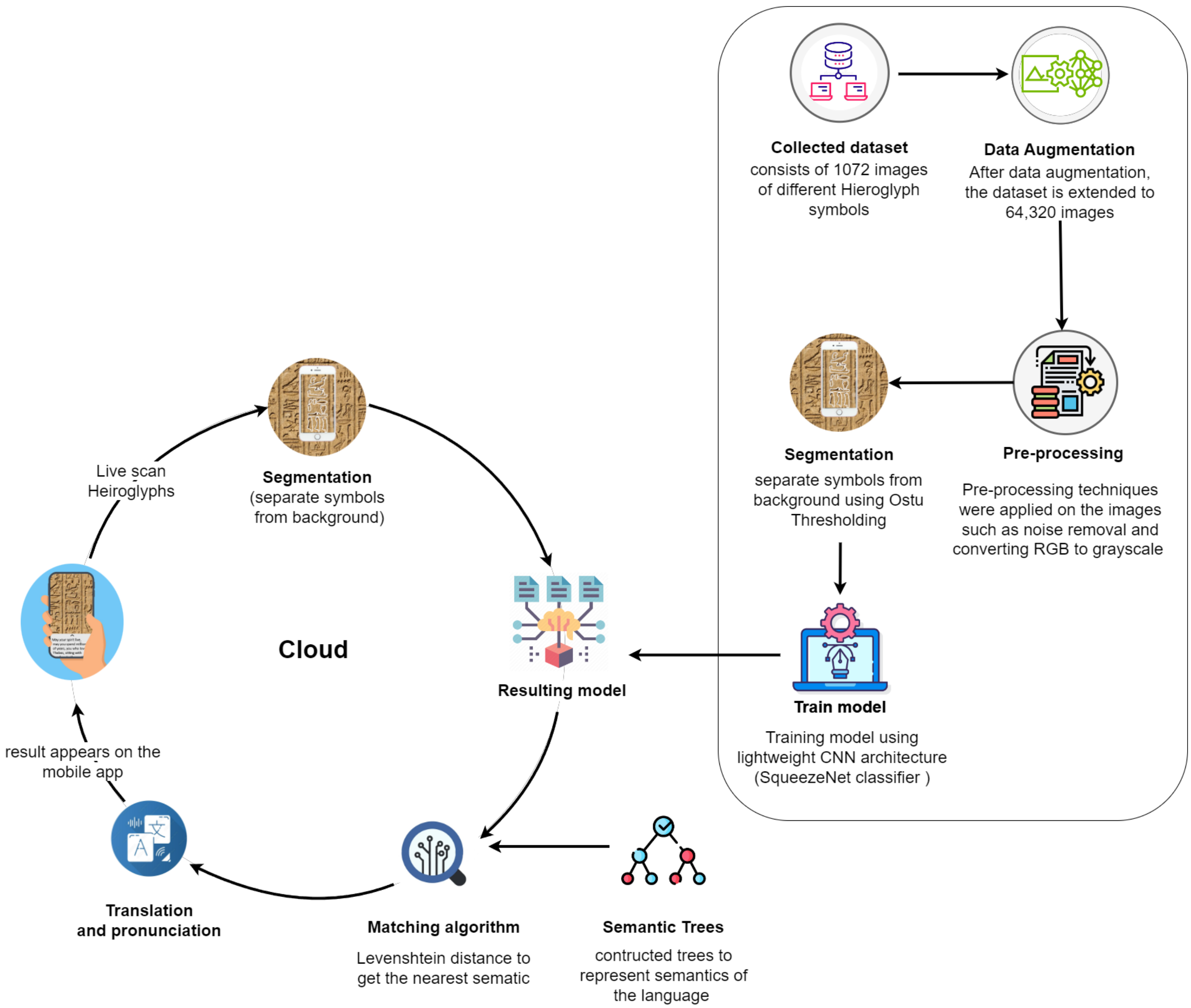

Figure 1: Project Overview Diagram

3.2.1 Otsu Thresholding & Gaussian Blur

Otsu's Thresholding technique is an image segmentation technique that automatically and in a rapid way determines the values for the threshold values by searching for and calculating the values of pixels for each side of the threshold. The input image is a gray-scale image and the values for the threshold and the output would be a binary image segmenting the symbols in the picture[1]. If the intensity of the pixel is greater than the inserted threshold then it is put in the foreground and colored white, while if the intensity of the pixel is lower than the inserted threshold then it is put in the background and colored black[2]. The techniques tried on the two datasets were Contour Detection, Canny Edge Detection, K-means, and Otsu Thresholding. Otsu's Thresholding provided the best output out of all of them. The results are shown in Figures 2 and 3.

Gaussian Blur is a way to smooth an image and remove noise using Gaussian Function. Gaussian Blur was used on the datasets to remove as much noise as possible before applying the Otsu Thresholding technique to the images in intend to improve the results.

