IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 March 2023, accepted 10 April 2023, date of publication 17 April 2023, date of current version 24 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3267981

RESEARCH ARTICLE

An AI Based Automatic Translator for Ancient Hieroglyphic Language–From Scanned Images to English Text

ASMAA SOBHY[1], MAHMOUD HELMY[1], MICHAEL KHALIL[1], SARAH ELMASRY[1], YOUTHAM BOULES[1], AND NERMIN NEGIED[2,3]

[1]Faculty of Electrical and Computer Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada
[2]School of Communication and Information Engineering, Zewail City of Science and Technology, Giza 12578, Egypt
[3]School of Engineering and Applied Science, Electronics and Computer Engineering Department, Nile University, Giza 12677, Egypt

Corresponding author: Nermin Negied (nnegied@debi.gov.eg)

ABSTRACT Recent advancements in the fields of Machine Learning and Deep Learning made a huge transformation in other fields that are not related to Computer Science. In this work, a new framework is proposed to tackle the problem of translating the old Egyptian Hieroglyphic writings to English language through deploying both Image Processing and Natural Language Processing techniques combined with AI approaches. Our primary goal is to design an application that completely revolutionizes a tourist's experience while navigating Egyptian Historical sites. This work utilize different AI techniques to automatically convert the scanned photos of hieroglyphic language to understandable and readable English language, through two main sub-tasks: The automatic detection and recognizing of the scanned glyphs images and the translation of them into English language. Different data sources of this low-resource language were explored and augmented to train and test our models. Results of different models and algorithms are assessed and analyzed to evaluate our work. State-of-the-art results are achieved compared to literature in both automatic glyphs recognition, and glyphs-to-English translation.

INDEX TERMS Machine translation, glyph, image processing, classification, object detection, natural language processing.

I. INTRODUCTION

Old Egyptians used Hieroglyphic language to record their findings in medicine [1], engineering, sciences, achievements, their religious views, beside facts from their daily life. Thus, it is fundamentally important to understand and digitally store these scripts for anyone who wants to understand the Egyptian history and learn more about this great civilization. In this work, the aim is to decipher this remarkably interesting language to make it easier for tourists to understand the ancient Egyptians' scripts, through the automatic detection and recognition of hieroglyphs then and translating them into English. That way people will be able to read what ancient Egyptians wrote in the Paranoiac era with-out referring to Egyptologists who are very rare and expensive.

In ancient Egypt, hieroglyphs were the official writing system. Nearly 1,000 symbols were used in the language. Until Jean Champollion deciphered the Rosetta Stone, hieroglyphic knowledge was lost. The main characteristics of this interesting language are stated below.

A. CHARACTERISTICS OF THE LANGUAGE

1) Because each symbol has its distinct sound, there is no link between knowing a hieroglyph and understanding how to read it. The glyph of the lower leg, for example, means nothing about the leg, but it sounds like the letter ''b'' in the English alphabet. These sound representations are the *''transliteration''* of the Ancient Egyptian language as they reflect how people spoke it. Transliteration is mapping a phrase's phonetics to the desired alphabet (e.g., English) based on phonetic similarity with no regard to the meaning of the sentence.

The associate editor coordinating the review of this manuscript and approving it for publication was M. Venkateshkumar.

2) Hieroglyphs can appear in various directions - Horizontally (from left to right or right to left), and vertically.
3) The presence of determinations plays a crucial role in the language.

There are symbols that do not have a sound but provide meaning to the words, such as the type of word or whether it is a singular or plural word. This language characteristics cause ambiguity in reading the language and understanding it. In other words it needs a lot of study in the linguistic knowledge to be able to know the linguistic challenges before automating the task. We learned a lot about the language to be able to understand it before trying to give the machine the ability to understand it, and we also investigated the available ways to simplify understanding this amazing ambiguous language. We found the Gardiner code's list which was first introduced and compiled by Sir Alan Gardiner to simplify the language understanding process. The Gardiner code's list contains all the common Egyptian glyphs and their subcategories of Egyptian glyphs. Reference [2] The signs are organized into 26 main categories followed by 3 sections that list hieroglyphs by their shape.

Collecting an appropriate dataset for this work was not an easy task, as it is not common to find a complete dataset for Hieroglyphic language but fortunately, we succeeded to gather the scripts written on Unas pyramid [3],this pyramid is very significant for having the first example of funerary texts known as Pyramid Texts. The gathered data set contains 172 different symbols. The remaining of the paper is organized as follows; Section II demonstrates the work done in literature related to our work. Section III describes the datasets used in this work and the data processing steps done to prepare the data for processing and testing. Section IV explains the proposed methodologies for glyph classification and translation in details. Section V discusses the results obtained by the work proposed in this work, and finally the paper is concluded in section VI which includes suggestions for future work also.

II. LITERATURE REVIEW

From surveying literature it was easy to find some past work investigated the visual descriptors and how to segment and recognize the characters, while other researchers tried to translate the language and understand the hieroglyphic text. However, no previous research have been done to tackle the end-to-end process, starting with recognizing the visual descriptors until translating it to English.

