

T.C.  
İSTANBUL AYDIN ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



**KARAKTER TANIMA SİSTEMLERİ VE OSMANLICA İÇİN BİR  
UYGULAMA**

**YÜKSEK LİSANS TEZİ**

**Utku Can AYTAÇ**

**Bilgisayar Mühendisliği Anabilim Dalı  
Bilgisayar Mühendisliği Programı**

**Şubat 2016**



T.C.  
İSTANBUL AYDIN ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



KARAKTER TANIMA SİSTEMLERİ VE OSMANLICA İÇİN BİR  
UYGULAMA

YÜKSEK LİSANS TEZİ

Utku Can AYTAÇ  
(Y1313.010016)

Bilgisayar Mühendisliği Anabilim Dalı  
Bilgisayar Mühendisli Programı

Tez Danışmanı: Prof. Dr. Ali GÜNEŞ

Şubat 2016





T.C.  
İSTANBUL AYDIN ÜNİVERSİTESİ  
FEN BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

**Yüksek Lisans Tez Onay Belgesi**

Enstitümüz Bilgisayar Mühendisliği Ana Bilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı Y1313.010016 numaralı öğrencisi **Utku Can AYTAÇ**'ın "KARAKTER TANIMA SİSTEMLERİ VE OSMANLICA İÇİN BİR UYGULAMA" adlı tez çalışması Enstitümüz Yönetim Kurulunun 29.12.2015 tarih ve 2015/30 sayılı kararıyla oluşturulan jüri tarafından *ayhan* ile Tezli Yüksek Lisans tezi olarak *başarıyla* edilmiştir.

Öğretim Üyesi Adı Soyadı

İmzası

Tez Savunma Tarihi :09/02/2016

1)Tez Danışmanı: Prof. Dr. Ali GÜNEŞ

.....  
*Ali Güneş*

2) Jüri Üyesi : Yrd. Doç. Dr. Vassılya UZUN

.....  
*Vassılya Uzun*

3) Jüri Üyesi : Yrd. Doç. Dr. Ferdi SÖNMEZ

.....  
*Ferdi Sönmez*

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

## YEMİN METNİ

Yüksek Lisans tezi olarak sunduđum “**KARAKTER TANIMA SİSTEMLERİ VE OSMANLICA İÇİN BİR UYGULAMA**” adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Bibliyografya’da gösterilenlerden oluştuđunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. (09/02/2016)

Aday/İmza





*Canım Annem ve Babama,*





## ÖNSÖZ

Bu çalışma, İstanbul Aydın Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak hazırlanan “KARAKTER TANIMA SİSTEMLERİ VE OSMANLICA İÇİN BİR UYGULAMA” isimli tezi içermektedir. Çalışmalarımın her aşamasında bilgi ve deneyimleri ile yardımcı olan kendisinden çok şey öğrendiğim, karşılaştığım problemlerde özgün fikirlerinden çokça istifade ettiğim danışmanım Sayın Prof. Dr. Ali GÜNEŞ'e teşekkür ederim. Yoğun iş tempomun haricinde onlara ait olduğuna inandığım vakitlerden çalarak tez çalışmamı tamamladığım için bana anlayış ve sabır gösteren, müdürüm Sayın Sinem Ucal ÇEPNİ'ye sonsuz teşekkür ederim

ŞUBAT 2016

Utku Can Aytaç

## İÇİNDEKİLER

### Sayfa

ÖNSÖZ.....	iv
İÇİNDEKİLER .....	v
KISALTMALAR .....	vii
ÇİZELGE LİSTESİ.....	viii
ŞEKİL LİSTESİ.....	ix
ÖZET.....	x
ABSTRACT .....	viii
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1 Tezin Amacı.....	2
1.2 Literatur Araştırması.....	2
<b>2. BÜTÜNLEŞİK VERİ</b> .....	<b>9</b>
2.1 Karakter Tanıma Sistemlerinin Genel Yapısı ve Kullanım Alanları .....	9
2.2 Osmanlıcanın Yapısı .....	17
2.3 Osmanlı Arşivleri .....	22
2.4 Osmanlıcanın Latin Harflerinden Farkı .....	24
<b>3. GÖRÜNTÜ İŞLEME</b> .....	<b>27</b>
3.1 Ön İşleme .....	28
3.2 Özellik Çıkarma .....	41
3.3 Dilimleme .....	43
<b>4. KULLANILAN YÖNTEMLER</b> .....	<b>47</b>
4.1 Özellik Çıkarma .....	47
4.2 Şablon Eşleme .....	47
4.3 Destek Vektör Makinaları .....	48
4.3.1 Doğrusal DVM için en uygun ayırıcı düzlem.....	49
4.3.2 Doğrusal olarak ayıramayan durumlar.....	51
4.4 Dalgacık analizi.....	53
<b>5. UYGULAMA VE MATERYAL</b> .....	<b>59</b>
5.1 Kullanılan Yazılımlar .....	59
5.2 Uygulama .....	61
5.2.1 Satır segmentasyonu .....	61
5.2.2 Kelime segmentasyonu .....	62
5.2.3 Akıllı okuma sistemi geliştirilmesi .....	64
5.2.4 Benzerlik matrisi .....	67
5.2.5 Programın algoritması.....	68
5.2.6 Programın arayüzü .....	71
<b>6. SONUÇ VE ÖNERİLER</b> .....	<b>75</b>
6.1 Test Sonuçları .....	75
6.2 Tartışma ve Öneriler .....	81
<b>KAYNAKLAR</b> .....	<b>85</b>
<b>EKLER</b> .....	<b>87</b>
<b>ÖZGEÇMİŞ</b> .....	<b>93</b>

## KISALTMALAR

<b>Bkz.</b>	: Bakınız
<b>BM</b>	: Benzerlik Matrisi
<b>Dİ</b>	: Doküman İşleme
<b>DTV</b>	: Dikey Tarama Vektörü
<b>DVM</b>	: Destek Vektör Makinesi
<b>Gİ</b>	: Görüntü İşleme
<b>K-EYK</b>	: K-En Yakın Komşuluk
<b>KFSD</b>	: Kısa Süreli Fourier Dönüşümü
<b>KT</b>	: Optik Karakter Tanıma
<b>NB</b>	: Naive Bayes
<b>OCR</b>	: Optical Character Recognition (Optik Karakter Tanıma)
<b>RO</b>	: Rastgele Orman
<b>SB</b>	: Satır Belirleme
<b>SP</b>	: Satır Parçalama
<b>VTD</b>	: Vertical Traverse Density (Dikey Çizgi Hareketliliği)
<b>YSA</b>	: Yapay Sinir Ağı
<b>YTV</b>	: Yatay Tarama Vektörü

## ÇİZELGE LİSTESİ

**Sayfa**

<b>Çizelge 6.1.</b> Satır Segmantasyonu Sonuçları .....	80
<b>Çizelge 6.2.</b> Kemile Segmantasyonu Sonuçları.....	81



## ŞEKİL LİSTESİ

### Sayfa

Şekil 2.1: Makineyle baskılanmış yazıya ait, düşük kaliteli görüntü .....	11
Şekil 2.2: Birleşik el yazısına bir örnek.....	12
Şekil 2.3: Osmanlıca harfleri .....	18
Şekil 2.4: Osmanlıca'da harflerin baş, son ve ortada gösterimi .....	19
Şekil 2.5: Osmanlıca Alfabe. ....	21
Şekil 2.6: Osmanlıca Harflerin Başta, Ortada ve Sonda Yazılışları .....	22
Şekil 3.1: İkili Görüntü .....	27
Şekil 3.2: Medyan filtre uygulama örneği .....	31
Şekil 3.3: Orjinal görüntü (üst sol) salt & pepper gürültüsü (üst sağ), ortalama alma teoremiyle (alt solda) ve medyan filtresi ile (alt sağda) .....	32
Şekil 3.4: Orijinal görüntü ve filtre uygulamalı görüntüler.....	32
Şekil 3.5: Latin Harfleriyle Örnek Satır Belirleme.....	37
Şekil 3.6: Silme kuralları .....	40
Şekil 3.7: Muhafaza şablonlarının gösterimi .....	41
Şekil 4.1: Özellik Çıkama Aşamaları .....	47
Şekil 4.2: İki boyutlu uzayda iki sınıf veri kümesi örneği.....	50
Şekil 4.3: Eğitim verilerinin doğrusal olarak ayrılmadığı durum örneği.....	52
Şekil 4.4: Dalgacık dönüşümünde bir işaretin altbandlara ayrışımı .....	55
Şekil 4.5: İki boyutlu ADD kullanılarak tek seviyeli görüntü ayrıştırma işlemi.....	57
Şekil 4.6: ADD ile 2 seviyeli görüntü ayrıştırma .....	57
Şekil 5.1: Toplam Satır Sayısı .....	61
Şekil 5.2: Satır Örnek.....	62
Şekil 5.3: Satır Örnek2.....	62
Şekil 5.4: Kayıt Edilen Satırlar .....	62
Şekil 5.5: Satır ve Kelime Sayısı Toplamı.....	63
Şekil 5.6: Bölünen Kelimeler.....	64
Şekil 5.7: Eğitim kümesinde kullanılan bazı harfler.....	65
Şekil 5.8: Matlab Ortamında Eğitim.....	65
Şekil 5.9: Orjinal Metin .....	66
Şekil 5.10: Ocr işlemine tabi tutulan metin .....	66
Şekil 5.11: Ocr işlemine tabi tutulan metin2 .....	67
Şekil 5.12: Algoritma1 .....	70
Şekil 5.13: Algoritma2.....	71
Şekil 5.14: Ana Menü. ....	72
Şekil 5.15: Kelime Düzeltme.....	72
Şekil 5.16: Kelime Ekleme .....	73
Şekil 6.1: Örnek1.....	75
Şekil 6.2: Örnek2.....	76
Şekil 6.3: Örnek3 .....	76
Şekil 6.4: Seçilen Metin.....	77

<b>Şekil 6.5:</b> Ana Menü .....	77
<b>Şekil 6.6:</b> Çeviri Ekranı .....	78
<b>Şekil 6.7:</b> Çeviri Sonucu .....	78
<b>Şekil 6.8:</b> Kelime Ekranı.....	79
<b>Şekil 6.9:</b> Bulunamayan Kelimeler.....	79
<b>Şekil 6.10:</b> Bulunan Kelimeler.....	80
<b>Şekil A.1:</b> Osmanlıca Matbu Gazete Örneği .....	89
<b>Şekil A.2:</b> Takvim-i Vekayi Gazetesi Örnekleri.....	90
<b>Şekil A.3:</b> Takvim-i Vekayi Gazetesi Örnekleri2.....	91



## KARAKTER TANIMA SİSTEMLERİ VE OSMANLICA İÇİN BİR UYGULAMA

### ÖZET

Bu tezde Karakter Tanıma Sistemleri ele alınmıştır. C# programlama dili kullanılarak, Osmanlıca için çeviri sistemi geliştirilmiştir. Osmanlıca harfler, çeviri sistemine ASCII kod olarak girdi üreten bir optik karakter tanıma sistemi tarafından tanınmıştır. Girdi metin yükleme aşamasında, girdi metnin tüm bileşik yapılarının sistem de olup olmadığı kontrol edilir. Dokümanın ve sözlüğün içeriğine göre olası alfabe çevirimlerinin tamamı üretilir. Eğer bir çevirim elde edilemezse metnin tanınmamış parçaları kullanıcıya sorulur. Kullanıcıyla bilgisayar arasındaki etkileşimi kolaylaştırmak artık günümüzde zorunlu hale gelmiştir. Kâğıt üzerine basılı metinlerin otomatik olarak bilgisayar ortamına aktarılması kullanıcıların zaman sarfiyatını azaltmak açısından çok önemlidir. Karakter Tanıma Sistemleri bilgisayar aracılığıyla, basılı metinlerin okunarak metinlerine dönüştürülmesi işlemidir.

Tez, altı bölümden oluşmaktadır: Giriş, Bütünleşik Veri, Görüntü İşleme, Kullanılan Yöntemler, Uygulama ve Materyal, Sonuç ve Öneriler kısmından oluşmaktadır.

Giriş bölümünde, bu tezin amacı ve bu çalışma hakkında literatür araştırması yapılmıştır. Bütünleşik Veri bölümünde, Karakter Tanıma Sistemlerinin Genel Yapısı, Karakter Tanıma Sistemlerinin Çalışma Prensipleri, Günümüzde Kullanılan Karakter Tanıma Sistemleri, bu sistemlerin kullanım alanları, avantaj ve dezavantajları ile Osmanlıca'nın Yapısı ve Osmanlı Arşivleri incelenmiştir. Görüntü İşleme bölümünde, görüntü işleme sistemlerinin çalışma prensipleri ve özellikleri incelenmiştir. Kullanılan Yöntemler bölümünde, Karakter Tanıma Sistemleri için kullanılan yöntemler belirtilmiştir. Uygulama ve Materyal bölümünde, Uygulamanın özellikleri, programın algoritması ve arayüzü anlatılmıştır.

Sonuç ve Öneriler bölümünde test sonuçları değerlendirilmiş ve görüşler ve öneriler belirtilmiştir.

**Ana Kelimeler:** Karakter Tanıma, Osmanlıca Çeviri, Görüntü İşleme

# CHARACTER RECOGNITION SYSTEMS AND AN A STUDY FOR OTTOMAN

## ABSTRACT

Character Recognition System are discussed in this dissertation research. Translation system of the Ottoman language was developed according to the C# programming language. Ottoman letters are recognized by translation system as an optical character recognition system that is generating input as ASCII code. All compounds of structure of the input text is controlled whether the system during input text installation phase. All of the possible alphabet translations are produced according to content of document and dictionary. The unknown pieces of text is asked the user if a translation can not be obtained. Nowadays, it has become imperative to facilitate of the interaction between computer and users. Automatically, transferred to a computer system of the text printed on paper is important according to time saving for the users.

Character Recognition System that is the process reading of the printed text is converted to text through the computer.

The dissertation research is divided into six sections: Sections consists of Introduction, Integrated Data, Image processing, Assessment Methods, Applications and Materials, Conclusions and Recommendations

Introduction section is mentioned about the aim of this dissertation research and literature review about this study. Integrated Data section is mentioned about general structure of character recognition system, working principle of character recognition system, the current character recognition systems and usage areas of this systems. Ottoman language structure with its advantages and disadvantages and examining the ottoman archives. Image processing section is mentioned about principles and characteristics of the image processing system. Assessment Methods section is mentioned about the methods used for character recognition system are indicated. Characteristics of the application is described algorithm and interface of the program in the Applications and Materials section. Finally, Test results are evaluated and the suggestions and recommendations determined in the Conclusions and Recommendations section.

**Keywords:** Character Recognition, Ottomon Translate, Image Processing



## 1. GİRİŞ

Bilgi, günümüzde insanın ihtiyaç duyduğu en önemli araçtır. Çağdaş insanlar, hayatın her alanında bilgi almakta, bunları başkalarıyla paylaşmaktadır. Yazılı, görsel ve elektronik ortamda her gün bilgi üretilmekte ve insanlarla paylaşılmaktadır. Tüm bu olanlarla birlikte bilgiye duyulan ihtiyaç sürekli artmaktadır. İnsanoğlu son yüzyılda tüm insanlık tarihinden ürettiğinden daha fazla bilg üretmekte, bilgi teknolojileri son yıllarda hızla gelişmekte ve yaygınlaşmaktadır. Bilgiyi elektronik ortama taşımak ve bilgisayar desteğinden yararlanmak, insanın en büyük çabasıdır. Bilginin bilgisayar ortamına aktarılmasında bilgisayarlar ile yapılmaktadır. Çeşitli imge tanıma teknolojileri ile karakterlar bilgisayar ortamında tanınmaktadır. Farklı ihtiyaçlar doğrultusunda farklı imge tanıma sistemleri geliştirilmektedir.

İmge tanıma teknolojilerinin kullanım alanları aşağıdaki gibi sıralanabilir.

- Resim tanıma
- Parmak izi tanıma
- Karakter tanıma
- Kenar tanıma
- Plaka tanıma
- Yüz tanıma vs.

Bu çalışmada Karakter Tanıma sistemleri kullanılarak Osmanlıca için bir çalışma ele alınmıştır. Karakter Tanıma (KT) karakter tanıma sistemlerinde kullanılan, basılı belgeler üzerindeki karakterlerin ya da metinlerin okunarak bilgisayar ortamında dönüştürülmesi işlemidir.

Bu çalışma ile Osmanlıca yazılmış olan matbu metinlerin Türkçe diline çevirisini sağlayan, görüntü işleme tekniğinin kullanıldığı bir yazılım programı, destekleyici bir sözlük ve veritabanı kullanan etkin bir yöntem önerilmiştir.

## 1.1 Tezin Amacı

Bu tezin amacı, Osmanlıca matbu eserlerin, Latin alfabesi harflerine ve dolayısıyla Türkçe diline çevrilmesi için gerçek zamanlı, özgün, bir programın geliştirilmesini hedeflemektedir. Türkiye’de Osmanlıca çeviri ile ilgili çalışmalar yeterli bilimsel ve teknolojik altyapıya sahip değildir. Bu çalışmaların yeterli olmamasının asıl nedeni; problemin öneminin kavranmamış olması, sadece harf tanıma ile Osmanlıca’nın çözülebileceğine inanılması, disiplinler arası ekip çalışmasının sürekliliğinin olmaması, çalışmalarda yeterli sabır ve ciddiyetin gerekliliğinin algılanamamasıdır. Bu tez ile birlikte konunun mühendislik ve sosyal bilimler açısından bütünsel analizi sağlanacak ve Osmanlıca metinlerin taranarak bilgisayar ortamına, sık kullanılan belge formatlarında (PDF, vb) aktarılması gerçekleştirilecektir. Kullanılacak olan birden fazla algoritma sayesinde, az hata oranı ile çalışacak, çeviri önerilerini alternatifli olarak sunabilecek bir program geliştirilmesi hedeflenmektedir. Konu ile ilgili özgün olarak tasarlanacak olup, taratılmış olan Osmanlıca matbu yazıların Türkçe diline gerçek zamanlı olarak çevrilmesi hedeflenmektedir. Böylece iki milyona yakın Osmanlıca metin Türkçe diline çevrilerek geçmiş bilgi birikimi yeni nesillere kazandırılmış olacaktır. Ayrıca kaybolmuş eserlerimizin ve kültürel mirasımızın, oluşturulacak yazılımı kullananlar vasıtasıyla tekrar ülkemize dönmesi, bu alanda disiplinler arası çalışmayla oluşturulmuş bir envanterin oluşturulması sağlanacaktır.

## 1.2 Literatür Araştırması

Literatürde doğrudan Osmanlıca ya da Osmanlıca belgeler için yapılan çalışmaların sayısı çok azdır. Bu sebeple her ne kadar Osmanlıca Arapçadan farklı bir dil olsa da, genel yapısı itibariyle Arap dillerine olan benzerliklerinden dolayı, Arapça çeviri ile ilgili literatürün de taranması faydalı görülmüştür.

Atıcı ve Yarman-Vural’ın çalışmasında [5] karakter darbelerine zincir kodu dönüşümü uygulanmış, tanıma işlemi içinse gizli markov modeli kullanılmıştır.

Şaykol ve Ark.’nın [7] çalışmasında ise amaç, öncelikle eldeki belge kümesinden bir “simge” kütüphanesi oluşturulması ve sonrasında da her belgenin içerdiği simgeleri listeleyen bir metin dosyasına dönüştürülmesidir. Simge çıkartıcı alt modülü temel olarak şu işlevleri gerçekleştirmektedir: Belge-resmi üzerinde bulunan tüm izole

işaretlerin bulunup çıkartılması işlemi literatürde bilinen algoritmalar kullanılmıştır. Çıkartılan işaretlerin simge kütüphanesinde daha önce kaydedilmiş simgelerle karşılaştırılarak benzer simgenin tespit edilmesi. Bu amaçla bahsedilen çalışmada bir yönlendirilmeyen yaklaşım kullanılmıştır. Bu yaklaşımda simge çıkartıcı modül Osmanlıca karakterler hakkında herhangi bir ön bilgiye sahip değildir, yani başlangıçta simge kütüphanesi boştur. Modül resimden her bir izole işareti çıkardığında kütüphaneye o ana dek kaydedilmiş simgelere bakar, eğer kayıtlı simgelerden biri eldeki işarete belli bir eşik değerinin üzerinde benziyorsa o işaretin bilinen bir simge olduğuna hükmeder, koordinatlarını ve hangi simge olduğunu kayıt ederek bir sonraki işarete geçer. Eğer eldeki işaret öncekilere benzer bulunmamışsa bunu yeni bir simge olarak kütüphaneye ilave eder. Böylece simge kütüphanesi sabit bir stildeki alfabe harflerini değil, belgelerde bulunma durumuna bağlı olarak harflerin çeşitli biçimlerdeki hallerini uyarlamalı bir şekilde kapsayabilecektir[3].

Keselj ve arkadaşları (2003), yazar tanınması yaptıkları çalışmalarında değişik boyutlarda n- gram yöntemini kullanmışlar ve İngilizce, Yunanca ve Çince'ye uygulayarak karşılaştırmalı sonuçlarını açıklamışlardır [12].

Diri ve Amasyalı (2003), Türkçe dokümanlar üzerinde yaptıkları çalışmada metin içeriğine bağlı sınıflamada Navie Bayes yöntemini, 22 farklı stil özelliğini kullanan diğer bir sınıflamada ise kendi geliştirdikleri 'Automatic Author Detection for Turkish Text' metodunu kullanmışlardır [13].

Doğan ve Diri (2006), Türkçe bir dokümanın türü yazarı ve cinsiyetini belirlemek için üç farklı veri seti üzerinde yaptıkları çalışmada n-gram yöntemini kullanmışlardır. Naive Bayes (NB), Destek Vektör Makinesi (DVM), Rastgele Orman (RO), K-En Yakın Komşuluk (K-EYK) gibi sınıflandırıcıların yanında kendi geliştirdikleri Ng-ind' yöntemini de kullanarak testler yapmış ve başarı performanslarını birbirleri ile karşılaştırmışlardır. 'Ng-ind' yönteminin cinsiyet ve tür belirlemede diğer yöntemlere göre daha iyi sonuçlar verdiğini gözlemlemişlerdir [15].

Khreisat (2006), Arapça metinlerin sınıflandırılması için çalışmıştır. Bunun için Arapça çevrimiçi gazetelerden derlediği metinlerde n-gram frekansı kullanarak "Manhattan Benzerliği" ve "Dice" benzerlik ölçütünü kullanmıştır [2].

Huang ve arkadaşları (2003), resim tabanlı belgelerde sınıflandırma için kelime şekil analizine dayalı bir yöntem teklif etmişlerdir. Bu çalışmada, OCR yerine doğrudan doğruya kelime resim özellikleri çıkarılır ve belge görüntüleri kelime birimleri halinde dik yönde parçalanır. Daha sonra dikey çubuk desenleri elde edilerek doküman özellik vektörleri oluşturulur. Son olarak doküman özellik vektörlerinin skaler çarpımlarından ikili benzerlik ölçütleri bulunarak sınıflandırma yapılır [4].

Özhan (2005), çalışmasında el yazısı ayrı yazılmış Osmanlıca harfleri tanımaya ilişkin bir yapay sinir ağı (YSA) tasarlamış ve uygulamıştır. Osmanlıca harflerin yazılı olduğu taranmış bir belge görüntü işleme teknikleri kullanılarak sayısal verilere dönüştürülmüştür. Verilerin düzenlenmesi için bir normalizasyon işleminden geçirilerek, YSA için giriş-çıkış değerleri elde edilmiştir. YSA'nın eğitim işlemi uygulamaları yapıldıktan sonra deneysel sonuçların değerlendirilmesi yapılmıştır [14].

Tan ve arkadaşları (2006), OCR kullanmadan görüntülü belgelere erişim için bir metod önermişlerdir. Belgeler karakter bölümlerine ayrılarak, her bir karakterin resim görüntüsü alınır. Her sütundaki dikey çizgi sayısı ile her satırdaki yatay çizgi sayısı birer vektör şeklinde alınarak karakter resimlerinin görüntü özellikleri çıkarılmıştır. Bu özelliklere bağlı olarak bir n-gram tabanlı belge vektörü oluşturularak, belgeler arasında metin benzerliği, vektörlerin skaler çarpımı ile ölçülmüştür [8].

Yalnız ve arkadaşları (2009), resim formatında taranmış, basılı Osmanlı belgelerindeki çalışmalarında bağlı harfler için entegre edilmiş segmentasyon ve bir karakter tanıma modeli önermişlerdir. Önerilen yöntem %90 doğruluk sağlamıştır [7].

Zaki ve arkadaşlarının (2010), Arapçada yazılmış belgeler üzerindeki çalışmaları; klasikleşmiş Boole modelini baz alarak; belgeyi meydana getirmiş terimlerin aralarında bulunan yakınlık durumunun, fuzzy ilkesi vasıtasıyla anlamlandırıp sınıflandırarak yapılmasını önermiştir.[1]

Rath ve Manmatha tarihsel belgelerin alımı için bir kelime-görüntü eşleştirme tekniğini önerdi ve bu karakterlerin eşleşmesinde önemli bir çalışma gerçekleştirdi. Gerçekleştirdiği bu çalışmada kelimelerin mürekkep farklılıklarının, font ve yazı stillerinin, çeviride önemli rol oynadığını anlattı [4]

Saykol ve arkadaşları Osmanlı belgelerinin içerik tabanlı erişim için sıkıştırma fikrini

kullanmışlardır. Çalışmada veri kümesi içinde karakterler ve semboller için, bir kod kitabı oluşturulur ve bu kod çizelgesi yardımıyla sorguları işlenir.[15]

Bunların yanı sıra, el yazı karakterlerini tanımada kullanılan diğer çalışmalara da göz gezdirmek doğru olacaktır.

Fukushima'nın (1982) gerçekleştirdiği çalışmada, el yazısının rakamları tanınması hususunda ilk örnek verilmiştir. Çalışma içeriğinde, konumun yahut şeklin deformasyonu ile alakalı değişimlerin karşılandığı, istatistiksel bir algoritmanın geliştirilmesi sağlanmıştır. Algoritmanın, piksellerin konumuna bağlı durumlarını referans alarak, şekiller ortaya çıkan muhtemel farklılıkları incelemesi ve karakterlerin ölçeklerden ve konumlardan ziyade şekle sınıflandırılmasını sağladığı görülebilir.(2)

Lam ve Suen'in çalışması (1988) el yazısıyla yazılmış olan posta kodlarını otomatik şekilde değerlendirmeyi amaçlayarak bir tanıma sistemini geliştirmiştir. Sistem, karakterlerdeki yapı özelliklerinin kullanılmasıyla çalışmaktadır. Söz konusu çalışma, sistemdeki hız yönünden, iki aşamadan oluşan bir karar tekniğini kullanır. İlk aşama, karakterdeki temel özelliklerin ve süratli tanıma sisteminin denenmesi gerçekleştirilir. Bu ilk safhadan sonra, ortaya çıkan tanınabilirlik oranı düşükse, diğer aşamada bu oranı artırabilmek için daha detay içerikli çalışmalara girilir.(3)

Fukushima'nın (1988) çok katmanlı ve hiyerarşik yapısıyla tasarlanan bir ağ yapısını(neocognitron) kullanmak suretiyle, piksel tabanlı bir tanıma sistemini geliştirdiği çalışmada, giriş katmanında uygulamaya tabi tutulan verilerin, ağda bulunan bütün tabakalarda aşamalı olarak işlenmesi ve bağlantı üzerinde bulunan başka hücreler üzerine aktarılması sağlanır. Bütün katmanlar, karakterdeki değişik bir özelliği saptamak için kullanılır. Katmanların ilerlemesiyle seçicilik artar ve karmaşıklaşan özellikler ayrıştırılarak, nihai olan son katman üzerinde tanımın gerçekleştirilmesi sağlanır.(4)

Mai ve Suen'in (1990) karakterlerin yapı özellikleri üzerinden elde edilecek özelliklerin, karakterlerin sınıflandırılmaları uyarınca, tekrarlanması/oluşması durumlarının sıklığıyla alakalı istatistiksel bilgilerin toplandığı bir veri tabanı oluşturdukları çalışmalarında, sınıflandırma için YSA kullanımının yanı sıra tanımlanamayan örnekleri teşhis amacıyla, üretilmiş olan veri tabanı uyarınca kararların alınması sağlanabilmektedir. (5)

Le Cun'un (1990) çalışması, rakamların karakterlerine ait normalleşmiş formları kullanmak suretiyle, piksel tabanda bir tanıma uygulamasını meydana getirmiştir. Uygulama, 4 katmandan oluşan geri yayılım ağını(backpropagation network) içermektedir.(6)

F. Kimura ve M. Shridhar'ın (1991) çalışmaları, rakamsal içerikteki karakterleri tanıma için iki nitelik setinin birleştirilerek kullanılmasına yöneliktir. Bu setlerden birincisi, karakterin sınır eğrisi üzerinden üretilmiş olan yerelleşmiş doğruyu bileşenini ifade eder. Diğer set ise, karakterin dış sınırının eğrisindeki sağ ve sol profiller üzerinden elde edilmiş olan yapı özelliklerini ifade eder. Sınıflandırmanın yapılma aşaması ise tanımadaki başarıyı yükseltmeyi amaçlayarak çoğul yöntemi kullanmayı seçmiştir. Sınıflandırmada illuminated, istatistikî yöntemlerin ve karar ağları kullanımının sonucunda, bu verileri, verimliliğe bağlı olarak ayrı ayrı değerlendirmektedir. Yüksek tanıma oranının (%98) sağlanmasıyla beraber, işlemdeki yük fazlalığının da tespit edildiği görülmüştür.(7)

S. Whan Lee'nin (1995) çalışması ise piksel tabanda tanıma yöntemini benimsemektedir. Bunun yanında, maskeleyerek karakterdeki alt bölgelere göre ayrı ayrı yerelleşmiş doğruyu bilgilerinin de çıkarılması söz konusudur. Tüm bunlarla beraber, alt bölge doğrularındaki nitelik uyarınca, karakterlerin tamamını temsil etmekte olan matris bilgileri üretilir. Sınıflandırmanın yapıldığı aşamada ise, ele geçen yerel ve global bilgiler, grupların yapıları uyarınca, bütün grupları ayrı ayrı karşılamakta ve ayrı ayrı kümeler şekliyle bağlantılar içermekte olan çok katmanlı ihtiva eden yapaylaşmış sinir ağları tasarlanır. (8)

J. Basak (1995) karakterdeki iskelet formlarının doğruca giriş şeklinde kullanıldığı, çok katmanlı yapaylaşmış sinirsel ağları kullanmak suretiyle, el yazılarının karakterlerinin tanınmasını gerçekleştiren bir sistem üretmiştir. Karakterdeki çizgi yönlü bileşenlerin özneliklerinin oluşumu, öz bağlantımı (connectionist) olan ağ yapılarının üstünde gerçekleşir. 6 katmanlı ağda, eğitimin yapıldığı aşama, her katmanla ilgili özneliklerin seçiciliğini meydana getirir. En son katman ise, söz konusu bilgileri entegre ederek, tanımayı sağlar. Özelliklerin çıkartımı doğrultusunda düzenlenen özellikli hücresel bağlantılar vasıtasıyla oluşturulmuş olan ağ yapısı, görüntünün matrisinin oluşturulduğu noktaların sayısı miktarınca girişin kullanılmasını gerekli kılar. (9)

X. Li (1997) yazmanın yapıldığı anda (on-line) karakterlerin tanınma yöntemini

geliştirme çalışması yapmıştır. Yöntem uyarınca, karakterleri oluşturmuş olan çizgilerin üzerindeki baskınlaşmış noktaların belirlenmesi sağlanır. Söz konusu noktalar, köşedeki noktalar ve düz çizgilerdeki orta noktalar şeklinde tanımlanır. Sınıflandırmanın benzerliğe dayalı ölçümler vasıtasıyla gerçekleştirildiği görülmektedir. (10)

Y.S. Hwang'ın (1997) çalışmaları ise piksel tabanda verisel girişleri benimsemektedir. Rakamsal karakterlerin üstünde yapılan söz konusu çalışma, sınıflandırıcılık açısından, RBF (RadialBasisFunction) ile eğitilmiş olan ağ yapısını kullanır. (11)

D. Cheng ve H. Yan'ın (1998) rakamsal karakterlerin üstünde gerçekleştirmiş oldukları çalışma, karakterlerin sınır eğrilerinin temel alındığı bir yaklaşımı tercih eder. Sınır eğrileri kullanılmak suretiyle çevresel uzunluğun, alan oranlarının, piksel uzaklık parametrelerinin ve buna benzeyen yapı özelliklerinin yanı sıra, sınır eğrilerini Fourier açılımının vasıtasıyla üretilmiş olan bilgilerin tanınması amacıyla da kullanılmaktadır. Sınıflandırıcılık için ise karar ağaçları kullanılır.(12)

K. Yaman'ın ve diğerlerinin (2001) çalışması, Ankara Hızlı Raylı Sistemde kameraların algıladığı gri seviyedeki görüntülerin, bilgisayara aktarımı ile alakalıdır. Ele geçirilmiş olan sayısal değerler ise modellemede kullanılmıştır.(13)

A.Erdem'in ve Emre Uzun'un (2005) çalışmaları ise, yapaylaşmış sinirsel ağlar vasıtasıyla, Türkçe Times New Roman, Arial ve El yazısı karakter tanımlarıyla alakalı bir çalışmaları olmuştur. Çalışmada el yazısı karakter tanımada başarı elde edilmiştir.(14)





## 2. BÜTÜNLEŞİK VERİ

### 2.1 Karakter Tanıma Sistemlerinin Genel Yapısı ve Kullanım Alanları

El yahut makine vasıtasıyla yazılıp hazırlanan yazıların bilgisayarca tanınması işine, genelleşmiş manasıyla Karakter Tanıma (KT) adı verilir.

