Automatic Egyptian Hieroglyph Recognition by Retrieving Images as Text

Nivas Narayanasamy Varun Rajavelu Smitha Desai

Advisor: Dr. Zoran Duric

May 2017

Abstract

The aim of the project is to develop a system that automatically recognizes ancient Egyptian Hieroglyphs from pyramid images. The process consists of two stages: training and testing. Training phase involves, computing descriptors of the training set, clustering them to obtain the visual words and modelling the data using support vector machine. Testing involves locating the hieroglyphs from the pyramid slab, cutting and preprocessing the individual hieroglyph, computing its descriptors and histogram, then predicting the hieroglyph. The project is entirely based on paper Egyptian Hieroglyph Recognition by Retrieving Images as Text by Morris Franken and Jan C. van Gemert's [1]

1 Introduction and Related Work

The ancient Egyptian hieroglyphs has been a mysterious writing system. During 1822 Thomas Young and Jean-Francois Champollion discovered that each hieroglyphs resembles a sound and multiple hieroglyphs form a word. By deciphering the ancient Egyptian hieroglyphs, much of their culture has been uncovered.
The images are taken from one pyramid to eliminate issues with different writing style. The hieroglyphs are chosen from pyramid of Unas. The Egyptian writing system does not contain any spaces. So, it is difficult for inexperienced eye to distinguish between different words. Each word can be written in a number of different ways as long as the pronunciation remains the same. Most of the hieroglyphs do not have any sound related to them and are used to illustrate the meaning of the word.
The multimedia tools have aided analysis and study of cultural, historical and artistic content. Current work on scene text detection and recognition [4]. Another example is proposed by Karaoglu et al.[5] where text recognition is used to aid in object recognition task. Related work on automatic hieroglyph recognition focuses on mesoamerican culture and Mayan hieroglyhs. The HOOSC

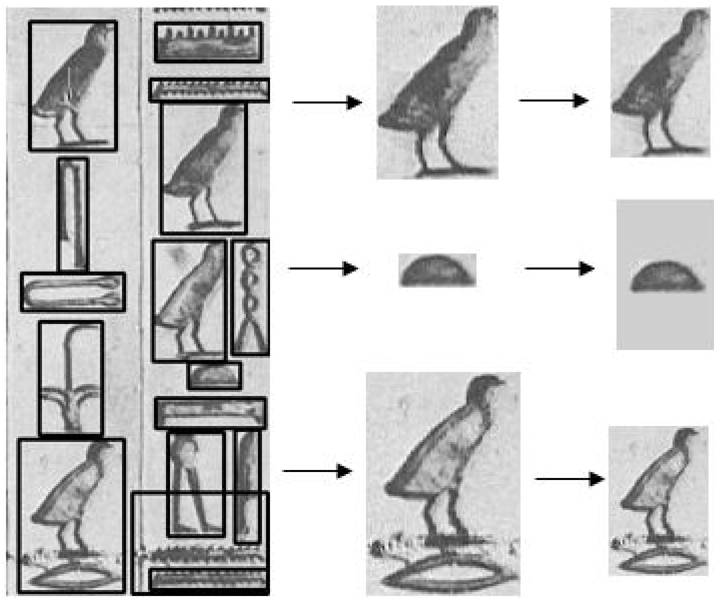

Figure 1: Locating, Extracting and Patching

descriptor was developed[2], which is combination of HOG[3] and shape content. The HOOSC descriptor can be used for direct matching or with bag of words. Other work extracts detailed line segments for Maya hieroglyph matching.

2 Technical Approach

2.1 Locating Hieroglyphs

Every hieroglyph on the slab is located and saved into an image. The threshold is applied to the image to convert into a binary image. The image is then inverted and eroded to get a solid path of the object. The contour is calculated for the image and bounding box is drawn around it. All the input images are resized into 50 x 75. As the resizing distorted the images, patching is applied around the detected image to maintain information. An image of size 50 x 75 pixels with uniform background is generated and later the hieroglyph cut from the slab is placed on this image. The process is illustrated in Figure 1.

2.2 Image features

Feature descriptors are used to extract image features that can be used for various tasks like detection, segmentation. The feature descriptor is a representation of an image that simplifies an image by extracting useful information and disposing extraneous information.

