



T.C.
MALTEPE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ÖĞRENME GÜÇLÜĞÜ ÇEKEN ÇOCUKLAR İÇİN EL YAZISI
TANIMA İLE ÖĞRENMEYİ KOLAYLAŞTIRICI BİR MOBİL
ÖĞRENME UYGULAMASI TASARIMI**

BANU YILMAZ

Yüksek Lisans Tezi

Tez Danışmanı

Yrd. Doç. Dr. Turgay Tugay Bilgin

İSTANBUL – 2014

**T.C.
MALTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**ÖĞRENME GÜÇLÜĞÜ ÇEKEN ÇOCUKLAR İÇİN EL YAZISI
TANIMA İLE ÖĞRENMEYİ KOLAYLAŞTIRICI BİR MOBİL
ÖĞRENME UYGULAMASI TASARIMI**

YÜKSEK LİSANS TEZİ

BANU YILMAZ

**Tez Danışmanı
Yrd. Doç. Dr. Turgay Tugay Bilgin**

İSTANBUL – 2014

ÖZET

Yüksek Lisans Tezi, Öğrenme Güçlüğü Çeken Çocuklar İçin El Yazısı Tanıma İle Öğrenmeyi Kolaylaştırıcı Bir Mobil Öğrenme Uygulaması Tasarımı, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.

Bu tez çalışması, okuma zorluğu çeken çocuklar için kullanılan Kelime Kutusu Stratejisi'ne göre mobil cihazlar üzerinde çalışabilmeye olanak sağlayan bir uygulamayı ve uygulama içerisinde el yazısı tanımayı içermektedir.

Bu tez toplam 5 bölümden oluşmaktadır. Birinci bölümde tezin amacından, kapsamından ve konuyla ilgili daha önce yapılan çalışmalardan bahsedilmiştir. İkinci bölümde öğrenme zorluğu için kullanılan yöntemlerden ve günümüz teknolojisiyle ilişkisinden bahsedilmiştir. Üçüncü bölümde mobil öğrenme ve e-öğrenme kavramlarından, tasarımlarından ve mobil öğrenme içeriklerinden bahsedilmiştir. Dördüncü bölümde el yazısı tanıması, tanıma yöntemleri ve kullanım alanları anlatılmıştır. Son bölümde uygulama altyapısından, içeriğinden, mimarisinden ve uygulama adımlarından, el yazısı tanıma için kullanılan Tesseract'ın eğitiminden bahsedilmiştir.

Bu tez 2014 yılında tamamlanmıştır ve 84 sayfadan oluşmaktadır.

Anahtar Kelimeler: mobil öğrenme, el yazısı tanıma, örüntü tanıma, öğrenme zorluğu, okuma zorluğu

ABSTRACT

Master's Thesis, Design of A Mobile Device Application with Handwriting Recognition to Make Learning Easy For Students Who Have Learning Disabilities, T.C Maltepe University, Institute of Sciences, Department of Computer Engineering.

This master thesis includes handwriting recognition and an application which runs on mobile devices by using Word Box Strategy for students who have learning disabilities.

This thesis consists of 5 sections. Earlier studies about the subject, the purpose and scope of the thesis have been presented in the first section. In the second section, definition of reading disability, used methods for the disability and relations up to date technologies have been mentioned. In the third section, concept and design of mobile learning and e-learning, mobile learning content have been mentioned. In the fourth section, recognizing of handwriting, the way of recognizing of processes and its areas of use have been presented. In the last section; application's infrastructure, content, architecture, implementation of stages and training of Tesseract which is used for recognition have been discussed.

This thesis has been completed in 2014 and consists of 84 pages.

Keywords: Mobile learning, handwriting recognition, pattern recognition, learning disability, reading disability

TEŐEKKÜR

Tez konumu seımeme yardımcı olan, alıőmaya teővik eden, bu srete yol gstericilięini ve bilgisini benden esirgemeyen deęerli danıőman hocam Yrd. Do. Dr. Turgay Tugay BİLGİN'e, sevgisi ve ilgisiyle attıęım her adımda arkamda olan aileme anlayıőları iin teőekkr ederim.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
ŞEKİLLER.....	vi
ÇİZELGELER.....	viii
KISALTMALAR	ix
1. GİRİŞ	1
1.1 Tezin Amacı	1
1.2 Tezin Kapsamı.....	1
1.3 Benzer Çalışmalar	3
2. PROBLEMİN TANIMI	8
2.1. Okuma Zorluğu Nedir?	8
2.2. Öğrenme Zorluğunu Aşmak İçin Kullanılan Yöntemler.....	9
2.3. Öğrenme Zorluğu ve Günümüz Teknolojisi	10
3. MOBİL ÖĞRENME	12
3.1 Mobil Öğrenme Kavramı	12
3.2 E-Öğrenme ve Mobil Öğrenme Farkları	15
3.3 Mobil Öğrenme Sistemleri Tasarımı.....	18
3.4 Mobil Öğrenme Sistemleri İçeriği.....	22
3.4.1 E-Öğrenme, Mobil Öğrenme Standartları.....	22
3.4.2 Öğrenim Materyalleri.....	24
4 EL YAZISI TANIMA.....	26
4.1 El Yazısı Tanıma	26
4.1.1 Çevrimiçi Yazı Tanıma.....	27
4.1.2 Çevrimdışı Yazı Tanıma	31
4.2 El Yazısı Tanımının Kullanıldığı Alanlar.....	44

5	ÖĞRENME ZORLUĞU ÇEKENLER İÇİN MOBİL UYGULAMA GELİŞTİRİLMESİ.....	45
5.1	Uygulama Altyapısı.....	45
5.1.1	Android İşletim Sistemi	45
5.1.2	SVG Dosya Biçimi ve Kullanımı.....	47
5.1.3	Tesseract OCR Yazılımı	51
5.2	Uygulamanın İçeriği.....	57
5.3	Sistem Mimarisi ve Uygulamanın Adımları	58
5.4	Kullanıcı Arayüzü	61
5.5	Eğitim Seti Hazırlama	63
5.5.1	Box dosyası oluşturma	64
5.5.2	Yeni Karakter Seti Yükleme	67
5.5.3	Tesseract'ı Eğitme.....	69
5.5.4	Karakter Setini Sayma.....	70
5.5.5	Font Özellikleri	72
5.5.6	Kümeleme	73
5.5.7	Sözlük Verisi.....	74
5.5.8	Belirsiz Karakterler	75
5.5.9	Bir Araya Toplama.....	77
6	SONUÇLAR VE DEĞERLENDİRME.....	78
6.1	Elde Edilen Sonuçlar	78
6.2	Değerlendirme	78
6.3	Öneriler.....	79
	KAYNAKLAR	80
	ÖZGEÇMİŞ	84

ŞEKİLLER

Şekil 1-1.Tezin kapsamı.....	3
Şekil 1-2.IMLIS'ın mimarisi.....	5
Şekil 1-3.Örnek uygulamanın eğitim ekranı	6
Şekil 1-4.Örnek uygulamanın yazma ekranı	7
Şekil 2-1.CellWriter arayüzü	11
Şekil 2-2.CellWriter eğitim ekranı	11
Şekil 3-1.E-öğrenme, çevrimiçi öğrenme ve mobil öğrenme arasındaki ilişki	18
Şekil 3-2.Mobil öğrenme sisteminde öğrenci merkezli tasarım	19
Şekil 3-3.Öğrenim nesnelerinin yaşam döngüsü	25
Şekil 4-1.Çevrimiçi tanımda analizi yapılmış yazı örneği	28
Şekil 4-2.Dynabook'un illüstrasyonu	31
Şekil 4-3.Piksel tabanlı çevrimdışı el yazısı tanıma	32
Şekil 4-4.Eşikleme ile arka plan renginin değiştirilmesi	33
Şekil 4-5.Gürültüsü azaltılmış metin	34
Şekil 4-6.El yazısında eğrilik ve eğim	35
Şekil 4-7.Karakter gövdesi bulma	36
Şekil 4-8.Histograma göre eşik bulma	37
Şekil 4-9.MR ve SH parametrelerinin kullanıldığı bir el yazısı örneği	37
Şekil 4-10.Bölgeleme ile karakter tanıma	40
Şekil 4-11.Piksellerdeki yoğunluk	40
Şekil 4-12.Karakterin izdüşümü	41
Şekil 4-13.Karakter profili	41
Şekil 4-14.Karakterin iç ve dış noktalarının vurgulanması.....	41
Şekil 4-15.Karakterlerin kesişme ve uzaklık özellikleri	42
Şekil 5-1.Android versiyonlarının kullanım grafiği.....	47
Şekil 5-2.SVG ile JPEG/GIF arasındaki fark	48
Şekil 5-3.SVG formatının diğer formatlardan farkı	49
Şekil 5-4.HTML içerisine gömülecek SVG kodu ve tarayıcıda görünen şekil	49

Şekil 5-5.<polyline> elementi ile SVG dosyası kodu.....	51
Şekil 5-6.Sınırları belirlenmiş bir metin örneği	53
Şekil 5-7.Metnin sabit aralıklarından faydalanılarak bölünen kelime	54
Şekil 5-8.Aralıkları orantısız metin örneği	54
Şekil 5-9.Aday bölme noktaları	55
Şekil 5-10.Tanımlanabilecek örnek metin	56
Şekil 5-11.Farklı yazılarla tanımlanan karakterler	56
Şekil 5-12.Uygulama üzerinde el yazısı denemesi	59
Şekil 5-13.Yazılan “A” harfinin SVG kodu.....	60
Şekil 5-14.Sistem mimarisi	61
Şekil 5-15.Uygulama arayüzü.....	62
Şekil 5-16.Eğitim seti için veri toplama arayüzü.....	63
Şekil 5-17.box dosyası oluşturan QT Box Editor arayüzü.....	69

ÇİZELGELER

Çizelge 3-1.Toplam internet abone sayıları	12
Çizelge 3-2.Esnek Öğrenme Modellerinin gelişimi.....	14
Çizelge 5-1.Android işletim sistemi versiyonları ve kullanım yüzdeleri.....	46
Çizelge 5-3.Eğitim seti için karakter özellikleri	65
Çizelge 5-4.Düzenlenen karakter seti	66
Çizelge 5-5.Karakter setlerinin tekrar düzenlenmesi.....	67
Çizelge 5-6.unicharset içeriği	71
Çizelge 5-7.DAWG sözlük dosyası için kullanılan kelimeler	74
Çizelge 5-8.unicharambigs dosyasının içeriği.....	75
Çizelge 5-9.Tip belirleyicisinin değerleri	76

KISALTMALAR

Kısaltma	İngilizcesi	Türkçesi
BSM	Bosinovic and Srihari Method	Bosinovic ve Srihari Metodu
CORDRA	Content Object Repository Discovery □And Registration/Resolution Architecture	İçerik Nesne Deposu Keşfi ve Kayıt Mimarisi
CSR	Cursive Script Recognition	Bitişik El Yazısı Tanıma
DAWG	Directed Acyclic Word Graph	Direk Döngüsüz Kelime Grafığı
DP	Dynamic Programming	Dinamik Programlama
FIRT	The Fuzzy Item Response Theory	Bulanık Öğelerin Tepki Kuramı
HMM	Hidden Markov Models	Saklı Markov Model
ICR	Intelligence Character Recognition	Akıllı Karakter Tanıma
IMLIS	Intelligent Mobile□Learning Interaction System	Akıllı Mobil Öğrenme Etkileşim Sistemi
IMS	International Management System	Uluslararası Yönetim Sistemi
JDK	Java Development Kit	Java Geliştirme Aracı
LOR	Learning Object Repository	Öğrenim Nesne Deposu
LTSC	Learning Technology Standards Committee	IEEE Öğrenme Teknoloji Standartları Komitesi
MR	Max Run	Maksimum Eğilim
MUI	Mobile User Interface	Mobil Kullanıcı Arayüzü
OCR	Optical Character Recognition	Optik Karakter Tanıma

PIMS	Personalized Intelligent Mobile Learning System for Supporting Effective English Learning	Etkili İngilizce Öğrenmeyi Desteklemek için Kişiselleştirilmiş Akıllı Mobil Öğrenme Sistemi
SCORM	Sharable Content Object Reference Model	Paylaşılabilir İçerikli Nesne Referans Modeli
SDK	Software Development Kit	Yazılım Geliştirme Kiti
SH	Strip Height	Sınır Uzunluğu
SOFM	Self-organizing Feature Map	Kendi Kendini Düzenleyen Haritalar
SVG	Scalable Vector Graphics	Ölçeklenebilir Vektör Grafikleri
UI	User Interface	Kullanıcı Arayüzü
W3C	World Wide Web Consortium	Dünya Çağında Ağ Birliği

1. GİRİŞ

Bu tez çalışmasında el yazısı tanıma algoritmalarını kullanan Tesseract OCR motoru kullanılarak el yazısı tanıma işlemi yapılmıştır. Okuma zorluğu çeken çocukların okuma zorluğunu aşabilmesi için yazılı olarak gerçekleştirilen yöntemlerin mobil cihazlar üzerine aktarılması amacı ile yola çıkılmıştır. Yazılı olarak kullanılan yöntemlerin mobil cihazlar üzerinde kontrol edici kişi olmaksızın, el yazısının program üzerinden tanımlanmasıyla, öğrencilerin kendi kendine çalışmalar yapabilmesi sağlanmıştır.

1.1 Tezin Amacı

Türkiye’de ve dünyada mobil cihazların önemi ve kullanımı her geçen gün artmaktadır. Mobil cihazlar çoğunlukla sosyal ağlara bağlanmak için kullanılmakta ve bu cihazlara genellikle oyun ve müzik uygulamaları indirilmektedir. Hayatın her alanına girmeye başlayan bu cihazlar eğitim ve öğretim alanında da kullanılmaya başlanmıştır. Geliştirilen bu yazılım okuma zorluğu çeken öğrenciler için kullanılan klasik yöntemlerin mobil ortama aktarılmış bir örneği olmasının yanında, zaman, mekan, okuma için yardımcı kısıtlaması olmaksızın, daha eğlenceli bir ortam sunmaktadır.

1.2 Tezin Kapsamı

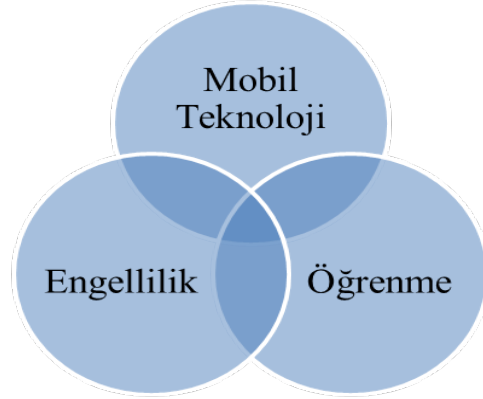
Okuma ve öğrenme kavramları insanların hayatında önemli iki temel kavramdır. Çağı yakalayabilmenin yanı sıra, yaşam içinde pek çok aktivitenin gerçekleştirilmesi için okumaya ve öğrenmeye ihtiyaç duyulmaktadır. Öğrenme zorluğu çeken çocuklar kelimeyi yanlış okuma, kelimeyi tekrarlama ya da kelimeyi atlama gibi okuma hataları yapar. Kelimeleri okumada zorluk çeken öğrencinin öncelikle hangi

kelimelerde sorun yařadığı belirlenmeli ve güçlük çekilen kelimeler üzerinde durulmalıdır. Bu problemin giderilmesi için ortaya atılmış birçok yöntem bulunmaktadır [1]. Bu yöntemlerin bazılarında öğrenciye okumada eşlik edecek, yanlışlarını düzeltecek, takıldığı yerden devam edecek yardımcıların bulunması gerekmektedir. Bazı yöntemlerde ise zorluk çekilen kelimeler parçalara ayrılarak harf bazında çalışmalar yapılmaktadır.

Ülkemizde son yıllarda sağlanan devlet teşviklerinin de etkisiyle eğitim kurumlarında mobil cihazların sayısı hızla artmaya başlamıştır. Fakat cihazlar üzerinde kullanılacak mobil öğrenme yazılımları maalesef aynı hızla artmamaktadır. Bu sebeple mobil öğrenme alanında büyük bir potansiyel bulunmaktadır.

Mobil öğrenme, mekân ve zaman kısıtlaması olmaksızın soyut bilgi ile pratiğı bir araya getiren köprü olarak değerlendirilebilir. Mobil teknolojileri kullanan bir öğrenme ortamı salt bilgi sunumundan ziyade model ve kavramları kullanan uyarlanabilir bir altyapı olmalıdır. Gerçekleştirilen mobil uygulama, teknolojiye ilgi duyan yeni nesil için daha eğlenceli bir çalışma ortamı sunmaktadır.

Bu tezin kapsamı üç ayrı bölümün kesişimidir: Mobil teknoloji, eğitim ve engellilik.



Şekil 1-1. Tezin kapsamı

Şekil 1-1 de gösterildiği üzere mobil teknoloji, engellilik ve eğitimin kesişimi vurgulanmıştır. Öğrenci merkezli tasarım modelinde engelliler için eğitim amaçlı mobil teknolojilerin kullanımında zorluklarla karşılaşılır. Direk öğrenci katılımındaki tasarımlarda kullanıcı ihtiyaçlarına öngörüler yardımcı olur, eğitimsel ve fonksiyonel gereksinimlerle eşleştirilir. Bu uygulama öğrenme zorluğu çeken çocukların mobil öğrenme ortamında problemlerini giderebilmek için geliştirilmiştir.

1.3 Benzer Çalışmalar

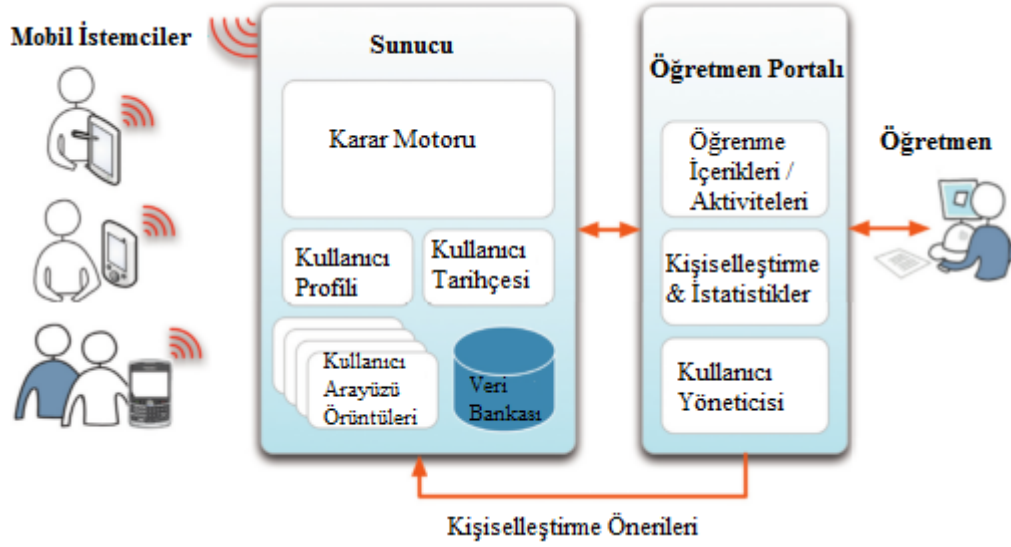
Birçok farklı platformda mobil öğrenme alanında yapılmış çalışmalar bulunmaktadır. Bunların çok küçük bir kısmı mobil cihazlarla uyumlu çalışmaktadır ancak teknolojinin donanım açısından geliştirilmesiyle, yazılım da aynı gelişmeyi göstermekte ve her geçen gün bu cihazlarla uyumlu uygulamaların sayısı artmaktadır. Bunlardan bazılarından bu bölümde bahsedilmiştir.

Etkili İngilizce Öğrenmeyi Desteklemek için Kişiselleştirilmiş Akıllı Mobil Öğrenme Sistemi (Personalized Intelligent Mobile Learning System for Supporting Effective English Learning - PIMS) adı verilen sistem okuma yeteneklerine göre kullanıcılara İngilizce haber başlıkları tavsiye eden bir yaklaşımdır ve bu yaklaşımla yapılan

sistem Bulanık Öğelerin Tepki Kuramı'nı (The Fuzzy Item Response Theory - FIRT) temel alarak benzer bir şekilde yabancı dili İngilizce olan kullanıcılar için okuma yeteneklerini geliştirmektedir [2].

M-CALL, Korece öğrenenler için ders içerikleri bulunduran, mobil bilgisayar destekli ve oyun tabanlı olarak geliştirilmiş bir projedir. Öğrencilerin motivasyonunu arttırmak amacıyla sanal hayvan oyunu şeklinde tasarlanmıştır [3].

