

**BAŐKENT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**HİTİT ÇİVİYAZISI İŐARETLERİNİN BİLGİSAYAR DESTEĐİ  
İLE OKUNMASI VE VERİ MADENCİLİĐİ UYGULAMA  
ÖRNEKLERİ**

**TUNÇ AŐUROĐLU**

**YÜKSEK LİSANS TEZİ**

**2015**



**HİTİT ÇİVİYAZISI İŞARETLERİNİN BİLGİSAYAR DESTEĞİ  
İLE OKUNMASI VE VERİ MADENCİLİĞİ UYGULAMA  
ÖRNEKLERİ**

**COMPUTERIZED HITTITE CUNEIFORM SIGN  
RECOGNITION AND DATA MINING APPLICATION  
EXAMPLES**

**TUNÇ AŞUROĞLU**

Başkent Üniversitesi  
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin  
BİLGİSAYAR Mühendisliği Anabilim Dalı İçin Öngördüğü  
YÜKSEK LİSANS TEZİ  
olarak hazırlanmıştır.

2015

“Hitit iviyazısı İřaretlerinin Bilgisayar Desteęi ile Okunması ve Veri Madencilięi Uygulama rnekleri” bařlıklı bu alıřma, jrimiz tarafından, 15/07/2015 tarihinde, **BİLGİSAYAR MHENDİSLİęİ ANABİLİM DALI** 'nda **YKSEK LİSANS TEZİ** olarak kabul edilmiřtir.

Bařkan : Prof. Dr. İ. Cem KARASU

ye (Danıřman) : Prof. Dr. A. Ziya AKTAř

ye : Yrd. Doę. Dr. Emre SMER

**ONAY**

...../...../.....

Prof. Dr. Emin AKATA  
Fen Bilimleri Enstits Mdr

## TEŐEKKÜR

Yazar, bu alıőmanın gerekleőmesinde katkılarından dolayı, aőađıda adı geen kiőilere itenlikle teőekkür eder.

Tez danıőmanı Sayın Prof. Dr. A. Ziya AKTAŐ'a, alıőmanın sonuca ulaőtırılmasında ve karőtılaőtılan glklerin aőtılmasında her zaman yardımcı ve yol gsterici olduđu iin...

Sayın Prof. Dr. İ. Cem KARASU ve Prof. Dr. Yasemin ARIKAN'a Hititler ve Hitite gibi eski ama yazar iin yepyeni bir alana girerken elinden tutup yolunu aydınlattıkları iin...

Sayın Yrd. Do. Dr. Emre SÜMER'e alıőmalardaki katkıları iin...

Aileme, destek oldukları iin...

Sayın Emel BOYACI'ya her zaman yanımda olduđu iin...

Sayın Araő. Gör. Koray AICI'ya yardımlarından dolayı...

## ÖZ

# HİTİT ÇİVİYAZISI İŞARETLERİNİN BİLGİSAYAR DESTEĞİ İLE OKUNMASI VE VERİ MADENCİLİĞİ UYGULAMA ÖRNEKLERİ

Tunç AŞUROĞLU

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Anadolu'da M.Ö. 1650 - 1200 yılları boyunca hüküm süren Hitit krallığı ve imparatorluğu o dönem dünyasının en büyük güçlerinden birisi sayılmaktadır. Hititlerin kullandığı Hititçe, Hint-Avrupa dil ailesinin bilinen en eski üyelerinden biridir. Hititler dünyada arşiv-kütüphane uygulamasını ortaya koyan ilk toplumlardan biridir. Hititler çeşitli konulardaki metinleri Hitit çiviyazısı ile yaş kil tabletler üstüne yazıp tabletleri çoğunlukla fırınlayarak kalıcı hale getirmişlerdir.

Hitit çiviyazılı metinlerin okunması, çevrilmesi, yorumlanması ve gramer kurallarının kullanımı yaklaşık yüz yıldır "ele ve insana" dayalı olarak yapılan, uzun süre ve emek isteyen, yorucu bir uğraştır. Anadolu'da yerüstünde ve hâlâ yeraltında bulunan kil tabletler üstündeki çiviyazısı işaretlerini günümüz bilgi ve bilgisayar destekli tekniklerle okuyabilmek sadece Anadolu değil tüm insanlık tarihi ve kültürü açısından son derece önemlidir. Bu çalışmada, imge işleme yöntemleri ile Hitit çiviyazılı tabletlerde bulunan çiviyazısı işaretlerinin okunması gerçekleştirilmiştir. Ayrıca çalışmada veri madenciliği teknikleri kullanılarak çiviyazılı işaretlerin sahip olduğu geometrik özelliklere göre sınıflara ayrılması ile ilgili uygulama örneklerine de yer verilmiştir.

**ANAHTAR SÖZCÜKLER:** Çiviyazısı işareti okuma, Hitit çiviyazısı, İmge işleme, Karakter Tanıma, Veri Madenciliği.

**Danışman:** Prof. Dr. A. Ziya AKTAŞ, Başkent Üniversitesi, Bilgisayar Mühendisliği Bölümü.

## **ABSTRACT**

### **COMPUTERIZED HITTITE CUNEIFORM SIGN RECOGNITION AND DATA MINING APPLICATION EXAMPLES**

Tunç AŞUROĞLU

Başkent University Institute of Science and Engineering

Department of Computer Engineering

In Anatolia the kingdom and empire of the Hittites had ruled nearly half a millenium during the years BC 1650-1200. It was considered one of the greatest world power of that time. Hittite language that the Hittites used is one of the oldest member of the Indo-European language family. The Hittites were one of the first communities that had adapted the concept of archive-library. The Hittites used cuneiform signs to write on various topics on wet clay tablets and baked them to be permanent and durable.

The study of Hittite language grammar rules followed transliteration, transcription and translation phases manually on the Hittite cuneiform tablets. It takes a long time, it requires financial support and a special know-how and expertise for processing. It is a tedious job. Many more tablets are still waiting under and over ground to be read and translated. Being able to read the signs on cuneiform clay tablets still in Anatolia, using computer-aided techniques would be a significant contribution not only to Anatolian but also to human history. In this study, recognition of Hittite cuneiform signs is performed by using image processing techniques. Also in this study; using data mining, applications related to classification of Hittite cuneiform signs based on their geometrical features are performed.

**KEYWORDS:** Cuneiform sign recognition, Data Mining, Hittite cuneiform script, Image processing, Character recognition.

**Advisor:** Prof. Dr. A. Ziya AKTAŞ, Başkent University, Department of Computer Engineering

## İÇİNDEKİLER LİSTESİ

	<u>Sayfa</u>
ÖZ.....	i
ABSTRACT .....	ii
İÇİNDEKİLER LİSTESİ.....	iii
ŞEKİLLER LİSTESİ.....	v
ÇİZELGELER LİSTESİ.....	vii
SİMGELER VE KISALTMALAR LİSTESİ.....	viii
<b>1 GİRİŞ.....</b>	<b>1</b>
1.1 Problemin Tanımı .....	1
1.2 Literatür İncelemesi .....	2
1.3 Tezin Amacı .....	5
1.4 Tezin Yapısı .....	6
<b>2 HİTİTLER VE HİTİT ÇİVİYAZISI.....</b>	<b>8</b>
2.1 Hititler ve Hitit İmparatorluğu .....	8
2.2 Hitit Çiviyazısı .....	13
<b>3 HİTİT ÇİVİYAZISI İŞARETLERİNİN OKUNMASI.....</b>	<b>17</b>
3.1 Hitit Çiviyazısı İşaretlerinin Dijital İmajının Alınması.....	17
3.2 Hitit Çiviyazısı İşaretlerinin Okunmasında Kullanılan İmge İşleme Algoritmaları.....	19
3.2.1 Algoritma 1: B.U. algoritması.....	21
3.2.2 Algoritma 2: Regionprops Kütüphanesi.....	23
3.2.3 Algoritma 3: SIFT.....	23
3.2.4 Algoritma 4: SURF.....	23
3.2.5 Algoritma 5: FAST.....	24
3.2.6 Algoritma 6: BRISK.....	24
3.2.7 Algoritma 7: MSER.....	24
3.2.8 Algoritma 8: ORB.....	25
3.2.9 Algoritma 9: Harris.....	25
3.2.10 Algoritma 10: Hausdorff uzaklığı.....	25
3.2.11 Algoritma 11: Hough Dönüşüm ile yapısal özniteliklerin bulunması.....	26
3.2.12 Algoritma 12: H.C. (Hierarchical Centroid) algoritması.....	29
3.2.13 Algoritma 13: HOG algoritması.....	30



<b>4 HİTİT ÇİVİYAZISI İŞARETLERİ ÜZERİNDE VERİ MADENCİLİĞİ</b>	
<b>UYGULAMALARI</b>	<b>32</b>
4.1 Veri Madenciliği Nedir?	32
4.2 Regionprops Kütüphanesi	34
4.3 Hitit Çiviyazısı İşaretlerinden Veri Kümesinin Oluşturulması	37
4.4 Hitit Çiviyazısı İşaretleri Üzerinde Kullanılan Veri Madenciliği Algoritmaları	37
4.4.1 K-Means kümeleme algoritması	37
4.4.2 J48 karar ağacı sınıflandırma algoritması	39
4.4.3 K-Nearest Neighbor sınıflandırma algoritması	40
4.4.4 Yapay sinir ağları sınıflandırma algoritması	41
<b>5 BULGULAR VE TARTIŞMALAR</b>	<b>43</b>
<b>6 ÖZET VE SONUÇLAR</b>	<b>58</b>
KAYNAKLAR LİSTESİ	60
WEB SAYFALARI LİSTESİ	64
EKLER	65
EK 1. Vanseveren Dijital Çiviyazısı İşareti Listesindeki İlk 16 İşaret Örneği	65
EK 2. Hata Oranları Çizelgesi	66

## ŞEKİLLER LİSTESİ

### Sayfa

Şekil 2.1 Kadeş antlaşmasının Boğazköy'de bulunan kopyası .....	10
Şekil 2.2 Hitit döneminde Anadolu (M.Ö 1650-1800).....	11
Şekil 2.3 Hititlerin başkenti Hattuşa.....	12
Şekil 2.4 Çiviyazısının gelişim süreci.....	14
Şekil 2.5 Örnek bir kil çiviyazısı tableti.....	15
Şekil 2.6 Hitit çiviyazısındaki en temel işaretler.....	16
Şekil 3.1 Portal Mainz kaynağında bulunan bir tablet resmi.....	17
Şekil 3.2 Tablettten alınan işaretin dijital işaret listesindeki karşılığının bulunması.....	18
Şekil 3.3 HZL No.180 olan işaretin iskelete dönüştürülmüş hali.....	21
Şekil 3.4 İşaret resminin 9 eşit parçaya bölünmesi.....	22
Şekil 3.5 Hausdorff Uzaklığı algoritmasında uygulanan ön işlem aşamaları.....	26
Şekil 3.6 Hough dönüşüm algoritmasında uygulanan ön işlem aşamaları .....	27
Şekil 3.7 HZL No.180 olan çiviyazısı işaretinin farklı derinlik değerlerine göre parçalara ayrılması .....	30
Şekil 4.1 Bilgi keşfi aşamaları.....	33
Şekil 4.2 Bir Bounding Box örneği .....	35
Şekil 4.3 Bölgedeki objelerin ağırlık merkezi.....	36
Şekil 4.4 K-Means algoritması akış diyagramı.....	38
Şekil 4.5 Örnek bir karar ağacı.....	39
Şekil 4.6 Bir yapay sinir ağının yapısı.....	41
Şekil 5.1 Birbirine benzeyen örnek işaretler.....	44
Şekil 5.2 HZL No.49 olan işaret için sorgu örneği.....	45
Şekil 5.3 HZL No.180 olan işarete SIFT algoritması uygulanması.....	49
Şekil 5.4 Sınıflara ayrılmış örnek işaretler.....	51
Şekil 5.5 K-means algoritmasının görselleştirilmesi.....	52
Şekil 5.6 Oluşturulan J48 karar ağacı modeli.....	53
Şekil 5.7 J48 karışıklık matrisi.....	53

Şekil 5.8 K=6 değeri için K-NN karışıklık matrisi .....	54
Şekil 5.9 Sınıflandırmada kullanılan yapay sinir ağı modeli .....	55
Şekil 5.10 Yapay sinir ağı karışıklık matrisi .....	56

## ÇİZELGELER LİSTESİ

### Sayfa

Çizelge 3.1 Örnek çivi yazısı işaretlerinin yatay ve dikey çizgilerin bulunması .....	28
Çizelge 4.1 İki sınıf için karışıklık matrisi .....	34
Çizelge 5.1 Algoritmaların işaretlerin karşılığını bulma performansları .....	43
Çizelge 5.2 Algoritmaların geri getirim puanları .....	46
Çizelge 5.3 Algoritmaların çalışma süreleri .....	46
Çizelge 5.4 Ön eleme algoritmaları uygulanan örnek işaretler .....	48
Çizelge 5.5 Hitit Çivi yazısı işaretlerinin sınıf dağılımları .....	51
Çizelge 5.6 Seçilen K değerlerine göre K-NN doğruluk oranları .....	54
Çizelge 5.7 Kullanılan sınıflandırma algoritmalarının doğruluk değerleri .....	56

## SİMGELER VE KISALTMALAR LİSTESİ

IEEE	Institute of Electrical and Electronics Engineering
HZL	Hethitisches Zeichenlexikon: Inventar und Interpretation der Keilschriftzeichen aus den Bogazkoy –Texten
V.S.	Vanseveren
B.U.	Başkent Üniversitesi
SIFT	Scale Invariant Feature Transform
SURF	Speed Up Robust Features
FAST	Features From Accelerated Segment Test
BRISK	Binary Robust Invariant Scalable Keypoints
MSER	Maximally Stable External Regions
ORB	Oriented Fast and Rotated Brief
H.C.	Hierarchical Centroid
HOG	Histogram of Oriented Gradients
DP	Doğru Pozitif
YP	Yanlış Pozitif
YN	Yanlış Negatif
DN	Doğru Negatif
K-NN	K-Nearest Neighbor (Yakın Komşu)
YSA	Yapay Sinir Ağı

# 1. GİRİŞ

## 1.1 Problemin Tanımı

Bilindiği üzere, Anadolu, yüzyıllardan beri birçok kavmi ve bu kavimlerin meydana getirdikleri kültürleri coğrafyasında yaşatmıştır. Bu kavimlerden pek azı bir imparatorluk düzeyine geçebilmiştir. Bu düzeye ulaşabilen kavimlerden Anadolu tarihi için en önemlisi Hititlerdir [Memiş, 1995].

Milattan önce 1650-1200 yılları arasında Anadolu'da hüküm süren Hititler o dönem dünyasının büyük güçleri arasında sayılmışlardır. Hititlerin kullandığı ve Hititçe diye bilinen dil, Hint-Avrupa (Indo-European) dil ailesinin bir üyesidir. Daha da ilginç, Hititçe bu dil ailesinin hâlâ okunabilir ve dil bilgisi (gramer) kuralları bilinen en eski örneklerinden biridir. Bu özellik Hititleri ve Hititçeyi başta ABD, Almanya ve İngiltere olmak üzere çeşitli Batı ülkelerinde ilginç ve tarihsel olarak değerli kılmaktadır.

Sümerler tarafından icat edilen resim yazısı tarihsel süreçte gelişerek Hititler tarafından kullanılan kendi çiviyazılarına dönüşmüştür. Hititler bu çiviyazısını kullanarak kral yıllıkları, devlet antlaşmalarını, kanunları, dini törenleri, mektuplar gibi bilgileri tabletlere işleyerek yazılı hale getirmişlerdir. Hitit çiviyazısında işaretler özel olarak hazırlanmış, farklı boy ve şekillerdeki yaş kil tabletler üzerine kamış veya benzer aletlerle yazılmıştır. Metinler kil tabletler üzerine yazılıp sonra da fırınladığı için toprak altında binlerce yıl yapısı bozulmadan kalabilmiştir. Anadolu'da Boğazköy'de 20. yüzyılın başlarında başlamış olan ve günümüzde hala devam eden kazı çalışmalarında yaklaşık 30.000 adet çiviyazılı tablet bulunmuştur. Bu tabletler Anadolu'da ve dünyanın birçok yerindeki müzelerde korunmaktadır [Alp, 2000] .

Hititçe dilinin gramer kurallarını ilk çözen ve ortaya çıkaran kişi 20. yüzyılın başlarında Çek bilim adamı Bedrich Hrozny olmuştur [Karasu, 2013]. Gramer kuralları Hrozny tarafından ortaya konduktan sonra, günümüze kadar Hitit çiviyazılı metinlerin okunması, çevrilmesi, yorumlanması ve yeni gramer kurallarının kullanımı genel olarak "ele ve insana" dayalı olarak yapılagelmiştir. Metinlerin okunup gerekli

çevirilerin yapılabilmesi özel olarak yetişmiş, deneyimli ve uzman kişiler gerektirdiği için uzun süre ve emek isteyen, yorucu ve zaman alıcı bir uğraş olmuştur.

Hititler dünyada arşiv-kütüphane uygulamasını ortaya ilk toplumlardan biri olarak da tanımlanabilirler. Hititçe metinlerin yazıldığı tabletlerin şimdiye kadar göreceli olarak az bir bölümü gün ışığına çıkarılabılmış, daha da azı okunup değerlendirilebilmiştir. Çorum Boğazköy'de ele geçen Hitit çiviyazılı tabletler UNESCO tarafından 22 Ocak 2002 tarihinde Dünya Kültür Mirasında Dünya Belleğine (Memory of the World Register) kaydedilmiştir<sup>1</sup>.

Hitit çiviyazılı tabletlerin üzerindeki metinlerin günümüz dillerine çevirisi toplamda üç temel süreçten oluşmaktadır. Bu süreçler sırasıyla işaretlerin ve işaret dizilerinin Latin alfabesindeki karakterlere dönüştürülmesi (transliteration), Latince karakter dizilerinin Hititçeye dönüştürülmesi (transcription) ve Hititçeye dönüşmüş olan metinlerin Türkçe, Almanca ve İngilizce gibi dillere çevirisi (Translation) 'dir [Van den Hout, 2011]. Bahsedilen üç aşama da zorlu ve vakit alan süreçlerdir. Günümüzde bu süreçler, konusunda uzman olan kişiler tarafından gerçekleştirilmektedir. Dolayısıyla bu süreçlerin bilgi ve iletişim teknolojileri kullanılarak otomatik bir hale getirilmesi; hem maliyeti düşürmesi hem de harcanan zamanı kısaltmasından dolayı büyük önem taşımaktadır. Bu süreçlerin bilgi ve iletişim teknolojileri kullanılarak yapılmasının bir diğer artısı ise müzelerde bulunan ve kazı yerlerinde ortaya çıkartılan tablet metinleri üzerine kolayca uygulanabilmesidir.

## 1.2 Literatür İncelemesi

Literatürde Hitit çiviyazısı üzerine bilgisayar destekli olarak yapılan ilk tez 1988 yılında ODTÜ'de yapılmıştır [Gürsel, 1988]. Bilgisayar desteği ile Hititçe işaretlerin okunması ile ilgili bir diğer tez çalışması da 2014 yılında Başkent Üniversitesinde gerçekleştirilmiştir [Dik, 2014]. Bilgisayar desteği ile Hitit çiviyazısı üzerine yapılan en son tez çalışmalarından birisi de 2015 yılında Başkent Üniversitesinde

---

<sup>1</sup> <http://whc.unesco.org/en/list/377>

gerçekleşmiştir [Yeşiltepe, 2015]. Ayrıca Hitit çiviyazısı bir semboller dizisi olduğu için Çince, Arapça, Japonca, Banglaca gibi Latin alfabesinden farklı alfabeler içeren diller üzerine yapılan karakter tanıma ve okuma çalışmaları da önceki çalışmalara dâhil edilebilir. Son olarak tabletlerin dijital ve üç boyutlu hale getirilmesi çalışmaları da çiviyazılı tabletlerin bilgisayar ortamına aktarılması işleminin önemli bir parçasını gerçekleştirdiği için önceki çalışmalara örnek gösterilebilir.

Dik [2014] Hitit çiviyazısı işaretlerinin otomatik çevirisini gerçekleştiren bir çalışma yapmıştır. Bu çalışmada Hitit çiviyazısı işaretlerini içeren bir dijital sözlük veritabanı tasarlamış ve Hausdorff uzaklığı yöntemiyle çiviyazısı işaretlerinin okunması ile ilgili bir çalışma yapmıştır. Çalışmalarını Hrozny'nin ilk çözdüğü Hititçe cümle üstünde gerçekleştirmiştir.

Hahn et al. [2006] Digital Hammurabi adı verilen çalışmada kilden yapılan çiviyazısı tabletleri dijitalleştirmek için üç boyutlu bir tarayıcı sistemi geliştirilmiştir. Bu tarayıcıyı kullanarak tabletlerin üç boyutlu bir obje olarak bilgisayar ortamında saklanmasını sağlanmıştır. Daha sonra kullanıcı bu tableti istediği yönde ve ışık düzeyinde bilgisayar ortamında görüntüleyebilmektedir.

Tyndall [2012] kazılarda çıkarılan parçalar halinde ve tek metine ait olan tabletlerin Hititçe metinlere dönüşmüş kısımlarını kullanarak veri madenciliği yöntemleriyle birleştirilmesi üzerine bir çalışma yapmıştır. Tabletlere uzmanlar tarafından verilen envanter numarasını sınıf olarak atadıktan sonra uzmanlar tarafından eşleştirilen kırık parçaları o sınıfa ait olarak kabul edip kırık parçalardan bir veri kümesi oluşturmuştur. Naive Bayes ve Maximum Entropy sınıflandırıcılarını kullanarak deneyler yapmış ve sınıflandırıcı performanslarını karşılaştırmalı olarak vermiştir.

Edan [2013] Sümerce çiviyazısı tabletlerindeki çiviyazısı işaretler üzerinde veri madenciliği yöntemlerini uygulamıştır. Bir tarayıcı yardımıyla işaretleri elde edip, işaretler üzerinde gürültü giderme ön işlemleri yapmıştır. Daha sonra öznitelik vektörleri oluşturulmuştur, bunlar işaretlerin dikey ve yatay dağılımları, ağırlık merkezi koordinatları ve birbirine bağlı obje sayılarıdır. İşaretlere K-means kümeleme



algoritması uygulanarak işaretlerin ait olduğu sınıflar bulunmuş, daha sonra yapay sinir ağları kullanılarak sınıflandırma performansı ölçülmüştür.

Yousif et al. [2006] “Intensity Curve” yöntemini kullanarak Sümerce çivi yazısı işaretlerinin okunması üzerine çalışmalar yapmışlardır. Intensity Curve adı verilen bir yöntemle işaretin öznitelikleri çıkarmışlardır. İşaretler önce yatay eşit bölümlere ayrılır ve her bir bölümdeki piksellerin değeri ve konumu hesaplanır. Daha sonra bu değerler bir eğriye dönüştürülür ve eğrinin yerel minimumları öznitelikleri oluşturur. Aynı işlem dikey bölümler için de gerçekleştirilir. Seçilen bir işaretin gürültülü, boyutu büyütülmüş ve boyutu küçültülmüş hali ile veritabanında bahsedilen öznitelikler üzerinden sorgular yapıp orijinal işareti bulma performansı raporlanmıştır.

Ahmed [2012] “Symbol Structural Vector” yöntemi ile Sümerce çivi yazısı işaretlerinin çevrimiçi okunması üzerine çalışmalar yapmıştır. Bu yöntem çivi yazısı işaretlerinin iskeletini çıkararak işleme başlamaktadır. Daha sonra ise iskelet üzerinden şeklin kırılma, bağlantı noktaları gibi çeşitli öznitelikler çıkarılmaktadır. Bu öznitelikler bir vektör olarak veritabanında saklanmaktadır. Çalışmada gerçek zamanlı olarak çizilen çivi yazısı işaretinin veritabanında bulunan diğer işaretlerle karşılaştırma yapılarak eşleşme performansı raporlanmıştır.