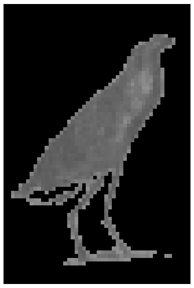

Figure 2: Otsu on the first dataset

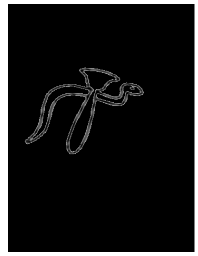

Figure 3: Otsu on the second dataset

3.2.2 Cropping images

After applying the Gaussian Blur and Otsu Thresholding, we obtain a binary image that then we sort the contour from left to right using a Python function called contour.sort_contours. It helps that when an iteration occurs through each contour, we know we have each character/symbol in order. In addition to it we use minimal thresholding after it to remove any additional noise left. After that, the characters/symbols are extracted and become a .png file of its own after looping through the images. The full image and the cropped and extracted characters are shown in Figures 7,5, and 6

Figure 4: Full image with outline boxes

Figure 5: Example 1 of cropped symbol

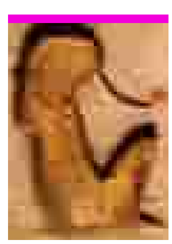

Figure 6: Example 2 of cropped symbol

3.2.3 Matching Algorithm

A matching Algorithm was used with a Python package called Fuzzywuzzy that was created to compute the differences between sequences and patterns using the Levenshtein distance. It figures out how similar two strings are to one another. With a percentage for the similarity to tell you the best match that it could find. As we used it to get the word of a string that consists of multiple Gardiner codes together.

Figure 7: Matching Algorithm

```
match_ratios = process.extract('A021 S029 G017 T034 N035 AA001', str_list, scorer=fuzz.token_sort_ratio)
print(match_ratios)

best_match = process.extractOne('I002 G001 Z002 AA001 D021 E023 G043 P008 A002', str_list, scorer=fuzz.token_sort_ratio)
print(best_match)
```

```
[('A021 S029 G017 T034 N035 AA001', 100, 'Friend'), ('T035 S029 N035 A002', 69, 'Call out'), ('O004 D021 G001 N005', 61, 'Day'), ('T035 N035 K001 T034 G017 D003', 61, 'Skin'),
('I002 G001 Z002 AA001 D021 E023 G043 P008 A002', 100, 'Noisy')
```

3.2.4 SqueezeNet

SqueezeNet is an 18 layers deep convolutional neural network that employs design strategies to reduce the number of parameters. The pre-trained network (on the ImageNet dataset) can classify images into 1000 various object categories. It has the ability to reduce feature map depth which is considered one of its benefits. reduces the amount of computation required by the 3x3 filters in the expanded layer. It increases speed as the 3x3 filter requires nine times as much computation as a 1x1 filter. Therefore, SqueezeNet is an AlexNet-level accuracy with 50 times fewer parameters and less than 0.5 MegaBytes sized model

3.2.5 EfficientNet

EfficientNet is a Convolutional Neural Network that has multiple versions, signifying scalability, starting from EfficientNetB0 until EfficientNetB7. The total number of layers of EfficientNetB0 is 237, while on reaching EfficientNetB7, the total number of layers reaches 813. Scalability is very important when it comes to deep learning classifiers. To scale up models in a quick and easy way, EfficientNet uses a method called a compound coefficient. Compound scaling uniformly scales each dimension with a predetermined fixed set of scaling coefficients as opposed to randomly increasing width, depth, or resolution. The developers of EfficientNet created seven models in different dimensions using the scaling approach and AutoML, outperforming most convolutional neural networks' state-of-the-art accuracy while operating far more effectively. The base EfficientNet-B0 network is based on the inverted bottleneck residual blocks of MobileNetV2, in addition to squeeze-and-excitation blocks. EfficientNets also transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets, with an order of magnitude fewer parameters

3.3 System Scope

Many people are curious to know about the hieroglyphic symbols, therefore the Aegyptos application can help you do that regardless of your age, whether you're a student or even an adult guy, you can use the Aegyptos application.

- Translate the Hieroglyphics pictures to the English language to be able to understand them.
- To make it simple for users to use it offline if an internet connection is unavailable.
- Lightweight Convolutional Neural Network (CNN) will be used to classify the hieroglyphs.
- Using noise removal followed by image segmentation techniques on the images to help figure out the symbol outline.
- The user gets the option of listening to the story rather than reading it.
- Matching Algorithm is used to detect Gardiner's code and get its corresponding meaning.

3.4 System Context

As shown in Figure 8, the user will be able to upload an image or live to scan the walls of temples. The system will be able to show the translation of these scanned images with full translation as sentences, not just characters. The user, also, can choose the translation to be in another language other than English. Also, the feature of hearing the correct pronunciation of the Hieroglyphs is supported.