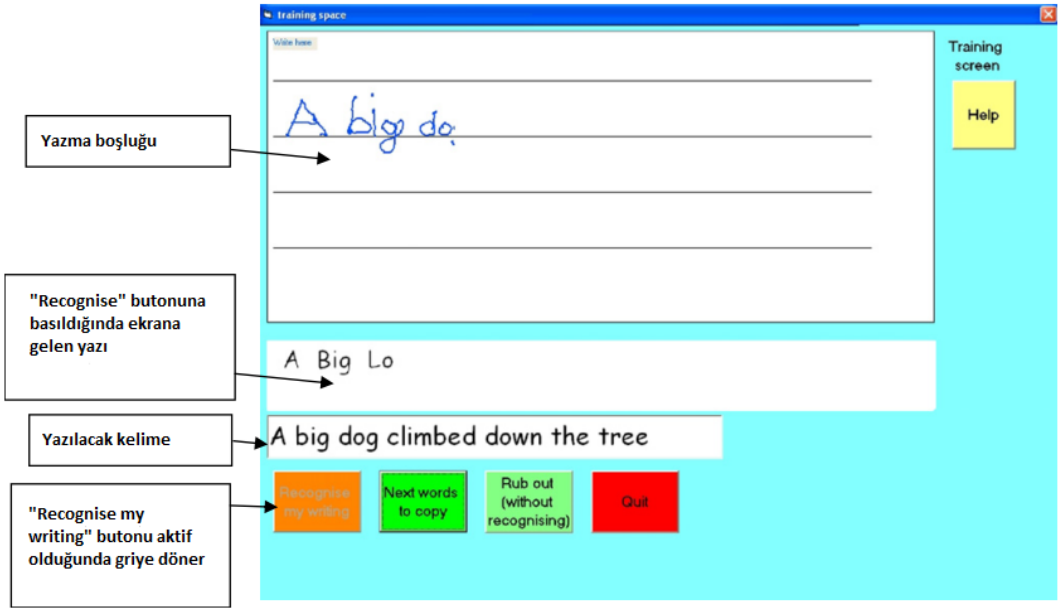
Akıllı Mobil Öğrenme Etkileşim Sistemi (Intelligent Mobile Learning Interaction System - IMLIS), zihinsel engellilere mobil öğrenme ortamı sunan bir mimaridir. Kullanıcı yetenekleri, öğrenme geçmişini kullanarak tasarlanan karar motoruyla kişiselleştirmeyi amaçlamakta ve farklı öğrenme seviyelerine göre farklı arayüzlerle, öğrenme aktiviteleriyle birlikte adaptasyonu sağlamaktadır [4]. Çalışmanın mimarisi Şekil 1-2'de gösterilmiştir.



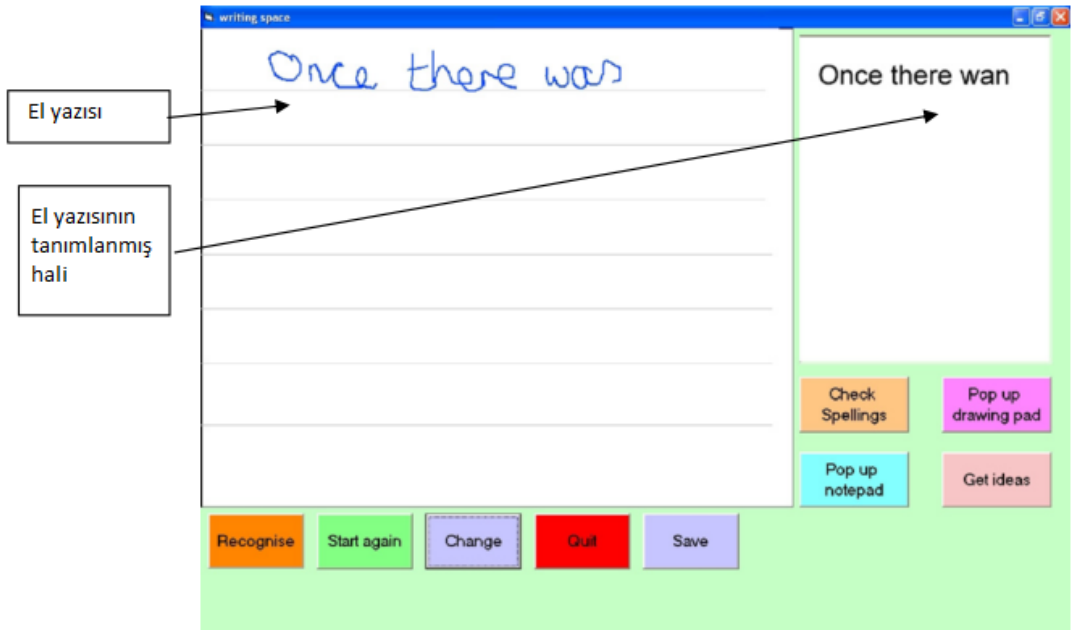
Şekil 1-2.IMLIS'in mimarisi [4]

Mimaride de gösterildiği üzere istemciler sunucuya bağlanarak, öğretmenin portala koyduğu eğitim içeriklerini kullanabilmekte, öğretmen ise öğrencilerin istatistiklerini görebilmektedir. Uygulama donanımsal olarak günümüz mobil cihazlarıyla uyumlu değildir.

Çocukların el yazısının ön plana alındığı başka bir çalışma ise, kelime işlemcilerde çalışmanın klasik yöntemlere göre verimliliği ölçülmektedir. Çocukların kalem kağıtla çalışmaya oranla, teknolojik ortamlarda daha hevesle çalıştığı [5], bilgisayar ortamındaki kelime işlemcide çalıştıktan sonra tekrar kalem kağıt ile çalışmaya döndüklerinde ise motivasyonlarında düşme olduğu gözlenmiş [6], dolayısıyla çocukların daha verimli çalışabilmesi, takım çalışmasına yatkınlıklarının artması için bu proje geliştirilmiştir. Geliştirme ortamı olarak Visual Basic, el yazısı tanımda yardımcı olması için CalliGrapher kullanılmıştır. Şekil 1-3'te eğitim, Şekil 1-4'te ise yazma ekranı gösterilen sisteme girişler bilgisayar ortamında fare yardımıyla yapılmaktadır [7]. Uygulama mobil cihazlarda değil, bilgisayar ortamında kullanılacak şekilde geliştirilmiştir.



Şekil 1-3.Örnek uygulamanın eğitim ekranı [7]



Şekil 1-4.Örnek uygulamanın yazma ekranı [7]

2. PROBLEMİN TANIMI

Bu bölümde öğrenme ve okuma zorluğu, çözüm yöntemleri ve günümüzde yapılan teknolojik çalışmalar anlatılmıştır.

2.1. Okuma Zorluğu Nedir?

Okuma ve öğrenme kavramları insanların hayatında önemli iki temel kavramdır. Çağı yakalayabilmenin yanı sıra, yaşam içinde pek çok aktivitenin gerçekleştirilmesi için okumaya ve öğrenmeye ihtiyaç duyulmaktadır. Bu yeteneklerimizin geliştirilip kazanılması, geliştirilmesi eğitim öğretim programlarının içeriğinde bulunmaktadır.

Okuması akıcı insanların kelime bilgisi de geniş olur, dolayısıyla kelimeleri daha seri kullanırlar ve anlarlar. Okumada zorluk çeken insanlar anlamada sorunlar yaşayabilirler. Enerjilerini ve dikkatlerini kelimenin anlamına değil, seslendirilmesine verirler. Oysa anlayabilmek başarının temelidir [8].

Ülkemizdeki okullarda okuma zorluğu çeken çocuklara rastlamak mümkündür. Sesi doğru tanıma ve çözümlenme, akıcı okuma, okuduğunu anlama, yeterli kelime hazinesine sahip olma gibi gerekli okuma becerilerinden herhangi birinin kazanılmamış olmasından dolayı bireyin okuma sırasında yaşadığı güçlükler okuma güçlüğü olarak tanımlanmaktadır [9].

Öğrenme zorluğu çeken çocuklar okuma hataları yapmak, kelimeyi tekrarlamak ya da kelimeyi atlamak gibi okuma hataları yapar. Öğrenci okumakta zorluk çektiğinde ise, vazgeçip ders dışında farklı yönlere ilgi göstermektedir. Bu da öğrencinin eğitim açısından geri kalması, kendini yetersiz hissetmesi, karşısındakilerle iletişim kurmasında dahi zorluklar yaşaması anlamına gelir.

2.2. Öğrenme Zorluğunu Aşmak İçin Kullanılan Yöntemler

Öğrenme zorluğu özellikle ilkokul çağında rastlanabilecek bir sorundur. Öğrencilerin sıkıntı çekmesinin yanında okul yaşantısında arkadaşlarından geride kaldığını düşünmesi ile moral bozukluğu yaşayabilir, dahası yardım alamazsa okuldan soğuyarak başarısını olumsuz yönde etkileyebilir. Bu yüzden zorluk çeken öğrencinin ilk önce hangi kelimelerde sorun yaşadığı belirlenmeli ve günlük çekilen kelimeler üzerinde durulmalıdır. Bu problemin giderilmesi için ortaya atılmış birçok yöntem bulunmaktadır. Bunlarda en önemlileri tekrarlı okuma, eşli okuma, eko okuma ve ahenkli okumadır [9]:

- a- **Tekrarlı Okuma:** Hatalı okunan kelimenin öğrenciye tekrar tekrar okutulması ile gerçekleştirilir [10].
- b- **Eşli Okuma:** Eşli okumada, çocuğa okumada yardımcı olarak birinin olması gerekmektedir. Uygun seviyede bir okuma parçası seçildikten sonra öğrenci parçayı yüksek sesle okur. Kelimelerin okunması sırasında en ufak duraksamalarda, yardımcı kişi devreye girerek yardımcı olmalı, okuma devam ettirilmelidir [10].
- c- **Eko Okuma:** Okumada yardımcı olacak kişinin kelimeyi ya da cümleyi okuduktan sonra, öğrencinin tekrar ettiği okuma yöntemidir.
- d- **Ahenkli Okuma:** Okumada yardımcı olacak kişinin cümleyi bir yere kadar okuduktan sonra kalan kısmı öğrencinin okuduğu okuma yöntemidir [11].

En sık kullanılan yöntemlerden yukarıda bahsedilmiştir. Bunların dışında “Kelime Kutusu Stratejisi” (Word Box Strategy) ile “Paragrafın Önceden Dinlenmesi Stratejisi” (Listening Passage Preview Strategy) de kullanılan yöntemler arasındadır.

“Kelime Kutusu Stratejisi (Word Box Strategy)”, söylenen kelimedeki seslerin karelere bölünmesidir. Kelime Kutusu’nun tamamlanması sesin parçalara ayrılması, harfleri seslerle eşleştirme ve harfleri yazma olmak üzere üç aşamadan oluşur [12]. Yapılan çalışmalarda okuma güçlüğü çeken öğrencinin yanlış okuduğu kelimeleri kutulara yazması istendiğinde eksik yazdığı, bu doğrultuda “Kelime Kutusu Stratejisi (Word Box Strategy)” yoluyla kelimelerin doğru öğrenildiği saptanmıştır [1].

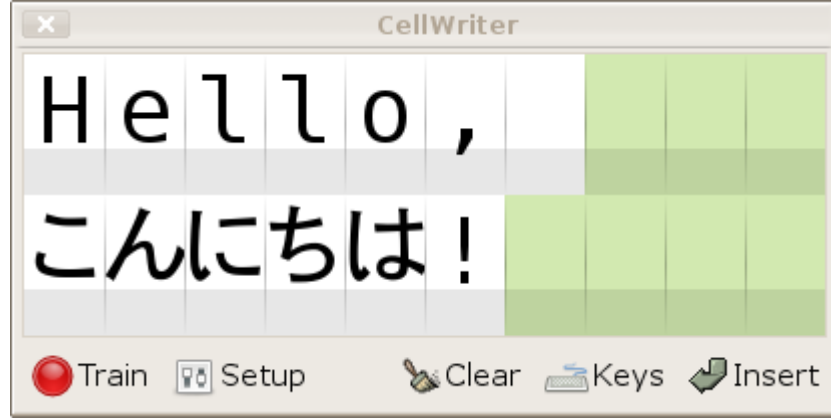
2.3. Öğrenme Zorluğu ve Günümüz Teknolojisi

Öğrenme zorluğunun giderilebilmesi için bazı yöntemler kullanılmaktadır. Sıklıkla kullanılan yöntemlerde hatalı okunan, okuma yapılırken atlanılan ya da tekrarlanarak okunan kelimelerin tekrar tekrar okutulması, yazılması şeklinde yapılan çalışmalar bunlardan bazılarıdır. Hatalı kelimelerin öğrenciye tekrarlanarak okunması teknolojik açıdan zorlayan bir işlem değildir, yalnız kelimenin yazılması işlemi için kullanıcı hareketlerini algılayabilecek bir ortam gereklidir. Donanımsal olan bu ihtiyacın yanında, algılanan hareketlerin anlamlandırılabilmesi için de yazılıma ihtiyaç duyulur.

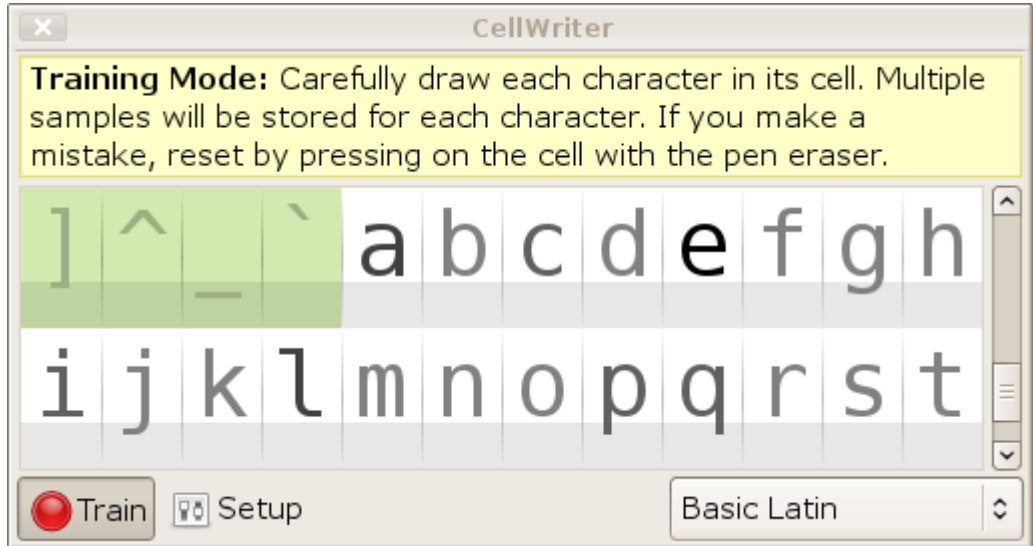
Yazı tanımak için kullanılan yazılımlardan bazıları CellWriter, CalliGrapher’dır. CalliGrapher Windows CE uygulamaları için kullanılan yazı tanıma yazılımıdır. En son sürümü olan CalliGrapher 7, 2003’de piyasaya sürülmüştür [13]. Cep bilgisayarlarında kullanılan yazılım el yazısının tanımlanmasına yöneliktir.

CellWriter ise Linux tabanlı olup girişlerin ızgaralarla yapıldığı yazı tanıma panelidir [14]. Karakter bazında tanıma işlemi yapan programda gerektiğinde eğitim yapılabilmektedir. Ayrıca çoklu dil desteği ile farklı bir alfabe ile yazılan karakterlerin latin harflerine çevirilmesi işlemi eğitilerek yapılabilmektedir. Sadece bilgisayar üzerinde çalışan bu program, fare yardımıyla yazılan karakterlerin tanıma

işlemini gerçekleştirmektedir. Şekil 2-1’de CellWriter’ın ana paneli, Şekil 2-2’de ise eğitim ekranı gösterilmektedir.



Şekil 2-1.CellWriter arayüzü [14]



Şekil 2-2.CellWriter eğitim ekranı [14]

3. MOBİL ÖĞRENME

Bu bölümde mobil öğrenme, e-öğrenme kavramları, aralarındaki farklar, mobil öğrenme sistemlerinin tasarımları ve içerikleri anlatılmıştır.

3.1 Mobil Öğrenme Kavramı

Kablosuz ağlarla internete bağlanan cihazların kullanımı oldukça yaygın hale gelmeye başlamıştır. 2010 yılı ortalarından beri başlayan mobil, fiber ve kablo internet abone sayılarının artışı gün geçtikçe devam etmektedir. Türkiye’de 2013 yılının birinci çeyreğinde toplam internet aboneliğinde bir önceki üç aylık döneme göre %1,6 artış gerçekleşmekte olup fiber, kablo ve özellikler mobil internet abonelerinin artmasıyla birlikte internet abone sayısındaki genel artış eğilimi de devam etmiştir [15]. Çizelge 3-1’de gösterildiği üzere internet abonelerinin çoğu cepten mobil internet kullanmaktadır. Yıllık büyüme oranında ise fiberden sonra en büyük yüzdeleri mobil bilgisayardan ve cepten internet almıştır. Mobil teknolojinin internette bu kadar yaygın kullanılması, iletişim, eğitim ve öğretim, bilgiye erişme gibi konuların mobil platformda her geçen gün artacağına göstergesidir. Eğitim alanında mobil öğrenme devlet teşvikiyle arttırılmaya çalışılmaktadır. Bu yönde yapılan çalışmalar mobil öğrenmenin de gelişmesine katkı sağlayacaktır.

Çizelge 3-1. Toplam internet abone sayıları [15]

	2012-1	2012-4	2013-1	Çeyrek Büyüme Oranı (2012-4...2013-1)	Yıllık Büyüme Oranı (2012-1...2013-1)
xDSL	6.736.138	6.643.299	6.678.907	0,5%	-0,8%
Mobil Bilgisayardan İnternet	1.780.895	1.909.530	1.977.612	3,6%	11,0%
Mobil Cepten İnternet	7.161.092	10.252.370	10.380.563	1,3%	45,0%
Kablo İnternet	483.843	500.658	501.201	0,1%	3,6%
Fiber	378.475	645.092	741.675	15,0%	96,0%
Diğer	139.858	139.665	137.256	-1,7%	-1,9%
TOPLAM	16.680.301	20.090.614	20.417.214	1,6%	22,4%

Günümüzde yaygın öğrenim tipleri aşağıdaki gibidir:

- Açık öğrenim,
- Uzaktan öğrenim,
- Yüz yüze öğrenim,
- Esnek öğrenim.

Yüz yüze öğrenim, öğretmen merkezli, kontrolün öğretmen üzerinde olduğu öğrenim sistemidir. Öğretmen, eğitici odaklı öğretimden uzaklaştıkça öğretmenin yaptığı işlemler teknoloji destekli uzaktan eğitim şeklini alır. Uzaktan eğitim, öğrencinin çalışma ve konumu açısından esnek olabildiği öğretim sistemidir [16].

Teknolojik gelişmeler, eğitimdeki duvarları kaldırarak uzaktan eğitimin gelişmesine yardım eder ve esnek eğitim ortaya çıkar. Esnek eğitim, çalışma ortamının esnekliği sayesinde hayat boyu öğrenmeyi destekler [17]. Esnek eğitimde ayrıca, öğrenci ne çalışacağını kendisi kontrol eder.

Esnek ve uzaktan öğrenme, öğrencilere öğrenimlerini kendilerinin kontrol etmesini sağlar. Bu iki tip öğrenme "açık öğrenme" olarak tanımlanabilir [18]. "Açık öğrenci" özelleştirilmiş ders materyalleri ile eğitim görür ve günümüzde gelişen eğitim teknolojileri araçları ile bu gerçekleştirilmektedir.

Esnek bir çatıda öğrenme yöntemleri beş açıdan kıyaslanır. Öncelik sırasına ve öğrenme seviyesine göre sorular aşağıdaki gibidir:

- Zaman dilimi (Öğrenme ne zaman gerçekleşir?)
- Coğrafi Konum (Öğrenme nerede yapılır?)
- Kişisel öğrenme tarzı (Öğrenme nasıl yapılır?)
- Öngörülen konu içeriği (Ne öğretilecek?)
- Varsayılan öncelikli öğrenim (Öğrenen kim?)

