

OSMANLICA ELYAZISI HARFLERİ ÇEVİRİMİÇİ TANIMA

Yılmaz EROĞLU

**YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

**TEMMUZ 2007
ANKARA**

Yılmaz EROĞLU tarafından hazırlanan OSMANLICA ELYAZISI HARFLERİ
ÇEVİRİMİÇİ TANIMA adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu
onaylarım.

Yrd. Doç.Dr. Mustafa BURUNKAYA
Tez Yöneticisi
(2.Danışman)

Prof.Dr. Muammer NALBANT
Tez Yöneticisi
(1.Danışman)

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Elektronik ve Bilgisayar
Eğitimi Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan: : Prof.Dr. Ömer Faruk BAY

Üye : Prof.Dr. Muammer NALBANT

Üye : Yrd.Doç.Dr. Halil İbrahim BÜLBÜL

Üye : Yrd.Doç.Dr. Nursal ARICI

Üye : Yrd.Doç.Dr. Mustafa BURUNKAYA

Tarih : 05/07/2007

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Yılmaz EROĞLU

OSMANLICA ELYAZISI HARFLERİ ÇEVİRİMİÇİ TANIMA**(Yüksek Lisans Tezi)****Yılmaz EROĞLU****GAZİ ÜNİVERSİTESİ****BİLİŞİM ENSTİTÜSÜ****Temmuz 2007****ÖZET**

Bu tezde, hazırlanan arayüzdeki sınırlandırılmış alana fare ile çizilen Osmanlıca harflerin tanınması ve Osmanlıca metin editörüne aktarılması sağlanmıştır. Arayüzün tasarlanması için Delphi 6.0 görsel programlama dili kullanılmıştır. Tanınan harfler metin editörüne Unicode karakter sisteminde aktarılmıştır. Hazırlanan programda yapısal yaklaşıma ait yön kodlaması yöntemi kullanılmıştır. İlk olarak çizilen şekle ait yön bilgisi, başlangıç ve bitiş noktaları arasındaki eğim, noktaların harfe ve birbirine olan uzaklıkları gibi karakteristik özellikleri çıkarılmıştır. Daha sonra çıkarılan bu öznitelikler harf tablosundaki her harfe ait özelliklerle karşılaştırılmış ve öznitelikler hangisine uyuyorsa çizilen şekil o karakter olarak tanınmıştır. Tanımlanan özellikler o harfe ait temel karakteristikler olduğu için genel kurallara göre çizilen her harf tanınmıştır. Harflerin genel doğru tanınma ortalaması %92'dir. Bu çalışmada öznitelik tabloları Nesih Hattına göre hazırlandığı için bu hat üzerinde doğru sonuçlara ulaşılmaktadır. Fakat öznitelik tabloları diğer hatlara göre oluşturulduğunda diğer hatların da tanınması sağlanabilmektedir. Tanınan karakter, Osmanlıca olarak metin editörüne aktarılarak düzenlenebilecek hale getirilmiştir.

Bilim Kodu : 704.3.013

Anahtar Kelimeler : Çevrimiçi karakter tanıma, Osmanlıca elyazısı, yön
temelli öznitelikler, yön kodlama, el yazısı tanıma

Sayfa Adedi : 65

Tez Yöneticisi : Prof.Dr.Muammer NALBANT, Yrd.Doç.Dr. Mustafa
BURUNKAYA

ONLINE HANDWRITTEN OTTOMAN CHARACTER RECOGNITION
(M.Sc. Thesis)

Yılmaz EROĞLU

GAZİ UNIVERSITY
INFORMATICS INSTITUTE
JULY 2007

ABSTRACT

The subject of this thesis is online recognition of Ottoman letters that drawn on program interface by a mouse and transferring to Ottoman text editor. The program was prepared with Delphi 6.0 programming language. Recognized letters were transferred to Ottoman text editor in Unicode character set. Direction coding method related to structural approach has been used in the program. First of all, the characteristic features of drawing such as; directional features, gradient of starting and ending points, and the distance of points to letters and each other are calculated. Then, this calculated features are compared with the letters' features in the letter table. The drawing is defined with the most similar letter. As the main features are the defined features of that letter, all the letters which are drawn according to general rules are recognized. General recognition rate of Ottoman letters is 92%. In this work, as the feature tables are prepared according to Naskh fonts, the correct results could be found on this font. When the feature tables are occurred according to other fonts we could supply the recognition of the other fonts. Finally, the recognized letters are transferred to Ottoman text editor in an editable form.

Science Code : 704.3.013

**Key Words : Online character recognition, Ottoman handwriting characters,
direction based features, direction coding, handwriting
recognition**

Page Number: 65

**Adviser : Prof.Dr.Muammer NALBANT, Assist. Prof.Dr. Mustafa
BURUNKAYA**

TEŐEKKÜR

Çalıőmalarım süresince çok deęerli yardım ve katkılarıyla beni yönlendiren hocam Prof.Dr. Muammer NALBANT'a ve yine tecrübelerinden faydalandığım hocam Yrd.Doç.Dr. Mustafa BURUNKAYA'ya, ayrıca kıymetli bir ağabeyim Vedat AYTEK'e ve manevi destekleriyle her zaman yanımda olan çok deęerli eşime teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ	xi
ŞEKİLLERİN LİSTESİ	xii
RESİMLERİN LİSTESİ	xiii
KISALTMALAR	xiv
1. GİRİŞ	1
2. OSMANLICA’NIN GENEL YAPISI VE LATİN HARFLERİNDEN FARKI	3
3. OSMANLICA VE ARAPÇA KARAKTER TANIMA KONUSUNDA YAPILAN ÇALIŞMALAR	8
3.1. Osmanlıca Karakterlerin Bilgisayar Destekli Tanınması	8
3.2. Osmanlıca Karakter Tanıma	9
3.3. Osmanlıca Metnin Bölütlenmesi, Öznitelik Çıkarımı ve Tanınması	9
3.4. Arapça Elyazısı Karakter Tanıma Sistemi	10
3.5. Yapay Sinir Ağları Yardımıyla Basılı Arapça Karakterlerin Tanınması	11
4. KARAKTER TANIMA SİSTEMİ	12
4.1. Ön işlemler (Preprocessing)	13
4.1.1. Eşikleme (Binarization)	14
4.1.2. Gürültü oranının düşürülmesi	14
4.1.3. Genişletme (Dilation)	15
4.1.4. İnceltme (Thinning)	16
4.1.5. Onarım	17
4.1.6. Normalizasyon	17
4.2. Karakter Ayırma (Segmentation)	17
4.3. Özellik Çıkarımı (Feature Extraction)	18
4.4. Tanıma (Recognition)	19

	Sayfa
4.4.1. Matris eşleme (Matrix matching).....	19
4.4.2. İçerik analizi (Feature analysis).....	19
4.4.3. Özgün tanımlama (Self-assertion).....	20
5. EL YAZISI TANIMA.....	21
6. OSMANLICA ELYAZISI HARFLERİ ÇEVİRİMİÇİ TANIMA.....	24
6.1. Önışlemler.....	25
6.2. Öznitelik Çıkarımı.....	25
6.2.1. Yön bilgisinin hesaplanması.....	25
6.3. Karakter Tanıma (Recognition).....	33
6.4. Harflerin Osmanlıca Metin Editörüne Aktarılması.....	37
6.5. Program Düğmelerinin Tanıtımı.....	38
7. SONUÇLAR.....	40
KAYNAKLAR.....	43
EKLER.....	45
EK-1.....	46
EK-2.....	63
ÖZGEÇMİŞ.....	65

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Osmanlıca hat örnekleri	7
Çizelge 6.1. Yön bilgileri hesaplama kriterleri	28
Çizelge 6.2. Osmanlıca elyazısı harflerin tanınma oranları	34
Çizelge 7.1. Osmanlıca elyazısı harflerin tanınma oran aralıkları	41

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 4.1. Karakter tanıma sistemlerinin temel aşamaları.....	13
Şekil 4.2. Bir genişletme işlemi maskesi	15
Şekil 4.3. Zhang-Suen, Stentiford inceltme algoritmaları.....	16
Şekil 6.1. Osmanlıca elyazısı karakterleri çevrimiçi tanıma sisteminin aşamaları ...	24
Şekil 6.2. Osmanlıca karakterlere ait tanımlanan yönler	26
Şekil 6.3. Harf tanınma oranları ile ilgili dağılım grafiği ve kutu grafiği.....	35

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 2.1. Osmanlıca harfler	3
Resim 2.2. Osmanlıca harflerin başta, ortada ve sondaki halleri	4
Resim 3.1. Eğitim setinde bulunan 3 değişik fonta ait matbu harflerden yanlış sınıflandırılanlar	9
Resim 3.2. Eğitim setinde bulunmayan farklı bir fonta ait matbu harflerden yanlış sınıflandırılanlar	9
Resim 3.3. Arapça bir karakterin yapay sinir ağı giriş katmanı için genel gösterimi 1	
Resim 4.1. Bir resme ait ikili görüntü	14
Resim 4.2. Osmanlıca bir metnin gürültülü ve gürültüsü düşürülmüş hali	15
Resim 4.3. İnceltme algoritmaları uygulanmış bir görüntü	16
Resim 4.4. Muhammed kelimesinin bölütlenmiş hali	18
Resim 6.1. Elif harfine ait yön bilgisini gösteren ekran görüntüsü	26
Resim 6.2. Be harfine ait yön kodlarının gösterimi	27
Resim 6.3. Ha harfine ait yön bilgilerinden biri	29
Resim 6.4. Ha harfine ait yön bilgilerinden bir başkası	29
Resim 6.5. Mim harfinin tanınması	35
Resim 6.6. Çim harfinin tanınması	36
Resim 6.7. Dat harfinin tanınması	36
Resim 6.8. Şin harfinin Osmanlıca editörüne aktarılması	37
Resim 6.9. La İlahe İllallah cümlesinin editöre aktarılmış hali	38
Resim 6.10. Harf tanımlama uyarısı	39

KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
OCR	Optik karakter tanıma
PDA	Avuçiçi bilgisayar
YSA	Yapay sinir ağı

1. GİRİŞ

İnsanođlu yıllardır kađıt ortamındaki el yazısı veya matbaa ıktısı metinleri ile eř zamanlı olarak bilgisayara girilen el yazılarını tanımak ve bilgisayar ortamında dzenlenebilecek hale getirmek iin uđrař vermektedir. Bu tanıma iřlemleri ierisinde en zor olan ve arařtırmacıları da yıllardır meřgul eden el yazısının tanınmasıdır. Bu alanda birok alıřma yapılmasına rađmen optimum bir zm retilmiř deđildir. El yazısı karakterlerin kullanıcıların yazım tarzlarına gre farklılık gstermesi, harflerin birleřmesi ve matbaa ıktılarına gre tam bir standardının olmayıřı el yazılarının daha zor tanınmasının temel sebepleridir. Matbaa ıktıları zerinde yapılan alıřmalar ise daha tatminkar gzmektedir. Bunlardan Latin harflerinin matbaa ıktıları zerinde elde edilen sonular gnmzde yeterli dzeye ulařmıřtır. Fakat ince, Japonca ve Osmanlıca karakterler zerinde sorun hala devam etmektedir. zellikle Osmanlıca ve Arapa karakterlerin tanıtılması dilin yapısı ve yazım řekli gz nnde tutulduđunda olduka zorlařmaktadır. Hatta matbaada hazırlanmıř bir Osmanlıca metnin tanınması el yazısı ile yazılmıř bir Latince metinden daha zor olabilmektedir [1].

Osmanlıca matbuat metinleri zerinde bazı alıřmalar olsa da Osmanlıca el yazısı metinler zerinde řu ana kadar yapılmıř tatminkar bir alıřma bulunmamaktadır [2]. Hal byle olunca Osmanlıca el yazısı karakterlerin bilgisayar ortamında iřlenebilecek hale getirilmesi problemi ok byk nem arz etmektedir.

Bu alıřmanın amacı hazırlanan arayzndeki sınırlandırılmıř alana fare ile izilen Osmanlıca harflerin evrimii tanınması ve Osmanlıca metin editrne aktarılmasıdır.

Arayzn tasarlanması iin Delphi 6.0 grsel programlama dili kullanılmıřtır. Tanınan harfler metin editrne Unicode karakter sisteminde aktarılmıřtır. Hazırlanan programda yapısal yaklařıma ait yn kodlaması yntemi kullanılmıřtır.

Hazırlanan program kişisel bilgisayarlarda (PC), Elektronik tahtalarda, Tablet PC lerde kullanılabilir durumdadır. Cep telefonları ve avuçiçi bilgisayarlarda (PDA) ise gerekli uyarlamalar yapıldığında kullanılabilir.

Bu tez 7 ana bölümden oluşmaktadır.

1. bölümde tezin amacı, önemi ve kapsamı anlatılmıştır.
2. bölümde Osmanlıca'nın temel özelliklerinden bahsedilmiş ve Osmanlıca ile Latin harfleri arasındaki farklar üzerinde durulmuştur.
3. bölümde Osmanlıca ve Arapça karakter tanıma konusundaki literatür taramaları değerlendirilmiş ve öne çıkan bazı çalışmalar üzerinde durulmuştur.
4. bölümde genel olarak karakter tanıma konusunun sistematığı incelenmiş yöntem ve tekniklere değinilmiştir.
5. bölümde karakter tanımanın bir alt alanı olan el yazısı tanıma üzerinde durulmuştur.
6. bölümde ise tezin konusunu oluşturan Osmanlıca el yazısı karakterleri çevrimiçi tanıma sistemi gerçekleştirilmiş, sistemin tüm adımları anlatılmış, bazı önemli program kodları açıklamalarıyla birlikte verilmiştir.
7. bölümde çalışmanın sonuçları değerlendirilmiş ve bazı önerilerde bulunulmuştur.