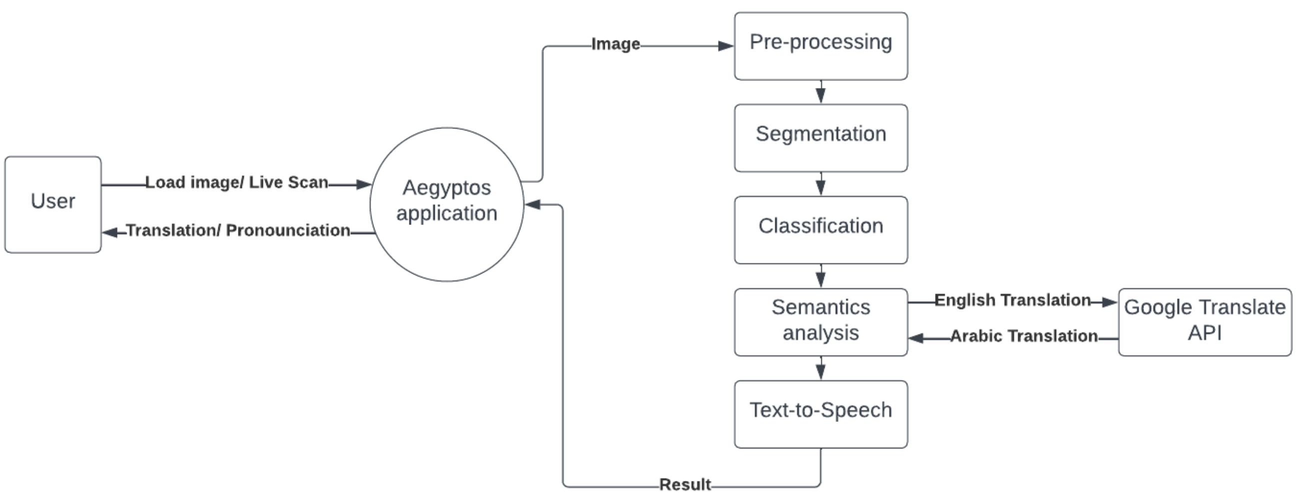

Figure 8: System Context

3.5 Objectives

- To translate the Hieroglyphs written on ancient walls with their semantics.
- Simple and fast mobile application that is accessible to everyone.
- To help users easily access the features in an offline mode if there is no internet connection.
- Instead of reading the translation, the user has the option to listen to it using text-to-speech.
- The user has the choice to read more about any king whose name occurred.
- The user can slide the phone's camera on the picture for real-time translation.

3.6 User Characteristics

- The user must have some basic knowledge of how to use a smartphone.
- The user must have some basic English knowledge as it will be the default language of the mobile application.
- The user can be of any age group.
- The user could be a resident or a tourist.
- The admin should have some basic knowledge of Administrator roles (CRUD operations).
- The developers should have a strong mobile development background to be able to fix bugs and keep the system updated.

4 Functional Requirements

4.1 System Functions

- **ID:01** The user shall be able to create an account.
- **ID:02** The user shall be able to Login and have access to his/her account.
- **ID:03** Each user shall edit or delete their account.
- **ID:04** The user shall be able to access the camera through the application to scan Hieroglyphics symbols for translation.
- **ID:05** The user shall be able to upload an image from his/her device for translation.
- **ID:06** The user shall be able to select the language to be translated into.
- **ID:07** The user shall be able to listen to the scanned Ancient Egyptian text.
- **ID:08** The admin shall be able to create, edit and delete a User account.
- **ID:09** The admin shall be able to manage the data of the system.
- **ID:10** The admin shall be able to view all users.

- **ID:11** The system shall segment hieroglyphic symbols from a scanned image.
- **ID:12** The system shall be able to classify each Hieroglyphic symbol.
- **ID:13** The system shall be able to detect the Hieroglyphic symbol's direction.
- **ID:14** The system shall be able to search data through a matching algorithm
- **ID:15** The system shall provide a precise translation in different languages.