Karakter tanımayı; metinsel tabandaki örüntülerin makinece işlenip anlam ifade eden çıktılar şeklinde üretim olarak da tanımlamak mümkündür.

Dökümantasyonların, makinelerce okunabilmesi ve işlenebilmesine yönelik formlara kavuşturulması amacıyla kullanılan, tarama, kamerayla görüntü elde etme yahut dokunmaya hassas ekranlara dosdoğru yazma gibi yöntemlerin kullanıldığı söylenebilir.

Karakter Tanıma Sistemleri (KT),günümüzde günlük hayatta kullanılan birçok akıllı sistemlerden biridir. KT sistemleri masaüstü yayıncılık, mahkemeler, vergilendirme ve tahsilat, personel kayıt yönetimi, nüfus sayımı formlarının işlenmesi, çek işleme, ödeme işlemleri, sipariş işlemleri gibi alanlarda kullanılmaktadır. Kâğıt bazlı formların bilgisayar tarafından kullanılabilir formatlara çevrilme gereksinimi her geçen gün artmakta, buda beraberinde yazı formatını okuyup tanıyabilecek yazılımların ihtiyacını doğurmuştur[1]. KT sistemlerinin sağladığı kolaylıklar, diğer birçok iş alanında bu teknolojilerin hızla uygulanmaya başlanmasına yardımcı olmuştur. Örneğin; mektupların üstlerindeki adreslerin tanınip posta koduna göre otomatik olarak ayrıştırılması, bankalara yollanan çeklerin otomatik olarak tanınip gerekli hesap işlemlerinin elektronik ortamlarda gerçekleştirilmesi gibi projeler karakter tanıma teknolojileri ile gerçekleştirilmektedir.

Karakter Tanıma, elektronik ortamda taranmış karakterların üzerindeki karakterlerin veya metin bilgilerinin çeşitli algoritmalar yardımıyla okunarak bilgisayar metinlerine dönüştürülmesi işlemidir[3].Karakterlar önce cihazlar yardımıyla taranarak resim formatında bilgisayara kayıt edilmelidir. Bundan sonra belgedeki karakter tespit edilmeli ve tanınan karakterler birleştirilerek bir metin halinde sunulmalıdır.

KT sistemleri, kâğıt vb. üzerindeki siyah noktaları yani harfleri, rakamları, sembolleri algılayabilecek bir yapıya sahiptirler. Her kelimeyi ayrı ayrı tutarak ve sayfayı satırlara bölerek analiz ederler. Programın kendi veritabanı, her bir karakter için tanımlanmış bazı parametlerele doludur.

Karakter Tanımının çalışma prensibi birkaç adımdan oluşmaktadır. Bunlar sırasıyla; Tarama, Eşik Çıkartma, Segmentasyon, Normalizasyon, Sınıflandırma kısımlarından oluşmaktadır.

Tarama aşamasında metin imgesi tarayıcı veya dijital kamerayla taranarak siyah beyaz imge elde edilmektedir. Tanıma işleminin kolaylığı açısından tarama kalitesinin yüksek olması önemlidir. Buna rağmen kaliteli tarama çok nadiren elde edilir. Genellikle olumlu sonuç almak için belgeler 100 dpi ve üzeri çözünürlük kalitesi ile taranmalıdır.

Eşik çıkartma işleminde genel metod pixel histogramlarının kullanılması prensibidir. Giriş imgesindeki aynı rengi temsil eden piksellerin toplam sayıları bulunarak histogram oluşturulur. Bu histogramlara dayanarak 0 ve 1 arasındaki değer hesaplanır. Bu değer, genelde histogramda iki zirve arasındaki aralıkta seçilir.

Segmentasyon aşamasında her bir karakterin imgesi ikili imgeden ayrılır. Ayrılan resim iki yönlü dizin halinde sunulur. Normalizasyon aşamasında resimdeki karakterlerin belirli bir ölçüye düşürülmesi önemlidir. Bu kısımda 3 farklı yöntemden yararlanılır: İnceltme, genişletme. İnceltme/Genişletme işlemlerinde karakter imgesinin her parçası belli ölçülere kadar inceltir ve gerekirse yeniden genişletilir. Böylece, karakter belli bir standart ölçüye getirilir.

Karakterin imgesi üzerinde incelenen, gerekli işlemler yapıldıktan sonra tanıma kısmına geçilir. Karakter karşılaştırılması, önceden sisteme tanımlı olan ve veri tabanında bulunan karakter imgelerine göre sınıflandırılması ile gerçekleşir.

Karakter tanıma için yapılan çalışmaları ise üç farklı tip altında inceleyebilmemiz mümkündür. Bunlar;

- Makinelerde yapılmış baskı karakteri tanıma
- Ayrık şekildeki el yazılarındaki karakterleri tanıma
- Birleşik şekildeki el yazılarındaki karakterleri tanıma şeklinde sıralanabilir.

El yazılarında karakterlerin tanınması işlemi ise kendi içerisinde;

- Çevirimiçi(on-line) tanımlar
- Çevirimdışı(off-line) tanımlar şeklinde sınıflandırılır.

Makinelerde yapılan baskıların, el yazısıyla yapılan baskılarla, karakterleri açısından mühim farklılıkları bulunmaktadır. El yazı karakterlerinde kendi aralarında görülen değişimlerin, makine baskı değişikliklerine oranla çok daha fazla ve farklılaşmış olduğu söylenebilir. Makine baskı değişikliklerinin sadece fontlarda ve yazı tiplerindeki değişiklikleri ifade ettiği görülürken, el yazılarında çok daha farklı ve kapsamlı değişikliklerin görülebilmesi mümkün olmaktadır.

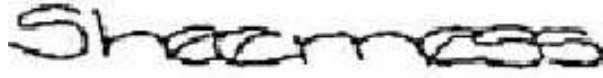
Dünya üzerinde hiçbir yazar, el yazısı açısından bir diğeriyle bire bir örtüşmez. Hatta aynı yazar tarafından, farklı farklı zamanlarda yazılmış olan yazılarda bile önemli değişikliklerin görülebilmesi mümkündür. Karakterler arasındaki büyüklük-küçüklük durumlarının farklılık göstermesinin yanı sıra, karakterlerde şeklen farklılaşmalar da gözlemlenebilmektedir. Yazıyı yazanın içerisinde bulunmakta olduğu ortamsal şartlar, yazarın moraline veya sağlığına bağlı durumların dahi, yazının biçimsel özelliklerine etkisinden bahsetmek mümkündür. El yazısını tanıma konusunda karşılaşılan problemlerin ve el yazısını tanıma konusunun henüz tam bir netliğe kavuşmamasının sebeplerinden bazılarını da bu etmenler oluşturmaktadır. Bu sebeplerle, makinelerde yapılmış baskıları tanıma oranının, el yazısını tanıma oranından daima fazla olması kaçınılmazdır.

Tanıma sistemi karakteri tanıyabilmek için, karakterleri görüntülerden ayırıştırma işlemi yapmak zorundadır. Kelimelerin içerisindeki harflerin birbirlerinden ayrı olmaları nadiren görülen bir durumdur. Bu tarz durumların, sadece karakterleri ayırmada kullanılan kutucuk formlarının varlığı durumunda görülmesi mümkündür. Öte yandan, makinelerin yaptıkları baskılar içerisinde bile karakterler birbirleriyle bitişik şekilde sıralanabilmektedirler. İlk izlenimde, makinelerde yapılan baskı karakterlerinin birbirlerinden kolaylıkla ayrıştırılabileceği düşünülse de, bazı sebeplerin varlığı bu tarz çalışmalarda problemler çıkarabilmektedir.



**Şekil 2.1.** Makineyle baskılanmış yazıya ait, düşük kaliteli görüntü (25)

Şeklin de göstermiş olduğu gibi, makine baskılarda da görüntünün kalitesinin her daim düzgün olmayabileceği durumlar vardır. Nokta vuruş tekniğiyle çalışan yazıcıların, bilhassa yüksek hızda çalışanlarında, aşırı gürültü içeren görüntüleri meydana çıkarabilmektedir. İki karakterin arasındaki görüntünün, bir piksel boyutundan küçük olması durumunda karakterlerin bitişik şekilde görüntülenmesi söz konusu olmaktadır.(15)



**Şekil 2.2.** Birleşik el yazısına bir örnek (25)

Şekil 2.2’de açıkça görülmekte olduğu gibi; birleşikleşmiş karakter probleminin çok daha kompleks bir haldedir. İnsanların yazı yazarken, karakterleri birleşikleşmiş şekilde yazmaları, onlar açısından son derece olağan durumdadır. Bu tip yazılar içerisinde, karakterlerin aralarında boşluğun olmamasının yanı sıra bir takım karakterlerin de üst üste binmesi durumunu barındırmaktadır. Söz konusu durumlar da el yazısı karakterinin tanınması zorlaştıracak bir hal ortaya çıkarmaktadır. Karakterlerin tanınmasına yönelik sistemlerin bir kısmı, yalnızca izolasyon sürecinden geçmiş karakter tanımlamaları hususunda çalışmakta, bir kısmı hem dilimlemeyle hem de tanımayla alakalı fonksiyonlar görmekte, bir kısım sistemler ise bu iki aşamayı ayrı ayrı yerine getirebilmektedir.

Kâğıt üzerinde olan yazıları tanımlayabilmek çevirimdışı(off-line) yöntemler sayesinde mümkün olabilmektedir. Bunun için de ilk olarak belgeyi sayısal veri haline dönüştürmek gerekir. Belgeyi analiz etmek, önemli olan aşamaların ilkidir. Bu analiz esnasında belge, öncelikle paragraflara ve cümlelere, ardından bu paragraf ve cümleler de kelimelere bölünmelidir. Şayet ki, söz konusu belge doldurulmuş halde bir formdan oluşuyorsa, aynı şekilde öncelikle analizin yapılması, akabinde ise tanınma amacıyla olan kelimelerin ayıklanması işlemi gerçekleştirilir.(16)

Offline yöntemlerin perspektifinin genişliğine karşın, yöntemler kullanılırken işlemlerin sırası çoğunlukla benzer şekildedir. Bu sıralamayı şu şekilde yapabiliriz;

1. Ön-işleme
2. Dilimleme
3. Öznitelikleri çıkarma
4. Tanıma
5. Son-işleme

Bir aşama olarak ön-işlemeyi oluşturan işlemleri, gürültünün azaltılması, normalleştirim, referansa kaynaklık edecek çizgiyi bulma gibi işlemler şeklinde sıralayabiliriz. İşlem öncesi yapılan sayısallaştırmanın esnasında yazının oluşumunu sağlayan eğrilerin arasında boşlukların oluşması, başka yeni çizgi ve noktaların ortaya çıkması hali meydana gelebilir. Söz konusu problemlerin de filtrelemeyle, gürültünün modellenmesiyle veya morfolojik operatörlerin vasıtasıyla giderilmesi mümkün olabilmektedir. Normalleştirmeye bağlı süreci, yazının karakterlerinin kaynaklık ettiği farklılık durumlarının giderilmesi ve bir standarda getirilmesi işlemi olarak açıklayabiliriz. Yazının düzgün bir çizgide değil de, yukarıya yahut aşağıya eğilimli yazılması durumunun düzeltilmesi normalleştirme sürecinin işlevini teşkil eder. Benzer biçimde, sağ veya sola eğik şekilde yazılmış olan yazıların düzeltilmesini de normalleştirme süreci içerisinde değerlendirebiliriz. Bir diğer normalleştirme faaliyeti ise, boyutları birbirinden farklı olan yazı karakterlerini aynı boyut şekline getirmektir. Bir takım sistemlerin de öznitelikleri çıkarma amacıyla inceltme yöntemi kullanması gerekebilmektedir. Referans çizgisinin ise tüm karakterleri üzerine oturtan çizgi biçiminde tanımlamak mümkündür. Söz konusu işlemin önemi, bir takım karakterleri ayırt etmekle alakalıdır. Örnek olarak “g” harfi ve “9” sayısı, karakter ayrıştırılmasında birbirleriyle karışabilmektedir. Referans çizgisinin varlığı ise bu iki karakterin, çizgideki yerleri sayesinde karışmamasını sağlar.(16)

Bir diğer aşama olan dilimlemede ise sözcükler, rakam ve harflere karşılık olabilecek şekilde parçalanır. Bir takım sistemlerin dilimleme aşamasını, harf arasında bulunan boşluk referanslarıyla yaptıkları görülmektedir. Fakat bu metodun, harfleri parçalara ayırdığı görülmüştür. Bu sebeple de tanımayla dilimlemenin birleştirildiği, yalnızca muhtemel olan karakterlerin denkleğini sağlayan sistemlerin kullanılması önerilir. Başka bir yaygınlaşmış yöntemi ise öncelikle olabildiğince küçük parçalara ayırmak ve sonrasında gizli markov modellerle (HiddenMarkov Model-HMM) birleştirmek şeklinde görebilmekteyiz.(17)

Bir takım sistemler ise, dilimlemeye dair aşamaları yok sayarak, doğruca tanımaya dair aşamalara yönelirler. Dilimlemenin sistemde olup olmamasına göre de öznitelik çıkarmaya dair aşamalar farklılaşmaktadır. Öğrenmenin yapıldığı aşama içerisinde tüm piksellerin işlenmeden kullanılmasıyla uzay büyüklüğü ve gürültü kaynaklı problemler meydana gelebilir. Özniteliklerin çıkarılması aşamasının esas amacı, veriyi kısıtlayarak söz konusu problemleri engellemektir. Çeşitlilik gösteren yöntemlerin izlenebilmesi mümkündür. Fourier, Wavelet gibi dönüşümlerin uygulanması, histogramı yahut izdüşümü temel alan metotlar veyaharflerin çizgileri, eğrileri, köşeleri veya buna benzer basitleşmiş şekillerde ve bütüncül olarak tanımlama yoluna gitmek, bu yöntemlerin bir kısmıdır.(17)

Tanımanın yahut harflerin sınıflandırmasının yapılmış olduğu aşamalarda, çok değişik metotların kullanılabilmesi mümkündür. Yapaylaştırılmış sinirsel ağların, istatistiki yahut kurallarla temellendirilmiş öğrenmenin, şablonları eşleştirmenin bu metotlardan bazıları olduğu söylenebilir. El yazılarının tanınmasında ortaya çıkan en büyük sorunlardan birinin, sözcükleri bulmak amacıyla harfleri tanımanın gerekliliğidir. Fakat hususiyetle, gürültü içeren bir verinin varlığında harflerin, yalnızca içerisinde bulunmuş oldukları bağlamın bilinmesiyle tanımlanabilmesi mümkün olabilmektedir. Söz konusu problemleri bir miktar da olsa azaltabilmek amacıyla bir takım sistemler, tanımlanmış olan harfleri anlamlı bir şekle getirebilmek için, tanım işlemi sonrası bir son-işleme safhası gerçekleştirmektedirler.

Diğer bir tanıma yöntem türü olan çevirimiçi(on-line) tanıma yönteminde ise, karakter yazımının anında tanınması mümkün olmaktadır. Bu yöntemin kullanıldığı işlemlerde, yazımın yapıldığı anda ele geçen bilgilerin kullanılması söz konusudur. Tipikleşmiş şekilde, karakterlerin bir çubuk vasıtasıyla ekrana yazılması ve bilgisayar sisteminin de söz konusu çubuk üzerinden geçmekte olan pikselleri boyamasıyla süreç işler. Söz konusu yöntem, yazıdaki hız, yön ve sıra gibi verilerin elde edilmesini sağlar. Günümüz dünyasında, bilhassa taşınabilir sistemlerin de yaygınlaşması, söz konusu metodu kullanan ticari sistemlerin de kullanımını yaygınlaştırmıştır. Bu ticari sistemler, farklı karakter ve hareketleri de algılayabilmektedir. Örnek olarak; bir kelime üzerinde çizgiyle geçilmesinin akabinde, kelime silme fonksiyonu çalışmakta, el yazılarının karakter tanıma görüntüleri ele alındığı vakit, reel zamanlı yazı bilgilerinin elde edilemeyeceği ve dolayısıyla da bu yöntemin kullanılamayacağı görülecektir.

Çevirimiçi sistemlerin geneli, elektronik tabletlerin üzerinde kalemle yapılan hareket koordinasyonları elde ederek el yazılarının ve çizimlerin otomatikleşmiş şekilde algılanması prensibiyle çalışırlar. Mevcut bir belgeyi bilgisayar ortamına geçirme işi klavyeyle daha basit şekillerde yapıyorsa da, yaratıcılığın gerekli olduğu konular ve belge düzenlemelerinde kalemin ve kâğıdın kullanımı daha çok tercih edilmekte, bu durum da söz konusu çevirimiçi yöntemlerin kullanımını zorunlu kılmaktadır. Öte yandan tabletler sayesinde Latin Alfabeti'nin dışında kalan, Çin, Japon alfabeleri gibi alfabelerin de çizimlerinde son derece kolaylıklar sağlanmaktadır.

Tabletlerdeki teknoloji bazlı sınıflandırma adına iki temel sınıfın varlığı mevcuttur. Bu sistemler; elektromanyetik/elektrostatik ve basınç duyarlı sistemler olarak sıralanabilir. Son yıllar içerisinde yaşanan gelişmeler, girdiler ve çıktıları aynı yüzeyde görünür hale getirmiştir.(18)

Çevirimiçi sistemlerin de çevirimdışı sistemlere metodoloji olarak benzerliği vardır. Çevirimiçi sistemler içerisinde gürültü kirliliğinin kaynağı genel olarak tabletteki yetersizlikten kaynaklanır. Yumuşatmanın ve inceltmenin, yöntemsel olarak kullanımıyla gürültünün azaltılması faaliyeti yapılır.

Dilimleme probleminin kaynağı ise, sistemi çevirimiçi kullanma durumu nedeniyle, farklılıklar ihtiva eder. Çevirimiçi sistemlerdeki en önemli avantajı, sistemi kullananların sistem işleyişine yardımının sağlanabilirliği olarak açıklayabiliriz. Dilimlemede, ilk zaman kullanılan metotlardan birisini, harflerin bitiminde sistemi kullananın bir işaret vasıtasıyla bunu belirtebilmesi şeklinde görebiliriz. Bir takım sistemler ise, kullanıcının harfi kutucuğa yazmasında zorluklarla karşılaşılabilmekte, bu durum da dilimlemenin önemli bir soruna kaynaklık etmesi olarak meydana çıkmaktadır. Başka bir takım metotlarda da çevirimdışı sistemlerde olduğu gibi harflerin arasındaki boşluklar kullanılmakta, bu halde de boşlukları bulmak ve harflerin yazım sırasına konması konularında zaman kayıpları yaşanmaktadır

Günümüzde çok sayıda KT sistemleri geliştirilmiştir. Başarı oranı 100% olmasa da yüksek bir seviyede olarak piyasada bulunan ticari amaçlı KT sistemlerine örnek olarak Fine Reader, AIOCR, JOCR ve başkaları gösterebiliriz. Bu yazılımlar metin tanıma ve belge dönüştürme aracı olup, kullanıcılarına yüksek doğruluk ve verimlilik sunan bir tarama ve yazılımlardır. Bu KT sistemleri kâğıt evrakları, PDF'leri ve metin içeren fotoğrafları tarayarak düzenlenebilir ve içinde arama yapılabilir dosyalar oluşturur.

Günümüzde, özellikle doküman işleme işlemlerinde çalışanlar, geniş anlamda maliyetlerini düşürmek ve verimlilik oranlarını arttırmak için, KT teknolojilerinden yararlanmak zorundadır.

Karakter tanıma konusu ABD'de 1950'li yılların başında geliştirilmiş ve belge sayısallaştırma amaçları için kullanılmaya başlanmıştır. 1960-1970'li yıllardan beri posta servislerinde bu tür sistemleri kullanan yabancı birçok ülkeye göre karakter tanımayı günlük hayatta kullanmak ülkemizde pek yaygın bir konu değildir.

KT teknolojisi genellikle aşağıdaki alanlarda kullanılmaktadır:

- Masaüstü yayıncılık
- Adliyeler
- Vergi daireleri ve Tahsilat
- Personel kayıt yönetimi
- Nüfus sayımı formlarının işlenmesi
- Çek işleme
- Ödeme işlemleri
- Emeklilik Fonu işlemleri
- Sipariş İşlemleri
- Hastaneler

Günümüz ekonomisinde, elektronik doküman yönetimi sistemlerinin, hem iş dünyasında hem de devletin yönetim mekanizmalarında, görüntüleme ve KT teknolojilerini birlikte kullandıklarını görebiliriz. Yaşadığımız bu dönemde KT ile tanıma işlemleri görüntüleme teknolojisi ile birleştirilerek kullanmak önemli bir basamak olmuştur[5].

Karakter tanıma sistemlerinin tarihi bilgisayar tarihinden daha eskidir. İlk zamanlar tüm KT sistemleri analog tabanlı yapılmaktaydı. Bunun nedeni o zamanlar analog/donanım teknolojisinin, dijital/yazılım teknolojisine göre daha çok gelişme göstermesiydi. 1962 senesinde, ilk olarak elektron tüp üzerine 91 kanallı, İngiliz ve Rus alfabesinin tüm harflerini tanıyabilen karmaşık bir KT sistemi geliştirildi fakat henüz o yıllarda ticari bir sistem yoktur [3].



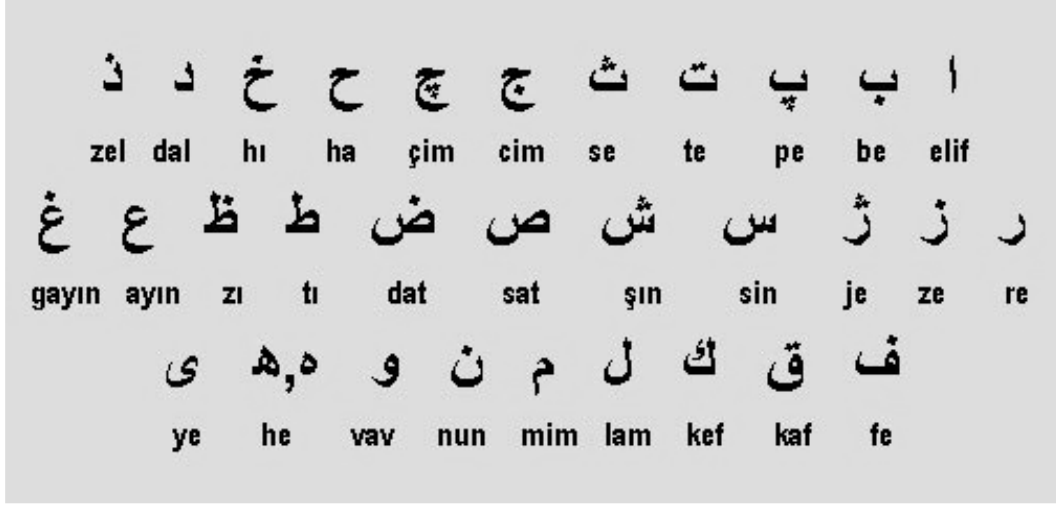
İlk bilgisayar destekli, 23 harfi tanıyabilen KT sistemi GİSMO, 1950'lerde David Shepard tarafından Amerika'da geliştirilmiştir. Bu alanda ilk patent sahibi alman Tauscheck'tir. 1929 senesinde Tauscheck, karakter tanıma için kalıp kıyaslama yöntemini ve mekanik teknolojileri kullanmıştır. Işık, maske üzerinden geçerek fotodetektörle yakalanırdı. Kalıpla tanınacak karakter tam olarak üst üste geldiği takdirde ışık maskeden geçmez ve böylece fotodetektörle algılanamazdı. Bu prensip 75 yıldan bu yana, günümüzde bile kullanılmaktadır [2].

## 2.2 Osmanlıca'nın Yapısı

13-20. yüzyıllar arasında Anadolu'da ve Osmanlı Devleti'nin hüküm sürdüğü yerlerde yaygın olarak kullanılmış olan, özellikle 15. yüzyıldan sonra Arapça ve Farsça'nın etkisinde kalan Türk yazı dili, 'Osmanlıca' olarak adlandırılır. Osmanlı Türkçesi ya da eski yazı olarak da bilinen Osmanlıca Arapça, Farsça ve Türkçe'nin karışımıdır ve Arap alfabesiyle yazılır [10].

Osmanlıca'da yazılar bitişik şekilde, sağ taraftan sol tarafa doğru yazılır. Harflerin birbirleriyle birleşmesi, ana çizgilerin üzerinden sağlanır.

Osmanlıca'da el yazması ve baskılar, Farsça temelinde bitişik ve ayrık şekillerde yazılmaktadır. Arapça alfabe 28 harf ile meydana gelmektedir. Osmanlıca'nın karakteristiği olarak, Arap harflerinin yanında Farsça'da mevcut olan, pe (پ), çe (چ) ve je (ج) harflerinin de kullanımı vardır. Bu 31 harften başka Türkçe'de bulunan ince g ünsüzü yerine geçebilecek kef harfi üzerine bir çizginin eklenmesi suretiyle oluşan "gef", genizsi n ünsüzünün yerine geçmesi amacıyla da üç noktanın eklenmesiyle "nef" (sağır kef, kâf-ı nunî), lam ile elifin birleşiminden lamelif, hemze ile h'nin ünlü şekli olarak hâ-i resmiye harflerinin de oluşturulduğu söylenebilir.



Şekil 2.3. Osmanlıca harfleri (26)

Sonuç olarak Osmanlıca alfabede 35 harfin varlığı mevcuttur. Bunların 25 tanesi baş tarafta, 26 tanesi orta tarafta ve 34 tanesinin de son tarafta olabilmesi mümkündür. Bir harfin yazımı müstakilleşmiş şekilde olup, başka harfler ile birleşim içerisine girmez. Harflerin dışında 10 tane de noktalama işaretiyle rakamlar ve boşluklar mevcuttur.

Öte yandan bir takım harflerin (örneğin ز, ر) ana çizgi altında yer alırken, bazılarının (örneğin ط, ظ, لا) iki parçayla oluştuğundan söz etmek gereklidir. Sesleri gösterirken hareketlerden faydalanılır. Harekelerin varlığı yahut yokluğu, kelimeleri farklı şekillerde anlamlandırır.

Osmanlıca'daana yahut küçük harf ayrımları gibi ayrımlar mevcut değildir. Noktalamalarda da kesinleşmiş bir kaide yahut kaideler dizisi yoktur. Osmanlıca'da harflerin sözcük başı, ortası yahut sonunda olabilmeleri de mümkündür.

Tek başına	Sözcük sonunda	Sözcük ortasında	Sözcük başında	Adı	Günümüz Türkçesiyle
ا	ا	—	—	elif	a, e
ء	—	—	—	hemze	' , a, e, i, u, ü
ب	ب	ب	ب	be	b
پ	پ	پ	پ	pe	p
ت	ت	ت	ت	te	t
س	س	س	س	se	s
ج	ج	ج	ج	cim	c
چ	چ	چ	چ	çim	ç
ه	ه	ه	ه	ha	h
ح	ح	ح	ح	hı	h
د	د	—	—	dal	d
ذ	ذ	—	—	zel	z
ر	ر	—	—	re	r
ز	ز	—	—	ze	z
ج	ج	—	—	je	j
س	س	س	س	sin	s
ش	ش	ش	ش	şin	ş
ص	ص	ص	ص	sat, sad	s
ض	ض	ض	ض	dat, dad	d, z
ط	ط	ط	ط	tı	t

**Şekil 2.4.** Osmanlıca'da harflerin baş, son ve ortada gösterimi (27)

Bir konum altında özdeş büyüklük sahibi durumunda bulunan az sayıda harfin varlığından söz edilebilir. Bir takım harflerin, kelime fonetiğiyle alakalı şekilde, kendilerine dair konumlardan farklılaşmış şekiller altında yazıldıkları görülebilir.(mesela ا, ع, أ, و, ئ, ی)

Başka başka Osmanlıca harflerin, tamamıyla özdeş şekillere sahip oldukları ve biriyle diğeri ayıran şeyin, tamamlayan karakterler (harflerle bütün şekilde yazılan noktalara dair konumlar ve sayılar) vasıtasıyla ayrılıyor olabilirler.

