# A Sustainable Decipher of Egyptian Hieroglyphs

**Karthik Variath Divakaran[1], Shiyas Ahamed S[1], Tushar Renji Thoonkuzhy[1], Yadul Manoj[1], Mr. Ajith S.[2]**

*[1]B.Tech student, Department of Computer Science and Engineering, Rajagiri School of Engineering & Technology, Kerala, India*
*[2]Assistant Professor, Department of Computer Science and Engineering, Rajagiri School of Engineering & Technology, Kerala, India*

---***---

**Abstract -** *This paper presents various approaches for recognizing Egyptian hieroglyphs using machine learning techniques. Egyptian hieroglyphs represent a complex system of writing used in Ancient Egypt for over 3,000 years, which poses a substantial challenge for recognition due to the huge number of symbols and their variability in form and form. Documentation of the culture and language of different civilizations has been a critical component of human history, allowing us to preserve and communicate knowledge across time and space. Our system, which utilizes machine learning and image recognition to recognize and interpret ancient Egyptian hieroglyphs, is an important part of this ongoing movement to preserve and share knowledge for future generations. Our approach leverages the ability of neural networking to accurately recognize and classify hieroglyphs based on their visual features. Our system is able to recognize hieroglyphs from hand-drawn images/doodles of hieroglyphs and a live video feed. The proposed system is created entirely from scratch, with no usage of any pre-existing models or any pre-existing datasets. We trained and evaluated our model on a large dataset of manually created hieroglyphic images. We follow our own architecture of the neural network for training purposes. Experiments show that our model achieves a covetable accuracy for hieroglyph recognition. Our application has the potential to revolutionize the field of Egyptian hieroglyphs decipherment and make them more accessible to scholars and the general public.*

***Key Words*: Ancient Egyptian hieroglyphs, Image recognition and classification, Machine learning, Convolutional Neural Networks, JavaScript, p5.js, ml5.js.**

## 1. INTRODUCTION

Egyptian hieroglyphs are a system of writing that was used in ancient Egypt from about 3200 BCE until the end of the 4th century CE. The hieroglyphic script was used to compose a variety of texts, including administrative documents, religious texts, and literature [7]. Despite the script being ancient, there is still much to be learned about it, and advances in machine learning have created new opportunities for understanding and interpreting these symbols.

Documentation has played a crucial part in the progress of civilization, allowing us to learn from the past and build upon the achievements of previous generations. Throughout history, documentation has taken several forms, ranging from stone tablets and papyrus scrolls to books and electronic media. In recent years, the proliferation of digital technology has brought about a new era of documentation, with huge amounts of data being created and stored digitally every day [17]. As we continue to produce and consume an ever-increasing amount of information, it is more important than ever to develop efficient documentation systems that can manage and maintain this knowledge for future generations. Our work is an important component of this effort, as it seeks to produce a robust and effective system for recognizing and translating ancient Egyptian hieroglyphs. By accurately documenting and translating hieroglyphs, our work can help preserve and share the knowledge and culture of one of the oldest civilizations in human history. This is especially crucial given the fragile nature of many hieroglyphic artifacts and the limited number of experts who are able to study and translate them [17]. By leveraging the recent advancement in machine learning and image recognition, our paper aims to make hieroglyphic translation more accessible and accurate than ever before, contributing to the broader goal of preserving and sharing knowledge across generations.

Neural networks are a type of machine learning that has become increasingly popular in recent years. They are designed to mimic the way that the human mind works, and are especially effective at recognizing patterns in data [1]-[5]. This makes them ideal for image classification tasks, such as identifying hieroglyphs [6]. The use of neural networks for image classification has opened up new possibilities for understanding the hieroglyphic script and its significance. The study of hieroglyphs has a long and rich history. The hieroglyphic script is a complex system of writing that combines both phonetic and ideographic elements. The script consists of over 700 individual signs, each of which have its own significance and significance [7]. The script was first deciphered in the early 19th century by Jean-Francois Champollion, who used the Rosetta Stone to unlock the secret of the ancient script. Since then, scholars have made large strides in understanding the grammar and vocabulary of hieroglyphs. However, there is still much that remains obscure about the script, especially in regard to the way that it was used in daily life [7].