2. OSMANLICA'NIN GENEL YAPISI VE LATİN HARFLERİNDEN FARKI

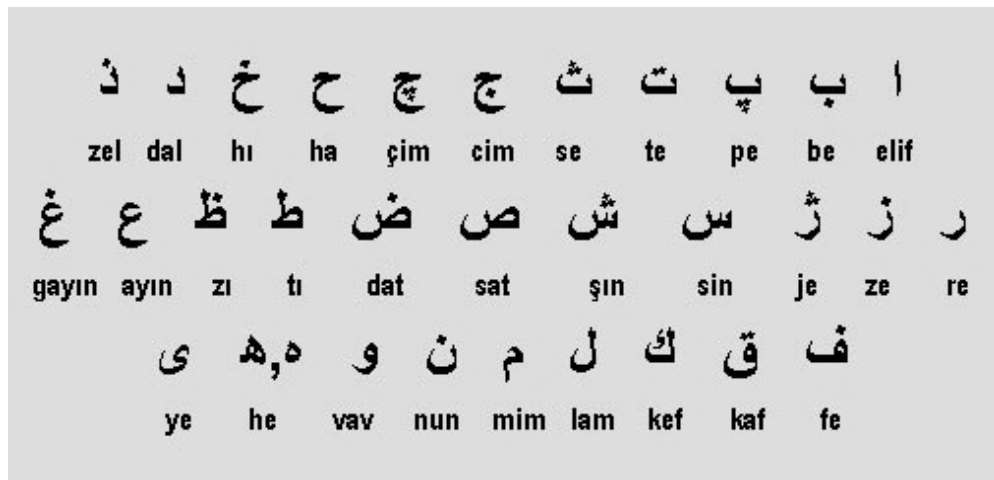
Bu bölümde Osmanlıca'nın temel özelliklerinden bahsedilmiş ve Osmanlıca ile Latin harfleri arasındaki farklar üzerinde durulmuştur.

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 (ژ) 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 nokta eklenerek nef (sağır kef, kâf-ı nunî), lam ile eliften lamelif, hemze ile h harfinin ünlü şekli olan hâ-i resmiye 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 noktalama işaretleri, boşluklar ve özel semboller kullanılır.



Resim 2.1. Osmanlıca harfler

- 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.

İsimleri	Harfler	Sonda (sağdan bitişik)	Ortada (her iki taraftan bitişik)	Başta (soldan bitişik)
elif	ا	ا	-	-
hemze	ء	ء	ء	ء
be	ب	ب	ب	ب
pe	پ	پ	پ	پ
te	ت	ت	ت	ت
se	ث	ث	ث	ث
cim	ج	ج	ج	ج
çim	چ	چ	چ	چ
ha	ح	ح	ح	ح
hu	ه	ه	ه	ه
dal	د	د	-	-
zel	ذ	ذ	-	-
re	ر	ر	-	-
ze	ز	ز	-	-
je	ج	ج	-	-
sin	س	س	س	س
şın	ش	ش	ش	ش
sat	ص	ص	ص	ص
dat	ط	ط	ط	ط
tı	ظ	ظ	ظ	ظ
zı	ظ	ظ	ظ	ظ
ayın	ع	ع	ع	ع
gayın	غ	غ	غ	غ
fe	ف	ف	ف	ف
kaf	ك	ك	ك	ك
kef	ك	ك	ك	ك
gef	ك	ك	ك	ك
nef, sağır kef	ك	ك	ك	ك
lam	ل	ل	ل	ل
mim	م	م	م	م
nun	ن	ن	ن	ن
vav	و	و	-	-
he	ه	ه	ه	ه
lame lif	ي	ي	-	-
ye	ي	ي	ي	ي

Resim 2.2. Osmanlıca harflerin başta, ortada ve sondaki halleri

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 (mesela ا, آ, إ, ؤ, !).

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. Bir çok 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 ا, آ, إ, ؤ, !) 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 bir çok fonta ve yazı biçimine sahiptir. Aynı fonttaki (hat) harfler farklı büyüklüklere sahiptir. Bundan dolayı sabit genişliğe dayalı bölümlenme uygulanamaz.

HAT

Belli kurallara göre güzel yazı yazma sanatına hüsn-i hat (Arapça'da "hat" yazı, "hüsn" güzel) ya da kısaca hat, bu işle uğraşanlara da hattat denir. Yüzyıllar boyunca Arap yazısı özellikle İranlı ve Türk sanatkarlarca geliştirilmiştir. Kullanılan kaleme, harflerin büyüklüğü ve şekline göre ya da iki yazının karıştırılmasıyla çeşitli yazı şekilleri ortaya çıkmıştır. Başlıca yazılar arasında kûfi, muhakkak, rêyhanî, sülüs, nesih, tevkî, rik'a, talîk, dîvanî, sayılabilir. Çizelge 2.1'de bazı hat örnekleri verilmiştir.

Çizelge 2.1. Osmanlıca hat örnekleri [1]

لا إله إلا الله محمد رسول الله
RİK'Â
لا إله إلا الله محمد رسول الله
TA'LİK
لا إله إلا الله محمد رسول الله
DİVANİ
لا إله إلا الله محمد رسول الله
CELİ DİVANİ
لا إله إلا الله محمد رسول الله
REYHANİ
لا إله إلا الله محمد رسول الله
SÜLÜS
لا إله إلا الله محمد رسول الله
KUFİ
لا إله إلا الله محمد رسول الله
NESİH
لا إله إلا الله محمد رسول الله
BİLGİSAYAR ÇIKTISI NESİH

3. OSMANLICA VE ARAPÇA KARAKTER TANIMA KONUSUNDA YAPILAN ÇALIŞMALAR

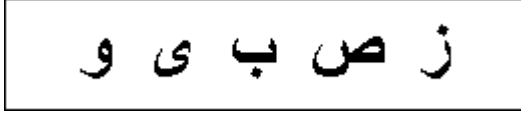
Literatür taramasında Arapça veya Farsça karakter tanıma konusunda oldukça fazla çalışmaya rastlanmıştır. Fakat Osmanlıca karakterler konusunda fazla çalışmaya rastlanmamıştır. Bu çalışmaların ortak özelliği genelde matbu metinler üzerinde çalışılmış olmasıdır. El yazması metinler üzerinde çok az çalışma vardır. Karakter tanıma konusunda ekseriyetle Saklı Markov Modeli , Farklı yapay sinir ağları modelleri (İleri ve Geri Beslemeli Çok Katmanlı Algılayıcı, Adaptif Rezonans Teorisi Ağı, LVQ Ağları, Kohonen Ağları vb.) ve Momentler yardımıyla Optik karakter tanıma yöntemleri kullanılmıştır. Matbu metinler üzerinde yapılan bazı çalışmalarda mesafe kat edilmesine rağmen henüz tatminkar bir sonuç elde edilememiştir.

3.1. Osmanlıca Karakterlerin Bilgisayar Destekli Tanınması

Bu tez çalışmasında üç değişik fonttan 28 değişik Osmanlıca matbaa harfini tanıyan sistemin yapay sinir ağını yardımıyla gerçekleştirilmesi amaçlanmıştır [2].

İlk olarak giriş verileri ile yapay sinir ağı eğitilmiş daha sonra sistem test edilmiştir. Bu iki adımın her birinde de verilerin yapay sinir ağının işleyeceği şekle getirilmesi ve ağın çıkış katmanından elde edilen sonuçların sistem kullanıcısının (insan) doğasına uygun bilgilere çevrilmesi gerekmektedir. Bunun için sırasıyla sistem ön işleme (preprocessing) ve son işleme (postprocessing) aşamalarından geçirilmiştir.

Eğitim setinde bulunan üç değişik Osmanlıca fonttan 28'er adet olmak üzere toplam 84 harf için normalize edilmiş değerler bulunmaktadır. Sonuç olarak bu üç değişik fonta ait matbu karakterlerin tanınmasında %95 düzeyinde tanıma oranlarına ulaşılırken, eğitim setinde kullanılmayan tamamen farklı bir fonttaki Osmanlıca harflerin tanınması %60 düzeyinde sağlanabilmiştir. Yanlış sınıflandırılan harfler Resim 3.1 ve Resim 3.2'de verilmiştir.



Resim 3.1. Eğitim setinde bulunan 3 değişik fonta ait matbu harflerden yanlış sınıflandırılanlar



Resim 3.2. Eğitim setinde bulunmayan farklı bir fonta ait matbu harflerden yanlış sınıflandırılanlar

3.2. Osmanlıca Karakter Tanıma

Özçilingir, bu çalışmada Osmanlıca el yazısı için bir metin tanıyıcı geliştirmiştir [3].

İkili hale getirilen Osmanlıca yazısı, inceltilmiş ve kalem vurgularına ayrıştırılmıştır. Bunlar kalemin bir hareketini yaklaşık olarak bildirmektedir. Bu kalem vurguları geometrik ve topolojik özelliklerine göre sınıflandırılır ve vurgu kütüphanesindeki en yakın bilinen vurgu ile eşleştirilerek onun numarasını alır.

Sonuçta numaralanmış bu vurgu zinciri çeşitli adımlarda istatistik ve kural temelli metotları uygulayarak alternatif karakter zincirlerine dönüştürülmüştür. Osmanlıca karakterler belirli bir tarza bağlı olarak yazıldığı sürece elde edilen sonuçların geçerli olduğu görülmüştür.

3.3. Osmanlıca Metnin Bölütlenmesi, Öznitelik Çıkarımı ve Tanınması

Atıcı, bu tez çalışmasında, C++ programlama dilini kullanarak, PC tabanlı bir donanım üzerinde Osmanlıca metnin bölütlenmesi, öznitelik çıkarımı ve tanınması amacıyla bir optik harf tanıma sistemi geliştirmiştir [4].

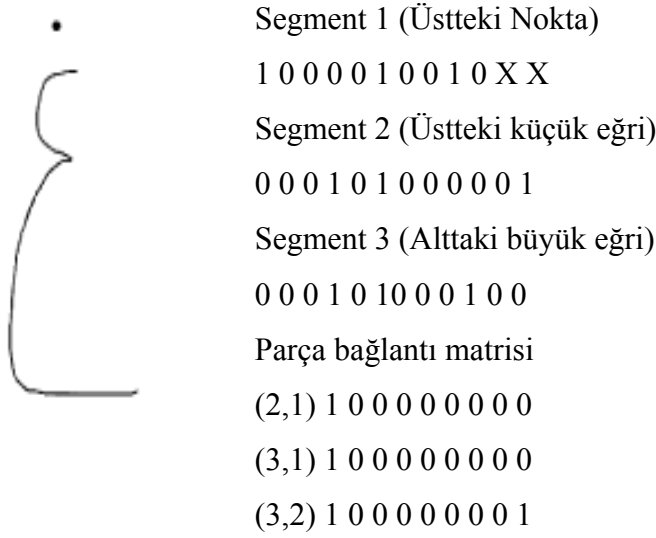
Geometrik ve topolojik öznitelik analizine dayalı bölütleme ve öznitelik çıkarımının ardından, harflerin ana gövdelerine zincir kodu transformasyon uygulanmıştır. Bölütleme aşaması ile elde edilen bölütler, farklı yöntemlerle işlenebilirler.

Zincir kodu dizilimleri Saklı Markov Model ile sınıflandırılmaktadır. Elde edilen özellikler, “trie” veri yapısı ağacı ile temsil edilen alfabede tam eşleme yöntemiyle işlenerek harf tanınması yapılmaktadır. Trie düğümleri, harfleri ayırtedici özniteliklere karşılık gelmektedir. Saklı Markov Model ile yapılan deneylerde ilk iki sırada görülen doğru seçeneklerin oranı oldukça yüksektir.

3.4. Arapça Elyazısı Karakter Tanıma Sistemi

Amin ve Al-Sadoun, el yazısı Arapça karakterleri tanımak için yapısal bir yaklaşım geliştirdiler [5]. Bu yaklaşıma göre ilk önce karakterin ikili görüntüsü paralel inceltme algoritması kullanılarak inceltilir. Sonra görüntünün iskeleti, karakteri temsil etmek için bir grafik oluşturmak üzere 3x3 penceresi kullanılarak soldan sağa doğru izlenir. Düz çizgi, eğri ve kapalı eğri benzeri unsurlar grafikten elde edilir. Son olarak karakter sınıflaması için 5 katmanlı yapay sinir ağı kullanılır.

Her karakter nokta, hemze, çizgi, eğri, kapalı eğri gibi sistemde bulunan parçalara dayalı olarak sınıflandırılır. Parçalar arasındaki ilişki nesnelere arası ilişki matrisine kodlanır. Giriş katmanının genel olarak 150 nöron kullanılır. Resim 3.3’te bu giriş katmanı tertibini kullanan Arapça bir karakterin yapay sinir ağı için genel gösterimi bulunmaktadır.



Resim 3.3. Arapça bir karakterin yapay sinir ağı giriş katmanı için genel gösterimi

3.5. Yapay Sinir Ağları Yardımıyla Basılı Arapça Karakterlerin Tanınması

Amin ve Mansoon, Arapça basılı metinleri tanımak için yapay sinir ağlarını kullanarak bir sistem geliştirdiler [6]. Bu sistem temel olarak üç adımda gerçekleştirilmiştir. Birinci adım 300 dpi tarayıcı kullanarak ikili görüntü haline dönüştürülen orijinal görüntüdeki ön işlem aşamasıdır. Bu aşamada inceltme, genişletme, normalizasyon gibi ön işlemler uygulanmıştır. Daha sonra birbirine bağlı parçalar biçimlendirilir.

İkinci olarak girilen Arapça kelimenin alt kelime sayısı, alt kelime içindeki tepe noktalarının sayısı, birbirini tamamlayıcı karakterlerin konumu ve sayısı gibi genel özellikleri hesaplanır.

Son olarak kelime sınıflandırılması için kullanılan 3 katmanlı yapay sinir ağının giriş katmanında 270 nöron kullanılmaktadır.

4. KARAKTER TANIMA SİSTEMİ

Karakter tanıma temel olarak kağıt ortamındaki veya eş zamanlı olarak bilgisayara tablet veya fare ile girilen verilerin ikili hale getirilip bunların karşılığı olan sembollere dönüştürülmesi işlemidir.

Dönüştürülen sembollerin karakter kod sistemlerinde karşılıkları vardır. Amerikan İngilizce'sinde kullanılan tüm küçük ve büyük harfleri, 0 ile 9 arasındaki rakamları, noktalama işaretlerini ve özel denetleme karakterlerini göstermek için standart ASCII kod tablosu yeterli iken, Japonca, Arapça ve Osmanlıca gibi diğer diller için UNICODE karakter sistemine ihtiyaç duyulmaktadır. Standart ASCII kod tablosu 7 bit sayı bileşimini kullandığı için 128 olası karaktere olanak sağlarken, genişletilmiş ASCII karakter setinde 8 bit kullanılarak 256 farklı karakteri tanımaya olanak sağlanmaktadır. Bununla beraber UNICODE karakter sistemi ise 16 bitlik sayı bileşimi ile dünyanın çoğu dilini kodlamada kullanılmak üzere kabul edilmiştir.