Das et al. [2011] Çoklu sınıflandırıcılar kullanarak Bangla karakterlerinin okunması üzerine bir çalışma yapmışlardır. Bangla alfabesinden basit geometrik şekillere sahip olan işaretler kullanılmıştır. İşaretlere siyah-beyaza çevirme, gürültü giderme ve ölçeklendirme gibi ön işlem aşamaları uygulanmıştır. Daha sonra farklı sınıflandırıcılar için farklı öznitelik vektörleri oluşturulmuştur. Bu öznitelik vektörleri işaretlerin geometrik özellikleri ve piksel dağılımları gibi elemanlardan oluşmaktadır. İki aşamalı bir sınıflandırma sistemi yaratılıp her bir sınıflandırıcı için seçilen öznitelik vektörlerinin sınıflandırmaya olan etkisi, sınıflandırma performansı olarak raporlanmıştır.

Aghav et al. [2012] çalışmasında dökümanlar üzerindeki karakterlerin okunmasında kullanılan yöntemleri ve aşamaları irdelemiş ve her bir aşamada hangi işlemlerin

gerçekleştirildiğini göstermiştir. Kısaca bu çalışma kâğıt üzerinde olan karakterlerin okuma tekniklerinin bir özetini sunmaktadır.

Sundar ve John [2013] Tamil<sup>2</sup> karakterlerinin okunması ile ilgili bir çalışma gerçekleştirmiştir. Her bir işaret için iki farklı öznitelik vektörü oluşturulmuştur, bunlardan ilki HOG algoritmasıyla oluşturulan vektör ikincisi ise işaretin geometrik özelliklerini içeren vektördür. Yapay sinir ağları kullanarak bu iki öznitelik yöntemi karşılaştırılmış ve sonuçları sınıflandırma performansı olarak verilmiştir.

Holambe ve Tholl [2013] birden çok sınıflandırıcı ve öznitelik kullanarak Devanagari<sup>3</sup> karakterlerinin okunmasını gerçekleştirmişlerdir. Üç tip öznitelik vektörü kullanmışlardır. İki işarettaki çizgi sayısı ve çizgi uzunluğu gibi öznitelikleri içeren istatistiksel öznitelikler, ikincisi Euler sayısı ve alan gibi öznitelikleri içeren yapısal öznitelikler, son olarak da küresel dönüştürme momentlerinden olan Zernike moment kullanılmıştır. Sınıflandırıcı olarak da SVM (Support Vector machine) ve K-nearest neighbor yöntemleri kullanılmıştır. Her bir öznitelik vektörünün iki sınıflandırıcıda ki sınıflandırma performansları ve iki sınıflandırıcının birleşmesiyle oluşan sınıflandırıcıdaki sınıflandırma performansları ölçülmüştür.

### **1.3 Tezin Amacı**

Bu tez kapsamında yapılan çalışmalar Hitit çiviyazısı tabletler üzerinde bulunan işaretlerin Latin yazısına dönüştürülmesinde yardımcı olabilecek çalışmaları içermektedir. Yapılan çalışma tablet metinlerinde bulunan çiviyazısı işaretlerin çeşitli görüntü işleme yöntemleri ile okunup veritabanında bulunan işaretlerle eşleştirilerek Latin yazısına dönüştürülmesi sürecini gerçekleştirmektedir. Çalışmada Hitit çiviyazısı işaretlerinin okunmasında kullanılan yöntemlerin işaret okuma performansları karşılaştırmalı olarak verilmiştir. Ayrıca İşaretlerin okunma performansını hızlandıran yöntemler de bu tez çalışmasında yer almaktadır.

---

<sup>2</sup> Tamil alfabesi Hindistan, Sri Lanka ve Malezya ülkelerinde kullanılan bir alfbedir.

<sup>3</sup> Devanagari Alfabesi Nepal ve Hindistan ülkelerinde kullanılan bir alfbedir.

Bu tez çalışmasının bir diğer boyutu olan veri madenciliği uygulamalarında ise Hitit çiviyazısı işaretlerinin geometriksel özelliklerine göre kategorilere ayrılması amaçlanmıştır. Çalışmanın bu boyutunun amacı birbirine benzer işaretlerin kategorilere ayrılmasıyla tabletlerdeki işaretlerin okuma işlemini hızlandırmaktır. Daha sonra kategorilere ayrılan işaretlere veri madenciliğinde kullanılan çeşitli sınıflandırma yöntemleri uygulanmıştır. Sınıflandırma uygulanmasının amacı kategorisi belli olan işaretlerden bir model oluşturularak, kategorisi belli olmayan işaretlere bu model üstünden kategori ataması yapabilmektir. Uygulanan algoritmaların sınıflandırma performansları ölçülüp karşılaştırmalı olarak verilmiştir.

#### **1.4 Tezin Yapısı**

Hitit çiviyazısı işaretlerinin okunması ve Hitit çiviyazısı işaretleri üzerinde veri madenciliği uygulama örneklerinin gerçekleştirildiği bu tez çalışmasının ilk bölümünde tez çalışmasının konusu anlatılmış, konu hakkında daha önce yapılan çalışmalara değinilmiş sonrasında ise yapılan tez çalışmasının amacı ve kapsamı hakkında bilgiler verilmiştir.

Tezin ikinci bölümünde ilk olarak Hititler ve Hitit imparatorluğu hakkında bilgiler verilip Hitit uygarlığı tarihçesi anlatılmıştır. Daha sonraki bölümde ise çiviyazısının tarihsel gelişimi ve Hitit çiviyazısının özelliklerinden bahsedilmiştir.

Tezin üçüncü bölümünde Hitit çiviyazısı işaretlerinin okunması ile ilgili çalışmalar yer almaktadır. İlk olarak tabletlerde yer alan çiviyazısı işaretlerinin dijital imajının alınması ve işaret veritabanını oluşturan dijital işaret listesi anlatılmıştır. Daha sonra tabletlerde bulunan çiviyazısı işaretlerinin okunmasında kullanılan algoritmalar açıklanmıştır.

Tezin dördüncü bölümünde ise Hitit çiviyazısı işaretleri üzerinde veri madenciliği uygulamaları ile ilgili çalışmalara yer verilmiştir. İlk olarak veri madenciliğinin kısa bir tanımı verilip, veri madenciliğinde kullanılan sınıflandırma ve kümeleme teknikleri konularına değinilmiştir. Daha sonra Hitit çiviyazısı işaretlerinin veri madenciliği

algoritmalarında kullanılmak üzere hazırlanan veri kümesinin nasıl oluşturulduğundan bahsedilmiştir. Bu bölümde son olarak Hitit çiviyazısı işaretleri üzerine uygulanan veri madenciliği algoritmaları anlatılmıştır.

Tezin son bölümü olan beşinci bölümde ise örnek olarak seçilen Hitit çiviyazısı işaretlerinin tabletlerden okunması ile ilgili çalışmaların sonuçları ile beraber Hitit çiviyazısı işaretlerine uygulanan veri madenciliği algoritmalarının sonuçlarına yer verilmiştir. Son olarak da yapılan tez çalışmasının geliştirilmesine yönelik önerilerden bahsedilmiştir.

## 2. HİTİTLER VE HİTİT ÇİVİYAZISI

### 2.1 Hititler ve Hitit İmparatorluğu

Milattan önce 1650-1200 yılları arasında Anadolu'da hüküm süren Hititler o dönem dünyasının büyük güçleri arasında sayılmışlardır. Hititlerin Anadolu'ya göç tarihi kesin olarak saptanamamıştır ama Hititlerin Anadolu'ya geliş hikâyeleri hakkında değişik görüşler bulunmaktadır. Hititler, Anadolu'ya bir görüşe göre Kafkaslar üzerinden diğer bir görüşe göre ise Boğazlar üzerinden gelmişlerdir. Hititlerin Kuzey Suriye üzerinden Anadolu'ya geldikleri de düşünülmektedir. Ayrıca, Orta Anadolu'nun yerlisi bir millet olabileceği de düşünülmektedir [Memiş, 1995].

Hititçe Hint-Avrupa kökenli bir dil olmasına rağmen Sümerce, Akadça gibi çok sayıda yabancı kelime de içerdiği için yabancılaşmıştır; fakat bu durum Hitit ana dilinin yapısını bozmamıştır. Dil üstüne yapılan çalışmalar ve kültür etkileşimi hakkındaki bilgiler ışığında, Hititlerin, Anadolu'da yaşayan topluluklarla kaynaşarak karma bir uygarlık ortaya çıkardıkları kanısına varılabilir [Memiş, 1995]. Hitit uygarlığının tarihi Eski Krallık/Eski Hitit (MÖ 1660-1460) ve Yeni Krallık/Hitit imparatorluğu dönemi (MÖ 1650-1200) olarak ikiye ayrılmaktadır [Çığ, 2000]

Eski Krallığın ilk kurucu hükümdarı Tabarna unvanına sahip I.Hattuşili'dir. "Tabarna" Hatti<sup>4</sup> dilinde hükümdar anlamına gelmektedir [Ünal, 2002]. Hititler bu dönemde Hatti prensliklerini ele geçirmeye başlamışlardır ve hükümdar, Hattilere ait olan Hattuşa şehrine yerleşmiş, kendisine de Hattuşili denmiştir<sup>5</sup>. I Hattuşili'nin MÖ 1660-1630 yılları arasında hüküm sürdüğü varsayılmaktadır. I. Hattuşili askeri seferleri Kuzey Suriye bölgesine ve isyan eden beylikler üzerine yaparken, işgal edilen toprakları daha çok vasal/bağımlı olarak kendine bağladı. Böylece Hitit Devleti her seferde gitgide büyümeye ve sınırlarını genişletmeye başladı.

---

<sup>4</sup> Hatti'ler Hititlerden önce Anadolu'da yaşamakta olan halktır.

<sup>5</sup> <https://tr.wikipedia.org/wiki/Hititler>

Hitit Devleti kuruluşundan itibaren Mezopotamya ve Mısır kapılarını aralayan Kuzey Suriye topraklarını el altında tutmaya önem vermiştir. Öte yandan Hititler, Kuzey Suriye ve Hurri devletine karşı yaptıkları seferlerde fethedilen uygarlıkların kültürü ve yaşam tarzından etkilenmişlerdir. Aynı dönemde eski Babil çiviyazısı Anadolu'ya getirilmişti ve Akadça + Hititçe olarak tabletlere yazılıyordu [Akurgal, 2005].

I.Hattuşili'den sonra birçok hükümdar Hitit devletini başarı ile yönetmiştir. Bunlardan biri olan I.Murşili, sefeli I. Hattuşili'nin yayılma politikasını izlemiş Suriye topraklarını alarak Babil'e kadar ilerlemiştir (yaklaşık olarak MÖ 1594).

Hitit İmparatorluk döneminde, MÖ 1315-1282 yılları arasında yaşamış Hitit Hükümdarı II. Mutavalli Hitit İmparatorluğunun en büyük ve en başarılı krallarından biridir. II. Ramses'in Mısır'ı yönettiği dönemde Mısır'la savaştı. Kadeş (Kuzey Suriye) Meydan Savaşı MÖ 1274 tarihinde yapıldı. Bu savaşla ilgili bilgiler birçok Mısır tapınağının duvarlarında kazılıdır. Bu savaş sonucu Mısır devleti topraklarına geri dönmüş, Hititler Suriye bölgesinde Şam'a kadar ilerlemiştir. Hititler Suriye ve Amurru devletlerinde Mısır egemenliğini sona erdirip bu devletleri kendi bünyesine katmıştır [Çiğ, 2000].

Hitit İmparatorluğunun bir diğeri önemli hükümdarı MÖ 1267-1237 yılları arasında yaşamış olan III. Hattuşili'dir. Mısırlılarla barış antlaşmasını yapan kraldır. Kadeş Barış Antlaşması MÖ 1269 yılında yapılmıştır [Çiğ, 2000]. Kadeş Antlaşması olarak bilinen bu antlaşma iki süper devlet arasında barış sağlamak amacıyla yapıldığı bilinen en eski yazılı bir diplomatik harekettir. Antlaşmanın içeriğinde, iki kralın isimleri, dostluk ilişkileri, şahit olarak Hitit ve Mısır tanrılarının gösterilmesi, antlaşmayı bozacak olanın lanetlenmesi, koruyana mutluluğun gelmesi gibi maddeler yer almaktadır [Çiğ, 2000]. Bu antlaşmanın bir kil kopyası Boğazköy/Hattuşa 'da bulunmuştur. Antlaşmanın aslı ele geçmeyen Gümüş tablettir. Bulunan bu kil tablet kopyası İstanbul Arkeoloji Müzesinde sergilenmektedir; sergilenen kopya Şekil 2.1 'de görülmektedir<sup>6</sup>.

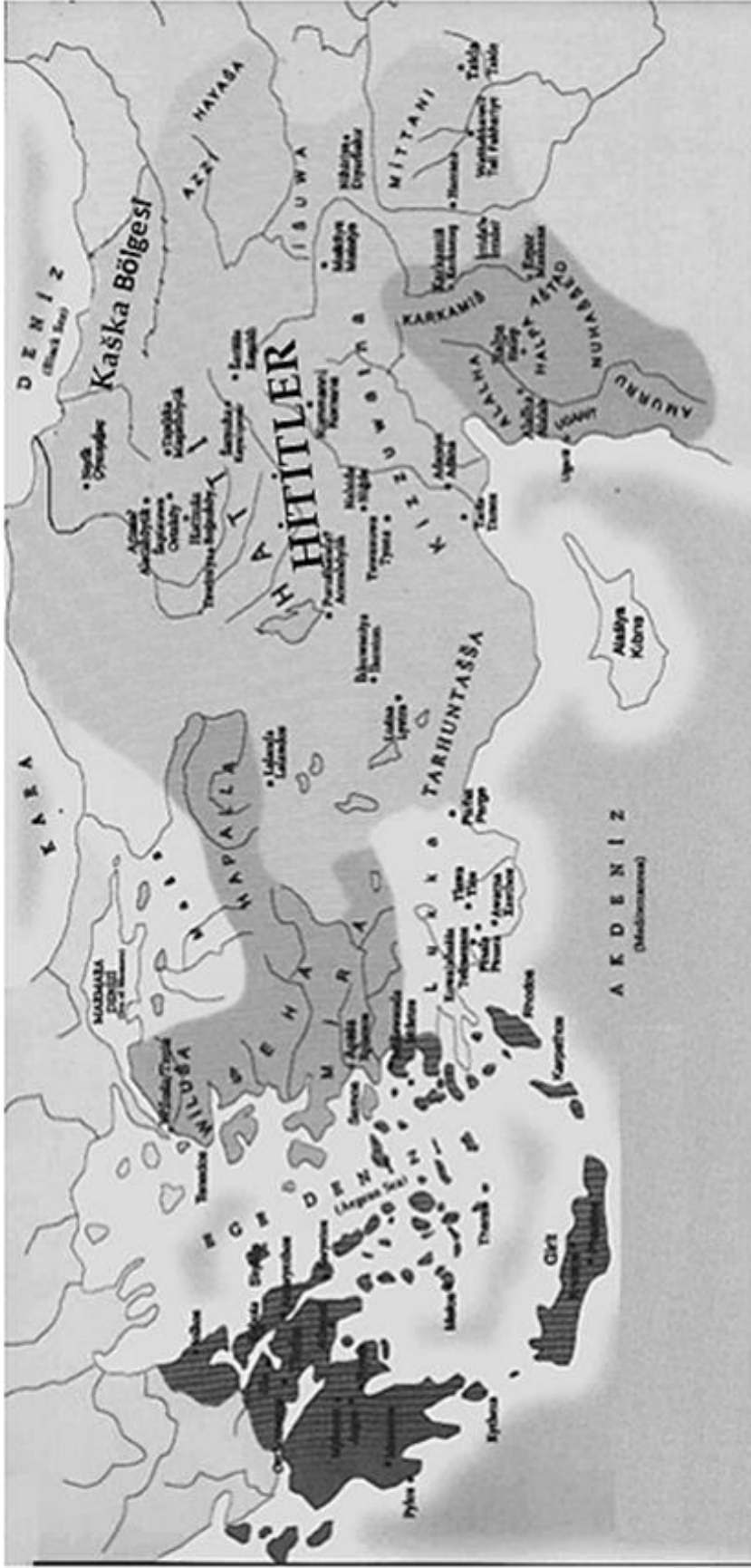
---

<sup>6</sup> [https://tr.wikipedia.org/wiki/Kadeş\\_Antlaşması](https://tr.wikipedia.org/wiki/Kadeş_Antlaşması)



Şekil 2.1 Kadeş antlaşmasının Boğazköy'de bulunan kopyası

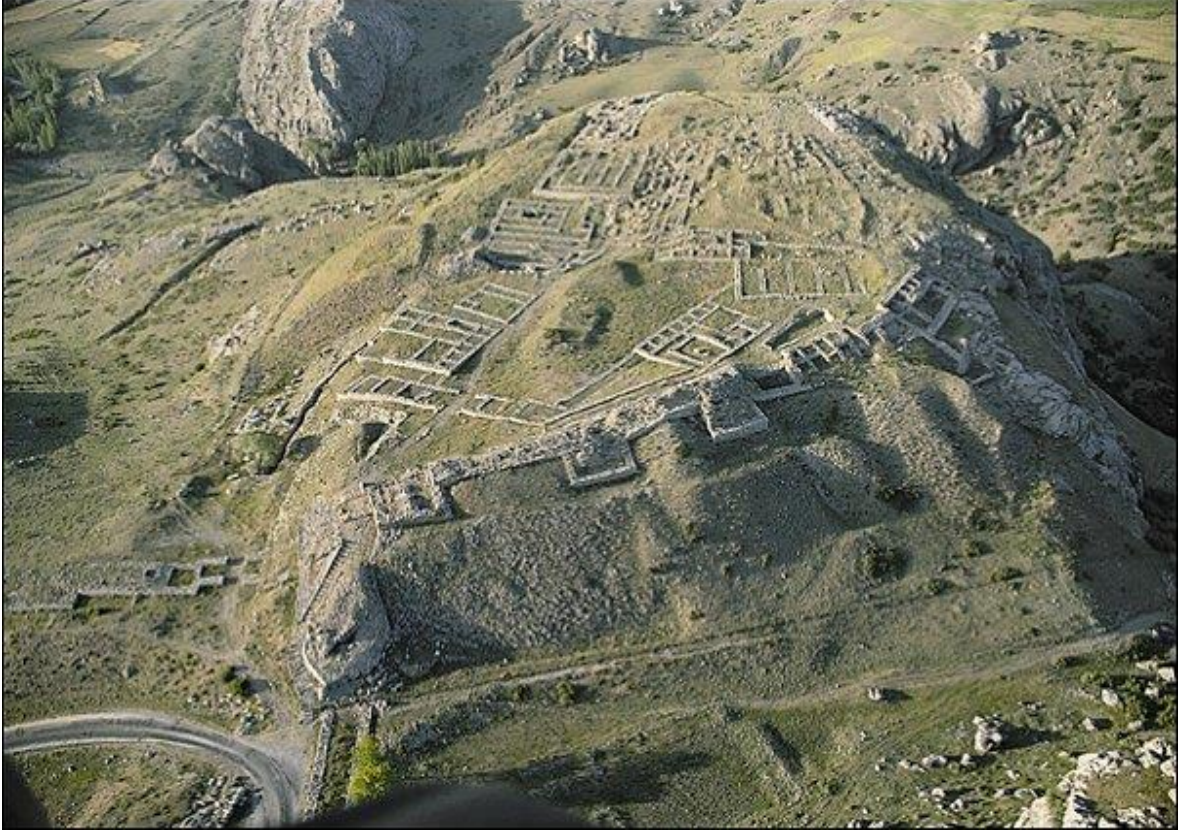
Hititler Döneminde Anadolu'daki devletlerin durumu ve Hitit Etkileri Şekil 2.2 ile gösterilmektedir [KARASU, 2013].



Şekil 2.2 Hitit döneminde Anadolu (M.Ö 1650-1800)



Hitit Devleti'nin başkenti olan Hattuşa İç Anadolu'da Çorum ilinin yaklaşık 80 kilometre güneybatısında bulunan Boğazkale ilçesinde yer almaktadır. Günümüzde bu bölge Boğazköy olarak geçmektedir. Döneminin en büyük şehirleri arasında gösterilen Hattuşa, Hitit devleti'nin kurulmasından sonra I. Hattuşili tarafından devletin başkenti olarak seçilmiştir. Hattuşa Şehrinin günümüzdeki görünümü Şekil 2.3 ile verilmiştir<sup>7</sup>.



Şekil 2.3 Hititlerin başkenti Hattuşa

Hitit devletinin çöküşüne yol açan temel nedenin deniz kavimleri göçü olabileceği söylenmektedir. Bu göçler denizlerden gelen istilacıların zayıflayan devleti ele geçirmesine sebep olmuştur. Diğer bir görüşe göre de Hitit devletinin halktan alınan vergiyi arttırarak Anadolu halkını fakirleştirdiği ve halkın isyanlar başlatarak Hitit devletinin yok olmasına sebep olduğu söylenmektedir [Ünal, 2002].

---

<sup>7</sup> <http://www.kulturvarliklari.gov.tr/TR,44427/hattusas-bogazkoy---hitit-baskenti-corum.html>

## 2.2 Hitit Çiviyazısı

Yazı, insanın konuşma dışında duygularını, düşüncelerini ve isteklerini aktarabilmek için başvurduğu, birtakım işaretlerden oluşmuş ve söze göre daha kalıcı olan bir anlatım aracı olarak düşünülebilir [Karasu, 2006]. Kınal [1971] “yazı uygarlıklar tarihinde o kadar önemli bir kültür ögesidir ki tarih onunla başlar” demektedir. Yazı ilk olarak tarımsal ürün, hayvan, insan, gibi varlıkları resmetmek için ortaya çıkan resim yazısı olarak üretilmiş ve daha sonra gelişerek dilde bulunan hece, ses ve sözcükleri temsil etmeye başlamıştır.

Sümerler, zamanında yüce bir uygarlık oluşturmuşlardı. Bu uygarlığın dünyada bilinen en eski yazılı kayıtları, Irak’ın güneyinde bulunan Uruk kentinde ki Eanna tapınağında keşfedilmiştir. Burada yaşayanların ekonomik aktivite bilgilerini içeren çok sayıda resim yazısı/piktografik tablet bulunmuştur. Sümerler bu dönemde “teokratik devlet sosyalizmi” adı verilen bir yönetim şekli ile yönetiliyorlardı ve bu yönetim şeklinde insanlar dâhil olmak üzere her şey tanrının malıydı ve tanrıya aitti [Kınal, 1971]. İnsanlar tanrılara hizmet ediyorlardı ve dolayısıyla kazançlarını tapınaklara getiriyorlardı. Bu tapınaklara gelen ürünlere göre işaretler belirlenmişti. İnsanlar yetiştirdiği ürünleri bu tapınaklara getiriyor ve kâtipler bu ürünleri kil tabletlere yazıyorlardı. Böylelikle Sümerler “piktografik” denilen resim yazısını icat etmiş oldular (M.Ö. 3200) [Kınal, 1971].

Sümerler artık resim yazısının yetersiz kalmaya başladığını fark etmeye başlamış ve dilin fonetik özellikleri çiviyazısına aktarmaya başlamışlardı. Resim ile ifade edilen varlıklar artık birleşerek heceleri ve sözcükleri oluşturarak dili zenginleştirmeye başlamıştır. Çiviyazısının yıllar içindeki değişimi ve gelişim süreci Şekil 2.4 ile gösterilmektedir<sup>8</sup>.