For instance, Gemert et al. [3] implemented a solution that automatically recognize Hieroglyphic text from an image. For glyph localization, the authors used a saliency-based text-detection algorithm [4]. Then they used an appearance matching approach with an advanced version of the Histogram of Oriented Gradients method (HOG) which is HOOSC [5]. Finally, they performed a pairwise matching with a labeled patch. Their detection approach detected only 83% of the glyphs and glyph matching was only 74% successful.

Barucii et al. [6], used ResNet-50 to develop a classification method to classify the glyphs. They didn't find the sufficient dataset to train their approach, so they used a different dataset then they used transfer learning. They also implemented a novel architecture called Glyphnet and trained it on a small hieroglyphic dataset which is designed for the specific task of hieroglyph classification and trained the network on it. The result showed that Glyphnet achieved an accuracy rate of 96% which is the highest accuracy found literature. But the data used in their work is unfortunately not available for other researchers to validate the results.

Hossam et al. [7] used simple image processing techniques to detect and segment the glyph like canny edge detector and Region of Interest segmentation (ROI), then they tried several image matching algorithms to match the segmented glyph with the glyphs in the dataset. The authors then confirmed that the Histogram of Oriented Gradients (HOG) obtained the best matching results. The authors at the end of the paper touched the translation of glyphs into English, but they confessed that they didn't obtain any noticeable results because of the fact that their main focus was on the image processing part and they did not consider anything regarding the linguistic language, leaving this task to researchers interested in Natural Language Processing. The authors demonstrated their glyph recognition results for every hieroglyphic Gardiner's code individually with the highest one reaching 87% and the lowest reaching 33%. Using the average of their glyphs recognition results, we can affirm that they achieved an overall classification accuracy of 66.7%.

Regarding the translation task, Neural Machine Translation (NMT) was used to translate the Sumerian language [8]. The Sumerian language is another old language. The lack of both language resources and the full understanding of it results in obtaining distorted English phrases from the translation process. That issue makes semi-supervised approaches have difficulties, since phrases from the large available corpora of English phrases out there won't be similar to the English phrases that exist in the parallel corpora that have been translated word by word. This incoherence could confuse the machine translation model. Other research [8] referred to languages that suffer from target-side incoherence as ''extremely'' low-resource languages.

Some other researchers tried to translate the Ancient Egyptian language. They used both transliteration and Gardiner signs. It is worth mentioning that there is an open-source repository of Fayrose in GITHUB under EgyptianTranslation project, which contains Egyptian transliterated hieroglyphs corpus.

III. DATASETS AND DATA PREPARATIONS

This section introduces the different datasets used in this work to train and test our different system's modules like object detection, image classification, word segmentation and machine translation.

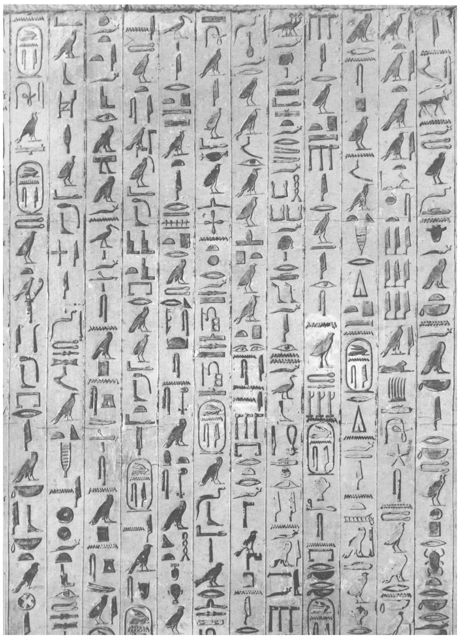

FIGURE 1. Example piece of Unas pyramid wall.

Gardiner Code					
Image	D1	f13	M4	M29	E1

FIGURE 2. Sample of the dataset.

A. GLYPHS DATASET

The dataset used in this work is called Morris Franken dataset and it is one of the few publicly available datasets. It covers 4210 glyphs, representing Egyptian hieroglyphs found on the walls inside the Pyramid of Unas, which is characterized by its vertical columns of hieroglyphic writings. Figure 1 shows an example piece of Unas pyramid. The resolutions of the images in this dataset are approximately 1150 × 1600 pixels in width and height, respectively. The images are manually annotated providing the bounding box for each glyph.

Individual glyphs in this dataset are labeled according to their Gardiner's codes, and each has an image of dimensions of 75 × 50 pixels. Figure 2 shows some sample images from the dataset. The images cover 172 different Gardiner's codes. The distribution of the images among the 172 labels is unbalanced (see figure 4) where most of the labels are having less then 10 images, meanwhile some labels have a larger number of images. A dataset augmentation was held in this work in two different ways for the tasks of detection and classification separately. In the following subsections, a demonstration of how the dataset was prepared for each of the two tasks respectively.

1) GLYPH DETECTION

Random cropping was used to create an augmented dataset using Albumentations [9]. The resulting dataset contained 2000 images. We defined the aspect ratios of the cropped patches as the most common aspect ratios for smartphone cameras in both the portrait and landscape orientations, which are 4:3 and 16:9 respectively. The distribution of the aspect ratios is equal, and they span 3 different resolutions each. The 4:3 aspect ratio's resolutions are (512, 384), (640, 480), (800, 600), while the 16:9 aspect ratio has the (426, 240), (640, 360), (854, 480) resolutions. The resolution was chosen to maintain a useful number of glyphs in the crop. Figure 3 shows an example of the generated crops in addition to the bounding boxes inside it.