**Noktalar:** Osmanlıcada harfler, çoğunlukla noktalar alırlar. Bunlar, tek nokta, iki nokta ve üç nokta şeklinde olabilirler. Noktaların sayısı ve harflere göre yerleriyle, harfler farklılaşabilir.

**İşaretler:** Harflerin üst veya alt kısmına ve bazen isteğe bağlı olarak yerleştirilirler.

Karakterlerdeki genişlik, bir karakterden diğeri bir karaktere ve bir hat üzerinden başka bir hat üzerine değişkenlik göstermektedir. Pek çok karakter ( 35 karakterin 21'inin), karakter gövdesiyle ve tamamlayıcılık ihtiva eden noktalarla beraber, iki

parça şeklinde oluşturulmuştur. Noktaların, karakterin gövdesinde yahut aşağısında bulunması hali görülmektedir. Noktaların, tekli, ikili yahut üçlü gruplarla ifade edilmesi mümkün olabilmektedir. Noktayla yazılan harfler, noktasız şekillerde de yazılabilmektedir. Başka seslerin, üstün, esre, ötre biçimlerindeki çizgiler vasıtasıyla, istisnai de olsa gösterilebildiği haller vardır. Büyük yahut küçük harflerin varlığı söz konusu değildir.

Osmanlıcada yazılar, bitişik şekildedir ve kelimelerin boşluklar vasıtasıyla birbirleri üzerinden ayrılırlar. Bir takım harfler (dal, zel, re, je, vav و, ز, ژ, ر, د, ذ, ) kendisinden sonra gelen harf ile birleşmemektedir. Dolayısıyla, söz konusu harflerin birisi bir kelimeye bulunuyorsa, kelimenin iki alt kelime şeklinde bölünmesi hali ortaya çıkar. Söz konusu karakterlerin, alt kelimeye yalnızca son tarafta görüldüğü ve onun ardından gelmiş olan harfin, bir diğer alt kelimedeki baş harfi biçimlendirdiği görülür.

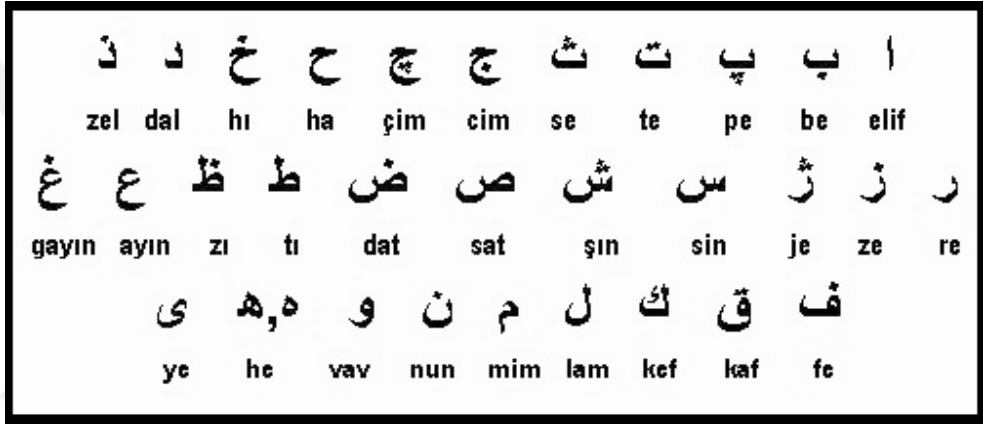
Osmanlıcada yazının, pek çok font ve yazı tipiyle gösterilebilmesi mümkündür. Özdeş fontta(hat) harfler, farklılaşmış büyüklüklerde görülebilirler. Bu sebeple, sabitlemiş genişlik uygulaması mümkün değildir.

Arapça'da 28 harf, Osmanlıca'da ise 32 harf vardır. Osmanlıca'da Arap harflerinin yanı sıra Farsçadaki 'p' (پ), 'ç' (چ), ve 'j' (ج) harfleri de mevcuttur. Bu 31 harfin dışında Türkçe'deki ince 'g' ünsüzünü belirtmek için kef harfine bir çizgi ekleyerek gef (گ) harfi de kullanılmıştır. Osmanlıca da Arapça gibi sağdan sola doğru yazılır.

Matbaanın Osmanlı'ya gelmesiyle ve basın-yayın hayatında kullanılmaya başlanmasıyla birlikte 19. yüzyılda basılı yayın oranlarında büyük bir artış görülmüştür. Özellikle II. Meşrutiyet döneminin getirdiği özgürlükçü hava sayesinde pek çok gazete ve dergi basımı gerçekleştirilmiştir. Bu konuda resmi yayınların dışında özel teşebbüsün de çeşitli basın yayın organları kurduğu ve hatta pek çok yeni yayına imza attığı görülür. Devletin resmi yayını olan Takvim-i Vekayi dışında Sabah, İkdam, Alemdar ve Tanin gazetelerinin basıldığı bilinmektedir. Bugün İstanbul'daki çeşitli kütüphanelerde bu gazetelere dijital ortamda ulaşılabilmektedir.

İstanbul Büyükşehir Belediyesi Atatürk Kitaplığı ve İstanbul Beyazıt Kütüphanesi bu konuda öncü rol oynayan iki kuruluş olarak bilinmektedir. Bu dönemde üretilmiş tüm gazete ve dergiler de sosyal bilimciler için önemli bir başvuru kaynağı olagelmıştır. Araştırmacıların bu kaynaklara erişiminin dışında Osmanlıca'ya da

hâkim olmaları gerekmektedir. Ancak işin zor tarafı Osmanlıca bilmek kadar o kadar fazla dokümanı tarayabilme zorluğudur. Her araştırmacının Amerika'yı yeniden keşfe çıkararak zamanının pek çoğunu bu dokümanları okumaya çalışarak geçirmesi, araştırmacının farklı kaynaklara inmesini ve daha da önemlisi tezleri üzerinde verimli çalışabilmesini zorlaştırmaktadır. O yüzden bu araştırmacıların dijital kaynakların Latin harfli karakterlere dönüştürülmüş ve hatta günümüz Türkçesiyle ulaşılabilir olmuş hallerine ihtiyacı vardır. Zaman tasarrufu, daha fazla araştırmacıların bu alanda çalışma yapabilmesi, dil bilme zorluğu gibi konularda araştırmacılara büyük ufuklar kazandırılmış olacaktır. Buradan hareketle matbaada basılmış bu gazetelerin taranması önem arz etmektedir.



Şekil 2.5. Osmanlıca Alfabe

Osmanlıca'yı Latin alfabesinden ve dolayısıyla bilgisayar tarafından da tanınmasını zorlaştıran bazı özellikler vardır. Osmanlıcada harfler çoğunlukla bitişik yazılır. Bir harf grubu bazen bir kelimeyi gösterir, bazen de bir kaç parçalı harf grubudur. Dolayısıyla kelimeleri birbirinden ayırmak zordur. Harflerin başta, ortada ve sonda yazılışları farklıdır (Bkz.Şekil 1.2). Eserler arasında harf kalınlıkları farklı olabilmektedir. Latin harflerinin matbaa çıktıları üzerinde elde edilen sonuçlar günümüzde yeterli düzeye ulaşmıştır. Fakat Çince, Japonca ve Osmanlıca karakterler üzerinde sorun hala devam etmektedir.

Özellikle Osmanlıca ve Arapça karakterlerin tanıtılması, dilin yapısı ve yazım şekli göz önünde tutulduğunda oldukça zorlaşmaktadır. Hatta matbaada hazırlanmış bir Osmanlıca metnin tanınması, el yazısı ile yazılmış bir Latince metinden daha zor olabilmektedir. [19].

İsmi den	Harfler	Sonda (sağdan bitişik)	Ortada (harf iki taraftan bitişik)	Başta (soldan bitişik)	Karşılıkları
elif	ا	ا	-	-	a, e
hemze	ء	ء	ء	ء	ʾ(a, e, i, u, ü)
be	ب	ب	ب	ب	b
pe	پ	پ	پ	پ	p
te	ت	ت	ت	ت	t
se	س	س	س	س	s
cim	چ	چ	چ	چ	c
çim	چ	چ	چ	چ	ç
ha	ح	ح	ح	ح	h
hu	ح	ح	ح	ح	h
dal	د	د	-	-	d
zel	ذ	ذ	-	-	z
re	ر	ر	-	-	r
ze	ز	ز	-	-	z
je	ج	ج	-	-	j
sin	س	س	س	س	s
şın	ش	ش	ش	ش	ş
sat	س	س	س	س	s
dat	د	د	د	د	d, z
tı	ط	ط	ط	ط	t
zı	ظ	ظ	ظ	ظ	z
ayın	ع	ع	ع	ع	ayın
gayın	ع	ع	ع	ع	g, ğ (kalm)
fe	ف	ف	ف	ف	f
kaf	ك	ك	ك	ك	k
kef	ك	ك	ك	ك	k, g, ğ(y), n
gef	ك	ك	ك	ك	g, ğ
nef, sağır kef	ك	ك	ك	ك	n
lam	ل	ل	ل	ل	l
mim	م	م	م	م	m
nun	ن	ن	ن	ن	n
vav	و	و	-	-	v, o, ö, u, ü
he	ه	ه	ه	ه	h, e, a
lame lif	ل	ل	-	-	la
ye	ي	ي	ي	ي	y, i, i

Şekil 2.6. Osmanlıca Harflerin Başta, Ortada ve Sonda Yazılışları [19]

### 2.3 Osmanlı Arşivleri

Osmanlı Devleti'nin 700 yılı aşkın tarihi boyunca ortaya çıkarılan ve arşivlerde yer alan matbu dokümanlar, birçok tarihsel araştırmada kullanılmakta ve bilimsel çalışmalar için zengin bir altyapı oluşturmaktadır. Avrupa, Kuzey Afrika ve Ortadoğu'da 30'a yakın günümüz devletinin tarihi ile ilgili önemli verileri barındıran bu eserler, evrensel bir kültür mirasını oluşturmaktadır. Harf devrimi sonrasında Osmanlıca'nın günlük hayatta kullanılmamasıyla birlikte bu dile hâkim olan insanların sayısı azalmış, çeviri konusunda yetkin uzmanlar bulmak giderek zorlaşmış ve çeviri kaynaklarının düzensizliği nedeniyle sistematik çalışmaların yapılması imkânsız hale gelmiştir. Matbu eserlerde kâğıt ve harf boyutu, el yazması

eserlerde ise kâğıt ve harf boyutunun yanı sıra eseri yazanların yazı karakteri, ayrıca kâğıdın ve mürekkebin zamana bağlı olarak yıpranmış olması, çeviri işlemlerini zorlaştıran bir başka faktördür.

Dergi, gazete ve kitap gibi kaynaklar da eklendiğinde Osmanlıca matbu eserlerin iki milyon civarında olduğu tahmin edilmektedir. Ancak Osmanlıcanın 33 harften oluşan özgün bir dil olması ve alfabedeki karakterlerin karmaşıklığı çoğunlukla bu eserlerin günümüz Türkçesine, yani Latin harflerin kullanıldığı bir alfabetik sisteme çevirisinin çok zor şartlarda ve uzun sürelerde yapılmasına neden olmaktadır. Buna ek olarak, 33 harfin kelimenin başına, sonunda ya da üst üste kullanımı ile birlikte 200'e yakın farklı alternatif oluşmakta ve bu durum çevirinin süresini ve doğruluğunu olumsuz yönde etkilemektedir. Bu zorluk nedeniyle Osmanlıca – Türkçe ve Türkçe çeviriler aracılığıyla da İngilizce çeviriler yalnızca araştırmacıların ilgilendikleri ve özel olarak araştırmak istedikleri metinler düzeyinde yapılmakta, dolayısıyla istenen objektiflik düzeyinin altında kalmaktadır.

Osmanlı İmparatorluğunun ilk yıllarından yıkılışına kadar geçen süredeki tüm belgeler Osmanlı Arşivi olarak adlandırılmaktadır. Bu arşivler Türkiye Cumhuriyeti Başbakanlık Devlet Arşivleri Genel Müdürlüğü'ne bağlıdır. Arşiv binası İstanbul'un Kağıthane ilçesinde bulunmaktadır.

95 milyon farklı belge bulunan Başbakanlık Osmanlı Arşivleri'ndeki belgeler halen taranarak elektronik ortama aktarılmaktadır. Arşivlerdeki belgeler 400 uzman tarafından elle tasnif edilmeye çalışılmaktadır [17]. Milyonlarca belgenin elle tasnifinin ne kadar zor olduğu ve ne kadar uzun süre gerektirdiği aşikârdır. Osmanlı arşivlerinin ülkemiz ve dünya için önemi düşünüldüğünde, günümüzün teknolojik gelişmelerinden yararlanılarak, belgelerin en kısa sürede araştırmacıların hizmetine sunulması gerektiği anlaşılmaktadır.

Osmanlı arşivlerindeki eserlerin bir kısmı matbu olup, büyük çoğunluğu el yazmasıdır. El yazısı karakterlerin kişiden kişiye farklılıklar göstermesi, harflerin birleşmesi, matbaa çıktılarına göre tam bir standardının olmaması ve özellikle Osmanlıcadaki çok çeşitli ve süslü yazı tipleri nedeniyle, el yazması eserlerin bilgisayar tarafından algılanması çok zordur. Bu alanda yapılan başarılı çalışmalar [2,3] olmasına rağmen henüz optimal bir sistem geliştirilememiştir.

## 2.4 Osmanlıcanın Latin Harflerinden Farkı

Diğer bölümlerde Osmanlıca'nın temel özelliklerinden bahsedilmiş, bu bölümde ise Osmanlıca ile Latin harfleri arasındaki farklar üzerinde durulmuştur. Evvela Latin alfabesinin Arap alfabesine tercih edilmesinin ve Arapça'dan vazgeçilmesinin nedenini Arap alfabesindeki yapısal özellikler ile alakalı olduğu bilinmelidir. Arapçanın yapısı, ünsüz harflerden oluşan bir dizi şeklindedir. Bu sebeple de Türklerin okuması için uygun değildir. Arapçada yazılan harekesi olmayan Türkçe içerikli sözcükleri okumada ortaya çıkan güçlüklerin azaltılması amacıyla elif ve vav harflerinin ündeşçe kullanımının yanında, ünlü harf göreviyle de kullanımı başladı. Hakikaten de Türkçenin sekiz adet ünlü harfi bünyesinde bulundurmasının yanında Arapça, muhteviyatında ünlü harf bulundurmamaktadır. Arapçada yeri geldiğinde “a, e, i” seslerini tek bir ünlü harfle, yani “Elif” harfiyle ifade etme durumu vardır.

Bu sebeple, örnek olarak “gel, gül, kel ve kil” sözcüklerinin yazımı, Arapçada bire bir benzer şekildedir. Bir metni yahut metin içerisindeki cümleyi okuyan kişi, kastedilen manayı düşünerek kendisi bulmak zorundadır. Haliyle, bir sözcüğü yanlış anlamlandırmamanın yolu da cümledeki anlamın geliş şekliyle alakalıdır. Cümledeki anlamın tamamıyla kavranamayışı, söz konusu olan sözcüğün cümlede olan mevcut vazifesinin idrak edilememesi, yanlış okumalara sebebiyet vermektedir. Yani Osmanlıcada bir sözcüğün, yapısı, çekimi, cümle görevi bakımıyla farklı farklı okunup anlamlandırılabilmesi mümkündür.

Maddelerin başında bulunan her bir sözcüğün, ne şekilde okunması gerektiğinin belirtilmesi zaruridir. Bu belirtmede kullanılan iki farklı geleneğin varlığından söz edilebilir. Bunların ilki; dil içerisinde bariz biçimde bulunan, örnek bir sözcüğü ölçü olarak okunacağı söylenmesi, bir diğer gelenek ise harflerin birinin yahut ikisinin harekesini belirtme yöntemidir. Arapça sözlükleri yazanlar, bu hususta şu ana dek tam manasıyla bir çözüme ulaşamamışlardır.

Sonunda “i, ı, u ya da ü” harfleri olan sözcüklerin “y” harfiyle bitirilmesi durumu vardır. Örnek olarak; “sütlü” sözcüğünü “sütli”, “okullu” sözcüğünü “okulli” şeklinde yazma hali görülür. Öte yandan Arapçada iki farklı “t”, üç farklı “s”, üç farklı “h”, iki farklı “n”, dört farklı “z” harfinin varlığından bahsedilebilir; oysa Türkçe, bu Latin alfabesiyle bu harfler için birer sesi bulundurmasıyla, örnek olarak ; “saat”, “sel”, “siz” ve “su” yazmak için başka başka “s” harflerine ihtiyaç



duymamaktadır. Sonuç olarak Arapçada bu “s” lerin birisi “sat”, birisi “sin” ve birisi de “se” harfleriyle yazılırken, Latin alfabesiyle kurulan modern Türk alfabesinde, bir adet “s”, bütün bunlarla aynı işlevi görebilmektedir. Öte yandan Arapçada “g” harfi olmadığından, “kef” harfinin, hem “ke” hem de “g” harfi işlevleriyle kullanıldığı da örnek olarak verilebilir.

Latin alfabesinde tüm karakterler, yatay düzlem doğrultusunda, birbirlerini takıp eder şekildedirler, oysa Arapça karakterler bazen üst bazen de alt uzantıları sebebiyle birbirleriyle düşeyleşmiş bir etkileşime girebilmektedirler.

Osmanlıcadökümanlarda genelde satırlar birbirleriyle bağımsız durumdadırlar. Fakat bazen bir satırın alt satırla yahut üst satırla olan alt sınırı aşabildiği görülür. Bu nedenle de satırları birbirinden ayıran net bir hattın söz edilemez.

Yukarıda sayılan nitelikler nedeniyle, dökümanları işlemede ana damar durumundaki “Alan Parçalama” ve “Alan Etiketleme” zor bir hale gelmekte ve bu durum da sistem çalışmasının hata verme oranında yükselmelere sebep olmaktadır.

Osmanlıca'nın temel özellikleri aşağıda özetlenmiştir:

- 1) Osmanlıca yazı (baskı veya el yazısı) bitişiktir ve sağdan sola doğru yazılır. Harfler, normalde birbirine temel çizgi üzerinde birleşir.
- 2) Osmanlıca baskı ve el yazmaları, farsî temelde bitişik ve ayrık yazılır. Arap alfabesi 28 harften oluşur. Osmanlıca'da Arap harflerinin yanı sıra Farsça'daki pe (پ), çe ( چ ) ve je ( ج ) harflerini de kullanmışlardır. Bu 31 harfin dışında Türkçe'deki ince g ünsüzünü belirtmek için kef harfine bir çizgi eklenerek gef, genizsi n ünsüzü için üç nKTA eklenerek nef (sağır kef, kâf-ı nunî), lam ile eliften lamelif, hemze ile h harfinin ünlü şekli olan hâ-i resmîye harfleri oluşturulmuştur. Dolayısı ile Osmanlıca'da 35 harf vardır. Bunlardan 25 adeti başta, 26 ortada ve 34 adedi sonda olabilir. Bir harf müstakil yazılır, diğer harflerle birleşmez. Harflerden başka 10 adet rakam ve nKTalama işaretleri, boşluklar ve özel semboller kullanılır.
- 3) Harflerden bazıları, temel çizginin altında yer alır (mesela ج, ز).
- 4) Harflerden bazıları, iki parçadan müteşekkildir (mesela ط, ظ, لا).
- 5) Seslerin gösteriminde harekeler kullanılır. Harekelerin varlığı veya yokluğu aynı kelimenin anlamını farklı kılar.
- 6) Osmanlıca harflerinde temel ve küçük harf ayrımı yoktur. Noktalama işaretlerinde

kesin kurallar bulunmamaktadır. Osmanlıca harfleri sözcüklerin başında, ortasında ve sonunda farklı biçimde yazılır.

7) Osmanlıca harflerinin Türkçe'deki zengin ünlü sistemini karşılamada yetersiz olduğu düşünülür. Örneğin Osmanlı alfabesindeki elif (ا) Türkçe'deki a ve e ünlüsünün karşılığıdır ya da Türkçe'deki u, ü, o, ö ünlülerinin yerine Osmanlıca'da yalnızca (و) harfi vardır, bu aynı zamanda v ünsüzünün de karşılığıdır.

8) Herhangi bir konumda aynı büyüklüğe sahip olan çok az harf vardır. Bazı harfler, kelimenin fonetiğine bağlı olarak kendi konumunda farklı şekillere sahip olabilir. (ا, ء, اُ, اِ, اَ mesela)

9) Farklı Osmanlıca harfler, tam olarak aynı şekle sahip olabilir ve birinden diğerine sadece tamamlayıcı karakter (harfle bütünleşen noktaların konumu ve sayısı) ilavesiyle ayrılırlar. Noktalar: Harflerin önemli bir kısmı bir nokta, iki nokta ve üç nokta alır. Bu noktaların sayısı ve harfin üstünde veya altında olmasına göre harfler farklılaşır. İşaretler: Harflerin üst veya alt kısmına ve bazen isteğe bağlı olarak yerleştirilirler.

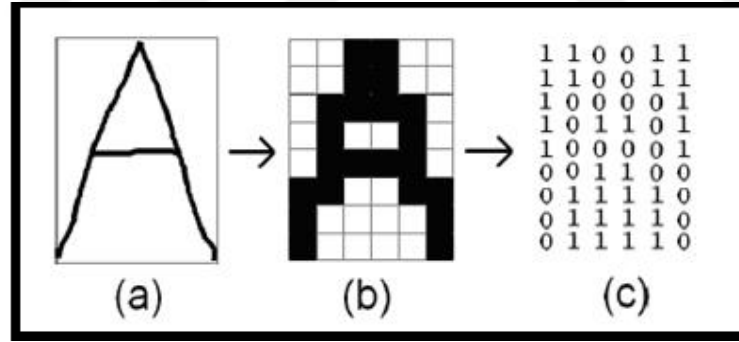
10) Karakterlerin genişliği, bir karakterden diğerine ve bir hattan diğerine değişir. Birçok karakter (35'in 21'i), karakterin gövdesi ve tamamlayıcı noktalar olmak üzere iki parçadan müteşekkildir. Noktalar, karakter gövdesinin üzerinde veya aşağısında bulunabilir. Noktalar, bir, iki veya üçlük gruplar halinde olabilir. Noktalı harflerin noktasız olanı da vardır. Diğer sesler, üstün, esre, ötre biçiminde çizgilerle nadiren gösterilebilir. Büyük harf veya küçük harf durumu mevcut değildir.

11) Osmanlıca yazı, bitişiktir ve kelimeler boşluklarla birbirinden ayrılmıştır. Bazı harfler (dal, zel, re, je, vav د, ذ, ر, ز, ج, و) bir sonraki harfle birleşmez. Bundan dolayı bu harflerden biri, kelime içinde mevcutsa, kelime iki alt kelimeye bölünür. Bu karakterler, alt kelimenin sadece sonunda görünür ve bundan sonra gelen harf, bir sonraki alt kelimenin baş harfini biçimlendirir.

12) Osmanlıca yazı birçok fonta ve yazı biçimine sahiptir. Aynı fonttaki (hat) harfler farklı büyüklüklere sahiptir. Bundan dolayı sabit genişliğe dayalı bölümlenme uygulanamaz.

### 3. GÖRÜNTÜ İŞLEME

Görüntü işleme, genel terim olarak resimsel bilgilerin manipulasyonu ve analizi demektir. Bir başka ifade ile görüntü işleme, verilerin yakalanıp ölçme ve değerlendirme işleminden sonra, başka bir aygıtta okunabilir bir biçime dönüştürülmesi ya da bir elektronik ortamdan başka bir elektronik ortama aktarmasına yönelik bir çalışma olan "Sinyal işlemeden" farklı bir işlemdir. Görüntü işleme, daha çok, kaydedilmiş olan, mevcut görüntüleri işlemek, yani mevcut resim ve grafikleri, değiştirmek, yabancılaştırmak ya da iyileştirmek için kullanılır. Bir resmin sayısallaştırılmasının açıklanması amacı ile öncelikle siyah–beyaz resim göz önünde bulundurulmuştur. Siyah–beyaz resim sadece iki gri değerden oluşan bir resimdir. Böyle bir görüntüde her bir piksel ya siyah ya da beyaz olarak oluşur. Bu şekilde 0 ve 1 kodlanmış piksellerden oluşan görüntülere ikili görüntü (binary image) adı verilir (Şekil 3.1).



Şekil 3.1. İkili Görüntü

Gri tonlu görüntülerde; görüntü farklı gri ton değerlerinden oluşur. Gri değer aralığı  $G=\{0,1,2,\dots,255\}$  şeklinde ifade edilir. Bunun anlamı; gri tonlu bir görüntüde 256 tane farklı gri ton değeri daha doğrusu gri değer bulunabilir. 0 gri değeri kural olarak siyah renge, 255 gri değeri ise beyaza karşılık gelir. Bu değerler arasında ise gri tonlar oluşur.

Gri düzeyli bir görüntüde gri değer aralığı o görüntünün radyometrik çözünürlüğü ile doğrudan ilgilidir. Yani bir başka deyişle, bir görüntüdeki gri değer aralığı o

görüntünün radyometrik çözünürlüğü ile ilgili bilgi içerir. Örneğin 256 gri değer aralığına sahip bir görüntüde 256 gri değere karşılık gelen değer  $2^8$  dir. Yani bu görüntüyü 8 bitlik görüntü şeklinde ifade ederiz. Aynı geometrik çözünürlüğe sahip iki görüntüden 8 bitlik görüntü ile 6 bitlik görüntüyü karşılaştıracak olursak 8 bitlik görüntüde 256 tane gri değer aralığı olmasına rağmen 6 bitlik görüntüde 64 tane gri değer aralığı söz konusu olacaktır. Dolayısıyla, objeden gelen enerji, radyometrik çözünürlüğü düşük olan görüntüde daha az bir gri değer aralığı şeklinde algılanacak ve objenin ayırt edilebilirliği ve yorumlanabilirliği azalacaktır. Bunun sonucunda, sayısal görüntü işlemenin bazı otomatik işlemlerinde gri değer düzeyinin azlığı elde edilecek sonuçların kalitesini ve doğruluğunu doğrudan etkileyecektir[5].

### 3.1 Ön İşleme

Ön işlemeye dair algoritmalar, ihtiyacı hissedilen verilerin, ileri aşamalarda kullanımlarını mümkün kılan bir takım işlemleri yapmaya yarar. Başka bir ifadeyle, gerçeklikle tanıma motorlarının arasında bir köprü vazifesi görür.

Tarama aşamasında, gerçek nesnelere bilgisayar formatlarına dönüştürülür ve bilgisayarın bu nesnelere anlamlandırması sağlanır. Tarayıcıların, off-line el yazısı karakterlerin tanınması işlemi içerisinde, en fazla tercih edilen araç konumunda oldukları söylenebilir.

Tarayıcıların, pek çok format biçiminde görüntüler üretmesi mümkündür. Bu format türlerinden en fazla kullanılanı BMP format türüdür. BMP format dosyası, iki temel parçayla oluşur. Bunlardan ilki, başlığın olduğu bölümdür. Başlık bölümü, genişliği, yüksekliği, piksel görüntüsündeki derinliği, her piksel üzerindeki bit sayısını ve piksel bilgisinde kullanılacak olan işaretçiler gibi, görüntüyle alakalı gereken bir takım enformasyonları bünyesinde barındırır.(19)

Asıl piksel bilgisi, ilk parçanın ardından başlamaktadır. Bütün piksellerin, kırmızıyla, yeşille yahut maviyle (RGB) gösterilmekte olan değerleri mevcuttur. Söz konusu değerler piksel renginin belirlenmesi için gereklidir. BMP dosyaları 24 bit olarak boyutlanır. Kırmızılar için 8, yeşiller için 8 ve maviler için de 8 bitlik kullanım vardır. Toplamda 24 bit RGB değerinin bir tek pikselin temsilinde kullanılması söz konusudur.

İkili seviyeye indirgeme, asıl görüntüyü gri seviyelere indirgemeye alakalıdır. İkili

seviyeye indirgenmiş olan görüntünün, enformasyon karmaşasını düşürmek amacıyla bir takım bilgileri budamasına karşın, nesnede bulunan şeklin, pozisyonun ve piksellerin sayısının bilgilerini içermesi durumu vardır.

Bir RGB görüntüsünden, ikili seviyeye indirgenmiş bir görüntünün sağlanması iki aşamayla mümkün bir süreçtir. İlk aşama, 8 bitten oluşan kırmızının, yeşilin ve mavinin değerlerinin kullanılmasıyla, bütün piksellerin ortalama gri değerlerini hesaplamak yoluyla yapılır.

$$\text{Gri}(i, j) = 0,299 * k(i, j) + 0,587 * y(i, j) + 0,114 * m(i, j) \quad (1)$$

Buradaki, i satır numarası, j ise sütun numarasını ifade eder; k, kırmızının değerini; y, yeşilin değerini, m ise mavinin değerini açıklar. Gri ise alakalı olan pikseldeki gri değerini gösterir.

Bir sonraki aşama ise, gri seviyelere indirgenen görüntünün, bir eşikleme işlemine tabi tutulmasıdır. En sık kullanılmakta olan metod, tek bir eşik değerinin belirlenmesi metodudur. Bu değer altındaki tüm piksellerin siyah (0), üstünde kalan tüm piksellerinse beyaz (1) şeklinde sınıflandırılması yapılır. İkileştirmede, eşik değeri doğru saptanması oldukça önemlidir. Söz konusu değer hesaplanması, görüntüdeki ortalama gri değerinin vasıtasıyla olabileceği gibi, kullanıcıların verdikleri değerler vasıtasıyla da yapılabilmektedir.

Eşik değerinin küçük seçilmesi, bilginin kaybına yol açar ve fayda içeren noktaların da silinmesi durumu meydana gelir. Değerin büyük seçilmesi durumunda ise gürültü fazlalığı durumu ortaya çıkar ve siyah noktaların fazlalığı sayesinde, karakterlerin tanınması işlemi zorlaşır. Eşik seviyesinin uygun seçimi durumunda, bilgilerde kayıp yaşanmamakla beraber, lüzum olmayan noktaların da ortaya çıkması engellenir.(20)

Filtreler görüntü düzenlemelerinde de sıkça kullanılan tekniktir. Filtreleme algoritması resimdeki gürültüleri yok etmek ve resimdeki renk değişikliklerini düzeltmek için kullanılır.