---

<sup>8</sup> <http://tarih.tumders.com/wp-content/çivi-yazısı.jpg>

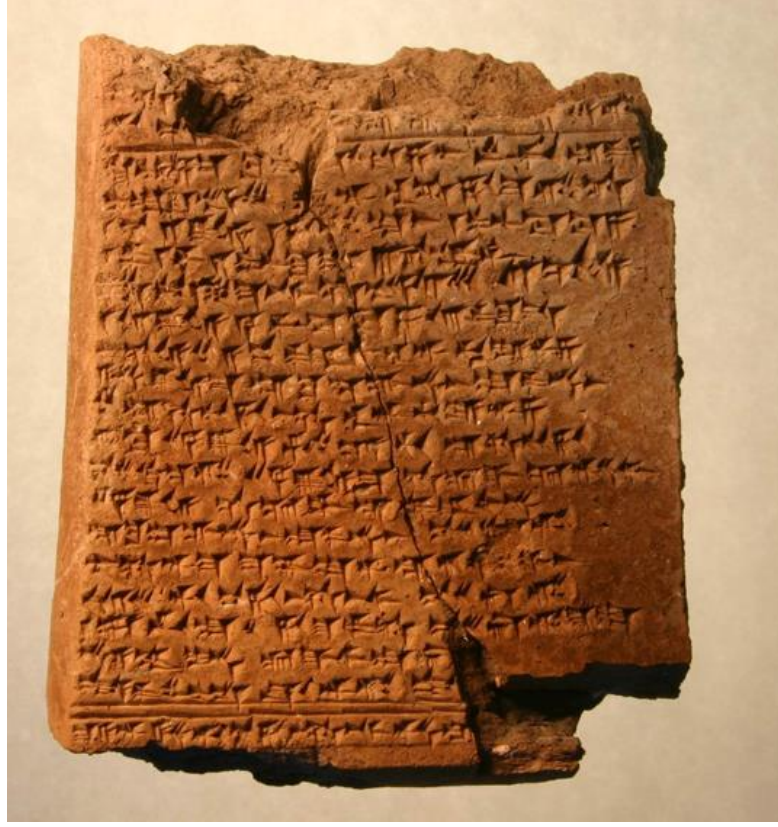
					SAG baş
					NINDA ekmek
					GU <sub>7</sub> yemek
					AB <sub>2</sub> boğa
					APIN saban
MÖ 3100	MÖ 3000	MÖ 2500	MÖ 2100	MÖ 700	Sümerce okunuşu ve anlamı

Şekil 2.4 Çiviyazısının gelişim süreci

Çiviyazısının Anadolu'ya gelişi, Mezopotamya'da ilk tabletlerin ortaya çıkmasından sonra, Asurlu tüccarların Orta Anadolu'ya gelip ticaret yapmasıyla başlamıştır. Ticaretlerin detayları, ne kadar ürün alındığı, ürünlerin türü gibi veriler tabletlerde tutuluyordu. Yani yazının Anadolu'da kullanılmasının başlangıç nedeni ticareti takip etmek olduğu söylenebilir. Asurlu tüccarlarla ticaret yapan Anadolu kralları tabletler yazdırmaya başladılar. Anadolu'da M.Ö. 2000'li yılların başında Hitit egemenliği ile birlikte çiviyazısı ve tablet kullanımı yaygınlaşmıştır. Yazışmalarda çiviyazısını kullanan Hititler, mühürlerde ve kaya anıtlarında resimyazı (hiyeroglif) kullanmışlardır [Hırçın, 2000].

Hitit çiviyazısında, yazıyı oluşturan temel işaretler özel olarak hazırlanmış yaş kil tabletler üzerine kamış veya benzeri aletlerle yazılmıştır. Kil tabletler üzerine metinler yazıldıktan sonra da fırınladığı için toprak altında binlerce yıl bozulmadan kalabilmiştir. Örnek bir kil tablet Şekil 2.5 'de gösterilmektedir<sup>9</sup>.

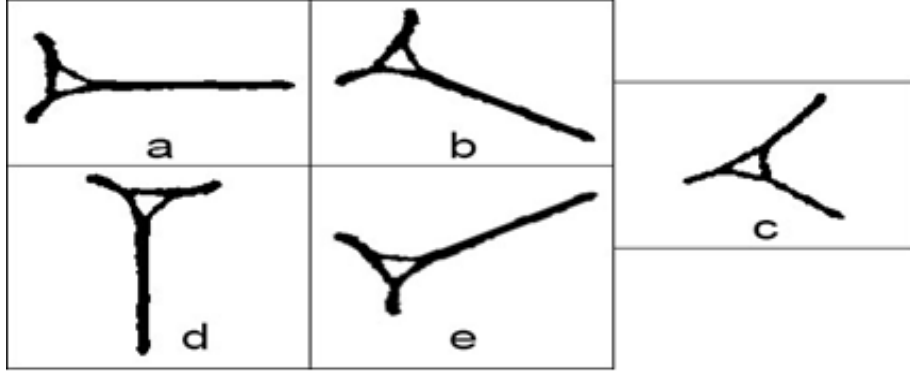
<sup>9</sup> <http://www.kulturvarliklari.gov.tr/Resim/104902,hitit-civi-yazili-tablet---kayalipinar-2014-1.png>



Şekil 2.5 Örnek bir kil çiviyazısı tableti

Hitit çiviyazısında toplam 375 adet farklı işaret bulunmaktadır [Ruster and Neu, 1989]. Bu işaretlerin hepsi de en temel 5 işaret ile tanımlanmışlardır [Aktaş ve Gürsel, 1988], [Karasu, 2013]. Şekil 2.6 içinde Hitit çiviyazısına ait en temel beş işaret verilmiştir. Her bir işaret bir kelimeyi veya bir heceyi temsil ettiği gibi birkaç işaret birleşerek de bir kelimeyi oluşturabilir. En temel işaretlerden biri olan yatay işaret, çamur halindeki kil tablet üzerinde yazı aletinin eğik olarak kullanılması ile oluşturulan bir işarettir (Şekil 2.6 a). Diğer temel işaretler bu işaretin farklı açılarda ( $-45^\circ$ ,  $-90^\circ$ ,  $+45^\circ$ ) uygulanması ile oluşturulmuş olan işaretlerdir (Şekil 2.6 b, d, e ). İşaretlerin temel bileşenleri arasında farklı olan ve “köşe çengeli” olarak adlandırılan işaret ise (Şekil 2.6 c) yazı aletinin çamur halindeki kil tablet üzerinde dikey olarak bastırılması ile oluşturulan bir işarettir.

1989 yılında C. Ruster ve E. Neu adlı araştırmacılar HZL (Hethitisches Zeichenlexikon) adında Hitit çiviyazısı işaretlerini ve işaretlerin anlamlarını içeren bir nevi işaret sözlüğü yayımlamışlardır [Ruster and Neu, 1989].



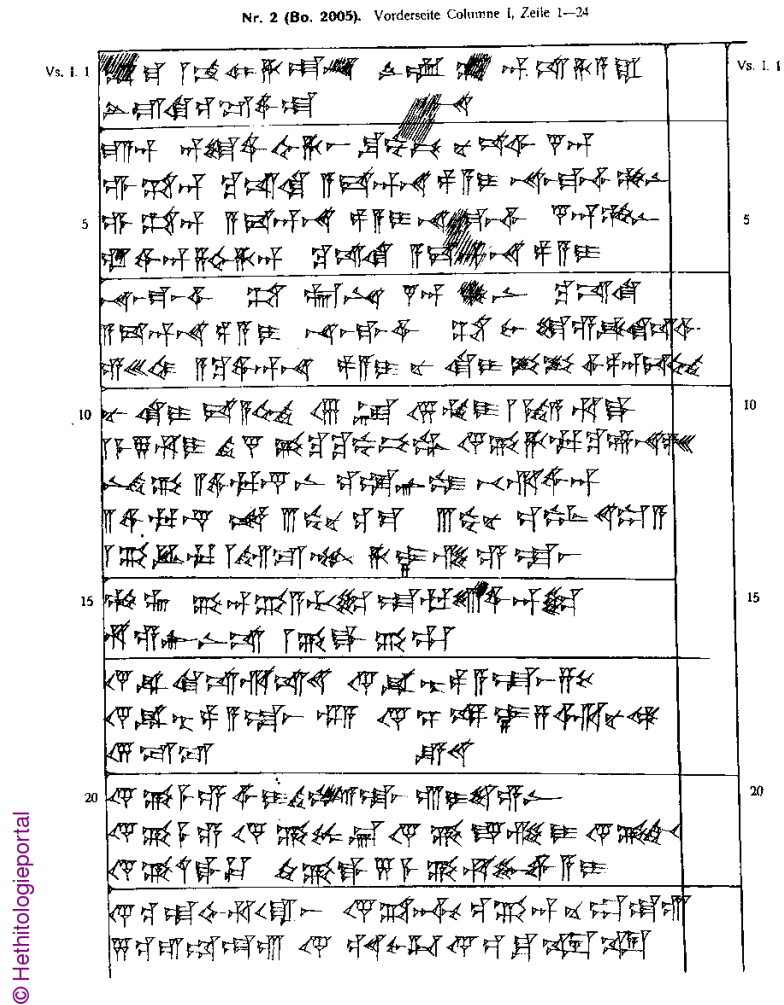
Şekil 2.6 Hitit çiviyazısındaki en temel işaretler

HZL sözlüğünde işaretler numaralar verilerek indekslenmiştir. Bu numaraya işaretin HZL numarası adı verilmiştir. Hititçe üzerine yapılan çalışmalarda işaretler HZL işaret numaraları üzerinden referans gösterilerek çalışılmıştır.

### 3. HİTİT ÇİVİYAZISI İŞARETLERİNİN OKUNMASI

#### 3.1 Hitit Çiviyazısı İşaretlerinin Dijital İmajının Alınması

Hitit çiviyazısı işaretlerin dijital ortama aktarılması için Würzburg Üniversitesinin internet sayfası bünyesinde bulunan Portal Mainz<sup>10</sup> adlı internet sayfası kaynak olarak alınmıştır. Ayrıca bu internet sayfasında çok sayıda tablet resmi de bulunmaktadır. Tez çalışmasında bu kaynaktan bulunan metin kopyaları kullanılmıştır. Portal Mainz'daki bir metin kopyası örneği Şekil 3.1 ile verilmiştir.



Şekil 3.1 Portal Mainz kaynağında bulunan bir tablet resmi

<sup>10</sup> <http://www.hethport.uni-wuerzburg.de/HPM/index.html>



Çalışmada veritabanı vazifesi gören V.S. işaret listesinde bulunan işaretlerin, ekran görüntüsü alınıp işaretler Portal Mainz'da bulunan tablet resimlerindeki işaretlerin alınmasında kullanılan aynı işlem ile kesilerek çalışmalarda kullanılmıştır.

İşaretlerin dijital imajlarının alınmasından sonra çiviyazısı işaretlerinin okunmasında kullanılan algoritmalarda işaretler çeşitli ön işlem aşamalarından geçirilip algoritmalarda kullanılmıştır.

### **3.2 Hitit Çiviyazısı İşaretlerinin Okunmasında Kullanılan İmge İşleme Algoritmaları**

Hitit çiviyazısı işaretlerinin okunmasında toplam 13 algoritma kullanılmıştır. Kullanılan algoritmaların bazıları MATLAB<sup>13</sup> (araç kutusu) yazılım geliştirme ortamında bulunan bazı fonksiyonların kullanılması ile hazırlanan algoritmalar (Örn: Algoritma 1). Örneğin, Algoritma 2 ise doğrudan MATLAB Kütüphanesi içinden alınıp kullanılmıştır. Ayrıca Algoritma 3, 4, 5 gibi algoritmalar da sıra ile verilmiştir ve MATLAB dışındaki kaynaklardan alınmıştır (OpenCV<sup>14</sup>).

**Algoritma 1:** ((Başkent Üniversitesi) B.U. Algoritması) İşaret resminin parçalara bölünmesi ve her parçadaki siyah piksel sayıları arasında fark kullanılarak hata oranı hesaplanır.

**Algoritma 2:** MATLAB Regionprops<sup>15</sup> Kütüphanesi kullanılır. Bu kütüphane bir resmin geometrik özniteliklerini hesaplamaya yardımcı olur.

**Algoritma 3:** SIFT Algoritması (Scale Invariant Feature Transform) [Lowe, 2004].

**Algoritma 4:** SURF Algoritması (Speeded Up Robust Features) [Herbert et al., 2006].

---

<sup>13</sup> <http://www.mathworks.com/products/matlab/>

<sup>14</sup> <http://opencv.org/>

<sup>15</sup> <http://www.mathworks.com/help/images/ref/regionprops.html>



**Algoritma 5:** FAST Algoritması (Features From Accelerated Segment Test)  
[Rosten and Drummond, 2006].

**Algoritma 6:** BRISK Algoritması (Binary Robust Invariant Scalable Keypoints)  
[Leutenegger et al., 2011].

**Algoritma 7:** MSER Algoritması (Maximally Stable Extremal Regions)  
[Matas et al., 2002].

**Algoritma 8:** ORB Algoritması (Oriented FAST and Rotated BRIEF)  
[Rublee et al., 2011].

**Algoritma 9:** HARRIS Köşe Bulma Algoritması [Harris and Stephens, 1988].

**Algoritma 10:** Hausdorff Uzaklığı Algoritması: İki işaret karşılaştırılırken işaretler arasındaki belirli bir mesafenin saptanıp küçük olanın alınmasına dayalı bir algoritmadır [Huttenlocher et al., 1993].

**Algoritma 11:** Hough Dönüşüm ile Yapısal Özniteliklerin bulunması  
[Chunhavittayatera et al., 2006].

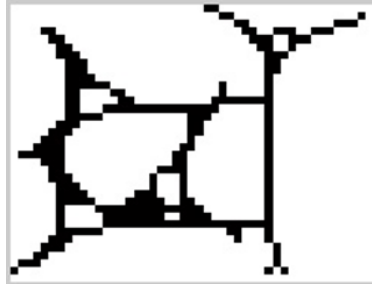
**Algoritma 12:** Hierarchical Centroid (H.C.) Algoritması: Resmin parçalara bölünüp her bir parçanın ağırlık merkezinin öznitelik olarak çıkartılmasıdır [Armon, 2011].

**Algoritma 13:** HOG (Histogram of Oriented Gradients) Algoritması  
[Dalal and Triggs, 2005].

Bu algoritmalar arasından örnek olarak B.U. algoritması adı verilen Algoritma 1 aşağıdaki alt bölümde anlatılmıştır.

### 3.2.1 Algoritma 1: B.U. algoritması

Uygulama için gereken ön işlem için örnek olarak HZL numarası 180 olan çivi yazısı işareti alınmıştır. Bu işaret üzerinde öncelikle aşağıda özetlenen ön işlem uygulanır. Bu ön işlem; tabletlerden ve işaret listesinden alınan RGB niteliğindeki (renkli) işaret resimlerini ikili (siyah beyaz) resim formatına çevirmeyi amaçlar. Öncelikle RGB resimler MATLAB'ın `rgb2gray`<sup>16</sup> fonksiyonuyla gri seviyeye çevrilir. Elde edilen gri seviyeli resim için bir eşik (threshold) değeri belirlenir ve eşik altında kalan değerler 0 yani siyah, eşik üstündeki değerler 1 yani beyaz piksel olacak şekilde ayrıştırılır. Böylece resim ikili (binary) yani siyah-beyaz resme dönüştürülmüş olur. İşaret siyah beyaza dönüştükten sonra MATLAB 'da bulunan `skel`<sup>17</sup> komutuyla iskeleti çıkarılır. Bu işlem resimdeki objelerin sınırlarını küçültürerek, objelerin kopmasına izin vermeden resmin iskeletinin çıkarılması işlemidir. Şekil 3.3'de V.S. işaret listesinden örnek olarak seçilen işaretin iskeleti görülmektedir.



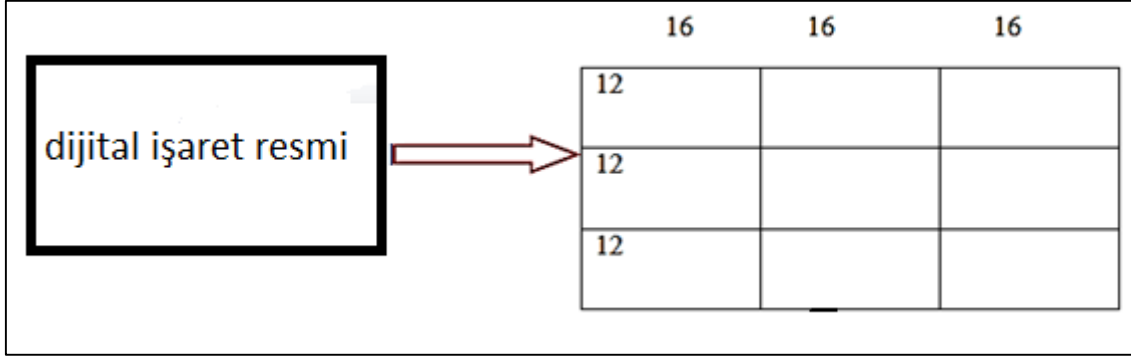
Şekil 3.3 HZL No.180 olan işaretin iskelete dönüştürülmüş hali

Boyutu 36x48 piksele indirilmiş bulunan işaret yukarıda özetlenen ön işlemden geçtikten sonra her biri 12x16'lık çözünürlükte olacak şekilde 9 eşit parçaya bölünür. Bu algorithmada herbir çivi yazısı işareti piksel sayıları  $m$ =satır sayısı,  $n$ =kolon sayısı olmak üzere  $(m \times n)$  ile tanımlanan  $p$  sayıdaki bölgeye ayrılır. Örneğin Şekil 3.4 içinde 36x48 piksel boyutunda olan bir resim için işaretin 9 bölgeye ayrılması anlatılmıştır. Örnekte  $m=12$  ve  $n=16$  piksel sayıları vardır.

---

<sup>16</sup> <http://www.mathworks.com/help/matlab/ref/rgb2gray.html>

<sup>17</sup> <http://www.mathworks.com/help/images/ref/bwmorph.html>



Şekil 3.4 İşaret resminin 9 eşit parçaya bölünmesi

İşaret eşit parçalara bölüldükten sonra her bir parça içindeki siyah piksel sayısı bulunur. Daha sonra V.S. ve tablettten alınan çiviyazısı işaretlerinin birbirine denk gelen parçalarının içindeki siyah piksel sayılarının farkı bulunur. Bu farkların toplamı resimdeki toplam piksel sayısına (36x48) bölünerek bir hata oranı bulunur. Bu hata oranı 0 ile 1 arasında bir değerdir. Bu işlem Denklem 3.1 ile ifade edilmiştir.

$$Hata Oranı 1 = \frac{\sum |Parçaların siyah piksel sayılarının farkı|}{İşaretin toplam piksel sayısı} \quad (3.1)$$

B.U. algortimasında kullanılan ikinci hata oranı hesaplaması aşağıda anlatılmıştır. İşaret eşit parçalara bölünüp her bir parça içindeki siyah piksel sayısının bulunmasından sonra V.S. ve tablettten alınan çiviyazısı işaretlerinin birbirine denk gelen parçalarının içindeki siyah piksel sayılarının farkı bulunur. Daha sonra her bir parçadaki bu fark, parçanın toplam piksel sayısına (12x16) bölünür. Son olarak hesaplanan bu oranların toplamının parça sayısına bölünmesi ile bir hata oranı elde edilir. Bu hata oranı 0 ile 1 arasında bir değerdir Bu işlem Denklem 3.2 ile ifade edilmiştir.

$$Hata Oranı 2 = \frac{\sum \frac{|Parçaların siyah piksel sayılarının farkı|}{Parçanın toplam piksel sayısı}}{Parça sayısı} \quad (3.2)$$

İşaretler okunurken bu algoritma V.S. dijital listesiyle direkt karşılaştırma ölçütü olarak kullanılmamaktadır. Çünkü tablette ve V.S. işaret listesindeki aynı HZL numaralı işaret farklı kaynaklardan alındığı için piksel sayıları teorik olarak birbirine eşit olamamaktadır. Bu algoritma dijital işaret listesinde arama ve karşılaştırma yapılırken

hata oranı belirli bir eşik değerden büyük olan işaretleri arama uzayına dâhil etmeyerek arama uzayını daraltmak için kullanılmaktadır. Ayrıca işlenecek işaret sayısını azaltarak, işaret okumada kullanılan diğer algoritmaların daha kısa sürede çalışmasını sağlamaktadır.

### **3.2.2 Algoritma 2: Regionprops Kütüphanesi**

Bu algoritma tezin veri madenciliği kısmında çiviyazısı işaretlerin özniteliklerini çıkarmak için kullanılmıştır. Algoritmanın detayları tezin 4.2 alt bölümünde verilmiştir.

### **3.2.3 Algoritma 3: SIFT**

SIFT obje tanıma ve bilgisayarlı görü sistemlerinde kullanılan popüler bir algoritmadır. Bu algoritma resimde var olan anahtar noktaları (keypoint) bulup bu noktalar üzerinden betimleyiciler (descriptor) yardımıyla öznitelikleri hesaplamaktadır. Bu algoritmanın en büyük özelliklerinden biri ise bu anahtar noktaları bulurken resmin farklı yönlerde çevrilmesi, boyutunun değişik olması ve resimdeki ışık yoğunluğu durumlarından etkilenmemesidir [Lowe, 2004]. Bu algoritma tez çalışmasında kullanılırken anahtar nokta ve betimleyici olarak SIFT seçilmiştir.

### **3.2.4 Algoritma 4: SURF**

SURF algoritması SIFT algoritması temel alınarak geliştirilen bir algoritmadır. SURF algoritması anahtar nokta bulma ve öznitelik bulma aşamalarında SIFT algoritmasından farklılık göstermektedir. SURF'te anahtar nokta bulunurken Hessian Matrisi yapısı kullanılmıştır bu sayede SIFT algoritmasına göre daha hızlı ve etkili çalışmaktadır [Herber et al., 2006]. Bu algoritma tez çalışmasında kullanılırken anahtar nokta ve betimleyici olarak SURF seçilmiştir.

### **3.2.5 Algoritma 5: FAST**

FAST algoritması gerçek zamanlı sistemlerde SIFT, HARRIS gibi algoritmaların çok işlemci zamanı aldığı gerçeğini ortaya koyarak, gerçek zamanlı sistemler için geliştirilmiş hızlı bir köşe bulma algoritmasıdır. Burada anahtar noktalar köşe noktalarından oluşmaktadır [Rosten and Drummond, 2006]. Bu algoritma tez çalışmasında kullanılırken anahtar nokta bulmada FAST ve betimleyici olarak da ORB seçilmiştir.

### **3.2.6 Algoritma 6: BRISK**

BRISK algoritması SURF algoritmasından işlemci zamanı olarak daha az zaman alan ve SURF'e göre daha iyi performans gösteren bir algoritmadır. Daha az süre almasının sebebi FAST algoritmasında kullanılan anahtar nokta bulma yönteminin kullanılması ve ek olarak her bir anahtar noktasındaki komşuluğunda bulunan piksellerin yoğunluklarının karşılaştırmasında kullanılan bit dizileridir [Leutenegger et al., 2011]. Bu algoritma tez çalışmasında kullanılırken anahtar nokta ve betimleyici olarak BRISK seçilmiştir.

### **3.2.7 Algoritma 7: MSER**

MSER algoritması resimlerde çember veya elips benzeri şekilleri (blobs) bulmak için kullanılan bir algoritmadır. Algoritma anahtar noktaları bu şekilleri dikkate alarak seçer ve öznelikleri bu anahtar noktalar üzerinde hesaplar [Matas et al., 2002]. Bu algoritma tez çalışmasında kullanılırken anahtar nokta bulmada MSER ve betimleyici olarak da ORB seçilmiştir.