Esnek öğrenme herhangi bir elektronik iletişim vasıtasıyla gerçekleştirilebilir ve e-öğrenme olarak da bilinir. Çevrimiçi öğrenme ve mobil öğrenme, e-öğrenmenin iki şeklidir. Çizelge 3-2’de esnek öğrenme modellerinin gelişimi gösterilmektedir. Bu çizelge uzak ve çevrimiçi öğrenme açısından, mobil öğrenmeyle genişletilmiş halini içermektedir. Her öğrenme modeli bir öğrenme pedagojisiyle ilişkilidir [16].

Çizelge 3-2.Esnek Öğrenme Modellerinin gelişimi [19]

Zaman Aralığı	Teknoloji ve Platformlar	Öğrenme modelleri
1970 ortaları	Geniş alan ağı	Uzaktan öğretim
1980 başları	1980 lerin başında internet (kişisel bilgisayarlar)	Çevrimiçi öğrenme
1990 başları	www (kişisel bilgisayarlar, sunucu/istemci mimarisi)	Çevrimiçi öğrenme
21 yy başları	Mobil ağlar (kişisel mobil cihazlar)	Mobil öğrenme

3.2 E-Öğrenme ve Mobil Öğrenme Farkları

Çevrimiçi öğrenme interneti ve kurumsal intraneti kullanır. Çevrimiçi öğrenme uygulamaları multimedya öğeleri içerir ve bilgi kısa ve net değildir.

Uygulama açısından mobil öğrenme sahip olduğu farklı özelliklerinden ötürü, çevrimiçi öğrenmeden farklıdır. Benzer olarak her ikisinde de iletişim kanalında öğrenci ve öğretmen vardır, fiziksel olarak her ikisi birbirinden ayrıdır. Buna rağmen, mobil öğrenme çevrimiçi öğrenmeye göre artık daha yaygındır, yer ve zaman kısıtlaması yoktur. Ek olarak, mobil öğrenme daha kişiselleştirilebilir.

Mobil öğrenme; mekan ve zaman, ağ erişimi gibi belirli mobil altyapılarını kullanarak aşağıda açıklanan maddelerle diğer öğrenme metotlarından farklı bir konumda bulunur:

- **İletişim ağ erişimi:** Mobil öğrenmede yeni nesil cep telefonları, tablet bilgisayarlar gibi portatif cihazların kullanımı ile kullanıcıların ihtiyaçlarına göre erişim sağlanabilmektedir.
- **İletişim Tipi:** Kullanıcı olarak, mobil öğrencinin kişiselleştirilmiş servisi kimse ile paylaşılmamalıdır. Kullanıcıların güvenliği, mobil ağ şifreleme yöntemleriyle sağlanmalıdır.
- **Mekan ve Zaman Özgürlüğü:** Kullanıcı mobil öğrenmede mekandan bağımsızdır, 7/24 erişimi aynı cihazdan sağlayabilirler.

Mobil öğrenmenin diğer sistemlere göre avantajları aşağıda listelenmiştir:

- Eğitimde öğrencilerin kendilerine güvenmelerine yardımcı olma

- E-öğrenmenin avantajlarını sağlama
- Öğrenme süresince mekan ve zamandan bağımsız olarak çalışabilme
- Kişiselleştirilmiş eğitim desteği verebilme
- Sosyal iletişim ile ister bireysel ister grup çalışması şeklinde çalışabilme
- Öğretmen merkezli içeriği kullanabilme
- Durumlu öğrenmeye (situated learning) yardımcı olma
- Mobil ödev ve araştırmalar için bir araç olarak kullanılabilme
- Farklı ve yeni etkileşim yolları sağlama
- Öğrenme işlemi süresince iletişimi sağlama
- Kolay öğrenme materyalleri ile yönetimi ve güncellemeleri sağlama [20] [21] [22].

Mobil öğrenme bu avantajlarının yanında bazı zorlukları da yanında getirmektedir:

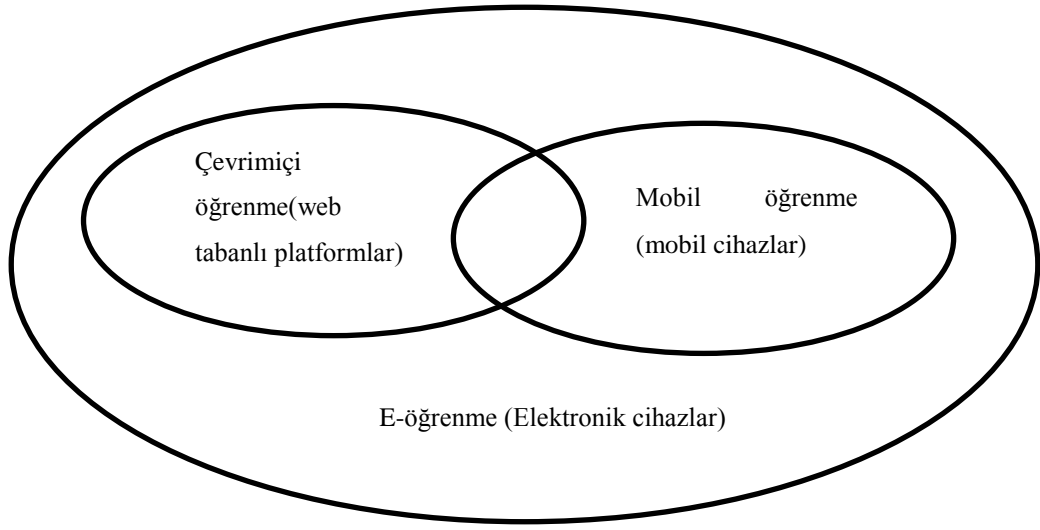
- Ekranların, bilgilerin yerleştirilmesi için yeterli büyüklüğe sahip olmaması
- Limitli depolama alanı
- Bazı durumlarda işletim sistemi açıkları
- Sınıf arkadaşları ya da ilgili diğer kişiler tarafından dışlanma hissi
- Farklı cihaz ve platformlarda çıkabilecek sorunlar

- Farklı cihazlarda öğrenme materyallerinin güncellenememesi
- Kopya çekilebilme olasılığı
- Teknolojinin gelişmesi nedeniyle mobil cihazların eskimesi
- Kablosuz internet iletimi ile ilgili problemler [20] [21] [22].

Mobil öğrenme aşağıdaki durumların gereksinim ve ihtiyaçlarına göre kullanılabilir [22] [20]:

- Sanal öğrenme ortamında eğitim ve öğretime katılma
- Faklı sayısal kütüphane ve arşivlere erişme
- Canlı yayın takibi ya da podcast kullanımı
- Farklı öğrenme materyallerine erişme (Testler, anketler...)
- Eğlenceli öğrenme ortamı sunma

E-öğrenme, çevrimiçi öğrenme ve mobil öğrenme arasındaki ilişki Şekil 3-1'de gösterilmektedir. Çevrimiçi öğrenme ve mobil öğrenme, e-öğrenmenin alt kümesi olarak gösterilmiştir.



Şekil 3-1.E-öğrenme, çevrimiçi öğrenme ve mobil öğrenme arasındaki ilişki [19]

Eğer bir çoklu platform yaklaşımı geliştirilirse, çevrimiçi ve mobil öğrenmeyi aynı eğitim içeriğinde kullanmak mümkün olur. Bu model öğrenme modellerinde hibrit ya da karışık model olarak adlandırılır [23].

3.3 Mobil Öğrenme Sistemleri Tasarımı

Geleneksel eğitim anlayışı olan öğretmen merkezli eğitim son zamanlarda öğrenci merkezli eğitime doğru kaymaktadır. Bu yüzden eğitimde öğrencinin yetenekleri üzerine odaklanması gerekmektedir [24]. Kullanıcı merkezli tasarım ile öğrenci merkezli tasarım arasındaki farktan aşağıda bahsedilmiştir:

- **Büyüme:** Eğitimin öğrenciler tarafından yapıldığı kısma odaklanması gerekir. Geliştiriciler ve tasarımcılar için ana hedef öğrenciler olmalıdır.
- **Çeşitlilik:** Öğrencilerin algılamaları, sosyal ve kültürel geçmişleri, öğrenim şekilleri birbirinden farklıdır. Öğrenciler, kullanıcı merkezli tasarım modelindeki kullanıcılardan daha heterojen görünebilirler. Kullanıcıların

çalışma tarzı benzer ya da farklı olabilir ama pratikteki uzmanlık seviyeleri tecrübelerine göre değişiklik gösterebilir.

- **Birlik:** Öğrencilere içeriğin uygun olması tasarım tarafından desteklenmelidir. Anlaşılabilir, motive edebilir ve odaklanılabilir olmalıdır. Profesyonel tasarımcı, kullanıcının ve kendi bakış açısının arasındaki fark büyük olmamalıdır.

Öğrenci merkezli tasarımın temelinde, öğrencinin yetenekleri, bilgi birikimi ve tecrübelerini desteklemek yatmaktadır.



Şekil 3-2.Mobil öğrenme sisteminde öğrenci merkezli tasarım [4]

Şekil 3-2’de kişiselleştirilmiş mobil öğrenme ortamında öğrenci merkezli tasarım farklı açılarıyla gösterilmektedir. Bu metot kullanıcıların ihtiyaçlarını gidermeye ve tasarımcı açısından daha verimli bir öğretim yapılmasına katkı sağlamayı amaçlamaktadır.

Eğitim içeriklerinin kullanılmaya başlanması ile bazı özel ihtiyaçları olan hedef kitle zorluklar yaşayabilir. Bu tarz problemleri çözmek için öğrencilerin ihtiyaç ve kapasitelerine göre etkili bir sistem uygulanmalı ve öğrenme durumları, motivasyonları tanımlanmalıdır. İlk adım olarak kişisel ihtiyaçlar ve kısıtlamalar konusunda yönlendirme ve kılavuz gereklidir. İkinci olarak öğrenme sisteminde

egzersiz ve eğitim materyallerine erişim sağlanabilmelidir. Oryantasyon ve kılavuzluk gerekmektedir [4].

Ekran tasarımı faktörleri ve tekniklerinde kullanıcı arayüzü önemli bir yer teşkil eder. Günümüzde kullanıcı arayüzü (User Interface – UI) teknolojileri mobil ve taşınabilirliğe kaymaktadır. Sonuç olarak, alışkın olunan büyük ekranların tersine mobil cihazların küçük ekranlarına uygun tasarımlar yapılmalıdır. Mobil cihazların ekran boyutu ile ilgili kısıtlaması, mobil öğrenme konusunda önemli zorlukları da beraberinde getirmektedir [26].

Mobil kullanıcı arayüzü (Mobile User Interface - MUI) mobil cihazlarla etkileşim halindeki karma öğrenci grubu için olumlu bir kullanıcı tecrübesi sağlayacak şekilde geliştirilmelidir. Genellikle UI tasarımcılarına ekranda gereksiz nesne, obje kullanımından sakınılması tavsiye edilir. Bu da daha anlaşılır bir sunum sağlar. Son kullanıcının çok fazla düşünmediğini varsayarsak, basit bir arayüz daha kolay bir kullanım sağlayacaktır. Tasarım süresince bazı faktörler göz önünde bulundurulmalıdır [19]:

- Hedef grup için simetri
- Etkileşimli
- Ölçeklenebilir
- Kullanıcı dostu
- Erişilebilirlik
- Sürdürülebilirlik

- Uyarlanabilirlik

MUI ekran boşluğu göz önüne alınarak basitleştirilip tasarlanabilir. Mobil uygulamalarda basitlik tasarımda büyük bir rol oynamaktadır ve engeli olan öğrenciler için bu durum daha da önemli hale gelmektedir. Mobil uygulamalar için tasarım tavsiyeleri şu şekildedir:

- Uygulama basit olmalıdır.
- Aşağı açılan kutu yerine seçilebilen veri kullanılmalıdır.
- Büyük veri bloklarından sakınılmalıdır.
- Kısa cümleler ya da kelimeler kullanılmalıdır.
- Ana menüye dönebilmek için ya da yönlendirebilmek için kısayol olmalıdır.
- Kısayollarda kafa karışıklığı olma ihtimaline karşı metnin altı çizili olmalıdır.
- Uygulamadaki bilgiler tutarlı olmalıdır.

Yönlendirmeler için başlıklar kullanılmalıdır [25]. Bunlara ek olarak, tasarımcı simülator ve cihaz emülatörlerinde sistemi derleyip test etmelidir. Eğer içerik sisteme uygunsa, ses ve video gibi eğitim materyallerinin çalışması da sorunsuzsa, etkileşim akıcı bir şekilde sağlanıyorsa ve ekran boşluğu da uygunsa, sistem kabul edilebilir seviyededir [26].

3.4 Mobil Öğrenme Sistemleri İçeriği

Mobil öğrenme sistemlerinin içeriğinde öğrenme materyalleri ve uygulanabilmesi için belirli standartlar bulunmaktadır.

3.4.1 E-Öğrenme, Mobil Öğrenme Standartları

Son yıllarda tasarımcılar ve öğretmenler için farklı standartlar ortaya çıkmaya başlamıştır.

Bu standartlar birçok öğrenme yönetim sisteminden ayrı düşünülemez. Farklı standart organizasyonları her e-öğrenme ve mobil öğrenme standartları aşağıdaki özelliklere sahiptir [27]:

Tekrar Kullanılabilirlik: Öğrenim materyallerinin farklı sistemlerde benzer özelliklerle tekrardan kullanılabilmesi

Erişilebilirlik: Kişilerin engelleri ya da yetenekleri doğrultusunda farklı kullanıcılar için eşit erişim hakkı oluşturulabilmesi

Dayanıklılık: Uygulamanın yeni sürümü çıktığında her seferinde önemli modifikasyonlar ya da yeni tasarımlara gerek kalınmaması

Birlikte çalışılabilirlik: Çoklu platformlar tarafından bilginin paylaşılabilir olması

Satın alınabilirlik: Uygulama kullanımı süresince ve geliştirme safhasında hem zamandan hem de paradan kazanç sağlayabilmesi

Uyarlanabilirlik: Kullanıcıların ihtiyaçlarına göre içeriklerin ve aktivitelerin kişiselleştirilebilmesi

Yönetilebilirlik: Sistemin görüntülenebilmesi ve bakımının yapılabilmesi

Ölçülebilirlik: Öğrenme uygulamalarında, farklı bilgi ve veri ölçeklerine uyum sağlayabilmesi

E-öğrenme ve mobil öğrenme çözümleri için kullanılan genel kapsamlı standartlar bu şekildedir ve birçok içerik yönetim sistemi tarafından da desteklenmektedir.

3.4.1.1 SCORM

Paylaşılabilir İçerikli Nesne Referans Modeli (Sharable Content Object Reference Model – SCORM), web tabanlı kullanımlar için e-öğrenme içerikli paket metodolojisidir [26]. SCORM, XML tabanlı teknolojisiyle özelleştirilmiş paketler içerir ve öğrenciler için tasarlanan öğrenim sayfalarında gömülü interaktif yönlendirmelerin anlık olarak çalışmasını sağlar. SCORM, e-öğrenme sistemlerinden çalışmasından ötürü mobil öğrenme sistemlerinde de kullanılabilir. SCORM, e-öğrenme sistemlerinden çalışmasından ötürü mobil öğrenme sistemlerinde de kullanılabilir.

Bu teknik sayfalar arasında yönlendirmenin kolay olmasını ve öğrenim içeriklerinin öğretmen tarafından hazırlanmasına yardım edilmesini sağlar. SCORM paketi her öğrenme ortamında aynı özelliklerle çalışabilmektedir [27].

3.4.1.2 CORDRA

İçerik Nesne Deposu Keşfi ve Kayıt Mimarisi (Content Object Repository Discovery and Registration/Resolution Architecture - CORDRA), öğrenme içeriğinin tekrar

kullanılması ve paylaşılmasını sađlayan e-öđrenme ve mobil öđrenme sistemlerinde kullanılan standart bir modeldir.

3.4.1.3 IMS Küresel Öđrenme Birliđi

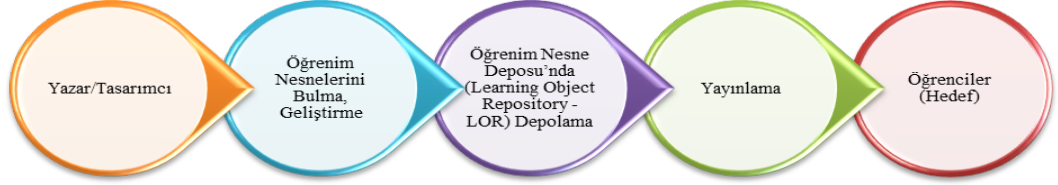
Uluslararası Yönetim Sistemi (International Management System - IMS); küresel, kar amacı gütmeyen, eđitim teknolojileri endüstrilerinin büyümesi, koordine edilmesini, standartlar ve özelleştirmeler dođrultusunda destek veren bir kuruluştur. IMS, mobil öđrenme ve e-öđrenme çözümlerinde içerik paketlerine büyük katkı sađlar [28].

3.4.1.4 IEEE LTSC

IEEE Öđrenme Teknoloji Standartları Komitesi (Learning Technology Standards Committee - LTSC) dünya çapında farklı akredite standartlarında ve kılavuzluđunda öđrenme teknolojileri geliştirmektedir. LTSC, eđitim teknoloji sistem mimarileri, öđrenim yönetim sistemlerindeki içerikler için farklı standartlarda veri modelleri yayınlamıştır [29].

3.4.2 Öđrenim Materyalleri

Öđrenim materyalleri, mobil öđrenmenin önemli kısımlarından biridir. Öđretmen ve geliştirici tarafından oluşturulan, öđrenci tarafından kullanılan elementlerdir [30]. Öđrenim nesnelerinin yaşam döngüsü Şekil 3-3'te gösterilmektedir.



Şekil 3-3.Öğrenim nesnelerinin yaşam döngüsü [31]

Öğrenim nesneleri içerikten bağımsız olmalı, açıklayıcı bilgi içermelidir. Öğrenim nesneleri aşağıdakileri içerebilmelidir:

- Öğrenim nesneleri hakkında genel bilgi (tanımlayıcılar, versiyonlar, durumlar, diller)
- Öğrenme nesnelerinin seviyesi
- İçeriğin tipi (metin, resim, medya, HTML)
- Sorular ve cevaplar
- Diğer öğrenim nesneleri ile alakalı olanlar (daha önceki derslerde kullanılan nesnelere)

4 EL YAZISI TANIMA

Bu bölümde el yazısı tanıma ve tanıma metotlarından, günlük hayatta kullanılan alanlardan bahsedilmiştir.

4.1 El Yazısı Tanıma

Yazı, kültür ve uygarlık anlamına da gelmektedir. Her bir simge kümesi (karakter ya da harf) temel şekillere sahiptir. Harfleri bir araya getirme kuralları dil bilimi olarak adlandırılmaktadır [32].

El yazısı son yıllarda teknolojinin de artmasıyla arka plana atılmaya, kalem ile kağıdın uyumunun yerini klavyeler almaya başlamıştır. Yazı, uzun zamandan beri insan hafızası geliştirmiş ve iletişimi kolaylaştırmıştır. Teknolojinin gelişmesi ile birlikte her ne kadar el yazısı kullanılsa bile klavyenin yeterli olmadığı ya da kullanışlı olmadığı durumlar meydana gelmektedir. Denklem ya da grafiklerin çizimleri buna örnek olarak verilebilir. Böyle bir durumda dijital kalemler ya da dokunmatik yüzeylerde çizimler yapmak kolaylık sağlamanın yanında el yazısının teknolojiyle tamamen kaybolmadığını göstermektedir.

Yeni teknolojilerin hayatımıza girmesiyle birlikte el yazısı önemini arttırmaya devam etmektedir. Mobil cihazların artmasıyla, kullanıcı arayüz etkileşimi için fare ve klavye gibi araçlar ortadan kalmaya başlamıştır. El yazısı teknolojiyle birleşerek mobil ortama taşınmıştır. Bununla beraber mobil ortamda yazının anlamlandırılabilmesi için yazı tanıma işlemleri yapılmaktadır.

El yazısının yorumlanması, metnin anlamlandırılmasına denir. Zarfların üzerindeki adresler buna örnek olarak verilebilir.

El yazısı verisinin sayısal şekle çevrilmesi için ya kağıt üzerine yazılmış metnin taratılması ya da elektronik yüzeyde yazılmış olması gerekmektedir. Mobil ortamdaki yazının tanımlanma işlemi; çevrimiçi ve çevrimdışı olarak ikiye ayrılmıştır. Yazım anında metnin tanımlanma işlemi çevrimiçi, taratılmış metinlerin tanımlanması çevrimdışı olarak nitelendirilir.