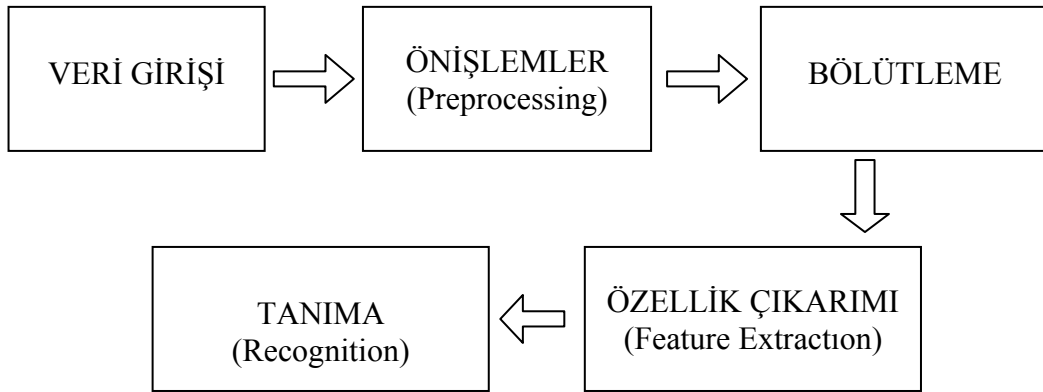
Karakter tanıma işlemi ile tamamlanan veriler artık metin olarak kaydedilebilir ve düzenlenebilir hale gelmiştir. Örneğin üzeri satırlarca yazı ile dolu bir kağıt, okuma bilmeyen bir çocuk için nasıl bir şey ifade etmiyorsa, tarayıcı ile taranıp bilgisayara aktarıldığında da bilgisayarlar için sizin vesikalık resminizden bir farkı yoktur. Bu resmin üzerindeki yazıları kullanabilmesi için bilgisayarın anlayacağı sembollere çevrilmesi gerekmektedir. İşte bu işlemler karakter tanımanın konusunu içermektedir.

Karakter tanıma basılı ve el yazısı olmak üzere iki ayrı kategoride ele alınmaktadır. Kullanılan yazı tipi ve stili bilinen basılı Latin harfler için tanıma yüzdesi %99'lar düzeyindedir. Ama alışılmadık yazı tipleri veya düşük kalitede baskılar ile karşılaşıldığında başarı düşmektedir. El yazısında ise insanın elde ettiği başarıya ulaşmak oldukça uzak görünmektedir.

Karakterler, her yazı sisteminde farklılık gösterdiğinden her biri için farklı karakter tanıma tekniklerine ihtiyaç vardır. Şu anda dünyada yaklaşık olarak iki düzine yazı sistemi kullanılmaktadır.

Her yazı sistemi belirli temel bir şekli olan kendi harflerine sahiptir. Bu harflerin yanı sıra rakamlar, noktalama işaretleri ve klavyede yer alan çeşitli semboller de yazıların içinde kullanılmaktadır. Dolayısıyla karakter tanıma tüm bu sembolleri kapsamalıdır [7].

Temel karakter tanıma aşamaları şema halinde Şekil 4.1’de gösterilmiştir.

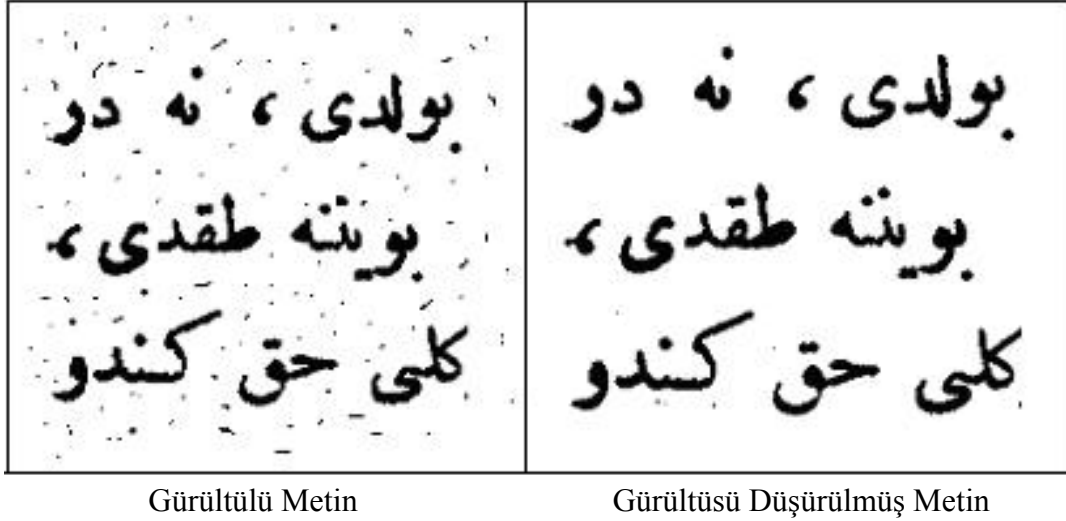


Şekil 4.1. Karakter tanıma sistemlerinin temel aşamaları

4.1. Önışlemler (Preprocessing)

Karakter tanıma sistemleri ilk olarak sisteme girilen işlenmemiş ham veriler üzerinde temel bazı düzenlemeler yaparlar. Bu işlemler karakter tanıma sisteminin ileriki aşamalarının doğruluğuna ve güvenilirliğine katkıda bulunmaktadır. Bu nedenle karakterlerin tanınmasında sayfa üzerinde yapılacak önışlemler aşaması çok önemlidir. Önışlemler eşikleme, gürültü oranının düşürülmesi, inceltme (thinning), genişletme (dilation), iskelet çıkarma gibi adımlardan oluşur.

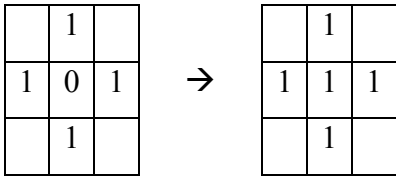
Gürültüsü düşürülmüş bir metine örnek olarak Resim 4.2 verilebilir.



Resim 4.2. Osmanlıca bir metnin gürültülü ve gürültüsü düşürülmüş hali

4.1.3. Genişletme (Dilation)

Genişletme işleminin amacı, resim içerisinde olabilecek kopuklukları ve küçük boşlukları gidermek ve bu işlemden sonra gerçekleşecek olacak inceltme işleminin yanlış sonuçlar vermesine engel olmaktır. Bu amaçla çerçevelenmiş görüntü üzerinde farklı maskeler kullanılmaktadır. Örnek olarak dört yönde komşuluklar ele alınarak merkez boş (beyaz) pikselin doldurulmasına yardımcı olan Şekil 4.2.'deki maskeyi verebiliriz.



Şekil 4.2. Bir genişletme işlemi maskesi

Bu maskeler değiştirilerek tek pikseli görüntüler de yok edilebilmektedir [7].

4.1.4. İnceltme (Thinning)

İnceltme işleminin amacı, harfleri tek piksel kalınlığında bir görüntüye dönüştürmektir. Bu işlem tanıma işlemini daha kolay ve hızlı hale getirmekle birlikte oluşan hataları da en aza indirmeyi amaçlar. Harf içerisindeki piksel sayısı en aza indiği için, işleme sokulacak nokta sayısı da azalmış olur. Bu da hızı artıran en önemli etmendir. Fakat bu işlemin kendisi de zaman almaktadır.

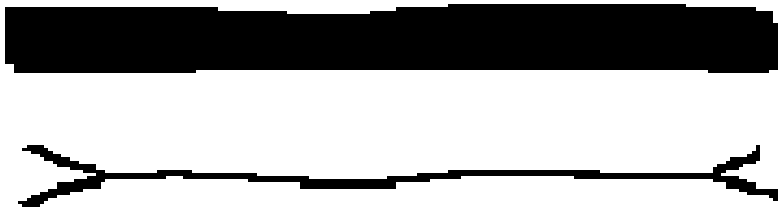
İnceltme işlemi için farklı maskeler kullanılmaktadır. Bunlara örnek olarak Zhang-Suen, Stentiford inceltme algoritmaları örnek verilebilir (Şekil 4.3).

			x			1	x		1	1	x
x	1	x	1	1		1	1		1	1	
1	1	1	1	1	x	1	x		x		

1	1	1	x	1	1		x	1			x
x	1	x		1	1		1	1		1	1
					x		x	1	x	1	1

Şekil 4.3. Zhang-Suen, Stentiford inceltme algoritmaları

Önce maskelerin 1 değerli merkez noktaları görüntüde bilgi taşıyan piksel üzerine oturtulur ve her maske ardışık olarak görüntü üzerinden geçirilerek inceltme işlemi gerçekleştirilir. Fakat bu maskeler uygulandığında uç noktalarda çatlama oluşmakta (Resim 4.3) ve karakterin bozulmasına sebep olmaktadır. Çatlama oluşmasının önlenmesi için de yeni maskelere ihtiyaç duyulmaktadır [7].



Resim 4.3. İnceltme algoritmaları uygulanmış bir görüntü

4.1.5. Onarım

Bu işlem aşamasında kopuk çizgilerin birleştirilmesi, eğri ve doğruların biçimlerinin düzenlenmesi işlemleri yapılmaktadır. Parçaları elde ederken, iskelete ait anahtar noktalar tanımlanır (köşe noktaları ve kesişen noktalar gibi). İki anahtar nokta arasındaki parça böylece elde edilir.

4.1.6. Normalizasyon

Normalizasyon, karakter geometrisindeki değişim ve standardizasyon olarak tanımlanabilir. Bu değişimler pozisyon, konum ve boyutta olabilir. Örneğin 12x7 ve 14x8 boyutlarında bir karakterin 40x20'lik bir alana taşınması bir normalizasyon (ölçekleme) işlemidir.

4.2. Karakter Ayırma (Segmentation)

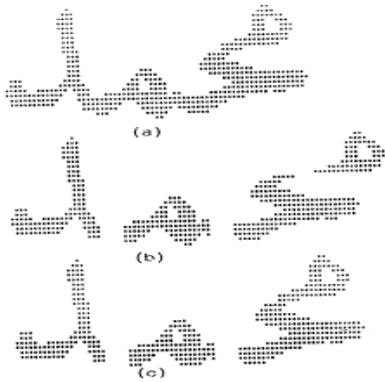
Osmanlıca'da matbaa harfleri ile de yazılsa bir metni oluşturan karakterler birbirine bağlı gruplar şeklinde karşımıza çıkmaktadır. Bu yüzden karakterlerin ayrılması oldukça önemli bir yer tutmaktadır. Osmanlıca harfler bir satırdaki kelime grubunda başta, ortada, sonda ya da diğerlerinden ayrı biçimde olmak üzere 4 farklı şekilde bulunmaktadır. Bu da ihtimallerin artması anlamına gelir [8].

Bazı harflerin buldukları yere göre şekilleri önemli ölçüde değişmektedir. Karakter ayırma işleminde izole harfler ve birbirine bağlı harfler grubu için siyah piksel sınırları kolaylıkla bulunabilir. Fakat bu, ön işleme aşamasına ek yük getirmektedir.

Harflerin bulunma konumlarından bağımsız bir ayırma işlemi için şöyle bir yöntem uygulanabilir; Öncelikle her satır için siyah noktaların yoğun olarak bulunduğu bir hat (baseline) belirlenir. Daha sonra bu hat üzerinde düşey olarak en az pikselin bulunduğu noktalardan harfler ayrıştırılabilir. Bu yöntemin temel sakıncası bazı harflerin baseline üzerinde ikiye veya üçe bölünmesidir. Bu durumda harf ayırma

işlemi harfin yanlış bölünmesine ve harfin bozulmasına neden olmaktadır Bu da harf tanıma aşamasında yanlış sonuçlara neden olmaktadır.

Karakter ayırma işlemi karakter tanıma sistemlerinin en önemli aşamalarından birisidir. El yazısı metinlerde kelime aralarında boşluk bulunduğundan kelime sınırlarını bulmak kolaydır. Fakat harflerin ayrıştırılması oldukça zor bir işlemdir. Çünkü yazış biçimine göre iki harf tek harf olarak görülebildiği gibi, tek bir harf de iki farklı harf gibi görülebilmektedir. Bilinen yöntemler bu gibi durumlarda başarısız kalmaktadır. Resim 4.4'te karakter ayırma işlemi doğru yapılmış bir kelime görülmektedir [2].



Resim 4.4. Muhammed kelimesinin bölütlenmiş hali [1]

4.3. Özellik Çıkarımı (Feature Extraction)

En genel ifadeyle özellikler karakteri oluşturan ve karakterden çıkarılabilen ölçüm değerleridir. Özellikler iki kategoride sınıflandırılmaktadır:

- Geometrik özellikler (iç ve dışbükey parçalar, kesişim noktaları v.b)
- Topolojik özellikler (bağlantı noktaları, birleşik bileşen sayısı, deliklerin sayısı v.b)

Özellik çıkarım işlemi, karakter tanıma sisteminin ana bölümlerinden biridir. Burada dikkat edilmesi gereken noktalar ise aşağıda verilmiştir:

- Özelliklerin bilgisayar tarafından işlenebilir olması,
- Doğru sınıflandırmaya yardımcı olması,
- Ham ölçümleri işlenebilir hale getirirken, karakteri belirleyen özellikleri yok etmemesi.

4.4. Tanıma (Recognition)

Karakter tanıma için temel olarak kullanılan yöntemleri üçe ayırabiliriz. Matris eşleme, içerik analizi ve özgün tanımlama.

4.4.1. Matris eşleme (Matrix matching)

Bu yöntemde ilk önce karşılaştırılması muhtemel tüm karakterlerin görüntüsü tablolar halinde tutulur, daha sonra taranmış dokümandaki karakterleri sırasıyla bu kütüphanedeki şekillerle eşleyerek hangisine uygun olduğunu bulmaya çalışır. Hızlı çalışan bir yöntemdir, fakat okunacak fontun şekli biraz olsun değiştiğinde hassasiyeti kaybolur. Bundan dolayı bu yöntemin başarısı için her yazı tipine ait bir şablon kümesi bulunmalıdır.

4.4.2. İçerik analizi (Feature analysis)

Bu yöntemde ise karakterlerin görüntüsü değil, tanımları tutulur. Karakteri tanımak için öznitelik çıkarımı aşamasında hesaplanan özelliklere bakılır. Örneğin karakterde kaç tane düz çizgi var, bunların kaç dikey kaç yatay, yuvarlak köşelerin konumları neler, karakterde delik ve nokta var mı gibi.

Tanımlama kriterleri böyle her karakter için karakteristik özellikler olduğundan genel kurallara göre yazılan hemen hemen her karakterin tanınması mümkün hale gelir. Bu durumda karakterin şekli, genel karakter kurallarına uyduğu sürece karakter tanımlama için fontların fazla önemi yoktur. Bu aşamanın güvenilirliği doğrudan özellik çıkarımı aşamasına bağlıdır.