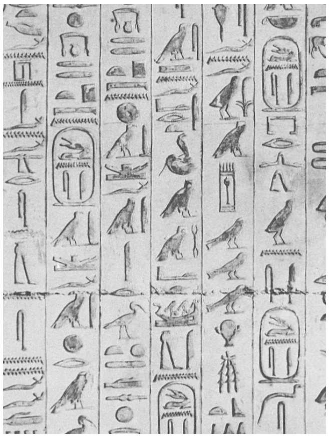

FIGURE 3. Example of a generated crop.

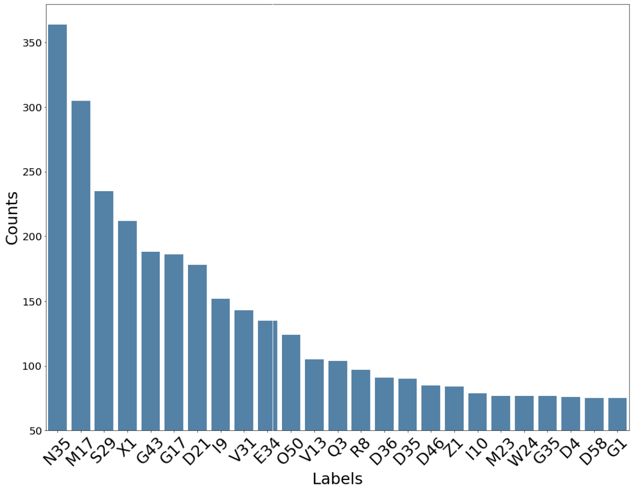

FIGURE 4. Distribution of the dataset.

2) GLYPH CLASSIFICATION

For the Classification task, the 4210 image crops were split into training and testing data with ratios 70:30 respectively in a stratified manner. Given small size data, a data augmentation were applied to to the training data.

Data Augmentation was done using the convolutional neural networks (CNN) to increase the size of the training data as it is very efficient in increasing size of data and avoid over-fitting problem at the same time.

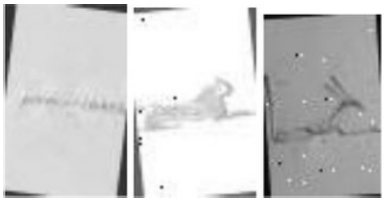

FIGURE 5. Augmented data-set samples.

In addition to that a random zoom with minimum zoom of 0% and maximum zoom of 15% was also used to enlarge the dataset. As well as rotation between −11° and +11°, different brightness levels of the image with an average starting from −21% to 21% were also used for the same purpose, and some noise pixels (3%) were added to some images too.

At the end, we succeeded to increase the size of the training dataset from 2947 images to 8226 images. Figure 5 shows some examples of the augmented images.

B. TEXTUAL CORPUS

The textual corpus is a hybrid dataset constructed using two sources: Fayrose/Lauren Fay on Github[1], and the dataset obtained from Hugging Face datasets [8]. In Fayrose's dataset, each sample is an ancient Egyptian phrase in transliteration format, along with the corresponding English translation. The Hugging Face dataset is the same as Fayrose's, but the translation is in German for most samples. Some samples have English translation instead of German, and some others have translations that are a mix of German and English. The data-set has an extra field for the Gardiner codes associated with the sample, but unfortunately, this field is empty for most samples.

Some challenges here include that the transliteration formats are different between the two datasets. Also, the Hugging Face dataset has some interpretations and transliterations that are damaged, missing, or unclear, for example, they might surround a part of the transliteration with question marks to represent that this part was missing from the source.

To unify the data, some data pre-processing steps were applied to the Hugging face dataset. First, a translation from German to English was done, then any brackets were removed along with the transliteration inside. Researchers in literature [10] confirmed that it is better to keep as much information as possible, but through this research, it was found that manipulating the data in a different way could be very useful, so a mapping of some characters from the Hugging face dataset to match the other format was done.

For the machine translation part, different data sources were used, following are the datasets used in this stage and how they have been augmented:

1) Hugging Face dataset [10]
2) Fayrose's dataset [11]
3) The entire corpus: combination of (1) and (2)
4) BERT augmented dataset
5) Synonyms augmented dataset
6) Back-translation augmented dataset
7) All-augmented dataset

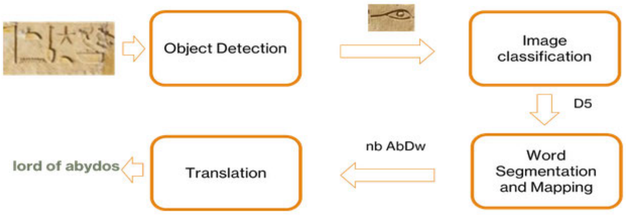

FIGURE 6. The overall diagram of the project's workflow.