Çevrimiçi el yazısında noktaların iki boyutlu koordinatları tutulur. Çevrimiçi işlemde koordinatlar üzerinde çalışılırken çevrimdışı işlemde ise görüntüler üzerinde işlem yapılır.

4.1.1 Çevrimiçi Yazı Tanıma

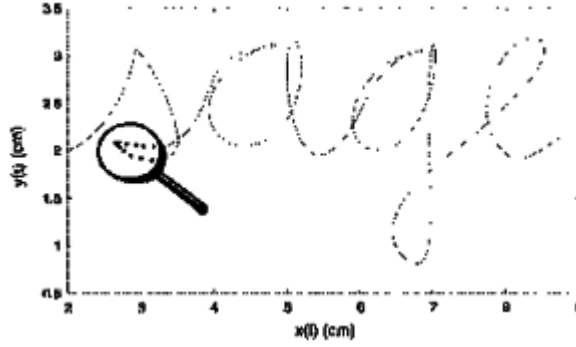
Çevrimiçi tanıma metotları ve teknikleri sayısal ortamdan alınan metnin otomatik olarak işlenmesi şeklindedir. Verinin pozisyonu, ivmesi, hızı gibi özellikleri belirlenebilir.

60'lardan beri el yazısı tanıma üzerinde zorluklar yaşanmaktadır [33]. Birçok yaklaşımlar denenmiş ve bunlar hakkında makaleler yazılmıştır [34].

Yıllar boyunca yazılan bu makaleler zamanla gelişerek yeni teknoloji üzerinde uygulanmaya koyulmuştur. Tanıma için öncelikle yazının tanımlanması, arayüz ile hareketlerin etkileşimi sağlanmalıdır. Daha sonra algılanan yazının kime ait olduğu bilgisi öğrenilebilir. Bu sistemler ışığında eğitim ve rehabilitasyon gibi alanlarda çalışmalar yapılabilir.

Şekil 4-1'de analizi yapılmış bir yazı örneği gösterilmektedir. Bitişik yazılmış el yazısı için veri gereksinimi; çevrimiçi tanımda 100 bayt veri, saniyedeki 100

örnekle örneklendirilmiştir. 21000 kelimededen oluşan bir sözlükle yapılan çevrimiçi tanıma işleminde %80 başarı sağlanmıştır. Tanıma üzerine son yıllarda yapılan çalışmalar daha yüksek performanslar vermektedir ancak tüm tanıma performansları test setlerinin verimine göre değişiklik göstermektedir [35].



Şekil 4-1.Çevrimiçi tanımda analizi yapılmış yazı örneği [35]

Birçok tanıma algoritmasında standart el yazısı karakterleri ya da normalleştirilmiş hareketler temel alınır. Genel normalizasyon çalışmaları içinde eğimin göz ardı edilmesi, temelin düzeltilmesi ve metin boyutunun ayarlanması vardır.

Tanıma algoritmalarında bölme işlemi için bazı ön işlemlerin yapılması gerekir. İki basamaktan ilki; metnin tamamına odaklanmaktır. Örneğin; metnin çevrimiçi algılanması, kelimelere bölme işlemleri gibi. İkincisi; karakter bölmeye dayalı metodolojilerdir. Bu işlemde en zorlayıcı kısım ise el yazısıyla yazılmış metnin tanımlanmasıdır. Birçok durumda, bölme işlemi daha önceki örneklere dayandırılarak ya da sınıflandırmadan sonra yapılmaktadır. Bu yaklaşımın daha anlamlı olabilmesi için sözlük uygulamaları kullanılabilir.

Karakter bölmede karşılaşılan en önemli problem; karakterlerin başlangıç ve bitiş kısımlarını belirlemektir. Birçok tanıma algoritmalarında sistemler tam verimle

çalışmamakta ve bu problemlerle uğraşmaktadır. Bu yüzden bu sistemler eğitilmeli ve büyük veri tabanları kullanılarak test edilmelidir.

El yazısı tanıma 2 farklı sınıflandırma metoduna ayrılmıştır [35]:

- Yapısal ve kural tabanlı metotlar
- İstatistiksel metotlar

4.1.1.1 Yapısal ve Kural Tabanlı Metotlar

Karakter şekillerin tanımlanabildiği, alakasız şekiller üzerinde çok da durulmamış bir yaklaşımdır. Kural tabanlı yaklaşım, geniş bir veri tabanındaki karakter ve kelimelerin otomatik olarak oluşturulması ve formüle edilmesinde karşılaşılan zorluklar yüzünden kullanımı bırakılmıştır. Bu yaklaşım istatistikleri kullanarak belirli özelliklerin oluşum sıklığını kullanarak yenilenmiştir.

4.1.1.2 İstatistiksel Metotlar

İstatistiksel yaklaşım 3 grup metodu temel alır:

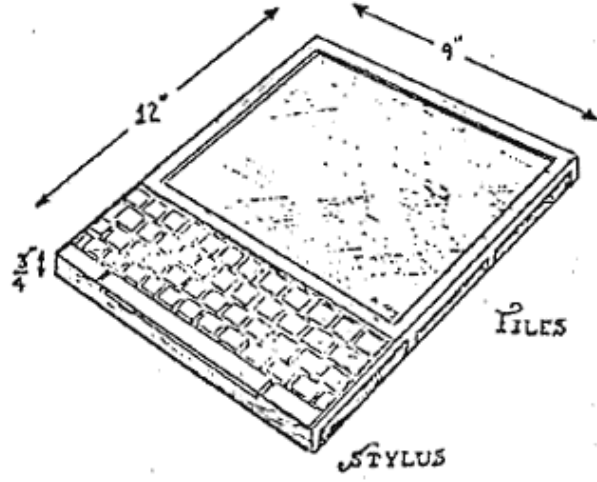
- **Kapalı Metotlar:** Kapalı metotlar lineer diskriminant analizi, ana bileşen analizi ve hiyerarşik kümeleme analizinden türemiştir. Bu yaklaşımların ana problemi; öncelikle yaklaşımların hipotezlere dayanması ya da istatistiksel dağılımı tanımlayan parametrelerdir. İkinci olarak ise büyük hesaplama ve bellek kaynaklarına ihtiyaç duyulmasıdır.

- **Açık Metotlar:** Açık istatistiksel metotlar genellikle yapay sinir ağlarına dayanır. Bu metotların sınıflandırma yöntemi, eğitim seti karakterlerinin istatistiklerine göre belirlenir. Birçok sistem çok katmanlı algılayıcılardan faydalanır. Geliştirilen yöntemler daha çok Kendi Kendini Düzenleyen Haritalar (Self-organizing Feature Map - SOFM) tarafından ve zaman gecikmeli sinir ağları üzerine yoğunlaşmıştır [35]. Önceki metotlar büyük bir eğitim setindeki karakter prototip şekillerinin otomatik olarak tanımlanmasıyla çalışır. Bu yaklaşım k-means ya da hiyerarşi kümeleme yöntemlerinde kullanır.

Çevrimiçi sistemleri daha dikkat çekici yapmak için kullanıcılara kişisel uyum sağlayabilen yaklaşımlar geliştirilmelidir. Kullanıcıdan bağımsız temel bir sistem, her bir karakter için farklı kullanıcıların el yazılarının tanımlandığı bir set ile gerçekleştirilebilir, her kullanıcı kendi eğitim setini kendi oluşturmalıdır. Başarılı olmak için, kullanıcıların yeni semboller ekleyebildiği ve sembol karmaşasının olmadığı bir yaklaşım olmalıdır [35].

4.1.1.3 Çevrimiçi Yazı Tanıma Mantığı ile Çalışan Donanım ve Yazılımlar

Dynabook: Tabletlerin atası sayılmaktadır. 1968 yılında Alan Kay tarafından tasarlanan Dynabook, kalemle kullanılan, grafik arayüze sahip, zamanının güçlü bir mikroişlemcisiyle çalışan günümüz tabletlerine benzer bir alettir. Şekil 4-2’de Dynabook’un tasarımcısı Alan C. Kay’ın hazırladığı taslak gösterilmektedir.



Şekil 4-2.Dynabook'un illüstrasyonu [36]

Elektronik tabletlerde çevrimiçi el yazısı tanıma işlemi yapılır, Dynabook da benzer şekilde çalışmaktaydı.

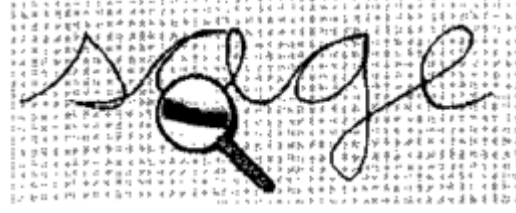
Herhangi bir tanımada öncelikle, elde edilen verinin gürültüsü azaltılır, normalize edilir ve sinyaller anlamlı parçalara bölünür.

Gürültü; verinin sayısallaştırılması esnasında, kullanıcı elinin değişmesi ya da parmak hareketleri gibi etmenler yüzünden oluşur. Gürültüyü azaltmak için veriyi düzeltme, sinyalleri filtreleme gibi yaklaşımlar kullanılır [35].

4.1.2 Çevrimdışı Yazı Tanıma

Çevrimdışı el yazı tanımada anlık olarak yazı tanımlanmaz, genellikle basılı dokümanların taratılması ile tanıma işlemi yapılır. Piksel tabanlı çalışır ve çevrimiçi el yazısı tanımada olduğu gibi yazının kalınlık, ivme gibi özellikleri alınamaz. Çevrimdışı el yazısı tanıma ikiye ayrılır:

Basılı görüntülerde Optik Karakter Tanıma (Optical Character Recognition - OCR) sistemleri, el yazısında ise Akıllı Karakter Tanıma (Intelligence Character Recognition - ICR) kullanılır.



Şekil 4-3. Pikel tabanlı çevrimdışı el yazısı tanıma [35]

Bitişik yazılmış el yazısı için veri gereksinimi; Şekil 4-3'te gösterilen çevrimdışı yazı tanımada 100 kilobayt inç başına 300 nokta ile örneklendirilmiştir. Genel açıdan bakıldığında küçük yazı tiplerinin iyi bir şekilde tanımlanabilmesi; bant genişliği ve görüntünün depolanması gibi durumlarla alakalıdır. Daha önce yapılan çevrimdışı çalışmalarda sırasıyla 10, 100, 1000 kelimedenden oluşan sözlükler göz önüne alındığında elde edilen verim sırasıyla %95, %58 ve %75 olmuştur [37].

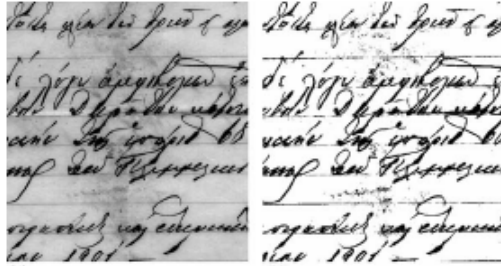
4.1.2.1 Ön İşleme

Doküman analizleri işlemleri daha çok taratılmış dokümanlardaki metnin tanımlanmasında kullanılır. Tanıma işleminde eşikleme kullanılır. Eşikleme, siyah-beyaz bir görüntünün grinin tonlarına çevrilmesi ile yapılır. Gürültülerden temizlenir, metin olan ön plan çıkarılır. Kelimeyi bölme, karakter bölme yapılır. Bu işlemler bitişik el yazısından çok harflerin ayrı ayrı yazıldığı yazılarda kullanılır.

4.1.2.1.1 Eşikleme

Eşiklemenin mantığında arka plandan ön plana çıkarma vardır. Bir dokümanın gri skalalı histogramında iki tepe noktası vardır; yüksek tepe noktası beyaz arka plan, küçük olan siyah ön planın tepe noktasıdır. Eşik değerinin belirlenme işleminde, gri skalanın en uygun değeri bu iki tepe noktasına göre seçilir. Bu değere göre beyaz ya da siyah olarak belirlenir.

Arka plan ve ön plan noktaları iki sınıfa ayrılır. Bu iki sınıf arasındaki değişkenliğin maksimum değeri optimal eşik değeri olarak belirlenir. Bu yöntem banka çeklerinde de kullanılan bir yöntemdir. Şekil 4-4'te verilen metnin tanıma işlemine tabi tutulabilmesi için arka plan renginin ön plan rengiyle karışmaması için eşikleme işlemi yapılmıştır.

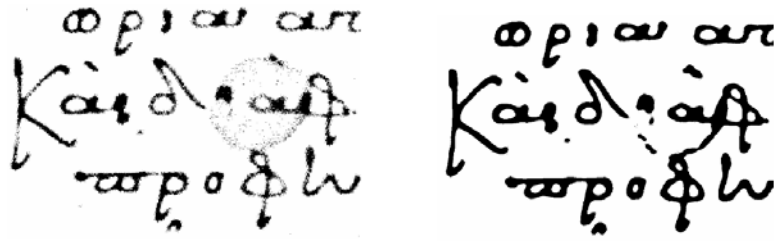


Şekil 4-4.Eşikleme ile arka plan renginin değiştirilmesi [38]

4.1.2.1.2 Gürültüyü Azaltma

Yazılı ya da makine yazısı ile basılmış dokümanlar için kullanılan bir doküman analiz yöntemidir. El yazısı dokümanlarda yazının vurguları korunur ancak dijital görüntülerde ya da taratılmış metinlerde gürültü oluşabilir.

Metin satırlarının vurgularının karışması sıklıkla karşılaşılan bir durumdur. Sinyallerden karıştırılan vurguların bölünmesi yapılmaktadır. Şekil 4-5’de metinde görülen damga, gürültüyü azaltmak için yapılan işlemden sonra kaybolmuş ve tanıma işleminden daha verimli sonuçlar elde etmek için ise; metin çizgileri daha belirgin hale getirilmiştir [38].



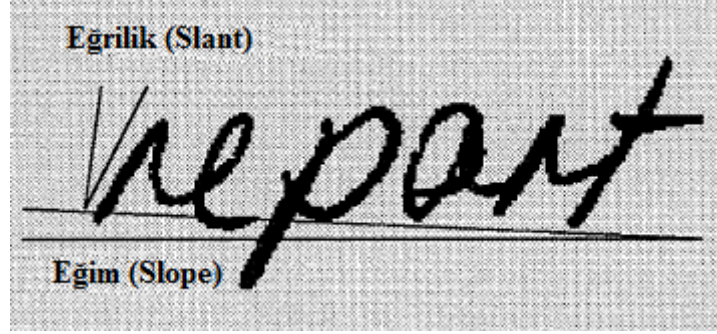
Şekil 4-5. Gürültüsü azaltılmış metin [38]

Gürültü piksellerinin giderilmesi amacıyla kullanılan yöntemlerden birisi 5x5 boyutlarında bir filtre elemanı kullanmaktır. Maskeleye yöntemi ile gerçekleştirilen filtreleme işlemi 5x5 maskeleye elemanı, karakter alanı içerisindeki bütün piksel noktalarına uygulanır. Maske eleman değerleri 1 olarak seçilir. Komşuluk sayısı eşik değerinin altında olan piksellerin silinmesi sonucu gürültüyü oluşturan pikseller silinmektedir [38].

4.1.2.1.3 Normalizasyon – Eğimi Düzeltme ve Eğriliği Kaldırma

Çevrimdışı Bitişik El Yazısı Tanıma (Cursive Script Recognition - CSR) banka çeklerinin ya da posta adreslerinin okunmasında kullanılan önemli bir yöntemdir. Hemen hemen her CRS sisteminde eğriliği (slant) ve eğimi (slope) düzeltme işlemi kullanılır. Olması gereken dikey eksen ile yazılan metnin dikey eksenindeki açıya eğrilik (slant), olması gereken yatay eksen ile yazılan metnin yatay eksenini

arasındaki açığa da eğim (slope) denir. Şekil 4-6'da metnin eğriliği ve eğimi gösterilmektedir.



Şekil 4-6.El yazısında eğrilik ve eğim [39]

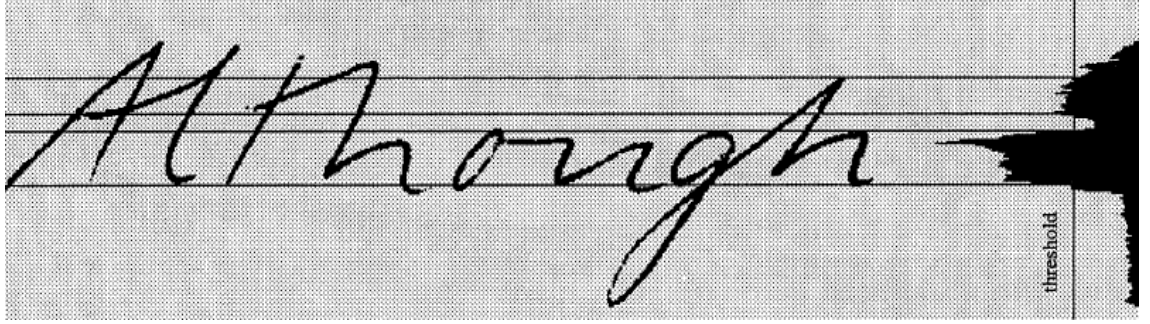
Bu iki kavramın kullanılarak el yazısının düzeltilmesine normalizasyon denir.

Tanıma tekniklerinde kelimeyi bölmek gerekmektedir. Bu da Dinamik Programlama (Dynamic Programming - DP) ve Saklı Markov Model (Hidden Markov Models - HMM) ile yapılabilir [38]. DP sistemlerinde şekillerin değişkenliğinin normalizasyonla azaltılması örüntü tanıma teknikleri tarafından sınıflandırılmasında kolaylık sağlar. HMM tabanlı sistemlerde bölme işlemi ön planda olduğundan, normalizasyon işlemleriyle yazılan metin parçalara bölünür. Normalizasyon işlemlerinin ilk basamakları genellikle Bosinovic ve Srihari Metodu (Bosinovic ve Srihari Method - BSM) ile yapılır.

BSM, el yazısı tanıma işlemlerinde kullanılan, yazıdaki eğikliği ve yazının hizalamasındaki eğriliği ortadan kaldıran bir yöntemdir.

BSM eğim düzeltme metodu ilk olarak karakterin gövdesine yakın olan bölgeyi alır. Şekil 4-7'de bu yaklaşım göz önüne alınarak karakter gövdesinin al ve üst sınırları belirlenmiştir. Bunu yapmak için tahmini karakter gövdesinin alt sınırına en yakın

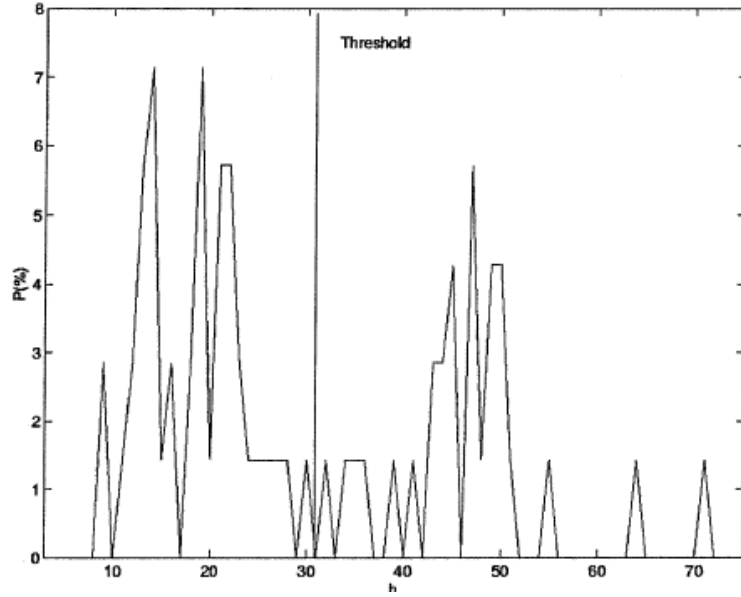
noktadan alt nokta çizilir. Bu çizgi en alt taban nokta (lower base line) olarak adlandırılır. Çizgi doğru olduğunda, görüntünün de eğriliği düzelmiş olur. Eğriliği düzeltme basamağından sonra tahmini karakter gövdesi metnin yatay olduğunu görmek için tekrar hesaplanabilir. Yatay yoğunluk histogramıyla düzeltildiği ispatlanabilir [39].



Şekil 4-7.Karakter gövdesi bulma [39]

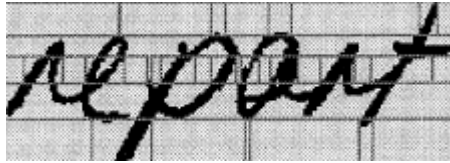
Şekil 4-7’de yazılan metnin yataydaki histogramı görülmektedir. Karakter gövdesi çizgileri genellikle en yüksek yoğunlukları baz alır. Karakterlerin dikey uzunluklarını da gözönüne aldığından gerçek karakter gövdesi bölgesini hesaplamak bazen sorun oluşturabilmektedir [39].