Filtrelemede, resim üstünde bir filtrenin olduğu varsayılarak, bütün piksellerin baştan değerlendirilmesi işlemidir. Filtrelerin, işlevsel olarak görüntüyü zenginleştirme, ayrıntıları ayıklamaya hüt belirginleştirme gibi operasyonları gerçekleştirme amacını taşıyan operatörleri ifade ettiği söylenebilir. Resimlerdeki ayrıntıların meydana

çıkartılması yahut resmin içindeki gereksiz gürültüleri önlemek, filtrelerin fonksiyonlarındandır. Bunun yanında, değişik amaçların, farklı filtrelemeleri gerekli kıldığı da söylenebilir. Bu filtrelemeler; kenarların keskinleştirilmesi, kenarların yakalanması, görüntünün yumuşatılması ve buna benzeyen pek çok amacı içerebilmektedir.

Keskinleştirme filtrelerinin işlevi ise, görüntülerdeki detayları belirginleştirme ve nesne sınırlarını belirginleştirmektir. Hususiyetle medikalde kullanılan görüntülemelerde, keskinleştirme filtresi sıkça kullanılır. Keskinleştirme filtresi, kenar belirleme metodu şeklinde de bilinmektedir.

Kenar bulmada 3 temel aşama vardır:

- 1- Gürültü Azaltma: Anlam ifade eden kenarların etkilenmemesini sağlayarak, mümkün mertebe gürültü bastırma işinin yapıldığı aşamadır.
- 2- Kenar Pekiştirme: Kenarlar üzerinde, tepkimesi daha yüksekte olan bir filtre kullanma aşamasıdır.
- 3- Kenar Tespiti: Filtrelerin çıktılarının hangisinin anlam ifade ettiğini, hangisinin gürültülü olduğunun tespit edilmesi aşamasıdır.

Yüksek geçirgenlik gösteren filtreler, türevsel filtreler şeklinde isimlendirilir. En yaygınları, Prewit ve Sobel filtreleri şeklinde karşımıza çıkmaktadır. Kullanılmakta olan bir diğer filtre ise Gaussian filtresidir. Gauss filtreler, görüntü yumuşatması yapmak, detaylar ve kirliliklerden uzaklaşmak açısından uygulanmaktadır.

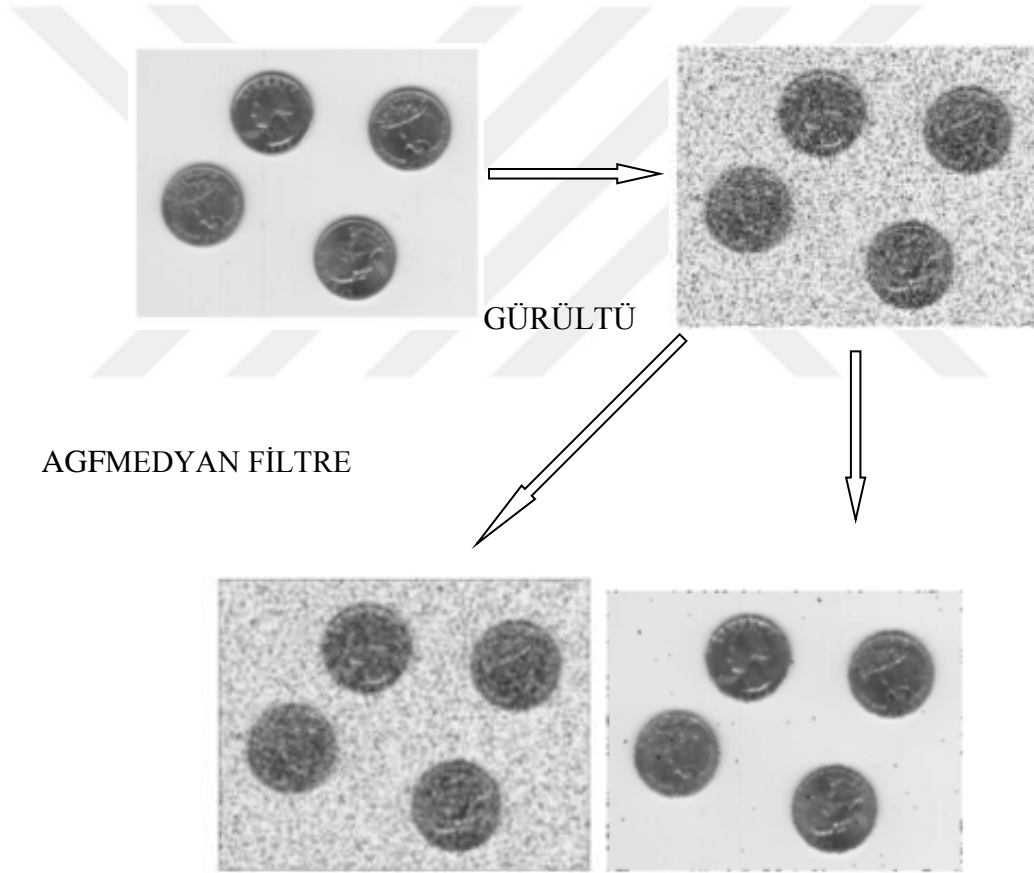
Ortalama değer filtrelerindeyse, orijinal görüntünün gösterildiği bir piksel, 8 komşu pikselle beraber, büyükten-küçüğe sıralanarak, ortada saptanan değer o piksele atanması sağlanır. Görüntü kirliliğini yok etmede alternatif bir metot şeklinde kullanılmaktadır.

Filtrelerin içinde kullanılabilir başka bir filtre yöntemi ise, meydan filtreleme yöntemi olarak karşımıza çıkmaktadır. Söz konusu yöntem sayesinde,  $3 \times 3$ ,  $5 \times 5$  gibi piksellerin sıralanması ve merkezdeki piksel olarak kabul edilmesi söz konusudur. Piksellerin, yine küçük olandan büyük olana doğru sıralanması hali vardır ve bu sıralamanın yapılması amacıyla, bilgisayar sıralama algoritmaları kullanılır. Denklem 2'de  $3 \times 3$  için seçilmiş maskeye ait merkez görülmektedir.

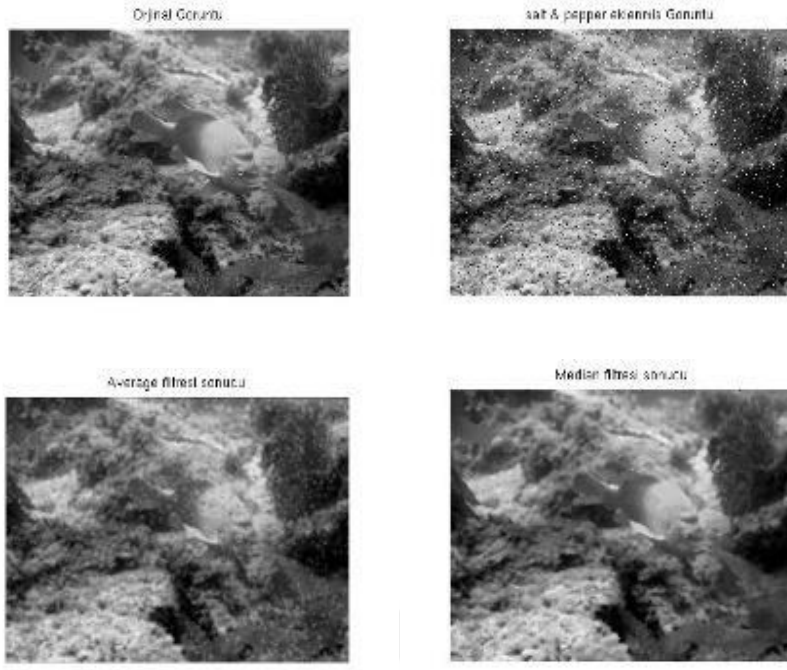
$$\left\{ \begin{array}{l} 646576 \\ 566861 \\ 345568 \end{array} \right\}$$

Denklem (2)

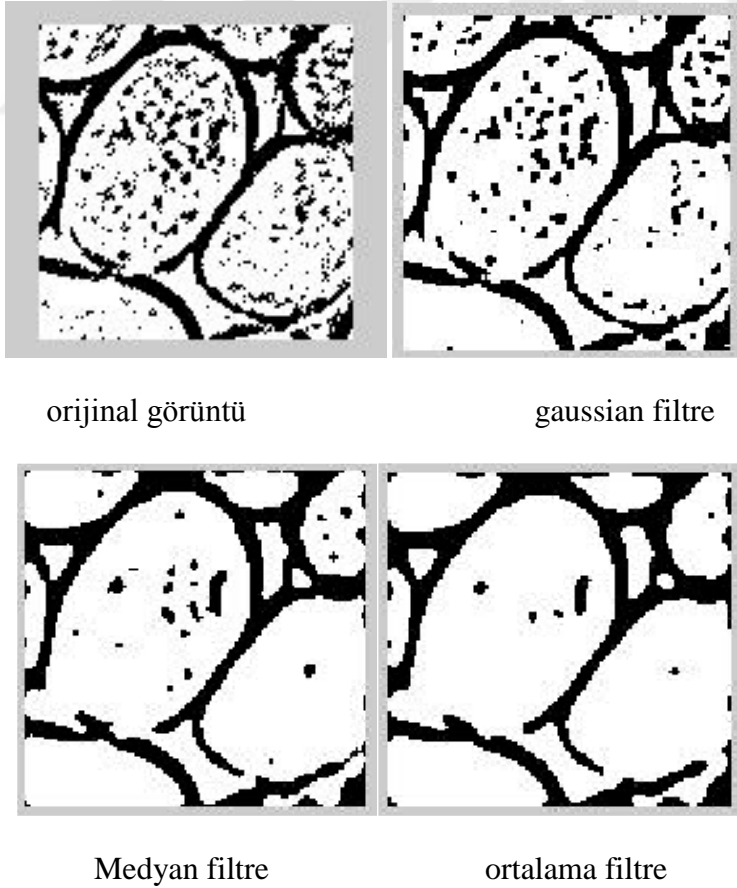
Küçük olandan büyük olan doğrultusunda yapılmış olan sıralamanın sonucunda, 34 – 55 – 56 – 61 – 64 – 65 – 68 – 68 – 76 bulunduğu görülmüştür. Medyan filtresinin sonucunda ortaya çıkan merkez piksel değerinin 68'den 64'e doğru değiştiği görülmektedir. Böylelikle, merkez piksel meydan filtresinin doğrultusu ekseninde yeniden işlenip, komşusu olan pikseller yönüne yaklaştırılır. Medyan filtresi uygulaması, Şekil 3.2'de ve Şekil 3.3'de görülmektedir. Medyan, gaussian ve ortalama filtre örnekleri de Şekil 3.4'de görülmektedir.



Şekil 3.2. Medyan filtre uygulama örneği (28)



Şekil 3.3.Orjinal görüntü (üst sol) salt &pepper gürültüsü (üst sağ), ortalama alma teoremiyle (alt solda) ve medyan filtresi ile (alt sağda) (29)



Şekil 3.4. Orijinal görüntü ve filtre uygulamalı görüntüler (30)



Görüntülerin, teknikler vasıtasıyla sayısallaştırılmasının akabinde, YSA'da eğitilip hedeflenen verilere ulaşılmıştır. Bu sayede de YSA'larla görüntü işleme tekniklerinin koordinasyonu sağlanabilmiştir.

Görüntü işlemede, veriler üzerindeki istem dışı değişimler gürültü olarak adlandırılır. Söz konusu işlemler segmentasyonun hızının artmasını sağladığı ve başarının oranını da yükselttiği söylenebilir.

Elde edilmiş olan ikili düzen görüntüsü üzerine yansımış olan gürültüleri gidermek, ardından gelen işlemlerin gerçekleşmesine pozitif katkılar sağlaması hususunda oldukça fazla önem arz etmektedir. Ekseriyetle gürültülerin bir bölümünü, kâğıt yüzeyde oluşan çok detaylanmış görüntülerde varlığı görülen küçük lekeler, bir başka bölümü ise kalem hatalarının sebebiyet verdiği çizikler oluşturmaktadır. İlk gruptaki görüntülerin daha fazla, piksellerin serpilmesi şekliyle ortaya çıkan tuz biber gürültüsü şeklinde ortaya çıktığı, bu gürültünün biçiminin artış gösteren algılama çözünürlüğüyle beraber, piksel gruplarının şekliyle de görüntülere yansiyabilmesi mümkündür. Kalem hatalarının sebebiyet verdiği ve karakterlere dair olmayan noktaların ve çiziklerinse, önceki kısımdakine oranlı olarak, genelde daha geniş çapta piksel gruplarıyla meydana geldiği söylenebilir. Gürültülerden oluşan pikselleri gidermek amacı ile kullanılmakta olan bir diğer yöntem de, 5x5 boyutlarıyla ortaya çıkan bir filtre elemanını kullanmak olarak karşımıza çıkar. Maskelemenin gerçekleştirilmesiyle meydana gelen filtrelemede 5x5 maskeleme elemanının, karakterdeki tüm piksel noktalarına uygulanması söz konusudur.(21) Maske eleman değerlerinin 1 olarak seçilmesi ve komşuluk sayısının eşiğe dair değerlerin altında olanlarının silinmesi sonucunda, gürültü oluşturmuş olan piksellerin de silinmesi sağlanır.

Filtrelemeyle beraber, gereksiz piksellerin yahut piksel gruplarının temizlenmesi öte yandan da, karakterlerin içinde yer bulan küçük piksel boşluklarının da doldurulması sağlanır. Söz konusu etkinin bunun yanında, karakterlere dair lakin karakterle bağlantısının kopuk olduğu görülmüş olan yakın piksellerin de, karakterin gövdesi üzerine bağlanması vasıtasıyla onarılması sağlanır. Öte yandan işlemlerde meydana gelen doğal sonuca göre de, karakterin şekil keskinliği, geçişli kenar noktaların da doldurulup/silinmesi işlemlerinin sonucunda, görelî şekilde daha yumuşaklaşmış geçişler içeren bir şekle indirgenir.(22)

Gürültülerin temizlenmesi amacı ile kullanılmış olan bir diğer yöntemdeyse, 3x3'lük

yahut 4x4'lük şablonların kullanımı sağlanır. Söz konusu şablonların kullanılmasıyla, gerekli olmayan noktaların da silinmesi sağlanır. Şablonlarda "0" beyaz pikselleri, "X" ise siyah pikselleri göstermektedir. Bu yöntem uyarınca, resimlerde bulunan pikseller üzerine 3x3'lük şablonların gezdirilmesi ve bu şartları sağlamakta olan siyah piksellerin silinmesi sağlanır. Öte yandan da bütün pikseller üstünde 4x4'lük şablonların gezdirilmesi ve bu şablonlar ile eşleşmiş olan siyah piksellerin silinmesi de sağlanır.

Belgenin içeriğine ulaşabilmek için öncelikle tek bir resim halinde mevcut olan belgenin satırlarının tespit edilmesi gerekir. Doküman görüntüsünün dikey başlangıç noktasından başlanarak her bir piksel satırı siyah pikselle karşılaşınca kadar yatay olarak taranır Görüntü matrisinde ilk kez 1 rakamıyla karşılaşılan satıra kadar tarama yapılır. İlk siyah piksele rastlanılan piksel satırı, metnin başlangıç satırı olarak kabul edilir. Böylece metnin başlangıç satırının üst sınırı tespit edilmiş olur. Tarama işlemine devam edilerek siyah piksel içermeyen ilk satır bulunduğu anda metnin başlangıç satırının bitimine ulaşılmış olur. Matrisin, üst sınırı veren satır numarası ile alt sınırı veren satır numarası arasındaki fark ilgili satırın yüksekliğidir (Bkz. Şekil 3.5). Satır belirlemenin en önemli problemi; bazı karakterlerin alt ve üst noktalarının bulunduğu piksel satırlarının ayrı birer metin satırı gibi algılanmasıdır. Bunu önlemek için tespit edilen her satırın yüksekliği ile alt ve üst satırlara olan uzaklıklarına bakılır.

Karakter çevrelemek amacıyla, öncelikle verilmiş olan metinleri satırlara ayırmak gerekir. Dokümanın sadece tek bir sütun metni içeriyor olduğu ve eğitimsizce ikileştirilmiş biçimlere dönüştürülmüş olduğu kabul edilir. Bir başka varsayımın ise, karakterleri birbirlerinden ayrı düşünme varsayımıdır. Satırları belirlerken, dokümanın görüntüsünü dikeyleşmiş başlangıcın olduğu noktadan başlaması şartıyla, sol yönden sağ yöne, bütün yataylaşmış piksellerin satır taraması yapılır. İlk siyah pikselin olduğu piksel satırının, metin satırlarında bir başlangıç(satır başı) olduğu varsayılır. Bu aşamanın ardından gelmekte olan ilk pikselsiz piksel satırının, metin satırınca kapsanan yataylaşmış piksel satırının, yani metindeki satırın yüksekliğinin belirlenmesinde kullanılması söz konusudur. Ayrılmış olan metin satırları içerisinde, Türkçe karakterlerde görülen üstü noktalı karakterlerin(İ, Ö, Ü, Ğ) üst kısımlarını imha etme ihtimalleri mevcuttur. Bu nedenle ikincil bir metin satırının, yine anlatılmış olan yöntem sayesinde tespiti sağlanır. Bu iki satır arasındaki yüksekliğe

dair oranlar ve birbirlerine olan uzaklıklarıyla, ikinci satır yönündeki uzaklığı karşılaştırılarak, ilk saptanan satırın üstü noktalı durumdaki karakterlerin satırı olup olmadığının araştırması yapılır.

Satırları tek satıra dönüştürebilmek için, sağlanması gereken şartları şu şekilde sıralamak mümkündür:

- İki satırın arasında bulunan uzaklık, ikinci satırla mesafesinin oranının  $1/5$ 'e eşit yahut küçük olmalı.
- İlk satırdaki piksel sayısının, ikinci satırdaki piksel sayısı ile oranının  $1/5$  e eşit yahut küçük olması gereklidir.

Söz konusu şartların sağlanamaması durumunda, ilk satırın, karakter ayırma modülüne gönderilmesi ve ikinci satırın da ardından gelen satır ile birleştirilme ihtimali karşısında elde tutulması sağlanır. Şartların sağlanması halinde ise, satırların birleştirilmesi ve karakter ayırma modülüne gönderilmesi işlemi gerçekleştirilir.

Karakterler çerçevelenmedeyse, satır başlangıcı ve son piksellerin arasındaki yataylaşmış başlangıç noktası üzerinden ( $x=0$ ) başlanıp, dikey doğrultuya doğru taramalar yapılmaktadır. Satırları belirlemede olduğu gibi, rastlanacak ilk siyah pikseli içermekte olan sütunun, karakterdeki sol başlangıç noktası kabul edilmesi hali vardır. Bu sütunun ardından rastlanacak ilk siyah piksel içermemekte olan sütunun ise karakter sonu olarak varsayılması şeklinde süreç devam eder. Böylelikle, karakterin solunda ve sağında bulunan kenarlarda, hiçbir boşluğun olmadığı bir çerçeve çizilmesi sağlanarak, karakterin eninin belirlenmesi gerçekleşir.

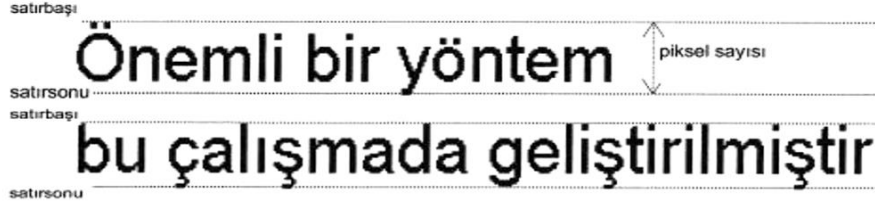
Takip eden aşamada, çerçeveye aşağı ve yukarı yönlü daraltılma işlemi uygulanır. Bunun nedeni, karakterin tümünden bir satır yüksekliğine erişmemiş olabilmesidir. Satırın hem üzerinde noktalar olan (İ,Ö gibi), hem de kuyruğu olan (ç,g,j gibi) karakterleri içermesi ihtimalinden dolayı, ilk olarak üst taraftan, ilk siyah piksel içermekte olan pikselin satırına varıncaya dek çerçevenin daraltılması sağlanır. Benzer biçimde, aşağı yönden yukarı yön doğrultusunda da, yataylaşmış taramanın yapılması ve ilk siyah pikseli içermekte olan pikselin satırına varıncaya dek çerçevenin yukarı tarafa çekilmesi işlemi yapılır. Bu işlemler sayesinde, karakterde çerçevenin kenarları üzerinde, siyah pikseli içermemekte olan satırların ve sütunların ortadan kaldırılması sağlanmaktadır. Karakterin boyu, çerçeve üstü ve altı sınırlar arasında kalan yatay piksellerin satırlarının sayısı kadar olacaktır.

El yazısı örnekleri üzerinde satırların ayıklanması işleminin, yukarı kısımda bahsedilen yöntemle yapılmasından dolayı, hataların ortaya çıkması görülebilmektedir. Bunun nedeni, satırlardaki eğimin var olabilmesi ihtimalidir. Yukarı kısımdaki yöntemin, yazı örneğinin yataylaşmış doğrultu yönünden taranması ve pikselleri içermemekte olan bir piksel satırını araması yolu izlenir. Ama iki satırın arasında bazı zamanlarda, her bir piksel satır, siyah piksel içerebilir. Haliyle, bu satırların fark edilememesi ve iki satır bir satır şeklinde taramak hali ortaya çıkabilir. Bunun önlenmesi için ise, histogramlardan faydalanılmaktadır. Girişteki satırın yataylaşmış histogramı alınır. Bu yapılırken de satırların yataylaşmış doğrultu yönünde taranması ve bütün piksellere dair en, boy satırlarında bulunan siyah piksel sayıları saptanır. Söz konusu siyah piksellerin sayısı, histograma dair satır değerini verir. Bunun ardından histogramın üstünde, dikeyleşmiş doğrultu yönünde bir eşik seviyesinin belirlenmesi işlemi yapılır. Bu eşik üzerinde tarama işlemleri gerçekleştirilir. Eşik seviyesinden geçmekte olan tepelerin belirlenmesi sağlanır. Bu tepeler arasında oluşan boşluklardaki orta noktalar üzerinden yataylaşmış çizgilerin çizilmesi gerçekleştirilir. Söz konusu çizgiler, iki satırın ayrıldığı çizgileri ifade eder. Bu siyah piksel sayısı histogramın o piksel satırındaki değerini verir.

Daha sonra histogram üzerinde, dikey doğrultuda bir eşik seviyesi belirlenir. Bu eşik seviyesi üzerinde tarama yapılır. Eşik seviyesini geçen tepeler belirlenir. Bu tepelerin arasındaki boşlukların orta noktalarından yatay çizgiler çizilir. Bu çizgiler iki satır ayıran çizgilerdir.

Satırların ayıklanma işleminin ardından, harfleri de ayıklamak gereklidir. Ancak el yazılarında genelde harflerin bitişikleşmiş şekillerde yazılması dolayısıyla, ilk metodun bu noktada işlevsel olmayacağı açıktır. Haliyle harflerin, bir takım farklı yöntemler ile ayrıştırılması gerçekleştirilir.

Kelimeleri ayrıştırmak amacıyla eşik değeri belirlenmesi yapılır. Söz konusu değerin, iki kelimenin arasında ortaya çıkan, olası boşluğun büyüklüğü şeklinde olduğu söylenebilir. Bütün satırlar dikeyleşmiş doğrultularda taranıp, geçilmiş olan boş piksellerin kolon sayıları, eşik seviyesini aşarsa, boşluğun tespit edildiğine kanaat getirilir. Böylelikle kelimelerin ayıklanması gerçekleşir ve bu kelimelerin, harflerin ayrıştırma algoritmalarına sunulması sağlanır.(23)



Şekil 3.5. Latin Harfleriyle Örnek Satır Belirleme [19]

Satır belirleme işlemleri tamamlandıktan sonra her bir metin satırındaki kelimelerin tespit edilmesi gerekmektedir. Bu aşamada ilk olarak satırın üst ve alt sınırları arasında yatay başlangıç noktasından başlanarak dikey doğrultuda tarama yapılır. Osmanlıca'da sağdan sola tarama yapılır. Satır belirlemede olduğu gibi sadece beyaz piksellerden oluşan sütunlar geçilir, siyah piksel içeren ilk sütun, karakterin başlangıcı olarak kabul edilir. Tarama işlemine devam edilerek tamamı beyaz piksellerden oluşan ilk sütun ise kelime/harf grubunun sonu kabul edilir. Bu işlem tüm satırlara, satır başından sonuna kadar uygulanarak belgedeki kelimelere ulaşılmış olur. Latin alfabesi ile yazılı dokümanlarda karakter aralıkları ve kelime aralıkları standart olduğundan satır parçalama ile kelimelerin elde edilmesi kolaydır. Ancak Osmanlıca yazılarda karakter boşlukları standart değildir. Harfler genelde bitişik yazıldığından karakterlerin ayrı ayrı parçalanması zordur. Ayrıca bazı durumlarda dikey yönde uzantısı birbirinin üstüne gelen harflerden dolayı beyaz piksel sütununa rastlanmadığından kelime boşluğu tespit edilemez ve parçalama gerçekleşmez. Böylece iki farklı kelime, iki resim gibi algılanması gerekirken, tek bir resim gibi işlem görebilir. Bu yüzden Latin alfabesinden farklı olarak, Osmanlıca'da karakteri tespit etmek yerine kelime ya da harf grubunu tespit etmek, anlamsal çıkarımın gerekli olmadığı durumlarda daha doğru bir işlem olacaktır.

Elde edilen harf gruplarının çevresel boşluklardan arındırılması gerekir. Bunun için görüntünün resim matrisi, alt üst ve yanlardan taranarak tamamı 1'lerden oluşan satır ve sütunlar hariç tutularak; siyah piksel içeren ilk satır ve sütunun numaraları belirlenip bir alt matris oluşturulur. Bu alt matris, resmin çevresel boşluklarından arındırılmış şeklinin sayısal formudur.

El yazısıyla yazılan karakterlerin, farklılaşmış boyutlar içerebileceğinden dolayı, ölçeğin farklılıklarının düzenlendiği bir normalize etme işlemi uygulanır. Normalleştirme işleminin sonucunda farklı boyutlar içeren karakterlerin, öngörülmuş olan boyutlar içerisindeki sabitleşmiş çerçevenin içine, tam manasıyla yerleşebileceği

şekilde düzenlenmesi sağlanır.(24) Bu uyarlamanın sayesinde, farklılaşmış karakterler içerisinde çıkarılmış nitelik bilgiler arasında bulunan ilişkileri koruma durumu sağlanmış olur.

El yazısındaki karakterlerin tanınması uygulamasında, meyilin ve eğimin düzeltilmesi işlemi, son derece önemlidir. Meyili, ana hattın yataylaşmış yöne doğru sapması şeklinde açıklamak mümkündür. Eğimi ise dikey yöne yakınlaşmış çizgilere dair ortalamanın dikeyleşmiş doğrultu yönünde sapması şeklinde ifade edebiliriz.

Bu işlemde kullanılan, iki farklı metodun varlığından bahsetmek mümkündür. Bunlardan ilki, ilk başta meylin açısının daha sonra eğimin açısının düzeltilmesi, ikincisi ise ilk önce eğimin daha sonra meylin açısının düzeltilmesi şeklindedir.(25)

Kelime görüntüsünün, -40 ile 40 arasında 10'ar dereceyle kendi ekseninde döndürülmesi, bütün döndürmelerde ortaya çıkan kelime görüntüsünü yüksekliğe bağlı değerler açısından karşılaştırılması sonucunda, en küçük yükseklik neticesinde olan döndürme açısının, kelimeye dair olası meyil açısı olarak saptanmasıdır.

Kelime görüntüsünün üstünde, -40 ile 40 arasında 10'ar dereceyle eğim düzeltim işleminin uygulanması, her düzeltmenin akabinde elde edilecek görüntülerin, genişliğe dair değerler ile karşılaştırılması, en küçük genişlikte olan eğim düzeltme açısının, kelimeye dair muhtemel eğim açısı olarak saptanması işlemidir.

Bu aşama içerisindeki amacın, karaktere dair bir piksel kalınlığı iskeletinin üretilmesi işinin yapılmasıdır. Karaktere dair nitelik setini çıkarmanın gerektiği hallerde, inceltme işlemine gereksinim duyulmaktadır.

İnceltme algoritmalarının sahip olması gereken nitelikler; çatallanmanın mevcut olmaması, bir piksel kalınlığı elde edilecek olması, bağları muhafaza edebilmek, enformasyon kayıplarının yaşanmaması gibi hallerdir.

Son yıllar içerisinde pek çok inceltme algoritması geliştirilmiş, bir takım klasik algoritmaların müspet sonuçları elde etmesine karşın, eksiklerin de varlığı ortaya çıkmıştır. Örneğin bir takım bir algoritmalar, imajdaki bağları muhafaza etmede yetersizdirler. Datta tarafından yaratılan algoritmada, inceltmeye dair bir takım ana özelliklerin sağlandığı görülür.(26). Ancak Datta'nın algoritmasının, çok geçiş içeren tekrarlamalar kullanılmakta olduğu ve tam paralellik içerikli işleme yapabilen bir algoritma olmadığı görülmektedir.

Han ise, 5x5 maske bilgisi dayanaklı, tam paralel işleme yapan bir algoritmayı önermektedir.(27) Ancak Han tarafından yaratılmış algoritma, zamanı çok fazla harcamaktadır. Ek olarak da tüm bu algoritmaların bir takım istisnai durumlar altında verimsiz oldukları da söylenebilir.

Huang tarafından üretilmiş algoritmada, bağlanabilirliğe dair problemleri çözebilmek amacıyla, bir takım yeni kaideler ve enformasyon kaybının önlenmesi amacıyla şekil bilgileri kullanılmaktadır. Söz konusu algoritmayla, bir piksel, kalınlık ve bağlanabilirliğin elde edilmesi mümkündür. İskeletin orta eksen yönüne yakın konumlandırılması ve böylece de imaja dair topolojinin muhafaza edilmesinin mümkün olması söz konusu olmaktadır. İkili resim ise siyah ve beyaz pikseller vasıtasıyla temsil edilmektedir. 1 siyah piksel, 0 ise beyaz piksel göstergesidir.