4.4.3. Özgün tanımlama (Self-assertion)

Bu yöntem de iki yöntemin birleşmesinden oluşmaktadır. Önce doküman içerik analizi yöntemiyle taranarak genel bir tablo oluşturulur ve bu sayede keskinliği yüksek karakterler bir kenara ayrılarak bunlardan bir matris oluşturulur.

Daha sonra emin olunmayan karakterler, bu yeni elde edilen matristeki karakter tablolarıyla karşılaştırılarak “benzetilmeye” çalışılır. Yani sistem önce içerik analiziyle kesinlikle “A” olduğuna emin olduğu bir “A” yakalayarak bunun fotoğrafını çeker. Daha sonra da bu “A” harfinin görüntüsünü “A” olduğundan emin olmadığı fakat “A” ya benzediğini düşündüğü karakterlerle karşılaştırıp ortak noktalarına bakarak gerçekten “A” olup olmadığına karar verir.

Son zamanlarda çalışmalar bu yöntemlerin birleştirilerek tanıma oranlarında ilerleme elde edilebileceğini göstermektedir.

5. EL YAZISI TANIMA

Bilgisayar kullanılarak el yazısının tanınması bilgisayar biliminin başlangıcından bu yana araştırmacıları meşgul eden zor problemlerden biridir. Bu konuda çok sayıda çalışma yapılmasına rağmen halen problem devam etmektedir. Yazı içerisinde farklı karakterlerin benzerliği, harflerin birleştirilmesi, bilgilerin bulanıklığı ve çok çeşitliliği bu problemi zor kılan unsurlardandır.

El yazısı tanıma konusunda temel olarak iki sistem bulunmaktadır. Çevrimiçi ve çevrimdışı sistemler [9].

Çevrimiçi sistemler, el yazısını yazı yazıldığı sırada tanıyan, özel olarak tasarlanmış sistemlerdir. Genelde elektromanyetik ya da elektrostatik tabletler kullanılır. Kalem dokunuşları ve hareketlerin devamlılığı göz önünde tutulur. Avuçiçi bilgisayarlar (PDA) gibi günümüzde çok yaygınlaşan bir yöntemdir.

Çevrimdışı sistemler ise genelde kağıt üzerine daha önceden yazılmış bilgilerin sayısallaştırılarak, sonradan tanınmaya çalışılması işlemidir.

Çevrimiçi sistemlerin yazı hızına yetişebilmesi için çok hızlı olması gerekirken çevrimdışı sistemlerde yazının tanınması için bir süre kısıtlaması gerekmemektedir. Çevrimiçi sistemlerin avantajı harflerin şekil özelliklerinin yanında yazılma sırasındaki hareketlerin de gözlenebilmesidir. Ayrıca kullanıcıyla etkileşimin en büyük faydalarından biri gerektiğinde yanlışların anında düzeltilebilmesidir.

Çevrimdışı sistemlerse, yazının yazılması sırasındaki hareketler hakkında hiçbir bilgi olmadığı ve özellikle eski belgeler yeterince temiz ve okunaklı olmayacağı için hata oranı daha yüksek olan sistemlerdir.

Sayısallaştırıcılardan kaynaklanan gürültülerin etkisini azaltmak için çok daha detaylı bir ön işlemeye gerek duyulur. Ancak bu sistemlerin avantajı özel bir alete

gerek duyulmaması ve bu sayede yıllardır var olan bütün belgelere uygulanabilmesidir [10].

El yazısı karakterlerin tanınmasında kullanılan yöntemleri iki ana grupta toplayabiliriz:

- a. Holistik yöntemler, bir harfi analiz etmeden tüm kelimeyi tanımayı amaçlar. İnsanın okuma aktivitesinin incelenmesi yönünde yapılan son çalışmalar, genel tanıma temelinde bir sürecin mümkün olduğunu göstermektedir. Bu yaklaşımlar sözlüğün boyutuna bağlı olarak sistemde bir karmaşıklık artışına yol açabilmektedir. Bu açıdan tanıma genellikle sınırlı sözlükler üzerinde gerçekleştirilmektedir. Holistik yöntemler iki alt grupta toplanır.
 - i. Dinamik programlamayı kullanan mesafe ölçmeye dayalı yöntemler.
 - ii. Probalistik çerçeve çalışmasına dayalı yöntemler (Saklı Markov Modeli) [1].
- b. Analitik yöntemlerde ise el yazısı karakteri tanıma süreci bir kelimeyi onun harfleri ile tanı ve kelimeyi tanımak için her harfin yerini içerir. Bu yaklaşımda her kelime, sözlük kullanılmadan tanınabildiği için oldukça ilgi çekicidir. Fakat buradaki temel sorun el yazısının dilimlenmesi problemidir. Yapılmış birçok çalışma olmasına rağmen dilimleme yalnız sınırlı uygulama alanları için geçerlidir.

Son zamanlarda çalışmalar bu iki yöntemin birleştirilerek tanıma oranlarında ilerleme elde edilebileceğini göstermektedir [7].

El yazısının taranmasında holistik yöntemlerde genellikle el yazısının genel görünüşünü, sınırlarını belirledikten sonra el hareketinin üst ve alt satır sonlarının (overstrokes) ve kelime uzunluğunun belirlenmesine çalışılır. El yazısı bu hareketlerin yöne ilişkin artması ve azalması şeklinde yorumlanabildiği için, bir kelimeye ait temel bileşenleri olarak kullanılabilir. Bu nedenle söz konusu özellikler kelimedeki dayanak (anchor) noktası olarak seçilmektedir. Kelimenin çizim sınır hattı ile ortalama

kseninin keşime noktası dayanak noktasını belirlemektedir. Önemli bilginin gövdenin alt ve üst kısımları arasındaki geçiş bölgesinde yerleştiği hipotezi olmaktadır. Kelime ekseni bu iki bölümü ayırır ve dayanak noktası bu bölümler arasındaki geçişleri gösterir. El yazısı kelimenin ferdi bileşenleri, genellikle harflere benzemektedir [7].

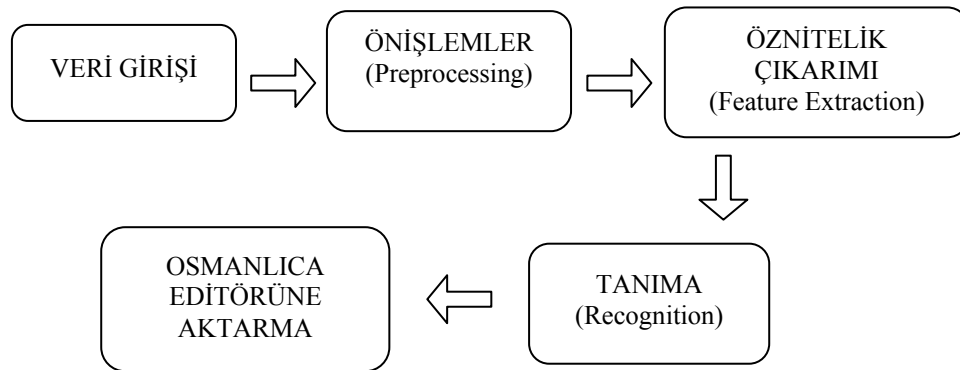
6. OSMANLICA ELYAZISI HARFLERİ ÇEVİRİMİÇİ TANIMA

Bu çalışmanın amacı, hazırlanan arayüzüne fare yardımıyla çizilen Osmanlıca el yazısı harflerin tanınması ve Osmanlıca metin editörüne aktarılacak hale getirilmesidir.

Çalışmada;

- Ön işlemler (Preprocessing)
- Öz nitelik çıkarımı (Feature extraction)
- Karakterin tanınması (Recognition)
- Editöre aktarılması

aşamaları takip edilmiştir.



Şekil 6.1. Osmanlıca el yazısı karakterleri çevrimiçi tanıma sisteminin aşamaları

Özetle ön işlemler aşamasında sınırlandırılan alan içerisinde çizilen şeklin özellikleri hesaplanmış. Çıkarılan özellikler harf tablosundaki özelliklerden hangisine uyuyorsa çizilen şekil o karakter olarak tanınmıştır. Tanımlanan özellikler o harfi tanımlayan temel karakteristikler olduğu için genel kurallara göre çizilen her harf tanınmıştır. Tanınan karakter de Osmanlıca olarak editöre aktarılacak hale getirilmiştir. Sonuçlarda %100'e varan tanınma oranlarına ulaşılmıştır. Genel harf tanıma oranı ise %92'dir.

6.1. Önışlemler

İlk olarak programda harfin girileceđi alan, 150x100 piksellik bir alanla sınırlandırılmıřtır. Çizilen harflerde hassasiyeti artırmak için farenin hareketi yavaşlatılarak fare hızından dolayı yakalanamayan piksellerin önüne geçilmiřtir. Böylece el titremesinden kaynaklanan gürültülerin de önlenmesi sađlanmıřtır.

6.2. Öznitelik Çıkarımı

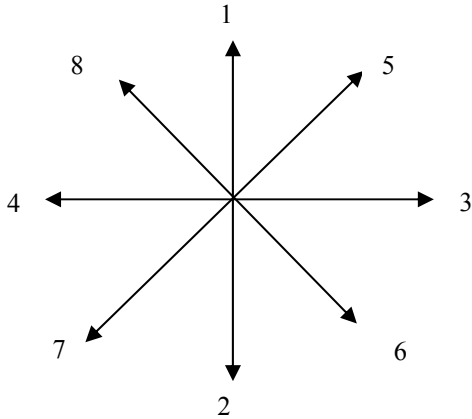
Bu aşamada çizilen řekile ait temel karakteristik özellikleri çıkarılmıřtır. Bunlar:

- Çizime ait tüm koordinat deđerleri
- Çizime ait dört yönlü sınır deđerleri
- Nokta sayısı
- Noktaların birbirine uzaklıkları
- Noktaların harfe uzaklıkları
- Bařlangıç ve bitiş noktası arasındaki eğim
- Harfe ait yön bilgileri

İlk olarak kullanıcı harfi çizerken fare hareketlerine göre piksel bilgileri x ve y koordinatları için ayrı ayrı tutulmuřtur. Bu piksel bilgileri ile dört yönlü sınır deđerleri, nokta sayısı, noktaların birbirine ve harfe uzaklıkları, bařlangıç ve bitiş noktaları arasındaki eğim hesaplanmıřtır [11]. Karakteristik özellikler içerisinde en temel özellik ise yön bilgisidir.

6.2.1. Yön bilgisinin hesaplanması

Yön bilgisinin belirlenmesi için yön kodlaması tekniđi [9] kullanılmıřtır. Eř zamanlı olarak kullanıcının fare hareketlerine göre gittiđi yönler belirlenmiř ve bunlara Şekil 6.2'deki numaralar verilmiřtir . Ana yönler 1, 2, 3, 4, arayönler ise 5, 6, 7, 8 dir.



Şekil 6.2. Osmanlıca karakterlere ait tanımlanan yönler

Yön bilgilerinin hesaplanması için mevcut koordinat bilgileri daha önceki koordinatların değerleri ile karşılaştırılmıştır. Bunun için fare hareket ederken seçilen en uygun aralık değeri kadar geri piksellerle karşılaştırma yapılmıştır. Farenin o anda bulunduğu x ve y piksel değeri ile karşılaştırma yapılan geri piksel değeri arasındaki fark yine uygun bir eşik değerini aşıyorsa yön değiştirildiği anlaşılmaktadır [12, 13].

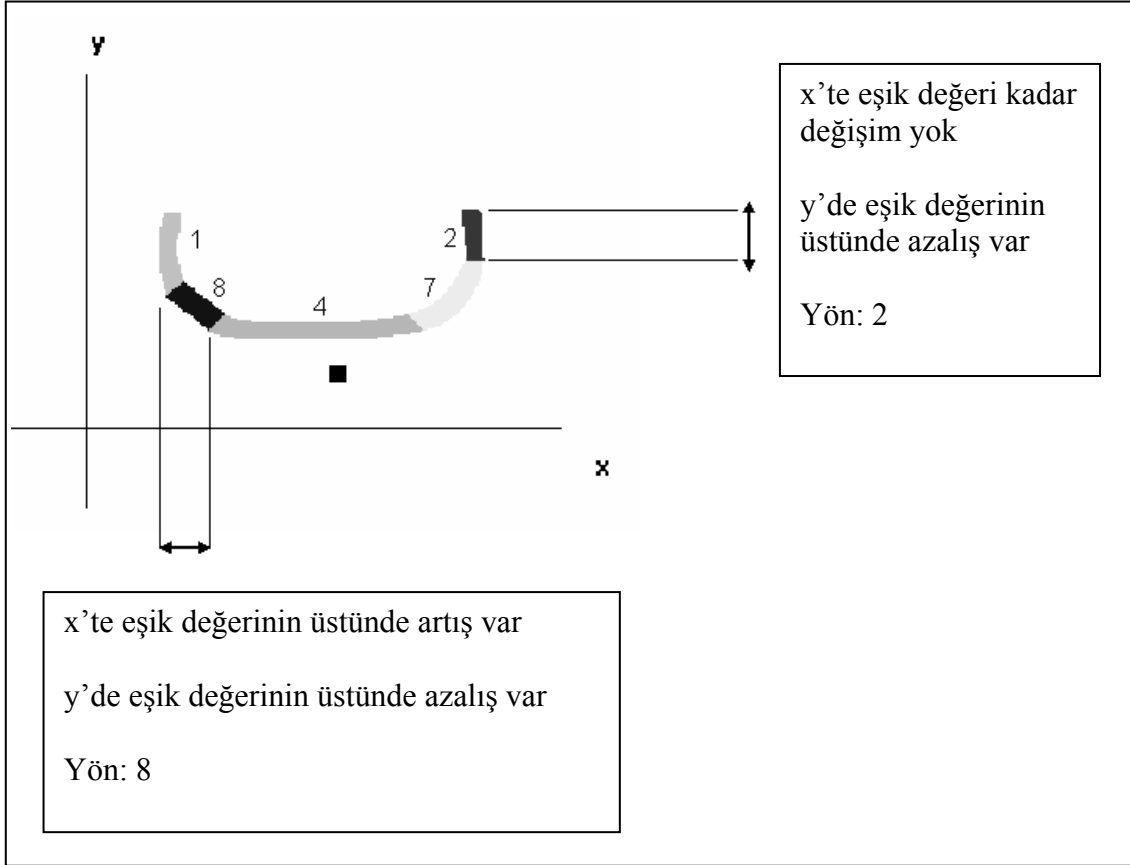
Yapılan denemelerde 150x100 piksellik bir alanda geri karşılaştırma aralığının 15 birim, fark olarak karşılaştırılacak eşik değerinin ise 7 birim olduğu gözlemlenmiştir.