Nlpaug library was used to augment data. BERT augmented dataset was constructed by inserting extra words that do not affect the meaning of the phrase. Synonyms augmented dataset was constructed using WordNet to replace some words with their synonyms. Back-translation augmented dataset was constructed by translating the target phrase to another language, then translating it back to English so that the wording change but the meaning does not. The All-augmented dataset was constructed by merging the three augmentation datasets. All the augmentations were done on the Fayrose dataset to increase its size. The augmentation is not performed on the full dataset but only to a subset of the dataset that was chosen randomly.

C. DICTIONARY

The dictionary created by Ancient Egypt and Archaeology Web Site [12] were used together in this work, the dataset is a CSV file that contains multiple columns of Hieroglyph words and its Gardiner's translation and transliteration. So, in the preprocessing phase, we removed the English, Hieroglyph, and duplicates, so, we had a dictionary of 10596 words.

IV. METHODOLOGY

The approach proposed in this work was splitted into two major tasks as mentioned before. The two tasks are hieroglyphic character recognition, and Hieroglyphic to English language translation. As shown in Fig.6, the input is a picture containing a group of hieroglyphs, which passes through the object detection block. The role of the Object detection part is to detect the bounding box of each glyph and crop it converting the picture into a set of ordered crops to be passed each to the classification block. The classification block then takes the images of single glyphs and classify them into their corresponding Gardiner's codes. The task of character recognition was further divided into two steps because the dataset used containing 4210 crops was insufficient to solve the problem in one step. The Second task which is Hieroglyphic - English language translation was also divided into two steps. The set of ordered classified Gardiner's codes enters the Segmentation and Mapping block to segment the codes into words because the Hieroglyphic language's words do not have spaces between them. Finally the translation block then translates the words to English.

Different open-source libraries for Machine learning and deep learning were used here like Tensorflow2.0 which used for creating architecture for convolution networks

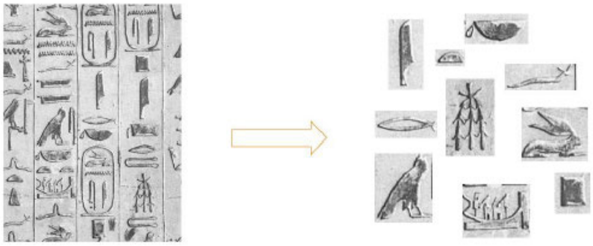

FIGURE 7. Example Input/Output for the glyph detection part.

relying of its powerful modules, also openNMT was used for machine translation part and META's open source library,Detectron,was mainly used for computer vision part.

A. GLYPH DETECTION

The goal of this part is to detect glyphs in an image, so that they could be first classified then translated to English text. Given an image as an input, the output should depict the glyphs present in the bounding box coordinates, as shown in figure 7.

For the object detection part, R-CNN [13] algorithm was employed. The algorithm adopts a four-way multi-task learning process: finding region proposals, predicting an objectness score (the membership to a set of object classes versus the background class), estimating class probabilities and finally correcting the proposed bounding box coordinates. What made us choose this algorithm over others is that it works better for smaller objects. Transfer learning from the ImageNet pre-trained weights was used, while freezing the first 2 stages of its 5-stage ResNet.

B. CLASSIFICATION

In the following part, the architectures used in this project is discussed in details. Siamese network and RestNet50 were used for image classification tasks.

1) RESNET50

ResNet was firstly developed by Microsoft Research labs in 2015 [14]. It is also proposed as 34, 50, and 101 layers network however we chose the 50 layers deep network which is pre-trained on 23 million parameters and has input to process images with dimension 224 × 224 pixels. The network represents residual learning, which solves the vanishing gradient problem in the deep neural network by allowing the other shortcut path for the gradient to go through. The networks can be trained from scratch or by using transfer learning [15]. In this work, new layers were added like: Input layer 70 × 75 to fit the images, dense layer with 256 neurons, and a SoftMax activation layer as output for classification (number of labels). Afterwards the model was compiled Stochastic gradient descent (SGD) with 0.9 momentum and learning rate of 0.001, batch size of 64 and 400 iterations and the loss function used is categorical cross-entropy.

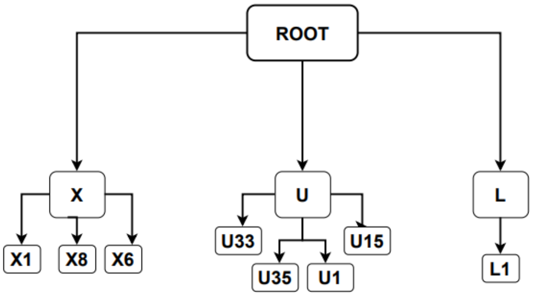

FIGURE 8. Part of the hierarchical classification.

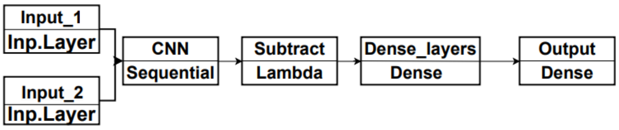

FIGURE 9. The Siamese network's architecture.