Şekil 4-8’de ise eşik değerinin histogramda yoğunluğun büyük olduğu kısma yakın olduğu, yoğunluğun az olmasının ise eşik belirlemede etkisinin az olduğu görülmektedir.



Şekil 4-8.Histograma göre eşik bulma [39]

BSM, kelimelerin yatay düzlemdeki eğriliğini baz alarak çalışmaktadır. Çalışma mantığı şu şekildedir: Öncelikle karakterlerin uzantıları olarak sayılan Maksimum Eğilim (Max Run - MR) parametresinden daha uzun olan tüm satırlar gözardı edilir. Kalan satırlarda Şekil 4-9’da gösterilen Sınır Uzunluğu (Strip Height - SH) parametresinden daha uzun olan bir uzunluk varsa sınır olarak kabul edilir.



Şekil 4-9.MR ve SH parametrelerinin kullanıldığı bir el yazısı örneği [39]

Yatay satırlar arasında kalan sınırlar ön plandaki pikselleri ayırmayacak şekilde birleştirilerek tahmini eğrilik seçilmiş olur. Her bir vurgunun alt ve üst yarısının ağırlık merkezi birleştirilerek tahmini yerel eğrilik bulunmuş olur. Bu yöntemin dezavantajı parametre ayarının gerekli olmasıdır. Parametre için optimum noktayı bulmak zor bir işlemdir. Nokta tanıma sisteminin en iyi performansı göz önüne

alınarak seçilir. Tahmini parametreler kalem kullanılarak el yazısı gibi farklı durumlarda deęişkenlik gösterebilir [39].

4.1.2.2 Parçalara Ayırma

El yazısı satırları, kelimeleri ve karakterleri bölme hakkında birçok yaklaşım bulunmaktadır. Elle yazılan kelime ve karakterlere bölme işlemi, makine yazısından farklıdır. Küçük bir eğrilik açısı aralığında, dikey histogram profili kullanılarak yapılabilir. Metnin satırları komşu satırlara artan ve azalan şekilde dalgalanıp keşilebilir. Başka bir yaklaşım ise görünmeyen bir çizgiyi var sayar. Bu varsayılan temelde her bir bileşenin minimum noktası tahmin edilir. Tüm bileşenlerin minimumu farklı el yazılarını belirlemek için kümeleme tekniğiyle gruplanır.

4.1.2.3 Satır Tanımı

Metindeki eğrilikler düzeltildikten sonra satır tanımı yapmak daha kolaylaşır. Satır hizalamasından sonra kelimelerin birbirinden ayrılması işlemi yapılır.

Görüntü analizi ve dijital görüntü işlemede kullanılan özellik çıkarımı yöntemi olan Hough dönüşümü, satır tanıma işleminde de kullanılmaktadır. Görüntü işleme, tanıma işlemlerinde elde edilen her nesne eksiksiz olarak bir şekle benzetilememekte ve bu yüzden işlemler yapılamamaktadır. Bu yöntemin amacı belirli bir şekil kümesindeki bozuk nesne örneklerini bulmakta, tanımlanmakta sorunlu olan nesnelerin tanımlanmasını sağlamaktır [40].

Doğrusal Hough dönüşüm algoritması akümülatör adı verilen iki boyutlu bir dizi kullanır. Bu dizi bir doğrunun varlığını;

$$r = x\cos\theta + y\sin\theta$$

şeklinde tanımlar. Akümülatörün boyutu bilinmeyen parametrelerin sayısına eşittir. Örneğin (r, θ) düşünelim. Eğer doğru olduğuna dair yeterince bilgi varsa, her bir (x,y) deki piksel ve onların komşuları için, Hough dönüşüm algoritması kullanılır. Doğrunun parametreleri (r, θ) hesaplanır. Akümülatörün değeri bir arttırılır. En yüksek değeri bulmak için akümülatör uzayında, doğrular çıkarılabilir ve yaklaşık geometrik tanımları yapılır. Hangi doğrudan ne kadar olduğunu bularak eşik değerinin bulunmasıyla en basit yoldan tepe noktaları bulunmuş olur. Uzunluk bilgisi içermeyen doğrularla karşılaşıldığında bir sonraki basamak; hangi doğrularla hangi doğruların eşleşeceğini bulmaktır. Hatta kenar tanıma basamağındaki hatalardan dolayı, akümülatör uzayında da hatalarla karşılaşılır. Bu da uygun tepelerin bulunmasını, dolayısıyla uygun doğruların bulunmasını zor hale getirir.

Doğrusal Hough dönüşümünün sonucu; iki boyutlu dizi(matriks) akümülatör ile benzerdir. Matriksin bir boyutu olan açı θ ile, uzaklık ise r ile gösterilir. Matriksin her bir elemanı doğru üzerinde bulunan noktaların ya da piksellerin sayısına eşittir. Bu yüzden en büyük değerli eleman düz bir doğruyu tanımlar [41].

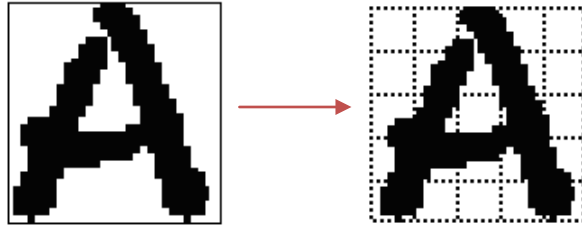
4.1.2.4 Özellik Çıkarımı

Özellik çıkarımının hedefinde en az miktarda element ile en yüksek tanıma oranına erişmek vardır. El yazısının kişilere göre değişkenlik göstermesinden dolayı tanıma işlemleri zordur.

Karakter tanımda bazen istatistiksel noktalar göz önüne alınır. Ana istatistik özellikleri şunlardır:

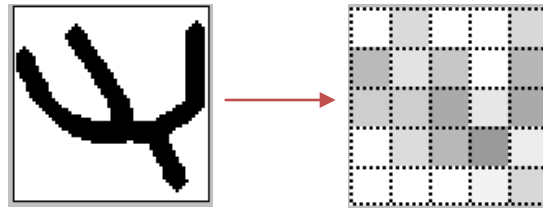
- Bölgeleme (Zoning)
- İzdüşüm ve profiller (Projections and profiles)
- Kesişme ve uzaklıklar (Crossing and distances)

Bölgeleme işleminde, Şekil 4-10'da görüldüğü üzere karakterin bulunduğu kısım $n \times m$ kadar bölgeye bölünür. Her bir bölümden özellik vektörü şekli çıkarılır. Bölgelemedenin mantığı global karakteristiklerin yerine lokal karakteristikleri elde etmektir.



Şekil 4-10. Bölgeleme ile karakter tanıma [38]

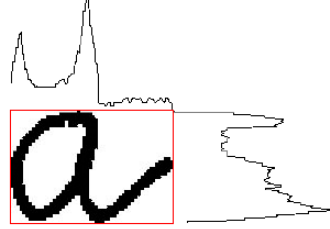
Arka plan piksellerinin sayısı yoğunluğu da göstermektedir. Şekil 4-11'de görüldüğü üzere koyu kareler yüksek yoğunluktaki piksel bölgelerini göstermektedir.



Şekil 4-11. Piksellerdeki yoğunluk [38]

İzdüşüm, iki boyutlu olan karakterlerin tek boyutlu olarak gösterilmesidir. Bu özellikler; gürültü, deformasyon ve rotasyona göre değişiklik gösterebilir. İzdüşümde

karakterin her bir satır ve sütunundaki piksel sayısına bakılır. Şekil 4-12’de izdüşümü alınan bir karakter gösterilmektedir.



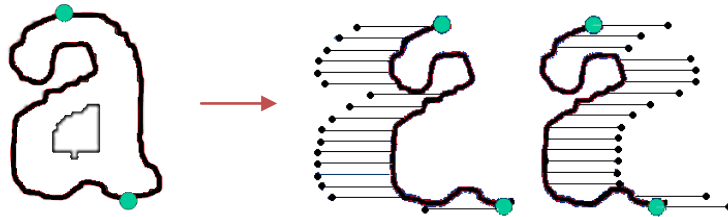
Şekil 4-12.Karakterin izdüşümü [38]

Profillerde kutu içine alınan karakter ile karakterin kenarları arasındaki mesafenin piksel sayısı kullanılır. Şekil 4-13’te profili gösterilen karakter örneği verilmiştir.



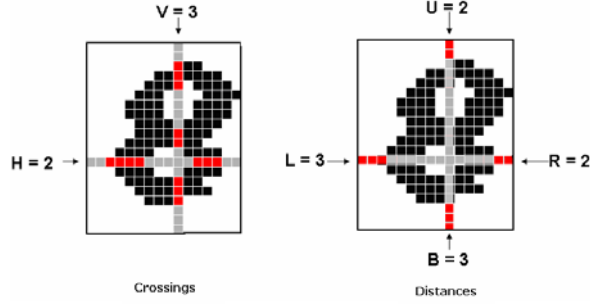
Şekil 4-13.Karakter profili [38]

Aynı zamanda bu özellik karakterin kontörlerinin de çıkarılmasını sağlamaktadır. Kontörlerin en alttaki ve en üstteki noktalarını belirleyebilir ve bunların iç ve dış kısımlarının hesaplayabilir. Şekil 4-14’de kontörleri çizilen karakter gösterilmektedir.



Şekil 4-14.Karakterin iç ve dış noktalarının vurgulanması [38]

Kesişme, kutulanan karakterin yatay ve dikey koordinatlar boyunca arka plandan ön plana geçen piksel sayısını kullanır. Uzaklık ise, kutulanan karakterin sağdan, soldan, alttan ve üstten sınırlara olan uzaklığını hesaplar. Bu iki özelliğin karakter üzerinde gösterimi Şekil 4-15'te gösterilmektedir.



Şekil 4-15.Karakterlerin kesişme ve uzaklık özellikleri [38]

Karakterler hakkında elde edilen bu bilgiler önemli veriler olup, daha düzgün tanıma yapılmasına yardımcı olmaktadır.

4.1.2.5 Sınıflandırma

Metnin yazım şekline göre daha önceden hazırlanmış sınıflara göre sınıflandırma işlemidir. Sınıflandırma için kullanılan bazı yöntemler:

En yakın k komşu algoritması: Sistemi eğitmeye gerek yoktur ve sınıflandırmak için k adet komşunun sınıf değerlerine bakılması yeterlidir. Gürültülü verilerde verimli sonuçlar alınabilmektedir. Bu algoritmanın kullanılabilmesi için k parametresine ihtiyaç duyulmaktadır.

Yapay sinir ağları: Bitişik el yazısı tanımda yapay sinir ağları ve genetik algoritması kullanılabilir. Öncelikle tanıma işleminde, el yazısının birbirinden ayrılması ile karakterlerin çıkarılması yapılır. Kelimedeki her bir karakterin

vektörleri seçilir ve bölme işlemi yapılır. Yapay sinir ağlarına bu vektörler girdi olarak verilir. Sinir ağlarından çıktı kümesi oluşturulur ve genetik algoritması bunlardan el yazısı tanıma modeline en çok uyan modeli seçer.

Algoritmada taratılmış gri tonlu görüntü monokromatik görüntüye çevrilerek piksellerin değerlerinden anlam çıkarılmaya çalışılır. Siyah pikseller için 0, beyazlar için 255 değeri verilir. (0,0) noktasından ilk siyah piksele kadar kontrol yapılır. Bulunan ilk siyah piksel, yazının ilk kelimesinin “üst noktası” olarak varsayılır. Üst noktası bulunduktan sonra, birbirine bitişik siyah noktaların değeri ise 999 olarak verilir. İlk basamak tamamlandıktan sonra, kelimedeki 999 değerli tüm karakterler birbiriyle bitişiktir. Tüm bitişik noktalar bulunduktan sonra, satır bazında alt piksellerden üst piksellere doğru 999 değerleri aratılmaya başlanır. Bu değer “alt nokta” olarak adlandırılır. Bu nokta tanımlandıktan sonra, alt ve üst noktalar arasındaki bölgenin sol tarafında kaçırılmış kelime olup olmadığını bulmak için kontrol edilir. Solda başka kelime varsa, o zaman yeni bir üst ve alt nokta belirlenir ve aynı prosedür devam ettirilir. Üst, alt, sağ ve sol nokta bulunduktan sonra kelime çıkarılıp farklı bir matriste saklanır. Bitişik el yazılarında farklı karakterler arasındaki ayırım noktalarının işaretlenmesinden sonra bu basamak izlenir. Kelimenin kaç sütuna ayrılacağı bulunur. Bütün karakterlerin kesimleri sağdan ve soldan grilik değeri 0,5 ile işaretlenen pikselden sonra kesilerek birbirinden ayrılır. Kesim işleminden sonra, dosyaya yazılan tüm karakterler bir döngü ile yapay sinir ağları karakter tanınması ile birlikte okunur [42].

Saklı Markov Modelleme: Sınıflandırma işleminde verilerin birbirinden farklı oldukları varsayılarak işlem yapılır. Her bir elemanın maksimum olabilirlik derecesine göre hangi sınıfa ait olduğu hesaplanır. Bu algoritmanın amacı, belirli bir sınıfın oluşma olasılığını bulmak ve verileri buna göre sınıflandırmaktır.

4.2 El Yazısı Tanımının Kullanıldığı Alanlar

Mobil cihazlarda el yazılarının ve zarfların üzerindeki adreslerin okunması, el yazısı ile doldurulmuş formlardaki yazıların tanımlanması verilebilecek örneklerdendir. Diğer uygulama alanları ise imza tanıma, yazan kişinin kimliğini doğrulama, el yazısı öğrenme araçları olabilir.

El yazısı kimliği tanıma, yazının var olan yazan kişi kümesinden kime ait olduğunun bulunması işlemine denir. İmza tanıma işlemi ise imzanın belirli bir kişiye ait olup olmadığının bulur.

Kimlik tanıma işleminde, yazı üzerindeki adli incelemelerde, el yazısı tanıma ve yorumlama işlemleri kullanılarak nesnel arasındaki farklılıklar belirlenerek kişilere özgü olan yazıların kime ait olduğu bulunmaktadır.

İmza tanıma işlemi de el yazısı tanıma sınıfına girer. Sistemde kaydedilmiş referans alınan bir imza ile test imzası karşılaştırılır. İmzadan, imza atan kişiye has belirli özellikler çıkarılır.

Teknoloji geliştikçe el yazısı tanıma işlemleri de gelişmeye devam edecektir. Çevrimiçi sistemler çevrimdışı sistemlere göre daha hassastır. Buna rağmen çevrimdışı sistemlerin, banka çeklerinin üzerindeki miktarları okuma ve zarfların üzerindeki adresleri yorumlama açısından yeterli sayılabilecek etkisi vardır.

Çevrimdışı olarak yazılan verinin çevrimiçi olarak tanıma algoritmalarını kullanması, çevrimiçi sistemlerin başarısını daha dikkat çekici hale getirmiştir [43].

5 ÖĞRENME ZORLUĞU ÇEKENLER İÇİN MOBİL UYGULAMA GELİŞTİRİLMESİ

Bu bölümde öğrenme zorluğu çeken çocuklar için hazırlanan uygulamada kullanılan bileşenler ve çalışma mantıkları, uygulamanın mimarisi ve içeriği anlatılmıştır.

5.1 Uygulama Altyapısı

Sorunun tanımlanması, sorun çözümüne yönelik yöntemlerin belirlenmesi ile şekillen uygulama için öncelikle gerekli birkaç aşama bulunmaktadır. Mobil cihazlar üzerinde çalışacak uygulamanın hangi işletim sistemi üzerinde çalışacağına karar verilmesi gerekmektedir. Bunun için piyasada bulunan birkaç işletim sisteminden en yaygın işletim sistemi [42] olarak kullanılmasından ve uygulama kolaylığından dolayı Android uygun görülmüştür. Android üzerinde hazırlanan arayüz ile kullanıcının el yazısı bilgileri alınacaktır. Kullanıcıdan alınan bu bilgi öncelikle cihaz üzerinde Ölçeklenebilir Vektör Grafikleri (Scalable Vector Graphics - SVG) dosyası olarak kaydedilecek, daha sonra sunucudaki Tesseract'a tanıma işlemi için gönderilecektir.

5.1.1 Android İşletim Sistemi

Android, Google ve özgür yazılım grupları tarafından geliştirilen Linux tabanlı, mobil cihazlar için kullanılan, açık kaynak kodlu mobil işletim sistemidir. Linux çekirdeği üzerine kurulan Android işletim sisteminin kütüphaneleri, ara katman yazılımları C dilinde yazılmıştır. Uygulama yazılımları Java uyumlu kütüphaneleri içeren uygulama iskeletinde çalışır. Derlenmiş Java kodunu çalıştırmak için Dalvik sanal makinası kullanılır.

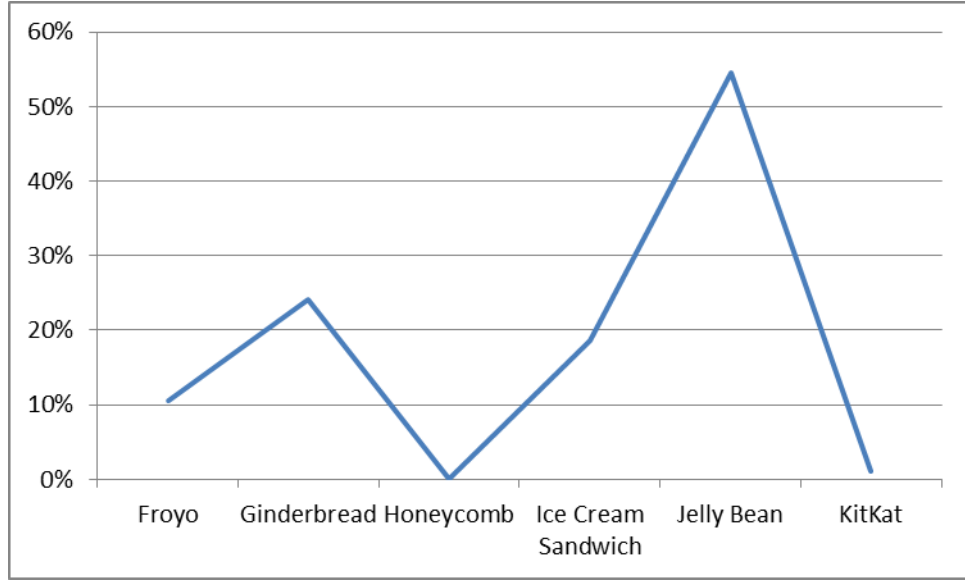
Geliştirme yapacak kişilerin Eclipse, Java Geliştirme Aracı (Java Development Kit - JDK) ve Android Yazılım Geliştirme Kiti (Software Development Kit - SDK) indirmesi gerekir. Eclipse üzerine Android bileşenlerinin yüklenmesi ile geliştirme işlemine başlanılabilir.

2013 Kasım ayında Android'in akıllı cihaz pazarındaki payı, Samsung cihazlar öncülüğünde %80'lere çıkmıştır [44][45].

Çizelge 5-1'de ve Şekil 5-1'de Android işletim sisteminin günümüze kadar piyasaya sürdüğü versiyonlar ve kullanım dağılımları gösterilmektedir.

Çizelge 5-1. Android işletim sistemi versiyonları ve kullanım yüzdeleri [46]

Versiyon	Adı	API	Kullanım Yüzdesi
2.2	Froyo	8	%1.6
2.3.3 - 2.3.7	Gingerbread	10	%24.1
3.2	Honeycomb	13	%0.1
4.0.3 - 4.0.4	Ice Cream Sandwich	15	%18.6
4.1.x	Jelly Bean	16	%37.4
4.2.x		17	%12.9
4.3		18	%4.2
4.4	KitKat	19	%1.1



Şekil 5-1.Android versiyonlarının kullanım grafiği

5.1.2 SVG Dosya Biçimi ve Kullanımı

SVG, XML’de iki boyutlu vektör grafikleri formatıdır. SVG, 1999 yılından bu yana Dünya Çapında Ağ Birliği (World Wide Web Consortium - W3C) tarafından geliştirilen açık standart olup, kullanımı tavsiye edilen bir dosya biçimidir [47].

