

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPAY SİNİR AĞLARIYLA OPTİK KARAKTER TANIMA
KULLANILARAK GÜNÜMÜZ TÜRKÇESİNİN
OSMANLICAYA ÇEVİRİLMESİ**

YÜKSEK LİSANS TEZİ

İshak DÖLEK

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Tez Danışmanı : Yrd. Doç. Dr. Serap KAZAN

Haziran 2016

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

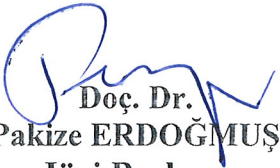
YAPAY SİNİR AĞLARI İLE OPTİK KARAKTER
TANIMA KULLANILARAK GÜNÜMÜZ
TÜRKÇESİNİN OSMANLICAYA ÇEVİRİLMESİ


YÜKSEK LİSANS TEZİ

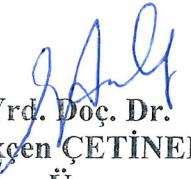
İshak DÖLEK

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM
MÜHENDİSLİĞİ

Bu tez 10.06.2016 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.


Doç. Dr.
Pakize ERDOĞMUŞ
Jüri Başkanı


Yrd. Doç. Dr.
Serap KAZAN
Üye


Yrd. Doç. Dr.
Gökçen ÇETİNEL
Üye

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

İshak DÖLEK

18.5.2016

TEŐEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Yrd. Doç. Dr. Serap Kazan'a teşekkürlerimi sunarım.

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	iii
ŞEKİLLER LİSTESİ	iv
TABLolar LİSTESİ	v
ÖZET	vi
SUMMARY	vii
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
KAYNAK ARAŞTIRMASI	2
2.1. Optik Karakter Tanıma	2
2.2. Yapay Sinir Ağları	5
2.2.1. YSA'ların kullanım alanları	8
2.2.2. Yapay sinir ağlarının sınıflandırılması	10
2.2.3. Öğrenme algoritmalarına göre yapay sinir ağları	11
2.2.4. Öğrenme zamanına göre yapay sinir ağları	12
2.3. Günümüz Osmanlı Türkçesi	13
BÖLÜM 3.	
MATERYAL VE YÖNTEM	16
3.1. Optik Karakter Tanıma	16

3.1.1. OCR ön işleme	16
3.1.2. Segmentasyon ve ölçeklenme	17
3.1.3. Özellik çıkartma	18
3.2. Yapay Sinir Hücresinin Yapısı	19
3.2.2. Matematiksel yapay sinir hücresi modeli	21
3.2.3. Geri yayılım algoritması	23
3.2.4. YSA'da geri yayılım çalışma prosedürü	25
3.2.5. Yapay sinir ağların eğitilmesi	27
3.2.6. Kullanılan YSA'nın test edilmesi	33
3.3. Osmanlı Türkçesi	35
3.3.1. Osmanlı Türkçesi genel özellikleri	36
3.3.2. Kelime çeviri algoritması	40
BÖLÜM 4.	
TARTIŞMA VE SONUÇ	47
KAYNAKLAR	48
ÖZGEÇMİŞ	51

SİMGELER VE KISALTMALAR LİSTESİ

ÇKA : Çok katmanlı algılayıcı

L1 : Giriş katmanı

L2 : Gizli katman

L3 : Çıkış katmanı

RGB : Red-Green-Blue

OCR : Optical Character Recognition

YSA : Yapay sinir ağları

YSH : Yapay sinir hücresi

ŞEKİLLER LİSTESİ

Şekil 2.1. OCR genel yapısı	4
Şekil 2.2. YSA ile geleneksel algoritmaların karşılaştırılması	8
Şekil 2.3. Osmanlı Türkçesinin alfabesi.....	15
Şekil 3.1. Her resmin ön işlemeden geçirilmesi.....	17
Şekil 3.2. Segmentasyon ve ölçekleme	18
Şekil 3.3. Binarize edilmiş imge	19
Şekil 3.4. YSA'nın giriş katmanı	20
Şekil 3.5. Sigmoid aktivasyon fonksiyonu	21
Şekil 3.6. Yapay sinir ağı matematiksel gösterimi.....	22
Şekil 3.7. Eğitim seti resim örnekleri	29
Şekil 3.8. Eğitim seti örneklerin ikili formattaki değerleri	29
Şekil 3.9. Eğitim ve test setlerin hazırlanması	30
Şekil 3.10. Eğitim seti örnekler ve sınıflandırma şekli	31
Şekil 3.11. Tercih edilen ağ yapısı	33
Şekil 3.12. YSA'nın test edilmesi	34
Şekil 3.13. Osmanlı Türkçesi çeviri algoritması	44
Şekil 3.14. Çeviri algoritmasının çıktısı	45
Şekil 3.15. Elif okutucusu için sistem çıktısı	46
Şekil 3.16. He okutucusu için sistem çıktısı	47