### **3.2.8 Algoritma 8: ORB**

ORB algoritması FAST algoritması ve BRIEF algoritmasının hibridi olan bir algoritmadır. Algoritma FAST algoritmasını kullanarak anahtar noktaları bulur ve BRIEF algoritmasını temel alan bir yöntemle bu anahtar noktalarından öznelikleri çıkarır. ORB algoritmasının diğer algoritmalara göre en önemli ve etkin yanı resmin farklı yönlerde çevrilmesinden ve resmin gürültülü olmasından etkilenmeyiştir. Ayrıca bilgisayarlı görü alanında popüler olan SIFT algoritmasından da iki kat daha hızlı çalışmaktadır [Rublee et al., 2011]. Bu algoritma tez çalışmasında kullanılırken anahtar nokta ve betimleyici olarak ORB seçilmiştir.

### **3.2.9 Algoritma 9: Harris**

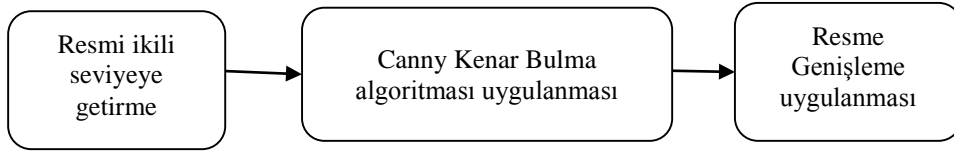
HARRIS algoritması resimde yer alan köşe ve kenar noktalarını bulmak için kullanılan ilk algoritmalarından biridir. Algoritma bir sinyalin üzerindeki farklı yönlerdeki yerel değişiklikleri ölçen yerel otomatik korelasyon (local auto correlation) fonksiyonuna dayanır. Bu algoritmada köşe ve kenar noktaları anahtar noktalarını oluşturmaktadır [Harris and Stephens, 1988]. Bu algoritma tez çalışmasında kullanılırken anahtar nokta bulmada HARRIS ve betimleyici olarak da ORB seçilmiştir.

### **3.2.10 Algoritma 10: Hausdorff uzaklığı**

Hausdorff uzaklığı algoritması tablettten alınan bir çiviyazısı işaretinin V.S. dijital işaret listesindeki karşılığını bulmak için kullanılmaktadır. Çiviyazısı işaretler karşılaştırırken iki işaret resmi arasındaki mesafenin hesaplanması şeklinde çalışan bir algoritmadır. Hausdorff uzaklığı algoritması işaretin karşılığını bulurken tabletteki işaretin dijital işaret listesindeki her bir işaretle olan uzaklığını bulup en küçük uzaklığa sahip işareti aranan işaretin karşılığı olarak kabul etmektedir. Bu algoritma obje eşleme, makine görüşü, imge işleme gibi uygulamalarda sıkça kullanılan bir algoritmadır [Huttenlocher et al., 1993].

Bu algoritma işaretlere uygulanmadan önce birtakım ön işleme aşamalarından geçmiştir. İlk olarak daha önce tezin 3.2.1 alt bölümünde bahsedilen ön işleme

aşamaları uygulanmıştır. Daha sonra resimdeki kenarları daha belirgin hale getirmek için 'Canny Edge Detection' algoritması kullanılmıştır. Canny algoritması resimdeki kenarları ortaya çıkartarak, resmi algoritmalarda kullanılmak için daha etkin bir hale getirir [Mall et al., 2013]. Bu yöntem resimdeki veri miktarını azaltmaya yardımcı olup resmin yapısal bilgisini kaybetmeden kenar bulma işlemini gerçekleştirir. Ayrıca resimdeki gürültü unsurlarını da giderir. Bu işlemden sonra görüntü işlemede kullanılan morfolojik işlemlerden biri olan genişleme (dilation) işlemi kullanılmıştır [Gonzales and Woods, 2002]. Şekil 3.5 ile çiviyazısı işareti resmine uygulanan ön işlem aşamaları verilmiştir.



Şekil 3.5 Hausdorff Uzaklığı algoritmasında uygulanan ön işlem aşamaları.

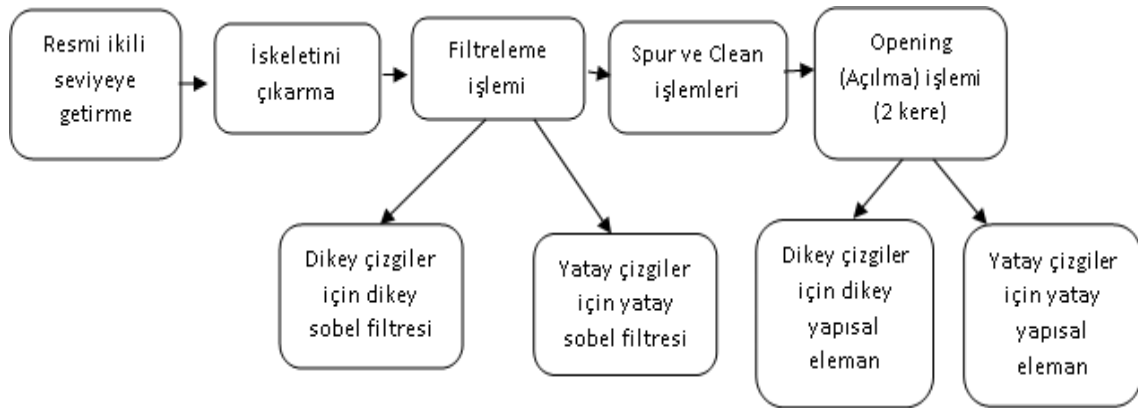
### 3.2.11 Algoritma 11: Hough dönüşüm ile yapısal özniteliklerin bulunması

Bu algoritma Hitit çiviyazısı işaretlerinin yapısal özniteliklerini çıkarımını sağlar. Bu yapısal öznitelikler işaretin sahip olduğu yatay ve dikey çizgi sayısıdır. Bu algoritmanın seçilme sebebi ise Hitit çiviyazısı işaretlerinin çoğunlukla yatay ve dikey çizgiler çizilerek oluşturulmasıdır. İşaretler okunurken bu algoritma V.S. dijital listesiyle direkt karşılaştırma ölçütü olarak kullanılamaz. Çünkü aynı dikey ve yatay çizgi sayısına sahip birden fazla işaret olabilir dolayısıyla bu algoritma dijital işaret listesinde arama ve karşılaştırma yapılırken çizgi sayısı farkı çok büyük olan işaretleri arama uzayına dâhil etmeyerek arama uzayını daraltmak için kullanılmaktadır. Ayrıca işlenecek işaret sayısını azaltarak işaret okumada kullanılan diğer algoritmaların daha kısa sürede çalışmasını sağlamaktadır.

Bu algoritmayı uygulamadan önce işaretler üzerinde tezin 3.2.1 alt bölümü içinde bahsedilen ön işleme aşamaları ile siyah ve beyaz pikseller belirlenir. Daha sonra yine tezin 3.2.1 alt bölümünde bahsedilen resmin iskeletini çıkarma ön işlemi işaretlere uygulanır. Bu algoritmada sadece çizgiler önemli olduğu için resmin

geometrik özelliklerine ihtiyaç vardır, dolayısıyla resimdeki işaretin iskeleti çıkartılıp, hatlarını inceltirerek ve çizgileri ortaya çıkartarak diğer detaylar azaltılıp daha etkili bir şekilde algoritmanın uygulanması sağlanır. Bu ön işlemlerden sonra işaretin yatay ve dikey bileşenlerinin ortaya çıkarmak için işaret resimlerine filtreleme işlemi uygulanır. Burada yatay kenarları belirtmek ve ortaya çıkartmak için yatay Sobel filtresi, dikey kenarları belirtmek ve ortaya çıkartmak için ise dikey Sobel filtresi kullanılmıştır [Hong Nguyen et al., 2014]. Daha sonra işaretlere MATLAB 'da bulunan Spur<sup>18</sup> özelliği uygulanmıştır. Bu özellik objede yani işarete bulunan çıkıntıları yok ederek gürültüyü azaltmaya yardımcı olur. Bu işlemden sonra MATLAB' da bulunan Clean<sup>18</sup> özelliği uygulanmıştır. Bu özellik ise izole olmuş pikselleri ve kopmuş pikselleri yok etmeye yarar. Bu özellik gürültüyü gidermede ve kopuk oluşumların silinmesinde rol oynar. Bu işlemlerden sonra resimlere açılma (opening) morfolojik işlemi uygulanır. Açılma gürültü giderme işlemini gerçekleştirir [Gonzales and Woods, 2002]. Bu işlemler küçük alana sahip alanların yok olmasına ve resmin genel hatlarına ulaşılmasına yardımcı olur.

İşaret resimlerine Hough dönüşüm algoritması uygulanmadan önce uygulanan ön işlem aşamaları Şekil 3.6 ile özetlenmiştir.



Şekil 3.6 Hough dönüşüm algoritmasında uygulanan ön işlem aşamaları

<sup>18</sup> <http://www.mathworks.com/help/images/ref/bwmorph.html>



Ön işlemlerden geçen işaretler Hough dönüşüm tarafından çizgileri tespit edilip yatay ve dikey çizgi sayıları belirlenmektedir. [Chunhavittayatera et al., 2006]. V.S. dijital listesinden ve tablet resimlerinden HZL numarası aynı olacak şekilde alınan Hitit çiviyazısı işaretlerine ön işlem aşamaları sonrası ve Hough dönüşüm algoritması uygulanması sonucu yatay ve dikey çizgilerin bulunması Çizelge 3.1 içinde gösterilmiştir.

Çizelge 3.1 Örnek çiviyazısı işaretlerinin yatay ve dikey çizgilerin bulunması

HZL Numarası	İşaretin resmi	Hough dönüşüm sonucu (dikey çizgilerin bulunması)	Hough dönüşüm sonucu (yatay çizgilerin bulunması)
8 (V.S.)			
8 (Tablet)			
180 (V.S.)			
180 (Tablet)			
218 (V.S.)			
218 (Tablet)			

### 3.2.12 Algoritma 12: H.C. (Hierarchial Centroid) algoritması

H.C. algoritması Hitit çiviyazısı işaret resminin parçalara bölünüp her bir parçanın ağırlık merkezi koordinatlarının öznitelik olarak çıkartılmasıdır [Armon, 2011]. Daha sonra çıkartılan bu öznitelikler tablet resminden alınan bir çiviyazısı işaretin dijital işaret listesindeki karşılığını bulmak için kullanılmaktadır. H.C. algoritması uygulanması sonucu tablettten alınan işaret için oluşturulan öznitelik vektörü dijital işaret listesinde bulunan bütün işaretlerin oluşturduğu öznitelik vektörüyle karşılaştırılmaktadır. Bu karşılaştırma iki ölçütle sağlanmaktadır. İlki iki vektör arasındaki Cosine benzerliği<sup>19</sup> diğeri ise iki vektör arasındaki Öklid Uzaklığıdır<sup>20</sup>. Karşılaştırılan iki işaret birbirine ne kadar benzer olursa Cosine benzerliği değeri o kadar büyük olur. Öklid uzaklığında ise bu durum tam tersidir işaretler ne kadar birbirine benzer ise uzaklık o kadar az olur. Karşılaştırma yapılırken tabletteki işaret ile dijital işaret listesindeki işaretler arasındaki Öklid uzaklığı en az bulunan veya Cosine benzerlik değeri en fazla bulunan işaret aranılan işarettir. Bu algoritmanın Hitit çiviyazısı işaretlerine uygulanmasından önce işaretlere tezin 3.2.10 alt bölümünde bahsedilen ön işlem aşamaları uygulanmıştır.

Algoritmayı gerçekleştiren fonksiyon girdi olarak bir resim alıp çıktı olarak ağırlık merkezinin x koordinatını döndürür (Ağırlık merkezi hesaplanırken beyaz pikseller ağırlık, siyahlar boşluk olarak kabul edilmektedir.). Daha sonra resim x koordinatından iki parçaya bölünür ve bu iki alt parçanın transpozu alınmış hali içinde fonksiyon özyinelemeli olarak çağrılır. Verilen bir derinlik değerine göre parçalara bölme sayısı ve hesaplanan öznitelik vektörünün boyutu değişmektedir. d derinlik değeri olmak üzere öznitelik vektörü  $2d-1$  elemandan oluşmaktadır. Ağırlık merkezlerinin y koordinatlarını elde etmek için ise resmin transpozu alınarak algoritmayı gerçekleştiren fonksiyona girdi olarak verilir. Sonuç olarak çıkan öznitelik vektörü parçaların x koordinatlarını tutan öznitelik vektörü ile birleştirilerek nihai öznitelik vektörünü oluşturur [Armon, 2011]. Bu nihai öznitelik vektörü verilen d derinliği için toplam  $2*(2^d-1)$  elemandan oluşur. Hitit çiviyazısı işaretlerine bu

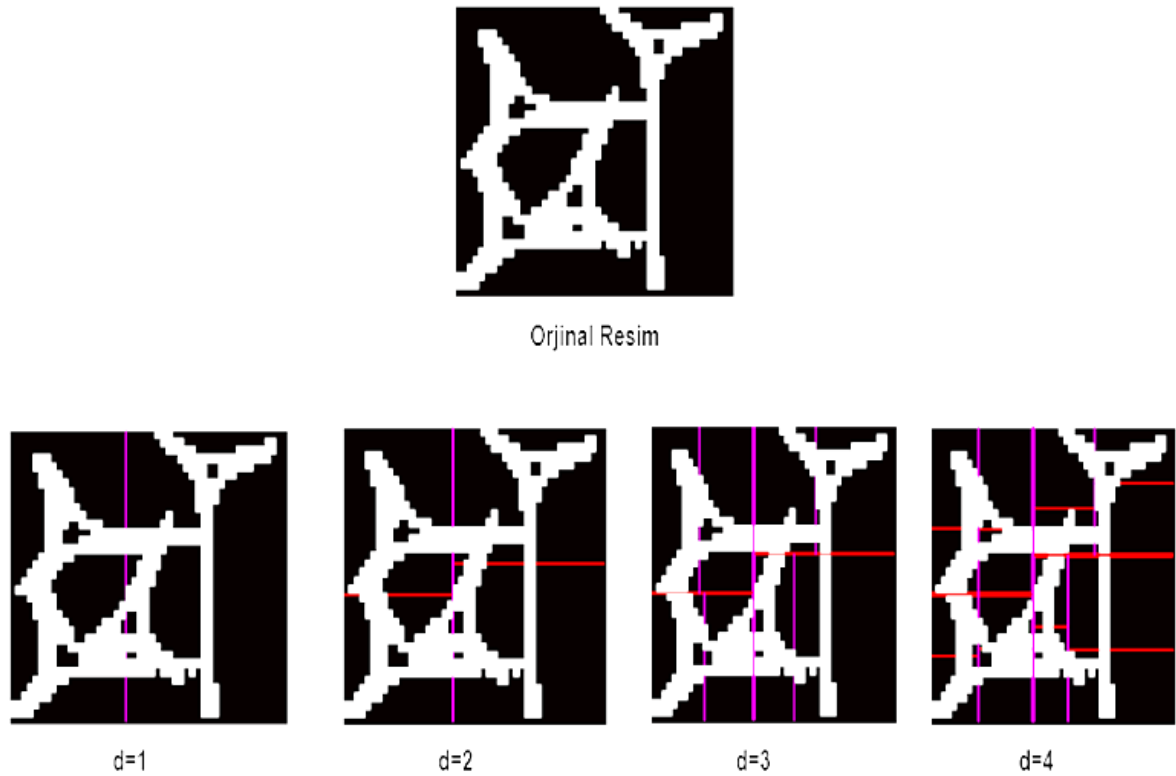
---

<sup>19</sup> [https://en.wikipedia.org/wiki/Cosine\\_similarity](https://en.wikipedia.org/wiki/Cosine_similarity)

<sup>20</sup> [https://en.wikipedia.org/wiki/Euclidean\\_distance](https://en.wikipedia.org/wiki/Euclidean_distance)

algoritmanın uygulanması sırasında derinlik değeri (d) 6 seçilmiştir ve sonuç olarak öznitelik vektörü 126 elemandan oluşmaktadır.

HZL numarası 180 olan ve V.S. dijital işaret listesinde yer alan Hitit çivi yazısı işaretine farklı derinlik (d) değerlerine göre algoritmanın uygulanması Şekil 3.7 ile gösterilmektedir. Bu şekilde görülen çizgiler parçaların ağırlık merkezinden geçen çizgileri ve parçalara ayrılma yerlerini temsil etmektedir.



Şekil 3.7 HZL No.180 olan çivi yazısı işaretinin farklı derinlik değerlerine göre parçalara ayrılması

### 3.12.13 Algoritma 13: HOG algoritması

HOG algoritması Hitit çivi yazısı işaret resminin HOG özniteliklerinin çıkarılmasını sağlayan bir algoritmadır [Dalal and Triggs, 2005]. Daha sonra çıkartılan bu öznitelikler tablet resminden alınan bir çivi yazısı işaretin dijital işaret listesindeki karşılığını bulmak için kullanılmaktadır. HOG algoritması uygulanması sonucu tablettten alınan işaret için oluşturulan öznitelik vektörü, dijital işaret listesinde bulunan

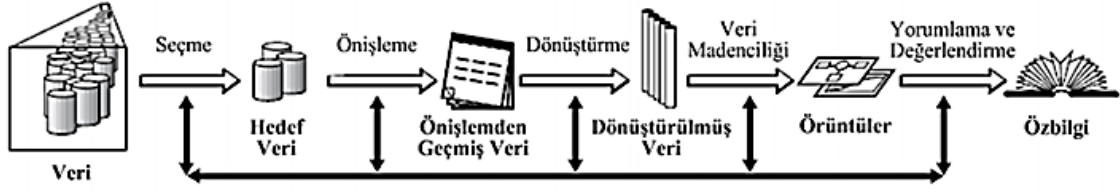
bütün işaretlerin oluşturduğu öznitelik vektörüyle karşılaştırılmaktadır. Tez çalışmasında her biri 20 derece olan 9 tane oryantasyon grubu (bin) ve 27 tane parça (cell) kullanılmıştır. Kullanılan öznitelik vektörü toplam 243 elemandan oluşmaktadır. Karşılaştırma ölçütleri tezin 3.2.12 alt bölümünde bahsedildiği gibidir. Bu algoritmanın Hitit çiviyazısı işaretlerine uygulanmasından önce işaretlere tezin 3.2.10 alt bölümünde bahsedilen ön işlem aşamaları uygulanmıştır.

## 4. HİTİT ÇİVİYAZISI İŞARETLERİ ÜZERİNDE VERİ MADENCİLİĞİ UYGULAMALARI

Hitit çiviyazısı işaretlerinde birbirine benzeyen geometrik şekillere sahip birçok işaret bulunmaktadır. Bu işaretlerin geometrik olarak birbirine benzeyenlerinin aynı kategorilerde toplanabilmesi düşüncesi bu tezin veri madenciliği uygulamaları tarafını yaratmıştır. İşaretlerin geometrik özelliklerinin çıkartılarak veri madenciliğinde popüler olarak kullanılan K-means kümeleme algoritmasıyla kategorilerinin belirlenmesi yani benzer geometrik özelliklere sahip işaretlerin aynı kategorilerde toplanması bu tez çalışmasında gerçekleştirilmiştir. Daha sonra kategorilere ayrılan işaretlere veri madenciliğinde popüler olarak kullanılan sınıflandırma algoritmaları uygulanmış ve sınıflandırma performansları raporlanmıştır.

### 4.1 Veri Madenciliği Nedir?

İnternetin ve bilgisayarın hayatımıza girmesiyle kullanılan veri miktarı gitgide artmaktadır. Bu veri miktarındaki artış verileri işlemekte zorluklar yaratmaktadır. Verileri işleme, maliyetli ve zaman alıcı olmaya başlamıştır. Büyük boyutlu verilerden kullanıcıların veya şirketlerin bütün veriyle uğraşmadan anlam ve bilgi çıkarması ihtiyacı doğmuştur. Veri madenciliği bu ihtiyaç neticesinde doğan bir disiplindir. Veri madenciliği büyük boyutlu verilerden bilgi çıkarma işlemine verilen addır [Han and Kamber, 2006]. Veri madenciliği bankacılık, e-ticaret, sigortacılık, sosyal medya, eğitim, genetik ve finans gibi birçok alanda kullanılmaktadır [Kumar and Bhardwaj, 2011], [Baykal, 2006]. Veri madenciliği veritabanı sistemleri, istatistik, makine öğrenme ve örüntü tanıma gibi birçok alanla yakından ilgilidir. Veri madenciliği bilgi keşfi (Knowledge Discovery (KDD)) adı verilen bir sürecin parçasıdır [Han and Kamber, 2006]. Bilgi keşfi verilerin birleştirilmesi, ön işlemden geçirilmesi, veri boyutunun küçültülmesi, veri madenciliği ve bilginin yorumlanıp değerlendirilmesi gibi süreçlerden oluşan bir yapıdır. Bu aşamalar Şekil 4.1 ile gösterilmektedir [Yeşilbudak vd., 2011].



Şekil 4.1 Bilgi keşfi aşamaları

Veri madenciliğinde veri yığınınından çıkartılacak bilgiye göre farklı özelliklere sahip modeller kullanılmaktadır. Veri madenciliği modelleri tahmin edici (predictive) ve tanımlayıcı (descriptive) olmak üzere iki başlık altında toplanmaktadır. Tahmin edici modellerin temeli, bazı varlıkların gelecekteki davranışlarını tahmin etmek için örüntülerin bulunmasına dayanmaktadır. Tanımlayıcı modellerde ise kullanıcıya insanın anlayabileceği ve yorumlayabileceği şekilde örüntülerin sunulması amaçlanmaktadır [Fayyad et al., 1996].

Tahmin edici modellerde birçok yöntem bulunmaktadır. Bunlardan birisi bu tez çalışmasında kullanılan sınıflandırma (classification) yöntemidir. Sınıflandırma; bir veri sınıfını veya konseptini tanımlayan ve ortaya çıkaran bir model (veya fonksiyon) bulunması ve bu modeli kullanarak sınıfı belli olmayan objelerin sınıflarını tahmin etme işlemidir. Model kural kümesi şeklinde, ağaç yapısında veya matematiksel bir formül olarak gösterilebilir. Çıkarılan model; sınıfı belli olan objeler analiz edilerek yaratılmaktadır [Han and Kamber, 2006]. Bu modeli oluşturan ve sınıfı belli olan objelere eğitim kümesi (training set) adı verilir. Bu model kullanılarak sınıfı tahmin edilecek ve sınıfı belli olmayan objelere ise test kümesi (test set) adı verilir. Eğitim kümesi kullanılarak yaratılan modelin test kümesindeki objeleri doğru sınıflandırma oranına doğruluk (accuracy) adı verilmektedir [Han and Kamber, 2006]. Bir modelin sınıflandırma performansı doğruluk üzerinden değerlendirilmektedir. Doğruluk hesabı karışıklık matrisi (confusion matrix) adında bir yapı vasıtasıyla hesaplanır. Bu matris sınıflandırma sonucu oluşan doğru ve yanlış sınıflandırmaları tutan bir tablodur. Örneğin 2 sınıflı (Pozitif/Negatif) bir sınıflandırma probleminde matris Çizelge 4.1'deki gibi görülmektedir.

