COMPUTERIZED HITTITE CUNEIFORM SIGN RECOGNITION AND DATA MINING APPLICATION EXAMPLES

A. Ziya Aktas [1]* and Tunc Asuroglu [2]

[1] Department of Computer Engineering, Baskent University, Ankara, Turkey
Email: zaktas@baskent.edu.tr
[2] Department of Computer Engineering, Baskent University, Ankara, Turkey
Email: tuncasuroglu@baskent.edu.tr

*** Corresponding Author**

ABSTRACT

Purpose *–In the paper a research on reading Hittite cuneiform signs using computer techniques and perform data mining examples is summarized.*

Design/methodology/approach *– To read Hittite cuneiform signs, 13 image processing algorithm used in this paper. Algorithms used on 149 signs. Also in data mining part, Clustering and classification algorithms are applied.*

Findings *– Results revealed that using image processing algorithms cuneiform signs can be read from tablets and also by using data mining techniques signs can be categorized by geometrical features.*

Originality/value *– Hittites were one of the greatest world powers during B.C.1600 – 1250 in Anatolia. Hittite language has been one of the oldest members of the Indo-European language family. The Hittites used cuneiform signs to write on wet clay tablets and baked them to be permanent and durable. The study of Hittite language grammar rules has followed reading manually the cuneiform tablets. It, however, takes a long time and effort, especially, expertise. Many more tablets are still waiting under and over ground in Anatolia to be read. Being able to read the signs on cuneiform clay tablets, using computer-aided techniques would be a significant contribution not only to Anatolian and Turkish but also to human history.*

KEYWORDS: CUNEIFORM SIGN RECOGNITION, DATA MINING, HITTITE CUNEIFORM SCRIPT, IMAGE PROCESSING AND COMPUTER VISION, OPTICAL CHARACTER RECOGNITION.

1. INTRODUCTION

In Anatolia the kingdom and empire of the Hittites had ruled nearly half a millennium during the years BC 1650-1200. They were considered one of the greatest world powers of that time. Hittite language that the Hittites used is one of the oldest members of the Indo-European language family. Member of this language family, the Hittite language, is one of the oldest examples that is still readable and grammar rules are known. Because of this property Hittites and Hittite language have become interesting and historically valuable in western countries like USA, Germany and England, including some others.

Grammar rules of Hittite language were revealed in the beginning of the 20th century (in 1915) by Czech scientist Bedrich Hrozny (Karasu, 2013). After Grammar rules of Hittite language were revealed; up to now reading, translating and interpreting of Hittite cuneiform scripts are based on human manual efforts. In order to read cuneiform scripts and to do necessary translations, expert people are needed, thus these processes require financial support, they take long time and are exhausting.

This paper includes a summary of a computerized work that could help translation of signs in Hittite cuneiform tablets to Latin script. The study carried out the process of reading Hittite cuneiform signs in tablets by using various computer based image processing techniques and match with signs that were already stored in databases and translated into Latin script. Techniques that are used in reading Hittite cuneiform signs are compared based on their sign matching performances. Some techniques that speed up the process of matching cuneiform signs were also proposed during the study that has been performed.

In Data mining part of the study, categorization of Hittite cuneiform signs based on their geometrical features were carried out to speed up the process of reading cuneiform signs in tablets by categorizing similar signs. After categorization of cuneiform signs, data mining classification algorithms are applied. Comparative classification performances of applied algorithms were reported. Paper finishes after conclusions with relevant references.

2. HITTITES AND HITTITE CUNEIFORM SCRIPT

The Hittites were one of the first communities that had adopted the concept of archive-library. The Hittites used cuneiform signs to write about various topics such as king diaries, state treaties, laws, religious ceremonies and letters on wet clay tablets. Relatively very few of those tablets were discovered and translated, most of them are still buried in the ground. Hittite cuneiform tablets that were obtained from Corum Boğazköy were added to memory of the world register by UNESCO in 22th January 2002.

In human history about 5000 years ago Sumerians discovered pictograph in Mesopotamia and many years later it is evolved to another type of script which is called cuneiform by Akkadians. The cuneiform script was brought to Anatolia by Akkadians during trading and by the time Hittites adopted this cuneiform and had developed a script of their own called “Hittite cuneiform”. In Hittite cuneiform, basic signs, that are given in Fig 1, form the cuneiform script had been written on wet clay tablets using sharp edged cane or similar tools by clerks. After clerks wrote scripts on tablets they baked tablets to become permanent and durable.

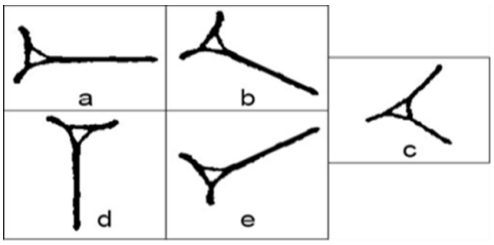


Figure1 Basic signs in Hittite cuneiform script

In 1989 C. Ruster and E. Neu published a Hittite cuneiform sign dictionary named HZL (Hethitisches Zeichenlexikon) which includes Hittite cuneiform signs and their meanings (Ruster and Neu, 1989). In HZL dictionary, signs are indexed by their sign numbers. This number is called HZL number. In Hittite studies, HZL numbers refer to signs.

3. RELATED WORK

In literature, the first study on computerized Hittite cuneiform signs in Turkey was made at METU in 1988 (Aktas and Gürsel, 1988), (Gürsel, 1988).Another study about computerized Hittite cuneiform sign recognition was made at Baskent University in 2014 (Dik, 2014). One of the most recent studies about computerized Hittite cuneiform script was made in Baskent University in 2015 (Yesiltepe, 2015). Another recent study was performed by Aşuroğlu (Asuroglu, 2015) which is summarized in this paper. Hittite cuneiform script is a collection of signs therefore character recognition studies based on Chinese, Arabian, Japanese, Bangla and Tamil alphabet, as well as Sumerian cuneiform script, which are different from Latin alphabet can be added as related work(Li and Woo, 2000), (Anthony et al., 2012) (Logar et al., 1994), (Pornpanomchai et al., 2001).

Dik (Dik, 2014) made a study on the automatic translation of Hittite cuneiform signs. In this study a digital dictionary database were developed which included Hittite cuneiform signs and an approach about Hittite cuneiform sign recognition by using Hausdorff Distance algorithm was proposed. She studied on the first Hittite sentence that Hrozny solved.

Tyndall (Tyndall, 2012) applied data mining algorithms to assembly transcripted cuneiform tablet parts that belong to a single tablet. Inventory number of tablet (given by Hittite experts) is assigned as class information and then broken parts matched by Hittite experts are accepted as single class and dataset is created from these broken parts. Experiments were made with Naïve Bayes and Maximum Entropy classifiers and classification performances were given.

Edan (Edan, 2013) applied data mining algorithms to Sumerian cuneiform signs. Signs were acquired by a scanner and a pre-processing is applied to reduce noise. Then feature vectors were created which consisted of horizontal and vertical distributions of cuneiform signs and number of connected components. K-means clustering algorithm was applied to find classes of cuneiform signs. After clustering, artificial neural network algorithm was applied to cuneiform signs and classification performance was evaluated.

Yousif et al. (Yousif et al., 2013) proposed an algorithm called Intensity Curve to perform recognition of Sumerian cuneiform signs. In that algorithm first all signs were divided into equal horizontal partitions and in every partition pixel values and locations were calculated. After calculations, these values were transformed into a curve and local minimum values of curve created a feature vector. The same procedures were applied to vertical partitions too. Noisy, enlarged and reduced size versions of signs were used to query database which holds original signs and matching performance of Intensity Curve algorithm is reported.