Örneğin Mouse ile ekrana “ELİF” harfi çizildiğinde yön bilgisi 2 olacaktır.



Resim 6.1. Elif harfine ait yön bilgisini gösteren ekran görüntüsü

Be harfine ait yön kodları Resim 6.2’de görülmektedir.



Resim 6.2. Be harfine ait yön kodlarının gösterimi

Resim 6.2’de de görüldüğü üzere kullanıcı ilk olarak fareyi hareket ettirirken y değeri 15 piksel öncesine göre eşik değerinin üzerinde azalış göstermiş, fakat x değerinde eşik değeri (7 birim) kadar değişim görülmemiştir ve kullanıcının 2 yönünde hareket ettiği anlaşılmıştır.

Daha sonra gittiği dördüncü yönde x değeri artarken y değeri ise azalmıştır. Buradan kullanıcının 8 yönünde ilerlediği anlaşılmaktadır.

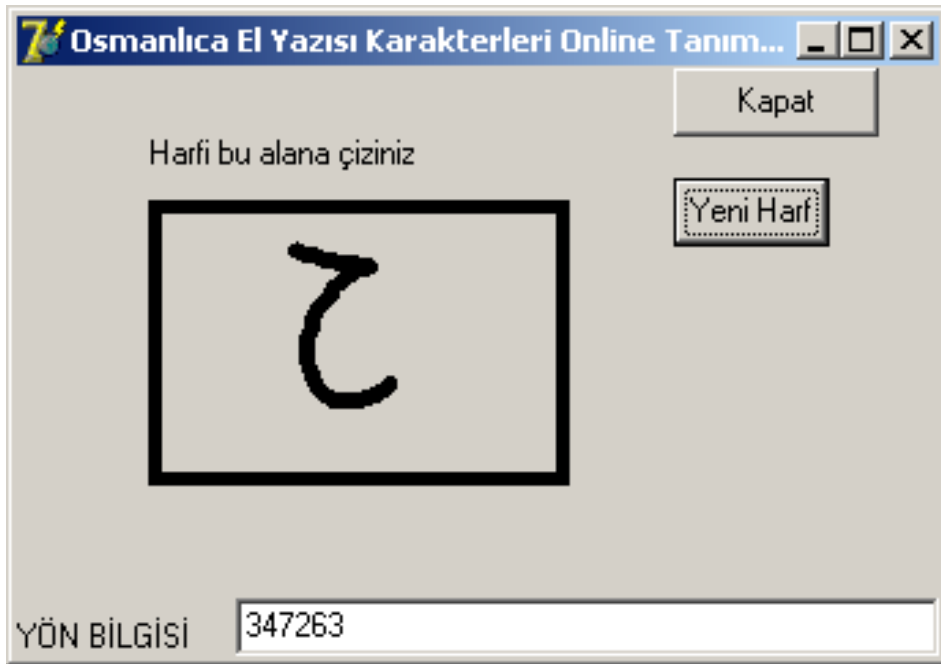
Harflerin yön bilgilerinin x ve y koordinatlarına göre hesaplama kriterleri Çizelge 6.1’de verilmiştir.

Çizelge 6.1. Yön bilgilerini hesaplama kriterleri

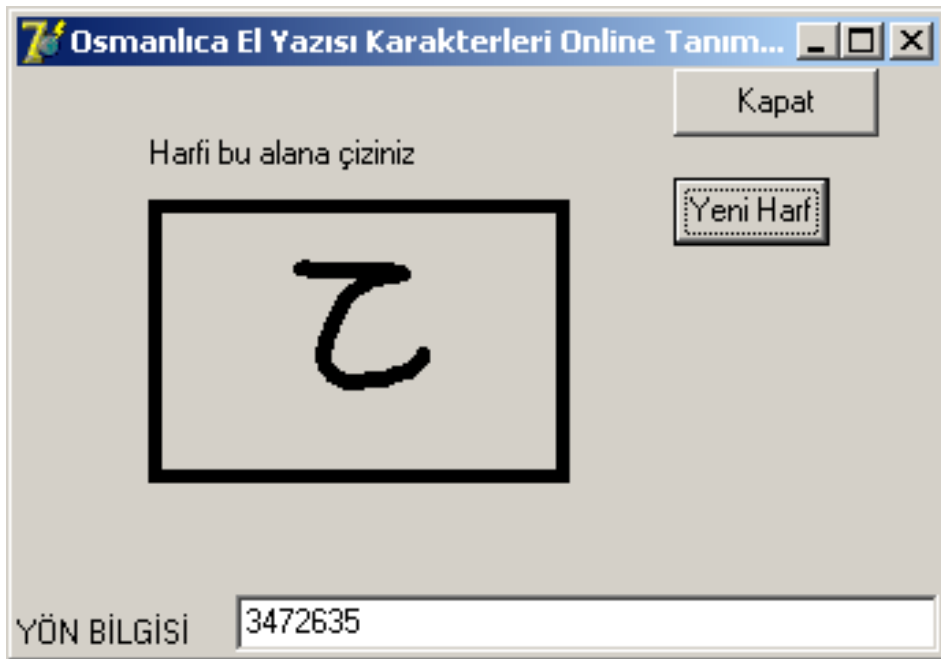
Yönler	Eşik değeri kadar değişim var mı ?	
	x	y
1	Yok	Artış
2	Yok	Azalış
3	Artış	Yok
4	Azalış	Yok
5	Artış	Artış
6	Artış	Azalış
7	Azalış	Azalış
8	Azalış	Artış

Her harfe ait bu şekilde yön bilgileri hesaplandığında çoğu harf için kullanıcı çizim sitillerine göre birden fazla yön bilgisine ulaşılmaktadır. Fakat her harf içinde harflere ait genel karakteristik özelliklerinden kaynaklanan temel yönler bulunmaktadır.

Bu farklı yön bilgilerini karakterin tanınması aşamasında kullanabilmek için 10 farklı kullanıcıya her harf için tatminkar sonuçlar elde edilinceye kadar çizimler yaptırılmıştır. Sonuçlar içinden harfin genel karakteristik özelliklerine uymayanlar elenmiş, doğru karakterlerin yön bilgileri her harf için yön bilgisi tablosuna işlenmiştir. Örneğin Resim 6.3'te ve Resim 6.4'teki farklı yön bilgileri HA harfine aittir.



Resim 6.3. Ha harfine ait yön bilgilerinden biri



Resim 6.4. Ha harfine ait yön bilgilerinden bir başkası

Yön bilgilerinin elde edildiği Delphi 6.0 programlama dilinde yazılmış yordamlar aşağıya çıkarılmıştır.

Bu kod fare her hareket ettiğinde tekrarlanmaktadır (programda daha önce aralık değeri 15, eşik (fark) değeri 7 olarak tanımlanmıştır).

```
procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
Y: Integer);
```

```
Begin
```

```
// Eğer Mousun Sol Tuşu Basılı Ve Çizim Belirtilen Alana Yapılıyorsa
```

```
if basili and (x>x1+xy)and(x<x2-xy)and (y>y1+xy)and  
(y<y2-xy) then
```

```
begin
```

```
xyon:=0;
```

```
yyon:=0;
```

```
yon:=0;
```

```
xdegisti:=false;
```

```
ydegisti:=false;
```

```
// Forma Çizim Yapılıyor Ve X,Y Koordinat Değerleri Diziye Aktarılıyor
```

```
canvas.LineTo(x,y);
```

```
yolx[adet]:=x;
```

```
yoly[adet]:=y;
```

```
adet:=adet+1;
```

```
// Çizilen Pikseller Aralık Değerini Aştığında Kontrol Yapılıyor
```

```
if adet>aralik then
```

```
if xyon=0 then
```

```
// Eğer Şu Anki X Değeri 15 Kayıt Önceki Değerden 7 Büyükse Yön Bilgisi "3" tür
```

```
if x-yolx[adet-aralik]>fark then
```

```

begin
  xyon:=3;
  xdegisti:=true;
end

```

// Eğer Şu Anki X Değeri 15 Kayıt Önceki Değerden 7 Küçükse Yön Bilgisi “4” tür

```

else if yolx[adet-aralik]-x>fark then
begin
  xyon:=4;
  xdegisti:=true;
end;

```

// Eğer Şu Anki Y Değeri 15 Kayıt Önceki Değerden 7 Büyükse Yön Bilgisi “2” dir

```

if adet>aralik then
  if yyon=0 then
    if y-yoly[adet-aralik]>fark then
      begin
        yyon:=2;
        ydegisti:=true;
      end
    end
  end

```

// Eğer Şu Anki Y Değeri 15 Kayıt Önceki Değerden 7 Küçükse Yön Bilgisi “1” dir

```

else if yoly[adet-aralik]-y>fark then
begin
  yyon:=1;
  ydegisti:=true;
end;

```

```

if (xdegisti=true) and (ydegisti=false) then

```

```

yon:=xyon
else if (xdegisti=false) and (ydegisti=true) then
  yon:=yyon
else if xdegisti and ydegisti then
  if (xyon=3) and (yyon=1) then      {X=3 Y= 1 ise Yön Bilgisi "5"}
    yon:=5
  else if (xyon=3) and (yyon=2) then  {X=3 Y= 2 ise Yön Bilgisi "6"}
    yon:=6
  else if (xyon=4) and (yyon=2) then  {X=4 Y= 2 ise Yön Bilgisi "7"}
    yon:=7
  else if (xyon=4) and (yyon=1) then  {X=4 Y= 1 ise Yön Bilgisi "8"}
    yon:=8;

```

//Yön Değişim Bilgileri "Sonuç" Değişkenine Aktarılıyor

```

if (yon<>0) and (copy(sonuc,sonucadet,1)<>inttostr(yon)) then
begin
  sonuc:=sonuc+inttostr(yon);
  sonucadet:=sonucadet+1;
end;
edtyon.Text:=sonuc;
end;

```

Öznitelik olarak harfe ait yön bilgisinden başka harfin dört yönlü sınır değerleri de hesaplanarak enbuyukx, enbuyuky, enkucukx, enkucuky değişkenlerine aktarılmaktadır. Ayrıca harfte bulunan nokta sayısı ve bunlarında birbirine ve harfe olan uzaklıkları bulunmaktadır. Öznitelik çıkarma aşamasında hesaplanan ve tabloya kaydedilen değerlerin doğruluğu ölçüsünde tanıma aşamasındaki oranlarında yükseldiği görülmektedir.

6.3. Karakter Tanıma (Recognition)

Bu aşamada çizilen karakterler karakter tablosuna girilen değerlerle karşılaştırılmış, doğru değerlere sahip olanlar tanınmış ve harfin adı formun tanınan harf kısmına yazılmıştır. Tanınmayan harf için “Tanımsız” ifadesi yazılmıştır. Tekrar harf denemesi yapmak için “YENİ HARF” butonu konulmuştur. Bu butona basıldığında çizimle beraber tüm değişkenler sıfırlanmaktadır.

Farenin sol tuşuna basılıp harf çizilmeye devam edildiği sürece yön bilgileri hesaplanmıştı. Sol tuş bırakıldığında ise ilk olarak diğer öz nitelikler hesaplanıp sonra harf tanıma aşamasına geçilmektedir.

Tanıma aşamasında ilk olarak noktasız harfler tablodan aranarak sonuç yazılmakta, daha sonra nokta sayısı, noktaların harfe ve birbirine olan uzaklıklarına göre noktalı karakterler arasında arama yapılmaktadır. Uygun sonuç bulunursa ekrana yazılmakta, eşleşen sonuç olmadığı zaman ise ekrana “Tanımsız” uyarısı gelmektedir.

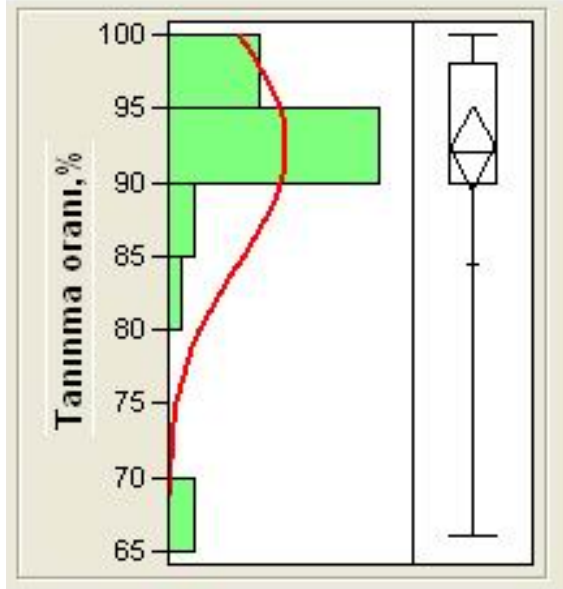
Tanınma oranlarının belirlenmesi için 10 farklı kullanıcıdan her harf için 10 doğru çizim yapması istenmiştir. Yanlış yapılan çizimler tekrarlatılmış, doğru yapılan çizimlerden ise sadece tanınanlar çizelgeye tanındı olarak işlenmiştir. Böylece her harfe ait 100 geçerli girişe göre tanınma oranları belirlenmiştir. Çizelge 6.2’ de bu bilgilere göre hazırlanmış harflerin tanınma oranları gösterilmektedir.

Çizelge 6.2. Osmanlıca elyazısı harflerin tanınma oranları

Harf	Tanınma Oranı (Yüzde)	Harf	Tanınma Oranı (Yüzde)
ELİF	100	DAT	87
HEMZE	96	TI	92
BE	98	ZI	93
PE	98	AYIN	90
TE	97	ĞAYIN	90
SE	98	FE	92
CİM	92	KAF	92
ÇİM	92	KEF	91
HA	94	GEF	90
HI	94	NEF	90
DAL	100	LAM	98
ZEL	100	MİM	92
RE	100	NUN	95
ZE	100	VAV	95
JE	100	HE	82
SİN	69	LAMELİF	91
ŞİN	66	YE	92
SAT	89		

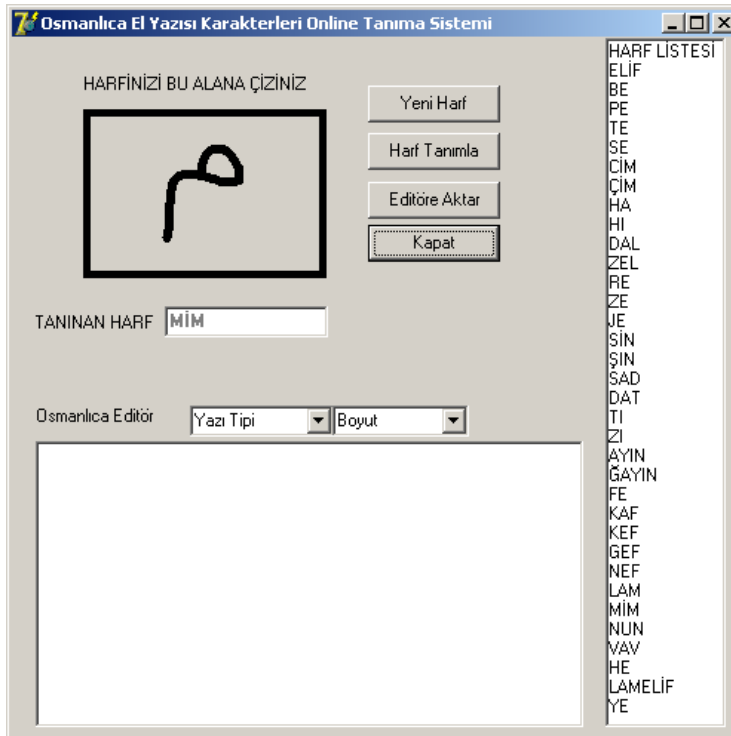
Osmanlıca harflerin genel tanınma ortalaması: %92'dir.