Çizelge 4.1 İki sınıf için karışıklık matrisi

Tahmin Edilen Sınıf	Gerçek Sınıf	
	Pozitif	Negatif
Pozitif	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
Negatif	Yanlış Negatif (YN)	Doğru Negatif (DN)

Burada DP ile gösterilen ifade sınıflandırıcının pozitif olarak doğru tahmin ettiği obje sayısıdır. YP ifadesi sınıflandırıcının pozitif olarak tahmin ettiği ama negatif sınıfa ait olan obje sayısıdır. YN, sınıflandırıcının negatif olarak tahmin ettiği ama aslında pozitif sınıfa ait olan objelerin sayısıdır. Son olarak DN ifadesi sınıflandırıcının negatif olarak doğru tahmin ettiği obje sayısıdır. Bu matris sınıf sayısı arttıkça daha büyük boyutlara ulaşır. Doğruluk, doğru olarak tahmin edilip sınıflandırılmış obje sayısının toplam sınıflandırılan obje sayısına bölümü ile bulunur ve (4.1) numaralı formül ile ifade edilir [Zaki and Meira Jr., 2014].

$$\text{Doğruluk} = (DP + DN) / (DP + YP + YN + DN) \quad (4.1)$$

Tez çalışmasında kullanılan ve tanımlayıcı modellerden biri olan Kümeleme (Clustering), sınıflandırmanın aksine objeleri sınıf etiketi olmadan analiz edebilmeye olanak sağlar. Kümeleme, veriler arasında birtakım benzerlik ve farklılıklara bakılarak gruplandırma yapılması prensibine dayanmaktadır [Ahamed and Hareesha, 2012]. Çoğu durumda ilk başta sınıf etiketli veri bulunmamaktadır. Kümeleme sınıf etiketi olmayan veriler için sınıf etiketleri yaratır. Kümelemede objeler; sınıflar arası (interclass) benzerlik minimum ve sınıf içi (intraclass) benzerlik maksimum olacak şekilde gruplanır veya kümelenir [Han and Kamber, 2006]. Sınıf etiketi atamaları yapıldıktan sonra sınıflama gibi tahmin edici modeller verilere uygulanabilir.

## 4.2 Regionprops Kütüphanesi

Bu algorithmada MATLAB yazılım geliştirme ortamı içindeki Regionprops kütüphanesi kullanılmıştır. Bu kütüphane verilen bir resmin bölgesel özniteliklerini hesaplamaya yardımcı olur. Regionprops, verilen bir resmi bir obje kümesi olarak görür ve bu obje kümesi üstünden objelerin özelliklerine ulaşabilmeyi sağlar. Bu hesaplamalar sadece siyah-beyaz resimlerde uygulanabileceği için daha önce 3.2.1 alt bölümü içinde yer

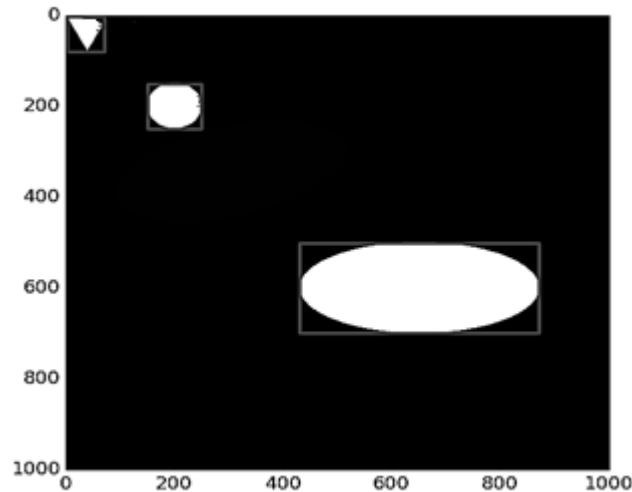
alan Algoritma 1’de uygulanan siyah-beyaza çevirme ön işlem aşamasını uygulamak gereklidir. Ayrıca resimde bulunan objelerden bu öznitelikleri çıkarmak için resmin siyah piksellerini beyaz, beyaz pikselleri ise siyaha dönüştürmek gerekir çünkü MATLAB yazılımı objeleri resimde beyaz piksellere sahip alanlar olarak değerlendirir.

Regionprops kütüphanesinin içerisinde resmin bölgesel özniteliklerini ölçmeye yarayan birden fazla öznitelik bulunur. Hitit çiviyazısı işaretlerinin okunmasında kullanılan öznitelikler: Area, Bounding Box, Centroid, Euler Number, Extent, Eccentricity, EquivDiameter’dır.

Bu algorithmada kullanılan öznitelikler ve açıklamaları aşağıda verilmiştir;

**Area:** Objeye ait bölgedeki toplam piksel sayısı.

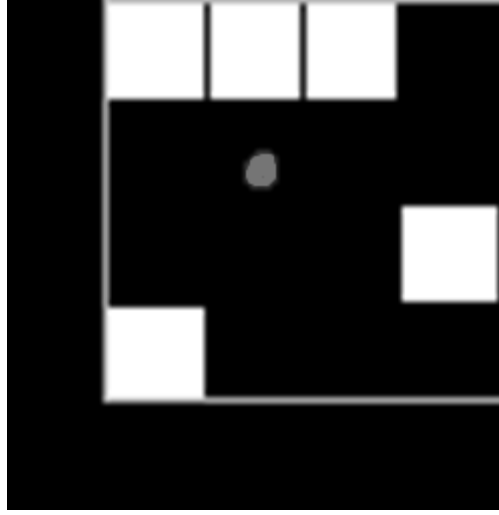
**Bounding Box:** Bölgenin ve objenin tümünü çerçeveleyen en küçük kutudur. Bölgenin beyaz pikselleri kutuya degecek şekilde oluşturulur. Bir vektör yapısında olup, bu vektörde kutunun x-y düzlemindeki sol üst köşesinin koordinatları ve her bir düzlemdeki boyutu bulunur. Çeşitli şekillerin bounding box örnekleri Şekil 4.2 ile gösterilmektedir.



Şekil 4.2 Bir Bounding Box örneği



**Centroid:** 2 elemanlı bir vektör olup, bölgenin ağırlık merkezinin x-y koordinatlarını içerir. Burada ağırlık merkezi hesaplanırken beyaz pikseller ağırlık, siyahlar boşluk olarak düşünülür. Örnek bir Centroid gösterimi Şekil 4.3 ile verilmiştir. Burada işaretli nokta Bounding Box içindeki bölgenin ağırlık merkezini göstermektedir.



Şekil 4.3 Bölgedeki objelerin ağırlık merkezi

**Euler Number:** Bölgede bulunan objelerin sayısından objelerin içindeki boşlukların sayısının çıkarılmasıyla elde edilen değerdir.

**Extent:** Bounding box'ta bulunan toplam piksel sayısının bölgedeki piksel sayısına oranını verir. Bölgenin alanının bounding box alanına bölünmesiyle hesaplanır. Skalar bir değerdir.

**Eccentricity:** Objeye ait bölgenin aynı ikinci momentlere sahip elipsin dış merkezi skalar değerini veren bir özneliktir. Odaklar arasındaki uzaklığın asal eksen uzunluğuna oranlanması ile elde edilen (0 ile 1 arasında) değerdir.

**EquivDiameter:** Objenin alanının 4 ile çarpılıp pi sayısına bölündükten sonra karekökünün alınmasıyla hesaplanır (4.2) .

$$EquivDiameter = \sqrt{4 * Alan / \pi} \quad (4.2)$$

### **4.3 Hitit Çiviyazısı İşaretlerinden Veri Kümesinin Oluşturulması**

Hitit çiviyazısı işaretleri veri madenciliği uygulamalarında kullanılırken işaretlerin geometrik öznitelikleri hesaplanarak veri kümesi oluşturulmuştur. Oluşturulan veri kümesi V.S. dijital işaret listesinden alınan işaretlerden oluşmaktadır. İşaretlerin dijital imajının alınması aşaması tezin 3.1 alt bölümünde anlatıldığı gibidir. Geometrik öznitelikler tezin 4.2 alt bölümünde bahsedilen MATLAB Regionprops Kütüphanesi kullanılarak hesaplanmıştır. Bu geometrik öznitelikler Area, Centroid X koordinatı, Centroid Y koordinatı, Euler number, Extent, Eccentricity ve EquivDiameter öğelerinden oluşmaktadır. Bu öznitelikler veri madenciliği algoritmalarında kullanılacak bütün çiviyazısı işaretleri için hesaplanmış ve toplamda her bir işaret için 7 öznitelik içeren bir veri kümesi oluşturulmuştur.

### **4.4 Hitit Çiviyazısı İşaretleri Üzerinde Kullanılabilecek Veri Madenciliği Algoritmaları**

#### **4.4.1 K-means kümeleme algoritması**

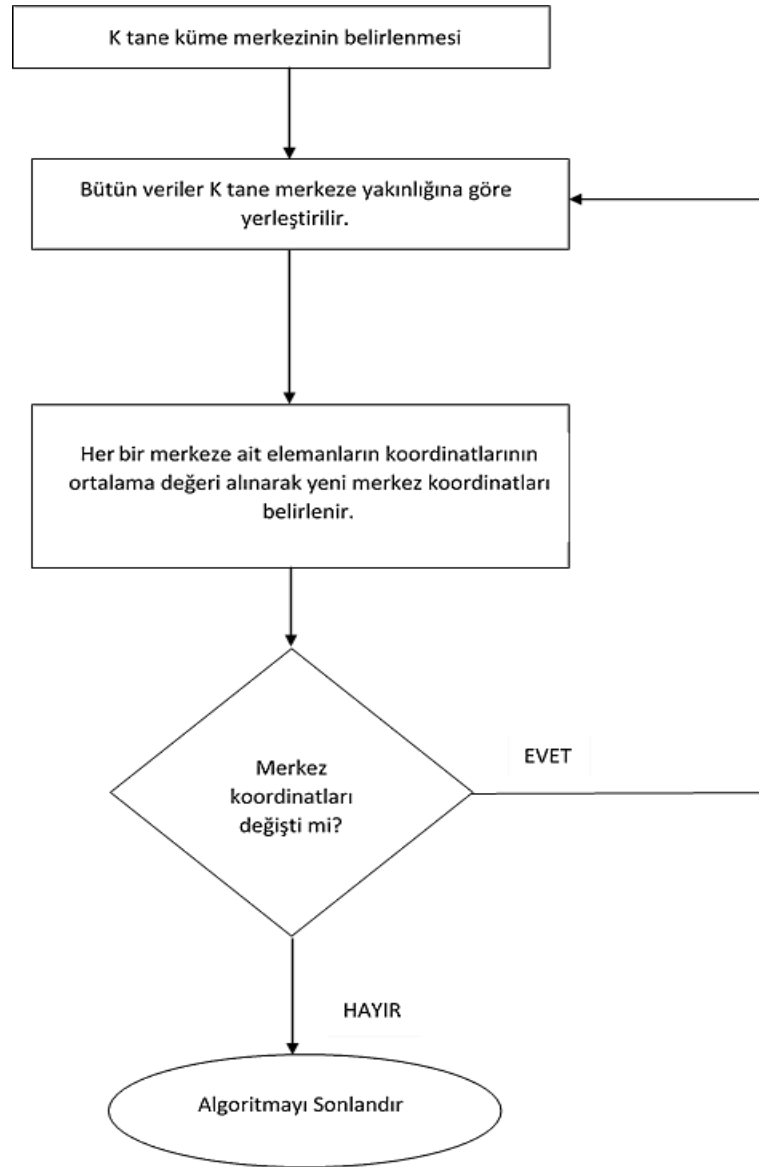
K-means algoritması veri madenciliğinde sınıfı bilinmeyen verilere sınıf atamak için kullanılan ve tanımlayıcı model yapısına sahip bir kümeleme yöntemidir. Yöntem en çok kullanılan kümeleme algoritmalarından biridir çünkü kolayca gerçekleştirilebilir ve çok fazla işlemci çalışma zamanı almamaktadır [Ahamed and Hareesha, 2012].

K-means'de ki temel amaç sınıf etiketi var olmayan verileri, verilerin sahip olduğu öznitelikler arasındaki yakınlığa göre K tane sınıfa bölmektir. Algoritma verileri bir öznitelik uzayına yerleştirir ve o uzay üstünden kümelemeyi gerçekleştirir. Algoritmanın çalışma aşamaları aşağıda özetlenmiştir [Han and Kamber 2006]:

- 1) Başlangıçta K tane küme merkezi koordinatı rastgele belirlenir;
- 2) Bütün veriler bu K tane merkeze yakınlığına göre yerleştirilir. Burada yakınlık küme merkezi ile veri arasındaki Öklid mesafesinin hesaplanmasıyla bulunur;

- 3) Daha sonra her bir sınıfa ait elemanların koordinatlarının ortalama değeri alınarak yeni sınıf merkez koordinatları belirlenir;
- 4) Bütün veriler tekrar yeni sınıf merkez koordinatlarına yakınlığına göre yerleştirilir;
- 5) Bu işlemler K tane sınıf için merkez koordinatlarında değişim olmamasına kadar veya belirli bir iterasyon sayısına ulaşılan kadar devam eder.

K-means algoritması akış diyagramı adımları Şekil 4.4 ile gösterilmektedir.

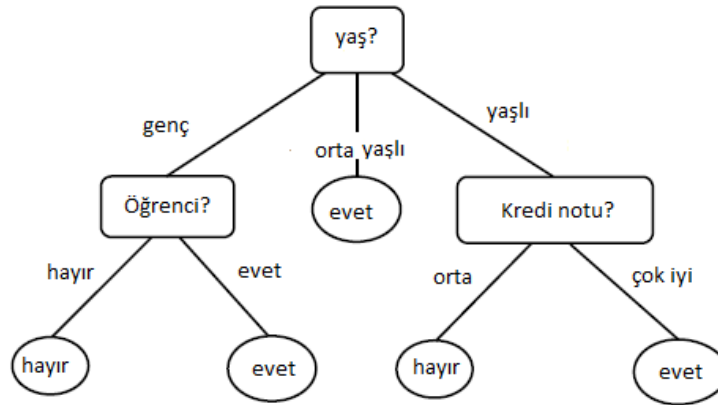


Şekil 4.4 K-Means algoritması akış diyagramı

#### 4.4.2 J48 karar ağacı sınıflandırma algoritması

J48 algoritması sınıflandırma için kullanılan bir karar ağacı (decision tree) algoritmasıdır. Karar ağaçlarında sınıflandırma yapılırken sınıflandırma modeli bir ağaç yapısı üzerinden oluşturulur. Karar ağaçlarının sınıflandırmada sıkça kullanılmasının birçok avantajı vardır; son kullanıcı tarafından modelin kolay anlaşılması, çok büyük veri kümelerini az bir çaba ile çok yüksek performansla sınıflandırabilmesi bu avantajlardan birkaçıdır [Bhargava et al., 2013]. Karar ağaçları, akış diyagramına benzeyen bir ağaç yapısına sahiptirler. Ağacın en üstteki düğümüne kök düğümü (root node) adı verilir. Ağaçtaki her bir iç düğüm (internal node) bir nitelik (attribute) testini göstermektedir. Her bir dal ise bu test sonucunda oluşabilecek değerleri göstermektedir. Son olarak da her bir yaprak düğüm (leaf node) ise bir sınıf değerini tutmaktadır [Han and Kamber, 2006].

Örnek olarak bir kişinin bilgisayar satın alma durumunu modelleyen bir karar ağacı Şekil 4.5 ile gösterilmektedir [Han and Kamber, 2006]. Burada bilgisayar satın alma sınıfı (evet, hayır) ile birlikte üç nitelik (attribute), yaş, öğrenci ve kredi notu kullanılarak bir karar ağacı modeli oluşturulmuştur. Kutu şekliyle gösterilen ifadeler nitelik testini temsil etmektedir. Dallarda bulunan ifadeler bu testler sonucu oluşabilecek değerlerdir. Yuvarlak şekille gösterilen ifadeler de sınıf değerini göstermektedir.



Şekil 4.5 Örnek bir karar ağacı

J48 karar ağacı, Quinlan [Quinlan, 1993] tarafından geliştirilen C4.5 karar ağacı algoritmasının Weka<sup>21</sup> veri madenciliği aracı için uyarlanmış halidir [Sharma and Sahni, 2011]. J48 sınıf etiketi var olan eğitim kümesi üzerinde bilgi entropisini [Han and Kamber, 2006] kullanarak karar ağacını oluşturur ve karar verme mekanizmasını verileri küçük parçalara bölerek gerçekleştirir [Thangalakshmi and Kamalesh, 2014].

#### 4.4.3 K-Nearest Neighbor sınıflandırma algoritması

K-Nearest Neighbor (K-NN) algoritması ilk olarak Cover and Hart [Cover and Hart, 1967] tarafından ortaya atılmıştır. K-NN algoritması günümüzde veri madenciliği, istatistiksel örüntü tanıma gibi birçok alanda kullanılan bir sınıflandırma algoritmasıdır<sup>22</sup>. Algoritmanın yaygın olarak birçok alanda kullanılmasının nedenleri arasında gürültülü verilerde iyi sonuç vermesi ve sınıflandırma modelinin hızlı oluşturulması söylenebilir [Bhatia, 2010]. Algoritma “En yakındaki komşuluğa göre sınıflandırma yap” prensibine göre çalışmaktadır. Algoritma sınıflandırma yaparken eğitim kümesinde ve test kümesinde var olan verileri öznitelik uzayına taşır [Suguna and Thanushkodi, 2010]. K-NN sınıflandırma algoritması test kümesine ait bir örneği sınıflandırırken örneğe en yakın K tane komşuya bakıp bir oylama yöntemiyle sınıf atamasını gerçekleştirir [Suguna and Thanushkodi, 2010].

Algoritmanın çalışma adımları aşağıdaki gibidir<sup>22</sup> :

- 1) K komşuluk sayısı belirlenir;
- 2) Sınıfı bilinmeyen obje ile eğitim kümesindeki bütün objeler arasındaki mesafe hesaplanır. Bu hesaplama iki vektör arasındaki uzaklığın hesaplanması şeklindedir. Genellikle bu hesaplamada Öklid uzaklığı yöntemi kullanılır;

---

<sup>21</sup> [www.cs.waikato.ac.nz/ml/weka/](http://www.cs.waikato.ac.nz/ml/weka/)

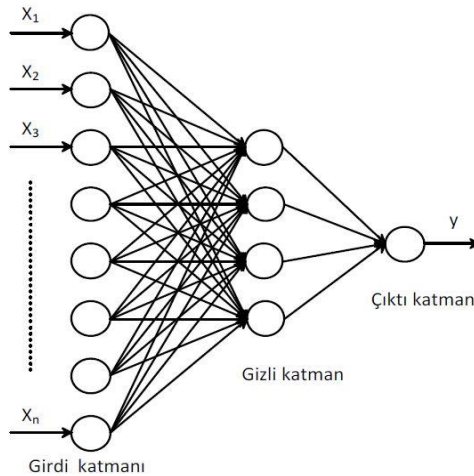
<sup>22</sup> <http://www.codeproject.com/Articles/32970/K-Nearest-Neighbor-Algorithm-Implementation-and-Ov>

3) Sınıfı bilinmeyen obje ile eğitim kümesinde bulunan her bir obje ile arasındaki mesafe en düşük olacak şekilde K tane eğitim kümesi objesi seçilir;

4) Bu K tane en yakın komşudan çoğunluk oylaması yöntemiyle yeni gelen objenin sınıfı belirlenir.

#### 4.4.4 Yapay sinir ağı (YSA) sınıflandırma algoritması

Yapay sinir ağları (YSA) son yıllarda popülerliğini arttırarak finans, tıp, mühendislik, jeoloji ve fizik gibi birçok alanda kullanılmaktadır [Pradhan and Lee, 2007], [Celik and Karatepe, 2007]. YSA; insanın en önemli özeliği olan öğrenmeyi, var olan bilgileri yorumlayıp sonuçlar çıkarabilmeyi temel alan ve bu işlemi otomatik olarak gerçekleştirmek için geliştirilmiş yapılardır. YSA, insan beyninden esinlenerek karar verme ve öğrenme süreçlerini matematiksel bir modele uyarlamayı amaçlar<sup>23</sup>. Yapay sinir ağları düğümlerden ve bu düğümleri birbirine bağlayan bilgi akışını sağlayan bağlantılardan oluşur. YSA'nın yapısı 3 katmandan oluşur. Bunlar Girdi, Çıktı ve Gizli katmanlardır. Girdi katmanına ait her bir düğüm gizli katmana bağlıdır ve gizli katmandaki her bir düğümün çıktı katmanı ile bağlantısı vardır<sup>24</sup>. Bu yapı Şekil 4.6 ile gösterilmektedir<sup>25</sup>.



Şekil 4.6 Bir yapay sinir ağının yapısı

<sup>23</sup> <http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.HaftaYapaySinirAglari.pdf>

<sup>24</sup> <http://www.dataminingmasters.com/uploads/studentProjects/NeuralNetworks.pdf>

<sup>25</sup> [http://3.bp.blogspot.com/-pCd6t1smidg/UF\\_9C43gfl/AAAAAAAAAn8/a8yIS3v98A0/s1600/YSA.jpg](http://3.bp.blogspot.com/-pCd6t1smidg/UF_9C43gfl/AAAAAAAAAn8/a8yIS3v98A0/s1600/YSA.jpg)

Her düğüm diğer düğümlerden aldığı ağırlıklı girdilere yanıt veren basit bir işlem elemanıdır [Pradhan and Lee, 2007]. YSA yapısında düğümlerin giriş ve çıkış değerleri mevcuttur, düğümdeki çıkış değerleri sigmoid fonksiyon adında bir fonksiyon ile hesaplanır. Her bir bağlantının bir ağırlığı mevcuttur. Girdi katmanı ağa giren ham verileri temsil etmektedir. Gizli katman girdi katmanından verileri alarak bir ağırlık değerine göre çıktı katmanına yönlendirir. Ağırlık ilk başta rastgele atanır ve ağ eğitildikçe ağırlıklar güncellenir<sup>26</sup>.

YSA sınıflandırmada kullanıldığı için bir model oluşturarak sınıflandırmayı gerçekleştirmektedir. Bu modeli de eğitim kümesinden öğrenerek oluşturmaktadır. Ağın eğitimi esnasında hem ağa gelen girdiler hem de o girdi değerlerine karşılık ağın üretmesi gereken çıktı değerleri ağa gösterilmektedir. Bu işlemin amacı ağın ürettiği çıktılar ile tahmin edilen çıktılar arasındaki hata oranının ağın ağırlıklarını güncelleyerek azaltılmasını sağlamaktır<sup>27</sup>. Öğrenme, üretilen çıktı ile beklenen çıktı arasındaki hatanın geriye dönüş yapıp ağın ağırlıklarının her geri dönüşte güncellenmesiyle azaltılmasını amaçlıyorsa işleme geriye doğru hesaplamalı öğrenme (back propagation learning) adı verilmektedir.

---

<sup>26</sup> <http://www.dataminingmasters.com/uploads/studentProjects/NeuralNetworks.pdf>

<sup>27</sup> <http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.HaftaYapaySinirAglari.pdf>

## 5. BULGULAR VE TARTIŞMALAR

Hitit çiviyazısında Hititçe anlamı olan işaretler ile birlikte Sümerce ve Akadça anlamları olan işaretler de bulunmaktadır. Bu tez çalışmasında Hitit çiviyazısı işaretlerinden seçilen ve sadece Hititçe anlamları olan 149 adet işaret kullanılmıştır. İşaretler V.S. işaret listesinden tezin 3.1 kısmında bahsedildiği şekilde dijital imajları alınarak çalışmalarda kullanılmıştır. Aynı şekilde işaretlerin tabletlerde bulunan karşılıkları da tezin 3.1 kısmında bahsedildiği şekilde Portal Mainz internet sayfasında bulunan çeşitli tablet resimlerinden dijital imajları alınarak kullanılmıştır.