2) HIERARCHICAL RESNET50

Since 172 labels are now available, which is a large number compared to the aforementioned small number of images in the dataset. A new approach that uses hierarchical classification using ResNet50 is proposed in this work. Basically, the hierarchical classification in the figure below divided the classification into two main steps. The first step started from the root, so the model predicts the Gardiner letter which represents the image. Second step then is representing each Gardiner letter using a small model that predicts especially the Gardiner group, that way a group of models instead of one model as shown in Fig.8 is available. In hierarchical approach; the previous architecture with the same three new layers was used. The essential change made here is allowing varying size of outputs per class, i.e. Class X has a SoftMax activation of 3 outputs and Class G has SoftMax activation of 4 outputs, etc.

3) SIAMESE NETWORK

As mentioned before, the dataset is small and unbalanced. For that reason, the traditional methods for classification did not perform well, especially for the less represented classes. As a solution, the Siamese network [16] was deployed here. The "Siamese twin" term means an identical twin, the network was given this name because it takes two inputs, feeds them into two identical CNNs with the same weights, extracting 2 output tensors which will then be subtracted, getting the difference tensor. This difference tensor was then fed into 2 dense layers with 512 neurons and 256 neurons and a last layer with 1 neuron and a sigmoid activation function. The network outputs 1 if the 2 inputs are similar and zero otherwise. The Siamese network is usually used in signature recognition and face recognition tasks, and it is also known as one shot learning. The architecture of the network is shown in Figure 9.

The model was compiled with Adam optimizer with a learning rate of 0.001 binary cross entropy loss. The network

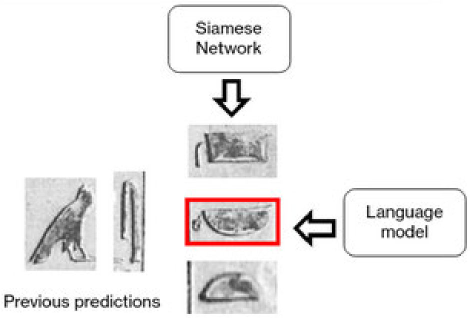

FIGURE 10. Prediction after adding the language model.

was trained with the dataset without augmentation for 20,000 iterations with batch size of 128. The model was then fine-tuned with the classes that have less than 10 images in the training data for 1,500 iterations with a reduced learning rate of 1e-6 and batch size of 32. The fine-tuning step increased the model's generalization to the less represented classes as this less represented set contained classes that had higher similarity thus increasing the difficulty of the training resulting in a better performing model.

A character level language model was then added to the Siamese network to rank the highest 3 predictions of the model as shown in Fig.10. A tri-gram language model was used to split the corpus of sentences into a new corpus of 3 characters in each entry. It takes two previous characters to calculate the probability of the current proposed character with equation 1. Where G1, G2 and G3 represent the Gardiner's codes in a proposed sentence in the correct order. C represent the count of the sequence in our corpus. This character level language model was added to take the previous predictions into consideration. The highest 3 scores of the Siamese network were then normalized with standard normalization, then the frequency scores of each prediction is calculated with the language model given the previous two predictions. The frequency scores of the language model was then normalized using the Softmax function 2 to avoid dividing by zero if none of the predictions were frequent. The Siamese network scores and the frequency scores are then summed to get the predicted Gardiner's code.

$$P(G3|G1, G2) = \frac{C(G1, G2, G3)}{C(G1, G2)} \tag{1}$$

$$p_i = \frac{exp(F_i)}{\sum_j exp(F_j)} \tag{2}$$

C. SEGMENTATION AND MAPPING USING DICTIONARY

The input to this module is the hieroglyphic characters as Gardiner's codes obtained from the previous module. The sentence then is segmented into words, so the output words are mapped into transliteration language to start the translation phase in English. The goal here is to try different segmentation techniques one based on dictionary and the other using subword tokenizer techniques. The work here is divided into 2 categories: First, the word segmentation using the Rule-based algorithms [17], where the segmentation is done based on a dictionary, and the other used sub-word tokenization or Sentencepiece. Beneath is a further explanation of both methods and a comparison between them.

1) DICTIONARY-BASED ALGORITHM

The rule-based algorithms are used to segment words based on a dictionary. The following subsections discuss two different Rule-based algorithms.

1) Forward maximum matching [18]: This algorithm works simply by taking longest m characters of the sequence that matches a word in the dictionary. A search for the word in the dictionary is done; if found, then it will be removed from the sequence, if not, then the last character of the sequence will be removed and a new sequence will be created. The algorithm iterates to segment all characters or words accordingly.
2) Reverse maximum matching [19]: The reverse matching algorithm works like the forward, but only removes a character from the beginning of the sequence, if the search cannot find a matched sequence.

2) SUB-WORD TOKENIZER

Sentencepiece is a language-independent sub-word tokenizer-detokenizer designed for neural-based text processing, including neural machine translation. Sentencepiece can train sub-word models directly from raw sentences.

In the Sentencepiece part, the model was trained on sentences of a series of Gardiner's codes like: (M11N5A13…). The segmentation was not excepted because it segmented letters and numbers separately, for example: 'M' and '11' were considered as two separate tokens, although the real Gardiner's codes should look like '11', '5', etc., another reason is the small number of sentences that were not enough for the model, as the training of this model needs rich dataset where the model can easily find combination between sequences of characters. Nothing in literature about this problems were found, and no attempts have been done to solve it according to the best of our knowledge, so a preprocessing step was implemented to substitute each number in the sequence with a corresponding combination of letters from a created dictionary. The dictionary is as (1: ab, 2: ac,…), so the sequence will be 'MagNae.'' As a result, the segmentation improved a lot, and the model overcame the problem of letters and numbers combinations.