Aşağıda SVG’yi diğer format tiplerinden ayıran özellikler bulunmaktadır:

- Vektör tabanlı olduğundan yakınlaştırıldığında görüntü kalitesi bozulmaz.
- Her bir bileşeni animasyonu destekler.
- DOM ve XSL gibi diğer W3C standartlarıyla entegreli olarak çalışmaktadır.
- SVG herhangi bir metin editörüyle oluşturulabilir.
- Kalitesini farklı çözünürlüklerde bozamaz.

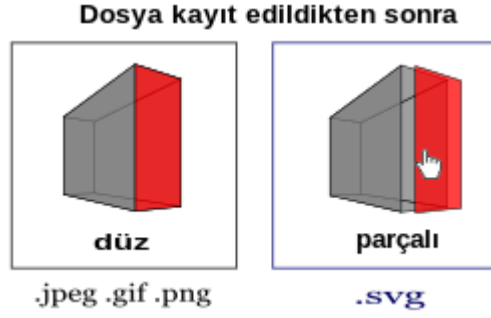
- SVG dosyaları tekrar eden birçok bileşen taşıdığından veri kaybı olmaksızın sıkıştırma yapılmasını sağlar.
- SVG filtreleri kullanılarak grafiklere renk değişimi, aydınlatma, gölgelendirme gibi özel efektler eklenebilmektedir.
- Metinler üzerinde rotasyon gibi işlemler yapılabilmesini de sağlar.

Şekil 5-2’de SVG formatının JPEG/GIF formatlarıyla karşılaştırıldığında, piksel tabanlı formatların yakınlştırıldığında görünümünün bozulduğu, vektör tabanlı SVG formatında ise bozulma olmadığı gösterilmektedir.



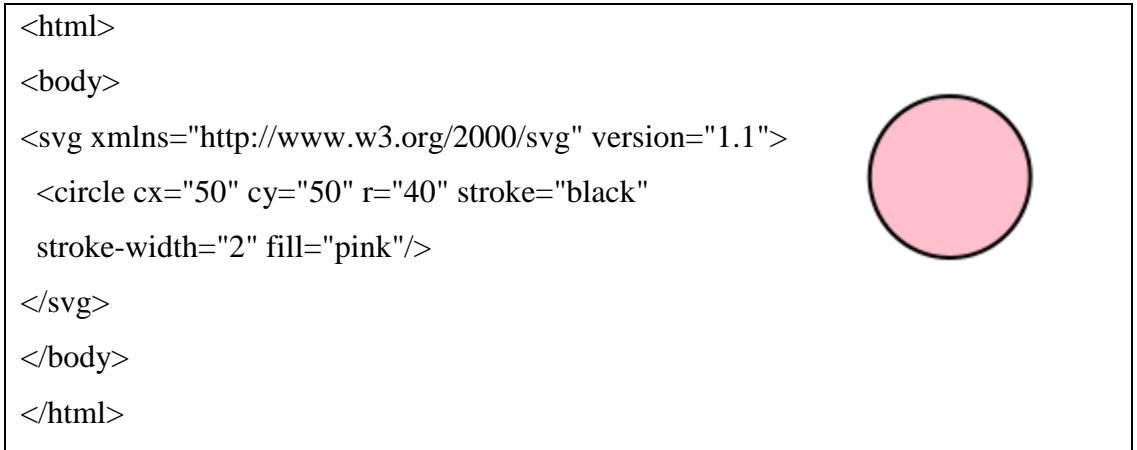
Şekil 5-2.SVG ile JPEG/GIF arasındaki fark [48]

Şekil 5-3’te ise diğer format türlerinden farklı olarak, parçalardan oluşan şekillerde parçaların ayrı olarak incelenebildiği gösterilmektedir.



Şekil 5-3.SVG formatının diğer formatlardan farkı [48]

Şekil 5-4'te daire çizilmesine yarayan SVG kodu ve tarayıcıda görünecek şekil gösterilmektedir.



Şekil 5-4.HTML içerisine gömülecek SVG kodu ve tarayıcıda görünen şekil

SVG'de kullanıcıların daha rahat kullanabilmeleri için hazır tanımlanmış şekiller vardır. Bu şekiller:

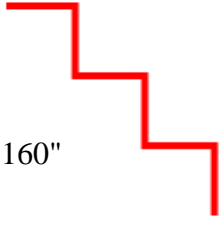
- Dikdörtgen <rect>
- Daire <circle>
- Elips <ellipse>
- Doğru <line>

- Çoklu doğru <polyline>
- Çokgen <polygon>
- Yol <path>

Uygulamada kullanıcıların kaç harekette karakter yazdığı bilgisi <polyline> etiketi ile belirlenmektedir. “polyline” elementi sadece doğrulardan oluşan bir şekil oluşturulmasını sağlar. Koordinatları verilen noktaların birleştirilmesiyle oluşturulan doğruların dolgu rengi, vurgu kalınlığı gibi stillerinin hazırlanması da bu elementle gerçekleştirilebilir. Birden fazla “polyline” elementi ile şekiller çizilebilir, her birine ayrı stiller uygulanarak birbirinden ayrı olarak değerlendirilmesi sağlanabilir.

Şekil 5-5’te gösterilen <polyline> elementinin kullanımıyla oluşturulan SVG kodu ve tarayıcıda görünecek şekil gösterilmektedir.

```
<!DOCTYPE html>
<html>
<body>
<svg height="180" width="500">
  <polyline points="0,40 40,40 40,80 80,80 80,120 120,120 120,160"
  style="fill:white;stroke:red;stroke-width:4" />
</svg>
</body></html>
```



Şekil 5-5.<polyline> elementi ile SVG dosyası kodu

5.1.3 Tesseract OCR Yazılımı

Tesseract doktora araştırma projesi olarak başlayan, 1984 ve 1994 yılları arasında HP tarafından geliştirilen, açık kaynak kodlu OCR motorudur [49]. 2005'de HP, Tesseract'ı açık kaynak kodlu OCR motoru olarak ilan etmiştir.

5.1.3.1 Tesseract'ın Mimarisi

İşlemler adım adım incelenir. İlk adım bileşenlerin ana hatlarına bağlı bileşenin analizidir. En önemli avantajı taslakların denetlenmesidir. Bu şekilde metin terse çevrilebilir ve beyaz üzerinde siyah metin haline kolaylıkla getirilebilir.

Tesseract siyah üzerine beyaz metin kullanan ilk OCR motorudur. Bu aşamada, anahatlar bir araya getirilir ve satırlarda analiz edilmiş metinlerin bölgelere ayrılmasıyla oluşan damlaların (blob) içine yerleştirilir. Satırlar karakterlerin boşluklarına göre kelimelere ayrılır. Sabit eğimli metinler karakter hücreleri tarafından bölünür. Oranlı metinlerde ise boşluklar kullanarak kelimeler bölünür.

Tanıma işlemi iki adımdan oluşur. İlk adım, her kelimenin tanımlanması işlemidir. Her kelime eğitim setine göre uygun sınıflandırıcıdan geçer. Uygun sınıflandırıcı metnin daha doğru bir şekilde tanımlanması ihtimalini artırır.

İkinci aşamada kelimeler tekrardan gözden geçirilir ve gerekirse yeniden tanımlanır. Son aşama kesin olmayan boşlukları ve büyük küçük harfleri kontrol ederek çözümlene yapar.

5.1.3.2 Satır Bulma

Tesseract'ın satır bulmak için bir kaç farklı algoritması vardır. Satırı bulma algoritması eğriliği kaldırmadan sayfadaki eğriliğin farkına varılmasını sağlayacak şekilde tasarlanmıştır. İşlemin anahtar noktaları ise damla filtreleme ve satır oluşturmaktır.

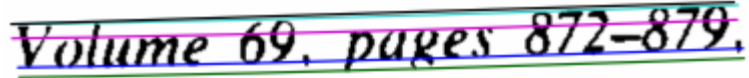
Dikey olacak şekilde harfler birbirinden ayrılır. Orta yükseklik yaklaşık olarak metin boyutunun alanı kadardır, bu sayede filtrelenen damlalar orta yüksekliğin bazı kırınımlardan daha küçüktür. Bunların çoğunluğu ya noktalama işaretleridir ya da gürültülerdir.

x koordinatları sayfa üzerindeki eğimi yakalamaya yardım ederek damlaları sıralamayı ve işlemeyi mümkün kılar. Filtrelenen damlalar, ortanın en küçük değeri alınarak damla kareleri oluşturulur, filtrelenen damlalar satırlara göre ayarlanır. Son basamakta ise satırın tanımlanmasında damlalar birleştirilir [51].

5.1.3.3 Eğriliği Düzeltme

Metin satırı bulunduktan sonra, eğim ikili şerit kullanarak daha hassas duruma getirilir.

Damlalara ayrılarak eğim ayarlanır. İkili şeridin avantajı, hesaplamının sabit olmasıdır. Dezavantajı ise, şeritler artırılması gerektiğinde devamlılığın olmamasıdır. Geleneksel kübik şerit bu durumda daha verimli çalışır [50].



Şekil 5-6.Sınırları belirlenmiş bir metin örneği [51]

Şekil 5-6'da eğimli bir metindeki harflerin boyutlarını, uzantılarını belirleyen şeritler görülmektedir.

Bütün çizgiler paralel ve belli belirsiz eğimlidir. Harflerin üst uzantıları açık mavi olarak gösterilmiştir. Onun üzerindeki siyah çizgi ise tamamıyla düzgündür. Yakından bakıldığında açık mavi renkli şerit, üzerindeki siyah çizgiye göre eğimlidir.

5.1.3.3.1 Sabit Aralıkları Tespit Etme Ve Bölme

Metin sabit aralıklı olduğunda, Tesseract alanı kullanarak kelimeyi harflere böler, daha sonra kelime tanıma aşamasına geçilir. Şekil 5-7'de sabit aralıklardan faydalanılarak parçalara ayrılmış bir kelime görülmektedir.



Şekil 5-7. Metnin sabit aralıklarından faydalanılarak bölünen kelime [51]

5.1.3.3.2 Orantıya Göre Kelime Bulma

Harfler arasında sabit olmayan ve orantısız boşluklar karşılaşılan zor problemlerden biridir. Şekil 5-8’de kelimeler arası boşlukları sabit olmayan bir örnek görülmektedir.

**of 9.5% annually while the Fed-
erated junk fund returned 11.9%
*fear of financial collapse,***

Şekil 5-8. Aralıkları orantısız metin örneği [51]

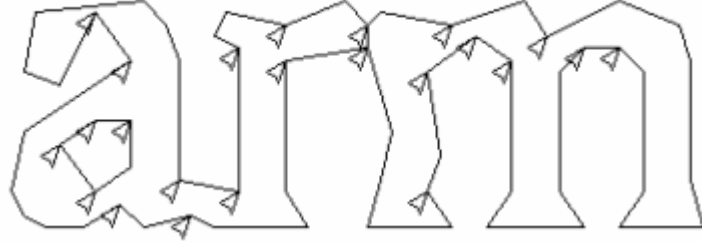
Tesseract, orta ve temel çizgilerin uzaklıkları arasındaki boşluğu ölçerek bu tür problemleri çözer. Bu aşamadaki boşlukların eşik değerleri birbirine yakındır ve son karar kelime tanımlandıktan sonra verilebilir.

5.1.3.4 Kelime Tanıma

Kelime tanıma motorlarının ilk yaptığı işlem, kelimenin harflere nasıl ayrılacağıdır. Satır bulunduktan sonra yapılacak ilk işlem sınıflandırmadır. Kelime tanımda geriye kalan basamak sadece sabit aralıkları olmayan metinlere uygulanır.

5.1.3.4.1 Bitişik Harfleri Birbirinden Ayırma

Tesseract harf sınıflandırmasında damlalara bölerek sonuçları geliştirir. Aday bölme noktaları, iç bükey köşelerinden bulunur. Çoğunlukla karşılıklarında da içbükey köşeler ya da satır segmentleri bulunur. Birleşik harflerin başarılı bir şekilde bölmek için genellikle 3 bölme noktası kullanılır. Şekil 5-9'da aday bölme noktaları oklar ile gösterilmiştir ve 'r' nin 'm' ye dokunduğu noktadaki çizgi, ayrılması gereken yer olarak seçilmiştir.



Şekil 5-9.Aday bölme noktaları [51]

Bölmeler ön işlemlerden geçirilirler. Başarısız olarak sonuçlanan herhangi bir bölme tamamlanmamış sayılır ama tekrar kullanılmak üzere ayrılır.

5.1.3.4.2 Eksik Parçaları İlişkilendirme

Eğer kelimeler yeterince düzgün değilse, “ilişkilendiriciler” kullanılır [50]. İlişkilendiriciler, bölme işlemi yapılmış aday olabilecek karakterler içerisinde mümkün olanları seçerek arama yapar. Şekil 5-10'da Tesseract tarafından kolaylıkla tanımlanabilen bir metin gösterilmektedir.

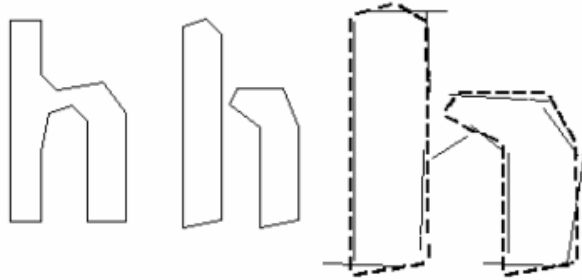


Şekil 5-10. Tanımlanabilecek örnek metin [51]

5.1.3.4.3 Özellik Çıkarımı

Tanımlar, karakterlerin boyutları ve yazı tiplerinden bağımsız olarak yapılabilmektedir.

Şekil 5-11'e bakıldığında bilinmeyen iki parça halindeki karakterin tanımlanmasında eğitim setindeki karakterle birebir eşlenmesi gerekmemektedir. Eğitim sırasında karakterlerin belirli özellikleri alınır. Alınan bu özellikler göz önünde bulundurularak bilinmeyen karakter ile eğitim setindeki karakter üst üste koyulur. Şekil 5-11'de bu iki karakter ile eşleştirilen karakter gösterilmektedir.



Şekil 5-11. Farklı yazılarla tanımlanan karakterler [51]

Aradaki fark fazla değilse, bunların benzer olduğu kanısına varılır. Dezavantaj ise, bilinmeyen karakter ile eğitim setinden alınan karakterin eşleştirilmesi sırasında farkın büyük olmasıdır. Eşleşmenin olmamasına neden olur.

5.1.3.4.4 Sınıflandırma

Sınıflandırmanın iki basamağı vardır. Tanımlanacak olan metnin oluşturulmuş bir karakter listesi sınıfıyla eşleşmesi gerekir. Her bir özelliğin 3 boyutu (x-y koordinatları, açı) vardır ve bu vektör bitleriyle sınıflara eşleştirme yapılır. Bütün özelliklerin vektör bitleri toplanır, sınıfların en yükseği ile bir sonraki basamakta kullanılacak liste hazırlanır.

Tanımlanacak olan metnin her bir özelliği prototipin vektör bitleri ile karşılaştırılır ve aralarındaki benzerlik miktarı ölçülür. Uzaklık hesaplamasında toplam benzerlik göz önüne alınır.

5.1.3.4.5 Veri Eğitimi

Sınıflandırıcı eksik, hasarlı verilerle eğitilmez. Eğer sağlıklı verilerle sistem eğitilmezse alınan sonuçlar var olandan daha kötü olabilir. Mümkün olduğunca eğitim setleri düzgün ve tanımlanması zor olmayan metinler olmalıdır. Sınıflandırıcı her bir yazı boyutundan 8 yazı tipinden 94 karakter ile eğitim yapabilmeyi desteklemektedir. Ayrıca normal, kalın, italik, kalın italik gibi özelliklerde eklendiğinde 60160 eğitim verisi oluşturulabilir [51].

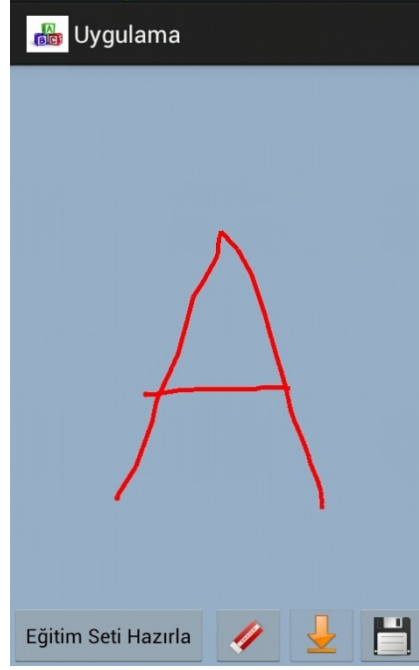
5.2 Uygulamanın İçeriği

Uygulama, öğrenme zorluğu çeken çocukların zorluk çektikleri sesleri yazarak sorunlarını düzeltilebileceğinden yola çıkarak teknik olarak el yazısı tanıma işlemi üzerinde yoğunlaşmaktadır. Mobil Android işletim sistemi üzerinde çalışan uygulamada, kullanıcı mobil cihaz üzerinde el yazısı ile yazılanın makine karakterlerine dönüşmesini sağlamaktadır. A-Z'ye ve 0-9'a kadar tanımlanmış özel bir eğitim seti kullanılan uygulamanın, eğitim seti verisi toplamak için ayrı bir arayüzü bulunmaktadır.

Uygulamayı kullanmaya başlayan kullanıcıların verileri birbirinden ayrı olarak bulunmakta, hangi karakteri nasıl ve kaç harekette yazdığı verisi, tarih ve saat bilgileriyle birlikte saklanmaktadır. Dolayısıyla kullanıcı bilgilerine sunucu üzerinden ulaşmak, üzerinde incelemeler yapmak, yanlış şekilde yazılan karakterlerin tespit edilmesi daha da kolaylaşmaktadır. Ayrıca uygulama ile çalışan öğrencin, klasik yöntemlerde olması gereken eğitimciye ihtiyacı olmadan çalışmasını gerçekleştirebilecektir.

5.3 Sistem Mimarisi ve Uygulamanın Adımları

Sistem istemci-sunucu mimarisinde tasarlanmıştır. Kullanıcının uygulamayı kullanması ile birlikte oluşmaya başlayan SVG dosyası kullanıcının parmağını ekrandan kaldırıp tekrar koymasıyla şekillenir. Her dokunuşta SVG dosyasına piksellerin x-y koordinatları kaydedilir. Bu işlem yazma işleminin sunucuya gönderilmesine kadar sürer. Şekil 5-12’de görüldüğü üzere “A” harfi iki hareket ile yazılmıştır. Bu yüzden Şekil 5-13’de görülen SVG kodunda iki adet <polyline> etiketi arasında dokunulan piksellerin koordinatları bulunmaktadır. Bu şekilde kullanıcının yazım stili hakkında da bilgi edinilebilir.