Harf tanınma oranları ile ilgili dağılım grafiği ve kutu grafiği Şekil 6.3’de verilmiştir.

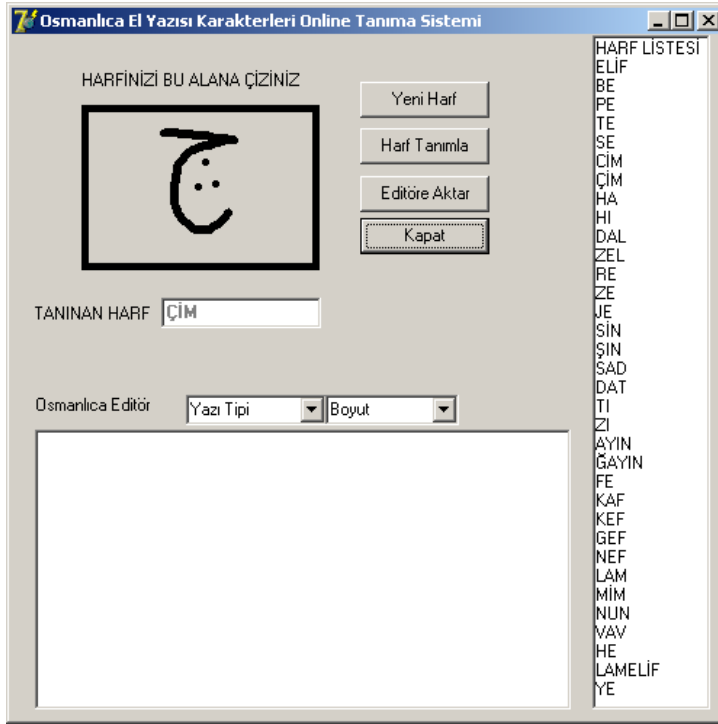


Şekil 6.3. Harf tanınma oranları ile ilgili dağılım grafiği ve kutu grafiği

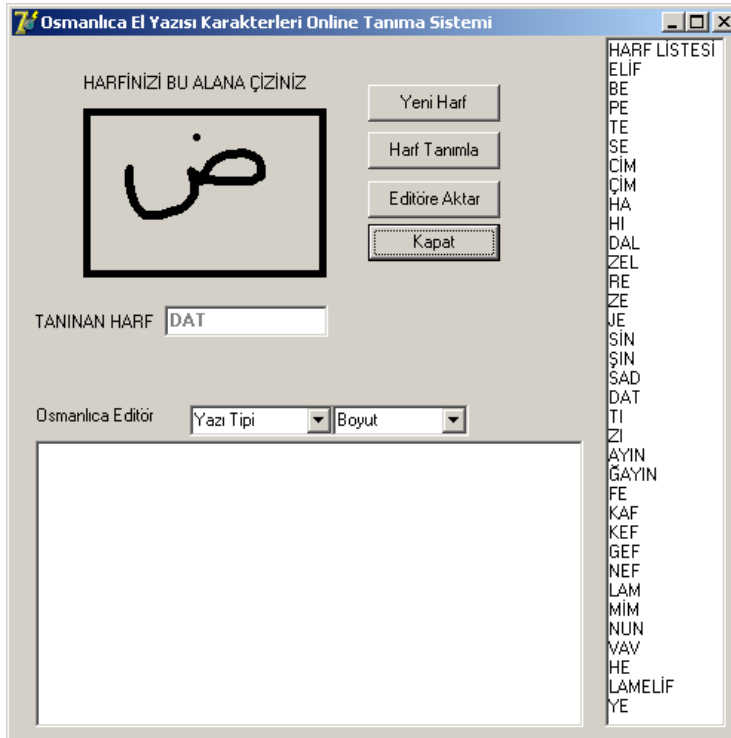
Tanınan bazı harfler Resim 6.5’te, Resim 6.6’da ve Resim 6.7’de gösterilmiştir.



Resim 6.5. Mim harfinin tanınması



Resim 6.6. Çim harfinin tanınması



Resim 6.7. Dat harfinin tanınması

6.4. Harflerin Osmanlıca Metin Editörüne Aktarılması

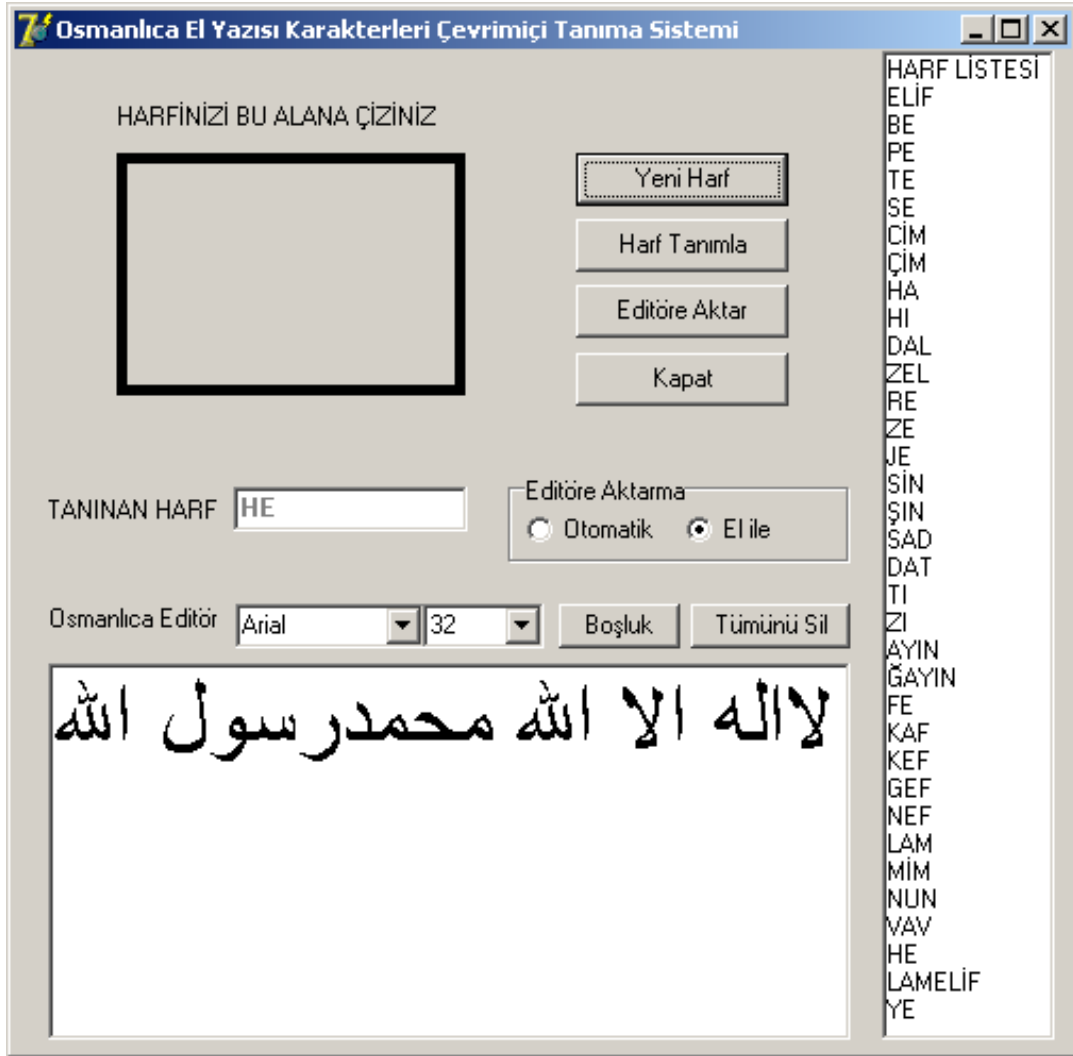
Bu aşamada da tanınan harfler UNICODE karakter setindeki karşılıklarına göre editöre aktarılmakta ve düzenlenebilecek hale gelmektedir. UNICODE karakter sisteminde karakterlerin sağdan ve soldan birleşme özellikleri tanımlı olduğu için harflerin birleşmesinde problem yaşanmamaktadır ve editöre eklenen harfler kendi özelliklerine göre sağa veya sola birleşebilmektedir.

Resim 6.8’de ŞİN harfinin editöre eklenmiş hali görülmektedir.



Resim 6.8. Şin harfinin Osmanlıca editörüne aktarılması

Resim 6.9'da ise sırasıyla La İlahe İllallah cümlesinin harfleri çizilmiş ve editöre eklenmiştir.



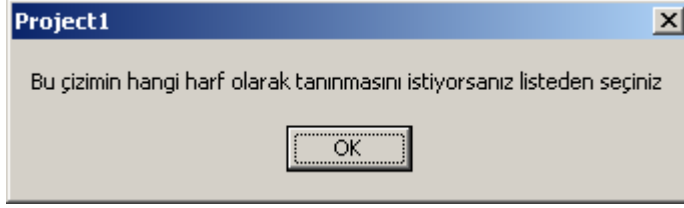
Resim 6.9. La İlahe İllallah cümlesinin editöre aktarılmış hali

6.5. Program Düğmelerinin Tanıtımı

a) Yeni Harf: Çizim alanını temizler ve yeni harf girişi için tüm değişkenleri sıfırlar

- b) Harf Tanımla:** Çizilen şeklin bir harfe ait özelliklere eklenmesini sağlar. Kullanımı için ilk olarak tanımlanacak harf çizilir. “Harf Tanımla” butonuna basılır.

Ekrana Resim 6.10’daki uyarı gelir.



Resim 6.10. Harf tanımlama uyarısı

Daha sonra tanımlanacak harf listesinden seçilerek işlem gerçekleştirilir.

- c) Editöre Aktar:** Tanınan karakterin Osmanlıca editöre aktarılmasını sağlar.
- d) Kapat :**Programı sonlandırır.

7. SONUÇLAR

Bu tezde, hazırlanan arayüzdeki sınırlandırılmış alana fare ile çizilen Osmanlıca harflerin çevrimiçi tanınması ve Osmanlıca metin editörüne aktarılması sağlanmıştır.

Program arayüzünün tasarlanması için Delphi 6.0 görsel programlama dili kullanılmıştır. Tanınan karakterler Unicode karakter sistemine dönüştürülmüştür. Hazırlanan programda yapısal yaklaşıma ait yön kodlaması yöntemi kullanılmıştır.

İlk olarak kullanıcının çizim yapacağı arayüzdeki alan 150x100 piksel ile sınırlandırılmıştır. Kullanıcının el titremeleri ve ani hareketlerinden etkilenmemesi ve gürültülere neden olmaması için farenin hızı yavaşlatılmıştır. Böylece farenin hızından dolayı yakalanamayan ve atlanan piksel değerlerinin de önüne geçilmektedir.

Kullanıcı fare ile arayüze çizimini yaparken eş zamanlı olarak hangi yönlere gittiği hesaplanmıştır. Bunun için kullanıcının takip ettiği x ve y koordinatları bir diziye aktarılmıştır. Yön değişimlerinin bulunabilmesi için aktif x ve y değerleri geri değerlerle karşılaştırılmış, belirli bir eşik değerini aştığında ise yön değiştirildiği anlaşılmıştır. Bu değer de harfin yön bilgisine kaydedilmiştir. Yapılan denemeler sonucunda 150x100 piksellik bir alana çizilen harfe ait uygun geri karşılaştırma aralığının 15 birim, aktif değer ile karşılaştırılan değer arasındaki uygun eşik değerinin ise 7 birim olduğu gözlenmiştir.

Karakterin tanınması için harfe ait yön bilgisinden başka başlangıç ve bitiş pikselleri arasındaki eğim, noktaların konumu, adedi, birbirine ve harfe olan uzaklık değerleri de hesaplanmıştır.

Çizim anında hesaplanan değerler daha önce hazırlanan harf tablosundaki değerlerle karşılaştırılmış, uygun eşleşmeler sağlandığında harf tanıma gerçekleşmiştir.

Sistemin önemli özelliklerinden biri de harflere ait karakteristik özelliklerin tutulduğu tablonun yine hazırlanan arayüz ile oluşturulmasıdır. Burada en temel özellik harfin yön bilgisidir. Harf karakteristik tablosunun hazırlanması için Osmanlıca yazmayı ileri derecede bilen 10 farklı kişiye tüm harfler doğru tanıma oranına ulaşıncaya kadar farklı şekillerde arayüze çizdirilmiştir. Harf tablosu Nesih hattına göre oluşturulmuştur. Çizilen her doğru şeklin özellikleri tabloya işlenmiştir.

Tabloya girilen değerler o harfe ait genel karakteristikler olduğu için genel yazım kurallarına göre Nesih Hattında çizilen her harf tanınmıştır.

Karakteristik tablosu oluşturulurken çizilen harfte yön bilgisi karmaşıklaştığında ve çok fazla yön değişimine ait harflerde tanıma oranı azalmaktadır. Tanımlamada en çok karıştırılan harf Sin ve Şın harfleri olmuştur. Sin ve Şın harflerinin tanıma ortalaması %67,5'tir. Elif, Dal, Zel, Re, Ze, Je harflerinin tanıma ortalaması %100'dür.

Harflere ait tanıma oranları Çizelge 6.2'de verilmiştir. Özet olarak tanıma oranları ise Çizelge 7.1'de görülmektedir.

Çizelge 7.1. Osmanlıca elyazısı harflerin tanıma oran aralıkları

Tanıma Oranları (Yüzde)	Harfler
100	Elif, Dal, Zel, Re, Ze, Je
95-99	Hemze, Be, Pe, Te, Se, Lam, Nun, Vav
90-94	Cim, Çim, Ha, Hı, Tı, Zı, Ayın, Ğayın, Fe, Kaf, Kef, Gef, Nef, Mim, Lamelif, Ye
85-89	Sad, Dat
80-84	He
65-70	Sin, Şın

Tüm harflerin genel tanıma ortalaması %92'dir.