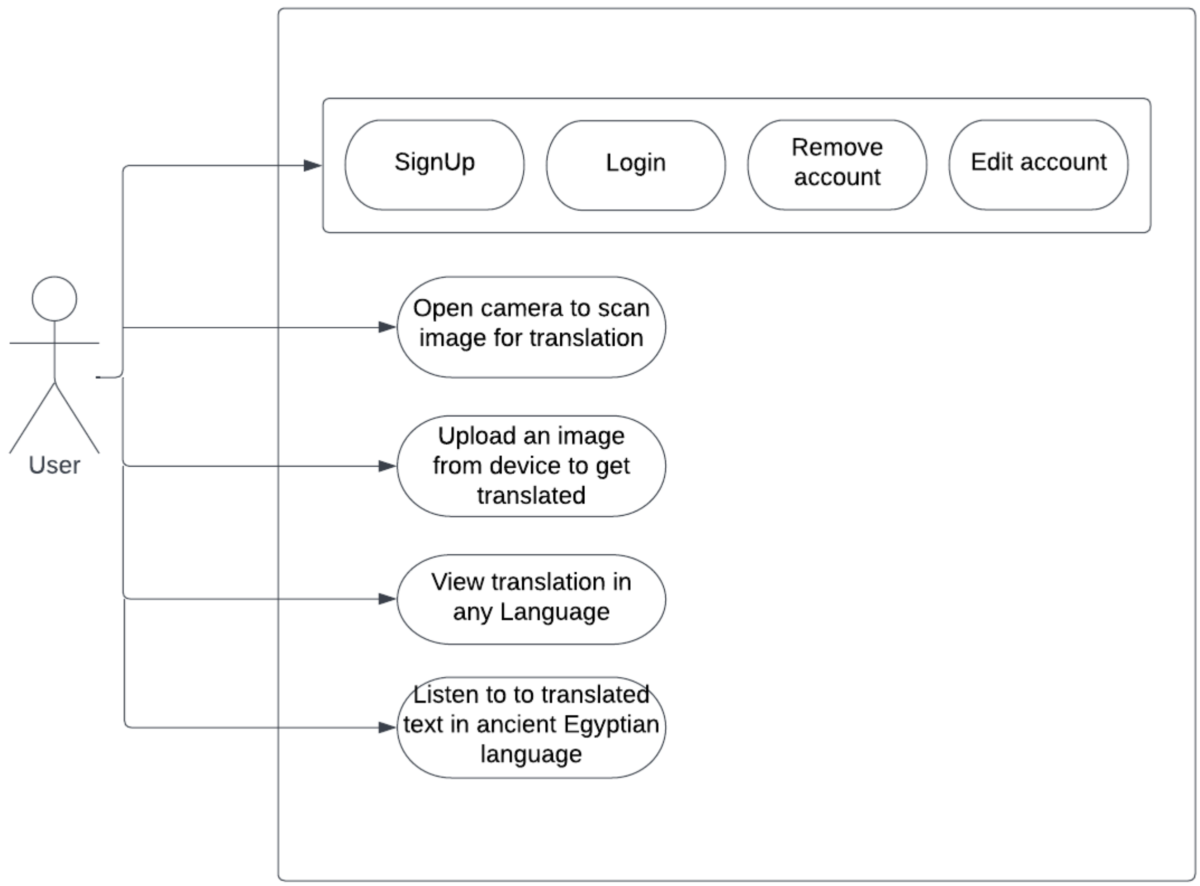

Figure 9: Use Case Diagram

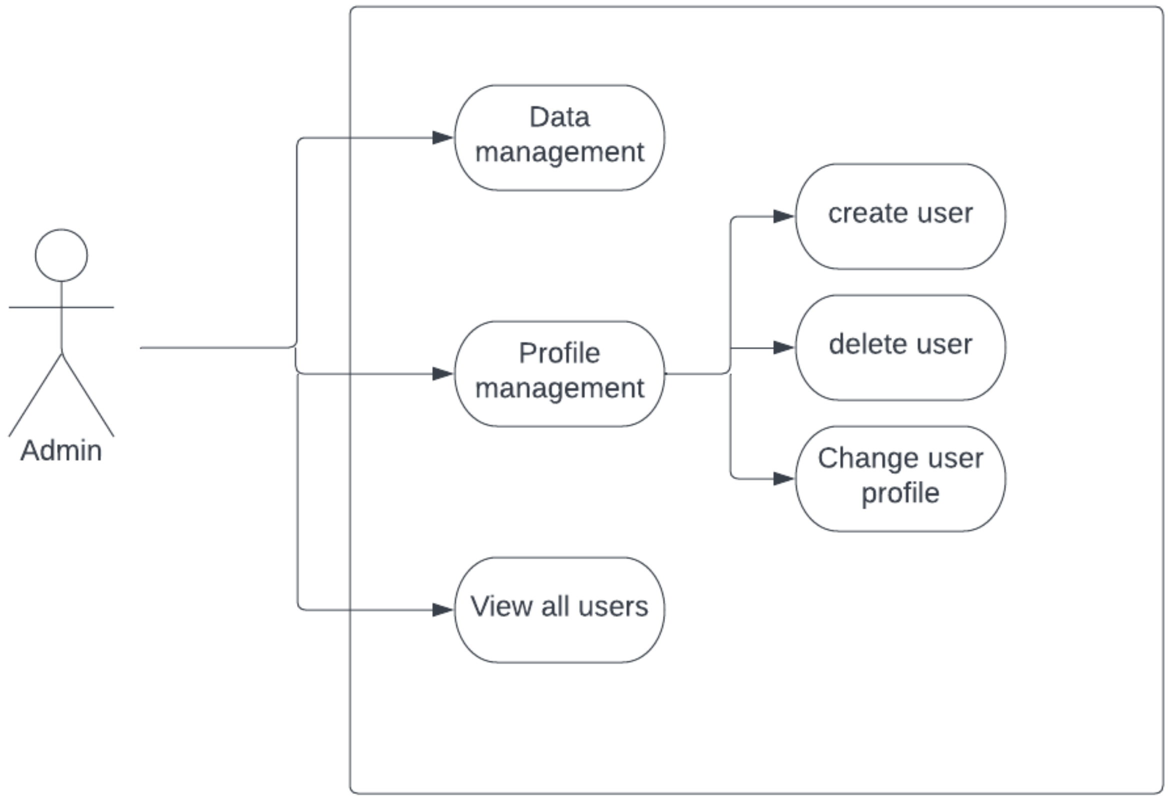

Figure 10: Admin Use Case Diagram

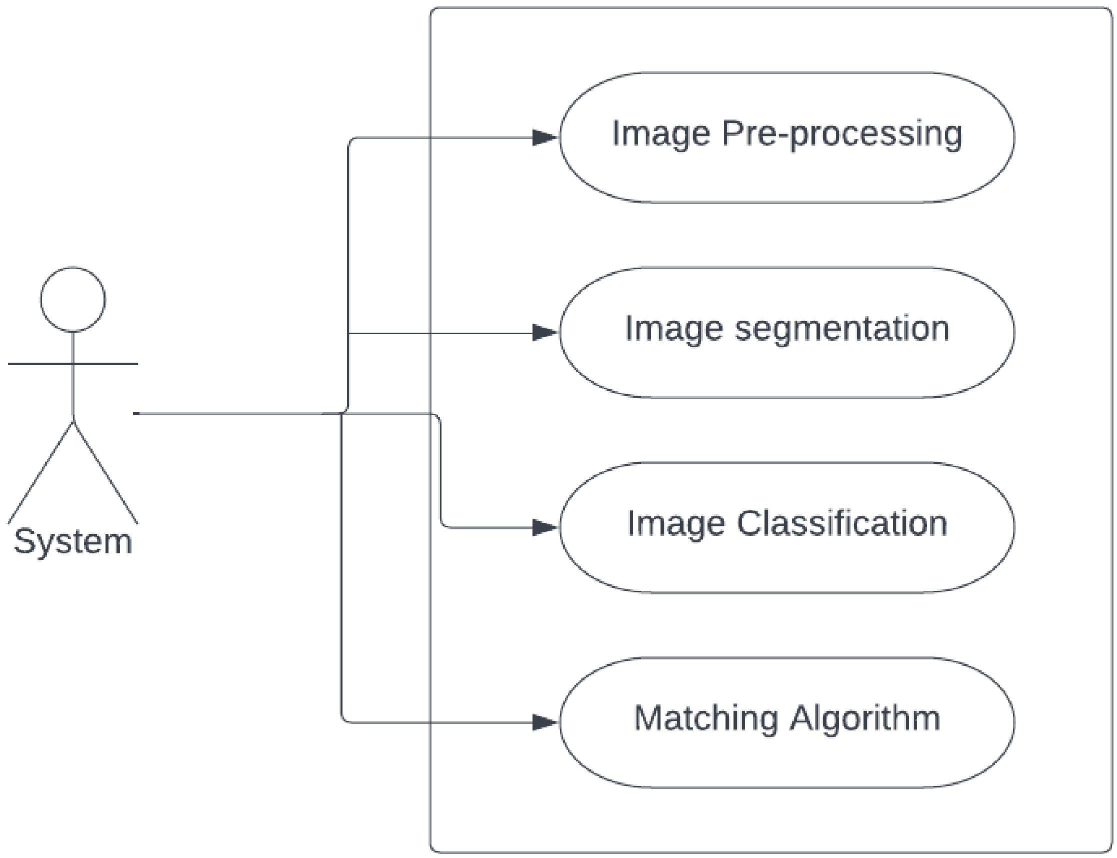

Figure 11: System Use Case Diagram

4.2 Detailed Functional Specification

Table 2: Scan Image

Name	Scan Image
Code	ID:04
Priority	High
Critical	to provide the application with data to classify and process.
Description	the function is responsible for extracting Hieroglyphics characters to be translated
Input	Image
Output	Translated text
Pre-Condition	User choose the language to be translated to.
Post-Condition	Image is processed
Dependency	Users will have to Allow access to the camera to be able to scan hieroglyphics symbols
Risk	Camera permission is denied

Table 3: Upload Image

Name	Upload Image
Code	ID:05
Priority	High
Critical	to provide the application with data to classify and process.
Description	The function is responsible for extracting Hieroglyphics characters from uploaded image to be translated
Input	Image
Output	Translated text
Pre-Condition	User choose the language to be translated to.
Post-Condition	Image is processed
Dependency	Users will have to accept permission to access device's photos
Risk	permission to access photos is denied

Table 4: Select Language

Name	Select Language
Code	ID:06
Priority	High
Critical	to provide the user with translations in a language he/she understands.
Description	The function is responsible for letting the user select the language in which he would like the translation to appear.
Input	User selection
Output	Text translated in the selected language
Pre-Condition	User choose the language to be translated to.
Post-Condition	...
Dependency	The translated text should appear in the language that was chosen by the user
Risk	The application does not support the selected language.

Table 5: Listen to text

Name	Listen to Hieroglyphics text
Code	ID:07
Priority	High
Critical	To provide the user pronunciation of Hieroglyphic text
Description	The function is responsible for providing pronunciation of the Hieroglyphic text
Input	Uploaded/scanned image
Output	Speech
Pre-Condition	The translation of the hieroglyphic text is accurate.
Post-Condition	...
Dependency	depends on the characters classification and the matching algorithm
Risk	None