Şekil 5-12.Uygulama üzerinde el yazısı denemesi

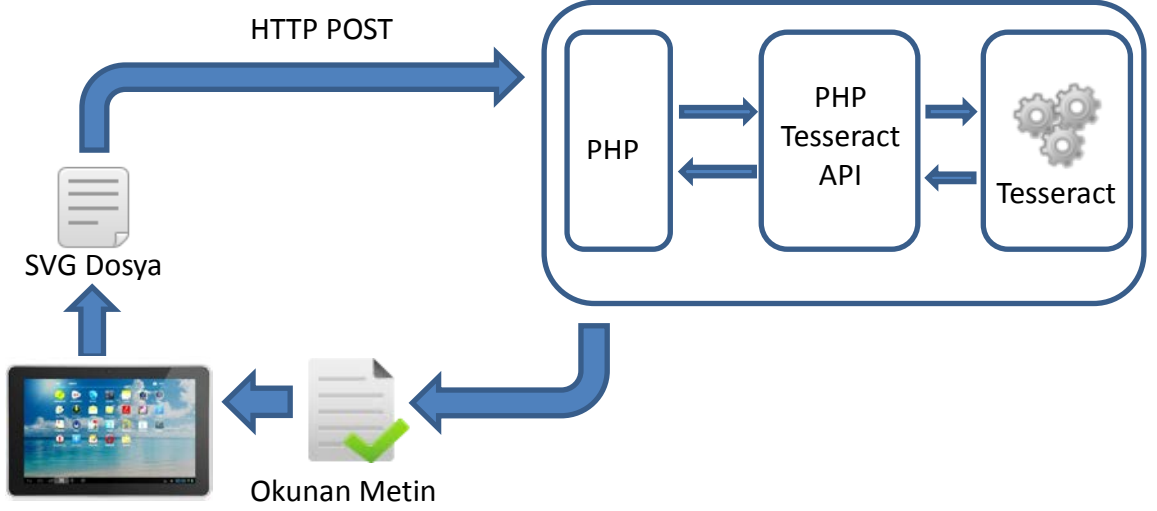
```
<!DOCTYPE html><svg xmlns="http://www.w3.org/2000/svg"
version="1.1"><polyline points=" 122.74428,496.24847
122.74428,493.2522 143.70062,460.2934 162.66112,408.35834
166.6528,388.38327 169.64658,379.39453 185.61331,345.43695
192.59875,329.45694 196.59044,314.47568 201.58005,296.49814
209.56342,265.53683 220.54054,248.55804 235.50935,221.59177
237.50519,217.59677 238.50311,203.61423 239.50104,197.62173
239.50104,196.62299 240.49896,194.62549 241.49689,191.62921
241.49689,189.63171 241.49689,190.63046 243.49272,192.62796
260.45737,210.6055 276.4241,239.5693 292.39084,287.50937
294.3867,295.4994 299.37628,313.4769 306.36176,337.44696
308.35757,344.4382 310.35342,350.43073 316.34094,371.4045
317.33887,378.39575 318.3368,381.39203 321.33057,395.37454
324.3243,404.3633 326.32016,409.35706 330.31186,418.34583
332.30768,423.3396 337.2973,435.3246 342.2869,449.30713
343.28482,455.29962 345.28067,459.29462 348.2744,467.28467
350.27026,471.27966 352.2661,476.27344 354.26196,483.2647
356.25778,488.25842 356.25778,492.25342 357.2557,503.2397
357.2557,504.23846 357.2557,507.23474 357.2557,508.23346
357.2557,508.23346" style="fill:none;stroke:red;stroke-
width:5"/><polyline points=" 155.67567,377.397 156.6736,377.397
159.66736,377.397 163.65904,377.397 168.64865,376.39825
187.60915,373.402 192.59875,372.40326 196.59044,372.40326
213.55508,371.4045 217.54678,371.4045 223.5343,371.4045
227.52599,371.4045 231.51767,371.4045 233.51352,371.4045
241.49689,371.4045 244.49065,371.4045 252.47401,371.4045
261.4553,371.4045 264.44907,371.4045 270.43658,371.4045
273.43036,371.4045 278.41995,371.4045 287.40125,369.407
290.39502,369.407 306.36176,369.407 313.3472,369.407
316.34094,369.407 317.33887,370.40576 319.33472,370.40576
320.33264,370.40576 320.33264,370.40576"
style="fill:none;stroke:red;stroke-width:5"/> </svg>
```

Şekil 5-13.Yazılan “A” harfinin SVG kodu

Hazırlanan SVG dosyası eğer internet bağlantısı aktifse sunucuya gönderilir. Sunucuda hazırlanan PHP sayfası ile farklı bir uygulamadan gelip gelmediği, gelen dosyanın uzantısının kontrolü yapılır. SVG dosyasının tarayıcı üzerinde görülen hali, kullanıcının mobil cihazda yazdığı şeklindedir. Tesseract ise statik yazıları piksel tabanlı olarak tanımaya yarayan OCR sistemidir. Burada farklı olarak yapılan ise, dinamik olarak kullanıcıdan alınan verilerin piksel tabanlı olarak sunucuda kurulu olan Tesseract’a işleme sokulmasıdır.

Uygulama Tesseract’ın kullanımı açısından çevrimdışı, kullanıcıdan verilerin dinamik olarak alınmasıyla, kaynak verinin basılı ya da makine yazısı olmaması, el yazısı olmasıyla, anlık olarak tanıma yapılabilmesiyle çevrimiçi el yazısı tanımına girmektedir. Tesseract’ta alınan karakterin ya da metinlerin tanımlanmış halleri

uygun formatta cihaza tekrar gönderilir. Şekil 5-14’de uygulamanın sistem mimarisi görülmektedir.



Şekil 5-14.Sistem mimarisi

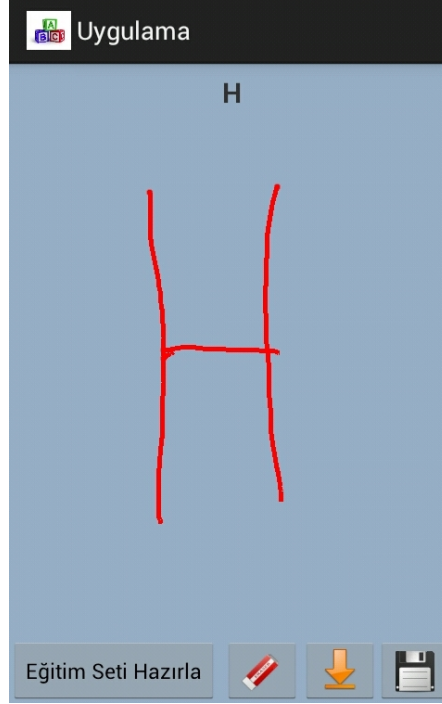
Sunucuda uygulamayı kullanan her cihaz için ayrı bir klasör bulunmakta, tarih ve saat bilgisiyle cihazlardan alınan SVG dosyaları, bu SVG dosyalarının ne olarak tanımlandıkları saklanmaktadır.

Eğitim seti hazırlarken A-Z’ye ve 0-9 arasındaki rakamların girilmesiyle yine kullanıcıların kendi klasörleri içinde eğitim seti verileri saklanmaktadır. Bu sayede kişilere özel eğitim setleri tanımlanabilir.

5.4 Kullanıcı Arayüzü

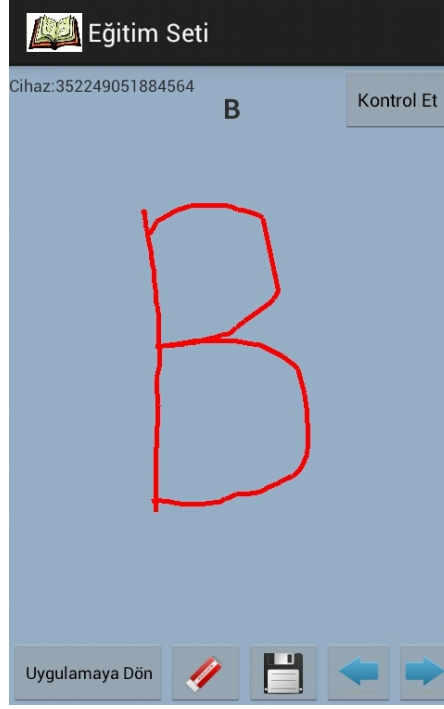
Şekil 5-15’te uygulama arayüzü, Şekil 5-16’da ise eğitim seti için veri toplama ekranı görülmektedir. Uygulama ekranının sağ alt kısımda görünen disket resimli buton ile eğer internet bağlantısı varsa sunucuya hazırlanan SVG dosyası gönderilmektedir. Yoksa kullanıcıya bağlantının olmadığı bilgisi verilir. Ok resmi olan buton, sunucuya gönderilmiş el yazısının tanımlanmış halini ekranın üst kısmına yazdırır. Tekrar yazabilmek için üzerinde silgi resmi olan bir buton da bulunmaktadır. Eğitim seti verisi oluşturmak içinse sol altta “Eğitim Seti Hazırla”

butonu bulunmaktadır. Bu butonla kişiye özel eğitim seti oluşturabilmek için farklı bir arayüze geçirmektedir.



Şekil 5-15.Uygulama arayüzü

Şekil 5-16’da gösterilen eğitim seti arayüzü uygulama arayüzüyle benzerdir. Farklı olarak veri girişi için A-Z ve 0-9 arası harfler ve rakamlar bulunmakta ve bunlar arasında ilerlemek, geri dönebilmek için ok tuşları bulunmaktadır. Her bir harf ya da rakam girişinden sonra kayıt yapılmalı ve bir sonraki karaktere geçilmelidir. El yazısının sunucu tarafında nasıl çevrildiğini “Kontrol Et” butonu ile öğrenilebilir.



Şekil 5-16.Eğitim seti için veri toplama arayüzü

5.5 Eğitim Seti Hazırlama

Başka bir dilde eğitim için, Tesseract'ın *tessdata* alt dizininin altında bazı dosyalar oluşturulmalıdır ve daha sonra *combine_tessdata* kullanılarak, bunlar tek bir dosya haline getirilmelidir. *dilkodu.dosya_adi* şeklinde adlandırılmalıdır. Dil kodları ISO 639-3 standartlarında yayınlanmıştır. Türkçe için kullanılan dosyalar:

- *tessdata/tr.config*
- *tessdata/tr.unicharset*
- *tessdata/tr.unicharambig*s
- *tessdata/tr.inttemp*

- tessdata/tr.pffmtable
- tessdata/tr.normproto
- tessdata/tr.punc-dawg
- tessdata/tr.word-dawg
- tessdata/tr.number-dawg
- tessdata/tr.freq-dawg [52]

Sonunda tessdata/tr.traineddata dosyası oluşturulur. Öncelikle karakterlerin tümünü içeren bir eğitim seti hazırlanmalıdır. Eğitim seti hazırlarken dikkat edilmesi gereken noktalar:

- Her bir karakter örneğinden yeteri kadar olmalıdır.
- Sıklıkla kullanılan harflerin en az 20 örneği olmalıdır.

5.5.1 Box dosyası oluşturma

Bir sonraki basamakta, Tesseract *box* dosyasına ihtiyaç duyar. *Box* dosyası, eğitim için kullanılacak görüntüdeki karakterleri listeleyen, her satırdaki karakterlerin sınırlarını koordinatları ile kutu içerisine alan bir metin dosyasıdır. Doğru sonuçlar alınabilmesi için karakter seti doğru karakterlerle oluşturulmalıdır.

Komut satırı kullanarak Tesseract'ın çalışması:

```
tesseract [lang].[fontname].exp[num].tif  
[lang].[fontname].exp[num] batch.no chop makebox
```

Örneğin; maltepe adında bir eğitim seti oluşturulmak istensin. Komut satırına aşağıdaki komut yazılarak “.box” uzantılı dosya elde edilir.

```
tesseract maltepe.timesitalic.exp0.png  
maltepe.timesitalic.exp0 batch.no chop makebox
```

Oluşan *box* dosyası, her bir karakter satır başında olacak şekilde düzenlenmesi gerekmektedir. Oluşturulan dosyanın içinde örnek olarak verilmiş bir görüntüdeki algılanan karakterlerin kutulandıktan sonra sol, sağ, üst, alt ve görüntünün sayfa sayısı Çizelge 5-2’de gösterilmektedir.

Çizelge 5-2.Eğitim seti için karakter özellikleri

Karakter	Sol	Alt	Üst	Sağ	Sayfa sayısı
s	734	494	751	519	0
p	753	486	776	518	0
r	779	494	796	518	0
i	799	494	810	527	0
n	814	494	837	518	0
g	839	485	862	518	0
t	865	492	878	521	0
u	101	453	122	484	0
b	126	453	146	486	0
e	149	452	168	477	0
r	172	453	187	476	0
d	211	451	232	484	0
e	236	451	255	475	0
n	259	452	281	475	0

Bazı durumlarda tanımlanan karakter değerleri doğru karakteri ifade etmeyebilir. Bu yüzden karakterlerin düzenlenebilmesi için uygun bir editör kullanılmalıdır.

Dosya oluşturulurken her karakter kutulanır. Çizelge 5-3'te çift tırnak işaretinin ayrı ayrı işleme tabi tutulduğu gösterilmektedir. Bunun için bu işaret gibi istisna işaretleri eğiten düzenlemelidir. Düzenlerken ayrı olarak kutulanan çift tırnak işaretinin koordinatları birleştirilir.

Çizelge 5-3.Düzenlenen karakter seti

Karakter	Sol	Alt	Sağ	Üst	Sayfa sayısı
D	101	504	131	535	0
e	135	502	154	528	0
r	158	503	173	526	0
,	197	498	206	510	0
,	206	497	214	509	0
s	220	501	236	526	0
c	239	501	258	525	0
h	262	502	284	534	0
n	288	501	310	525	0
e	313	500	332	524	0
l	336	501	347	534	0
l	352	500	363	532	0
e	367	499	386	524	0
”	389	520	407	532	0

İlk numara (sol sınır için) iki satırın minimum değerini alır. (197)

İkinci numara (alt sınır için) iki satırın minimum değerini alır. (497)

Üçüncü numara (sağ sınır için) iki satırın maksimum değerini alır. (214)

Dördüncü numara (üst sınır için) iki satırın maksimum değerini alır. (510)

En son düzenlemelerden sonraki karakter değerleri Çizelge 5-4'te gösterilmektedir.

Çizelge 5-4.Karakter setlerinin tekrar düzenlenmesi

Karakter	Sol	Alt	Sağ	Üst	Sayfa sayısı
D	101	504	131	353	0
e	135	502	154	528	0
r	158	503	173	526	0
„	197	497	214	510	0
s	220	501	236	526	0
c	239	501	258	525	0
h	262	502	284	534	0
n	288	501	310	525	0
e	313	500	332	524	0
l	336	501	347	534	0
l	352	500	363	532	0
e	367	499	386	524	0
”	389	520	407	532	0

5.5.2 Yeni Karakter Seti Yükleme

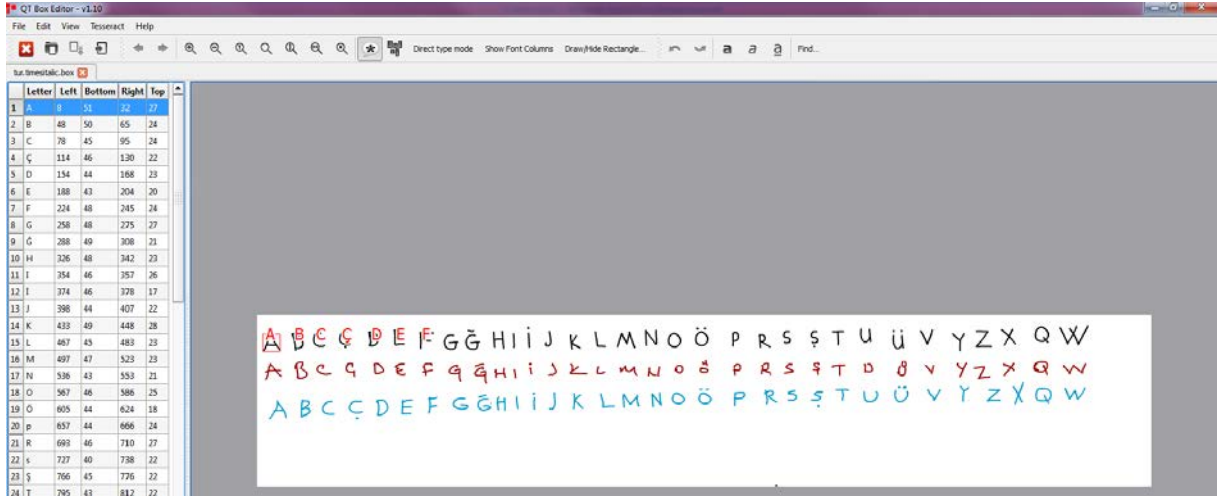
Eğer yeni bir karakter seti eğitilecekse, tek bir fonttan oluşan iyi bir *box* dosyasını kullanmak doğru olacaktır. Aşağıda bunun için kullanılması gereken komut bulunmaktadır.

```
tesseract [lang].[fontname].exp[num].png  
[lang].[font].exp[num] -l yenedilinadi batch.no chop makebox
```

Birden fazla *box* dosyası oluşturulabilir ve eğitim setine daha fazla font eklenebilir. Ancak varolan sete eğitilen yeni set eklenemez. Bunun anlamı her seferinde yeniden *mfTraining*, *cnTraining* çalıştırılmalı ve *tr* uzantılı dosyadan yeni veri dosyaları yapılmalıdır. *intproto/pffmtable/normproto* dosyaları da direkt olarak eklenemez.

Tesseract tarafından otomatik oluşturulan *box* dosyası içerisinde, eğitim için kullanılacak görüntüdeki karakterlerin düzgün olarak tanımlanamaması yahut düzgün kutulanamaması dolayısıyla yanlış *box* dosyaları oluşturulabilir. Dolayısıyla bu da eğitim setlerinin düzgün çalışmamasına neden olur ve tanımların verimini de etkiler. *Box* dosyasının daha düzgün oluşturulabilmesi amacı ile QT Box Editor programı kullanılabilir. Bu program ile eğitim verisi tek dosya üzerinden verilmelidir. Kullanıcı program arayüzünden hangi karakterin nasıl tanımlandığını, nasıl kutulandığını görebilmektedir. Yanlış tanımlanıp kutulanan veriler, kullanıcı tarafından elle düzeltilebilmekte, daha doğru *box* dosyaları oluşturulabilmekte ve tanımların verimi arttırılabilmektedir.

Şekil 5-12’de QT Box Editor arayüzü görülmektedir. Soldaki menüde eğitim setinde kullanılmak üzere el yazısı karakterlerinin alt, üst, sağ ve sol değerleri bulunmaktadır. Ayrıca karakterlerin üzerine tıklandığında karakterinin sınırlandıran kutu ve sistem tarafından ne olarak algılandığı gösterilmektedir. Eğer karakter yanlış tanımlanmışsa, soldaki menüden elle düzeltilebilmektedir.



Şekil 5-17. *box* dosyası oluşturan QT Box Editor arayüzü

5.5.3 Tesseract'ı Eğitime

Aşağıdaki komut, komut satırına yazılmalıdır.

```
tesseract [lang].[fontname].exp[num].png
[lang].[fontname].exp[num] nobatch box.train
```

Tesseract'a bu adım için varolan bir dil eklenmelidir. Eğer dil eklenmemişse İngilizce varsayılır. Aşağıda Türkçe eğitim setinin kullanımı için kullanılan komut bulunmaktadır.

```
tesseract [lang].[fontname].exp[num].png -l
tur.[fontname].exp[num] nobatch box.train

tesseract maltepe.timesitalic.exp0.png -l tur
maltepe.timesitalic.exp0 nobatch box.train
```

apply_box çıktılarında çıkan hataları kontrol etmek önemlidir. Eğer "FATALITIES" raporlanmışsa *box* dosyası düzeltilene kadar eğitime işlemi

yapılmamalıdır. FATALITY genellikle *box* dosyasında listelenen karakterlerden herhangi birinin örneğinin bulunamadığını gösterir. Koordinatların ya da ilgili karakterin görüntüsünde bir sorun olmasıyla alakalı olabilir. Eğer çalışabilecek bir karakter yoksa, tanımlanamamışsa, *inttemp* dosyası *unicharset* dosyasıyla uyuşmayacak, dolayısıyla Tesseract çalışmayı bırakacaktır.

Bir diğer sorun ise zararlı ve dikkat edilmesi gereken “Box file format error on line n” hatasıdır. Eğer öncesinde “Bad utf-8 char...” varsa, o zaman utf-8 kodları hatalıdır ve düzeltilmesi gerekmektedir. Hata “utf-8 string too long...” ise karakter tanımının 24 byte ı aştığını göstermektedir.

5.5.4 Karakter Setini Sayma

Tesseract çıkabilecek mümkün karakter sayısını bilmek ister. Bu yüzden *unicharset_extractor* komutunu kullanarak *unicharset* dosyası oluşturulur. Oluşturmak için kullanılan komut aşağıda gösterilmiştir.

```
unicharset_extractor maltepe.timesitalic.exp0.box  
  
unicharset_extractor lang.fontname.exp0.box
```

Tesseract büyük-küçük harf, rakam ya da harf gibi karakter bilgilerine erişebilir. Bu veriler *unicharset* dosyasının içinde şifrelenmiştir. Dosyadaki her satır bir karakteri temsil etmektedir. Her karakterin özelliği UTF-8 formatına göre ikilik sistemle gösterilir ve özellik şifrelenir. Her bit bir özelliği temsil eder ve eğer değeri bir ise o özellik doğru demektir. Alfabetik olma, rakam olma, büyük ya da küçük harf olma durumu bu özelliklerdendir.

Örneğin;

“;” noktalama işareti karakteridir. İkilik sistemde 10000 olarak gösterilir. (onaltılık sistemde 10 olarak gösterilir.)

“b” alfabetik bir karakterdir ve küçük harflidir. Bu özellikler ikilik sistemde 00011 olarak gösterilir (onaltılık sistemde 3 olarak gösterilir).

“7” rakamdır. Bu özelliği ikilik sistemde 01000 olarak gösterilir. (onaltılık sistemde 0 olarak gösterilir.)

“=” ne alfabetik, ne de rakamsal, ne de noktalama işareti karakteridir. Bu özellik ikilik sistemde 00000 olarak gösterilir. (onaltılık sistemde 0 olarak gösterilir.)