Hazırlanan program kişisel bilgisayarlarda (PC), Elektronik tahtalarda, Tablet PC lerde kullanılabilir durumdadır. Cep telefonları ve avuçiçi bilgisayarlarda (PDA) ise gerekli uyarlamalar yapıldığında kullanılabilir.

Sonuç olarak öznitelik çıkarımında harfe ait karakteristik özelliklerin doğruluğu arttıkça, yön bilgileri gereği kadar artırıldığında ve fare ile çizilen piksellerin yakalanma oranını artıracak şekilde bilgisayar sistemi hızlandırıldığında daha yüksek tanınma oranları elde edilebilecektir. Sistemi hızlandırmak için farklı programlama dilleri de kullanılabilir.

Hazırlanan programda kelime bazında da tanınma özelliği eklenmesi için, öznitelik tablolarında harflerin başta, ortada ve sondaki halleri de girilebilir veya harf ayırma özelliği eklenebilir.

KAYNAKLAR

1. Amin, A., "Offline Arabic Character Recognition: The State Of Art", *School of Computer Science And Engineering*, 31(5): 520-521 (1998).
2. Öztürk, A., "Osmanlıca Karakterlerin Bilgisayar Destekli Tanınması", Yüksek Lisans Tezi, *Gebze Yüksek Teknoloji Enstitüsü*, Gebze , 8-15 (1998).
3. Özçilingir, M., "An Ottoman Character Recognizer", Yüksek Lisans Tezi, *ODTU Fen Bilimleri Enstitüsü*, Ankara, 1-42 (1990).
4. Atıcı, A., "Segmentation, Feature Extraction and Recognition of Ottoman Script", Yüksek Lisans Tezi, *ODTU Fen Bilimleri Enstitüsü*, Ankara, 90-98 (1994).
5. Amin, A., Al-Sadoun, H., "Handprinted Arabic Character Recognition System", *Proc. 12th Int. Conf. on Pattern Recognition*, Japan, 536-539 (1994).
6. Amin, A., Mansoon, W., "Recognition of Printed Arabic Text Using Neural Networks", *Proc. 4th Int. Conf. on Document Analysis Recognition*, Germany, (1997).
7. Nabiyev, V.V., "Yapay Zeka", *Seçkin Yayıncılık*, Ankara, 534-570 (2005).
8. Amin, A., Kaced, A., Haton, J., Moher, R., "Handwritten Arabic Character Recognition by the I.R.A.C system", *Proc. 5th Int. Joint Conf. Pattern Recognition*, Germany, 729-731 (1980).
9. Plamondon, R., Srihari, S.N., "Online and off-line handwriting recognition: a comprehensive survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1): 63-84 (2000).
10. Duygulu, P., "El Yazısı Tanıma", *Türkiye Bilişim Ansiklopedisi*, Ören, T., Çölkesen R., *Papatya Yayıncılık*, İstanbul, 369-372 (2006).
11. Aizawa, H., Wakahara, T., Odaka, K., "Realtime handwritten character string segmentation using multiple stroke features", *IEICE Trans. Inf. & Syst*, 80(5): 1178-1185 (1997).
12. Okamoto, M., Yamamoto, K., "On-line handwriting character recognition method with directional features and direction-change features", *Fourth International Conference Document Analysis and Recognition (ICDAR'97)*, USA, 926 (1997).

13. Blumenstein, M., Liu, X.Y., Verma, B., “An investigation of the modified direction feature for cursive character recognition”, *Pattern Recognition*, 40(2): 376-388 (2007).

EKLER

EK-1 Program Kodları

{Program kodlarının önemli adımları aralarda açıklanmıştır}

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, ComCtrls, OleServer, Word2000;

type

TForm1 = class(TForm)

 memyol: TMemo;

 Button1: TButton;

 edtyon: TEdit;

 Label1: TLabel;

 Label2: TLabel;

 edtharf: TEdit;

 lstharf: TListBox;

 Edit1: TEdit;

 btnkapat: TButton;

 Button3: TButton;

 lsteskiharf: TListBox;

 lstenby: TListBox;

 lstenky: TListBox;

 lstnoktaliharf: TListBox;

 Button2: TButton;

 Button4: TButton;

 osmeditor: TRichEdit;

 cmbyazitipi: TComboBox;

```
cmbboyut: TComboBox;  
osmharf: TRichEdit;  
Isttumharf: TListBox;  
Label3: TLabel;  
Label4: TLabel;  
radioaktarma: TRadioGroup;  
Button5: TButton;  
Button6: TButton;  
Istkef: TListBox;
```

```
procedure FormMouseDown(Sender: TObject; Button: TMouseButton;  
    Shift: TShiftState; X, Y: Integer);  
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,  
    Y: Integer);  
procedure FormMouseUp(Sender: TObject; Button: TMouseButton;  
    Shift: TShiftState; X, Y: Integer);  
procedure Button1Click(Sender: TObject);  
procedure FormClose(Sender: TObject; var Action: TCloseAction);  
procedure btnkapatClick(Sender: TObject);  
procedure FormCreate(Sender: TObject);  
procedure FormPaint(Sender: TObject);  
procedure Button3Click(Sender: TObject);  
procedure cmbyazitipiChange(Sender: TObject);  
procedure cmbboyutChange(Sender: TObject);  
procedure Button2Click(Sender: TObject);  
procedure Button4Click(Sender: TObject);  
procedure IsttumharfClick(Sender: TObject);  
procedure Button5Click(Sender: TObject);  
procedure Button6Click(Sender: TObject);
```

```
private  
    { Private declarations }
```

```
public
  { Public declarations }
end;

//harfkume=Set of string;
var
  // harfyol:harfkume;
  Form1: TForm1;
  basili,xdegisti,ydegisti:boolean;
  yolx:array[0..100000] of integer;
  yoly:array[0..100000] of integer;
  yon,sonucadet,xyon,yyon,adet,aralik,fark,i,enbuyuky,enkucuky,noktasay:integer;
  harf,sonuc:string;
  yer:integer;

const
  x1:integer=50;
  y1:integer=50;
  x2:integer=210;
  y2:integer=160;
  xy:integer=3;

implementation

{$R *.dfm}

//MOUSA TIKLANINCA AKTİF OLAN PROSEDÜR
procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
```

```

//GİDİLEN YOL BİLGİSİ SIFIRLANIYOR
for i:=0 to 100000 do
begin
  yolx[i]:=0;
  yoly[i]:=0;
end;
//*****

//DEĞER ATAMASI YAPILYOR
adet:=0;
aralik:=15;
fark:=7;
sonucadet:=0;
sonuc:="";
edtyon.Text:="";
canvas.Pen.Width:=6;
//*****

//HARFİN KUTU İÇİNE ÇİZİLMESİ SAĞLANIYOR
if (x>x1+xy)and(x<x2-xy)and (y>y1+xy)and (y<y2-xy)then
begin
  basili:=true;
  yolx[adet]:=x;
  yoly[adet]:=y;
  adet:=adet+1;
  canvas.MoveTo(x,y);
  canvas.LineTo(x,y);
end;
//*****
end;

```

```

//MOUSE HAREKET EDERKEN AKTİF OLAN PROSEDÜR
procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
if basili and (x>x1+xy)and(x<x2-xy)and (y>y1+xy)and (y<y2-xy)then
begin
  xyon:=0;
  yyon:=0;
  yon:=0;
  xdegisti:=false;
  ydegisti:=false;
  canvas.LineTo(x,y);
  yolx[adet]:=x;
  yoly[adet]:=y;
  adet:=adet+1;

//MOUSUN TAKİP ETTİĞİ YÖNLER HESAPLANIYOR

  if adet>aralik then
    if xyon=0 then
      if x-yolx[adet-aralik]>fark then
        begin
          xyon:=3;
          xdegisti:=true;
        end
      else if yolx[adet-aralik]-x>fark then
        begin
          xyon:=4;
          xdegisti:=true;
        end;

    if adet>aralik then

```



```

if yyon=0 then
  if y-yoly[adet-aralik]>fark then
    begin
      yyon:=2;
      ydegisti:=true;
    end
  else if yoly[adet-aralik]-y>fark then
    begin
      yyon:=1;
      ydegisti:=true;
    end;
end;

```

```

if (xdegisti=true) and (ydegisti=false) then
  yon:=xyon
else if (xdegisti=false) and (ydegisti=true) then
  yon:=yyon
else if xdegisti and ydegisti then
  if (xyon=3) and (yyon=1) then
    yon:=5
  else if (xyon=3) and (yyon=2) then
    yon:=6
  else if (xyon=4) and (yyon=2) then
    yon:=7
  else if (xyon=4) and (yyon=1) then
    yon:=8;
end;

```

```

if (yon<>0) and (copy(sonuc,sonucadet,1)<>inttostr(yon))then
  begin
    sonuc:=sonuc+inttostr(yon);
    sonucadet:=sonucadet+1;
  end;
end;

```

```

if sonucadet>3 then
  if copy(sonuc,sonucadet-1,2)=copy(sonuc,sonucadet-3,2) then
    begin
      sonucadet:=sonucadet-2;
      delete(sonuc,sonucadet-1,2);
    end;
  edtyon.Text:=sonuc;
end;
end;

procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
var enby,enky: array[1..4] of integer;
begin

  //ÇİZİLEN ŞEKLİN ALT VE ÜST NOKTALARI HESAPLANIYOR (SON DÖRT
  ŞEKLE KADAR)

  enbuyuky:=0;
  enkucuky:=y2;
  for i:=0 to 10000 do
    if yoly[i]<>0 then
      begin
        if yoly[i]>enbuyuky then enbuyuky:=yoly[i];
        if yoly[i]<enkucuky then enkucuky:=yoly[i];
      end;

  lstenby.Items.add(inttostr(enbuyuky));
  lstenky.Items.Add(inttostr(enkucuky));

  if lstenky.Count>4 then lstenky.Items.Delete(0);
  if lstenby.Count>4 then lstenby.Items.Delete(0);

```

```

if lstenby.Count>0 then
begin
  enby[1]:=strtoint(lstenby.Items[0]);
  enky[1]:=strtoint(lstenky.Items[0]);
end;
if lstenby.count>1 then
begin
  enby[2]:=strtoint(lstenby.Items[1]);
  enky[2]:=strtoint(lstenky.Items[1]);
end;
if lstenby.count>2 then
begin
  enby[3]:=strtoint(lstenby.Items[2]);
  enky[3]:=strtoint(lstenky.Items[2]);
end;
if lstenby.count>3 then
begin
  enby[4]:=strtoint(lstenby.Items[3]);
  enky[4]:=strtoint(lstenky.Items[3]);
end;
//*****

harf:='Tanımsız';
basili:=false;
for i:=0 to 100000 do
if yolx[i]<>0 then
begin
  memyol.LineS.Add(inttostr(i+1)+'!'+inttostr(yolx[i])+'!'+inttostr(yoly[i]));
end;

if sonucadet<>0 then

```

```

begin
  insert('#',sonuc,1);
  insert('#',sonuc,sonucadet+2);
end;

if pos(sonuc,lstharf.items[1])<>0 then harf:='ELİF'
else if pos(sonuc,lstharf.items[2])<>0 then
begin
  if (yoly[adet-1]-yoly[0])<20 then harf:='BE'
  else harf:='LAM'
end
else if pos(sonuc,lstharf.items[3])<>0 then harf:='PE'
else if pos(sonuc,lstharf.items[4])<>0 then harf:='TE'
else if pos(sonuc,lstharf.items[5])<>0 then harf:='SE'
else if pos(sonuc,lstharf.items[6])<>0 then harf:='CİM'
else if pos(sonuc,lstharf.items[7])<>0 then harf:='ÇİM'
else if pos(sonuc,lstharf.items[8])<>0 then harf:='HA'
else if pos(sonuc,lstharf.items[9])<>0 then harf:='HI'
else if pos(sonuc,lstharf.items[10])<>0 then harf:='DAL'
else if pos(sonuc,lstharf.items[11])<>0 then harf:='ZEL'
else if pos(sonuc,lstharf.items[12])<>0 then harf:='RE'
else if pos(sonuc,lstharf.items[13])<>0 then harf:='ZE'
else if pos(sonuc,lstharf.items[14])<>0 then harf:='JE'
else if pos(sonuc,lstharf.items[15])<>0 then harf:='SİN'
else if pos(sonuc,lstharf.items[16])<>0 then harf:='ŞİN'
else if pos(sonuc,lstharf.items[17])<>0 then harf:='SAD'
else if pos(sonuc,lstharf.items[18])<>0 then harf:='DAT'
else if pos(sonuc,lstharf.items[19])<>0 then harf:='TI'
else if pos(sonuc,lstharf.items[20])<>0 then harf:='ZI'
else if pos(sonuc,lstharf.items[21])<>0 then harf:='AYIN'
else if pos(sonuc,lstharf.items[22])<>0 then harf:='ĞAYIN'
else if pos(sonuc,lstharf.items[23])<>0 then harf:='FE-KAF'

```

```

else if pos(sonuc,lstharf.items[24])<>0 then harf:='KAF'
else if pos(sonuc,lstharf.items[25])<>0 then harf:='KEF'
else if pos(sonuc,lstharf.items[26])<>0 then harf:='GEF'
else if pos(sonuc,lstharf.items[27])<>0 then harf:='NEF'
else if pos(sonuc,lstharf.items[28])<>0 then harf:='LAM'
else if pos(sonuc,lstharf.items[29])<>0 then harf:='MİM'
else if pos(sonuc,lstharf.items[30])<>0 then harf:='NUN'
else if pos(sonuc,lstharf.items[31])<>0 then harf:='VAV'
else if pos(sonuc,lstharf.items[32])<>0 then harf:='HE'
else if pos(sonuc,lstharf.items[33])<>0 then harf:='LAMELİF'
else if pos(sonuc,lstharf.items[34])<>0 then harf:='YE'
else if pos(sonuc,lstharf.items[35])<>0 then harf:='HEMZE'

```

```

else if (adet>0) and (adet<6)then harf:='Nokta';
lsteskiharf.Items.add(harf);
if lsteskiharf.Count>4 then lsteskiharf.Items.Delete(0);
noktasay:=0;
for i:=lsteskiharf.count-1 downto 0 do
  if lsteskiharf.items[i]='Nokta' then
    noktasay:=noktasay+1
  else break;
//showmessage(inttostr(noktasay));