Silme kurallarının inceltme algoritmasında bir çekirdek olduğu ve algoritmaya dair performansı oluşturduğu söylenebilir. Datta'nın çalışması, dört tane 1x3'lük şablonu ve bir tane 3x3'lük pencereyi kullanmaktadır.

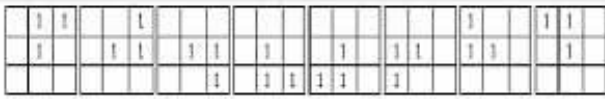

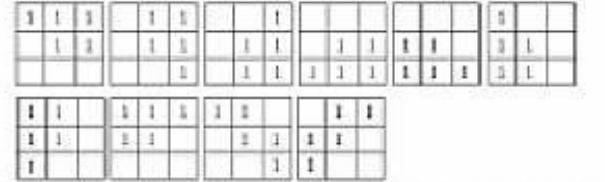



Han'ın algoritması ise, ağırlığın değerlerini(8-komşu toplamını) kullanmaktadır. Nesneleşmiş piksele ait 8-komşu ile meydana gelecek tüm çeşitler düşünülmektedir. Şekil 3.6'de tüm kuralların gösterildiği görülebilir. İlk kolonda piksele ait komşuların sayıları gösterilir. Pencerelemin birçoğunda Han'ın kurallarıyla özdeş durumların bulunmasına karşın, aralarında ana farklılıklar göze çarpar. Han'ın algoritması, 8-komşunun üzerine dayalıdır ve 8-komşuya dair ağırlık değerlerinin de hesaplanması hali vardır.

Buna ek şekilde, 5x5'lik pencerelerin bilgileri de kullanılmaktadır. Bu algoritmanın yalnızca 3x3'lük pencere bilgilerini kullandığı da bilinmelidir. Bunun yanında, şekil biçiminde bozulmalar yaratabilen ve bağların silinmesini sağlayacak bir takım elimine kuralların da kaldırılması sağlanır.

Tüm kuralların, tüm piksellere eş zamanlı olarak uygulanması söz konusudur. Ancak algoritmalar, bir takım istisnai durumlar altında, düzgün çalışmamaktadırlar. Örneğin dikeyleşmiş ve yataylaşmış yönler doğrultusunda, 2-piksel kalınlığındaki çizgilerin silinebilmesi ve bunun da nesne örüntüsünde meydana gelecek bağ kopmalarına neden olabilmesi mümkündür. Örneğin; iki piksel kalınlığa sahip olan dikdörtgen örüntüsünün tam manasıyla kaybolması hali vardır. Bağlanabilirlik halinin muhafazası amacıyla, söz konusu pikselleri silmek gerekebilir. Bunun yanında, 2-

piksel kalınlığına dokunulmaması durumunda, iskeletin bir piksel kalınlığını ifade etmesi mümkün olamamaktadır.

Datta'nın algoritması, her geçişin tek bir şablonla uygulanmasını ve bunun da çıkışının diğer geçişlere uygulanmasını içerir. Bağlanabilirliğin ve bir piksel kalınlığının garanti edilmesi hali mevcuttur.

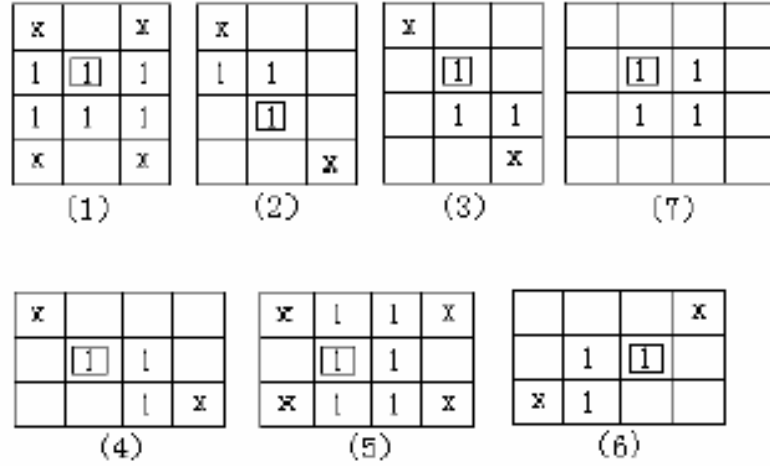
Değer	Silme Kuralları
0	Yok
1	Yok
2	
3	
4	
5	
6	
7	
8	Yok

Şekil 3.6. Silme kuralları. Bos kutu beyaz pikseli gösterir.(31)

Jang, söz konusu problemin çözümü açısından on tane 3x4, 4x3 ve 4x4'lük pencereler kullanmaktadır.(28)

LeiHuang'un algoritması ise, Jang'daki şablonların değiştirilmesiyle kullanımını içerir. Oluşacak yeni şablonların gösterimi Şekil 3.7'da görülmektedir.





**Sekil 3.7.** Muhafaza şablonlarının gösterimi. Boşluklar beyaz pikseli ifade eder. “X” dikkate alınmaz.(32)

İnceltme Algoritması:

1. Arama tablosunu oluştur.
2. Her bir pikselde indeks numaraları hesabı yap.
3. Piksel silinmesi olacaksa, indeks numarasını ve arama tablosunu kullan. Şayet pikselin silinmeyeceği bir durum varsa 5. Adıma git.
4. Nesnedeki kalınlığı hesap et. Şayet, 2-piksel kalınlığını sağlamazsa sil. Aksi halde 3x4, 4x3 ve 4x4'luk şablonlar kullan. Şayet şablonların birisine uyuyor ise pikseli muhafaza et, hiçbirisine uymuyor ise sil.
5. Silinebilecek pikselin kalmayacağı duruma kadar, 2-4 adımlarını tekrar et.

Nagendraprasad-Wang-Gupta inceltme algoritmasına dair simetrik bir model, ekstra hesaplama maliyetleri getirmeksizin, el yazısı karakterlerinin basit ve zarifleşmiş iskeletlerinin üretilmesini sağlar.(29)

### 3.2. Özellik Çıkarma

Görüntülerde yer almakta olan ve farklılık içeren her bilginin, özellik şeklinde tanımlanması mümkündür. Bu farklılıklar, görüntünün piksellerine yansımış olan karakteristiklik içeren niteliklerdir. Piksel gri tonlarında meydana gelen değişimleri, arka zeminlerdeki nitelikleri, dokuların yapısını ve lokal dokusal farklılıkları bunlara örnek şeklinde açıklayabiliriz.

Şekil tanıma yönünden ise özellikleri, rastgele bir nesnenin sahip olduğu fiziki biçimin karakterize edildiği bilgiler şeklinde tasvir edebilmemiz mümkündür. Nitelikler, söz konusu bilgileri görüntülerden yahut şekillerden ayrıştırmak sonucu seçebilme sonucunda ele geçen sayısal verileri ifade eder. Nitelik bilgilerini çıkartmada, şeklen yer alan köşe noktalar, kenar çizgiler gibi yapısallık içeren nitelikleri tanımlamakta olan verilerle beraber, yine şekil dönüşümü yöntemleriyle (moment, fourierv. b.) ele geçen verilerin de kullanılabilmesi mümkündür (22).

Şekil tanıma işlemlerinde, pek çok nitelik saptama metodunun hangisinin ya da hangilerinin kullanılacağıın, seçilmiş olan uygulamanın doğrultusuyla belirlendiği söylenebilir. Şekil tanıma işlevini görmesi açısından üretilmiş olan bilgiler ile bir nitelik setini meydana getirme işleminin, sağlamayı amaçlamakta olduğu bilgiler şunlardır;

a) Özdeş sınıflara dair örnekler için nitelik bilgisindeki değişimler küçük olmak durumundadır.

b) Başka başka sınıfların arasında olan değişimlerin ise büyük olması gerekmektedir.

Nitelik bilgilerindeki bu şartların taşınma oranı, tanımanın verimliliğiyle direkt bağlantılı bir faktörü oluşturur.

Bu nitelikler, gri tondaki görüntülerin üstünden ve histogram fonksiyonları kullanılmasıyla ele geçirilen niteliklerdir. Histogram fonksiyonunu ise bir görüntünün içerisinde olan gri tonlara dair piksel seviyesi değerlerindeki oluşum sıklığının tanımlanmasında bir ihtimal dağılım fonksiyonu (30). şeklinde açıklayabiliriz.

Öte yandan histogram fonksiyonunun, nesnelere arka zeminlerden ayrıştırmada kullanılma işlevleri de vardır. Eşik değerlerini, fonksiyonun vadi noktası seviyesine karşılık gelecek gri seviye değeriyle seçmek gerekmektedir. Histogram fonksiyonu birden çok vadi noktasının üretilmekte olduğu görüntüleme örnekleri açısından, çok seviyeli eşik işlemleriyle gerçekleştirilmektedir.(31)

Histogramla ele geçen nitelik bilgilerini şu şekilde sıralayabiliriz: Serpinti (dispersion) etkisi, ortalama değer, değişinti (variance), ortalama enerji, kaykılma (skewness) etkisi, ortalama değer çevresinde yoğunlaşma (kurtosis).

Histogramdaki darlık, düşük seviyede kontrastı tanımlar. Fonksiyonun gri tonun

ekseni üzerinde sağ yahut sola yaklaşık oluşu ise, ışığa doygunluk ve yetersizliğin derecesinin görülmesini sağlar.

Nesnedeki fiziki yapıyı yansıtmakta olan profile şekil ismi verilmektedir. Şeklin özellikleri ise, ikili forma dönüştürülen görüntünün aracılığıyla elde edilir.

Sınır eğrisi (contour), nesnenin şeklinin kuşatıldığı, birbirleriyle bağlantı içeren noktalarla oluşmaktadır. Bir nesneyi, sınır eğrisiyle temsil etmek, görüntü içerisinde dokularla ortaya çıkan niteliklerin yok sayılmasını ve şekli daha az veriyle tanımlamayı sağlamaktadır.

İkili görüntü içerisinde sınır eğrisi üretmek, sınır eğrisi çıkarmaya yarayan algoritmalar vasıtasıyla gerçekleştirilebilir. İşlemin sonunda, tek boyutta diziler halinde elde edilmiş olan sayısal verilerin, iki boyuttaki sınır eğrilerini temsil etmekte olan açılımlar oldukları söylenebilir. Zincir kodlar, sınır eğrisi açılımı yöntemlerinin birisini oluşturur.

İlk defa Freeman'ın tanımlamış olduğu zincir kodların, 4'lü yahut 8'li piksel komşulukları uyarınca tanımlanmış olan standart doğrultudaki vektörler ile oluştuğu söylenebilir.(32)

Sınır eğrilerindeki zincir kodların açılımları, sınır eğrisi piksellerinin sıralanmış diziliminin yansıtılmasıdır. Sıralamalar, piksellerin izleme yönlerine göre gerçekleşir. Bu şekil altında açılımın, sınır eğrisi noktalarındaki tek boyutlu bir dizi halinin tanımlanmasını sağladığı görülebilir.

### **3.3. Dilimleme**

Dilimleme, kelimedeki karakterleri ayrıştırma işlemi olarak karşımıza çıkar. Karakterleri, tanıma algoritmalarına gönderme işleminden önceki en mühim aşamayı oluşturur. Bunun da nedeni, doğru tanımlamaların yapılabilmesi açısından, doğru ayrıştırmaların yapılmasının gerekliliğidir.

Günümüz dünyasında, el yazılarındaki karakter ayrıştırmaları tam manasıyla çözümü sağlanamamış bir problemi teşkil eder. Son yıllar içerisinde bu konuyla alakalı pek çok çalışmanın varlığından bahsedilebilir. Karakterleri dilimleme stratejilerini dört grupta incelemek mümkündür(33):

**a)Parcalamayontemi ile dilimleme:** Karakterle benzeşen niteliklerin

kullanılmasıyla, dilimlerin tanımlanması gerçekleştirilir. Görüntünün, anlam ihtiva eden bileşenler vasıtasıyla kesilmesi işlemiparçalama(dissection) işlemidir. Parçalamanın, görüntüleri, şekil niteliklerinin özel sınıfları kullanılmaksızın, akıllı analizinin yapılması işlemi şeklinde de tanımlayabilmek mümkündür.

İyi bir dilimlemenin, geçerlilik içeren karakterlerin taşıdığı parçaları taşıması için mühim bir kriteri oluşturduğu söylenebilir. Kullanılmakta olan parçalama tabanındaki metotlar, karakterlerin sıklığıyla ve beyaz alanların dikeyleşmiş projeksiyon analizinin, birleşik bileşen analizinin dönüm noktalarını kullandığı söylenebilir.

**b)Tanıma tabanlı dilimleme:** Evveliden tanımlanmış olan sınıfların, eşleşmiş görüntü bileşenleriyle bulunmasına çalışılır. Dilimlemenin tanımadaki güvenceyle yapılması ve bu güvencenin, sonucun söz dizimi ve anlam içeren doğruluklarına bakılmak suretiyle kuvvetlendirilmesi ile devam etmesi söz konusudur. Bu yaklaşım altında, iki değişik metod vardır; bir takım arama işlemlerini gerçekleştiren metod ve görüntü özellikleri kullanan metod.

İlk yöntem, nitelik tabanda parçalama algoritmaları kullanılmaksızın, kelimeleri karakterlere ayırıştırma amacını taşır. Kavram şekliyle bu yöntemler, makinelerde basılmış yazıların tanınması amacını içeren bir takım semaları kullanmaktadırlar. Ana prensip, karakterin tanınmasıyla doğrulanmış, olası dilimleme dizileri sağlayabilmek amacıyla, değişebilmekte olan genişlikler sahibi bir pencereyi kullanmaktır. Bir başka tekniğiye, dinamik programlamalarıyla sinirsel ağları birleştirme olduğu söylenebilir.(34)

İkinci yöntem, görüntünün tamamından elde edilmiş uzaysal niteliklerin, altküme sınıflandırılması sayesinde görüntünün dilimlenmesidir. Söz konusu yöntem de kendi içerisinde ikiye ayrılmaktadır: Saklı Markov Modeli tabanındaki yaklaşımlar ve Markov olmayan yaklaşımlar. Saklı Markov Modelinde amaç, el yazısının, direkt olarak gözlenmesi mümkün olmayan olasılıktaki bir yapı olarak modelize etmektir. Literatür içerisinde, tanımaya dair uygulamalarda kullanılmakta olan söz konusu yaklaşımın hakkında çokça bilginin varlığından söz edilebilir.(35)

Markov olmayan yaklaşımların, kapalı nesnelere tanımlamada kullanılmakta olan, makine görüşü içerisinde kullanılmakta olan kavramlardan kaynaklandığı söylenebilir. Söz konusu tip tanıma tabanındaki yaklaşımlar, geri yönlü eşleştirme ve

düzenlilik kavramlarının kullanılmasıyla işler.

**c)Karışıklık içeren stratejiler:** Parçalama ve arama metotlarının hibrit bir metotla birleştirmektedir. Görüntülere parçalama algoritmaları uygulanmakta, ancak bunun sonucunda aşırı sayıda dilim meydana çıkmaktadır. Yani kelimeler içerisindeki karakterlerin sayısının da üstünde dilimin ortaya çıkması hali vardır. Bunun akabinde, söz konusu dilimlerdeki alt kümelerin bilgilerini taşıyan dilim noktalarının tespit edilmesi hali ortaya çıkar.(36)

**d)Holistikstratejiler:** Kelimelerin bütüncül olarak tanınmasına çalışır. Bu şekilde kelimeyi dilimlemeye olan gereksinim ortadan kalkar. Kelimelerin bütünü üzerinden çıkarılmış olan nitelik bilgileri, veritabanı içerisinde mevcut olan kelime nitelikleriyle eşleştirilerek tanınır. Söz konusu tekniklerin, dinamik programlamayla yahut Markov zincirleri tabanlı şekilde olmaları hali vardır. Holistic yöntemler, genel olarak özel bir sözlükle sınırlı şekildedirler.(37)

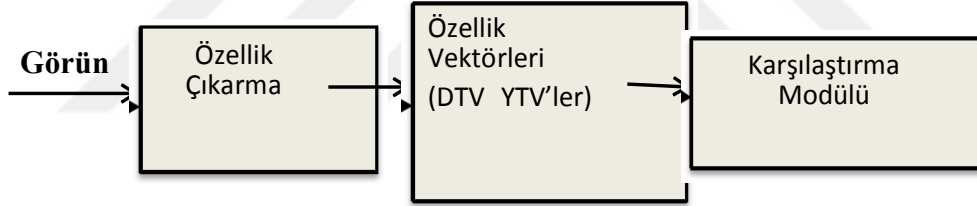


## 4. KULLANILAN YÖNTEMLER

### 4.1 Özellik Çıkarma

Özellik çıkarma; görüntü üzerinde bazı ölçümler yaparak, görüntünün özelliklerini çıkarmak ve bunları bir özellik vektörüne aktarmaktır.

SP den gelen görüntü dikey ve yatay olarak taranıp, görüntüye ilişkin Dikey Tanımlama Vektörü (DTV) ve Yatay Tanımlama Vektörü (YTV) elde edilir. Bu vektörler karakterlerin dikey ve yatay yönlerde sıkıştırılmış şeklidirler. Resmin boyut özellikleri ortadan kalktığından, aynı karakterin farklı boyutlardaki görüntülerinin özellik vektörleri birbirine benzerdir[3].



Şekil 4.1. Özellik Çıkama Aşamaları

### 4.2 Şablon Eşleme

Şablon Eşleme, görüntü içerisinde belirli işaretleri ayırmak için kullanılan bir tekniktir. Bu belirli işaretler; tek bir nokta, çizgi veya tüm bir nesne olabilir. Şablon Eşleme Yöntemi, görüntü içerisinde aranacak olan işaretin/işaretlerin şablon görüntülerini kullanarak karşılaştırma esasına dayanır. Şablon eşleme yöntemi büyük bir görüntüde şablon görüntüsüne benzeyen küçük parçaların bulunması esasına dayanmaktadır. Şablon görüntüsü büyük görüntü üzerinde dolaştırılır ve her pikselde eşleme yapılarak benzerlik ölçülür. Benzer şekillerin piksel koordinatları kaydedilir. Görüntü işlemede çok sık kullanılan bir yöntemdir. Görüntüde benzer küçük parçaların bulunması esasına dayanır. Şablon görüntüsü büyük görüntü üzerinde dolaştırılır ve her pikselde eşleme yapılarak benzerlik ölçülür. Benzer şekillerin

piksel koordinatları kaydedilir. Temel kelimeler belirlenir[3].

### 4.3 Destek Vektör Makinaları

Destek Vektör Makineleri (DVM) sınıflandırma ve kestirim problemlerinin çözümü için kullanılan istatistiksel bir yöntemdir. 1960'lı yılların sonlarında Vapnik tarafından teorik temelleri atılmıştır. 1995 yılında ise yine Vapnik tarafından ilk defa bir sınıflandırma probleminin çözümünde kullanılmıştır. DVM iki sınıfa ait örnekleri birbirinden ayırmaya çalışır. Bunun için bir örnek uzayında örnekleri mümkün olduğu kadar sınıflandıran en uygun ayırıcı düzlemi (hyperplane) arar. DVM, iki sınıfa ait örnekler arasındaki mesafeyi maksimum yapmayı amaçlamaktadır. İki sınıfa ait örnekleri ayıran birçok ayırıcı düzlem olabilir. En uygun ayırıcı düzlem ise iki sınıfa ait ve sınıra en yakın iki örnek arasındaki mesafenin en büyük olduğu durumu sağlayan aşırı düzlemdir. Optimum ayırıcı düzlemin sınırlarına (margin) en yakın örnek vektörlere destek vektörü denir.

DVM'ler sınıflama yaparken; yapay sinir ağları ve diğer öğrenme makinelerin kullanmış olduğu deneysel risk minimizasyonu (ERM) ile beraber aynı zamanda yapısal risk minimizasyonunu (SRM) da kullanır. Deneysel risk minimizasyonu örnekler arasındaki hata oranını azaltmaya çalışır. Yapısal risk minimizasyonu ise bütün örnekler üzerinde oluşabilecek hata oranı üst sınırının minimum olmasını sağlar.

DVM ile optimum ayırıcı düzlem bulunurken önce ortaya çıkan optimizasyon problemi formüleştirebilir daha sonra bu problem ikinci derece programlama (quadratic) ile çözülür. Örnek (eğitim) vektörler doğrusal olarak bir ayırıcı düzlem ile ayrılamiyorsa, bu örnekler giriş uzayından çok boyutlu öznitelik uzayına haritalanır ve bu uzayda en uygun ayırıcı düzlem aranır. Öznitelik uzayında yüksek dereceli işlem yükünden kurtulmak için çekirdek (kernel) fonksiyonları kullanılır ve en uygun ayırıcı düzlem bulunur.

DVM sinir ağları, radial tabanlı fonksiyon ağları ve polinomal sınıflandırıcıları kendi yapısında içermektedir. Doğrusal olmayan girişleri yüksek boyutlu öznitelik uzayına dönüştürüp orada doğrusal yöntemleri kullandığından matematiksel olarak analizi son derece basittir.

DVM'ler karakter tanıma, ses tanıma ve örüntü tanıma gibi konularda birçok



uygulama alanı bulmuştur.

### 4.3.1 Doğrusal DVM için en uygun ayırıcı düzlem

Bu bölümde Vapnik tarafından geliştirilen ve eğitim verilerini hatasız şekilde ayırabilen un uygun ayırıcı düzlem yönteminden bahsedilecektir

$l$  tane örnekten oluşan bir eğitim kümesi düşünelim.

$$(x_1, y_1), \dots, (x_n, y_n), \quad x_i \in R^d, \quad y_i \in \{-1, +1\} \quad (1)$$

uygun  $w$  ağırlık vektörü ve  $b$  eşik değeri için aşağıdaki eşitsizlikleri sağlıyorsa o zaman (4.1) deki eğitim kümesi doğrusal olarak ayrılabilir.

$$\begin{aligned} w \cdot x_i + b &\geq 1 & y_i &= 1 \\ w \cdot x_i + b &\leq -1 & y_i &= -1 \end{aligned} \quad (2)$$

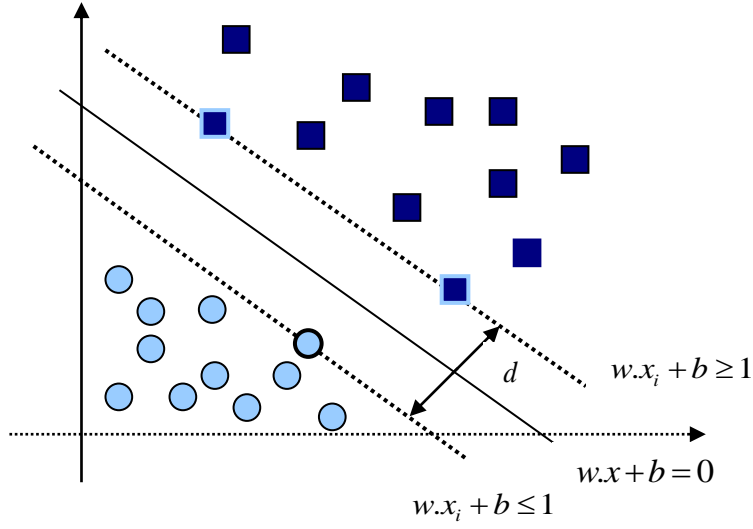
(2) deki eşitsizlik aşağıdaki şekilde genelleştirilebilir.

$$y_i(w \cdot x_i + b) \geq 1, \quad i = 1, \dots, l \quad (3)$$

En uygun ayırıcı düzlem ise karar sınırı (3) ile gösterilen doğrudur.

$$w \cdot x + b = 0 \quad (4)$$

En uygun ayırıcı düzlem tektir (unique) ve eğitim datasını en büyük sınır ile ayırır. Sınır (margin), ayırıcı düzlem ile eğitim verileri arasındaki en kısa uzaklık olarak tanımlanır. Eğer sınır maksimum uzunlukta ise ayırıcı düzlem en uygun (optimum) olur. Şekil 4.2'de iki boyutlu bir uzayda iki sınıfa ait verileri ayıran bir ayırıcı düzlem örneği gösterilmiştir. Noktalı doğrular ayırıcı düzlemin karar sınırlarını gösterir ve bu yüzeye en yakın vektörlere destek (support) vektörler denir. Karar sınırları sadece destek vektörleri ile belirlenir.  $d$  uzunluğu en uygun (optimal) sınır uzunluğunu gösterir.



**Şekil 4.2.** İki boyutlu uzayda iki sınıf veri kümesi örneği.

Ayırt edici düzlem üzerinde iki karar sınırı arasındaki  $d$  mesafesi aşağıdaki gibi yazılabilir.

$$d = \frac{2}{\|w\|} \quad (5)$$

$\|w\|$  sembolü,  $w$ 'nin normunu göstermektedir.  $d$  mesafesinin en büyüklenmesi  $\|w\|$ 'in en küçüklenmesine eşit olduğu için en uygun ayırt edici düzlem denklem (3)'deki kısıtları sağlayan ve  $\|w\|$ 'yi en küçükleyen en iyileme problemi ile elde edilebilir. Bu en iyileme problemini çözmek Lagrange çarpanları yöntemi kullanabilir.

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i [y_i (w \cdot x + b) - 1] \quad (6)$$

Burada  $\alpha_i \geq 0$  Lagrange çarpanlarını göstermektedir.  $L(w, b, \alpha)$  fonksiyonu  $\alpha_i \geq 0$  değerlerine göre en büyüklenerek,  $w$  ve  $b$  değerlerine göre ise en azlanarak çözülebilir.  $L(w, b, \alpha)$  fonksiyonunu  $w$  ve  $b$  değerlerine göre ise en minimum nKTalarını bulabilmek için  $L$  fonksiyonunun  $w$  ve  $b$  değerlerine göre türevleri sıfıra eşitlenir.

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \quad (7)$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \quad (8)$$

(7) ve (8)' deki türevler çözüldüğünde (9)'daki kısıta bağlı olarak, en uygun ayırıcı düzlemin çözümünün (10)'da gösterildiği gibi, eğitim vektörünün ( $x_i$ ) doğrusal kombinasyonu şeklinde yazılabildiği görülür.

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad \alpha_i \geq 0, \quad i = 1, \dots, l \quad (9)$$

$$w = \sum_{i=1}^l \alpha_i y_i x_i \quad \alpha_i \geq 0, \quad i = 1, \dots, l \quad (10)$$

Artık (6) denklemi fazla değişkenli olmaktan kurtulur ve  $\alpha_i$  Lagrange çarpanlarına bağlı olarak yazılabilir.

(3)'deki eşitsizlik, karar sınırları üzerindeki noktalarda yani destek vektörlerinin bulunduğu noktalarda aşağıdaki eşitlik haline geldiği açıktır. Khun-Tucker teoremi kullanılarak (3) eşitsizliğinin  $\alpha \neq 0$  farklı olduğu noktalarda aşağıdaki eşitliğe dönüştüğü bulunur.

$$\alpha_i [y_i (w \cdot x_i + b) - 1] = 0, \quad i = 1, \dots, l \quad (11)$$

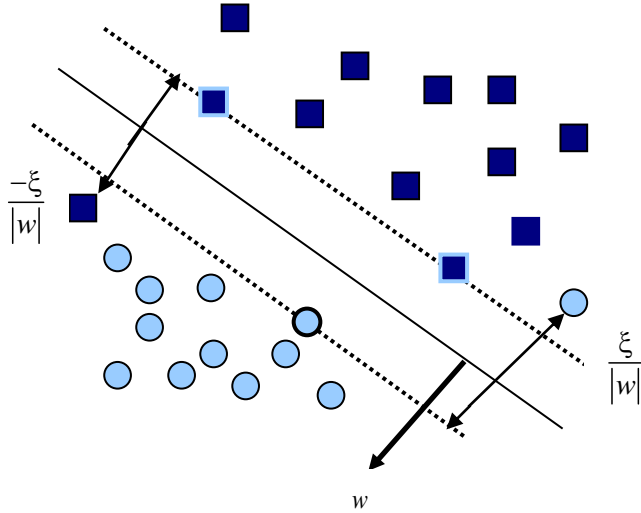
(11) eşitliğini sağlayan başka bir deyişle  $\alpha \neq 0$  farklı olduğu noktalar destek vektörleridir. (9) ve (10) denklemleri (6). Denklemden yerine konulup tekrar düzenlenirse

$$L(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (12)$$

olarak yazılabilir. Burada  $L(\alpha)$  fonksiyonu (9) kısıtına bağlıdır ve  $\alpha_i$  çarpanlarına göre en çoklanarak çözülür.

### 4.3.2 Doğrusal olarak ayıramayan durumlar

Eğitim verileri her zaman hatasız olarak ayıramayabilir. Bu tür durumlarda verilerin en az sayıda hata ile ayrılması istenir. Şekil 4.3'de doğrusal olarak ayıramayan durum gösterilmiştir.



**Şekil 4.3.** Eğitim verilerinin doğrusal olarak ayıramadığı durum örneği

Bir önceki bölümde sınıflandırma problemi için eğitim verilerinin doğrusal olarak ayrılabilirdiği ve en iyi ayırt edici düzlemin  $\|w\|$ 'nin en azlanarak bulunduğunu gösterilmiş ve tek bir destek vektörünün en iyi ayırt edici düzlemi tanımlamada yeterli olduğu söylenmişti. Şekil 4.3'de gösterilen ve verilerin doğrusal olarak ayıramadığı durumlarda en iyi ayırt edici düzlem  $\xi_i$  gevşek değişkenler tanıtılarak sağlanır.  $\xi_i$  Gevşek değişkeni herhangi bir eğitim örneği için uygun olan sınıfın sınırdan olan sapma miktarıdır. Ayırt edici düzlemde sınırın içine düşen veya karar sınırın yanlış sınıfa ait tarafta bulunan eğitim verileri için genelleme yapılırsa, ayırt edici düzlem kısıtları olan (3)'deki ifadeye  $\xi_i$  pozitif gevşek değerleri eklenir.

$$\begin{aligned} y_i(w \cdot x_i + b) &\geq 1 - \xi_i, & i = 1, \dots, l \\ \xi_i &\geq 0 & i = 1, \dots, l \end{aligned} \quad (13)$$

Bir hatanın oluşması için  $\xi_i$  biri geçmelidir.  $\sum_i \xi_i$  ise eğitim hataları sayısının üst sınırıdır. Bu durumda en genel anlamda en büyük sınır ile en iyi ayırt edici düzlem,

$$P = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (14)$$

(13) deki kısıtlara bağlı olarak,  $P$  eşitliğini en küçük yapan  $w$  ve  $b$  parametreleri ile tanımlanır. Burada  $C$  parametresi düzenleyici parametresi olup herhangi sabit bir

değer seçilebilir. (15) probleminin çözümü için bir önceki bölümde bahsedilen Lagrange çarpanları ile ikili probleme dönüştürülmelidir.

$$L(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j (x_i x_j) \quad (16)$$

$$\sum_{i=1}^l \alpha_i y_i \text{ ve } 0 \leq \alpha_i \leq C \quad i = 1, \dots, l \quad (17)$$

(16) eşitliği en büyük yapılarak, sınıflandırma için gerekli olan parametreler hesaplanır.