Ahmed (Ahmed, 2012) proposed an algorithm called Symbol Structure Vector to perform recognition of Sumerian cuneiform signs. This algorithm starts with skeleton extraction of cuneiform signs. After skeleton extraction, features such as bending points and connection points of sign were calculated. Features were saved in database for later use. Real-time drawings of a cuneiform signs are compared to sign database and matching performances were reported.

Sundar and John (Sundar and John, 2013) made a study on Tamil character recognition. For every Tamil character two different feature vectors were calculated. First one was calculated by using HOG algorithm, second one consists of geometric aspects of a character. Using artificial neural network these two feature vectors were compared and results were reported as classification performance.

4. Hittite Cuneiform Sign Recognition

4.1 Acquire Digital Image of Hittite Cuneiform Signs

Portal Mainz website is used as a source to acquire digital images of Hittite cuneiform signs. Portal Mainz (http://www.hethport.uni-wuerzburg.de/HPM/index.html) is a website that is part of the Wurzburg University website. There are many Cuneiform tablet pictures available in this website. In this paper these tablet pictures were used as a source for cuneiform signs. Also in Portal Mainz website there is a digital list that includes all of Hittite cuneiform signs and their HZL index numbers. This digital list is created by Sylvie Vanseveren (V.S.). We call this digital list V.S. sign list in the paper. V.S. list includes high resolution pictures of all Hittite cuneiform signs. So this list acts as a database for cuneiform signs in our study. When finding the equivalent of signs in tablet V.S. digital list is used as a baseline for cuneiform signs. This process is depicted in Fig 2.

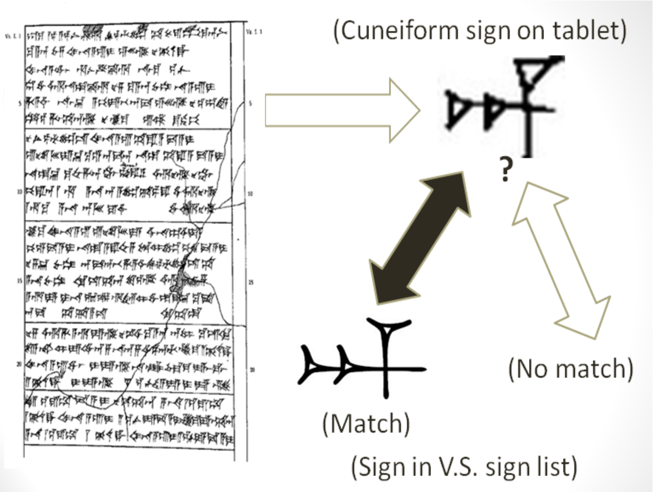


Figure 2Finding the V.S. list match of a cuneiform sign in a tablet

We take screenshot of tablet pictures and crop cuneiform signs in Microsoft Paintand resize them to 36x48 pixels. These signs are the signs that we perform queries on V.S. digital list to find a match. The same procedure applies to V.S. Digital list when cropping cuneiform signs from list. After acquiring digital image of cuneiform signs, several preprocessing phases applied to signs in recognition algorithms.

4.2 Image Processing Algorithms for Hittite Cuneiform Sign Recognition

In the study thirteen algorithms were used for computer based Hittite cuneiform sign recognition. Some of these algorithms were created by using functions that belong to the MATLAB toolbox (e.g. Algorithm 1). Another example is algorithm 2 that belongs to the MATLAB library. Algorithms like 3, 4 and 5 belong to Open CV library.

1. B.U. Algorithm (Baskent University): Division of sign image in to regions and calculation of an error rate (difference of number of black pixel in every region).
2. MATLAB Regionprops library. This library helps to calculate geometric features of an image.
3. SIFT algorithm (Scale Invariant Feature Transform) (Lowe, 2004).
4. SURF algorithm (Speeded up Robust Features) (Herbert et al., 2006).
5. FAST algorithm (Features from Accelerated Segment Test) (Rosten and Drummond, 2006).
6. BRISK algorithm (Binary Robust Invariant Scalable Keypoints) (Leutenegger et al., 2011).
7. MSER algorithm (Maximally Stable Extremal Regions) (Matas et al., 2002).
8. ORB algorithm (Oriented FAST and Rotated BRIEF) (Rublee et al., 2011).
9. HARRIS corner detection algorithm (Harris and Stephens, 1988).

10. Hausdorff Distance algorithm: When comparing two signs, distances between these two signs are calculated and minimum distance is selected (Huttenlocher et al., 1993).
11. Calculation of structural features using Hough transforms(Chunhavittayatera et al.,2006).
12. Hierarchial Centroid (H.C.) algorithm: Division of sign image into partitions and centroids of every partition are extracted as a feature (Armon, 2011).
13. HOG (Histogram of Oriented Gradients) algorithm (Dalal and Triggs, 2005).

In the following subsection one of the algorithms, namely Algorithm 1, will be briefly summarized. Short comments for other algorithms are also given later.

4.2.1 B.U. algorithm

In order to explain preprocessing phases in this algorithm, as an example cuneiform sign with HZL number 180 is selected. First, sign images that belong to V.S. and tablets are converted to gray level image by using rgb2grayfunction in MATLAB. After gray level conversion, sign images are converted to binary scale images, these images only contain black and white pixels. After binary conversion, skeleton of sign images are extracted by using skel function in MATLAB. This process shrink object boundaries but don’t let object connections break. Skeletonization of a sample cuneiform sign from V.S. digital list can be seen in Fig. 3.

Sign images that are cropped from V.S. and tablet are resized to 36x48 pixels and go into preprocessing phase as discussed before and then sign images are divided into 9 equal regions with each region having 12x16 pixels resolution. In this algorithm every cuneiform sign is divided into p regions and every region have mxn size (m=number of rows, n=number of columns). For example in Fig. 4 a sign image with 36x48 pixels resolution is divided into 9 regions. In this example m=12 and n=16 pixels.

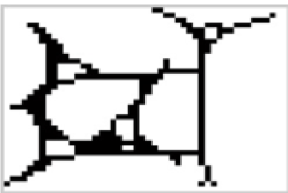


Figure 3 Skeletonization of cuneiform sign with HZL number 180.

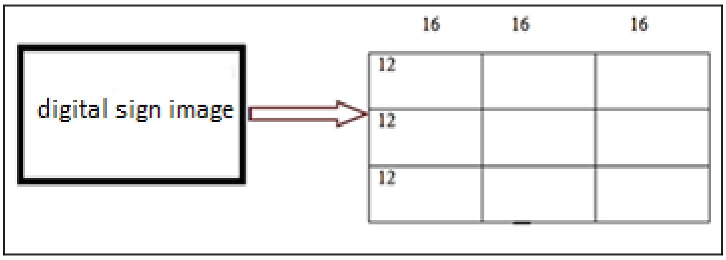


Figure 4 Division of a sign image into 9 equal regions

After sign is divided into equal regions, number of black pixels in every region is calculated. After this phase, difference of black pixel numbers in corresponding regions of signs that belong to V.S. and tablet are calculated. An error rate is calculated by dividing sum of these differences to total number of pixels (36x48) in sign. This error rate value is between 0 and 1. This process is summarized in (1).

$$Error = \frac{\sum |difference\ of\ black\ pixels\ of\ corrresponding\ regions|}{total\ number\ of\ pixels\ in\ a\ sign} \qquad (1)$$

Another error rate calculation is as follows: after division of sign into equal regions, number of black pixels in every region is calculated and difference of black pixel numbers in corresponding regions of signs that belong to V.S. and tablet are calculated. This difference value for every region is divided to total number of pixels (12x16) in a region. In last phase of error calculation, sum of difference ratios are divided to number of regions. This error rate value is between 0 and 1. This process can be seen in (2).

$$Error = \frac{\sum \frac{|difference\ of\ black\ pixels\ of\ corrresponding\ regions|}{total\ number\ of\ pixels\ in\ a\ region}}{number\ of\ regions} \qquad (2)$$

This algorithm can't be used when finding a match for signs in V.S. digital list because tablet and V.S. list versions of the same sign are extracted from different sources and theoretically they don't have the same black pixels. So this algorithm is used when searching and comparing signs in digital list. By using error rate, elimination of signs based on a threshold is carried out. B.U. algorithm narrows database space and reduce number of signs to search in V.S. so algorithm reduces runtime of matching algorithms.