This paper aims to explore the role of neural networks in the identification of Egyptian hieroglyphs. The paper will then describe the architecture of the system and the methodology used for training and testing the neural network. This will include a discussion of the dataset used for training the network, as well as the pre-processing steps that were taken to develop the data. The paper will also describe the architecture of the neural network, including the number of layers, and the activation function used. The paper will then demonstrate the results of the neural network, including its accuracy and performance. Finally, the paper will conclude with a discussion of the possible applications of the developed system for the study of hieroglyphs and linguistics.

## 2. LITERATURE SURVEY

Fujita M., [8], et al. propose a system for detecting hieroglyphs in images using deep convolutional neural networks. Their scheme consists of two stages: first, candidate regions are extracted from the image using a region proposal network, and then a CNN is used to classify each candidate region as either a hieroglyph or background. The authors gathered a dataset of 3,506 images containing hieroglyphs and non-hieroglyphic symbols, and used it to develop and evaluate their system. The authors experimented with various CNN architectures, including VGG-16, ResNet, and Inception-v3, and found that Inception-v3 achieved the best performance with an accuracy of 99.15% on their test set. One interesting aspect of this paper is the author’s use of a visualization technique called Grad-CAM to generate heatmaps showing the regions of the image that were most important for the CNN's classification decision. They found that the CNN was able to focus on characteristic features of hieroglyphs, such as certain shapes and patterns, to make its decisions.

Amin, M. A., [9], et al. propose a hieroglyph recognition system based on deep learning techniques. They use a dataset of 736 hieroglyph images, which were pre-processed using edge detection and binarization techniques. The authors then develop a CNN model on this dataset and achieve an accuracy of 95.65% on their test set. They also compare their approach to traditional computer vision techniques such as the Scale-Invariant Feature Transform (SIFT) and the Speeded Up Robust Feature (SURF) algorithms, which achieve lower accuracy of 89.13% and 91.3%, respectively. The authors conclude that deep learning-based approaches are more efficient for hieroglyph identification than traditional computer vision.

Kulkarni, A. R., [10], et al. the research discusses a hieroglyph recognition scheme based on CNNs. They use a dataset of 1536 hieroglyph images, which were split into 12 classes based on their form and structure. They also performed data augmentation to increase the size of the dataset. The authors use transfer learning to fine-tune pre-trained CNN models, including VGG-16, Inception-v3, and ResNet-50, on their hieroglyph dataset. They compare the performance of the different models and find that ResNet-50 achieved the highest accuracy of 96.9% on their test set. The authors also analyzed the confusion matrix to identify the most challenging hieroglyph classes for recognition. They found that some classes, such as "Eye" and "Reed Leaf," were easier to distinguish than others, such as "Scepter" and "Spear". The authors note that their proposed scheme has several limitations, including the limited size of the dataset and the lack of variability in the images. They suggest that future research should concentrate on collecting a bigger and more diverse dataset of hieroglyphs to improve the performance of recognition systems.

Elshamy, M., [11], et al. discusses a hybrid approach for hieroglyph identification that combines 2 types of features: shape-based features and texture-based features. They used a dataset of 3200 hieroglyph images, which were split into 32 classes based on their form and meaning. For the shape-based features, the authors use the Fourier-Mellin transform (FMT) to extract rotation, scale, and translation invariant features from the hieroglyph images. For the texture-based features, they use a local binary pattern (LBP) operator to extract texture features from the hieroglyph images. The authors use a k-nearest neighbor (k-NN) classifier to sort the hieroglyph image based on their extracted features. They compare the performance of their hybrid approach with that of using only shape-based features or texture-based features. They find that their hybrid approach achieved the highest accuracy of 98.8% on their test set. The authors note that their proposed hybrid approach is robust to variation in rotation, scale, and translation and can handle the identification of hieroglyphs with different textures.