#### 4.4 Dalgacık Analizi

1950’li yılların sonlarından beri Fourier Dönüşümü dönüşüm-tabanlı görüntü işlemenin başlıca dayanağı olmasına karşın son zamanlarda dalgacık dönüşümü birçok görüntünün analizinde, iletiminde ve sıkıştırılmasında daha kolaylıklar sağlamaktadır. Dalgacık dönüşümü, bir işaretin,  $\psi(t)$  örnek fonksiyonunun ölçekleme ve ötelemeleri ile oluşturulan taban fonksiyonu ile analizine dayanmaktadır. Bu taban fonksiyonları, Fourier dönüşümünde olduğu gibi sinusoidal değil, sonlu süreli ve titreşimli dalgalardır ki bu dalgalara dalgacık denir. Dalgacıkları ilk kez 1900’lü yılların başında Haar kullanmıştır

Fourier analizi ile işaretin frekans içeriği bilgisi elde edilir. Frekans bilgisi birçok sinyal için oldukça önemlidir. Fourier analizi durağan sinyallerin karakteristiğinin belirlenmesinde çok yararlı olmasına karşın durağan olmayan sinyaller için zaman bilgisi içermediğinden o kadar kullanışlı değildir. Çünkü Fourier dönüşümü ile belirli bir olayın ne zaman gerçekleştiği bilgisine ulaşmak imkânsızdır. Fourier dönüşümündeki bu eksiklik Gabor’un 1946 yılında, pencere fonksiyonu olarak tanımlanan bir sabit fonksiyonun zamanda ötelenmesi ile taranan herhangi bir işaretin Fourier dönüşümü alınarak giderilmeye çalışılmış ve bölgesel frekans analizinin yapılmasına imkân sağlamıştır. Gabor dönüşümü adı verilen bu dönüşüm 1965 yılında ortaya atılan yeni bir algoritma ile değişik pencere fonksiyonlarının kullanıldığı “Kısa Süreli Fourier Dönüşümü” (KSFD) olarak genişletilmiştir. KSFD’ler işaretin frekans bilgisinin yanı sıra zaman bilgisinide içermektedir. Fakat bu bilgi kullanılan pencerelerin sabit büyüklükte olmasından dolayı kısıtlıdır.

KSFD'deki bu kısıtlamayı ortadan kaldırmak amacıyla, sabit genişlikteki pencereler yerine işaretteki düşük frekansları yakalamak için geniş pencere fonksiyonları ve yüksek frekansları yakalamak için dar pencere fonksiyonlarının kullanımı düşünülmüş ve sonuç olarak Dalgacık dönüşümü ortaya çıkmıştır

Dalgacık zamanda bellibir aralıkta osilasyon yapan bir dalgayı ifade eder. Bundan dolayı dalgacık, hem zamanda hemde frekans alanında sınırlı olan bir fonksiyondur. Dalgacık dönüşümü ile ses ve imge sıkıştırma, çpklı çözünürlüklü işaret işleme gibi uygulama alanlarının yanı sıra, özellikle iki boyutlu işaret işlemedeki imge analizinde sağladığı kolaylıklardan dolayı geniş bir kullanım alanına sahip olmuştur.

Ayrık Dalgacık Dönüşümü Dalgacık katsayılarının ölçek değerlerinde hesaplanması çok sayıda katsayının hesaplanmasına ve dopal olarak işlem yükünün artmasına neden olmaktadır. Bu yüzden dalgacık katsayıların sadece seçilen ölçek ve zaman diliminde hesaplanması büyük avantaj sağlar. Seçilmiş olan yeni ölçek değerleri ile (ölçek ve öteleme parametreleri ayrık olan) ayrık dalgacık dönüşümünü (ADD) aşağıdaki gibi yazabiliriz.

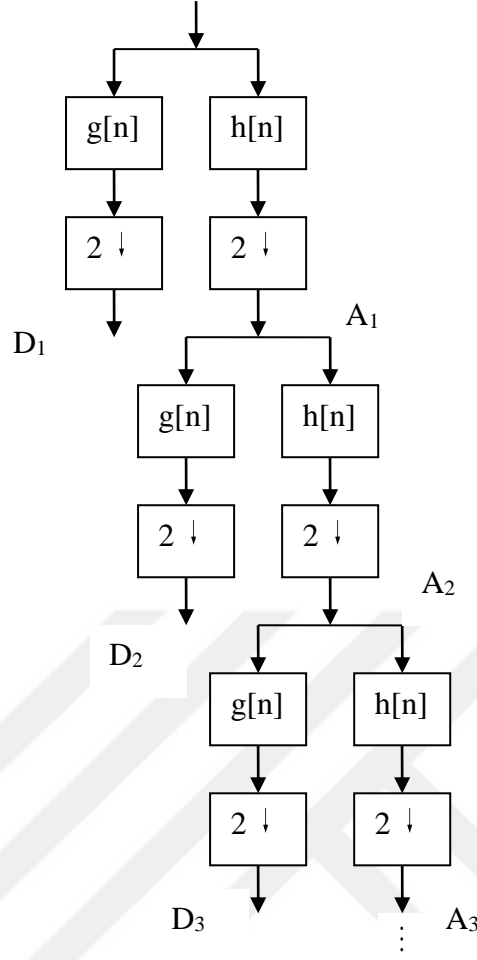
$$W[m, n] = \sum_x f[x] \psi_{m,n}[x] \quad (1)$$

burada dalgacık taban fonksiyonunu,  $a$  ve  $b$ ' yi ayrık değerler yani,  $a = 2^j$  ve  $b = k2^j$  ( $k, j \in Z$ ) olarak tekraradan yazabiliriz. O halde dalgacık taban fonksiyonu aşağıdaki şekilde yazılabilir.

$$\psi_{j,k}[x] = 2^{-j/2} \psi(2^j x - k) \quad (2)$$

Dalgacık dönüşümünde  $x[n]$  işareti yaklaşık  $A_1$  ve ayrıntı  $D_1$  altbandlara ayrılır. Daha sonra yaklaşık alt bandı alçak geçiren  $h[\cdot]$  ve yüksek geçiren  $g[\cdot]$  süzgeçlerden geçirilerek orjinal işaret belli sayıda ölçeklere ayrılır. Çoklu çözünürlük olarak isimlendirilen bu işlem şekil 3'te gösterilmiştir.

$x[n]$



**Şekil 4.4.** Dalgacık dönüşümünde bir işaretin altbandlara ayrışımı

Şekil 1’de yaklaşık altbandı işaretin yüksek ölçekli ve düşük frekanslı bileşenini, ayrıntı altbandları ise işaretin düşük ölçekli yüksek frekanslı bileşenini göstermektedir.

Tek boyutlu ayrık dalgacık dönüşümü kolaylıkla görüntü gibi iki boyutlu fonksiyonlar için genişletilebilir. Bunun için bir iki boyutlu ölçekleme fonksiyonuna ( $\varphi(x, y)$ ) ve 3 tane iki boyutlu dalgacık fonksiyonuna ( $\psi^H(x, y), \psi^V(x, y)$  ve  $\psi^D(x, y)$ ) ihtiyaç vardır. Aşağıda bu fonksiyonlar tanımlanmıştır.

$$\varphi(x, y) = \varphi(x)\varphi(y) \quad (3)$$

$$\psi^H(x, y) = \psi(x)\varphi(y) \quad (4)$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \quad (5)$$

$$\psi^D(x, y) = \psi(x)\psi(y) \quad (6)$$

Yukarıdaki (3-6) denklemlerindeki dalgacıklar, bir görüntü için yoğunluk değişimlerini ölçerler.  $\psi^H$ ; görüntüdeki sütun boyunca oluşan değişimleri  $\psi^V$ ; satır boyunca gözlenen değişimleri  $\psi^D$  ise köşegenler boyunca oluşan değişimleri göstermektedir.

İki boyutlu ölçekleme ve yönlü dalgacık fonksiyonlarını aşağıdaki gibi ifade edebiliriz.

$$\varphi_{j,m,n}(x,y) = 2^{-j/2} \varphi(2^{-j}x - m, 2^{-j}y - n) \quad (7)$$

$$\psi_{j,m,n}^i(x,y) = 2^{-j/2} \psi^i(2^{-j}x - m, 2^{-j}y - n) \quad i = \{H, V, D\} \quad (8)$$

(8) denklemindeki  $i$  yönlü dalgacıklardaki  $H, V$  ve  $D$  yönlerdeki değişimlerin göstermektedir. O halde  $M \times N$  boyutundaki  $f(x, y)$  fonksiyonunun ayrık dalgacık dönüşümü aşağıdaki gibi tanımlanabilir.

$$W_\varphi(j_0, m, n) = 2^{-j} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0, m, n}(x, y) \quad (9)$$

$$W_\psi^i(j, m, n) = 2^{-j} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j, m, n}^i(x, y) \quad i = \{H, V, D\} \quad (10)$$

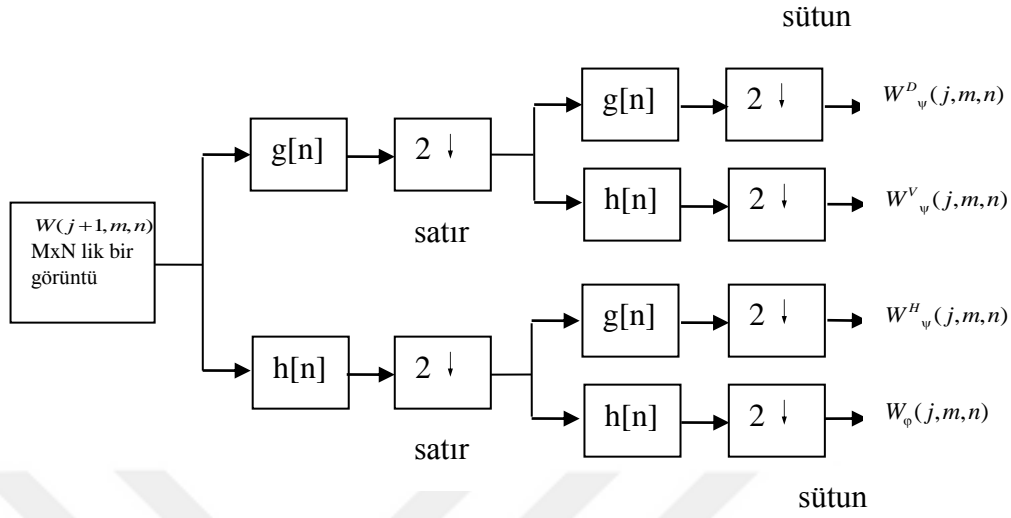
(9) denklemi  $f(x, y)$  fonksiyonunun  $j_0$  ölçeğindeki yaklaşık katsayılarını verir. Burada  $j_0$  keyfi bir ölçekleme değeri olup genellikle 0 seçilir. (10) denklemi  $j$  ölçeği için dikey, yatay ve köşegen ayrıştırma katsayılarıdır. Burada  $j = 0, 1, 2, \dots, j-1$  ve  $m, n = 0, 1, 2, \dots, 2^j - 1$  değerleri alabilir. Eğer  $W_\varphi$  ve  $W_\psi^i$  katsayıları verilmiş ise ters ayrık dalgacık dönüşümü kullanarak  $f(x, y)$  fonksiyonunu tekrar elde edebiliriz.

$$\begin{aligned} f(x, y) &= 2^{-j} \sum_m \sum_n W_\varphi(j_0, m, n) \varphi_{j_0, m, n}(x, y) \\ &+ 2^{-j} \sum_{i=H, V, D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j, m, n) \psi_{j, m, n}^i(x, y) \end{aligned} \quad (11)$$

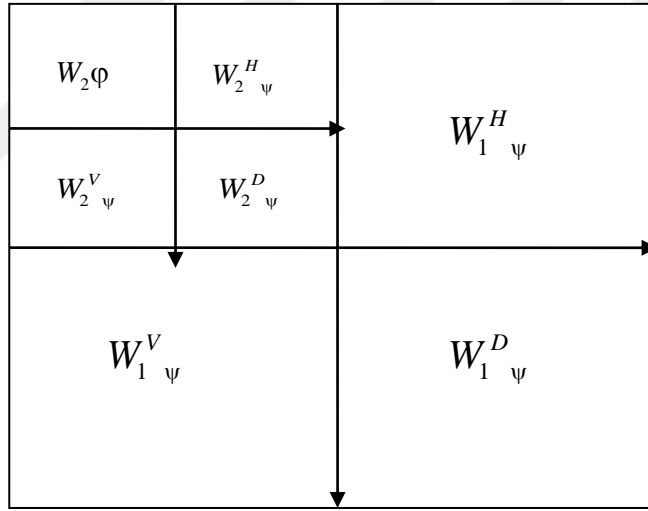
Tek boyutlu ayrık dalgacık dönüşümünde olduğu gibi, iki boyutlu ayrık dalgacık dönüşümü de sayısal süzgeçler ve veri azaltma (downsampling) işlemleri ile gerçekleştirilebilir. Her bir seviyede önce yatay olarak (satır), sonra dikey olarak



(sütun) alt geçiren ve üst geçiren süzgeçleme yapılır. Bu işlem şekil 2. ve 3 te gösterilmiştir



Şekil 4.5. İki boyutlu ADD kullanılarak tek seviyeli görüntü ayrıştırma işlemi



Şekil 4.6. ADD ile 2 seviyeli görüntü ayrıştırma

Şekil 2 te,  $h[\cdot]$  alçak geçiren  $g[\cdot]$  yüksek geçiren süzgeçleri,  $(2 \downarrow)$  ise veri azaltma işlemlerini göstermektedir.  $W(j+1, m, n)$  görüntüsü ilk kısımda yatay olarak, ikinci kısımda ise dikey olarak süzgeçlenmiştir. Bu işlemlerin sonucunda  $W(j+1, m, n)$  görüntüsü  $\frac{M}{2} \times \frac{N}{2}$  boyutlarında dört alt-görüntüye ayrıştırılmıştır.  $W^phi_psi(j, m, n)$  düşük frekan bileşeni olan yaklaşık görüntüyü,  $W^D_psi(j, m, n)$ ,  $W^V_psi(j, m, n)$  ve  $W^H_psi(j, m, n)$  alt görüntüleri ise sırasıyla; köşegen, dikey ve yatay ayrıntı görüntülerini

göstermektedir.

Ayrıştırma (decomposition) işlemi sonucunda elde edilen yüksek frekanslı ve düşük frekanslı alt-görüntüleri kullanarak eski orjinal görüntü tekrar elde edilebilir. Bu işleme birleştirme (synthesis) adı verilir. Birleştirme işleminde ayrıştırma işleminde yapılan işlemlerin simetriği yapılır.



## 5. UYGULAMA VE MATERYAL

Bu çalışmada materyal olarak, İstanbul Üniversitesi Türk İktisat ve İctimaiyat Tarihi Araştırmaları Merkezi, İBB Atatürk Kitaplığı, Diyanet Vakfı İSAM Kütüphanesi, İstanbul Beyazıd Kütüphanesinden alınan belgeler resm formatında taranmış. Her bir veri setinin 15'er sayfası eğitim, 15'er sayfası ise test verisi olarak kullanılmıştır.

Modelin geliştirilmesi için yazılım materyali olarak iki hazır programdan yararlanılmıştır.

### 5.1 Kullanılan Yazılımlar

Kod yazımı ve uygulamaların çalıştırılması için "Microsoft Visual Studio" paket programı, veritabanı için ise "Microsoft Access programı kullanılmıştır. Test araştırmaları içinde Matlab yazılımı kullanılmıştır.

Visual Studio, Microsoft'un C++, C#, VB, J#, Jscript dillerinde program yazmayı ve derlemeyi kolaylaştırmak için hazırladığı çok amaçlı bir arayüzdür. İki adet sürümü bulunmaktadır. Birisi profesyonel programcılar için lisanslı olan sürümüdür. Ötekisi, programlamaya yeni başlayanların bütün ihtiyaçlarını karşılayacak nitelikteki serbest sürümüdür.

Program ücretsizdir, lisans almaya gerek yoktur. Birçok uygulama geliştirme dili ve bu dillerle yazılmış uygulamaları çalıştırmak için ise farklı platformlar vardır. .net Framework, windows tabanlı uygulamalar geliştirebilmek için farklı programlama dillerinin birlikte çalışabileceği ortak bir geliştirme ortamıdır. Bu ortam hem yazılımları geliştirmek hem de çalıştırmak için gereklidir. .Net Framework bileşen yönelimli uygulamaların geliştirilmesini ve yürütülmesini destekleyen bir ortam tanımlamaktadır. Bu sayede; farklı programlama dilleri, farklı işletim sistemleri ile çalışabilme özelliğine sahiptir. .Net Standart ağ protokollerine ve standartlarına destek: TCP/IP, SOAP, XML, HTTP gibi standart internet protokolleri ve standartlarına destek verir. Farklı programlama dillerine destek: Programcılara istedikleri dilde geliştirme özgürlüğü sunan net bünyesinde birçok programlama

dilini barındırır. Farklı dillerde geliştirilen programlama kütüphanelerine destek: .net Framework, farklı dillerde oluşturulmuş programlama dillerinde oluşturulmuş program kütüphanelerine destek verir. Kütüphaneler uygulama geliştirmeyi kolaylaştıran bir kere yazılmış bir program parçasını tekrar tekrar yazmaktan kurtaran yapılardır. Farklı platformlara destek: .net Framework bütün windows platformları için mevcuttur. Farklı bilgi işlem platformlarından insanları, sistemleri ve cihazları birbirine bağlar[23].

Access, İlişkisel Veri Tabanı Yönetim Sistemi ile çalışan bir veri tabanı oluşturma programıdır. İlişkisel Veri Tabanı Yönetim Sistemi sisteminde bir veri tabanı dosyasında birden fazla tablo oluşturulabilir ve bu tablolar arasında birbirleriyle ilişki kurulabilir. Kurulan ilişkiler sayesinde farklı tablolardaki veriler sanki aynı tablodaymış gibi kullanılabilir. Microsoft Access bir İlişkisel Veri Tabanı Yönetim Sistemi uygulamasıdır[14]. Bir veri tabanını oluşturmak ve kullanmak Access ile diğer veri tabanı uygulamalarına göre çok daha kolaydır. Bunun nedeni Access'in, Windows ortamının Grafiksel Kullanıcı Arabiriminin sağladığı avantajların tümünden yararlanma imkânı vermesidir. Grafiksel Kullanıcı Arabirimi, karmaşık komut dizilerini öğrenmeyi gerektirmeden, ekran üzerindeki nesnelere ve simgelerin yardımıyla, fare desteğinden de yararlanarak kullanıcının çalışmasına olanak verir. Örneğin, geleneksel veri tabanı uygulamalarında iki tablo arasında bağlantı kurmak için oldukça karmaşık komut dizileri yazmak gerekirken, Access'te bu iş basit bir fare hareketiyle gerçekleştirilebilir [24].

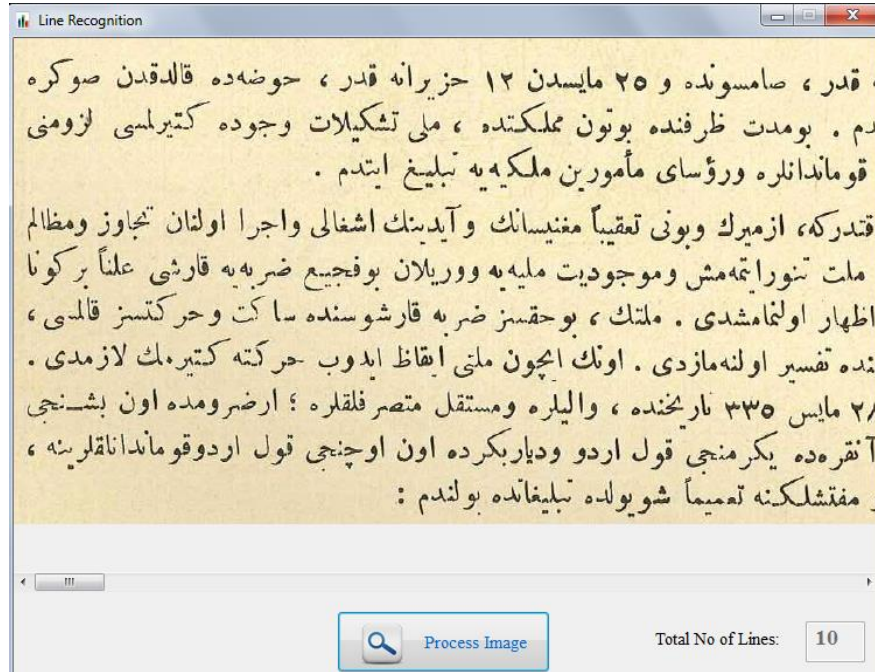
Matlab, çeşitli uygulamalar geliştirmek için tasarlanmış bir yazılım geliştirme aracıdır. Özellikle görüntü işleme konularında oldukça kullanışlıdır. Hazır fonksiyonlar yardımıyla diğer programlarda uzun süren işlemleri kendi içinde daha kısa sürede gerçekleştirmektedir. Matlab ile; Veri elde etme, veri analizi ve inceleme, görsellik ve görüntü işleme, algoritma prototipi oluşturma ve geliştirme, modelleme ve simülasyon, programlama ve uygulama geliştirme yapabiliriz. Bugün çok sayıda akademisyen, araştırmacı, bilim adamı ve öğrenci tarafından kullanılan Matlab, içinde gömülü pek çok ara yüzü ile bilgisayar dünyasının en gelişmiş teknik ve bilimsel problem çözme ve uygulama geliştirme aracıdır.

## 5.2 Uygulama

İlk olarak resim formatında taranmış belgeler bilgisayar ortamına aktarılmıştır. Osmanlıca metinlerin gerçek zamanlı olarak türkçe diline aktarılması için önemli 3 önemli süreç vardır. Bunlar sırasıyla; Satır segmentasyonu, Kelime segmentasyon, kelimelerin ve karakterlerin tanınmasıdır.

### 5.2.1 Satır segmentasyonu

Satır segmentasyonunun temel prensibi, metine ait imgenin sütunlarının sırasıyla taranması ve siyah piksellerin sütun eksenindeki konumlarının, yani satır cinsinden yerlerinin belirlenmesidir. Böylece her satırdaki alt ve üst siyah piksellerin konumuna göre resimdeki satır sayısı belirlenir. Yapılan çalışmalarda Arapça (özellikle matbu) kelimelerde karakterlerin birleşme noktalarındaki genişliğin karakterin genişliğinden çok daha az olduğu görülmüştür. Buradan hareketle, iki siyah piksel arasındaki aralık 25 değerinden büyükse kelime küçük ise harf olarak kabul edildi. Bu çalışmalar doğrultusunda öncelikle satır bölütleme işlemi gerçekleştirilmiştir. Sistemin bir metnin öncelikle satırlarını farketmesi gerektiğinden satır bölütleme işlemi çalışmasına başlandı.

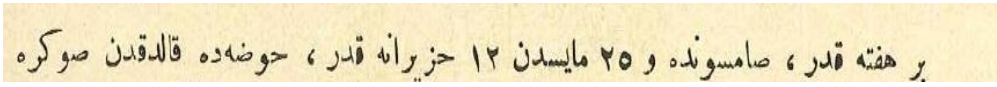


Şekil 5.1. Toplam Satır Sayısı

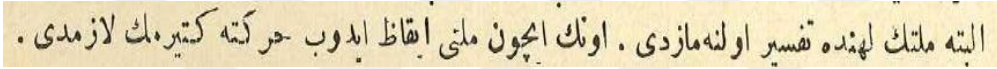
Yukarıdaki Osmanlıca metinde gördüğünüz üzere her satırın arasında belli boşluklar

bulunmaktadır. Bu boşluklar sayesinde yatay izdüşümleri hesaplanarak bir metinde toplam kaç adet satır olduğu büyük bir başarı oranıyla bulunmaktadır.

Histogram genel olarak yatay ve dikey eksenlerden oluşan bir grafikdir. Yatay eksen, tonaralığını temsil eder. Sol taraf saf siyahı (0) , sağ taraf saf beyazı (256) gösterir. Dikey eksen ise hangi tonda ne kadar piksel olduğunu temsil eder. Buna göre yukarı doğru çıkıntılar o tonda daha fazla piksel olduğunu gösterir. Bu yöntem sayesinde resmin pixellerinin azlığına, çokluğuna ve arasındaki mesafelere göre segmentasyon bölümünü gerçekleştirdik.



Şekil 5.2. Satır Örnek



Şekil 5.3. Satır Örnek2

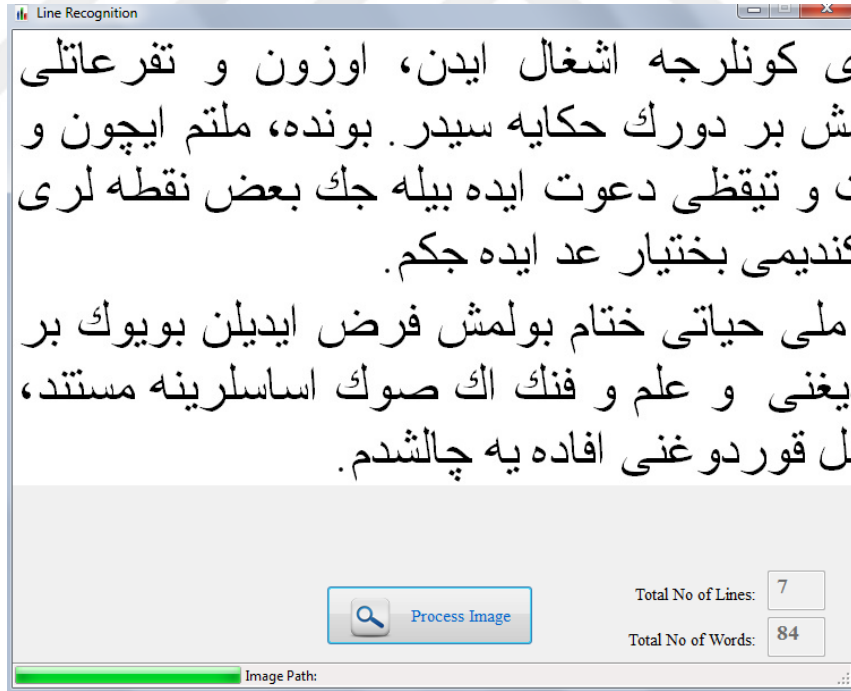
Ad	Değiştirme tarihi	Tür	Boyut
Image7(Line1words)	22.07.2013 21:54	Dosya klasörü	
Image7(Line2words)	22.07.2013 21:54	Dosya klasörü	
Image7(Line3words)	22.07.2013 21:54	Dosya klasörü	
Image7(Line4words)	22.07.2013 21:54	Dosya klasörü	
Image7(Line5words)	22.07.2013 21:54	Dosya klasörü	
Image7(Line6words)	22.07.2013 21:54	Dosya klasörü	
Image7(Line7words)	22.07.2013 21:54	Dosya klasörü	
Image(7)Line 1	22.07.2013 21:54	JPEG resmi	13 KB
Image(7)Line 2	22.07.2013 21:54	JPEG resmi	15 KB
Image(7)Line 3	22.07.2013 21:54	JPEG resmi	16 KB
Image(7)Line 4	22.07.2013 21:54	JPEG resmi	12 KB
Image(7)Line 5	22.07.2013 21:54	JPEG resmi	15 KB
Image(7)Line 6	22.07.2013 21:54	JPEG resmi	15 KB
Image(7)Line 7	22.07.2013 21:54	JPEG resmi	12 KB

Şekil 5.4. Kayıt Edilen Satırlar

## 5.2.2 Kelime segmentasyonu

Kelime segmentasyonu, satır segmentasyonunda olduğu gibi siyah piksellerdeki süreksizlik noktalarının belirlenmesini temel alır. Yapılan çalışmada Arapça (özellikle matbu) kelimelerde karakterlerin birleşme noktalarındaki genişliğin

karakterin genişliğinden çok daha az olduğu görülmüştür. Buradan hareketle, iki siyah piksel arasındaki aralık 25 değerinden büyükse kelime küçük ise harf olarak kabul edildi fakat kelime segmentasyonunda Osmanlıca gibi bağlamalı bir dil ortaya yeni bir problem çıkardı. Siyah pikseller hem kelimeler arasında hem de kelime içinde süreksizlik göstermektedirler. Bunun sebebi Osmanlıcada kelime ortasında kendisinden sonraki harflerle bitişmeyen “elif”, “dal”, “zel”, “re”, “ze”, “je” ve “vav” harfleridir. Bu harfler, kelime segmentasyonun karmaşık bir hal almasına neden oldu. Bu çalışmalar doğrultusunda öncelikle satır bölütleme işlemi gerçekleştirildi. Sistemin bir metnin öncelikle satırlarını fark etmesi gerektiğinden satır bölütleme işlemi çalışmasına başlandı. Bu işlemin ardından satırları ve kelimeleri kayıt etme işlemi gelmektedir. Kayıt edilecek kelimeler veritabanına gönderilecek, burada eğitilerek metinlerin çevrilmesine yardımcı olacaktır. Satır ve kelimeleri kayıt ederken gene kelimeler arası boşluklardan faydalanılmıştır. Program önce satırları ve kelimeler bulup, kullanıcıdan, kayıt edilecek klasör yolunu isteyerek, satırları ve kelimeleri kayıt etmektedir.



Şekil 5.5. Satır ve Kelime Sayısı Toplamı

Resimde görüldüğü üzere osmanlıca metnin toplam satır sayısı 7, metnin karakterleri toplamı 84 adettir.



Şekil 5.6. Bölünen Kelimeler

### 5.2.3 Akıllı okuma sistemi geliştirilmesi

Karakterlerin nasıl okunması gerektiği konusunda birçok kez çalışmalar yapılmıştır. Bu kısımda karşılaşılan en büyük zorluk, Osmanlıca metinlerin farklı yazı tiplerinde olmaları ve resimlerin gürültülerinin çok olmalarıdır. Gürültü; özellikle resim işleme yöntemlerinde sıkça karşılaşılan bir sorundur. Bir görüntü veya resim basit 2 değişkenin bir fonksiyonu olarak tanımlanır.  $a(x,y)$  gibi bir fonksiyonla ifade edilen bir resimde  $a$  bir şiddet birimi ve  $x$  ve  $y$  değişkenleri ise resmin gerçek koordinatları olmaktadır. Dijital metinler sayısal değerlerden oluşurlar. 1 ve 0'lerden oluşan sayısal resimler o resmin sayısallaştırılmış halidir. OCR yazılımlarının karakter tanımlaması için temel olarak üç yöntem vardır.