Table 6: Preprocessing and segment the Hieroglyphs

Name	Preprocess
Code	ID:11
Priority	High
Critical	To provide the system with segmented symbols without background or noise
Description	The function is responsible for segmenting the symbols from the background
Input	Uploaded/scanned image
Output	segmented image
Pre-Condition	The uploaded or scanned image.
Post-Condition	Segmented image to be aken to classification
Dependency	Depends on the user uploading an image
Risk	None

Table 7: Classifier

Name	Classify
Code	ID:12
Priority	High
Critical	To classify the symbol into the right class for the translation
Description	The function is responsible for classifying the symbol into its right class
Input	Segmented image
Output	Gardiner's code of the correct class with percentage of confidence
Pre-Condition	Segmented image.
Post-Condition	The image is converted to its representative Gardiner's code
Dependency	Depends on function ID:11
Risk	None

Table 8: Postprocessing

Name	PostProcess
Code	ID:14
Priority	High
Critical	To translate the Gardiner’s code into the closest translation.
Description	The function is responsible for connecting the Gardiner’s code to the closest translation from the matching algorithm and dictionary
Input	Gardiner’s code of classified image
Output	Translation of the Gardiner’s code in English
Pre-Condition	Gardiner’s code from the classifier.
Post-Condition	The Gardiner’s code is converted to its meaning
Dependency	Depends on function ID:11
Risk	None

5 Design Constraints

5.1 Standards Compliance

Only Android or iOS are supported by the Aegyptos mobile application.

5.2 Hardware Limitations

The user needs a smartphone with a working camera to take pictures of the Hieroglyphics and a working speaker to translate the text into speech.

5.3 Other Constraints as appropriate

The user can access the application whether he has an internet connection or not.

6 Non-functional Requirements

There are a number of non-functional requirements for the Aegyptos mobile application, starting with the

6.1 Security:

Each user will need a unique email and password to get into the application, and that password will be a hashed password.

6.2 Performance:

Moving on to the performance, it won’t take long to translate the hieroglyphs and show them to the user because they will be translated instantly.