Atallah, Y., [12], et al. bases their approach on a three-stage pipeline, where the input image is first preprocessed to remove interference and enhance the contrast. Then, feature extraction is performed on the preprocessed image using Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP) techniques. Finally, the extracted features are classified using a Support Vector Machine (SVM) algorithm. The authors evaluated their approach using a dataset of 28 hieroglyphs and achieved an accuracy of 89.29%. To further assess the performance of their approach, they compare it to other state-of-the-art techniques, such as Bag-of-Features (BoF) and Convolutional Neural Networks (CNNs). The results show that their proposed approach outperformed the other techniques, achieving high accuracy with low computational complexity. Additionally, the usage of HOG and LBP for feature extraction allows for the detection of texture and pattern information, which are crucial for hieroglyph recognition.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 05 | May 2023 www.irjet.net p-ISSN: 2395-0072

## 3. PROPOSED SYSTEM

The system we have developed is an application that allows users to input hieroglyphic characters, either through text, doodles, or through a live video feed, and get the corresponding English character or its meaning. The application is designed for people who are interested in learning the ancient Egyptian language or scholars who are studying ancient Egyptian texts. Our recognition system is an application that allows users to input hieroglyphic characters, either through text or doodles, and get the corresponding translation of the character. The application is designed for people who are interested in learning the ancient Egyptian language or scholars who are studying ancient Egyptian texts. The system consists of an English-to-hieroglyph converter, a hieroglyphic doodle recognizer, and a hieroglyphic word recognizer that recognizes hieroglyphs from a video feed.

The Hieroglyphic doodle recognizer is one of the core features of the application. It is designed to recognize hieroglyphs from hand-drawn sketches. The doodle recognizer is coded in p5.js, a JavaScript library for creative coding [13], and the model for the recognition is made from ml5.js, a machine learning library that offers a range of pre-trained models for image and audio recognition [14]. The dataset used for training the model is manually created by generating random images of a single hieroglyph character. The model is trained using a convolutional neural network (CNN) architecture, which is a deep learning architecture commonly used for image recognition tasks [15]. The doodle recognizer works by taking an input image of a hand-drawn hieroglyph character and processing it through the CNN model. The model then predicts the corresponding hieroglyphic character from the input image. The hieroglyphic doodle recognizer is a valuable feature for users who want to practice drawing hieroglyphic characters or who have encountered an unfamiliar hieroglyphic character and want to quickly identify it.

The hieroglyphic word recognizer is a feature that recognizes hieroglyphic words from a video feed. The word recognizer is also coded in p5.js, and the model for the recognition is made from ml5.js. The dataset used for training the model is manually created by recording videos of each hieroglyph word and extracting frames of the video to provide as input for the training. The word recognizer works by taking an input video stream and processing each frame through the CNN model. The model then predicts the corresponding hieroglyphic word from the input video. The Hieroglyph word recognizer is a valuable feature for users who want to translate ancient Egyptian texts that are in the form of hieroglyphic inscriptions on temple walls or papyri.

### 3.1 System Architecture

The architecture of our hieroglyphic recognition system consists of three main components: an English-to-hieroglyph converter, a hieroglyphic doodle recognizer, and a hieroglyphic word recognizer. The architecture diagram is depicted in Fig -1.

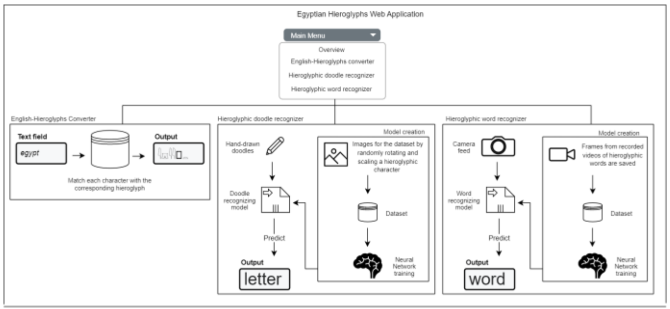


**Fig -1:** Architecture diagram of the proposed system

The English-to-hieroglyph converter is responsible for converting English words into Hieroglyphic symbols. This component uses an internally stored database of character mappings to generate a hieroglyphic symbol sequence for the corresponding English text. The hieroglyphic doodle recognizer is designed to recognize hieroglyphic symbols from freehand drawings or doodles made by users. This component uses machine learning techniques such as convolutional neural networks (CNNs) to recognize the shape and pattern of the doodles and match them to known hieroglyphic symbols in its database. The hieroglyphic word recognizer is responsible for recognizing hieroglyphic symbols that have a word meaning from a video feed. This component uses similar machine learning techniques by using a model trained from recorded video frames to distinguish individual hieroglyphic symbols from the video feed and then match them to known words.