4.2.2 Regionprops Library

This algorithm is used in data mining examples section (section 5) for extracting geometrical features of cuneiform signs.

4.2.3 SIFT Algorithm

SIFT is a popular algorithm and it is used in object recognition and computer vision systems. This algorithm finds keypoints in an image and extracts features in these keypoints by using descriptors. Most important aspect of SIFT is immunity to rotation, scale and light intensities when finding keypoints (Lowe, 2004). In this paper, SIFT is used as keypoint detector and descriptor.

4.2.4 SURF Algorithm

SURF algorithm is developed based on SIFT algorithm. SURF algorithm has differences when finding keypoints and extracting features from images. In SURF algorithm Hessian matrix structure is used for finding keypoints so it's more effective and faster than SIFT algorithm (Herbert et al., 2006). In this paper, SURF is used as keypoint detector and descriptor.

4.2.5 FAST Algorithm

FAST algorithm is a corner detection algorithm and it is developed for real time systems, because in real time systems algorithms like SIFT and HARRIS take too much processor time. Keypoints are made of corner points in FAST algorithm (Rosten and Drummond, 2006). In this paper, FAST is used as keypoint detector and ORB is used as descriptor.

4.2.6 BRISK Algorithm

BRISK algorithm takes less time than SURF and gives a better performance than SURF. It takes less time because it uses keypoint detection algorithm that FAST uses and algorithm uses bit arrays of neighborhood pixel intensity for every keypoint (Leutenegger et al., 2011). In this paper, BRISK is used as keypoint detector and descriptor.

4.2.7 MSER Algorithm

MSER algorithm is used for finding circle and ellipse (blobs) shapes. Algorithm chooses keypoints from these shapes and extracts features (Matas et al., 2002). In this paper, MSER is used as keypoint detector and ORB is used as descriptor.

4.2.8 ORB Algorithm

ORB algorithm is a hybrid of FAST and BRIEF algorithm. ORB finds keypoints by using FAST algorithm and extracts features from these keypoint by using BRIEF algorithm. Best aspect of ORB algorithm is that it isn't affected by rotation and noise. Also it works twice as fast as popular computer vision algorithm SIFT (Rublee et al., 2011). In this paper, ORB is used as keypoint detector and descriptor.

4.2.9 Harris Algorithm

Harris algorithm is one of the first algorithms that are used for finding corner and edge points. Algorithm is based on local auto correlation function; this function finds local changes on different angles in a digital signal. In Harris algorithm, keypoints are selected from corner and edge points (Harris and Stephens, 1988). In this paper, Harris is used as keypoint detector and ORB is used as descriptor.

4.2.10 Hausdorff distance Algorithm

Hausdorff distance algorithm is used to find a match of a cuneiform sign in cuneiform tablet by searching V.S. digital list. Algorithm principles are described below. When comparing cuneiform signs; distances between two signs are calculated. When finding a match, Hausdorff distance algorithm calculates distances between tablet cuneiform sign and every cuneiform sign in V.S. digital list and selects sign that has minimum distance to cuneiform sign. This algorithm is used frequently in many areas including object recognition, computer vision and image processing (Huttenlocher et al., 1993).

Before this algorithm is applied to cuneiform signs, signs go through several pre-processing phases. These phases are binary conversion of image, Canny edge detection algorithm and finally dilation. These phases can be seen in Fig. 5.

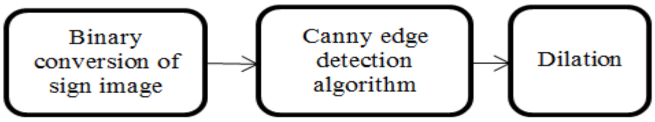

Binary conversion of sign image
Canny edge detection algorithm
Dilation

Figure 5 Preprocessing phases in Hausdorff distance algorithm

4.2.11 Calculation of structural features using Hough transform

This algorithm extracts structural features from cuneiform signs. These structural features are number of horizontal and vertical lines that cuneiform sign has. Main motivation for selecting this algorithm is that structures of Hittite cuneiform signs are usually made of horizontal and vertical lines. This algorithm can't be used directly when finding a match for signs in V.S. digital list because there can be more than one sign that has the same number of vertical and horizontal lines. So this algorithm is used when searching and comparing signs in digital list. By using number of vertical and horizontal lines, signs that don't satisfy a specific threshold (difference of lines) are eliminated. This algorithm narrows database space and reduce number of signs to search in V.S. so algorithm reduces runtime of matching algorithms.

Before this algorithm is applied to cuneiform signs, signs go through several preprocessing phases. These preprocessing phases are binary conversion of image, Skeletonization, filtering with Sobel filters, spur, clean and opening. These phases can be seen in Fig. 6.

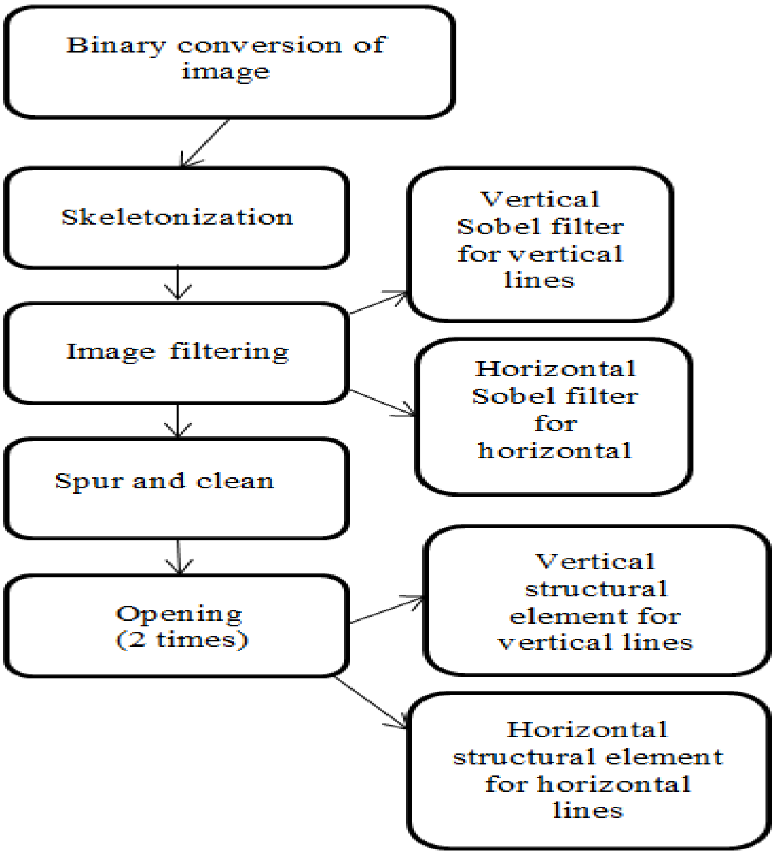

Binary conversion of image
Skeletonization
Image filtering
Spur and clean
Opening (2 times)
Vertical Sobel filter for vertical lines
Horizontal Sobel filter for horizontal
Vertical structural element for vertical lines
Horizontal structural element for horizontal lines

Figure 6 Preprocessing phases in Hough Transform algorithm

After signs went through preprocessing phases, their lines are detected by Hough transform and number of vertical and horizontal lines is determined (Chunhavittayatera et al., 2006).