6.3 Reliability:

Additionally, applying the squeezeNet algorithm will be faster and more accurate than using any other technique, with a 92% accuracy rate.

6.4 Usability:

Anyone, starting with teenagers interested in hieroglyphs, can use the application to learn their language because it has a user-friendly layout.

7 Data Design

7.1 Database Description

- Users Collection - This collection contains information about the users of your application. Each document in this collection represents a single user and contains the following fields: name, email, password, user role, and id. The "id" field could be used as a unique identifier for each user.
- History Collection - This collection is used to keep track of the uploaded images and their translations made by the users. Each document in this collection represents a single instance of an uploaded image and its translation and contains fields such as image URL, English translation, and Arabic translation.
- Nefertiti Collection - This collection contains information about Nefertiti, an ancient Egyptian queen. Each document in this collection represents a random piece of information about her, such as her birth date, family background, or notable accomplishments.
- Rosetta Stone Collection - This collection contains information about the Rosetta Stone, an ancient Egyptian artifact that helped scholars decipher hieroglyphics. Each document in this collection represents a random piece of information about the Rosetta Stone, such as its discovery, its inscription, or its significance.
- Tutankhamen Collection - This collection contains information about Tutankhamen, an ancient Egyptian pharaoh. Each document in this collection represents a random piece of information about him, such as his reign, his tomb, or his treasures.

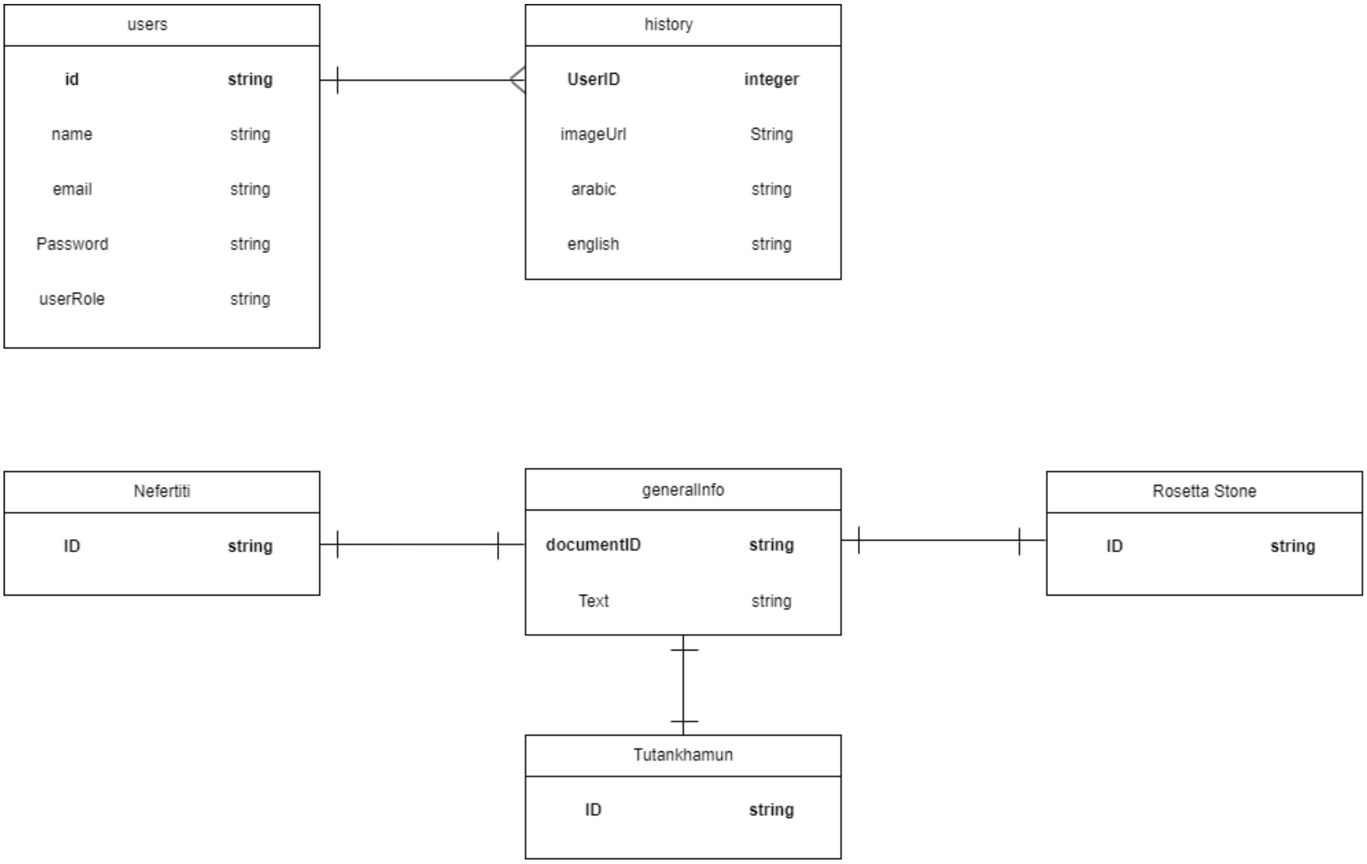

Figure 12: Database diagram

7.2 Data Description

- A new dataset was collected as shown in figure 13 using Gardiner's code that contains more than 60,000 images that are divided into 1072 classes, each image in this dataset has been saved with a special Gardiner code that uniquely identifies each hieroglyphic character. Additionally, the dataset has been run through a data augmentation algorithm to provide various visuals for each hieroglyphic character. All of the images have equal sizes of (472*354 px). And as shown in figure 14 this was the image after applying the Otsu thresholding segmentation across all the 1072 classes. This dataset is divided into 3 parts test, train, and val.