Figure -2 depicts the swimlane diagram of the same proposed system. The swimlane diagram would begin with a section where the user is able to choose between each part of the system. For the English-to-hieroglyph converter, the input is in the form of English text. The converter would then convert the English text into Hieroglyphic characters by phonetically matching each character. The Hieroglyphic doodle recognizer uses a trained CNN model for its recognition purpose, where the input is in the form of hand-drawn doodles. The third part is the word recognizer where a CNN model is used for classification purposes. There is an extra lane for the neural network training process for developing the model for the doodle and word recognizers.

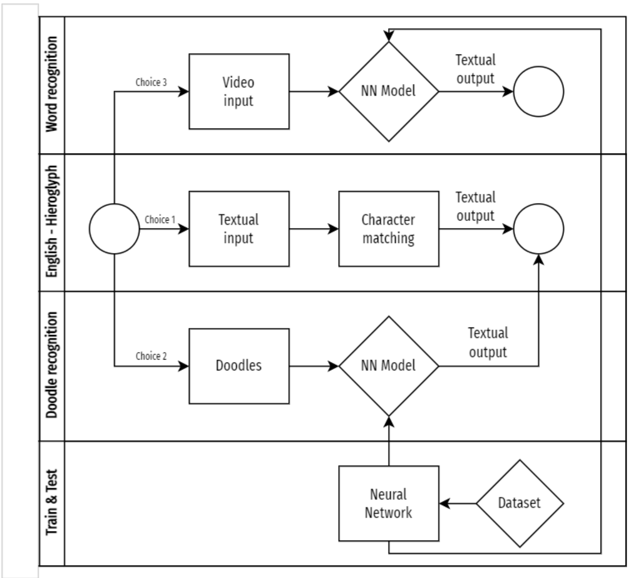


**Fig -2:** Swimlane diagram of the proposed system

The architecture of the model for the doodle recognizer is composed of several layers and is shown in Figure -3.

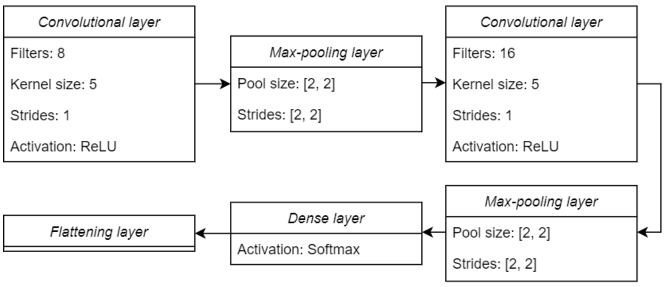


**Fig -3:** CNN architecture of the doodle recognizer

The first layer is a convolutional layer, which applies 8 filters of size 5x5 to the input image. The stride is set to 1 and the activation function is Rectified Linear Unit (ReLU). This layer extracts features from the input image. The second layer is a max pooling layer, which reduces the spatial dimensions of the feature maps by a factor of 2. The pooling size is set to 2x2 and the stride is also set to 2. The third layer is another convolutional layer, which applies 16 filters of size 5x5 to the feature maps from the previous layer. The stride is set to 1 and the activation function is ReLU. This layer further extracts more complex features from the previous layer. The fourth layer is another max pooling layer, which reduces the spatial dimensions of the feature maps by a factor of 2. The pooling size is set to 2x2 and the stride is also set to 2. The fifth layer is a flattening layer, which converts the feature maps into a 1-D vector by stacking them. The last layer is a dense layer with the softmax activation function. This layer takes the flattened feature vector as input and produces a probability distribution over the 26 output classes. The input to the network is a 64x64x4 image, where the last dimension corresponds to the 4 color channels (red, green, blue, and alpha). The output of the network is a probability distribution over 26 classes, which represents each letter of the English alphabet.