Tabletten alınan bir işaretin okunup, V.S. dijital işaret listesi karşılığının bulunması için HOG, Hausdorff Uzaklığı ve H.C.(Hierarchical Centroid) algoritması kullanılmıştır. Bu algoritmalar tabletten alınan işaretin, V.S. dijital işaret listesindeki karşılığını bulurken dijital listedeki tüm işaretlerle karşılaştırma yapmaktadır. Daha sonra bu karşılaştırmalara göre aranılan işareti sorgu sonucunda getirmektedir. Algoritmaların işaret okuma performanslarını karşılaştırmak için tabletlerden alınan 149 işaretin dijital işaret listesindeki karşılıklarının bulunma sayısı dikkate alınmıştır. Algoritmaların işaretlerin karşılığını bulma performansları Çizelge 5.1 içinde verilmiştir.

Çizelge 5.1. Algoritmaların işaretlerin karşılığını bulma performansları

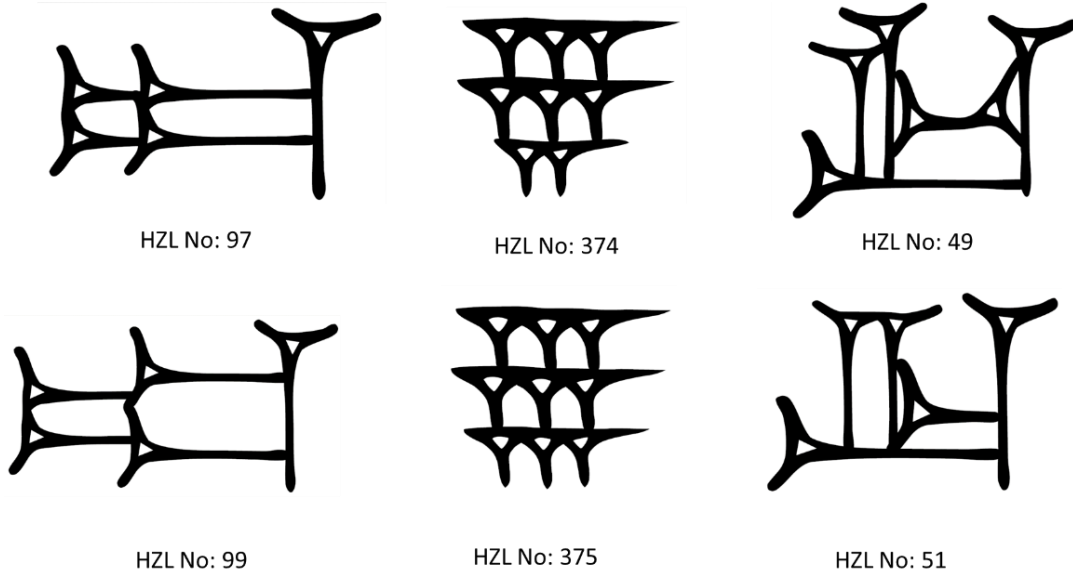
Algoritma Adı	HOG (Cosine)	HOG (Öklid)	Hausdorff Uzaklığı	H.C. (Cosine)	H.C. (Öklid)
<b>Karşılığı Bulunan İşaret Sayısı (149 işaret üzerinden)</b>	46	42	41	33	27
<b>İşaretlerin Karşılığının Bulunma Yüzdesi</b>	32%	28%	27%	22%	18%

Çizelge 5.1 'de görüldüğü üzere en fazla işaretin karşılığını bulan algoritma HOG (Cosine benzerliği ölçütü) algoritmasıdır. Algoritma tabletlerden alınan 149 işaret resminin 46 tanesinin V.S. dijital işaret listesinde aynı HZL numarasıyla indekslenen



karşılıklarını bulmuştur. Karşılığı bulunan işaret sayısına bakıldığında en kötü işaret okuma performansı H.C. (Öklid) algoritmasına aittir, algoritma tabletlerden alınan 149 işaret resminden 27 tanesinin V.S dijital işaret listesindeki karşılıklarını bulmuştur. Çizelge 5.1 'e bakılarak çıkarılabilecek bir diğer bulgu ise işaretlerin karşılaştırılmasında kullanılan benzerlik ölçütlerinden Cosine benzerlik ölçütünün Öklid uzaklığı benzerlik ölçütüne göre algoritmalarda daha çok işaret karşılığının bulunmasını sağlamıştır.

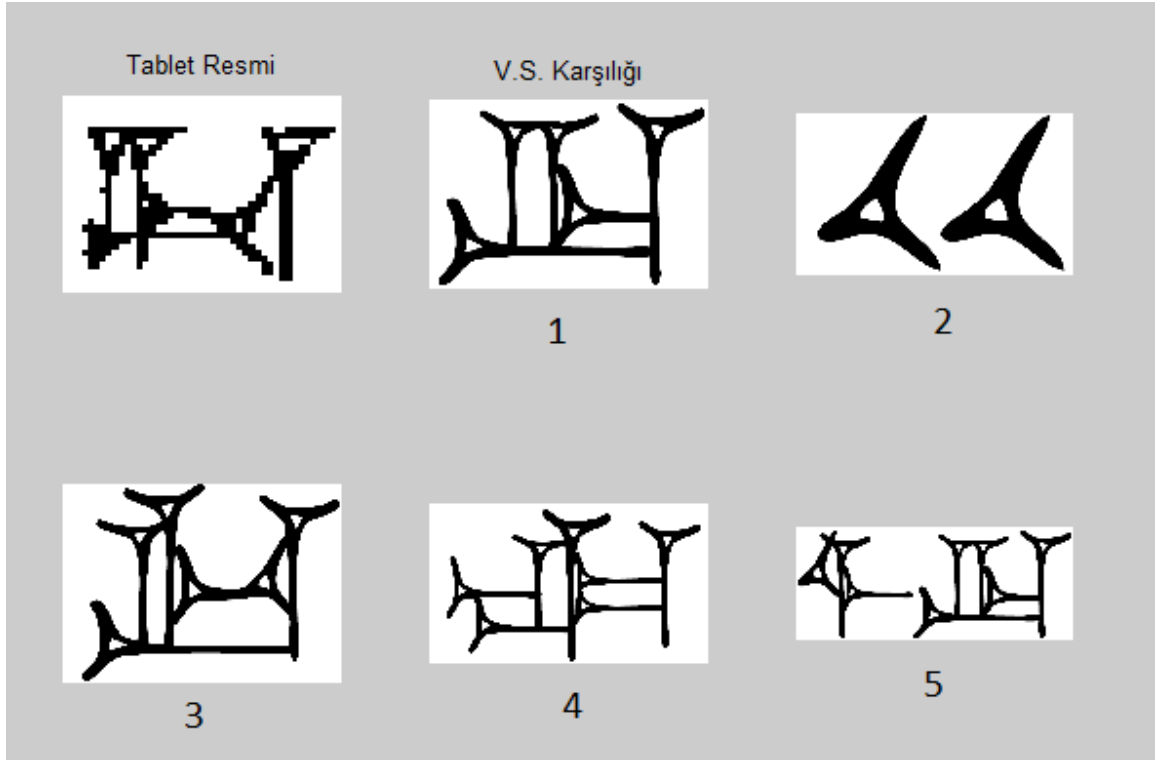
Hitit çivi yazısı işaret listesinde birbirine benzer işaretler bulunmaktadır. Bu benzer işaretler tabletlerde aranan işaretlerin karşılıklarının bulunmasında zorluklara yol açmaktadır. Bu durum ise işaret okumayı gerçekleştiren algoritmaların işaretlerin karşılıklarını bulma performansını etkilemektedir. Bahsedilen duruma neden olan V.S. dijital işaret listesindeki bazı işaretler Şekil 5.1 içinde gösterilmiştir. (Dikkate değer bir nokta HZL No. 97 ile 99 arasındaki farktır: çizgi boyları ve aralıkları arasında fark vardır.)



Şekil 5.1 Birbirine benzeyen örnek işaretler

Bahsedilen durum sebebiyle tablet üzerindeki işaretin dijital listedeki karşılığı olan işaret, karşılaştırmada ilk sırada (Öklid ve Hausdorff uzaklık için en düşük uzaklık değeri, Cosine için en büyük benzerlik değeri) çıkmayarak eşleşmesi bulunamamış

gözükmektedir. Hâlbuki aranılan işaret sorguda ikinci, üçüncü veya dördüncü sırada bulunmaktadır. Anlatılan duruma örnek bir sorgu Şekil 5.2 ile gösterilmektedir. Sorgu HOG (Cosine) algoritması kullanılarak gerçekleştirilmiş ve HZL numarası 49 olan tabletteki işaretin V.S. karşılığı aynı HZL numaralı işaret olması beklenirken benzer işaret olan 51 numaralı işaret sorguda ilk sırada getirilmiştir. İşaretin asıl karşılığı ise 3. sırada getirilmiştir.



Şekil 5.2 HZL No. 49 olan işaret için sorgu örneği

Şekil 5.2 'de görülen durumlardan dolayı algoritmaların aranılan işaret için geri getirim performanslarını incelemek de gerekmektedir. Algoritmaların geri getirim performansını incelemek için bir puanlama sistemi oluşturulmuştur. Bu sistemde tablette karşılığı aranan işaretin V.S dijital listesi karşılığı ilk sırada geri getiriliyor ise algoritma 5 puan, ikinci sırada geri getiriyorsa 4 puan, üçüncü sırada 3 puan, dördüncü sırada 2 puan ve beşinci sırada 1 puan kazanmaktadır. Bir algoritmanın 149 işareti de ilk sırada getirerek alabileceği toplam puan 745 'tir. Puanlar 3 algoritma için 149 işaret üzerinden hesaplanmıştır sonuçlar Çizelge 5.2 'de görülmektedir.

Çizelge 5.2 Algoritmaların Geri Getirim Puanları

Algoritma Adı	HOG (Cosine)	HOG (Öklid)	Hausdorff Distance	H.C. (Cosine)	H.C. (Öklid)
<b>Puan (745 üzerinden)</b>	350	350	317	252	240

Çizelge 5.2 'de görüldüğü üzere HOG algoritmasının Öklid ve Cosine karşılaştırma ölçütleri kullanılan versiyonları aynı puanı almıştır. HOG (Öklid) algoritmasının işaretin direkt karşılığını bulma performansı HOG (Cosine) e göre daha düşük olmasına rağmen geri getirim performansları eşit çıkmıştır. Algoritmaların karşılığı aranan işaretleri ilk sırada getirdikleri durumlar puan hesaplamasında çıkarılırsa, HOG (Cosine) algoritması  $350-5*46=120$  puan ve HOG (Öklid) algoritması  $350-5*42=140$  puan almaktadır. Yani HOG (Öklid) algoritması tablette karşılığı aranan işaretlerin V.S. liste karşılıklarını çoğunlukla ilk 5 sırada getirerek algoritmanın geri getirim safhasında HOG (cosine) algoritmasına yetişmiştir.

İşaretlerin karşılığını bulmak için kullanılan algoritmaların çalışma zamanları tablet üzerinde işlenecek işaret sayısı büyüdükçe önem kazanmaktadır. Bir Hitit çiviyazısı tablette tabletin büyüklüğüne göre yaklaşık 400 işaret bulunabilir. Algoritmaların bu tabletlerdeki işaretleri işlemesi çok işlemci zamanı alabilir. İşaretlerin karşılığını bulan algoritmaların tabletteki bir işaretin V.S. dijital işaret listesindeki karşılığını bulma süreleri saniye cinsinden Çizelge 5.3 ile gösterilmektedir.

Çizelge 5.3 Algoritmaların çalışma süreleri

Algoritma Adı	HOG (Cosine)	HOG (Öklid)	Hausdorff Distance	H.C. (Cosine)	H.C. (Öklid)
<b>Çalışma Süresi (saniye)</b>	3.77	3.35	2.6	3.41	3.25

Çizelge 5.3 'deki sonuçlardan görüldüğü üzere en iyi karşılaştırma performansını gösteren HOG (cosine) algoritması en fazla işlemci zamanı alan algoritmadır. Tek bir işaretin karşılığını 3.77 saniyede bulmaktadır, bu algoritma örneğin 400 işaretli bir













tablete uygulandığında tableti çözmek için geçen süre 1508 saniye yani yaklaşık 25 dakika olacaktır. Bu tableti çözmek için gereken süreyi azaltmak için iki algoritma önerilmektedir. Bunlar Algoritma 1 (B.U. algoritması) ve Algoritma 11 (Hough Dönüşümüyle yapısal özniteliklerin bulunması)'dir. Bu algoritmalar, karşılaştırma algoritmaları çalıştırılmadan önce bir ön eleme işlemi gibi çalışarak her bir işaret için algoritmalarındaki karşılaştırma sayısını azaltmaktadır. Karşılaştırma sayısını azaltmak algoritmaların çalışma süresini kısaltarak süreci hızlandırmaktadır.

Karşılaştırma sayısını azaltacak olan B.U. algoritması, tableten ve V.S. işaret listesinden alınan iki resim arasında hata oranı belirlenen bir eşikten büyük olan işaretleri aramaya dâhil etmeyerek arama uzayını daraltır ve dolayısıyla işlenecek işaret sayısı azaltılmış olur.

Hough dönüşüm algoritmasında ise tableten alınan işaretin yatay ve dikey çizgi sayısı belirlenir, yatay çizgi sayısının dikey çizgi sayısına olan oranı hesaplanır daha sonra aynı şekilde V.S dijital listesindeki işaretin de yatay ve dikey çizgi sayısı oranı hesaplanır. Bu iki oran arasındaki fark belirlenen bir eşğin üstünde ise işaret arama uzayına dâhil edilmez. Eğer bir işaretin dikey veya yatay çizgi bileşenlerinden biri yoksa (örneğin; 1 numaralı işaret tek yatay çizgiye sahiptir.) fark, var olan çizgi sayısı üzerinden hesaplanır.

Çalışmada B.U. algoritması için hata oranı eşik değeri 0.1 seçilmiştir, Hough algoritmasında ise fark eşik değeri 1 seçilmiştir. Bu eşik değerler belirlendikten sonra seçilen örnek işaretlere bahsedilen 2 ön eleme algoritması uygulanmış ve bu ön eleme işlemlerinden sonra işaretin karşılığını bulan algoritmaların toplam yapacakları karşılaştırma sayısı Çizelge 5.4 ile verilmiştir.

Çizelge 5.4 Ön eleme algoritmaları uygulanan örnek işaretler

HZL. No	İşaret Resmi (Tablet)	İşaret Resmi (V.S.)	B.U. Hata Oranı 1 (karşılaştırma sayısı)	B.U. Hata Oranı 2 (karşılaştırma sayısı)	Hough Dönüşüm (karşılaştırma sayısı)
218			108	44	128
375			116	104	124
371			129	120	115
20			46	110	82
364			53	45	121
358			82	38	124

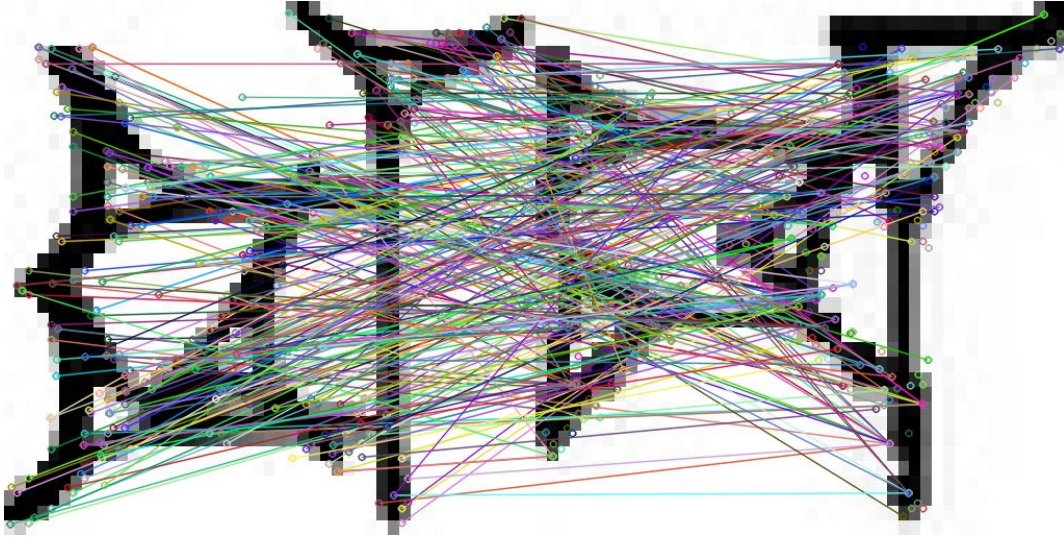
Çizelge 5.4'deki sonuçlara bakıldığında B.U. algoritması basit görünümlü olan 20, 358 ve 364 numaralı işaretlerde daha az karşılaştırma yapılacak işaret bulmuştur ve arama uzayını daha çok daraltmıştır, bu durumun nedeni çiviyazısı işaretlerinde basit ve karmaşık görünümlü işaretlerin piksel sayılarının çok farklı olmasındandır. Çizelge 5.4'ten çıkarılan diğer bir sonuç ise B.U. algoritmasında kullanılan ikinci hata oranının birinci hata oranına göre karşılaştırılacak işaret sayısını genelde daha çok azaltmış olduğudur. Hough Dönüşüm algoritmasında ise işaret karşılaştırma sayısının genelde daha büyük çıkmasının ve yeterli elemeyi yapamamasının nedeni tabletlerde bulunan işaret resimlerinin bozukluğu ve çizgilerin eğriliği olarak söylenebilir.

Tez çalışmasında ayrıca Hitit çiviyazısı işaretlerinin okunmasında kullanılan ve OpenCV kütüphanesinde bulunan 3, 4, 5, 6, 7, 8 ve 9 numaralı algoritmaların işaret okuma performansları ölçülmüştür. Algoritmalar işaretler üzerinde kullanılmadan önce 512 x 512 piksel boyutlarına getirilmiştir. Bahsedilen algoritmaların işaret okuma performansları; tablet ve V.S. dijital işaret listesinde karşılaştırılan işaret üzerinde

bulunan anahtar noktalardan (keypoint) çıkartılan betimleyicilerin (descriptor) eşleşme oranına bakılarak bir hata oranı hesaplanmıştır ve bu hata oranı üzerinden algoritmalar değerlendirilmiştir. Hata oranının karşılaştırılan iki işaret için hesaplanması (5.1) nolu formülle gösterilmektedir.

$$Hata\ Oranı = \frac{|\text{İşaretlerdeki Eşleşen Betimleyici Sayısı Farkı}|}{\text{Birinci İşaret Toplam Betimleyici Sayısı}} \quad (5.1)$$

Örnek olarak SIFT algoritmasının HZL numarası 180 olan işarete uygulanması ile ilgili sonuç Şekil 5.3 ile gösterilmektedir.



Şekil 5.3 HZL No.180 olan işarete SIFT algoritması uygulanması

Burada hata oranı hesaplanırken birinci resimden 323 tane ve ikinci resimden 339 tane betimleyici çıkartılmıştır. Bu betimleyicilerden 323 tanesi eşleşmiştir. Bu durumda hata oranı şöyledir:

$$Hata\ oranı = (339 - 323) / 323 = 0.0495 \sim 0.05$$

Örnekten görüldüğü üzere hata oranı 0 ile 1 arasında bir değer almaktadır. Hata oranları, 149 çivi yazısı işareti için tabletten alınan ve V.S. işaret listesinden alınan versiyonları için uygulanmış ve sonucunda hata oranları EK 2. 'de verilmiştir. Hata

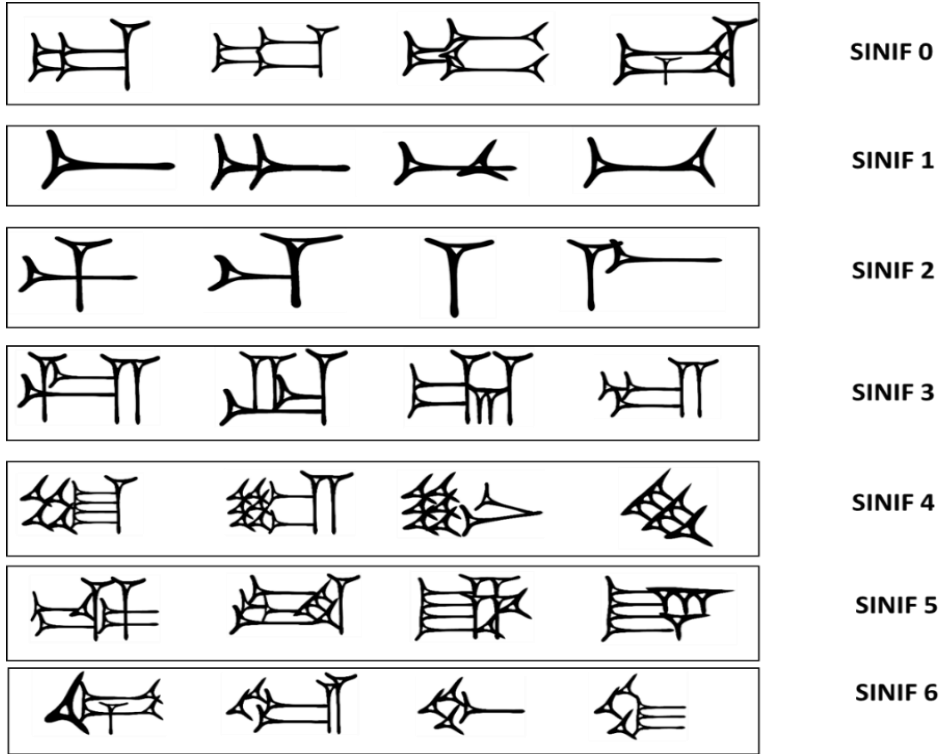
oranı ne kadar düşük ise algoritma işaret okumada o kadar başarılı kabul edilmiştir. EK 2.'deki sonuçlara bakılarak en düşük ortalama hata oranına sahip algoritmanın 0.17 hata oranıyla ORB olduğu görülmektedir. ORB'dan sonra gelen en iyi ikinci algoritmanın 0.24 hata oranıyla SIFT olduğu tespit edilmiştir. Bu iki algoritmanın düşük hata oranı üreterek çalışmasının sebepleri gürültü ve ışık yoğunluk farklarından etkilenmemesidir. En kötü algoritmanın ise 0.87 hata oranıyla MSER olduğu görülmektedir. MSER algoritmasının Hitit çivi yazısı işaretlerinde yüksek hata oranı vermesinin nedeni algoritmanın yuvarlak ve elips şekillere odaklı bir algoritma olması ve Hitit çivi yazısı işaretlerde bu yapıların bulunmamasıdır.

Hitit çivi yazısı işaretleri üzerine veri madenciliği uygulamalarında 149 işarete kümeleme ve sınıflandırma algoritmaları uygulanmıştır. İlk olarak çalışmada kullanılan 149 işaret için tezin 4.3 alt bölümünde bahsedilen öznitelikler hesaplanarak bir veri kümesi oluşturulmuştur.

Oluşturulan veri kümesi üzerine K-means kümeleme algoritması uygulanmıştır. Kümeleme algoritmasının veri kümesine uygulanmasının amacı sınıf bilgisi olmayan veri kümesinde benzer geometrik özelliklere sahip işaretlerin bir arada toplanarak işaretleri kategorilere (sınıflara) ayırabilmektir. Veri kümesine K-means kümeleme algoritması uygulanırken çeşitli K değerleri deneysel olarak seçilmiş ve işaretlerin görünüm ve geometriksel olarak birbirine benzer olanların aynı sınıfta toplanması durumları değerlendirilmiştir. Bu değerlendirmeler sonucunda K değerinin 7 olarak seçilmesi uygun görülmüştür. K-means kümeleme algoritması uygulanması sonucu veri kümesindeki işaretlere sınıf bilgisi eklenmiştir. Algoritmanın uygulanması sonucu 149 işaretin sınıf dağılımları Çizelge 5.5 ile gösterilmektedir. Belirlenen 7 sınıfa ait olan örnek çivi yazısı işaretleri Şekil 5.4 ile gösterilmektedir.