Çizelge 5-5'te örnek bir unicharset dosyasında karakterlerin ikilik gösterimi, hangi yazı tipinden olduğu ve bulunduğu tipe göre id'sinin ne olduğu gösterilmiştir.

Çizelge 5-5.unicharset içeriği

Karakter	İkilik Gös.	Yazı Tipi	Tipe göre karakter id'si
;	10	Common	46
b	3	Latin	59
W	5	Latin	40
7	8	Common	66
=	0	Common	93

inttemp, *normproto* ve *pfmtable* oluşturulduğunda *unicharset* dosyası tekrar oluşturulmalıdır. *Box* dosyası değiştirildiğinde ise bütün dosyalar tekrardan oluşturulmalıdır.

5.5.5 Font Özellikleri

Font özelliklerinde değişiklikler yapılmasını sağlayan *font_properties* dosyası eğitim için kullanılmaya başlanmıştır.

Font_properties dosyası aşağıdaki gibi formatlandırılır.

```
<fontname><italic><bold><fixed><serif><fraktur>
```

Sadece *<fontname>* kısmı stringdir. Diğer özellikler 0 ya da 1 değerini alırlar.

Mftraining dosyası çalıştığında, her bir “.tr” dosyası *font_properties* dosyasındaki girişlere uygun olmalıdır yoksa *mftraining* dosyası çalışmaz.

Örneğin; *font_properties* dosyasının aşağıdaki gibi olduğu varsayılırsa:

```
Timesitalic 1 0 0 1 0
```

Komut satırına aşağıdaki iki komut yazılır:

```
shapeclustering -F font_properties -U unicharset  
lang.fontname.exp0.tr
```

```
mftraining -F font_properties -U unicharset -O  
lang.unicharset lang.fontname.exp0.tr
```

```
shapeclustering -F font_properties -U unicharset  
maltepe.timesitalic.exp0.tr
```

```
mftraining -F font_properties -U unicharset
maltepe.timesitalic.exp0.tr
```

5.5.6 Kümeleme

Karakter özellikleri çıkarıldıktan sonra, prototipleri oluşturmak için kümeleme gerekmektedir. *shapeclustering*, *mftraining* ve *cntraining* komutlarını kullanarak karakter özellikleri kümelenebilir.

```
shapeclustering -F font_properties -U unicharset
lang.fontname.exp0.tr
```

shapeclustering ile master şekil tablosu oluşturulur ve *-shapetable* dosyasına yazılır.

```
shapeclustering -F font_properties -U unicharset
maltepe.timesitalic.exp0.tr
```

```
mftraining -F font_properties -U unicharset
lang.fontname.exp0.tr
```

```
mftraining -F font_properties -U unicharset
maltepe.timesitalic.exp0.tr
```

-U dosyası, *unicharset_extractor* komutu ile oluşturulur ve çıktısı *combine_tessdata* komutuyla birlikte kullanılır. *Mftraining* iki dosya oluşturur:

- inttemp (prototiplerin şekilleri)
- pffmtable (her bir karakterden beklenen özelliklerin sayısı)

cntraining lang.font.name.exp0.tr

Bu komut *normproto* dosyasını oluşturur (karakter normalizasyonu yapılmış hassas prototipler).

5.5.7 Sözlük Verisi

Tesseract her bir dil için 8 sözlük verisine kadar izin verir. Dosyaların yedisi Direk Döngüsüz Kelime Grafi (Directed Acyclic Word Graph - DAWG) olarak şifrelenir ve diğer dosya UTF-8 dosyadır. Çizelge 5-6'da sözlük dosyası için kullanılacak kullanılacak kelimeler ve açıklamaları bulunmaktadır.

Çizelge 5-6.DAWG sözlük dosyası için kullanılan kelimeler

Adı	Tipi	Tanımı
punc-dawg	dawg	Kelimeler arasındaki noktalama örüntüleriyle bir dawg oluşturur. Her kelime boşluk karakteri ile yer değiştirir.
word-dawg	dawg	Dildeki sözlük kelimelerinden bir dawg oluşturur.
number-dawg	dawg	Rakam içeren simgelerden bir dawg oluşturur. Her rakam boşluk karakteri ile yer değiştirir.
fixed-length-dawgs	dawg	Word-dawg'da da bulunan en sık kullanılan kelimelerden bir dawg oluşturulur.
bigram-dawg	dawg	Birbirinde farklı sabit uzunluklarda birkaç dawg oluşturulur.– Çince gibi dillerde kullanışlı olur
unambig-dawg	dawg	Kelimeler boşluk ile ayrılır ve her bir basamak ? ile yer değiştirilir.

DAWG sözlük dosyalarını yapmak için öncelikle kelime listesi hazırlanmalıdır. Kelime listesi her satıra bir kelime gelecek şekilde UTF-8 metin dosyası formatında hazırlanmalıdır. DAWG dosyaları “frequent words, the rest of the words” gibi

kümelerle birbirinden ayrılmalıdır. Aşağıda dosyanın nasıl hazırlanacağı gösterilmiştir.

```
Wordlist2dawg frequent_words_list lang.freq-dawg  
lang.unicharset
```

```
Wordlist2dawg word_list lang.word-dawg lang.unicharset
```

Sağdan sola yazılan dillerde (RTL) “-r l” kullanılmalıdır.

Kelime listeleri en az bir kelime içermek zorundadır. Eğer kelimeler noktalama işaretleri içeriyorsa (örneğin; maltepe.edu.tr) bunları sözlüğe eklemek mantıklıdır.

5.5.8 Belirsiz Karakterler

Oluşturulacak son dosya “*unicharambig*s” olarak adlandırılır. Karakter ve karakter kümeleri arasında karışıklığa neden olabilecek karakterlerden bir dosya oluşturulur. Çizelge 5-7’deki ilk sütun eşleştirilecek kaynak karakter sayısını, ikinci sütun eşleştirilecek kaynak karakterleri, üçüncü sütun eşleştirilecek hedef karakter sayısını, dördüncü sütun hedef karakterleri ve son olarak beşinci sütun tip belirleyicisini göstermektedir.

Çizelge 5-7.*unicharambig*s dosyasının içeriği

3	И0	2	uo	3
3	I-I	1	H	2
2	"	1	'	1
2	к06	1	к0e	1
1	m	2	rn	0
3	iii	1	m	0

Karakterlerin tanımlanabilmeleri için, her bir karakter *unicharset* dosyasında bulunan birkaç satırın ilk bölümleriyle eşleşmek zorundadır. Çizelge 5-8’de *unicharambig* dosyasındaki tip belirleyicisinin aldığı değerlerin ne anlama geldiği gösterilmektedir.

Çizelge 5-8.Tip belirleyicisinin değerleri

Değer	Tip	Tanım
0	NOT_AMBIG	Karakter çifti belirsiz değil
1	REPLACE_AMBIG	Doğru karakterler yanlışın yerine geçer.
2	DEFINITE_AMBIG	Sınıflandırma sonucunda doğru karakteri ekleme
3	SIMILAR_AMBIG	Çifti sınıflandırma
4	CASE_AMBIG	Belirsizlik durumu

Eğer beşinci sütun 1 ise, yerine geçme kuralına göre dördüncü sütündeki karakter ikinci sütündeki karakterin yerine geçer. Beşinci sütunda 0 var ise, yerine geçme önemsizdir. Beşinci bölümün 0 olmasının en önemli faydası yanlış tanımların azaltılmasıdır.

Çizelge 5-7’de verilen örneğe bakılırsa; çift tırnak, peş peşe gelen iki tek tırnağın yerini almalıdır. Çünkü beşinci sütunda yer değiştirmenin kullanılmak zorunda olduğu görülmektedir. Diğer bir satırda “m” çifti bazen yanlışlıkla “m” olarak algılanabilmektedir. Diğer satırda “m” harfi “iii” karakter dizisi olarak algılanabilir. “œ “ karakteri “œ 6” olarak algılandığından ve tip belirleyicisi 1 olduğundan yer değiştirme yapılmak zorundadır. *unicharambig* dosyası her zaman boş satırla bitmelidir.

5.5.9 Bir Araya Toplama

Oluşturulan bütün dosyaları (*shapetable*, *normproto*, *pfmtable*, *inttemp*) bir klasörün altına toplanarak, ilgili dizine gidip aşağıdaki komut yazılır:

```
combine_tessdata lang.
```

“*lang.*” kısmı hazırlanacak olan dilin adı olacaktır. Komut aşağıdaki gibi yazılacaktır:

```
combine_tessdata maltepe.
```

Klasör içinde “*maltepe.trainneddata*” adında bir dosya oluşur ve oluşan bu dosya Tesseract’ın kurulduğu dizin altındaki “*tessdata*” klasörünün altına koyulursa tıpkı bir dil gibi kullanılabilir. Bu sayede farklı girişlerden oluşan ya da var olan dili daha özelleştirilmiş şekliyle yeni bir dil oluşturulabilir. Hatta kişiye özel “*.trainneddata*” dosyaları oluşturularak el yazılarının tanımlanması da kişiselleştirilebilir.

6 SONUÇLAR VE DEĞERLENDİRME

Bu bölümde tez doğrultusunda hazırlanan uygulama ile yapılan denemelerin sonuçlarına, bu sonuçların değerlendirilmesine ve ileriki çalışmalarda neler yapılabileceğine yer verilmiştir.

6.1 Elde Edilen Sonuçlar

3 farklı deneğin yazdıkları harf ve rakam örnekleri ile yeni bir eğitim seti oluşturulmuş ve Tesseract'ın kendi içinde gelen Türkçe dili eğitim setinin performansları karşılaştırılmıştır. Tesseract'ın eğitim seti kullanılarak gerçekleştirilen karakter tanıma testinde toplam 39 karakterden 11'i yanlış tanımlanmıştır. Bu değer yaklaşık olarak %72 oranında doğru tanıma yapıldığını göstermektedir. Bu çalışma kapsamında oluşturulan yeni eğitim seti kullanıldığında 39 karakterden 4'ü hatalı tanımlanmıştır. Bu da yaklaşık olarak %90 oranında doğru tanıma yapıldığını göstermektedir. Sonuç olarak, parmak ile yazılmış harfler kullanılarak Tesseract yazılımı eğitilmiş ve mobil öğrenme uygulamasının doğruluğu arttırılmıştır.

6.2 Değerlendirme

Geliştirilen mobil öğrenme uygulaması, harfleri veya genel anlamda yazı yazmayı öğrenmede zorluk yaşayan öğrencilerin daha eğlenceli, görsel ve ölçülebilir yöntemler kullanarak eğitilmesine katkı sağlayacaktır. Eğitim alanı hedeflenerek tasarlanan proje, geliştirmeye açık yapısı sayesinde, el yazısı karakter tahlili ve örüntü tanıma gibi başka uygulama alanlarında da kullanılabilir şekilde tasarlanmıştır. Ayrıca her kullanıcının verisinin sunucuda ayrı tutulması ile gerekirse öğrencilerin tarihçesine erişilebilir çalışmaları takip edilebilir, yazım stillerinin incelenebilir, yanlış yazan öğrenciler tespit edilebilir. Tanıma işleminden daha verimli sonuçlar alabilmek için düzgün eğitim setleri kullanmak faydalı olacaktır. Aksi takdirde var olandan daha kötü sonuçlar elde edilebilir.

6.3 Öneriler

Uygulama, yazarak çalışmaya yönelik olduğundan öğrenme zorluğu çeken çocukların yanı sıra, yazmayı yeni öğrenen çocuklara da yardımcı olabilir. Bu durumda yazılan karakterin dinlenilmesi, çalışmanın görsel olmasının yanında işitsel olarak da öğrenciyi desteklemesi öğrenmeyi kolaylaştırabilir. Tesseract üzerinde karakter bazlı olarak çalışan uygulama, isteğe bağlı olarak karakter ya da kelime bazlı tanıma yapması için değiştirilebilir. Öğrenciler tarafından yapılan çalışmalar sunucu üzerinde tutulmakta ve gerektiğinde yapılan çalışmalar kontrol edilebilmektedir. Bunun için öğretmene ya da kontrol edecek kişiye özel ayrı bir kontrol paneli hazırlanabilir. Öğrencilerin çalışma istatistiklerinin bulunduğu, uygulamayı kullanan öğrencilerin görülebildiği, sisteme giriş-çıkış zamanlarının takibi gibi bazı özellikler eklenebilir.

KAYNAKLAR

1. Yüksel A., “Okuma Güçlüğü Çeken Bir Öğrencinin Okuma Becerisinin Geliştirilmesine Yönelik Bir Çalışma”. Kuramsal Eğitimbilim. 3 (1), 124-134, 2010
2. Chen C., Hsu S., “Personalized Intelligent Mobile Learning System for Supporting Effective English Learning”, Educational Technology & Society. 11 (3), pp. 153-180. ISSN 1436-4522
3. Cho S., Kim J., Lee S., “Mobile Computer-Assisted Language Learning Courseware for Korean Language Learners”, ISSN 0302-9743, pp. 173–178. (2004)
4. Zare S., “Intelligent Mobile Learning Interaction System (IMLIS) A Personalized Learning System for People with Mental Disabilities”, 2010
5. Jones L., Pellegrini A.D., “The effects of social relationships, writing media, and microgenetic development on first grade students written narratives”, American Educational Research Journal, 33, 691–718, 1996
6. Nash L., Schwartz L., “Making computers work in the writing class”, Educational Technology, 25, pp. 19–26, 1985
7. Read C.J., “A study of the usability of handwriting recognition for text entry by children”, Interacting with Computers, 19, pp. 57–69, 2007
8. Yılmaz M., “Kelime Tekrar Tekniğinin Akıcı Okuma Becerilerini Geliştirmeye Etkisi”, Türk Eğitim Bilimleri Dergisi, 6(2), pp.323-350, 2008
9. Özsoy Y., “Okuma yetersizliği”. Anadolu Üniversitesi Eğitim Fakültesi Dergisi. (1), 17-20, 1984
10. Akyol H., “Yeni Programa Uygun Türkçe Öğretim Yöntemleri”, Ankara: Kök Yayıncılık, 2006
11. Hudson R.F., Lane H.B., Pullen P.C., “Reading fluency assessment and instruction: What, why, and how? ”, The Reading Teacher, 58, 2005
12. Joseph L.M., “Helping Children Link Sound Top Rint: Phonics Procedues For Small- Group Or Whole- Class Settings”, Intervention In School And Clinix, 31, 2002

13. http://www.brighthand.com/article/CalliGrapher_7_Now_Available/
(17.07.2013)
14. <http://risujin.org/cellwriter/> (17.07.2013)
15. Sektörel Araştırma ve Strateji Geliştirme Daire Başkanlığı Bilgi Teknolojileri ve İletişimi Kurumu, 2013 Yılı 1. Çeyrek Ocak-Şubat-Mart Üç Aylık Pazar Verileri Raporu, Ankara, Mayıs 2013
16. Petrova K., “Mobile Learning as A Mobile Business Application”, Int. J. Innovation and Learning, Vol. 4, No. 1, 2007
17. Garrick J., Usher R., “Flexible learning, contemporary work and enterprising selves”, Electronic Journal of Sociology, Vol. 5, No. 1, 2000
18. Brophy P., Craven J., Fisher S., “The Development of UK Academic Library Services in the Context of Lifelong Learning”, LITC, South Bank University, London, UK, 1998
19. Raskin J., “The Humane Interface: New Directions for Designing Interactive Systems”, Addison-Wesley Professional. 2nd Edition. ISBN-13: 978-0201379372, 2000
20. Ally M., ”Mobile Learning. Transforming the Delivery of Education and Training”, AU Press, Athabasca University, ISBN 978-1-897425-43-5, 2009
21. Woodill G., Cunningham-Reid A., Nantel R., “Mobile Learning Comes of Age: How and Why Organizations are Moving to Learning on Mobile Devices”, Brandon Hall Research, USA, September, 2008
22. Keegan D., “Mobile Learning: The Next Generation of Learning”, Distance Educational International, 2005
23. Petrova, K. “Teaching differently: a hybrid delivery model”, Proceedings of the Global Business and Technology (GBATA 2001) International Conference, Istanbul, Turkey, pp.928–939, 2001
24. Soloway E., Guzdial M. J., Hay K. E., “E-Learner Centered Design: The Challenge for HCI in 21st Century”, Interactions, 1(2), 1994
25. Landers P., “The Advantages And Disadvantages of Using WAP In Developing mlearning Course”, 2002,
http://learning.ericsson.net/mlearning2/project_one/wap_article.html (15.08.2013)

26. Ballard B., "Designing the Mobile User Experience", Wiley. ISBN-13: 978-0470033616. 2007
27. <http://www.adlnet.gov> (20.08.2013)
28. <http://www.imsglobal.org> (20.08.2013)
29. <http://www.ieeeltsc.org> (30.08.2013)
30. Lapp D., Flood J., Farnan N., "Content Area Reading and Learning: Instructional Strategies", Routledge 3rd Edition. ISBN-13: 978-0805852059, 2007
31. Harman K., Koohang A., "Learning objects: standards, metadata, repositories & LCMS", Informing Science Press, ISBN: 83-922337-5-1, 2007
32. Bearne E., "Making Progress in English", Routledge, ISBN-13: 978-0415159968, London, 1998
33. Eden M., "Handwritng and Pattern Recognition", IRE Trans. Information Theory, vol. 8, 1962.
34. Nouboud F., Plamondon R., "On-Line Recognition of Handprinted Characters: Survey and Beta Tesis", Pattern Recognition, vol. 25, no.9, 1990
35. Plamondon R., Srihari S. N., "On-Line and Off-Lline Handwriting Recognition: A Comprehensive Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, No.1, January 2000
36. <http://en.wikipedia.org/wiki/Dynabook> (30.09.2013)
37. Srihari S.N., "High Performance Reading Machines", Proc. IEEE, vol. 80, pp. 291-302, 1993
38. https://www.iit.demokritos.gr/IIT_SS/Presentations/Off-Line%20Handwritten%20OCR.ppt (01.11.2013)
39. Vinciarelli A., Luettin J., "A new normalization technique for cursive handwritten words", Pattern Recognition Letters, vol. 22, pp. 1043-1050, 2001
40. <http://www.isikdogan.com/2010/hough-donusumu-ile-dairesel-sekil-tespiti/> (28/11/2013)
41. Jensen, Jeppe. "Hough Transform for Straight Lines", 2011. http://www.cvmt.dk/education/teaching/e07/MED3/IP/hough_lines.pdf (03.11.2013)

42. Mathur S., Aggarwal V., Joshi H., Ahlawat A., “Offline Handwriting Recognition Using Genetic Algorithm”, Information Science and Computing, pp.21-27
43. Nishida H., “An Approach to Integration of Off-Line and On-Line Recognition of Handwriting”, Pattern Recognition Letters, vol. 16, pp.1, 213-1, 219, 1995
44. <http://techcrunch.com/2013/07/01/android-led-by-samsung-continues-to-storm-the-smartphone-market-pushing-a-global-70-market-share/?ncid=tcdaily> (03.01.2014)
45. <http://www.highlightpress.com/android-tops-80-global-smartphone-market-share-windows-phone-up-156-year-on-year/6708/tharper> (03.01.2014)
46. <http://developer.android.com/about/dashboards/index.html> (04.01.2014)
47. http://www.w3schools.com/svg/svg_intro.asp (04.12.2013)
48. http://tr.wikipedia.org/wiki/%C3%96l%C3%A7eklenebilir_Vekt%C3%B6r_Grafikleri (04.01.2014)
49. Smith R.W., “The Extraction and Recognition of Text from Multimedia Document Images”, PhD Thesis, University of Bristol, 1987.
50. Schneider P.J., “An Algorithm for Automatically Fitting Digitized Curves”, Graphics Gems, pp. 612-626, 1990
51. Smith, R., “An Overview of the Tesseract OCR Engine”, Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2, 629–633, 2007
52. <http://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract3> (05.01.2014)

ÖZGEÇMİŞ

Banu Yılmaz, 1989 yılı Rize doğumludur. Özel Marmara Lisesi'nden mezun olduktan sonra 2007'de Maltepe Üniversitesi Bilgisayar Mühendisliği bölümünü kazandı ve fakülte derecesiyle 2011 yılında mezun oldu. Ağustos 2011 - Şubat 2014 tarihleri arasında Maltepe Üniversitesi Bilgi İşlem Daire Başkanlığı'nda Yazılım Uzmanı olarak görev yaptı. Şubat 2014'ten itibaren Intertech Bilgi Teknolojileri'nde Yazılım Mühendisi olarak çalışmaktadır. 2011 yılında eğitime başladığı Maltepe Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans programını 2014 yılında tamamladı.