D. MACHINE TRANSLATION

To be able to translate the transliterated text resulted from the segmentation and mapping stages, the transformer architecture was used here. It was trained using the different dataset settings of the textual corpus mentioned before. The used hyper-parameters are the same ones used in [10]. However, there still a lot of work to be made in the machine translation phase to get a complete model that translates hieroglyphic to English text, but a huge data is needed to achieve good results.

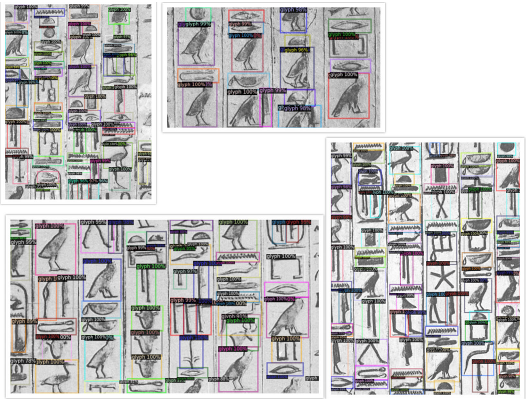

FIGURE 11. Example glyph detection output.

V. RESULTS

In this section, A discussion of the results obtained by each module separately is presented, each module as previously discussed solves an independent specific problem with its specific dataset. We then go over the evaluation of the entire workflow integrated together.

To assess the effectiveness of the glyph detection module, the aforementioned dataset was splitted randomly into training and validation sets in an 80:20 split ratio. The training process was run for 1000 iterations with a base learning rate of 0.01 which gets reduced by a factor of 0.1 at the 400th, 600th, and 800th iteration's mark as well as using a 100-step learning rate warmup. Training took about an hour to complete on an NVIDIA Tesla V100 GPU. Experiments yielded a final mean Average Precision (mAP) of 95.9% and an Average Recall (AR) of 74.4% at Intersection over Union (IoU) of 0.5. Figure 11 shows some output images after feeding them into the obtained model.

Using the Siamese model, a pairwise comparison with a labeled array is done. The anchor arrays were created to contain one anchor image and its corresponding label per every instance class and its corresponding Gardiner's code. For evaluation, the test image was compared with each entry in the anchor_img array to calculate the similarity score with the Siamese network then output the Gardiner's code corresponding to the entry with the highest similarity score. We then calculated the accuracy for the number of test images that were correctly classified over the total number of images tested.

The accuracy of the model after fine-tuning with the less represented classes is 85%.Since the evaluation method is so sensitive to our choice of the images in the anchor array, a multi layer anchor array with 3 images per class was created. The test image with the 3 images in each entry and we summed up the model scores to report the average highest score in matching the proper corresponding Gardiner's code. This method increased the accuracy to 87.5%. Finally adding the language model achieved the highest accuracy of 88.5%. For the classification stage different approaches were conducted and compared, ResNet50 which had some limitation because of the data size, so we shifted the Siamese network. Many approaches to increase the accuracy of the models were deployed and a comparison of the results of each approach is demonstrated in table 1. The Siamese network is a better fit for the problem of low resource language classification. However, for more number of classes and more data for training the ResNet50 might be a better fit because comparing the test image with more classes would be very time-consuming.

TABLE 1. Comparison of the methods used for classification.

Model	Accuracy
ResNet50 before data augmentation	61%
ResNet50 after data augmentation	72%
Hierarchical Classification	68%
Siamese Network	85%
Siamese with Multi-anchor array	87.5%
Siamese after adding a Language model	88.5%

TABLE 2. Comparison of results for word segmentation techniques.

Algorithm	Hamming distance	sequence Matcher
Forward Matching	6.6	0.58
Inverse Matching	6.7	0.57
Sentencepiece	3	0.21

To ensure that the dictionary's output is similar to the original dataset, hamming-distance and sequence-matching algorithms were used to evaluate the similarity between the outputted and original sentences.

1) The Hamming Distance [20]:
This algorithm compares two sentences, by comparing the number of characters positions in which the two sentences are different using a simple XOR.

2) SEQUENCE MATCHER [21]:
Sequencematcher is an algorithm that counts the number of matching characters between two strings and then output a ratio with a range between 0 and 1, where 1 means that the two strings are perfectly similar and 0 means the opposite.

$$Ratio = \frac{T}{M} \tag{3}$$

where M is the matched sequence and T is the total number of elements in both sequences.

Following the removal of all sentences that did not contain Gardiner's codes from the dataset, we had a total of 2538 sentences, and after removing duplicates and NAN values, we had 2350 sentences. The following table demonstrates the results of word segmentation techniques.

As shown in table 2, the champion algorithm is the Forward max matching as it achieved 60% correct segmentation ratio of the original sentence correctly mapped. The issue that Gardiner's codes sentences have unknown characters, so removing the symbols will require an Egyptologist to interpret the missing characters. The main challenge was the dependency on the dictionary in order to make good segmentation, so it is highly needed to have either a rich dictionary to cover most of the vocabulary or a dataset with

TABLE 3. Evaluation of translation results using different BLEU score techniques.