4.2.12 H.C. (Hierarchical Centroid) Algorithm

H.C. algorithm is a process of division of sign image into partitions and centroids of every partition are extracted as feature (Armon, 2011). These extracted features are used to find equivalent of a tablet cuneiform sign in V.S. digital list. Feature vector that belongs to tablet cuneiform sign are compared with feature vectors of all V.S. digital list signs. This comparison is achieved by two metrics: Cosine similarity and Euclidean distance between two vectors. Signs that similar to each other have large Cosine similarity. However in Euclidean distance it is the opposite, signs that similar to each other have small Euclidean distance. When comparing tablet sign to all V.S. digital list signs, minimum Euclidean distance and maximum Cosine similarity between signs are selected and sign that have minimum distance and maximum similarity is accepted as a match. Before this algorithm is applied to cuneiform signs, preprocessing phases given in Fig. 5 are applied to cuneiform signs.

The function that algorithm uses has image as input and x coordinate of centroid as output (white pixels are considered as weight and black pixels are considered as space). Image is divided into two by x coordinate of image centroid and function is called recursively for transpose of these two pieces (Armon, 2011). For y coordinates of centroid, transposition of image goes through function as input. Number of partitions and size of feature vector depends on depth value. In H.C. algorithm depth value is 6 and feature vector has 126 elements.In Fig. 7, algorithm is applied with different depth values (d) to sign with HZL number 180 (V.S. digital list). Lines that can be seen in Fig.7 represent lines that cross from centroid and segmentation location.

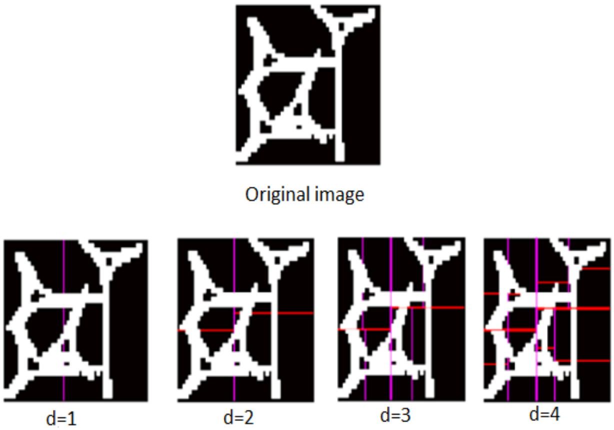


Figure 7 Segmentation of sign with HZL no 180 for different depth values

4.2.13 HOG algorithm

HOG algorithm is an algorithm that extracts HOG features from Hittite cuneiform signs (Dalal and Triggs, 2005). These extracted features are used to find equivalent of a tablet cuneiform sign in V.S. digital list. Feature vector that belongs to tablet cuneiform sign are compared with feature vectors of all V.S. digital list signs. Comparison metrics are the same as those given for Algorithm 12 in Section 4.2.12 of the paper. In HOG algorithm 9 orientation bins and 27 cells were used in Hittite cuneiform sign recognition. Feature vector had 243 elements. Before this algorithm is applied to cuneiform signs, preprocessing phases given in Fig. 5 were applied to cuneiform signs.

5. Data Mining Examples On Hittite Cuneiform Signs

In Hittite cuneiform script there are many geometrically similar signs. Thinking of gathering these similar signs in different categories has created data mining side of this study. In this study, geometric features were extracted and categorization of geometrically similar signs was carried out by K-means clustering algorithm which is a popular data mining algorithm. After categorization, popular data mining classification algorithms were applied to cuneiform signs and classification performances were reported in the following subsections.

5.1. Hittite Cuneiform Signs Dataset

In data mining examples, dataset consists of geometric features of Hittite cuneiform signs. These cuneiform signs were selected from V.S. digital list. Digital image acquisition phase of cuneiform signs is the same as Subsection 4.1 of the paper. Geometric features were extracted by Algorithm 2 of MATLAB Regionprops library. These geometric features are Area, X coordinate of centroid, Y coordinate of centroid, Euler Number, Extent, Eccentricity and EquivDiameter. Geometric features were extracted for every cuneiform signs that were used in data mining algorithms. Finally a dataset with 7 features is constructed.

5.2. Data Mining Algorithms That Were Used in Hittite Cuneiform Signs

5.2.1 K-means clustering algorithm

K-means clustering algorithm is an algorithm of data mining that has descriptive model structure. It is used for assigning class labels to data that class labels are unknown. K-means is one of the most popular data mining clustering algorithms because it can be easily implemented and doesn’t take too much processor time (Ahamed and Hareesha, 2012). Main purpose of K-means is to divide unlabeled data to K class by using features of data. Algorithm places data to a feature space and make clustering on this feature space (Han and Kamber, 2006).

5.2.2. J48 decision tree classification algorithm

J48 decision tree algorithm is the Weka implementation of Quinlan's C 4.5 (Quinlan, 1993) decision tree algorithm (Sharma and Sahni, 2011). J48 uses training data to build a decision tree model by using information entropy (Han and Kamber, 2006) and create decision mechanisms by dividing data into little partitions (Thangalakshmi and Kamalesh, 2014).

5.2.3. k-Nearest Neighbour (kNN) classification algorithm

K-nearestneighbour (kNN) algorithm proposed by Cover and Hart (Cover and Hart, 1967). Algorithm is used in many areas; reasons behind this popularity are fast classification model building and good classification results on noisy data (Bhatia, 2010). Algorithm works with principle of "Classify according to nearest neighbours" (Suguna and Thanushkodi, 2010).

5.2.4. Artificial Neural Network (ANN) classification algorithm

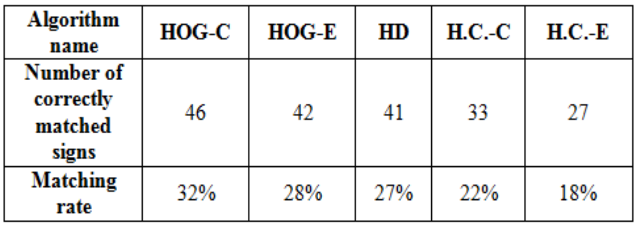
Artificial neural network is used in many areas including finance, engineering, geology and physics (Pradhan and Lee, 2007) and (Celik and Karatepe, 2007). ANN structure is based on human brain's most important aspects which are learning, interpretation of information and inference. ANN is developed to perform these processes automatically. ANN's mathematical model of decision and learning process are inspired by human brain.

6. RESULTS AND DISCUSSIONS

InHittite cuneiform script there are signs that have Hittite, Sumerian and Akkadians meanings. In this paper the results of a study only on 149 cuneiform signs that have only Hittite meanings were selected. Digital images of signs are acquired from V.S. digital list and from some cuneiform tablets. Digital image acquisition process is already described in subsection 4.1 of the paper.

In order to find the matching tablet cuneiform signs by searching V.S. digital list, three algorithms were used; HOG, Hausdorff Distance (HD) and Hierarchial Centroid (H.C.) algorithms. Matching performances of algorithms can be seen in Table 1.