Number of images	Number of classes (according to Gardiner's code)
64,320	1,072

Table 9: Dataset using Gardiner's code list

Figure 13: E029 (before segmentation)

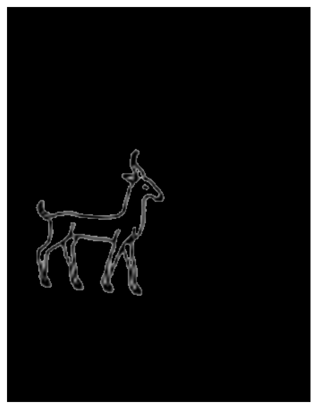

Figure 14: E029 (after segmentation)

8 Preliminary Object-Oriented Domain Analysis

9 Operational Scenarios

Scenario 1: The user can sign-up or log in to access the mobile application. The user then has the option of whether to capture an image or live to scan a video to get their translation as full sentences.

Scenario 2: The user has the ability to change the language as their preference, as the mobile application's default language is English.

Scenario 3: The user has the option to hear the correct pronunciation of the Ancient Egyptian Hieroglyphs, to live the full experience.

Scenario 4: The user can view their previously uploaded or scanned image translations and then store their preferred translations. The user always has the choice to delete their image history.

Scenario 5: The administrator must log in to the system when a user's information needs to be added, removed, or updated. The administrator will also be able to browse the user list and conduct a name-based search for a particular user.