HOG Descriptor

Here the descriptor is computed using the Histogram of Orientated Gradients(HOG). This feature descriptor converts an image of size width x height x 3(channels) into a feature vector of length n. The HOG descriptor uses the gradient and magnitude information of each pixel. They are computed using the equation (1) and (2).

$$Magnitude_{x,y} = \sqrt{dy_{x,y}^2 + dx_{x,y}^2} \tag{1}$$

$$Angle_{x,y} = atan(\, dy_{x,y},\, dx_{x,y}) \tag{2}$$

where $dy_{x,y}$ represents the y derivative at location (x,y) in the image.
Gradients is a useful information as the value is large around the edges and corners. The detection window size for the HOG descriptor is taken to be the size of the input image, since we needed to compute a single descriptor for the entire image. The cell size is 16x16, the block size is 1x1 and the number of orientations is 8. HOG descriptors returns a 1x96 array (feature vector of length 96).

2.3 Classification

The input images are classified based on the descriptors calculated in the previous stage. Here the classification is achieved by matching with bag of words. Other methods

Bag Of Words

All the features from the training set are stacked up. These features are clustered using K-means to obtain the K most important visual words. These visual words are then used to construct a histogram for all the images, this is done by finding the most similar visual word for each feature in an image and simply counting how many times each word occurs in that image. Based on the count value of each visual word in the test set image, it is classified into one of the classes.

In the penultimate step, we create the classifier using Linear Support Vector Classification(SVC). The histograms of each image and its classes, are fit into the SVC. Inorder to persist the fitting of the data, we dump this into a pkl file. The overall process is shown in Figure 2.

In order to classify, all the hieroglyphs in a pyramid slab, the entire pyramid slab is taken as an input, individual hieroglyphs are detected and processed. Then the descriptors and the histograms of the images are found using the similar techniques mentioned above. Each image is compared with the existing training model. Once a class is predicted , the class is outputted for each image.

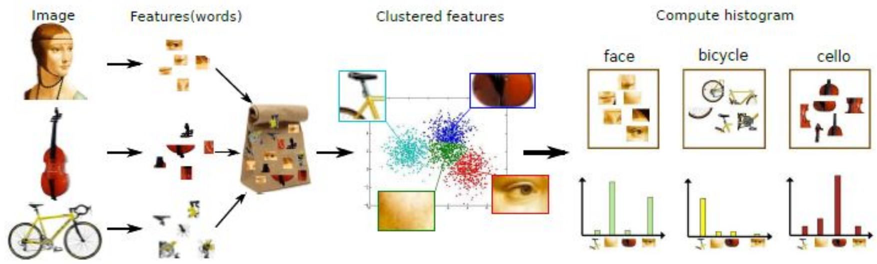

Figure 2: Bag of Words

3 Results and Discussions

For a preprocessed test data set of size 455, Our method (HOG/BOW/SWM without n-grams/Language model) fetched us an accuracy of 69.66%. Other descriptors such as SIFT/SURF gave us an accuracy of 45% - 50%. An accuracy of 40% is achieved when the raw images of the pyramid slabs are used as input. A confusion matrix for 4 classes and 10 test images is given in Figure 3. With 10 images , 6 images are classified correctly, resulting in an accuracy of 60%. Further metrics for the entire test data set is given in Figure 4.

Nivas worked on getting the descriptor of the image and generating the classifier.Varun implemented Bag of words model was implemented. The Confusion matrix and result metrics were generated by Nivas and Varun. Myself, worked on initial step of locating the hieroglyph, extracting and pre-processing and collaborated with other team members in documenting the process.

Nivas Narayanasamy: I believe the selection of the data set was very specific and relevant to the problem. We were so determined to use a single descriptor for the entire image, rather than using multiple descriptors and wanted the entire process to be not CPU intensive. We chose HOG(Histogram of Oriented Gradients) for descriptors and BOW(Bag of Words) method for classification. HOG/BOW with SVM for classification turned out to be a good choice and we achieved a accuracy of 68% without n-grams or language model (Original Paper: HOOSC/RANSAC with n-grams 88%) . With more time and resources I believe, we could have refined the classification using a language model/n-grams and could have used state of the art deep learning image classification models like Keras or AlexNet for a better classification.

Smitha Desai:The classification done using the pre-processed images ran efficiently and produced good accuracy. The stage where the hieroglyphs were located could not be completed accurately because of the time constraints. This stage was difficult because the extracted hieroglyphs were of different sizes which had to be resized for further processing in the next stage. Resizing was a challenge as the image would get distorted.