Table 1Recognition performances of algorithms



Algorithm name	HOG-C	HOG-E	HD	H.C.-C	H.C.-E
Number of correctly matched signs	46	42	41	33	27
Matching rate	32%	28%	27%	22%	18%

In the tables Table 1, Table 2 and Table 3 HOG-C stand for HOG-Cosine, HOG-E for HOG-Euclidian, H.C.-C for H.C.-Cosine and H.C.-E for H.C.-Euclidian.

As can be seen in Table 1 most successful algorithm for finding matches of cuneiform signs is HOG (cosine similarity measure) algorithm. Another result that can be inferred from Table 1 is that when finding matches of cuneiform signs Cosine similarity comparison measure performs better than Euclidean distance comparison measure.

In Hittite cuneiform script there are many similar signs. This situation leads to difficulty when finding matches of cuneiform signs in tablets and as a result it is affecting matching performance of algorithms. Sample signs that cause this situation can be seen in Fig. 8.

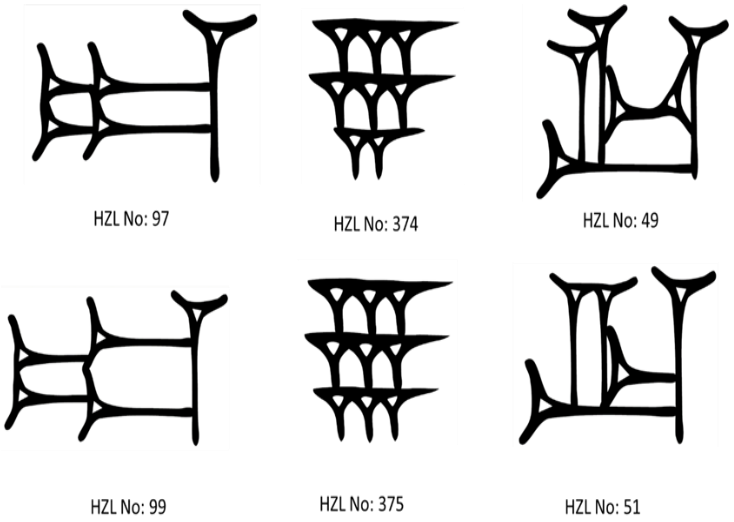


Figure 8 Sample similar signs

Because of similar sign situation, matching of a sign that belongs to tablet doesn't come up in the first place when comparing with other signs in V.S. digital list. So algorithms can't find a correct match in the first query, instead correct match of a sign appears in the second, third or fourth place in query. Sample query of discussed situation can be seen in Fig. 9 query is executed using HOG (Cosine similarity) algorithm and V.S. match of tablet sign with HZL number 49 is expected to be same but similar sign (HZL no. 51) appeared first in query so algorithm performed a wrong match. V.S. equivalent of tablet sign appear as third.

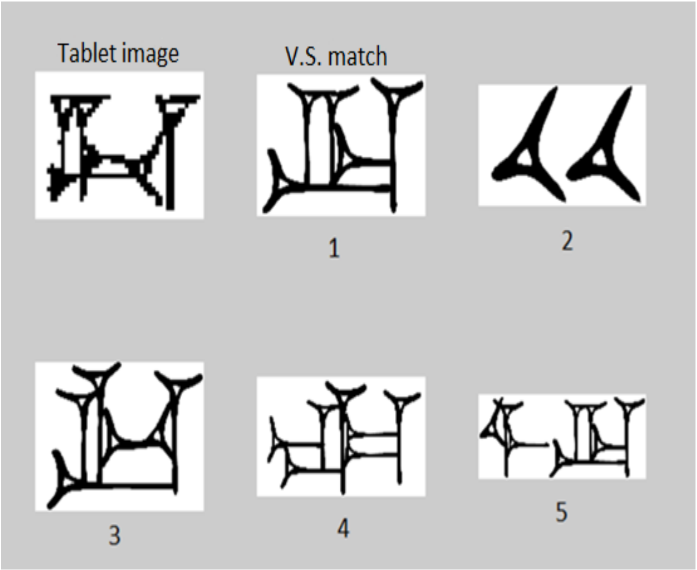
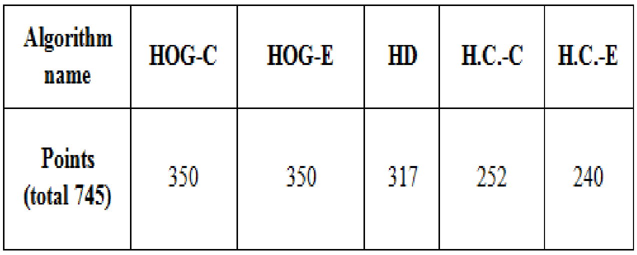


Figure 9 Query sample of a sign with HZL number 49

Because of situations like Fig. 9, sign retrieval performances of algorithms must be inspected. In order to measure sign retrieval performances of algorithms a scoring system is used. In this system, an algorithm gets five points if V.S. digital list match of a cuneiform sign is found in first place, four points if it is found in second place, three points for third place, two points for fourth place and one point for fifth place. If an algorithm retrieves all of 149 signs in first place algorithm gets 745 points. Points are calculated for three algorithms using 149 cuneiform signs. Results are shown in Table 2.

Table 2 Retrieval points of algorithms

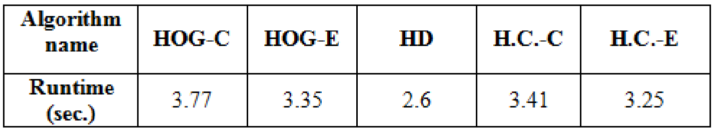


Algorithm name	HOG-C	HOG-E	HD	H.C.-C	H.C.-E
Points (total 745)	350	350	317	252	240

Although matching performance of HOG (Euclidean distance) algorithm is less than HOG (Cosine similarity) in Table 1, retrieval performances of both algorithms are equal as seen in Table 2. If we exclude situations that match of a sign is retrieved in first place HOG (Cosine similarity) gets 350-5*46=120 points and HOG (Euclidean distance) gets 350-5*42=140 points. Thus HOG (Euclidean Distance) retrieves matches of most cuneiform signs in V.S. digital list in first five places and catch up to HOG (Cosine similarity) algorithm in retrieval phase.

As number of cuneiform signs in a tablet grows bigger, runtime of algorithms is getting more important. In a Hittite cuneiform tablet there can be 400 signs depending on size of the tablet. Processing time of these signs can take a long time. Runtime of algorithms for finding a match of a sign in V.S. digital list can be seen in Table 3.

Table 3 Runtime of algorithms in seconds (sec.)