1-Template matching (şablon esleme):Sisteme kayıt edilen osmanlıca karakterleri, yani karşılaştırılması muhtemel tüm karakterlerin görüntüsünü içeriğinde tutar. Sonra taranmış karakterlerin karakterler ile içeriğini karşılaştırır.

2-Feature analysis (içerik analizi):Bu yöntemde ise karakterlerin görüntüsünü değil, tanımlarını aklında tutar. Karakterleri tanımak için birçok özelliğine bakar; kaç tane düz çizgi olduğu, bunların kaç dikey, kaç yatay, yuvarlak köşelerin konumlarını inceler.

3-Self-assertion (ozgun tanımlama): Bu yöntem ise yukarıdaki iki yöntemin birleşmesidir. Önce doküman içerik yöntemiyle taranarak genel bir tablo oluşturulur. Bu sayede kesinliği yüksek karakterler bir kenara ayrılarak bunlardan bir matris oluşturulur. Daha sonra bu kez emin olunamayan karakterler, bu yeni elde edilen karakter tablolarıyla karşılaştırılarak benzetilmeye çalışılır. Ocr yazılımlarına örnek vermek gerekirse bunlardan bazıları; Abby Fine Reader, Docscan ve Readiris, bunlardan en kullanışlı olanlarıdır.

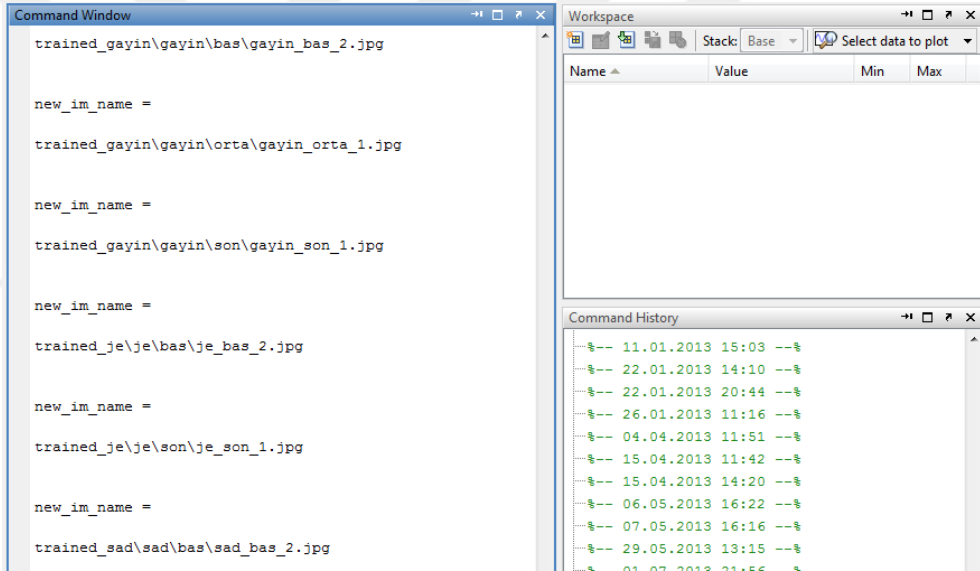
Ocr işleminde karşılaşılan en büyük sorun osmanlıca karakterler ile tanınması gereken karakterler arasındaki yazı tipi ve boyutu farkının çok büyük olmasıdır. Bu işlemleri yaparken Matlab programında çalışmalar yapılmıştır.



### Osmanlıca Eğitim Kümesinde Kullanılan Harfler



Şekil 5.7. Eğitim kümesinde kullanılan bazı harfler

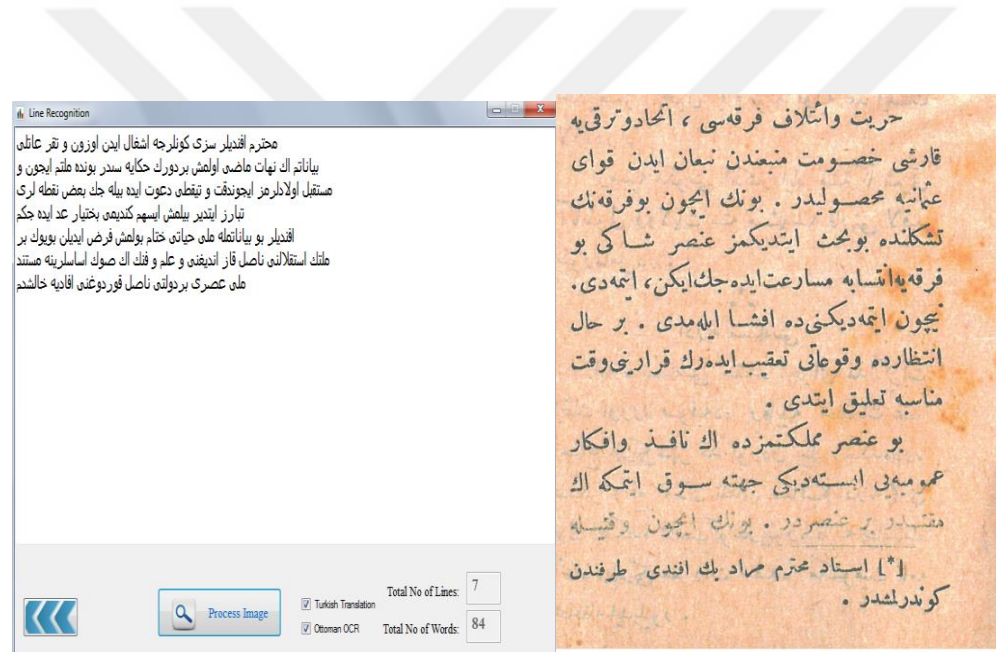


Şekil 5.8. Matlab Ortamında Eğitim

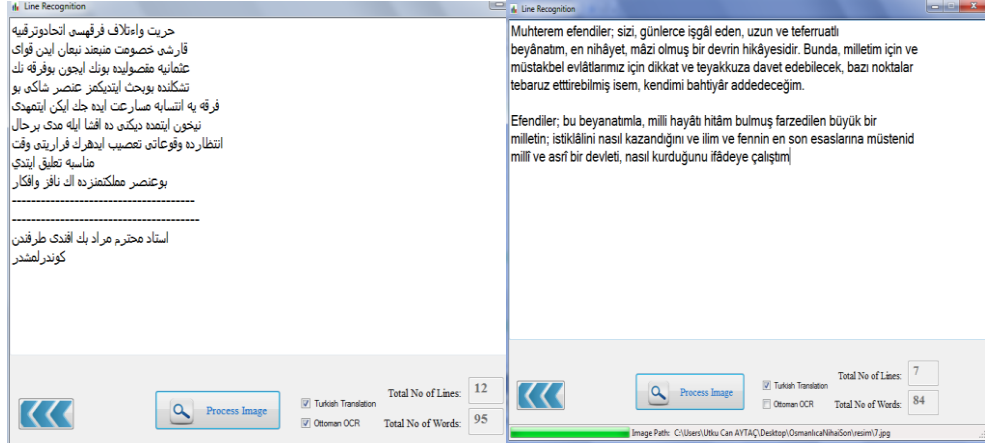
Matlab üzerinde yaptığımız çalışmaları c# diline aktardıktan sonra 3 adet metni OCR işlemine tabi tutulmuştur. Ekran çıktıları aşağıdaki gibidir.

بر هفته قدر ، صامسونده و ۲۵ مایسدن ۱۲ حزیرانه قدر ، حوضده قالدقندن صوکره  
 آماسیه کیتدم . بومدت ظرفنده بوتون مملکتده ، ملی تشکیلات وجوده کتیرلمسی لزومنی  
 تمیمماً بالجمله قوماندانلره ورؤسای مأمورین ملکیه به تبلیغ ایتدم .  
 شایان دقتدرکه، از میرک وبونی تعقیماً مغنیسانک و آیدینک اشغالی واجرا اولتان تجاوز ومظام  
 حقنده هنوز ملت تنورایتمه مش وموجودیت ملیه به ووریلان بوفجیع ضربه به قارشى علناً برکونا  
 تأروشکایت اظهار اولنمامشدی . ملتک ، بو حقسز ضربه قارشوسنده ساکت وحرکتسز قالدی ،  
 البته ملتک لهنده تفسیر اولنه مازدی . اونک ایچون ملتی ایفاظ ایدوب -حرکت کتیرمک لازمدی .  
 بو مقصدله ۲۸ مایس ۳۳۵ نارینخنده ، والیلره ومستقل متصرفلقلره ؛ ارضرومده اون بشنچی  
 قول اردو ، آتقرده یکرمنچی قول اردو و دیاربکرده اون اوچنچی قول اردو قوماندانقلرینه ،  
 قونیه ده اردو مفتشلکنه تمیمماً شو یولده تبلیغانده بولندم :

Şekil 5.9. Orjinal Metin



Şekil 5.10. Ocr işlemine tabi tutulan metin



**Şekil 5.11.** Ocr işlemine tabi tutulan metin2

Yukarıda görülen Ocr işlemine tabi tutulan metinler tutarlı sonuçlar vermektedir. Fakat resim kalitesi farklı olan metinlerde aynı işlem gerçekleştirilememekte, sonuçlar tutarsız olmaktadır. Bunun üzerine kelimelerin binary olarak kayıt edilmesi gerçekleştirilmiştir. Buna göre bölütlenen resimler binary hale getirilecek ve veritabanına Türkçe anlamlarıyla birlikte kayıt edilecektir. Sistem kendi kendini geliştirip, gelen resimlerin sayısal verilerini karşılaştırıp aynı resimden karşılaşırsa onun Türkçesini yazacaktır. Eğer karşılaşmaz ise kullanıcıya bunu kayıt edip etmek istemediğini soracaktır.

Bu işlemleri şablon eşleme yöntemiyle gerçekleştirdik. Bölütleme sonucunda elde edilen resimleri binary olarak veritabanına kayıt ettikten sonra bunların tanınması için bir threshold değeri atanmıştır. Threshold değeri bir resimde nesnelere ayırt edilmesini sağlamaktır. Bunu resmin gri seviye histogramında t(thresholding) değerinden büyük veya küçük olmasına göre piksellerin değerini siyah ya da beyaza çekere gerçekleştirir. Kısacası koyu ton renkleri siyaha açık ton renkleri beyaz renge dönüştürerek karakterlerin tanınmasını sağlamaktır.

#### 5.2.4 Benzerlik matrisi

Çalışmamızın bir sonraki aşaması, harflerin görüntülerini veritabanıyla karşılaştırarak, resimler arası benzerlik puanlarını hesaplamak ve benzer resimleri yani farklı yerlerde geçen aynı kelimeleri ortaya çıkarmaktır.

Osmanlıca'nın yapısal özelliğinden dolayı, yazı içerisinde karakterler kalınlık olarak değişebilmekte, aynı karakterin değişik konumlarda, değişik kalınlıkları olabilmektedir ya da bir karakter farklı formlarda yazılabilmektedir. Bu sebepten

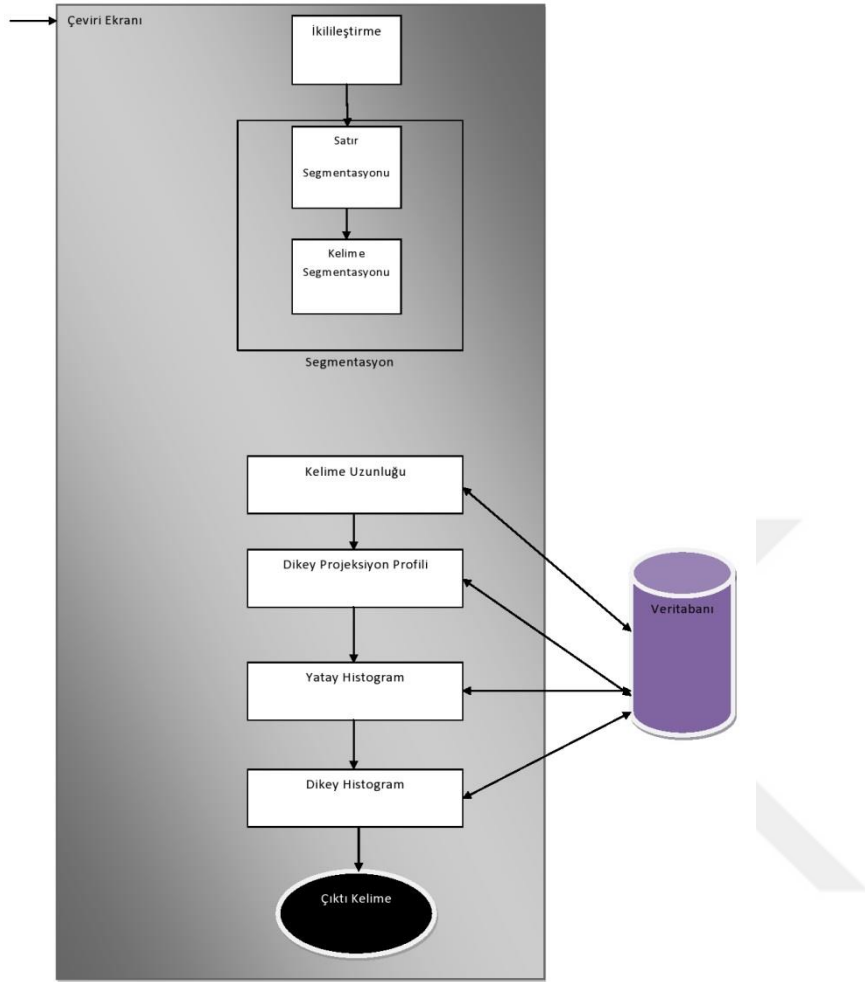
farklı boyutlardaki iki resmin, aynı kelimeyi içerse dahi sayısal görüntü matrisleri farklı olabilir ve bilgisayar bunları farklı resimler olarak algılayacağından yanlış etiketleme yapılmış olur. Sayısal görüntüsünü temsil eden boyuttan bağımsız, özellik vektörlerine ihtiyaç vardır. Bu yüzden resimlerin özellik vektörleri tespit edilerek ikili kombinasyonlar halinde karşılaştırılır ve elde edilen benzerlik/farklılık puanları 'Benzerlik Matrisi' ne alınır. Çalışmamızda her bir resim iki farklı vektör (Dikey Tarama Vektörü, Yatay Tarama Vektörü) ile tanımlanmıştır.

- Yatay Tarama Vektörü (YTV) : Herhangi bir ikili resmin yatay tarama vektörü; resmi oluşturan piksel satırlarındaki toplam çizgi sayılarını içeren vektördür. Aynı satırda yer alan kesintisiz siyah piksel bloklarının her biri bir çizgi olarak nitelendirilir[3].
- Dikey Tarama Vektörü (DTV) : Herhangi bir ikili resmin dikey tarama vektörü; resmi oluşturan piksel sütunlarındaki toplam çizgi sayılarını içeren vektördür. YTV tanımındaki ile benzer şekilde; aynı sütunda yer alan kesintisiz siyah piksel bloklarının her biri bir çizgi olarak nitelendirmektedir[3].

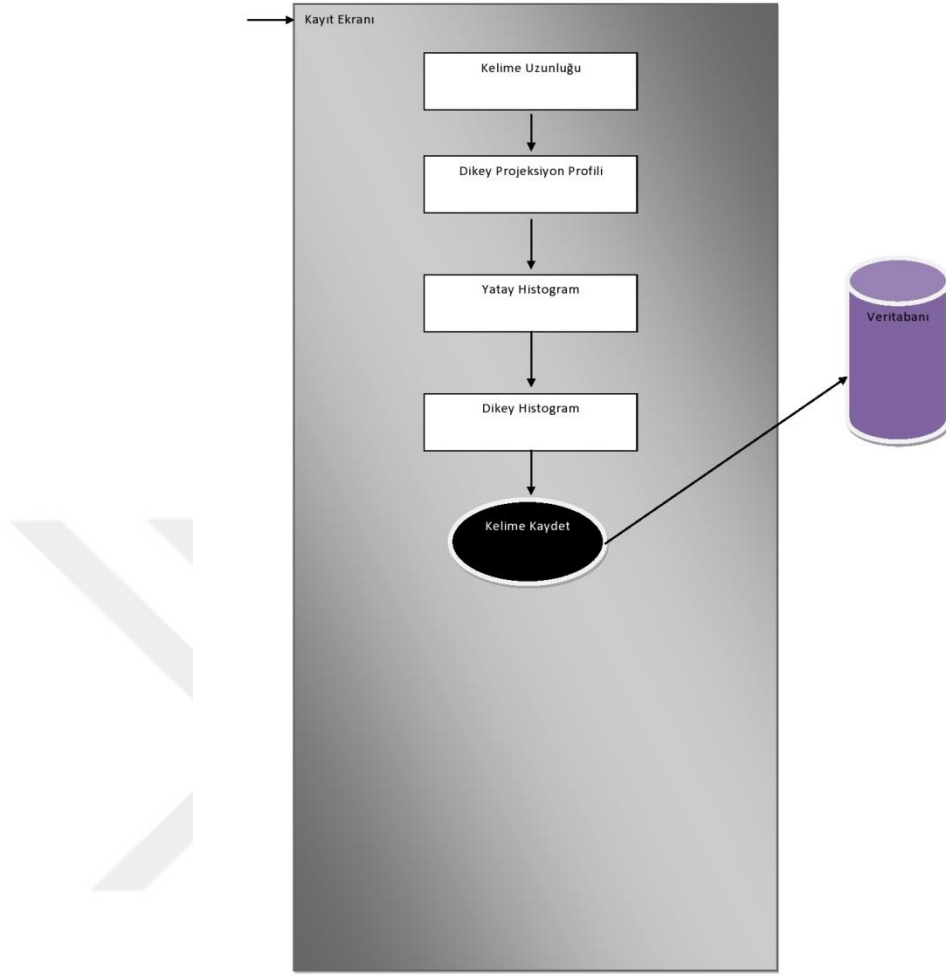
### **5.2.5 Programın algoritması**

Bu yenilikçi yöntem; kullanıcı, tarama, operatör, belge tarama, server, bilgisayar, çeviri ekranı, binarizasyon, satır segmentasyonu, kelime segmentasyonu, kelime uzunluğu, dikey profil izdüşümü, yatay profil izdüşümü, kelime çıkışı, kelime kayıt ekranı, kelime uzunluğu, kelime kayıt, veri tabanı ve internet aşamalarından oluşmaktadır. Çeviri işleminin başlayabilmesi için kullanıcı iki seçeneğe sahiptir. İlk olarak, kullanıcı, sahip olduğu Osmanlıca metni belirli kalite standartlarında tarayarak oluşturulan ilgili bölümünden sisteme yükleme işlemini yapabilir. Operatör, aldığı metni belge tarama işleminden geçirerek gerekli kalite standartlarında metni dijital ortama aktarmaktadır. Elden alınan metinler tarama ve belge tarama işlemlerinden sonra server üzerinde kayıt edilmektedir. Operatörün kullanacağı bilgisayar, internet ağı ile servere bağlı olarak çalışmaktadır. Operatör, dijital ortama aktarılan Osmanlıca metni özel olarak geliştirilen çeviri sistemine yükledikten sonra çeviri ekranı açılır. Çeviri ekranı içerisinde dijital ortama aktarılan belgeler binarizasyon işlemi içerisinde bilgisayar diline yani 1 ve 0'lara dönüştürmektedir. Bilgisayar diline dönüşen metin, satır segmentasyonu ile kaç adet satır barındırdığı hesaplanır. Kelime segmentasyonu ile kaç adet kelime içerdiği

hesaplanacaktır. Segmentasyon işlemi sonrasında hesaplanan kelimelerin kelime uzunluğu ve dikey profil izdüşümü bulunacaktır. Osmanlıca yazılış itibariyle belirli değişkenliklere sahip bir dildir. Matbu ve el yazması eserlerdeki kelimelerin yazılış şekillerinden dolayı veya el yazması eserlerde yazı tipinin kişiye özgü olması kelimelerin farklı uzunluklarda olmasını getirmektedir. Kelime uzunluğu ile bilgisayar dilinde bulunan kelimelerin boyutları hesaplanmakta dikey profil izdüşümü ile kelimelerin anlamları belirlenmektedir. Böylelikle eserin matbu veya el yazması olması kelimelerin sağlıklı biçimde çevrilmesine engel teşkil etmemektedir. Boyutları ve anlamı bulunan kelimeler, daha önceden oluşturulan veritabanı içerisindeki kelime kayıtları ile karşılaştırılacaktır. Bu karşılaştırma işlemi bilgisayar dilinde gerçekleşecektir. Karşılaştırma sonucunda bulunan kelime kayıtları kelime çıkışı ile anlamlı bir biçime dönüştürülerek Osmanlıca metnin Türkçe karşılığı olarak operatöre sunacaktır. Sistem, segmentasyon işlemi sonucunda bulunan kelime ve satırları ayrı ayrı kaydetme olanağı vermektedir. Bu olanak sayesinde operatör, sistemin çevirmiş olduğu metni hızlı ve kolaylıkla test etme imkânına sahip olacaktır. Sistem, karşılaştırma işlemi sonucunda veri tabanı üzerinde kayıtlı kelime bulamadığı zaman operatöre uyarı vererek kelimenin veri tabanına kayıt edilmesini istemektedir. Sıklıkla güncellenecek olan veritabanı bu uyarı sayesinde sürekli olarak genişleyen bir sözlük verisine sahip olacaktır. Veri tabanı üzerinde kayıtlı bulunmayan Osmanlı kelimenin kayıt edilmesi için kelime kayıt ekranı açılacaktır. Bilgisayar dilindeki Osmanlıca kelimenin kelime uzunluğu ve dikey profil iz düşüm değerleri hesaplanarak karşılaştırma parametreleri oluşturulacaktır. Operatör, kelimenin Türkçe karşılığını girdikten sonra kelime, karşılaştırma parametreleri ve Türkçe karşılığı ile veri tabanına kayıt edilecektir. Yukarı anlatılan çeviri ekranı ve kelime kayıt ekranı işlemleri sırasında yapılan aşamalar sistemin alt biriminde hızlıca gerçekleşecek olup operatör bu aşamaları görmeden, karşılaştırma sonucunda kayıdı bulunmayan bir kelime tespit etmediği zaman doğrudan Türkçe çevirisine ulaşacaktır. Sistemin kayıdı bulunmayan bir kelime tespit etmesi durumunda operatörden Osmanlıca kelimenin Türkçe karşılığını girmesini isteyecektir. Operatör, sistemden elde ettiği Türkçe çeviriyi internet üzerinden kullanıcıya aktarılacaktır.



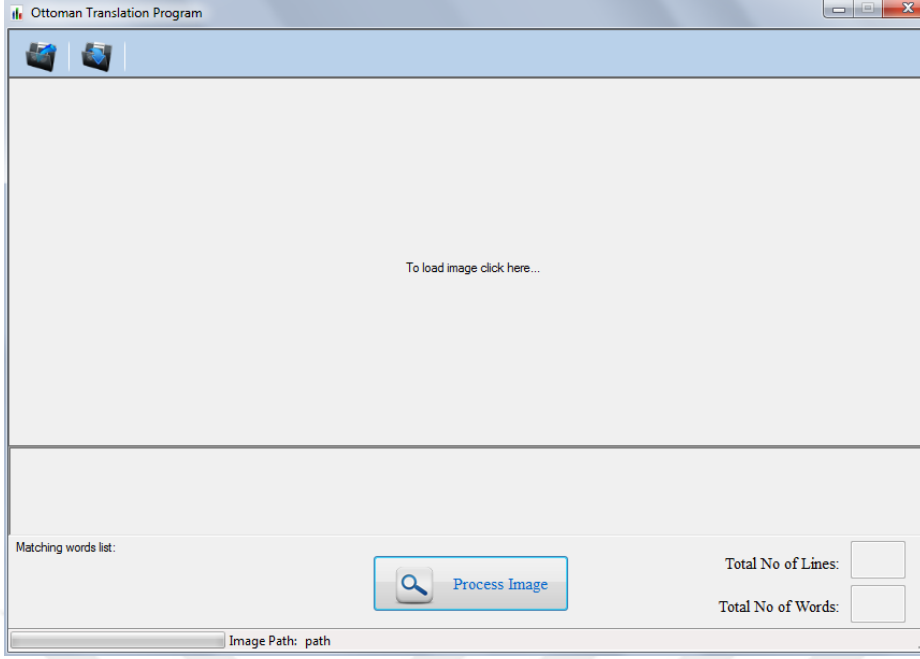
Şekil 5.12. Algoritma1



Şekil 5.13. Algoritma2

### 5.2.6 Programın arayüzü

Program arayüzü, kolay ve anlaşılabilir olmalıdır. Karmaşık arayüzler beraberinde yeni sorunlar getirmektedir. Programımızın test aşamalarında birçok arayüz geliştirdik. Arayüz.Net platformunun en önemli IDE'si (tümleşik geliştirme ortamı) olan Visual Studio 2010 da yapılmıştır. Microsoft Visual Studio, Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework ve Microsoft Silverlight tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve Windows Forms uygulamaları, web siteleri, web uygulamaları ve web servisleri ile birlikte konsol ve grafiksel kullanıcı arayüzü uygulamaları geliştirmek için kullanılır. Programın arayüzü aşağıdaki gibidir.

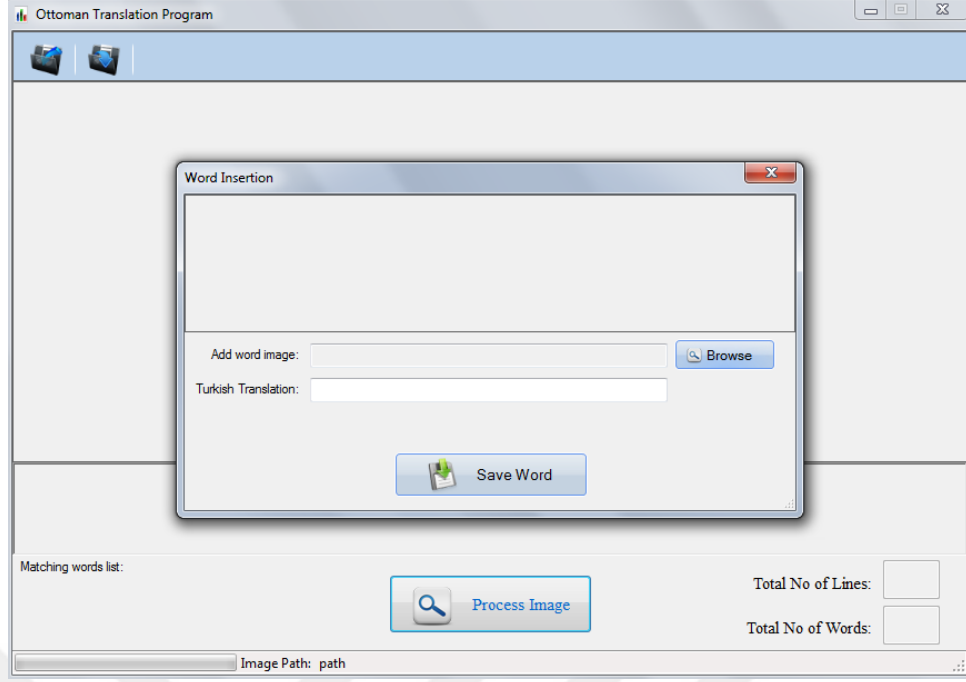


Şekil 5.14. Ana Menü



Şekil 5.15. Kelime Düzeltme





**Şekil 5.16** Kelime Ekleme

Programın arayüzü kullanımı kullanıcı dostu ve kolaydır. Sol üst köşede kelime ekleme ve yanında kelime düzeltme kısmı vardır. Bir resim yüklemek istediğimizde programa fare yardımıyla bir kez tıkladığımızda bize hangi resmi yükleyeceğimizi soracaktır. Yüklenicek resmi seçtikten sonra “Process Image” butonuna bastığımızda resim önce bölütlenecek, ardından veritabanında kayıtlı kelime varsa ekrana gelecektir. Program kullanıcıya kaç adet kayıt edilmemiş resim olduğunu, kaçının kayıt edildiği bilgisini vermektedir. Bir resmi bir kere türkçesi ile kayıt ettiğimizde bir daha kayıt etmemize gerek kalmıyacak, gelen her resimde eğer bulunursa otomatik türkçesi yazılacaktır.



## 6. SONUÇ VE ÖNERİLER

### 6.1 Test Sonuçları

Proje süresince test edilecek metinler Takvim-i Vekayi gazetesinden alınmıştır. Takvim-i Vekayi, Osmanlı İmparatorluğu sınırları dâhilinde 1831'de yayımlanmaya başlanan ilk Osmanlı Türk gazetesidir. Haftalık olarak yayımlanan ve Osmanlı Türkçesi dışında Arapça, Ermenice, Farsça, Fransızca, Rumca baskıları da yayımlanan bir gazeteydi. Resmî ilânlar ve gayri resmî duyurular dışında, iç ve dış gelişmelere ilişkin haberler de basılmaktaydı. Takvim-i Vekayi resmî bir gazete olması dolayısıyla makaleler esas olarak devletin görüşlerini yansıtıyordu. 1860'tan itibaren sadece resmî duyurular ve kabul edilen yasa metinleri yayınlanır oldu. İstanbul Üniversitesi arşivlerinden aldığımız resimler çalışma süresince kullanıldı. Genel olarak üzerinde resim olmayan düz yazı tipleri alındı. Üzerinde resim olan metinler programın çalışmasına engel olacağı için resimler kesilerek ayrı bir resim haline getirildi ve ön çalışmaya alındı. Aşağıda bazı Takvim-i Vekayi gazetesinden alıntılar vardır



Şekil 6.1. Örnek1

ايضاح ايتمك وبو صورتله ماهيتنى اظهار ايله  
توسيع نفوذ ايتمك جهتته كئتمه دى . بالعكس  
برايكى غزته، همده كئدينه منسوب غزته لر  
مؤاخذه نى داعى بندلر يازنجيه «فرقه نك مروج  
افكارى يوقدر» كى تاويل قبول ايتماستسكا .  
فلرله قوى، قطعى مؤاخذه لره مقاومت ايده جك  
درجه ده متين برمسلكه مالك اولما دىغنى افشا  
ايله مكدن چكئتمه دى . بودخى اتحاد جيلر طر-  
فدن اتحاد اولنوب خطرناك سمتره بيجاره  
وطنى سوروكله ين «ميسى سبزم» دن باشقه برشى  
دككلى . واويولده تاقى اولدى .