Çizelge 5.5 Hitit Çiviyazısı işaretlerinin sınıf dağılımları

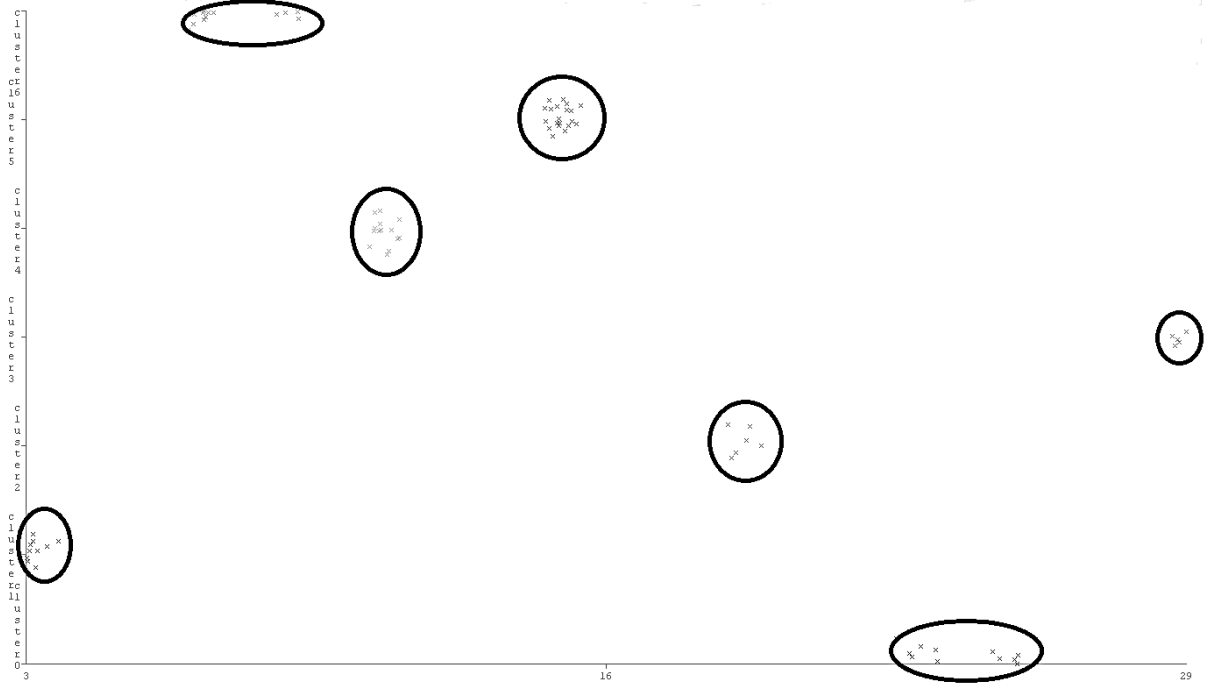
Sınıf Adı	İçerdiği İşaret Sayısı
Sınıf 0	15
Sınıf 1	29
Sınıf 2	29
Sınıf 3	19
Sınıf 4	19
Sınıf 5	12
Sınıf 6	26



Şekil 5.4 Sınıflara ayrılmış örnek işaretler

K-means algoritmasının uygulanması sonucu oluşan kümelerin (sınıfların) koordinat sisteminde gösterimi Şekil 5.5 'te görülmektedir. Bu koordinat sisteminde x koordinatı centroid (x-bileşeni) ve y koordinatı ise küme (sınıf) numarasını temsil etmektedir.

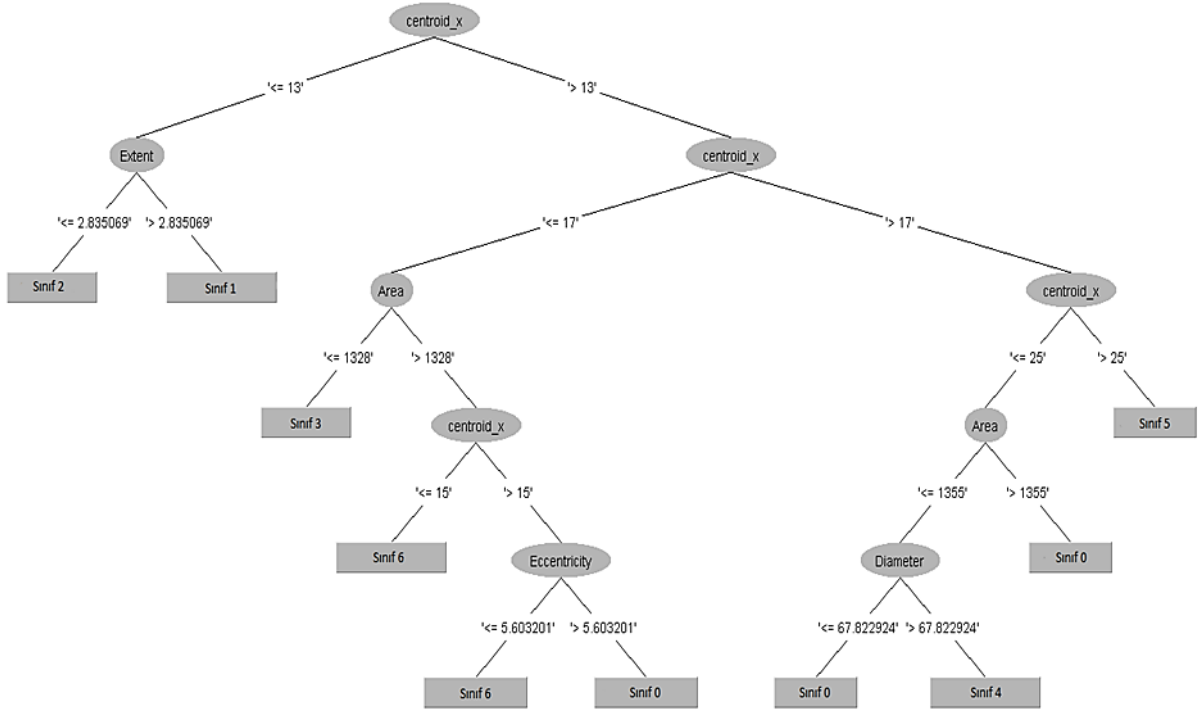




Şekil 5.5. K-means algoritmasının görselleştirilmesi

Çiviyazısı işaretlerine kümeleme algoritması uygulandıktan sonra sınıflandırma algoritmaları uygulanıp algoritmaların sınıflandırma performansları ölçülmüştür. Veri kümesine sınıflandırma uygulayabilmek için veri kümesi eğitim ve test kümesi olarak iki kısma bölünmüştür. Bu bölünme veri kümesinde bulunan elemanlarının %70'inin eğitim kümesine, %30'unun test kümesine aktarılmasıyla gerçekleştirilmiştir. Eğitim kümesinin eleman sayısı 104'tür. Test kümesi ise 45 elemandan oluşmaktadır.

Veri kümesine ilk uygulanan sınıflandırma algoritması J48 karar ağacı sınıflandırma algoritmasıdır. Eğitim kümesinde var olan verilerle karar ağacı model yapısı oluşturulmuştur. Bu karar ağacı yapısı Şekil 5.6 ile verilmiştir.



Şekil 5.6 Oluşturulan J48 karar ağacı modeli

Oluşturulan model üzerinden test kümesi kullanılarak sınıflandırma doğruluk değeri hesaplanmıştır. J48 karar ağacı algoritması oluşturulan model ile test verisinde bulunan 45 veri üzerinden 40 tanesinin sınıfını doğru tahmin etmiştir. Doğruluk oranı 89% olarak hesaplanmıştır. J48 algoritması için karışıklık matrisi Şekil 5.7 ile verilmiştir.

a	b	c	d	e	f	g	<-- Tahmin Edilen Sınıf
5	0	0	0	0	0	0	a = Sınıf 0
0	6	1	0	0	0	0	b = Sınıf 1
0	0	7	0	0	0	0	c = Sınıf 2
1	0	0	4	0	0	0	d = Sınıf 3
1	0	0	0	4	0	0	e = Sınıf 4
0	0	0	0	0	7	0	f = Sınıf 5
0	2	0	0	0	0	7	g = Sınıf 6

Şekli 5.7 J48 karışıklık matrisi

Veri kümesi uygulanan ikinci sınıflandırma algoritması K-NN algoritmasıdır. Sınıflandırma modeli belirlenen K değerine göre oluşturulmaktadır. K değerleri 1, 2,

3, 4, 5, 6, 7 olacak şekilde oluşturulan modellerin üzerinden test kümesi kullanılarak sınıflandırma doğruluk değerleri hesaplanmıştır. Çizelge 5.6 içinde farklı K değerleri için sınıflandırma doğruluğu ve sınıfı doğru tahmin edilen test kümesine ait eleman sayısı gösterilmektedir.

Çizelge 5.6 Seçilen K değerlerine göre K-NN doğruluk oranları

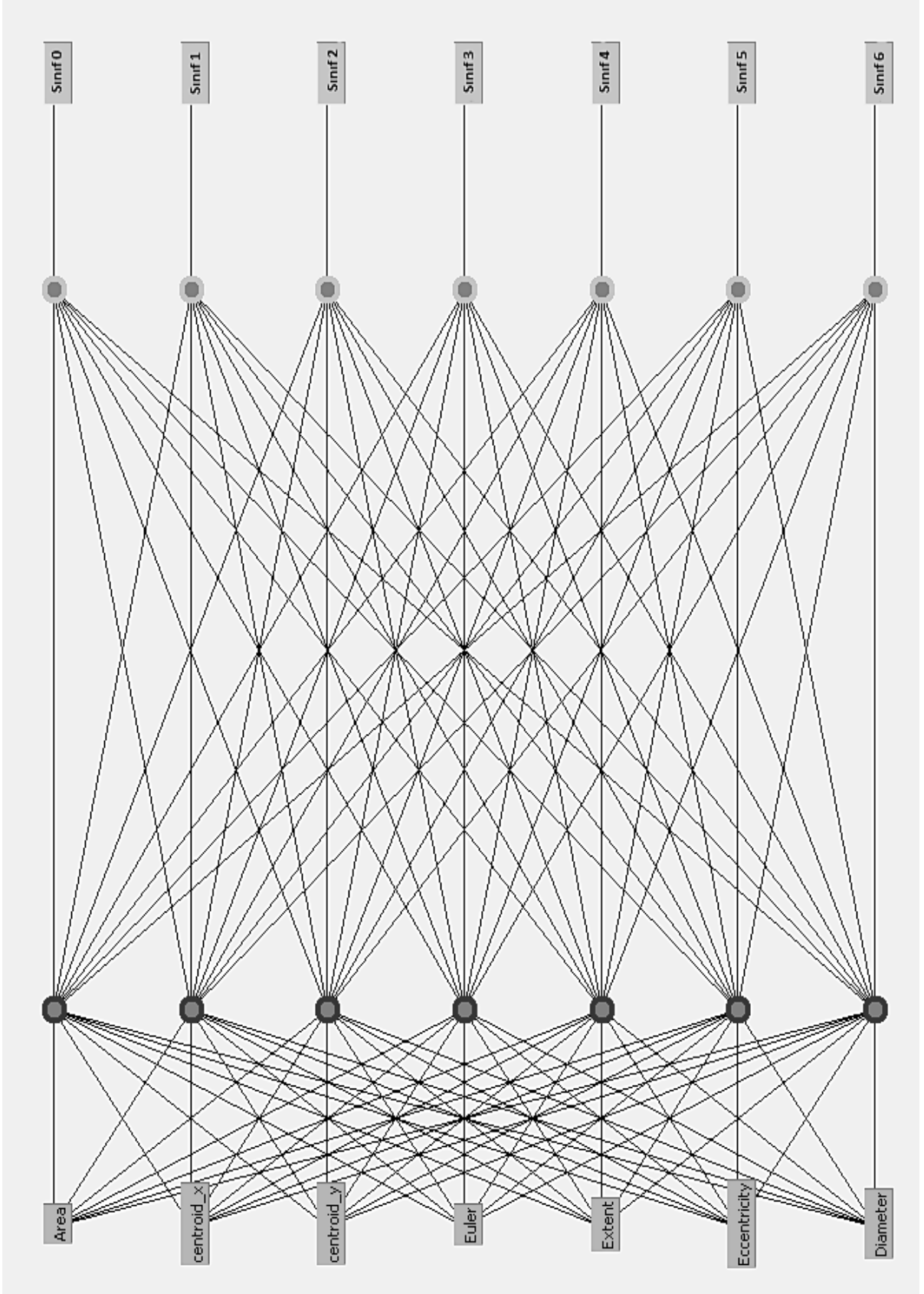
K değeri	Sınıfı Doğru Tahmin Edilen Test Kümesi Eleman Sayısı	Doğruluk Oranı
1	37	82%
2	37	82%
3	40	89%
4	41	91%
5	41	91%
6	42	93%
7	39	87%

Çizelge 5.6'da görüldüğü üzere en yüksek doğruluk oranı K değerinin 6 seçilmesiyle elde edilmiştir. K=6 değeri için K-NN algoritmasının karışıklık matrisi Şekil 5.8 ile gösterilmiştir.

a	b	c	d	e	f	g	<-- Tahmin Edilen Sınıf
5	0	0	0	0	0	0	a = Sınıf 0
0	7	0	0	0	0	0	b = Sınıf 1
0	0	7	0	0	0	0	c = Sınıf 2
0	0	0	5	0	0	0	d = Sınıf 3
0	0	0	0	5	0	0	e = Sınıf 4
0	0	0	0	3	4	0	f = Sınıf 5
0	0	0	0	0	0	9	g = Sınıf 6

Şekil 5.8 K=6 değeri için K-NN karışıklık matrisi

Veri kümesine uygulanan son sınıflandırma algoritması yapay sinir ağı sınıflandırma algoritmasıdır. Kullanılan yapay sinir ağı sınıflandırma modeli 3 katmandan oluşmaktadır. Bunlar Girdi, Gizli ve Çıktı katmanıdır. Kullanılan yapay sinir ağı yapısı Şekil 5.9 'da verilmiştir.



Şekil 5.9 Sınıflandırmada kullanılan yapay sinir ağı modeli

Burada ilk katman giriş katmanına ait düğümlerdir ve veri kümesindeki öznelikleri (Area, extent, vb.) göstermektedir. İkinci katman gizli katmana ait düğümlerden oluşmuştur. Son katman ile gösterilen düğümler ise çıkış katmanına ait düğümlerdir ve sınıf etiketlerine bağlıdır.

Oluşturulan model üzerinden test kümesi kullanılarak sınıflandırma doğruluk değeri hesaplanmıştır. Yapay sinir ağı sınıflandırma algoritması, oluşturulan model ile test verisinde bulunan 45 veri üzerinden 38 tanesinin sınıfını doğru tahmin etmiştir. Doğruluk oranı 85% olarak hesaplanmıştır. YSA sınıflandırma algoritması için karışıklık matrisi Şekil 5.10 ile görülmektedir.

a	b	c	d	e	f	g	<-- Tahmin Edilen Sınıf
4	0	0	1	0	0	0	a = Sınıf 0
0	7	0	0	0	0	0	b = Sınıf 1
0	0	7	0	0	0	0	c = Sınıf 2
0	0	0	4	0	0	1	d = Sınıf 3
0	0	0	0	5	0	0	e = Sınıf 4
0	0	0	0	5	2	0	f = Sınıf 5
0	0	0	0	0	0	9	g = Sınıf 6

Şekil 5.10 Yapay sinir ağı karışıklık matrisi

Hitit çiviyazısı işaretleri veri kümesine uygulanan sınıflandırma algoritmalarından hesaplanan doğruluk değerine göre en yüksek değere sahip algoritma K değerinin 6 seçilmesiyle uygulanan K-NN algoritmasıdır. Hitit çiviyazısı işaretleri veri kümesine uygulanan sınıflandırma algoritmalarının doğruluk değerleri Çizelge 5.7 ile verilmiştir.

Çizelge 5.7 Kullanılan sınıflandırma algoritmalarının doğruluk değerleri

Algoritma Adı	Doğruluk Değeri
K-NN (K=6)	93%
J48 Karar Ağacı	89%
Yapay Sinir Ağı	85%

## 6. ÖZET VE SONUÇLAR

Bu tez çalışmasında tablettten alınan bir çiviyazısı işaretinin okunup V.S. dijital işaret listesindeki karşılığının bulunması gerçekleştirilmiştir. Tabletteki işaretlerin karşılığının bulunmasında ve işaret geri getiriminde en iyi performansı gösteren algoritmanın HOG olduğu tespit edilmiştir. Ayrıca tablet üzerindeki işaretleri okurken algoritmaların çalışma zamanlarını kısaltacak iki ön eleme yöntemi önerilmiştir. İşaret okumada kullanılan OpenCV kütüphanesine ait olan algoritmaların işaret okumadaki performansları hata oranı cinsinden verilmiş ve en az hata oranına sahip algoritmanın ORB algoritması olduğu saptanmıştır. Daha sonra V.S. dijital işaret listesindeki işaretlere K-means kümeleme algoritması uygulanarak geometrik olarak benzer işaretlerin birarada toplanması gerçekleştirilmiştir ve işaretlere sınıf etiketleri atanmıştır. Son olarak oluşturulan işaret veri kümesine sınıflandırma algoritmaları uygulanmış ve doğruluk değerine göre sınıflandırma performansları ölçülmüştür. En yüksek doğruluk oranına sahip algoritmanın K-NN algoritması olduğu saptanmıştır.

Hitit çiviyazısı işaretlerinin bilgisayar destekli olarak hızlı ve hata oranı düşük bir yüzdeyle okunmasını ve veri madenciliğinin ilgili teknikleri yardımıyla otomatik anlam çıkarımını sağlayan bu tez çalışmasının daha iyi sonuçlar verebilmesi için varolan veri miktarının artırılması gerekmektedir. İşaretlerin dijital versiyonlarının daha fazla olması (aynı işaretin farklı kişiler tarafından ve/veya farklı ortamlarda yazılmış hali) ve bu verilerin çevrimiçi bir veritabanında saklanabilmesi çalışmanın performansı için çok yararlı olacaktır. Ayrıca Hitit çiviyazısının V.S. dijital işaret listesi dışında yüksek çözünürlüklü işaret resimleri içeren kaynakların var olması da çalışmadaki tablet üzerindeki işaretlerin doğru okuma oranını artıracığı gerçektir. Diğer bir konu ise tablet üzerindeki çiviyazısı işaretlerin çözünürlük kalitesidir. Portal Mainz'da bulunan tabletlerdeki işaretlerin düşük çözünürlüğe sahip olması çalışmada zorluklar yaratmıştır. Düşük çözünürlük sebebiyle işaretlerin detayları tam anlamıyla kullanılamamıştır. Bu durumu önlemek için yüksek çözünürlüklü tablet resimlerinden bir veritabanı oluşturulması, işaretlerin okunması ve detaylara ulaşılması konusunda iyileştirme sağlayacaktır. Ayrıca ön elemeyi gerçekleştiren algoritmalarda kullanılan eşik değerlerinin üzerinde ek çalışmaların yapılması algoritmaların performanslarını yükselteceği öngörülmektedir.

Bu tez çalışması sırasında veri madenciliğinin sınıflandırma tekniği ile işaretlerin sınıflara ayrılması gerçekleştirilmiştir. Tabletten işaret okuma sırasında işaretin sınıf bilgisini bir ön bilgi gibi kullanarak veritabanında gerçekleştirilen işaret karşılaştırma sayısı kısaltılabilir ve işaretin kategorisi bilindiği için veritabanında sadece o sınıfta arama yapılabilir. Ayrıca sınıflandırma teknikleriyle çalışmada kullanılacak yeni işaretlerin sınıfları tahmin edilerek tabletlerden işaret okuma süreçleri daha etkili ve hızlı gerçekleştirilebilir.

## KAYNAKLAR LİSTESİ

- AGHAV, Sushila, PAYGUDE, Shilpa, Computer Assisted Printed Character Recognition in Document Based Images, Procedia Engineering, vol.38, s.3222-3227, 2012.
- Ahamed, S., Hareesha, K., Dynamic Clustering of Data with Modified K-Means Algorithm, International Conference on Information and Computer Networks (ICICN), s.221-225, 2012.
- AHMED, Kawther K., Online Sumarians Cuneiform Detection Based on Symbol Structural Vector Algorithm, Journal Of The College Of Education For Women, vol.23, no.2, s.545-553, 2012.
- Aktaş, A. Z., Gürsel, H. ,Çiviyazısı Metinlerin Uzman Sistemler Yardımıyla Çözülmesi, 5. Türkiye Bilgisayar Kongresi, Ankara, 1988.
- AKURGAL, Ekrem, Anadolu Kültür Tarihi, TÜBİTAK, 409s., 2005.
- ALP, Sedat, Hitit Çağında Anadolu, Tübitak Popüler Bilim Kitapları, 2000.
- ARMON, Shahar, Handwriting Recognition and Fast Retrieval for Hebrew Historical Manuscripts, M.Sc. thesis, Hebrew University, 38s, 2011.
- BAYKAL, Abdullah, Veri Madenciliği Uygulama Alanları, D.Ü. Ziya Gökalp Eğitim Fakültesi Dergisi 7, s.95-107, 2006.
- BHARGAVA, Neeraj, SHARMA, Girja, BHARGAVA, Ritu, MATHURIA, Manish, Decision Tree Analysis on J48 Algorithm for Data Mining, International Journal of Advanced Research in Computer Science and Software Engineering, vol.3, no.6, s.1114-1119, 2013.
- BHATIA, Nitin, Survey of Nearest Neighbor Techniques, International Journal of Computer Science and Information Security, vol.8, no.2, s.302-305, 2010.
- CELIK, Arzum E., KARATEPE, Yalcin, Evaluating and forecasting banking crises through neural network models: An application for Turkish banking sector, Expert Systems with Applications, vol.33, s.809-815, 2007.
- Chunhavittayatera, S., Chitsobhuk, O., Tongprasert, K., Image Registration using Hough Transform and Phase Correlation , International Conference on Advanced Communication Technology (ICACT), s.973-977, 2006.
- COVER, Thomas M., HART, John P., Nearest neighbor pattern classification, IEEE Transactions on Information Theory, vol.3, no.1, s.21-27, 1967.
- ÇIĞ, Muazzez İ., Hititler ve Hattuşa İştar'ın Kaleminden, Kaynak Yayınları, 187s., 2000.



- Dalal, N., Triggs, B., Histograms of oriented gradients for human detection, IEEE Society Conference on Computer Vision and Pattern Recognition (CVPR), s.886-893, 2005.
- Das, P., Paul, S., Ghoshal, R., Recognition of Bangla Basic Characters using Multiple Classifiers, Computer and Communication Technology (ICCCCT), 2nd International Conference, s.88-93, 2011.
- DİK, Elif Ceren, Hitit Çiviyazısı İşaretlerinin Otomatik Çevirisi, Yüksek Lisans Tezi, Başkent Üniversitesi, 100s., 2014.
- EDAN, Naktal M., Cuneiform Symbols Recognition Based on K-Means and Neural Network, Journal of Computational Mathematics, vol.10, no.1, s.195-202, 2013.
- Fayyad, U., Piatetsky-shapiro, G., Smyth, P., Knowledge Discovery and Data Mining: Towards a Unifying Framework, Second international conference on KDD, s.82-88, 1996.
- GÜRSEL, H., An Expert System for Cuneiform Interpretation, ODTÜ-BMB, 1988
- GONZALEZ, Rafael C. and WOODS, Richard E., Digital Image Processing, 2<sup>nd</sup> Edition, Prentice Hall, 2002.
- Hahn, D.V., Duncan, D.D., Baldwin, K.C., Cohen, J.D., Purnomo, B., Digital Hammurabi: Design and development of a 3D scanner for cuneiform tablets, three-dimensional image capture and applications Conference 7th, 12s, 2006.
- HAN, Jiawei and KAMBER, Micheline, Data Mining Concept and Techniques, 2<sup>nd</sup> Edition, Elsevier Inc., 750s, 2006.
- Harris, C., Stephens, M., A combined corner and edge detector, Alvey Vision Conference, s.147-151, 1988.
- Herbert, B., Tuytelaars, T., Van Gool, L., SURF: Speeded Up Robust Features, Lecture Notes in Computer Science Volume 3951, s.404-417, 2006.
- HIRÇIN, Selen, Çivi Yazısı: Ortaya Çıkışı, Gelişmesi, Çözümü, Eskiçağ Bilimleri Enstitüsü Yayınları, 35s., 2000.
- HOLAMBE, Anilkumar N., THOOL, Ravindra C., Combining Multiple Feature Extraction Technique and Classifiers for Increasing Accuracy for Devanagari OCR, International Journal of Soft Computing and Engineering (IJSCE), vol.3, no.4, s.38-41, 2013.
- Hong Nguyen, T.K., Belleudy, C., Pham, T.V., Power evaluation of Sobel filter on Xilinx platform, IEEE Faible Tension Faible Consommation (FTFC), s.1-5, 2014.
- HUTTENLOCHER, Daniel P, KLANDERMAN, Gregory A., RUCKLIDGE, William J., Comparing images using the Hausdorff distance, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.15, no.9, s.850-863, 1993.