The architecture of the word recognizer contains extra layers for the network due to the requirement for classification from a live video feed. It is depicted in Figure-4.

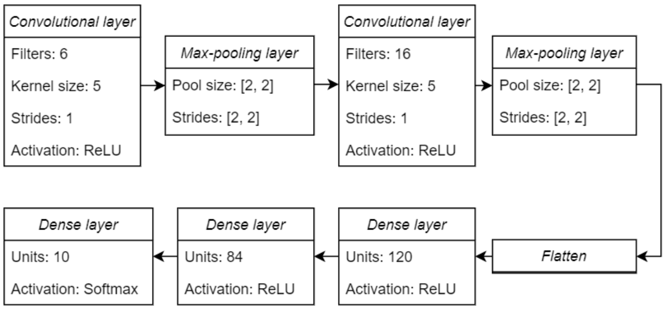


**Fig -4:** CNN architecture of the word recognizer

The first layer is a convolutional layer with 6 filters and a kernel size of 5x5 pixels. The layer applies the ReLU activation function to the output. The second layer is a max pooling layer with a pool size of 2x2 pixels and a stride of 2 in both dimensions. The third layer is another convolutional layer with 16 filters and a kernel size of 5x5 pixels. The ReLU activation function is applied to the output. The fourth layer is another max pooling layer with the same parameters as the previous one. The fifth layer is a flattening layer that converts the output of the previous layer into a 1-dimensional array. The sixth layer is a dense layer with 120 units and the ReLU activation function. The seventh layer is another dense layer with 84 units and the ReLU activation function. The eighth and final layer is a dense layer with 10 units and the softmax activation function, which produces a probability distribution over the 10 possible output classes. The input to the network is a 64x64x4 image, where the last dimension represents the four color channels (red, green, blue, and alpha) of the image. The output of the network is a probability distribution over the 10 possible output classes.

## 3.2 Experimental setup

One of the key challenges in developing our system was the limited availability of training data. As a result, we resorted to manually creating our own training dataset. This is done by generating images via code by the manipulation of already existing images of ASCII characters of hieroglyphs and randomizing them. This is done as follows: Initially, the canvas is created with a width and height of 400 pixels. The image file of the character is loaded before it is manipulated. The background of the canvas is set to white and a random x and y position is generated and the image is placed at the coordinates, while ensuring that the image will not be placed too close to the edges of the canvas. The image is also then randomly scaled and rotated. After sufficient manipulations are done, the current frame is saved as a PNG file in a folder after downscaling it down to a dimension of 64x64 pixels. The result of the image generation is shown in Figure -5.

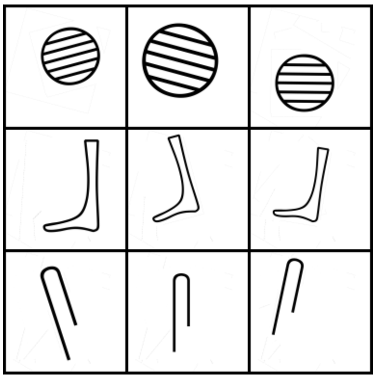

**Fig -5:** Sample images from the dataset for the hieroglyphic characters for the hieroglyphic doodle recognizer

We follow a similar methodology for generating the dataset for the hieroglyphic word recognizer. In this particular case, we created our own dataset by recording videos of the words we wanted to recognize. A webcam was used to capture these videos. From these videos, we then extracted frames to use as individual data points in their dataset. A frame is a still image that is captured from a video at a specific point in time, and in this case, it would represent a single observation in the dataset. Once the frames are extracted, they are labeled or annotated with information about what is in the image. Figure -6 shows the result of the mentioned steps.