Algorithm name	HOG-C	HOG-E	HD	H.C.-C	H.C.-E
Runtime (sec.)	3.77	3.35	2.6	3.41	3.25

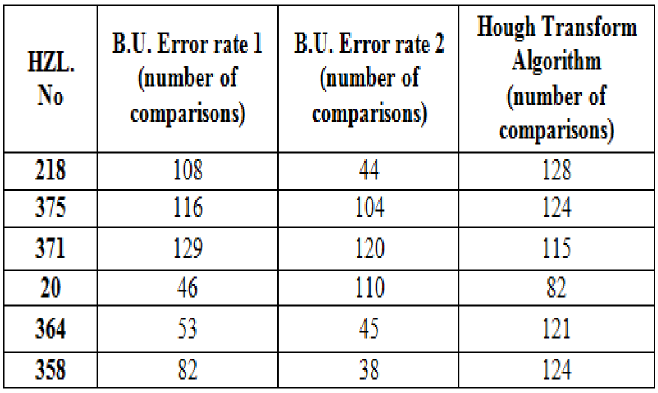
According to results in Table 3 the best matching algorithm HOG (Cosine) takes most of processor time. Algorithm finds match of a sign in 3.77 seconds. For example if this algorithm is applied to a tablet with 400 signs, runtime will be 1508 seconds which is approximately 25 minutes. In order to reduce runtime of reading a tablet, two algorithms are proposed. These algorithms are Algorithm 1 (B.U. algorithm) and Algorithm 11 (Calculation of structural features using Hough transform). These algorithms run before matching algorithms in order to reduce the number of comparisons in algorithms when performing match for every sign. Reducing number of comparisons causes runtime reduce and algorithm speed up.

B.U. algorithm uses error rate for reducing number of comparisons. If Error rate between two cuneiform signs obtained from tablet and V.S. list is greater than a specific threshold they aren’t included in comparison process for matching algorithms. So B.U. algorithm narrows database space and reduce number of signs to search in V.S.

In Hough transform algorithm, number of horizontal and vertical lines is determined for tablet cuneiform sign and ratio of number of horizontal lines to number of vertical lines is calculated. The same process is applied to V.S. digital list sign. If difference of these ratios is greater than a specified threshold, V.S. digital list sign isn’t included to comparison process for matching algorithms. If a sign doesn’t have vertical or horizontal lines (for example cuneiform sign with HZL number 1 only have one horizontal line) difference is calculated for number of available lines.

In this paper error rate threshold for B.U. algorithm is selected as 0.1 and algorithm difference threshold is selected as 1 for Hough transform algorithm. These two algorithms are applied to sample cuneiform signs and number of total comparisons that matching algorithms will do after elimination process are given in Table 4.

Table 4Application of elimination algorithms to sample signs



HZL. No	B.U. Error rate 1 (number of comparisons)	B.U. Error rate 2 (number of comparisons)	Hough Transform Algorithm (number of comparisons)
218	108	44	128
375	116	104	124
371	129	120	115
20	46	110	82
364	53	45	121
358	82	38	124

According to results in Table 4, B.U. algorithm has good performance for cuneiform signs that have simple shape (Signs with HZL number 20, 358 and 364). Reason behind this performance is that in Hittite cuneiform, complex and basic shape signs have so much difference in pixel numbers. As seen in Table 4 error rate 2 performs better than error rate 1 in general. Hough algorithm shows poor performance because of low resolution of tablet cuneiform sign images and curvature of these sign lines.

In this study matching performances of OpenCV algorithms with 3, 4, 5, 6, 7, 8, 9 numbers are measured. Sign images are resized to 512x512 pixels. Matching performances of OpenCV algorithms are calculated based on an error rate between two compared signs (tablet and V.S. digital sign). This error rate is based on matching rate of extracted descriptors from keypoints in an image. Algorithms are evaluated on this error rate. (3) shows error rate calculation for two compared signs.

$$Error\ rate = \frac{|Difference\ of\ matched\ descriptor\ of\ signs|}{Number\ of\ total\ descriptors\ in\ first\ sign} \qquad (3)$$

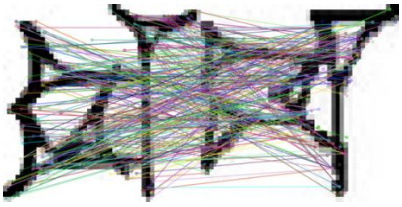
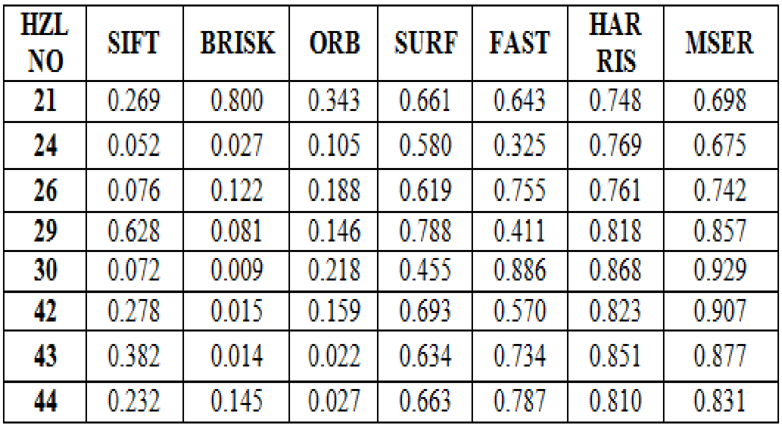


Figure 10 Application of SIFT algorithms on sign with 180 HZL

When calculating error rate 323 descriptors are extracted from first image and 339 are extracted from second image. 323 of these descriptors are matched. So in this case error rate is as follows: Error rate= (339 – 323) / 323 = 0.0495 ~ 0.05

As can be seen in example error rate value is between 0 and 1. Error rates are calculated for 149 cuneiform signs that are extracted from tablets and V.S. digital list. Lowest error rate accepted as highest matching performance for an algorithm. Error rates of algorithms for selected sample cuneiform signs can be seen in Table 5. Also average error rate of algorithms for 149 cuneiform signs can be seen Table 5.

Table 5 Error Rate of Algorithms



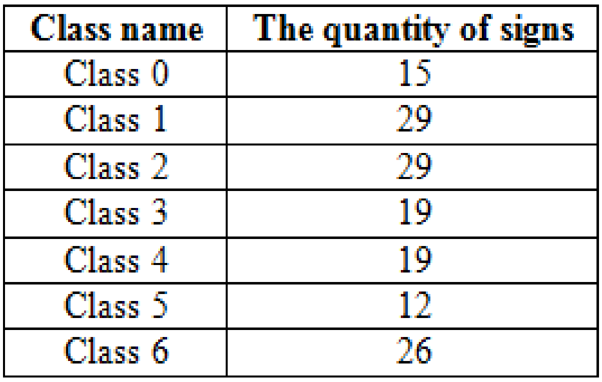
HZL NO	SIFT	BRISK	ORB	SURF	FAST	HAR RIS	MSER
21	0.269	0.800	0.343	0.661	0.643	0.748	0.698
24	0.052	0.027	0.105	0.580	0.325	0.769	0.675
26	0.076	0.122	0.188	0.619	0.755	0.761	0.742
29	0.628	0.081	0.146	0.788	0.411	0.818	0.857
30	0.072	0.009	0.218	0.455	0.886	0.868	0.929
42	0.278	0.015	0.159	0.693	0.570	0.823	0.907
43	0.382	0.014	0.022	0.634	0.734	0.851	0.877
44	0.232	0.145	0.027	0.663	0.787	0.810	0.831

As can be seen in Table 5, algorithm with lowest error rate is ORB algorithm with 0.17 error rate. Second lowest rate algorithm is SIFT algorithm with 0.24 error rate. The reason behind these low error rates is that ORB and SIFT have resistance to noise and light intensities. Algorithm with highest error rate is MSER algorithm with 0.87 error rate. Main reason for high error rate is that MSER focuses on round (blob) and elliptic shapes and Hittite cuneiform signs don't have these kinds of shapes.

In data mining part, Clustering and classification algorithms are applied to 149 Hittite cuneiform signs. First of all features that are discussed in subsection 5.1 of paper are extracted from 149 cuneiform signs. A dataset is constructed using these features.