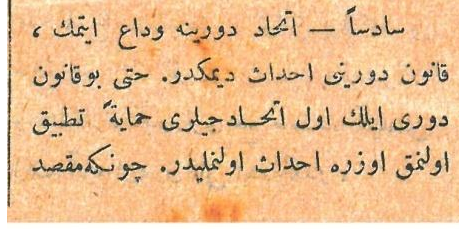
Şekil 6.2. Örnek2

لطفى فكري بك قازان دو برييور . مجاهد  
محترمك اشبو سپاهى زوربالنى بك موسمنده  
اولسه كر كدر . فضلله اوله رق . كئديسنگ شمدى به  
قدر كورنديكى وجهله - يالكىز (اوپورتونىست)  
اولمايوب اورتالغك احوال روجه سنى تدقيق  
وتقديره صالح برسياسى دور انديش اولديغنى

Şekil 6.3. Örnek3

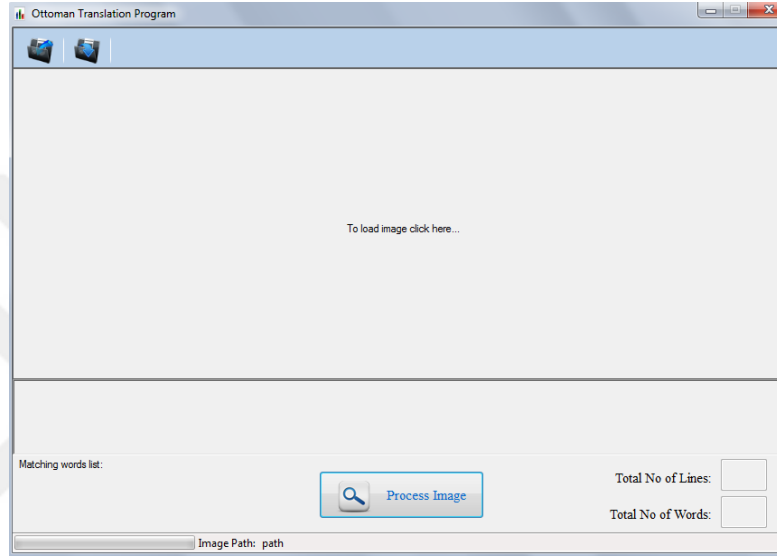
Gazetelerden alınan bu örnek metinler ile sözcükler oluşturulmuş, veritabanının başarımı test edilmiş, kelimeler oluşturulmuştur.

Osmanlıca metinlerin Türkçeye aktarılması konusunda önemli bir başarı elde ettik. Bu sayede geniş bir Osmanlıca sözcük veritabanı ile Osmanlıca metinlerin Türkçeye aktarılması mümkün olabilmektedir. Projenin sonunda geldiğimiz başarıyı ifade etmek gerekirse, elimizde bulunan Osmanlıca metinleri Türkçeye aktarma konusunda geldiğimiz nokta önemlidir. Öncelikle elimizdeki arşivlerde bulunan bir metni seçelim.



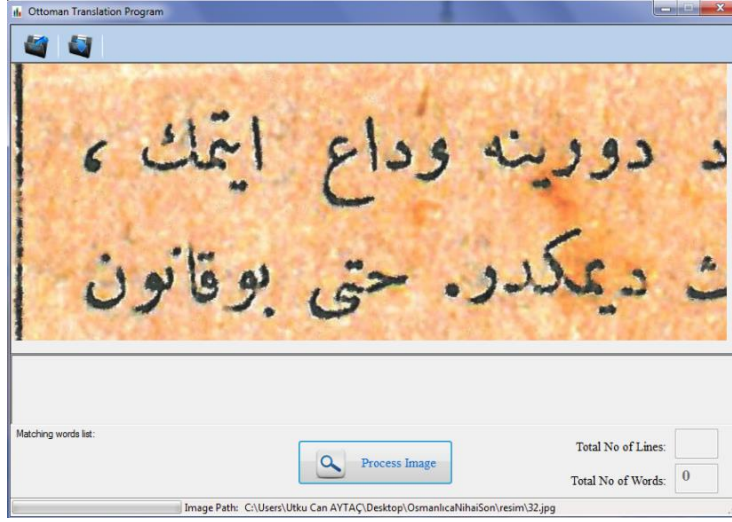
Şekil 6.4. Seçilen Metin

Osmanlıca Türkçe çeviri programımızı açtığımızda karşımıza şu ekran gelecektir;



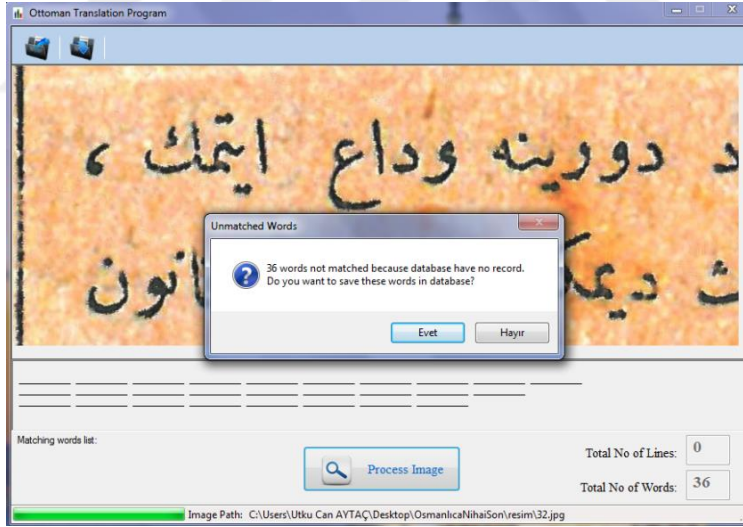
Şekil 6.5. Ana menü

Programda “To load image click here” yazan yere bir kere tıkladığımızda açılan pencerede seçmiş olduğumuz resmi ekleyelim. Karşımıza şu ekran gelecektir;



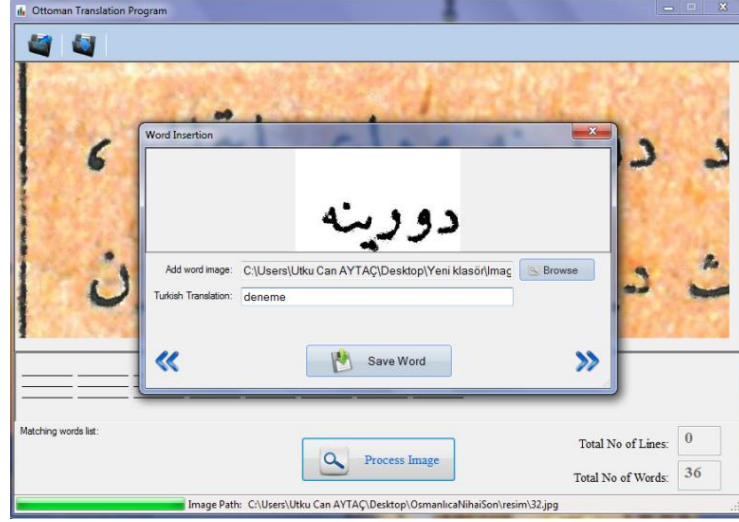
Şekil 6.6. Çeviri Ekranı

Yukarıda gördüğümüz ekranda “Process Image” butonuna tıkladığımızda program resimdeki kelimelerle veritabanındaki eğitilen kelimeleri test edecek, bulduklarının çıktısını vericek, bulamadıklarını da bizden sisteme eklememizi isteyecektir.



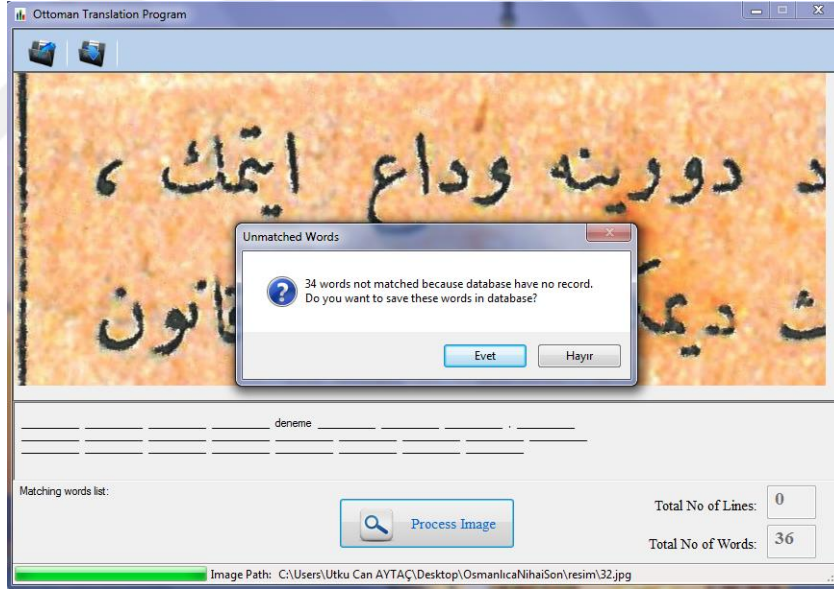
Şekil 6.7, Çeviri Sonucu

36 kelimelik bir metinde 36 tane kelimeyi sistemde bulamadı. Bunun için sisteme bunları eklememiz gerekiyor. Program kullanıcıya sisteme kelimeleri eklemek istermisiniz diye sorduğunda “evet” butonuna tıklıyoruz. Karşımıza şu ekran gelecektir;



Şekil 6.8. Kelime Ekleme

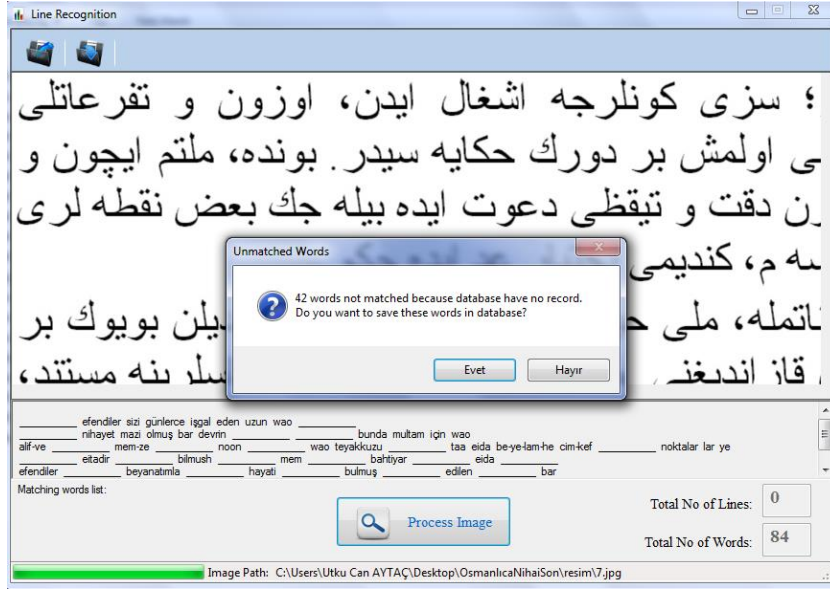
Gelen kelimelerden bir tanesini “deneme” adıyla kayıt ediyoruz ve sistemi tekrar çalıştırdığımızda bize “deneme” kelimesini vermeli, diğer yerlerin boş kalması gerekmektedir.



Şekil 6.9. Bulunamayan Kelimeler

Yukarıda gördüğümüz gibi sistem “deneme” yazısını vermekte, diğer kelimelerin sistemde olmağını, bunları kayıt etmemiz gerektiğini söylemektedir. Böylece bundan sonra herhangi bir metinde osmanlıca “deneme” yazısını gördüğünde her zaman bunu vericektir. Başka bir metnin örneğide aşağıdaki gibidir.





Şekil 6.10. Bulunan Kelimeler

Çizelge 6.1. Satır Segmantasyonu Sonuçları

Karakter No	Satır Sayısı	Segmentasyon		Başarı Oranı
		Yapılan Sayısı	Satır Sayısı	
1	4	4		100%
2	28	26		93%
3	7	7		100%
4	10	10		100%
Toplam	49	47		96%



**Çizelge 6.2.** Kelime Segmentasyonu Sonuçları

Karakter No	Satır Sayısı	Segmentasyon Yapılan Satır Sayısı	Başarı Oranı
1	31	30	96%
2	257	240	93%
3	84	70	83%
4	124	116	93%
Toplam	496	456	91%

Bu sonuçlara baktığımızda satır segmentasyonunda yüzde 96'lık bir başarı oranına sahip olduğu, kelime segmentasyonunda ise yüzde 91'lik başarı oranı elde edildiği görülmektedir.

## 6.2 Tartışma ve Öneriler

Altı yüzyıldan fazla süre boyunca üç kıtada hüküm süren ve tüm dünya tarihinde çok önemli bir yeri olan Osmanlı Devleti'nin arşivleri uluslararası alanda çözüm bekleyen birçok problem için başvurulacak temel kaynaklardır. Aktif olarak kullanılmayan bir dil olması bakımından güncelliğini yitirmeye başlayan Osmanlıca, çoğunlukla konu ile ilgili uzmanlar tarafından subjektif ve belirli alanlarda yapılan çalışmalarda kullanılmaktadır. Bu çalışmanın sahip olduğu veritabanı ve yazılımın bütünsel yapısı, öncelikle mevcut bilgilerin elektronik ortama başarı oranı test edilmiş biçimde aktarılmasını ve sürekli güncellenebilen bir yapıyla kullanılmasını sağlayacaktır. Böylece öncelikle ulusal, daha sonra da uluslararası düzeyde yapılan araştırmalarda, arşivlerde bulunan tarihsel metinlerin bir görsel işleme teknolojisi ile birlikte Latin alfabesini kullanan günümüz dillerine hızlı, objektif ve doğruluk oranı yüksek bir şekilde çevirisi mümkün olacaktır. Önceki yüzyıllarda yazılmış olan metinlerin çok kısa bir sürede günümüz dillerine aktarılması, çalışmaların önündeki en büyük engellerden biri olan zaman problemini ortadan kaldıracaktır. Proje ile hız ve maliyet bakımından Osmanlıca eserlerin yeni bir envanter haline getirilmesinin

kolaylaşacağı, geliştirilecek olan ürünün bir tersine kültür göçüne yardımcı olacağı öngörülmektedir.

Bundan önceki çalışmalarda metin formatındaki belgelerde uygulanan doküman işleme yöntemleri bu tezde resim formatındaki Osmanlıca belgelere uygulanmıştır.

Metin formatlı belgelerde, kelimeler bilgisayar tarafından ASCII kodlarıyla tanınmasına karşılık bu çalışmada resimler bilgisayara tanıtılmış ve her resim veritabanına kaydedilip adeta bir karakter gibi algılatılarak, şablon eşleme yöntemiyle eşleştirilmiştir. Sistem sürekli kendini geliştiren modüler bir yapıdadır.. Böylece savunduğumuz tezin Osmanlıca arşiv belgelerinde çalışabildiği gösterilmiştir. Elde edilen yüksek başarı oranları modelin doğru ve geliştirilebilir olduğunu kanıtlamaktadır. Ülkemizin sahip olduğu Osmanlı arşivlerinin önemli bir kısmının elektronik ortama aktarılmış olmasına rağmen çevirilerin hala elle yapıldığı düşünüldüğünde, sunduğumuz çalışmanın ve ileride geliştirilerek kullanıma uygun hale getirilmesinin önemi ortaya çıkmaktadır.

Bu çalışmanın uygulamasında karşılaştığımız ve ileride geliştirilmesi gerektiğine inandığımız diğer noktalar şunlardır:

- Yapılan çalışmada öncelikle 'karakter tanıma' tekniği seçilmiş fakat uygulamada istenilen başarı oranına ulaşılamadığı görülmüştür. Bunun nedeni olarak; karakter işleminde karşılaşılan en büyük sorun osmanlıca karakterler ile tanınması gereken karakterler arasındaki yazı tipi ve boyutu farkının çok büyük olmasıdır. Bunun dışında resim kalitesi farklı olan metinlerde aynı işlem gerçekleştirilememekte, sonuçlar tutarsız olmaktadır.
- Coğrafya, Anatomi, Matematik vb. gibi şekil ve resim içeren doküman ve gazetelerde bu şekillerin ayrıştırılabilmesi için özel çözümler üretilmelidir.
- Bu tezde matbu dokümanlar üzerinde çalışılmıştır. Karakterlar Takvim'i Vekayi gazetesinden alınmıştır. El yazısı ile yazılmış dokümanlarda, yazı karakterleri yazan kişiye göre değiştiğinden ve satırların çoğunlukla düz bir hat üzerinde olmamasından dolayı satır parçalama ve kelime tanıma daha zor çalışacaktır. Bunun için dile özgü kurallar işletilerek parçalamanın daha sağlıklı yapılması sağlanabilir.
- Mevcut programın diğer metinlere uyumlu hale getirilmesi için daha gelişmiş

normalizasyon, ön işleme çalışmalarının yapılması, Yapay sinir ağları ile kelime tanıma başarısının artırılması önerilmektedir.





## KAYNAKLAR

### Metin Kısmı

- [1]. **Ataer, E. ve Duygulu, P.** (2006). *Retrieval of Ottoman Documents*, Bilkent Üniversitesi, Ankara, Türkiye
- [2]. **Yalnız, Z. İ.** (2008). *Integrated Segmentation and Recognition of Connected Ottoman Script*, Ankara, (doktora tezi), Bilkent Üniversitesi, Türkiye
- [3]. **Eroğlu, Y.** (2007). *Osmanlıca El Yazısı Harfleri Çevrimiçi Tanıma*, Ankara, (yüksek lisans tezi), Gazi Üniversitesi, Türkiye
- [4]. **Adıgüzel, H.** (2013). *Segmentation Based Ottoman Text and Matching Based Kufic Image Analysis*, (yüksek lisans tezi), Bilkent Üniversitesi, Türkiye
- [5]. **Pehlivan, R.** (2014). *Resim Tabanlı Osmanlıca Belgelerde Sınıflandırma*, (yüksek lisans tezi), İstanbul Kültür Üniversitesi, Türkiye
- [6]. **Öztürk, A., Güneş, S. ve Özbay, Y.** (t.y). *Multifont Ottoman Character Recognition*, Selçuk Üniversitesi, Konya, Türkiye
- [7]. **Şaykol, E., Sinop, K. A., Güdükbay, U., Ulusoy, Ö. ve Çetin, E.** (2004). Content-Based Retrieval of Historical Ottoman Documents Stored as Textual Images, *Ieee Transactions on Image Processing*, Vol. 13, No. 3, Sf. 314-323.
- [8]. **Amin, A.** (1997). Offline Arabic Character Recognition: The State Of The Art, *Pattern Recognition*, Vol. 31, No. 5, Sf. 517-530.
- [9]. **Duygulu, P.** (2006), *El Yazısı Tanıma*, Bilişim Ansiklopedisi, Sf. 69-73.
- [10]. **Amin, A.** (1999). Recognition of Printed Arabic Text Based on Global Features and Decision Tree Learning Techniques, *Pattern Recognition*, Vol. 33, Sf. 1309-1313.
- [11]. **Mesleh, A., Sharadqh, A., Jaber T., Odeh, O. ve Hasn, M.** (2012). An Optical Character Recognition, *Contemporary Engineering Sciences*, Vol. 5, 2012, no. 11, Sf. 521 - 529.
- [12]. **Salmani Jelodar, M., Fadaeieslam, M., Mozayani, N. ve Fazeli, M.** (2007). A Persian OCR System Using Morphological Operations, *World Academy of Academy of Science*, Sf. 735-738.
- [13]. **Altıngövde, S. İ., Şaykol, E., Ulusoy, Ö., Güdükbay, U., Çetin, E. ve Göçmen, M.** (t.y). *Osmanlı Arşivleri İçerik-Bazlı Sorgulama (İBS) Sistemi*, Bilkent Üniversitesi, Ankara.
- [14]. **Yanıköğlü, B. ve Kholmatov, A.** (t.y). *Türkçe İçin Geniş Sözcük Dağarcıklı Doküman Tanıma Sistemi*, Sabancı Üniversitesi, İstanbul.
- [15]. **Quotli, F.** (2010). *Resim İçerik filtreleme ve İçerik tespiti*, (yüksek lisans tezi), Gazi Üniversitesi, Ankara.
- [16]. **Aba, F.** (2014). *Görüntü İşleme ve Yapay Sinir Ağları Kullanarak Mineral Tanıma*, (yüksek lisans tezi), Fırat Üniversitesi, Elazığ.
- [17]. **Polat S.** (t.y).Alındığı Tarih:01.02.2015 Başbakanlık Müsteşar Yardımcısı, “Önsöz”, Başbakanlık Osmanlı Arşivi Rehberi, adres:www.devletarsivleri.gov.tr/icerik/236/devlet-arsivleri-genel-mudurlugu-tarihcesi/

- [18]. **Tan, Chew L., Huang, W., Sung, Sam,Y., YU, Z., Xu,Y.** (2003). *Text Retrieval from Document Images Based on Word Shape Analysis*, Applied Intelligence 18, Sf:257–270.
- [19]. **Gharib, T. F., Habib,M. B., Fayed, Z. T.,(t.y).** Arabic Text Classification Using Support Vector Machines, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt.
- [20]. **Doğan S.** (2006). *Türkçe Dokümanlar İçin N-Gram Tabanlı Sınıflandırma: Yazar Tür ve Cinsiyet*, (Yüksek Lisans Tezi), Yıldız Teknik Üniversitesi, İstanbul.
- Url-1**<[http://tr.wikipedia.org/wiki/Osmanlı\\_Türkçesi](http://tr.wikipedia.org/wiki/Osmanlı_Türkçesi)>, alındığı tarih: 12.09.2014.
- Url-2** <[http://tr.wikipedia.org/wiki/Osmanlı\\_Arşivi](http://tr.wikipedia.org/wiki/Osmanlı_Arşivi)>, alındığı tarih: 14.09.2014.
- Url-3**<[http://tr.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://tr.wikipedia.org/wiki/Microsoft_Visual_Studio)>, alındığı tarih:28.12.2014.
- Url-4** <[http://tr.wikipedia.org/wiki/Microsoft\\_Access](http://tr.wikipedia.org/wiki/Microsoft_Access)>, alındığı tarih: 29.12.2014.

## Şekiller

- [25]. **Mulgaonkar P. G.,Chen C., DeCurtins J. L.** (1994) Word Recognitionin a Segmentation-FreeApproachto OCR.
- [26]. [http://www.turkedebiyati.org/osmanlica\\_arap\\_alfabesi.html](http://www.turkedebiyati.org/osmanlica_arap_alfabesi.html)
- [27].<http://www.delinetciler.org/diger-diller/116035-osmanlica-harfler-ve-gunumuz-turkcesiyle-okunusu.html>
- [28]. **McAndrew, A.,** (2004), Digitalimageprocessing with MATLAB, School of Computer Science and Mathematics, Victoria University, Thomson Course Technology, Melbourne, Victoria, Australia
- [29]. **Özbek, U.,** (2005), MATLAB’da image processing toolbox kullanarak temel resim işleme ve gürültü ayıklama, Yüksek Lisans Semineri, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya
- [30]. **Demir, Ö.,** (2006), MATLAB gereçleri ile görüntü işleme uygulamaları, Yüksek Lisans Tezi, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul
- [31].<http://dspace.trakya.edu.tr/jspui/bitstream/1/629/1/MURAT%20%C5%9EEKER%C4%B0.pdf>
- [32].<http://dspace.trakya.edu.tr/jspui/bitstream/1/629/1/MURAT%20%C5%9EEKER%C4%B0.pdf>

## **EKLER**

### **EK A: Test İin Kullanılan Osmanlıca Gazeteler**







محترم افنديلر؛ سزى كونلرجه اشغال ايدن، اوزون و تفرعاتلى بياناتم، اك نهايت ماضى اولمش بر دورك حكايه سيدر. بونده، ملتيم ايچون و مستقبل اولادلرمرز ايچون دقت و تيقضى دعوت ايده بيله جك بعض نقطه لرى تبارز ايتديره بيلمش ايسه م، كنديمي بختيار عد ايده جكم.

افنديلر؛ بو بياناتمله، ملي حياتى ختام بولمش فرض ايديلن بويوك بر ملتك؛ استقلالنى ناصل قازانديغنى و علم و فنك اك صوك اساسلرينه مستند، ملي و عصرى بر دولتى ناصل قوردوغنى افاده يه چالشم.

بر هفته قدر، صامسونده و ۲۵ مایسدن ۱۲ حزیرانه قدر، حوضه ده قالدقندن صوكره آماسیه یه كیتدم . بومدت ظرفنده بوتون مملكتده ، ملي تشكيلات وجوده كتیرلمسى لزومى تعمیماً بالجمله قوماندانلره ورؤسای مأمورین ملكیه یه تبلیغ ایتدم .

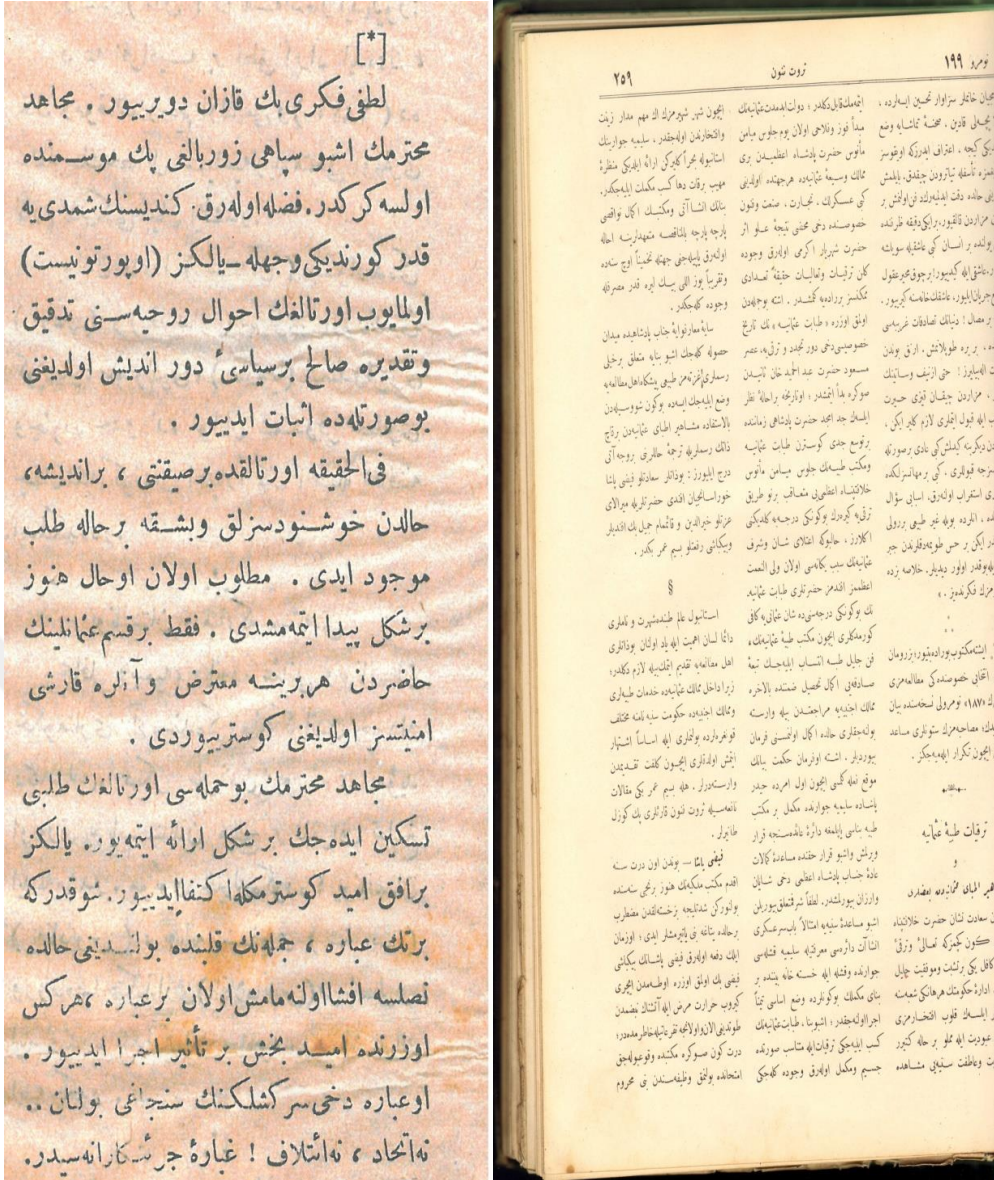
شایان دقتدرکه، از میرك وبونى تعقیماً مغنیسانك وآیدینك اشغالی واجرا اولنان تجاوز ومظالم حقننده هنوز مات تنورایتمه مش وموجودیت ملییه ووریلان بوفجیع ضربه یه قارشى علناً برکونا تأروشکایت اظهار اولمامشدى . ملتک ، بوحقسنز ضربه قارشوسنده ساکت وحرکتسنز قالدی ، البته ملتک لهنده تفسیر اولنه مازدی . اونك ایچون ملتى ابقاظ ادوب حرکته كتیرمك لازمدى . بو مقصدله ۲۸ مایس ۳۳۵ نارینخنده ، والیره ومستقل متصرفلقلره ؛ ارضرومده اون بشنجهی قول اردو ، آنقرده یکرمنجهی قول اردو و دیاربکرده اون اوچنجهی قول اردو قوماندانانقلرینه ، قونیه ده اردو مفتشلکنه تعمیماً شویولده تبلیغانده بولندم :

Şekil A.1: Osmanlıca Matbu Gazete Örneği



Şekil A.2: Takvim-i Vekayi Gazetesi Örnekleri





Şekil A.3: Takvim-i Vekayi Gazetesi Örnekleri2



## ÖZGEÇMİŞ

**Ad-Soyad** : Utku Can AYTAÇ

**Doğum Tarihi ve Yeri:** 31.08.1987, Bursa

**E-posta** : [utku.c.aytac@gmail.com](mailto:utku.c.aytac@gmail.com)

## ÖĞRENİM DURUMU:

- **Lisans** : 2005, Kadir Has Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yüksek Lisans** : 2011, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Mühendislik Yönetimi

## MESLEKİ DENEYİM VE ÖDÜLLER

06.2012- : İnovasyon Proje

Proje Uzmanı

06.2009 : INVENOA Yazılım&İletişim A.Ş Yazılım Bölümü

Yaz Stajı

08.2008 : Oyak Renault Otomobil Fabrikaları A.Ş Bilgi İşlem Bölümü

Yaz Stajı