Varun Rajavelu:Bag of words worked with good results (with HOG). The results of SURF/SIFT was not up to the mark to be used. Given more time,term

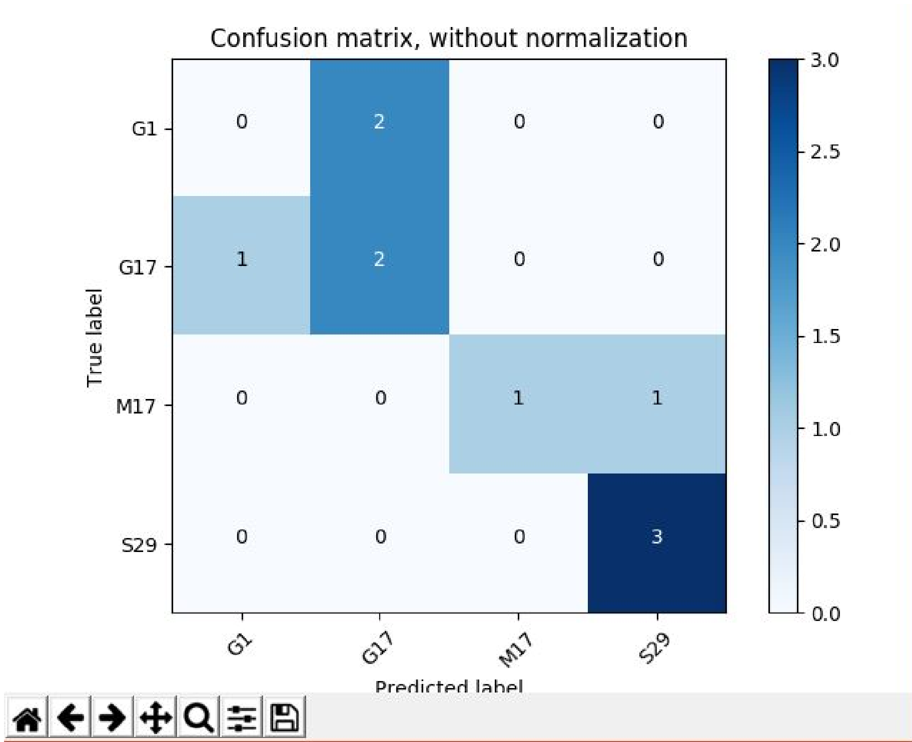

Figure 3: Confusion Matrix

frequency-inverse document frequency, could have been incorporated which would give better result (increased accuracy with classification). Also, due to time and the complexity involved, implementation of language model have been excluded in the current version.

4 Conclusion

The system classifies about 68% of the input correctly. A better classification was achieved by the author, incorporating language models. Two different methods have been implemented by the author where the first method uses a lexicon to find Egyptian words in the top-n proposed classification. The second method uses an n-gram approach which determines the likelihood of certain hieroglyphs occurring in a sequence. State of the art deep learning image classification models like AlexNet, Inception, Keras can be used to increase the accuracy of the classification task.

```
True Positive: 316
False Positive: 139
True Negative: 33986
False Negative: 139
True positive rate: 0.694505494505
True negative rate: 0.995926739927
Positive predictive value: 0.694505494505
Negative predictive value: 0.995926739927
False positive rate: 0.00407326007326
False negative rate: 0.305494505495
False discovery rate: 0.305494505495
Overall accuracy: 0.991960670908
```

Figure 4: Result

References

[1] Morris Franken and Jan C. van Gemert *Automatic Egyptian Hieroglyph Recognition by Retrieving Images as Text*, ACM Multimedia Conference, 2013.

[2] E. Roman-Rangel, C. Pallan Gayol, J.M. Odobez, and D. Gatica-Perez *Searching the past: an improved shape descriptor to retrieve maya hieroglyphs*, ACM MM, 2011.

[3] N. Dalal and B. Triggs *Histograms of oriented gradients for human detection*, CVPR, 2005.

[4] B. Epshtein, E. Ofek, and Y. Wexler *Detecting text in natural scenes with stroke width transform*, CVPR, 2010.

[5] S. Karaoglu, J. van Gemert, and T. Gevers *Object reading: Text recognition for object recognition*, ECCV-IFCVCR Workshop, 2012.