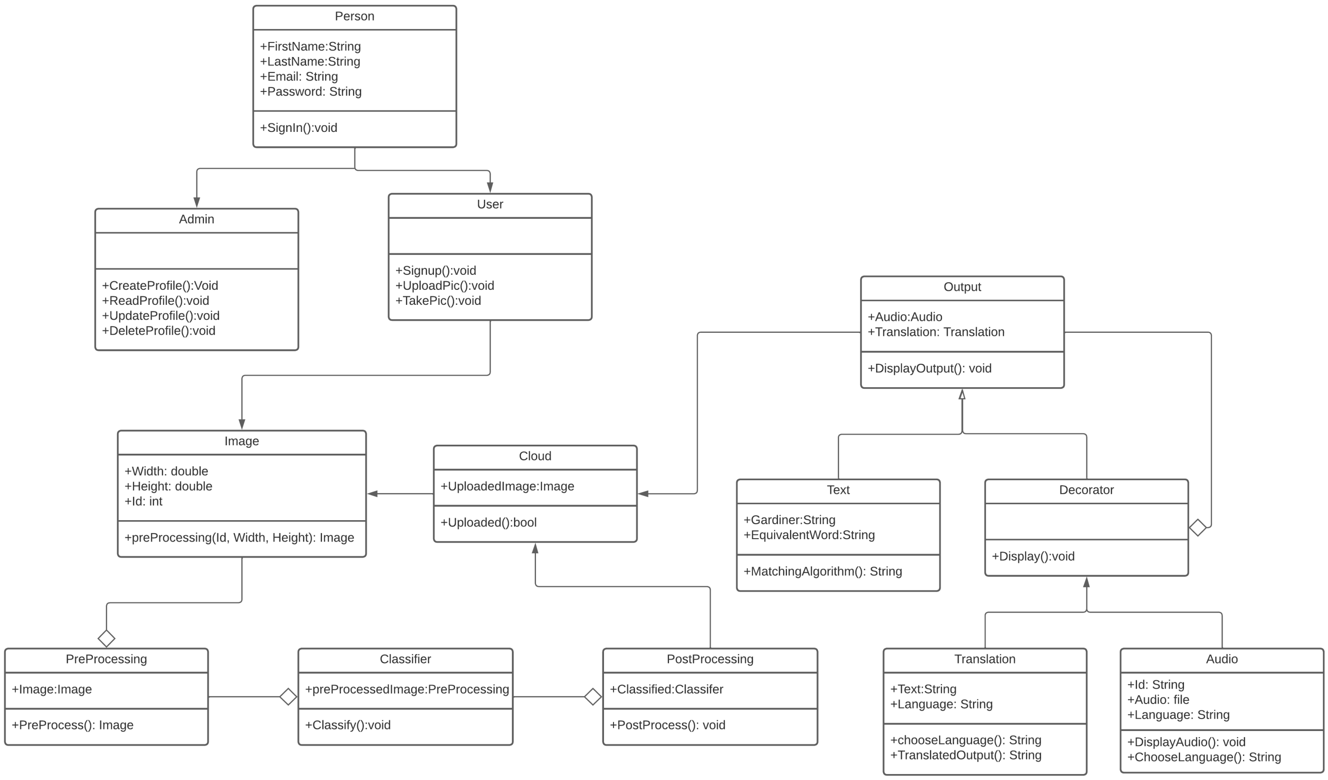

Figure 15: UML Class Diagram

10 Project Plan

Task	Start Date	End Date	Duration	Responsible member
Implement segmentation on our personal dataset folder	15-Nov-2022	22-Nov-2022	7 Days	Shehab
Design the UI and Front End of the mobile application	15-Nov-2022	22-Nov-2022	7 Days	Shehab
Look into the different classifiers	15-Nov-2022	22-Nov-2022	7 Days	Basma & Reem
Look into ways for implementing Decision Trees into code	15-Nov-2022	22-Nov-2022	7 Days	Ahmed
Implement segmentation on the real dataset folder	22-Nov-2022	29-Nov-2022	7 Days	Shehab
ResNet50	22-Nov-2022	29-Nov-2022	7 Days	Basma & Reem
Implement dot language for decision trees	22-Nov-2022	6-Dec-2022	14 Days	Ahmed
Implement morphology on the dataset segmentation	29-Nov-2022	6-Dec-2022	7 Days	Shehab
Test Google Translate API	29-Nov-2022	6-Dec-2022	7 Days	Shehab
SqueezeNet (Pretrained)	29-Nov-2022	6-Dec-2022	7 Days	Basma & Reem
EfficientNet(Untrained)	29-Nov-2022	6-Dec-2022	7 Days	Basma & Reem
Implement a way to crop symbols from a big image	6-Dec-2022	13-Oct-2022	7 Days	Shehab
Work on the semantic analysis	6-Dec-2022	13-Dec-2022	7 Days	Ahmed
Work on the matching algorithm	6-Dec-2022	13-Dec-2022	7 Days	Ahmed
SqueezeNet (Untrained)	6-Dec-2022	13-Dec-2022	7 Days	Basma & Reem
EfficientNet (Pretrained)	6-Dec-2022	13-Dec-2022	7 Days	Basma & Reem
Start working on the SRS document	6-Dec-2022	17-Dec-2022	11 Days	All Members

Table 10: Time Plan

11 Appendices

11.1 Definitions, Acronyms, Abbreviations

The following table explains the abbreviations used:

Term	Stands for
CNN	Convolutional Neural Network
API	Application Programming Interface
AR	Augmented Reality
CRUD	Create, Read, Update, Delete

11.2 Supportive Documents

- **Dell Competition:** We applied for the competition made by DELL technologies as shown in figure 14.
- **Egypt IOT & AI Challenge:** We applied for the "Egypt IOT & AI Challenge" competition as shown in figure 15.

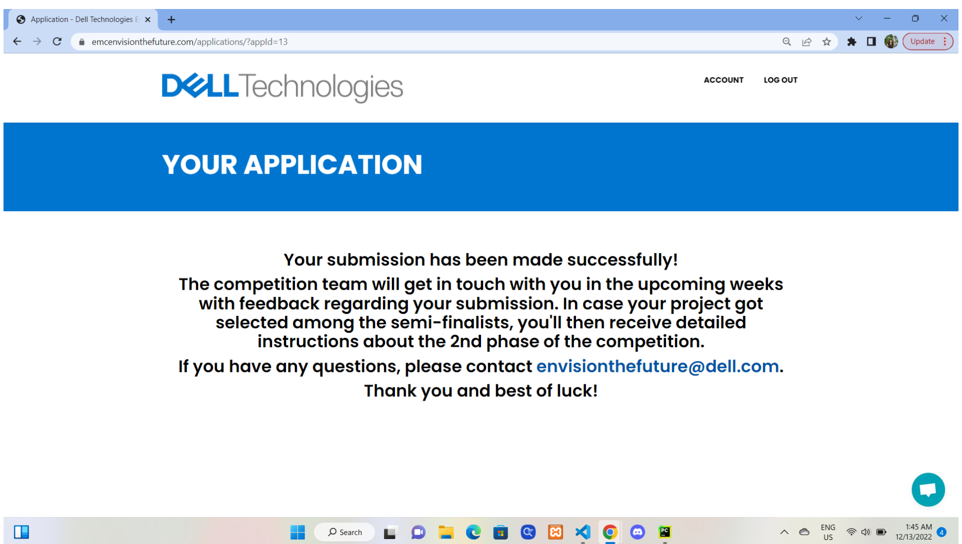

Figure 16: DELL competition acceptance

Figure 17: Egypt IOT & AI Challenge competition acceptance

References

[1] Xiangyang Xu et al. "Characteristic analysis of Otsu threshold and its applications". In: *Pattern recognition letters* 32.7 (2011), pp. 956–961.

[2] Jamileh Yousefi. "Image binarization using otsu thresholding algorithm". In: *Ontario, Canada: University of Guelph* (2011).

[3] Reham Elnabawy, Rimon Elias, and Mohammed Salem. "Image based hieroglyphic character recognition". In: *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE. 2018, pp. 32–39.

[4] David A Plecher et al. "ARsinoë-Learning Egyptian Hieroglyphs with Augmented Reality and Machine Learning". In: *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE. 2020, pp. 326–332.

[5] Andrea Barucci et al. “A Deep Learning Approach to Ancient Egyptian Hieroglyphs Classification”. In: *Ieee Access* 9 (2021), pp. 123438–123447.

[6] Reham Elnabawy et al. “Extending Gardiner’s code for Hieroglyphic recognition and English mapping”. In: *Multimedia Tools and Applications* 80.3 (2021), pp. 3391–3408.

[7] *Dua Khetty application on app store*. URL: https://apps.apple.com/us/app/dua-khety/id1466697097.

[8] *MS Windows NT Kernel Description*. https://www.macrotrends.net/countries/EGY/egypt/tourism-statistics. Accessed: 2010-09-30.

[9] *Scriba application on app store*. URL: https://play.google.com/store/apps/details?id=com.scriba.app.

[10] *Unravel the Symbols of Ancient Egypt.* URL: https://blog.google/outreach-initiatives/arts-culture/.