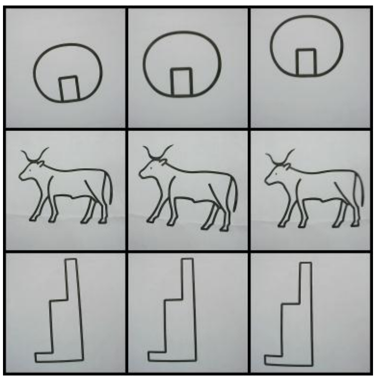

**Fig -6:** Sample images from the dataset for the hieroglyphic words for the hieroglyphic word recognizer

Our system is primarily coded in p5.js and ml5.js. p5.js is a library that simplifies the process of creating interactive and creative graphics, animations, and visualizations for the web. It is built on top of the JavaScript programming language and provides a set of functions and methods that make it easy to make shapes, manipulate images, play sounds, and make user interfaces [13]. ml5.js, on the other hand, is a library that simplifies the process of integrating machine learning into web applications. It is built on top of TensorFlow.js, a popular machine learning library for JavaScript, and provides a set of pre-trained models and tools for training and evaluating custom models. ml5.js can be used for a variety of machine learning tasks, such as image classification, object detection, text generation, and audio analysis [14]. The requirements for development in p5.js and ml5.js are relatively straightforward. They consist of a text editor or integrated development environment (IDE) to write your code, a web browser to view and test the p5.js sketches and ml5.js models, and a server or cloud-based platform to host the trained model [16].

## 4. PERFORMANCE EVALUATION

Both components of our system, i.e., the doodle recognizer and the word recognizer were trained for 35 epochs with a dataset consisting of 4,030 images and 2,000 images respectively. We found that the model producing the best accuracy were of the architecture mentioned in the previous section. The doodle recognizer and word recognizer produce a loss of 0.09 and 0.10 respectively. This is depicted in figure 6.1. As seen from figures 7 and 8, the end architecture fits the data well as the validation loss has more or less converged after the 30th epoch. Furthermore, fig 6.2 shows the accuracies achieved by doodle recognizer and word recognizer: 1. Final accuracy: 89.97% & 85.64. top-3 accuracy: 86.71% & 82.44%.