Setting	NLTK's BLEU	NLTK's GLEU	NLTK's corpus BLEU	sacreBLEU-BLEU
1	**49.73**	47.83	**59.19**	28.12
2	31.56	30.73	34.59	42.73
3	47.36	44.94	58.73	42.73
4	48.71	47.44	58.69	**70.71**
5	49.52	**48.34**	58.39	48.55
6	47.83	46.85	57.59	50.00
7	46.71	46.81	56.32	35.36

high number of well processed sentences and do not have any missing characters, which is impossible in such a low-resource language.

To evaluate the machine translation part, BLEU score was calculated in 4 different ways: NLTK's sentence BLEU, NLTK's GLEU, NLTK's corpus BLEU, and sacreBLEU-BLEU. For NLTK's BLEU, a smoothing function was used while computing the BLEU score. The following table demonstrates the results of translation.

The highest score was highlighted for each BLEU score method to be able to see which setting performed best across the obtained BLEU scores. The translation model proposed in this work achieved the highest BLEU score compared to previous work in literature. Compared to *Fayrose's work*, the NLTK's corpus BLEU reaches 59.19 in setting 1, which is more than the current max of *Fayrose's* NLTK's corpus BLEU of 42.22. Following are some analysis of the results to be used in the future to enhance the performance of this task.

The first setting of data scored better than the others in two methods, which suggests that a single data source with no augmentation might be best when it comes to low resource languages. Also the 5th setting results are close to the first setting with the highest GLEU score, which tells that augmentation using synonyms might be a good option to explore more. 6th setting proved to be the worst one. This might be because of the TSIC problem, since back translation translates the sentence to another language and then back to English, but the sentences in the training corpus do not map to the modern day English, so these translations might be misleading and confusing to the machine. Analysis has showed that Setting no. 6 is the reason of having a poor performance from Setting no. 7 where it uses all the augmentation techniques. Setting no. 4 has the highest score in sacreBLEU, but the sacreBLEU results seem unstable with a lot of variance compared to the others. For setting no. 3, analysis has showed that the results are lower than setting no. 1 because of two reasons: First, the performance of the dataset in setting 2 is not that good. Second, the dataset was translated from German to English, an English that would be different from the English in the dataset from Setting no. 1 due to TSIC, since translation engines tend to create a sentence that sound more like modern day English.

In conclusion, data augmentation would cause lowering translation performance, but augmentation such as using synonyms that merely replaces a word by a different word of the same meaning is worth exploring. Using other datasets to augment to the original dataset is worth exploring too, but it has to be using the same target language as the original dataset.

VI. CONCLUSION AND FUTURE WORK

In this project, a complete system to translate scanned Hieroglyphic symbols to readable English language was proposed. Many Machine Learning and Deep Learning algorithms were utilized in this work and analysis of results was conducted to determine the champion models and the best data settings. This complex problem was divided into two sub-problems: recognizing glyphs in a photo, then translating them into English. Speaking of glyphs recognition, The system proposed in this work have outperformed the state-of-the-art results in literature, as our approach was able to detect glyphs with a mAP of 95% and AR of 75%, and accurately classify 88.5% of the glyphs compared to 66.6% average classification rate obtained by the most recent research done in this area using the same dataset [7].

Regarding the translation task the proposed work succeeded to achieve BLEU score of 59.19 which states that our translation models also outperforms its equivalent models found in literature. As a future work, but higher score might be achieved in translation using more exhaustive data cleansing. Larger volume training data would improve the bias and variability. But since the language is a low-resource language, it will be hard to obtain more data. Since the language is TSIC, there is no much variation in the sentences of the textual corpus. Even though sentences could be different, a lot of words are common between the phrases. Analysis of results also has showed that the lack of variation between the sentences has resulted in some level of bias. Therefore, more analysis of the data and investigation on similarities between training and testing sets should be done. As a future work also more sources of dataset the fits the whole process from glyph recognition to glyphs translation should be found. Maybe finding another source of data written horizontally would be also useful to analyze the performance of our models dealing with it, as evaluation of the proposed work was limited to Pyramid of Unas in which the writings' direction are in vertical fashion.

ACKNOWLEDGMENT

The authors would like to acknowledge and appreciate the support provided by Microsoft company in Egypt, specially Dr. Ahmed Tawfik, and Digital Egypt Builders Initiative (DEBI), through their journey working on this project.

REFERENCES

[1] C. J. van Schaik, L. L. Boer, J. M. Draaisma, C. J. van der Vleuten, J. J. Janssen, J. J. Futterer, L. J. S. Kool, and W. M. Klein, "The lymphatic system throughout history: From hieroglyphic translations to state of the art radiological techniques," *Clin. Anatomy*, vol. 35, no. 6, pp. 701–710, 2022.

[2] B. Manley, *Egyptian Hieroglyphs for Complete Beginners*, 1st ed. London, U.K.: Thames and Hudson, 2012, pp. 1–100.