After dataset creation, K-means clustering algorithm is applied to dataset. Purpose of applying clustering algorithm to dataset without class information is to categorize cuneiform signs by gathering similar shape signs together. When applying K-means clustering algorithm to dataset, several K values are experimentally selected. Situations of gathering visually and geometrically similar cuneiform signs are assessed when selecting K value. After these assessments 7 is considered appropriate for K value. Class information is added to cuneiform signs in dataset as a result of K-means clustering algorithm. Class distributions of 149 cuneiform signs can be seen in Table 6. Fig. 11 shows sample cuneiform signs that belong to classes.

Table 6 Class Distributions of Hittite Cuneiform Signs



Class name	The quantity of signs
Class 0	15
Class 1	29
Class 2	29
Class 3	19
Class 4	19
Class 5	12
Class 6	26

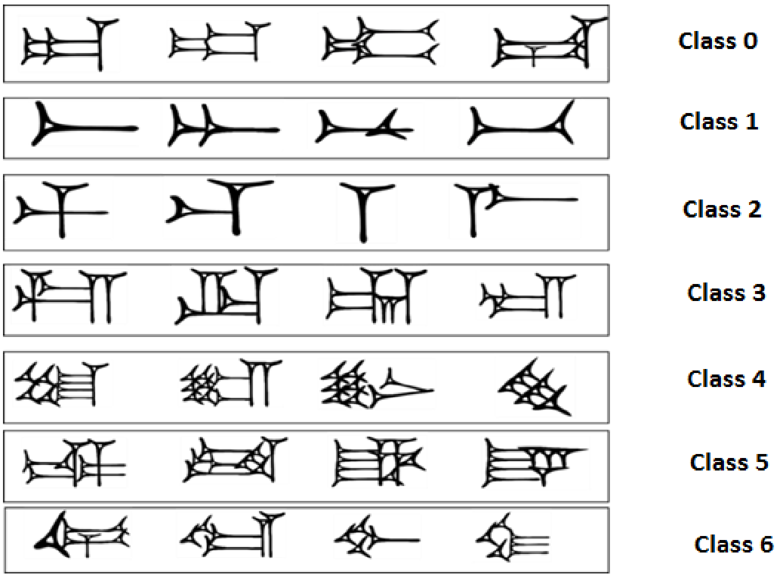
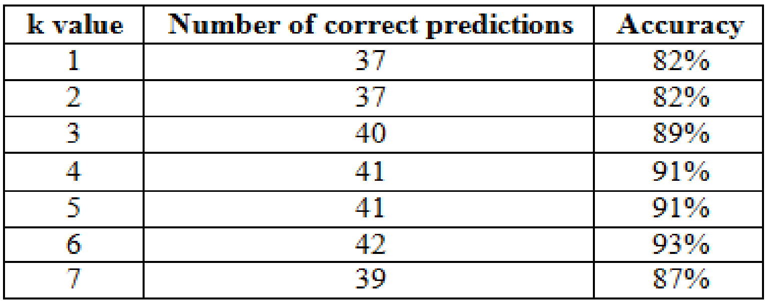


Figure 11 Sample cuneiform signs of every class

After clustering algorithm is applied to cuneiform signs, classification algorithms are applied and accuracies of classification algorithms are measured. In order to apply classification algorithms to dataset, dataset is split in two as training and test set. Split ratio is %70 as training set and %30 as test set. Training set has 104 elements and test set has 45.

First algorithm that is applied to dataset is k-NN classification algorithm. Classification model is based on k value selection. k values 1, 2, 3, 4, 5, 6, 7 are selected and classification models are created based on these values. Accuracy of models is calculated using test set. Table 7 shows accuracies and correct predictions of different k value models.

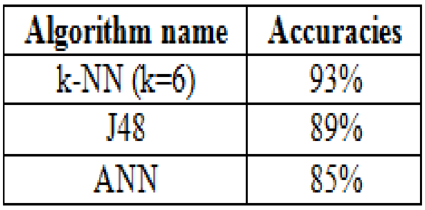
Table 7 k-NN Accuracy for Different k Values



k value	Number of correct predictions	Accuracy
1	37	82%
2	37	82%
3	40	89%
4	41	91%
5	41	91%
6	42	93%
7	39	87%

Other algorithms that are applied to dataset were J48 decision tree classification algorithm and artificial neural network. Accuracies of classification algorithms can be seen in Table 8.

Table 8 Accuracies of Classification Algorithms



Algorithm name	Accuracies
k-NN (k=6)	93%
J48	89%
ANN	85%

7. Conclusions

In the study by Aşuroğlu (Asuroglu, 2015) that is summarized in this paper, cuneiform signs from tablets are recognized and V.S. digital list matches are found. HOG algorithm is reported as best algorithm when finding a match of tablet cuneiform sign and cuneiform sign retrieval process. Also two algorithms are proposed to reduce runtime of matching algorithms. Matching performances of algorithms that belong to OpenCV library are reported as error rate and algorithm with minimum error rate is reported as ORB algorithm.

Categorization of geometrically similar cuneiform signs in V.S. digital list is achieved by using K-means clustering algorithm and class labels are added to cuneiform signs. Finally classification algorithms are applied to cuneiform sign dataset and classification performances are measured by algorithm accuracies. k-NN algorithm has the highest accuracy.

In order to get better results on computerized Hittite cuneiform sign recognition and perform better on automatic knowledge extraction by data mining techniques, available amount of data must be increased (Gaines, 2013). Availability of more digital versions of a sign (versions of same sign that are written by different persons or in different environments) and store these data in an online database will be beneficial for performance of this study. Also availability of sources besides V.S. digital list that include high resolution images of cuneiform signs will increase recognition rate of tablet cuneiform signs. Another situation is resolution of cuneiform signs on tablets. Cuneiform sign images from Portal Mainz have low resolution so this situation caused difficulties in this study. Because of low resolution, details of signs were not fully used. To prevent this situation, creation of a database with high resolution tablet images will increase recognition rate of cuneiform signs and eventually increase availability of details in cuneiform signs. Furthermore extra studies on threshold values that are used in elimination algorithms will increase algorithm performances.

In this study, categorization of cuneiform signs was achieved by using data mining techniques. When finding a match of table cuneiform sign, class labels can be used as foreknowledge to reduce number of comparisons in database. Because of class information of a sign is known, database searches can be made only in this class. Furthermore classes of new signs can be predicted by classification techniques and as a result recognition process can be fast and effective.

Acknowledgment

The authors would like to express their gratitude to Prof. Dr. I. C. Karasu and Prof. Dr. Y. Arikan for their kind support and encouragement during the project “Hittite Cuneiform Script and Further: Hittite Information System project (HIT-IS Project)”. Thanks are also extended to Asst. Prof. Dr. E. Sumer, Asst. Prof. Dr. M. Sert, Miss M.B. Yesiltepe, Miss S. Sahin, Mr. S. Isik and Mrs. N. Ugur for their contributions and support in the project.

References

- Ahamed, S. and Hareesha, K. (2012), “Dynamic Clustering of Data with Modified K-Means Algorithm”, *International Conference on Information and Computer Networks (ICICN)*, pp. 221-225.
- Ahmed, K. K. (2012), “Online Sumerians Cuneiform Detection Based on Symbol Structural Vector Algorithm”, *Journal Of The College Of Education For Women*, vol.23, no.2, pp. 545-553.