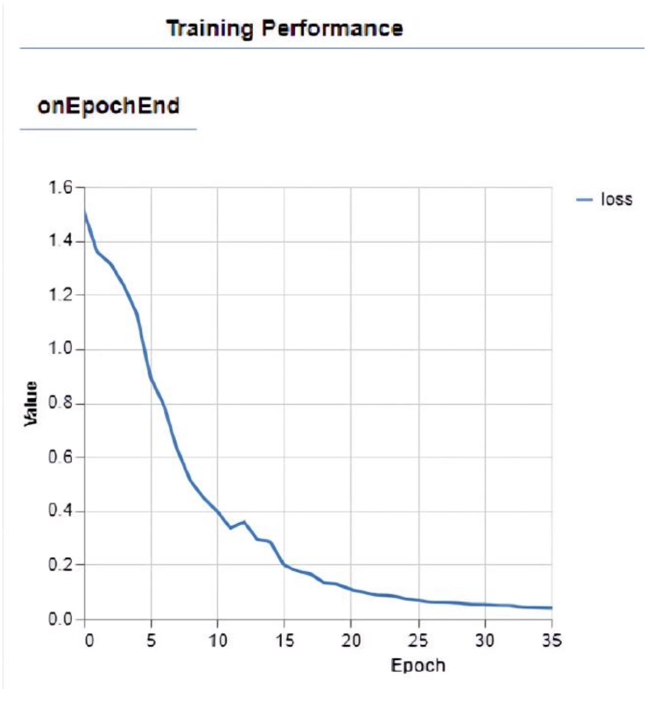


**Fig -7:** Graph depicting the training loss in each epoch for the doodle recognizer

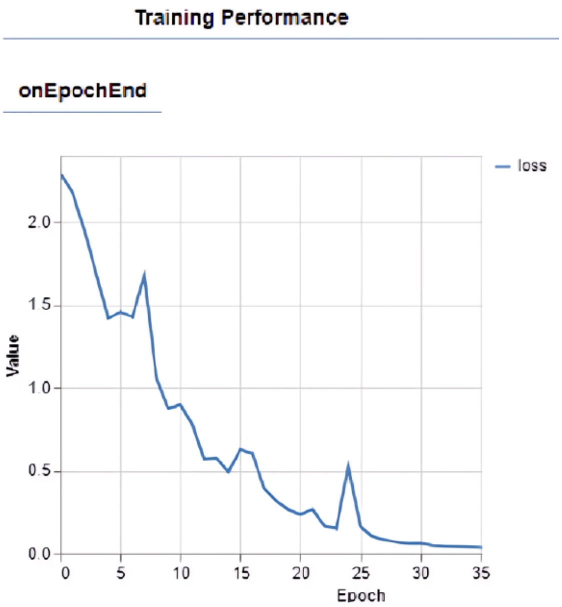


**Fig -8:** Graph depicting the training loss in each epoch for the word recognizer

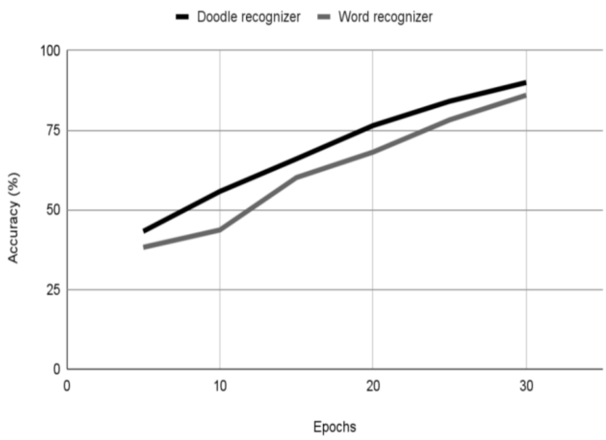


**Fig -9:** Graph representing the accuracy % of the doodle recognizer and word recognizer

## 5. RESULTS

Figures 10, 11, 12, 13, 14 demonstrate the Egyptian hieroglyphs recognition system we have developed. The application can be run on any normal web browser with webcam facilities.

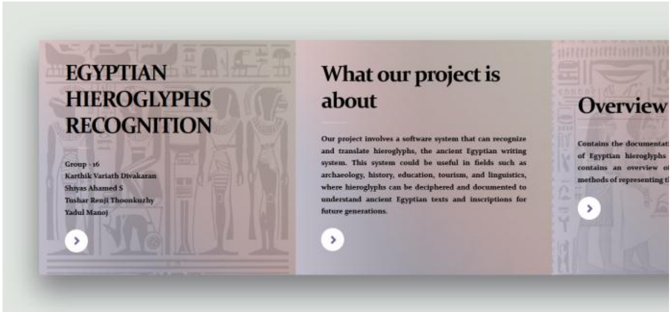


**Fig -10:** Main menu of the application


**Fig -11:** Overview section of the application


**Fig -12:** English-to-Hieroglyphs converter


**Fig -13:** Hieroglyphic doodle recognizer


**Fig -14:** Hieroglyphic word recognizer

## 6. CONCLUSIONS

In this paper, we proposed to develop a system that is able to recognize Egyptian hieroglyphs through two main methods, hand-drawn doodles and through a live video feed. Our system is developed purely from scratch without any assistance from pre-trained models or readily available datasets. Everything in our system is created manually, including the models used for classification purposes. Additionally, our application also includes an English-to-hieroglyph converter, which can be used to quickly familiarize with the hieroglyphic language system. Another important aspect to highlight in the application is the performance. The system was able to generate a final accuracy of 89.97%. As part of the potential areas for future research and development, the existing capabilities of our system could be expanded to recognize more complex hieroglyphic inscriptions or to translate hieroglyphic texts into other languages. The system's performance could also be improved, such as through the integration of additional machine learning algorithms or the use of more sophisticated computer vision techniques. Furthermore, our system can be expanded to a variety of fields like historical and archaeological studies, and language preservation and is applicable to other scripts and languages with a similar level of complexity. This could fundamentally alter the field of study of Egyptian hieroglyphs and increase public and scholarly interest in them. Our work is a testament to the enduring importance of documentation and its use in preserving and sharing knowledge across generations. By contributing to this effort, our work seeks to ensure that the rich history and culture of ancient Egypt are not lost to time and that succeeding generations have access to the knowledge and wisdom of our predecessors.