[3] M. Franken and J. C. van Gemert, "Automatic Egyptian hieroglyph recognition by retrieving images as texts," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 765–768.

[4] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2963–2970.

[5] E. Roman-Rangel, C. P. Gayol, J.-M. Odobez, and D. Gatica-Perez, "Searching the past: An improved shape descriptor to retrieve maya hieroglyphs," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 163–172.

[6] A. Barucci, C. Cucci, M. Franci, M. Loschiavo, and F. Argenti, "A deep learning approach to ancient Egyptian hieroglyphs classification," *IEEE Access*, vol. 9, pp. 123438–123447, 2021, doi: 10.1109/ACCESS.2021.3110082.

[7] R. Elnabawy, R. Elias, and M. Salem, "Image based hieroglyphic character recognition," in *Proc. 14th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Nov. 2018, pp. 32–39, doi: 10.1109/SITIS.2018.00016.

[8] R. Bansal, H. Choudhary, R. Punia, N. Schenk, J. L. Dahl, and E. Page-Perron, "How low is too low? A computational perspective on extremely low-resource languages," 2021, *arXiv:2105.14515*.

[9] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, Feb. 2020.

[10] P. Wiesenbach and S. Riezler, "Multi-task modeling of phonographic languages: Translating middle Egyptian hieroglyphs," in *Proc. 16th Int. Conf. Spoken Lang. Transl.*, 2019, pp. 1–7.

[11] (2021). *Machine Translation for Middle Egyptian-English*. [Online]. Available: https://github.com/fayrose/EgyptianTranslation

[12] (2021). *Ancient Egypt Dictionary*. [Online]. Available: http://www.ancient-egypt.co.uk/transliteration/ancient_egypt_dictionary.pdf

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Computer Vision and Pattern Recognition*. 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 1–9.

[16] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[17] (2021). *Natural Language Processing—Rule Segmentation*. [Online]. Available: https://programmer.group/natural-language-processing-rule-segmentation.html

[18] J. Tang, Q. Wu, and Y. Li, "An optimization algorithm of Chinese word segmentation based on dictionary," in *Proc. Int. Conf. Netw. Inf. Syst. Comput. (ICNISC)*, 2015, pp. 259–262.

[19] J. Wu and Z. Tu, "Reverse image segmentation: A high-level solution to a low-level task," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–13.

[20] M. Norouzi, J. D. Fleet, and R. Salakhutdinov, "Hamming distance metric learning," in *Proc. Neural Inf. Process. Syst. Conf.*, 2012, pp. 1–9.

[21] T. Mondal, N. Ragot, J.-Y. Ramel, and U. Pal, "Flexible sequence matching technique: Application to word spotting in degraded documents," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2014, pp. 210–215.

ASMAA SOBHY received the B.Sc. degree in computer engineering from Ain Shams University, Cairo, Egypt, in 2020, and the M.Eng. degree in electrical and computer engineering (data science and artificial intelligence) from the University of Ottawa, Canada, in 2022. His research interests include computer vision and natural language processing.

MAHMOUD HELMY received the B.Sc. degree in electronics and communication engineering from Alexandria University, Alexandria, Egypt, in 2019, and the M.Eng. degree in electrical and computer engineering (data science and artificial intelligence) from the University of Ottawa, Canada, in 2022. His research interests include physical simulation and reinforcement learning.

MICHAEL KHALIL received the B.Sc. degree in computer science and engineering from German University in Cairo, Egypt, in 2020, and the M.Eng. degree in electrical and computer engineering (data science and artificial intelligence) from the University of Ottawa, Canada, in 2022. His research interests include computer vision and logic in AI.

SARAH ELMASRY received the bachelor's degree in computer science from Alexandria University, Alexandria, Egypt, in 2020, and the M.Eng. degree in artificial intelligence from the University of Ottawa, Ottawa, Canada, in 2022. His research interests include natural language processing and machine learning.

YOUTHAM BOULES received the B.Sc. degree in computer science and engineering from German University in Cairo, Egypt, in 2020, and the M.Eng. degree in electrical and computer engineering (data science and artificial intelligence) from the University of Ottawa, Canada, in 2022. His research interest includes computer vision.

NERMIN NEGIED received the M.Sc. and Ph.D. degrees from the Computer Department, Faculty of Engineering, Cairo University, in February 2012 and July 2016, respectively. She worked as the Educational Quality Manager with the Faculty of Computer Science, October University for Modern Science and Arts (MSA), where she was an Assistant Professor. She was a Teaching Assistant with the Computer Engineering Department, Faculty of Engineering, 6th of October University, from September 2006 to September 2012, where she was a Lecturer, from September 2012 to September 2015. She is currently the Head of data science and artificial intelligence track with the Digital Egypt Builders Initiative (DEBI), Ministry of Communication and Information Technology (MCIT). She is also an Assistant Professor with the Zewail City of Science and Technology. She is also a former Assistant Professor with Cairo University, Nile University, the Arab Academy for Science and Technology and Maritime Transport (AASTMT), and the October University for Modern Science and Arts (MSA). She has published many international journals and conference papers and shared in reviewing many scientific papers. Her research interests include image processing and computer vision, machine learning, artificial intelligence, expert systems, natural language processing, and genetic algorithms.