- KARASU, Cem, Çivi Yazılı Belgelere Göre Eski Anadolu Kadını Hakkında Bazı Gözlemler, ÇÜ Sosyal Bilimler Enstitüsü Dergisi, Cilt 15, Sayı 3, s.45-66, 2006.
- KARASU, Cem, Hititçe ve Hitit Çivi Yazısı, Editörler: M. D. Alparslan ve M. Alparslan, Hititler, Yapı Kredi Yayınları, ss.84 – 93, 2013.
- Kınal, F., Çiviyazısının Doğuşu ve Gelişmesi, Türk Tarih Kurumu Konferansı, s.1-13, 1971.
- KUMAR, Dharminder, BHARDWAJ, Deepak, Rise of Data Mining: Current and Future Application Areas, International Journal of Computer Science Issues, vol.8, no.5, 2011.
- Leutenegger, S., Chli, M., Siegwart, R. Y., BRISK: Binary robust invariant scalable keypoints, IEEE International Conference on Computer Vision, s.2548-2555, 2011.
- LOWE, David G., Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, vol. 60, no. 2, s.91-110, 2004.
- Mall, V., Roy, A.K., Mitra, S.K., Shukla, S., Detection of structural tampering in a digital image using Canny Edge Detector, International Conference on Informatics, Electronics & Vision (ICIEV), s.1-7, 2013.
- Matas, J., Chum, O., Urban, M., Pajdla, T., Robust wide-baseline stereo from maximally stable extremal regions, British Machine Vision Conference, s.384-393, 2002.
- MEMİŞ, Ekrem, Eskiçağ Türkiye Tarihi, Çizgi Kitabevi, 1995.
- PRADHAN, Biswajeet, LEE, Saro, 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, s.143-152, 2007.
- QUINLAN, J. Ross, C4.5: programs for machine learning, Morgan Kaufmann Publishers Inc, 1993.
- Rosten, E., Drummond, T., Machine learning for high-speed corner detection, 8th European conference on Computer Vision (ECCV), s.430-443, 2006.
- Rublee, R., Rabaud, V., Konolige, K., Bradski, G., ORB: an efficient alternative to SIFT or SURF, IEEE International Conference in Computer Vision (ICCV), s.2564-2571, 2011.
- RUSTER Christel and NEU Erich, Hethitisches Zeichenlexikon: Inventar und Interpretation Der Keilschriftzeichen aus den Bogazkoy-Texten, O. Harrassowitz, 386s., 1989.

- SHARMA, Aman K., SAHNİ, Suruchi, A Comparative Study of Classification Algorithms for Spam Email Data Analysis, International Journal on Computer Science and Engineering (IJCSE), vol.3, no.5, s.1890-1895, 2011.
- SUGUNA, N., THANUSHKODI, K., An Improved k-Nearest Neighbor Classification Using Genetic Algorithm, International Journal of Computer Science Issues, vol.7, Issue 4, no.2, s.18-21, 2010.
- Sundar, K.A., John, M., A high precision printed character recognition method for Tamil script, IEEE International Conference on Computational Intelligence and Computing Research (ICCI), s.1-5, 2013.
- Thangalakshmi, P., Kamalesh, S., A Decentralized Service Discovery Approach on Peer-To-Peer Networks, International Conference on Innovations in Engineering and Technology (ICIET), s.2423-2429, 2014.
- Tyndall, S., Toward automatically assembling hittite-language cuneiform tablet fragments into larger texts, 50th Annual Meeting of the Association for Computational Linguistics, vol.2, s.243-247, 2012.
- ÜNAL, Ahmet, Hititler Devrinde Anadolu Kitap 1, Arkeoloji ve Sanat Yayınları, 224s., 2002.
- VAN DEN HOUT, Theo, The Elements of Hittite, Cambridge University Press, 2011.
- YEŞİLBUDAK, Mehmet, KAHRAMAN, Hamdi T., KARACAN, Hacer, Veri Madenciliğinde Nesne Yönelimli Birleştirici Hiyerarşik Kümeleme Modeli, J. Fac. Eng. Arch. Gazi Univ., vol.26, no.1, s.27-39, 2011.
- YEŞİLTEPE, Beste, Hitit Çiviyazılı Metinlerin Okunmasında Uzman Sistem Uygulama Örnekleri, Yüksek Lisans Tezi, Başkent Üniversitesi, 2015.
- YOUSIF, Hilal, RAHMA, Abdul Munim, ALANI, Haithem, Cuneiform Symbols Recognition Using Intensity Curves, The International Arab Journal of Information Technology, vol.3, no.3, s.237-241, 2006.
- ZAKI, Mohammed J. and MEIRA JR., Wagner, Data Mining and Analysis Fundamental Concepts and Algorithms, Cambridge University Press, 2014.

## WEB SAYFALARI LİSTESİ (Metindeki dipnot sırasına göre)

1. <http://whc.unesco.org/en/list/377>
5. <https://tr.wikipedia.org/wiki/Hititler>
6. [https://tr.wikipedia.org/wiki/Kadeř\\_Antlařması](https://tr.wikipedia.org/wiki/Kadeř_Antlařması)
7. <http://www.kulturvarliklari.gov.tr/TR,44427/hattusas-bogazkoy---hitit-baskenti-corum.html>
8. <http://tarih.tumders.com/wp-content/ęivi-yazısı.jpg>
9. <http://www.dogalrehber.com/image/bitki/kiltablet3.Jpeg>
10. <http://www.hethport.uni-wuerzburg.de/HPM/index.html>
11. <http://www.hethport.uni-wuerzburg.de/cuneifont/>
12. <http://windows.microsoft.com/tr-tr/windows7/products/features/paint>
13. <http://www.mathworks.com/products/matlab/>
14. <http://opencv.org/>
15. <http://www.mathworks.com/help/images/ref/regionprops.html>
16. <http://www.mathworks.com/help/matlab/ref/rgb2gray.html>
17. <http://www.mathworks.com/help/images/ref/bwmorph.html> (skeleton)
18. <http://www.mathworks.com/help/images/ref/bwmorph.html> (clean, spur)
19. [https://en.wikipedia.org/wiki/Cosine\\_similarity](https://en.wikipedia.org/wiki/Cosine_similarity)
20. [https://en.wikipedia.org/wiki/Euclidean\\_distance](https://en.wikipedia.org/wiki/Euclidean_distance)
21. [www.cs.waikato.ac.nz/ml/weka/](http://www.cs.waikato.ac.nz/ml/weka/)
22. <http://www.codeproject.com/Articles/32970/K-Nearest-Neighbor-Algorithm-Implementation-and-Ov>
23. [http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta\\_YapaySinirAglari.pdf](http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta_YapaySinirAglari.pdf)
24. <http://www.dataminingmasters.com/uploads/studentProjects/NeuralNetworks.pdf>
25. [http://3.bp.blogspot.com/-pCd6t1smidg/UF\\_9C43gfl/AAAAAAAAAn8/a8yIS3v98A0/s1600/YSA.jpg](http://3.bp.blogspot.com/-pCd6t1smidg/UF_9C43gfl/AAAAAAAAAn8/a8yIS3v98A0/s1600/YSA.jpg)

26. <http://www.dataminingmasters.com/uploads/studentProjects/NeuralNetworks.pdf>

27. [http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta YapaySinirAglari.pdf](http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta%20YapaySinirAglari.pdf)

## EK 1. Vanseveren Hitit Çiviyazısı İşaretleri Dijital Listesindeki İlk 16 İşaret Örneği

HZL	Borger	Unicode	UllikummiA	UllikummiB	UllikummiC	Hittite	Sumerian	Autotext
1	1	12038	𐎠			aš	AŠ	ac
2	3	1212C	𐎡			ḫal	ḪAL	hal
3	741 882	1227E	𐎢𐎣				PÉŠ	pec2
4	5	12044	𐎢𐎣			bal, pal	BAL	bal
5	115	122D3	𐎢𐎣			šir	NU <sub>11</sub> , SIR <sub>4</sub> , ŠIR	cir
6	6	12108	𐎢𐎣	𐎢𐎣			GÍR, UL <sub>4</sub>	gir2
7	9	122FB	𐎢𐎣	𐎢𐎣	𐎢𐎣	tar, ḫaš	KU <sub>5</sub> , KUD, TAR	tar
8	10	1202D	𐎢𐎣			an	AN, DINGIR	an
9	90	12033	𐎢𐎣	𐎢𐎣			APIN, ENGAR	apin
10	91	12224	𐎢𐎣	𐎢𐎣	𐎢𐎣	maḫ	MAḪ	mah
11	112	12261	𐎠			nu	NU	nu
12	117	121B0	𐎢𐎣	𐎢𐎣		kul	KUL, NUMUN	numun, kul
13	113	12041	𐎠			pát/d, pí/d/t, pè, pí	BAD, BE, SUMUN, SUN, TIL, ÚŠ	bad
14	167	12074	𐎢𐎣	𐎢𐎣		dim, tim		dim
15	110	1223E	𐎢𐎣			na	NA	na
16	18	12034	𐎢𐎣				ÀR, ARAD, NÍTA	arad

## EK 2. Hata Oranları Çizelgesi

(OpenCV kütüphanesinde bulunan 3, 4, 5, 6, 7, 8 ve 9 numaralı algoritmaların işaret okuma performanslarının hata oranları çizelgesi)

<b>HZL NO</b>	<b>SIFT</b>	<b>BRISK</b>	<b>ORB</b>	<b>SURF</b>	<b>FAST</b>	<b>HARRIS</b>	<b>MSER</b>
1	0.333	0.674	0.444	0.750	1	0.808	1
2	0.513	0.477	0.129	0.825	1	0.874	1
4	0.119	0.136	0.179	0.589	0.830	0.857	0.902
5	0.124	0.387	0.189	0.754	0.856	0.710	1
7	0.218	0.172	0.090	0.722	1	0.250	1
8	0.709	0.773	0.431	0.587	0.445	0.858	0.667
10	0.208	0.175	0.230	0.596	0.724	0.700	1
11	0.586	0.250	0.052	0.591	1	0.000	1
12	0.215	0.200	0.221	0.783	0.685	0.841	0.747
13	0.265	0.472	0.107	0.637	1	0.119	1
14	0.033	0.394	0.179	0.652	0.868	0.634	0.649
15	0.387	0.247	0.182	0.493	1	0.723	0.500
17	0.122	0.790	0.218	0.564	0.600	0.717	1
20	0.363	0.101	0.148	0.631	0.815	0.793	0.787
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
32	0.206	0.163	0.139	0.693	0.767	0.849	0.857
33	0.181	0.079	0.209	0.663	0.852	0.864	0.914
37	0.537	0.208	0.123	0.503	0.744	0.895	0.903
39	0.199	0.189	0.224	0.684	0.650	0.800	0.879
40	0.368	0.296	0.233	0.571	0.717	0.864	0.924
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

<b>HZL NO</b>	<b>SIFT</b>	<b>BRISK</b>	<b>ORB</b>	<b>SURF</b>	<b>FAST</b>	<b>HARRIS</b>	<b>MSER</b>
<b>49</b>	0.325	0.704	0.220	0.746	0.749	0.768	0.800
<b>51</b>	0.413	0.210	0.131	0.721	0.781	0.820	0.721
<b>61</b>	0.328	0.016	0.143	0.731	0.704	0.687	0.816
<b>67</b>	0.126	0.211	0.218	0.621	0.781	0.832	0.886
<b>68</b>	0.146	0.310	0.194	0.604	0.721	0.829	0.890
<b>72</b>	0.121	0.362	0.231	0.762	0.349	0.902	0.914
<b>77</b>	0.110	0.129	0.191	0.781	0.643	0.702	0.958
<b>81</b>	0.338	0.802	0.244	0.724	0.513	0.715	0.571
<b>91</b>	0.415	0.222	0.168	0.604	0.500	0.819	0.938
<b>92</b>	0.419	0.089	0.184	0.693	0.656	0.698	0.692
<b>93</b>	0.163	0.529	0.117	0.731	0.809	0.851	0.889
<b>95</b>	0.427	0.270	0.157	0.700	0.751	0.758	0.756
<b>97</b>	0.063	0.526	0.132	0.661	0.704	0.820	0.906
<b>98</b>	0.192	0.262	0.188	0.730	0.747	0.809	0.909
<b>99</b>	0.079	0.451	0.053	0.778	0.793	0.792	0.774
<b>100</b>	0.158	0.771	0.230	0.697	1	0.901	1
<b>105</b>	0.248	0.279	0.196	0.701	0.699	0.774	0.892
<b>108</b>	0.180	0.202	0.176	0.709	0.655	0.714	0.901
<b>112</b>	0.360	0.519	0.334	0.636	0.766	0.865	0.720
<b>113</b>	0.367	0.488	0.073	0.712	0.707	0.779	0.924
<b>117</b>	0.059	0.364	0.239	0.609	0.716	0.781	0.885
<b>120</b>	0.218	0.676	0.216	0.496	0.865	0.862	0.902
<b>122</b>	0.294	0.647	0.258	0.647	1	0.868	1
<b>124</b>	0.038	0.425	0.104	0.616	0.846	0.798	0.750
<b>125</b>	0.119	0.348	0.123	0.709	0.830	0.853	0.909
<b>128</b>	0.336	0.891	0.137	0.698	0.658	0.804	0.862
<b>131</b>	0.027	0.036	0.179	0.732	0.794	0.832	0.861
<b>132</b>	0.050	0.288	0.137	0.666	0.739	0.831	0.810
<b>133</b>	0.102	0.026	0.143	0.545	0.811	0.819	0.884



<b>HZL NO</b>	<b>SIFT</b>	<b>BRISK</b>	<b>ORB</b>	<b>SURF</b>	<b>FAST</b>	<b>HARRIS</b>	<b>MSER</b>
<b>151</b>	0.185	0.380	0.154	0.686	0.790	0.808	0.810
<b>152</b>	0.315	0.160	0.064	0.576	0.598	0.860	0.842
<b>153</b>	0.377	0.191	0.015	0.664	0.681	0.836	0.766
<b>158</b>	0.047	0.076	0.141	0.419	0.872	0.851	0.912
<b>159</b>	0.173	0.581	0.119	0.576	0.797	0.804	0.622
<b>160</b>	0.196	0.252	0.210	0.712	0.789	0.751	0.789
<b>168</b>	0.265	0.524	0.278	0.650	0.611	0.784	0.842
<b>169</b>	0.241	0.255	0.205	0.507	0.804	0.768	0.901
<b>171</b>	0.165	0.387	0.216	0.426	0.835	0.836	0.852
<b>173</b>	0.204	0.012	0.143	0.665	0.180	0.826	0.937
<b>174</b>	0.133	0.609	0.099	0.748	0.340	0.791	0.964
<b>178</b>	0.328	0.101	0.079	0.677	0.013	0.796	0.958
<b>179</b>	0.392	0.354	0.473	0.801	0.554	0.837	0.889
<b>180</b>	0.049	0.687	0.251	0.628	0.400	0.840	0.250
<b>183</b>	0.019	0.215	0.050	0.685	0.689	0.743	0.875
<b>185</b>	0.373	0.769	0.037	0.695	1	0.906	1
<b>187</b>	0.049	0.693	0.154	0.591	0.506	0.826	0.839
<b>191</b>	0.290	0.154	0.034	0.782	0.718	0.728	0.889
<b>192</b>	0.024	0.418	0.105	0.630	0.837	0.795	0.866
<b>195</b>	0.184	0.326	0.056	0.499	0.876	0.645	0.712
<b>196</b>	0.074	0.052	0.067	0.667	0.808	0.820	0.852
<b>197</b>	0.245	0.516	0.229	0.496	0.566	0.456	0.832
<b>198</b>	0.114	0.294	0.119	0.655	0.844	0.825	0.875
<b>202</b>	0.290	0.341	0.262	0.539	1	0.844	1
<b>205</b>	0.203	0.250	0.093	0.558	1	0.907	1
<b>206</b>	0.107	0.310	0.032	0.614	0.784	0.858	0.850
<b>208</b>	0.126	0.287	0.150	0.488	0.451	0.651	0.765
<b>209</b>	0.366	0.132	0.127	0.709	0.824	0.787	0.804
<b>210</b>	0.265	0.809	0.245	0.698	0.551	0.633	0.841

<b>HZL NO</b>	<b>SIFT</b>	<b>BRISK</b>	<b>ORB</b>	<b>SURF</b>	<b>FAST</b>	<b>HARRIS</b>	<b>MSER</b>
<b>214</b>	0.092	0.411	0.220	0.528	0.459	0.674	1
<b>215</b>	0.182	0.050	0.184	0.631	0.813	0.785	0.793
<b>217</b>	0.028	0.020	0.242	0.712	0.353	0.802	0.866
<b>218</b>	0.592	0.782	0.070	0.670	0.180	0.797	0.143
<b>233</b>	0.260	0.289	0.058	0.665	0.782	0.737	0.762
<b>240</b>	0.333	0.128	0.208	0.614	0.733	0.808	0.734
<b>241</b>	0.013	0.115	0.161	0.696	0.530	0.787	0.737
<b>242</b>	0.142	0.185	0.130	0.639	0.742	0.788	0.785
<b>243</b>	0.039	0.102	0.109	0.624	0.469	0.770	0.440
<b>244</b>	0.363	0.697	0.215	0.670	1	0.865	1
<b>245</b>	0.171	0.427	0.131	0.764	0.751	0.738	0.737
<b>251</b>	0.275	0.300	0.277	0.738	0.398	0.863	0.886
<b>260</b>	0.325	0.103	0.221	0.604	0.727	0.802	0.850
<b>261</b>	0.688	0.125	0.199	0.568	0.966	0.923	1
<b>267</b>	0.178	0.043	0.230	0.664	0.472	0.794	0.787
<b>271</b>	0.174	0.147	0.185	0.602	0.848	0.851	0.917
<b>273</b>	0.198	0.513	0.125	0.674	0.850	0.767	0.857
<b>275</b>	0.367	0.480	0.175	0.673	0.804	0.831	0.920
<b>286</b>	0.442	0.149	0.085	0.652	0.049	0.861	0.960
<b>288</b>	0.160	0.800	0.169	0.542	0.651	0.844	0.833
<b>289</b>	0.086	0.076	0.173	0.554	0.862	0.855	0.926
<b>290</b>	0.128	0.370	0.166	0.544	1	0.865	1
<b>296</b>	0.433	0.552	0.173	0.372	1	0.955	1
<b>297</b>	0.638	0.058	0.143	0.577	0.518	0.858	0.914
<b>298</b>	0.100	0.606	0.177	0.639	0.798	0.766	0.867
<b>300</b>	0.186	0.767	0.298	0.502	1	0.906	1
<b>304</b>	0.284	0.863	0.266	0.488	0.820	0.903	0.896
<b>306</b>	0.076	0.293	0.175	0.563	0.795	0.821	0.830
<b>307</b>	0.202	0.123	0.157	0.709	0.752	0.768	0.821

<b>HZL NO</b>	<b>SIFT</b>	<b>BRISK</b>	<b>ORB</b>	<b>SURF</b>	<b>FAST</b>	<b>HARRIS</b>	<b>MSER</b>
<b>310</b>	0.077	0.227	0.200	0.656	0.613	0.755	0.846
<b>312</b>	0.114	0.355	0.225	0.811	0.757	0.816	0.842
<b>313</b>	0.107	0.172	0.220	0.624	0.719	0.772	0.918
<b>316</b>	0.591	0.050	0.159	0.500	0.759	0.904	0.943
<b>317</b>	0.216	0.763	0.218	0.733	0.692	0.811	0.810
<b>329</b>	0.770	0.377	0.258	0.224	0.180	0.866	1
<b>330</b>	0.556	0.034	0.203	0.540	0.473	0.877	1
<b>331</b>	0.743	0.046	0.205	0.530	0.197	0.923	1
<b>332</b>	0.397	0.756	0.179	0.645	0.642	0.842	0.861
<b>333</b>	0.418	0.041	0.188	0.515	0.586	0.828	0.804
<b>335</b>	0.384	0.513	0.032	0.573	0.462	0.811	0.979
<b>337</b>	0.030	0.189	0.193	0.662	0.845	0.815	0.818
<b>338</b>	0.162	0.253	0.062	0.705	0.678	0.850	1
<b>339</b>	0.412	0.059	0.218	0.619	0.839	0.863	0.957
<b>340</b>	0.117	0.413	0.268	0.438	0.874	0.841	0.890
<b>343</b>	0.136	0.014	0.266	0.614	0.835	0.855	0.897
<b>344</b>	0.371	0.305	0.216	0.745	0.662	0.630	0.880
<b>346</b>	0.240	0.044	0.233	0.623	0.809	0.782	0.881
<b>353</b>	0.139	0.145	0.206	0.611	0.796	0.767	0.905
<b>354</b>	0.146	0.099	0.224	0.576	0.827	0.832	0.880
<b>355</b>	0.331	0.537	0.242	0.505	0.582	0.756	0.870
<b>356</b>	0.148	0.200	0.305	0.779	0.536	0.930	0.971
<b>357</b>	0.136	0.094	0.217	0.755	0.298	0.821	0.958
<b>358</b>	0.594	0.810	0.212	0.606	1	0.913	1
<b>360</b>	0.295	0.303	0.121	0.589	0.861	0.891	0.963
<b>364</b>	0.153	0.837	0.244	0.704	0.135	0.878	0.960
<b>366</b>	0.318	0.122	0.129	0.731	0.351	0.841	0.857
<b>367</b>	0.119	0.330	0.117	0.725	0.577	0.821	0.864
<b>369</b>	0.061	0.434	0.253	0.752	0.354	0.766	0.986

<b>HZL NO</b>	<b>SIFT</b>	<b>BRISK</b>	<b>ORB</b>	<b>SURF</b>	<b>FAST</b>	<b>HARRIS</b>	<b>MSER</b>
<b>370</b>	0.210	0.569	0.118	0.799	0.505	0.736	1
<b>371</b>	0.096	0.436	0.133	0.782	0.315	0.726	0.966
<b>372</b>	0.184	0.621	0.061	0.739	0.554	0.715	0.962
<b>373</b>	0.107	0.101	0.182	0.719	0.417	0.672	0.938
<b>374</b>	0.258	0.179	0.095	0.624	0.622	0.737	0.962
<b>375</b>	0.201	0.355	0.172	0.646	0.540	0.717	0.948
<b>Ort:</b>	<b>0.243</b>	<b>0.328</b>	<b>0.174</b>	<b>0.641</b>	<b>0.689</b>	<b>0.791</b>	<b>0.865</b>