## STUDY LIMITATIONS

Hieroglyphic inscriptions can be highly complex, with various grammatical and semantic structures. This made it challenging to develop a system to accurately recognize and interpret the meaning of individual Hieroglyphs and their interactions within a larger inscription. Due to this difficulty, we decided to develop an application that recognizes rudimentary hieroglyphic characters and words. Finally, the number of individuals who are proficient in reading and writing Hieroglyphs is relatively small, which can limit the resources available for maintaining and updating such a system.

## REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016. [Online]. http://www.deeplearningbook.org M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences," Rev. Mod. Phys., vol. 91, Dec. 2019, Art. no. 045002, doi:10.1103/RevModPhys.91.045002.

[3] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and M. Prabhat, "Deep learning and process understanding for data-driven Earth system science," Nature, vol. 566, pp. 195–204, Feb. 2019.

[4] K. Yu, A. Beam, and I. Kohane, "Artificial intelligence in healthcare," Nature Biomed. Eng., vol. 2, no. 10, pp. 719–731, 2018.

[5] C. Scapicchio, M. Gabelloni, A. Barucci, D. Cioni, L. Saba, and E. Neri, "A deep look into radiomics," La Radiol. Medica, vol. 4, pp. 1–16, Jul. 2021.

[6] J. Duque-Domingo, P. Herrera, E. Valero, and C. Cerrada, "Deciphering Egyptian hieroglyphs: Towards a new strategy for navigation in museums," Sensors, vol. 17, no. 3, p. 589, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/3/589

[7] A. Gardiner, Egyptian Grammar. Oxford, U.K.: Griffith Institute, 1957.

[8] Fujita, M., Tamura, Y., Takiguchi, T., & Aoki, Y. (2019). Hieroglyph detection based on deep convolutional neural networks. International Journal of Heritage in the Digital Era, 8(1), 129-138.

[9] Amin, M. A., Abdelsamea, M. M., & Mohamed, M. A. (2019). Hieroglyphic Recognition System Based on Deep Learning Techniques. Journal of King Saud University-Computer and Information Sciences, 31(1), 81-89.

[10] Kulkarni, A. R., & Chandak, M. S. (2019). Recognition of Egyptian Hieroglyphics using Convolutional Neural Network. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(2), 404-408.

[11] Elshamy, M., Helala, M., & Elgharib, M. (2018). Hieroglyph recognition using deep convolutional neural networks. International Journal of Computer Applications, 179(30), 25-30.

[12] Atallah, Y., Abd El Mageed, N. 2019. A hybrid approach for ancient Egyptian hieroglyph recognition. Journal of Imaging, 5(5), 60.

[13] B. Mechtley and A. Johnson, "Teaching and Learning Creative Coding with p5.js: A Case Study", Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, March 2017. DOI: 10.1145/3017680.3017724

[14] S. Yang, S. Goh, and G. Ressler, "ml5.js: Machine Learning for the Web in JavaScript", Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, April 2018. DOI: 10.1145/3173574.3174218

[15] L. Deng and D. Yu, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2014.2344017

[16] K. Santhosh and S. K. Venkatesan, "Visualizing Convolutional Neural Networks with ml5.js and p5.js", Proceedings of the 2019 International Conference on Computer Vision and Image Processing, December 2019. DOI: 10.1109/ICCVIP46957.2019.8977402

[17] R. Parkinson, "The Rosetta Stone and the Decipherment of Hieroglyphs," Antiquity, vol. 92, pp. 1484-1497, 2018.

## BIOGRAPHIES

**Karthik Variath Divakaran** is a B.Tech Computer Science and Engineering student at Rajagiri School of Engineering & Technology, Kerala, India.

**Shiyas Ahamed S** is a B.Tech Computer Science and Engineering student at Rajagiri School of Engineering & Technology, Kerala, India.

**Tushar Renji Thoonkuzhy** is a B.Tech Computer Science and Engineering student at Rajagiri School of Engineering & Technology, Kerala, India.

**Yadul Manoj** is a B.Tech Computer Science and Engineering student at Rajagiri School of Engineering & Technology, Kerala, India.

**Mr. Ajith S.** is an Assistant Professor in the Department of CSE of Rajagiri School of Engineering & Technology. He completed his B.E in CSE from SDMCET, Dharwad, and M.Tech in Computer Science from SRM University, Chennai.