- Aktas, A. Z. ve Gürsel, H. (1988), "Çiviyazısı Metinlerin Uzman Sistemler Yardımıyla Çözülmesi", *5. Türkiye Bilgisayar Kongresi,* Ankara, (In Turkish).
- Anthony L., Yang J., Koedinger K. R.(November 2012) , "A paradigm for handwriting-based intelligent tutors*", International Journal of Human-Computer Studies* Volume 70, Issue 11, pp. 866–887.
- Armon, S. (2011), "Handwriting Recognition and Fast Retrieval for Hebrew Historical Manuscripts", M.S. thesis, Hebrew University, Jerusalem, Israel.
- Asuroglu, T. (2015), "Hitit Çiviyazisi İşaretlerinin Bilgisayar Desteği İle Okunmasi Ve Veri Madenciliği Uygulama Örnekleri", Yüksek Lisans Tezi, Başkent Üniversitesi, supervised by Prof. Dr. A. Z. Aktaş (In Turkish).
- Bhatia, N. (2010) "Survey of Nearest Neighbor Techniques", *International Journal of Computer Science and Information Security,* vol.8, no.2, pp. 302-305.
- Celik, A. and Karatepe, Y. (2007), "Evaluating and forecasting banking crises through neural network models: An application for Turkish banking sector", *Expert Systems with Applications,* vol.33, pp. 809-815.
- Chunhavittayatera, S., Chitsobhuk, O. and Tongprasert, K. (2006), "Image Registration using Hough Transform and Phase Correlation", *International Conference on Advanced Communication Technology (ICACT),* pp. 973-977.
- Cover, T. and Hart, J. (1967), "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory,* vol.3, no.1, pp. 21-27.
- Dalal, N. and Triggs, B. (2005), "Histograms of oriented gradients for human detection", *IEEE Society Conference on ComputerVision and Pattern Recognition (CVPR),* pp. 886-893.
- Dik, E. C. (2014), "Hitit Çiviyazısı İşaretlerinin Otomatik Çevirisi", Yüksek Lisans Tezi, Başkent Üniversitesi, supervised by Prof. Dr. A. Z. Aktaş (In Turkish).
- Edan, N. M. (2013), "Cuneiform Symbols Recognition Based on K-Means and Neural Network", *Journal of Computational Mathematics,* vol.10, no.1, pp. 195-202.
- Gaines B. R. (February 2013), "Knowledge acquisition: Past, present and future", *International Journal of Human-Computer Studies* Volume 71, Issue 2, pp. 135–156.
- Gürsel, H. (1988), "An Expert System for Cuneiform Interpretation", M.S. thesis, Comp. Eng. Dept. METU, Ankara.
- Han, J. and Kamber, M. (2006), "Data Mining Concept and Techniques", 2nd Edition, Elsevier Inc.
- Harris, C. and Stephens, M. (1988), "A combined corner and edge detector", *Alvey Vision Conference,* pp. 147-151.
- Herbert, B., Tuytelaars, T. and Van Gool, L. (2006), "SURF: Speeded Up Robust Features", *Lecture Notes in Computer Science Volume 3951,* pp. 404-417.
- Huttenlocher, D. P., Klanderman, G. A. and Rucklidge, W. J. (1993), "Comparing images using the Hausdorff distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.15, no.9, pp. 850-863.
- J. E. Rosten and T. Drummond, (2006) "Machine learning for high-speed corner detection", *8th European conference on Computer Vision (ECCV),* pp. 430-443.
- Karasu, C. (2013), "Hititçe ve Hitit Çivi Yazısı", Editörler: M. D. Alparslan ve M. Alparslan, Hititler, Yapı Kredi Yayınları,, ss.84 – 93 (In Turkish).
- Leutenegger, S., Chli, M. and Siegwart, R. Y. (2011), "BRISK: Binary robust invariant scalable keypoints", IEEE International Conference on Computer Vision, pp. 2548-2555.

- Li F. and Woo P. (August 2000), "The coding principle and method for automatic recognition of JiaGu Wen characters", *International Journal of Human-Computer Studies* Volume 53, Issue 2, pp. 289-299.
- Logar A. M., Edward M. C., Oldham W. (March 1994)," Performance comparisons of classification techniques for multi-font character recognition", *International Journal of Human-Computer Studies* Volume 40, Issue 3, pp. 403-423.
- Lowe, D. G. (2004), "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110.
- Matas, J., Chum, O., Urban, M. and Pajdla, T. (2002), "Robust wide-baseline stereo from maximally stable extremal regions", *British Machine Vision Conference,* pp. 384-393.
- Pornpanomchai C., Batanov D. N., Dimmitt N. (September 2001), "Recognizing Thai handwritten characters and words for human–computer interaction", *International Journal of Human-Computer Studies* Volume 55, Issue 3, pp. 259-279.
- Pradhan, B. and Lee, S. (2007), "Utilization of Optical Remote Sensing Data and GIS Tools for Regional Landslide Hazard Analysis Using an Artificial Neural Network Model", *Earth Science Frontiers*, vol.14, no.6, pp. 143-152.
- Quinlan, J. R. (1993), "C 4.5: programs for machine learning", Morgan Kaufmann Publishers Inc.
- Rublee, R., Rabaud, V., Konolige, K. and Bradski, G. (2011), "ORB: an efficient alternative to SIFT or SURF", *IEEE International Conference in Computer Vision (ICCV),* pp. 2564-2571.
- Ruster, C. and Neu, E. (1989), "Hethitisches Zeichenlexikon: Inventar und Interpretation Der Keilschriftzeichen aus den Bogazkoy-Texten", O. Harrassowitz.
- Sharma, A. K. and Sahni, S. (2011), "A Comparative Study of Classification Algorithms for Spam Email Data Analysis", *International Journal on Computer Science and Engineering (IJCSE)*, vol.3, no.5, pp.1890-1895.
- Suguna, N. and Thanushkodi, K. (2010), "An Improved k-Nearest Neighbor Classification Using Genetic Algorithm", *International Journal of Computer Science Issues*, vol.7, Issue 4, no.2, pp. 18-21.
- Sundar, K.A. and John, M. A. (2013), "High precision printed character recognition method for Tamil script", *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1-5.
- Thangalakshmi, P. and Kamalesh, S. (2014), "A Decentralized Service Discovery Approach on Peer-To-Peer Networks", *International Conference on Innovations in Engineering and Technology (ICIET),* pp.2423-2429.
- Tyndall, S. (2012), "Toward automatically assembling Hittite-language cuneiform tablet fragments into larger texts", *50th Annual Meeting of the Association for Computational Linguistics*, vol.2, pp. 243-247.
- Yesiltepe, B. (2015), "Hitit Çiviyazılı Metinlerin Okunmasında Uzman Sistem Uygulama Örnekleri", Yüksek Lisans Tezi, Başkent Üniversitesi, supervised by Prof. Dr. A. Z. Aktaş (In Turkish).
- Yousif, H., Rahma, A. M. and Alani, H. (2006), "Cuneiform Symbols Recognition Using Intensity Curves", *The International Arab Journal of Information Technology*, vol.3, no.3, pp. 237-241.

A. Ziya Aktas had his BS and MS at METU in Ankara. He received his Ph.D. at Lehigh University, USA. He served as the first chairman of the Department of Computer Engineering at METU for a total of ten years. He is the author of a software engineering book published by Prentice Hall in the USA. He is a member of ACM. He serves as a Professor of Computer Engineering at Baskent University His recent interest areas are Software engineering, Cloud computing, IS modeling, Data Mining, Knowledge management and engineering.

Tunc Asuroglu received the bachelor’s degree from TOBB ETU in Ankara, at 2013 and the Master of Science degreefrom Baskent University in Ankara at 2015 in Computer Engineering. He is currently a Graduate student leading to Ph. D. and Research Assistant in Baskent University, Ankara, Turkey.