```

```
//NOKTALILAR
```

```
if (noktasay=0) and (lstnoktaliharf.Items.IndexOf(harf)>=0) then harf:='Noktalı';
```

```
//TEK NOKTALILAR BULUNUYOR
```

```
if (noktasay=1) and (lsteskiharf.Items.Count>1) then
```

```
begin
```

```
  if (lsteskiharf.Items[lsteskiharf.items.Count-2]='Tanımsız') then harf:='Tanımsız';
```

```
  if (lsteskiharf.Items[lsteskiharf.items.Count-2]='BE')
```

```
    and (enby[lstenky.Count-1]<(enky[lstenky.Count])) then harf:='BE';
```

```
  if (lsteskiharf.Items[lsteskiharf.items.Count-2]='HA')
```

```

    and(enky[lstenky.Count-1]>(enky[lstenky.Count]))then harf:='HI'
else if (lsteskiharf.Items[lsteskiharf.items.Count-2]='HA')
    and (enky[lstenky.Count-1]<(enky[lstenky.Count]))
    and (enby[lstenby.Count-1]>(enky[lstenby.count]))then harf:='CİM';
if (lsteskiharf.Items[lsteskiharf.items.Count-2]='DAL')
    and (enky[lstenky.Count-1]>(enky[lstenky.Count])) then harf:='ZEL';
if (lsteskiharf.Items[lsteskiharf.items.Count-2]='RE')
    and (enky[lstenky.Count-1]>(enky[lstenky.Count])) then harf:='ZE';
if (lsteskiharf.Items[lsteskiharf.items.Count-2]='SAD')
    and (enky[lstenky.Count-1]>(enky[lstenky.Count])) then harf:='DAT';
if (lsteskiharf.Items[lsteskiharf.items.Count-2]='TI')
    and (enky[lstenky.Count-1]<(enky[lstenky.Count]))
    and (enby[lstenby.Count-1]>(enky[lstenby.count]))then harf:='ZI';
if (lsteskiharf.Items[lsteskiharf.items.Count-2]='AYIN')
    and (enky[lstenky.Count-1]>(enky[lstenky.Count])) then harf:='ĞAYIN';
if (lsteskiharf.Items[lsteskiharf.items.Count-2]='FE-KAF')
    and (enky[lstenky.Count-1]>(enky[lstenky.Count])) then harf:='FE';
if (lsteskiharf.Items[lsteskiharf.items.Count-2]='NUN')
    and(enky[lstenky.Count-1]>(enky[lstenky.Count])) then harf:='NUN';
end;

```

//İKİ NOKTALILAR BULUNUYOR

```

if (noktasay=2)and (lsteskiharf.Items.Count>2) then
begin
if (lsteskiharf.Items[lsteskiharf.items.Count-3]='BE')
    and (enky[lstenky.Count-2]>(enky[lstenky.Count-1]))
    and (enky[lstenky.Count-2]>(enky[lstenky.Count])) then harf:='TE';
if (lsteskiharf.Items[lsteskiharf.items.Count-3]='FE-KAF')
    and (enky[lstenky.Count-2]>(enky[lstenky.Count-1]))
    and (enky[lstenky.Count-2]>(enky[lstenky.Count])) then harf:='KAF';
end;

```

```

//ÜÇ NOKTALILAR BULUNUYOR
if (noktasay=3)and (Isteskiharf.Items.Count>3) then
begin
  if (Isteskiharf.Items[Isteskiharf.items.Count-4]='BE')
    and (enby[Istenky.Count-3]<(enky[Istenky.Count-2]))
    and (enby[Istenky.Count-3]<(enky[Istenky.Count-1]))
    and (enby[Istenky.Count-3]<(enky[Istenky.Count])) then harf:='PE';
  if (Isteskiharf.Items[Isteskiharf.items.Count-4]='BE')
    and (enky[Istenky.Count-3]>(enky[Istenky.Count-2]))
    and (enky[Istenky.Count-3]>(enky[Istenky.Count-1]))
    and (enky[Istenky.Count-3]>(enky[Istenky.Count])) then harf:='SE';
  if (Isteskiharf.Items[Isteskiharf.items.Count-4]='HA')
    and (enky[Istenky.Count-3]<(enky[Istenky.Count-2]))and (enby[Istenby.Count-
3]>(enky[Istenby.count-2]))
    and (enky[Istenky.Count-3]<(enky[Istenky.Count-1]))and (enby[Istenby.Count-
3]>(enky[Istenby.count-1]))
    and (enky[Istenky.Count-3]<(enky[Istenky.Count]))and (enby[Istenby.Count-
3]>(enky[Istenby.count]))
    then harf:='ÇİM';
  if (Isteskiharf.Items[Isteskiharf.items.Count-4]='RE')
    and (enky[Istenky.Count-3]>(enky[Istenky.Count-2]))
    and (enky[Istenky.Count-3]>(enky[Istenky.Count-1]))
    and (enky[Istenky.Count-3]>(enky[Istenky.Count])) then harf:='JE';
  if (Isteskiharf.Items[Isteskiharf.items.Count-4]='SİN')
    and (enky[Istenky.Count-3]>(enky[Istenky.Count-2]))
    and (enky[Istenky.Count-3]>(enky[Istenky.Count-1]))
    and (enky[Istenky.Count-3]>(enky[Istenky.Count])) then harf:='ŞİN';
  if (Isteskiharf.Items[Isteskiharf.items.Count-4]='KEF')
    and (enby[Istenky.Count-3]>(enby[Istenky.Count-2]))
    and (enby[Istenky.Count-3]>(enby[Istenky.Count-1]))
    and (enby[Istenky.Count-3]>(enby[Istenky.Count])) then harf:='NEF';

```

```
end;

//SONUÇ GÖSTERİLİYOR
if harf<>'Nokta' then
begin
  edtharf.Text:=harf;
  if (radioaktarma.ItemIndex=0)and (harf<>'Tanımsız') and (harf<>'Noktalı')then
Button2.OnClick(Self);
  end
  else edtharf.Text:='Tanımsız' ;

end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  memyol.Lines.Clear;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  lstharf.Items.SaveToFile('harfler.txt');
  AddFontResource('osmanlica.TTF');
  SendMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0);
end;

procedure TForm1.btnkapatClick(Sender: TObject);
begin
  form1.Close;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  lstharf.Items.LoadFromFile('harfler.txt');
  AddFontResource('osmanlica.TTF');
```



```

SendMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0);
cmbyazitipi.Items:=screen.Fonts;
end;

procedure TForm1.FormPaint(Sender: TObject);
begin
  canvas.Pen.Style:=psSolid;
  canvas.Pen.Width:=5;
  form1.canvas.Rectangle(x1,y1,x2,y2);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  form1.FormPaint(self);
end;

procedure TForm1.cmbyazitipiChange(Sender: TObject);
begin
  osmeditor.font.Name:=cmbyazitipi.Items[cmbyazitipi.itemindex];
end;

procedure TForm1.cmbboyutChange(Sender: TObject);
begin
  osmeditor.Font.Size:=strtoint(cmbboyut.Items[cmbboyut.itemindex]);
end;

procedure TForm1.Button2Click(Sender: TObject);
var sira:integer;
begin
  if (edtharf.text<>'Tanımsız')and(edtharf.text<>'Noktalı')and((edtharf.text<>'')) then
  begin
    sira:=Isttumharf.Items.IndexOf(harf);
    if edtharf.Text='LAMELİF' then
    begin

```

```

osmharf.Lines.LoadFromFile('harf/LAM.rtf');
osmharf.SelStart:=0;
osmharf.SelLength:=1;
osmharf.Font.Size:=30;
osmharf.CopyToClipboard;
osmeditor.PasteFromClipboard;

osmharf.Lines.LoadFromFile('harf/ELİF.rtf');
osmharf.SelStart:=0;
osmharf.SelLength:=1;
osmharf.Font.Size:=30;
osmharf.CopyToClipboard;
osmeditor.PasteFromClipboard;
end
else
begin
osmharf.Lines.LoadFromFile('harf/'+Isttumharf.items[sira]+'rtf');
osmharf.SelStart:=0;
osmharf.SelLength:=1;
osmharf.Font.Size:=30;
osmharf.CopyToClipboard;
osmeditor.PasteFromClipboard;
end;
osmeditor.Font.Size:=30;
end;
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
if harf="" then showmessage('İlk önce harfinizi çiziniz')
else if harf='Tanımsız' then showmessage('Bu çizimin hangi harf olarak tanınmasını istiyorsanız listeden seçiniz')

```

```

end;
procedure TForm1.IsttumharfClick(Sender: TObject);
begin
if edtyon.Text<>" then
begin
  if pos('#',Istharf.Items[Isttumharf.itemindex])<>0 then
Istharf.Items[Isttumharf.itemindex]:=Istharf.Items[Isttumharf.itemindex]+edtyon.Text+'#'
  else
Istharf.Items[Isttumharf.itemindex]:=Istharf.Items[Isttumharf.itemindex]+'#'+edtyon.Text+'#';
  showmessage('Çizdiğiniz harf '+Isttumharf.Items[Isttumharf.itemindex]+' harfine eklendi');end;
  edit1.Text:=Istharf.Items[Isttumharf.itemindex];end;
procedure TForm1.Button5Click(Sender: TObject);
begin
  osmharf.Lines.LoadFromFile('harf/bosluk.rtf);
  osmharf.SelStart:=0;
  osmharf.SelLength:=1;
  osmharf.CopyToClipboard;
  osmeditor.SelStart:=0;
  osmeditor.PasteFromClipboard;
  osmeditor.SelStart:=0;
end;
procedure TForm1.Button6Click(Sender: TObject);
begin
  osmeditor.Clear;
end;

end.

```

Ek-2 Harf özellikleri tablosunun metin dosyası içeriği (harfler.txt)

HARFLER

ELİF:#1#2#

BE:#247481#241#2741#27481#274841#2748481#2481#2748#7481#

PE::#247481#241#2741#27481#274841#2748481#2481#2748#7481#

TE::#247481#241#2741#27481#274841#2748481#2481#2748#7481#

SE::#247481#241#2741#27481#274841#2748481#2481#2748#7481#

CİM::#347263#34723#37263#327263#3472635#372635#327263#347263535#6347
2635#627263#36272635#3272635#6347263#ÇİM::#347263#34723#37263#327263#3472635#372635#327263#347263535#6347
2635#627263#36272635#3272635#6347263#HA:#347263#34723#37263#327263#3472635#372635#327263#347263535#634726
35#627263#36272635#3272635#6347263#HI::#347263#34723#37263#327263#3472635#372635#327263#347263535#634726
35#627263#36272635#3272635#6347263#

DAL:#624#6274#

ZEL::#624#6274#

RE:#74#274#27#

ZE:74#274#27#

JE:74#274#27#

SİN:#24842427484#24812427481#2424827481#248127481#248248127481#24274
81#248124127481#274812748127481#748412748127481#274127427481#2481274
8127481#2481248427481#7481274812748#74812748127481#27481248127481#27
4827427481#242427481#27412748127481#ŞİN:#24842427484#24812427481#2424827481#248127481#248248127481#24274
81#248124127481#274812748127481#748412748127481#274127427481#2481274
8127481#2481248427481#7481274812748#74812748127481#27481248127481#27
4827427481#242427481#27412748127481#SAD:#536274241#5362742627481#5327427481#53627427481#153627427481#53
62742748#536274727481#3627472748#5362748427481#536274842627481#

DAT:#536274241#5362742627481#5327427481#53627427481#153627427481#53
 62742748#536274727481#3627472748#5362748427481#536274842627481#
 TI:#2153274#253274#21536274#2536274#2153627#253627#2353274#23536274#2
 723536274#723536274#263536274#636274#
 ZI:#2153274#253274#21536274#2536274#2153627#253627#2353274#23536274#2
 723536274#723536274#263536274#636274#
 AYIN:#726347263#72635347263#7263472635#726353472635#72635472635#726
 372635#7263272635#47263272635#4723472635#4726353472635#263472635#426
 347263#47263472635#4726372635#
 ĞAYIN: #726347263#72635347263#7263472635#726353472635#72635472635#7
 26372635#7263272635#47263272635#4723472635#4726353472635#263472635#4
 26347263#47263472635#4726372635#
 FE:#4813627481#8153627481#48153627481#815362748#4153274815#481536274
 815#4153627481#4136274848481#413627481#
 KAF:#4813627481#8153627481#48153627481#815362748#4153274815#4815362
 74815#4153627481#4136274848481#413627481#
 KEF:#723627481#72362748#7236274#72636274#726274#726362748#7262748#72
 624#723634#7264#72634#
 GEF:#723627481#72362748#7236274#72636274#726274#726362748#7262748#72
 624#723634#7264#72634#
 NEF:#723627481#72362748#7236274#72636274#726274#726362748#7262748#72
 624#723634#7264#72634#
 LAM:#247481#241#2741#27481#274841#2748481#2481#2748#7481#
 MĪM:#5327472#1532742#1327472#132742#132472#1362742#153627472#153247
 2#1362472#536242#5362472#536248472#53627472#15362742#
 NUN:#627481#62727481#
 VAV:#81536274#41536274#481536274#48153274#4815362747#48153627#
 HE:#472635147#4726351847#726351847#72635184727#47263518472#726351847
 2#472636363518472#
 LAMELĪF:#72635181#723518#7263184#726318#74318#7263518#72635184#7435
 18#

YE:#43274#42632741#426327481#42627481#4262748181#4263274815#84726362
7481#42627484#84726274815#1842627481#8472626274815#8472627481#184727
48481#842627481#184727481#84727481#8427481#
HEMZE:#7263547#4726357#84726347#72635347#726347#72637#4726347#72635
7#7263537#726327#72635327#

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : EROĞLU, Yılmaz
 Uyruğu : T.C.
 Doğum tarihi ve yeri : 19.01.1981 Bartın
 Medeni hali : Evli
 Telefon : 0 (505) 254 32 33
 e-mail : yilmazeroglu@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi, Bilgisayar Eğitimi Bölümü	2003
Lise	Anadolu Öğretmen Lisesi	1999

İş Deneyimi

Yıl	Yer	Görev
2005-2007	Gerede Anadolu İHL	Bilgisayar Öğretmeni

Bulunduğu Görevler

Yıl	Yer	Görev
2006-2007	Abant İzzet Baysal Üniversitesi Gerede Meslek Yüksek Okulu	Bilgisayar Dersi Öğretim Elemanı

Verdiği Kurslar

Yıl	Yer	Kurs Adı
2006-2007	Gerede Halk Eğitim Merkezi	Bilgisayar Operatörlük ve İşletmenlik Kursu

Yabancı Dil

İngilizce

Hobiler

Bilgisayar teknolojileri, kitap okuma