TABLULAR LİSTESİ

Tablo 3.1. YSA'nın farklı parametreleri için başarıml oranı	32
Tablo 3.2. Osmanlı türkçesi ile günümüz türkçesi karşılaştırması	35
Tablo 3.3. Arapça ve Farsça kelimelerin grammer yapıları ve yazılışları	43



ÖZET

Anahtar Kelime: Karakter Tanıma, Günümüz Türkçesi, Osmanlıca, Yapay Sinir Ağı

Bu çalışmada, yapay sinir ağları (YSA) ile Optik Karakter Tanıma (OCR) kullanılarak, günümüz Türkçe'sini Osmanlı Türkçe'sine çevirme işlemi gerçekleştirilmiştir. Bu çalışma iki aşamadan oluşmaktadır. Birinci aşama herhangi bir resim üzerindeki karakterlerin akıllı bir sistem tarafından tanınıp ayrıştırılması işlemidir. İkinci aşama ise ayrıştırılmış karakterlerin Osmanlı Türkçe'sine çevrilmesi işlemidir. Birinci aşamada OCR ve Yapay Zekânın alt dallarından olan YSA'ndan faydalanılmıştır. Yapay Zekâ, idealize edilmiş bir yaklaşıma göre insan zekâsına özgü olan, algılama, öğrenme, çoğul kavramları bağlama, düşünme, fikir yürütme, sorun çözme, iletişim kurma, varsayım yapma ve karar verme gibi yüksek bilişsel fonksiyonları veya otonom davranışları sergilemesi beklenen yapay bir işletim sistemidir. Yapay Zekânın alt dallarından olan YSA ise; insan beyninin bilgi işleme tekniğinden esinlenerek geliştirilmiş bir bilgi işlem teknolojisidir. YSA'nın eğitilmesi için kullanılacak olan resim formatındaki karakterler görüntü işleme tekniklerinden faydalanılarak, ikili resim elde edilmiş ve ölçeklendirilmiştir. Daha sonra bunların öznitelikleri çıkartılarak normalize edilmiştir. Normalize edilen veriler ağa sunularak ağ eğitilmiştir. Eğitilen YSA test edilerek başarıyı hesaplanmıştır.

Eğitilen ağ ayrıştırılmış karakterleri, sırası ile tanır. Tanınan kelimelerin Osmanlı Türkçe'sine çevrilme işlemi esnasında Osmanlıca gramer yapısına göre değiştirilmesi gerekir. Osmanlı Türkçesi gramer yapısına göre "a" seslisi için "ا" (elif), e seslisi için "ه" (he), o, ö, u, ü sesleri için "و" (vav) ve ı, i sesleri için ise "ی" (ye) kullanılır. Osmanlı Türkçe'sinde Arapça ve Farsça kelimeler de çok kullanılır. Bu kelimeler ise Arapça ve Farsça gramer yapısına göre çevrilir.

TRANSLATING CONTEMPORARY TURKISH TO OTTOMAN TURKISH BY USING ARTIFICIAL NEURAL NETWORK BASED OPTICAL CHARACTER RECOGNITION

SUMMARY

Keywords: Artificial Neural Networks, Character Recognition, Current Turkish, Ottoman Turkish

In this study translation of current Turkish characteristics into Ottoman Turkish is performed by using Artificial Neural and Optical Character Recognition. This study consists of two levels. The first level is after recognized any character on the picture which will be known and will be separated. Another step is separated characters will be converted to Ottoman language. In that first step were used neural artificial networks that belongs to OCR system, artificial intelligence illiterately; the theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages. Subset of the artificial intelligence which is some of them is a neural artificial networks were gotten as idea of the human brain system, it include kind of microprocessor technology. In this technology were preferred expansionary networks as web structure and inside this technology there are some different Technics effects perfectly such as image process techniques. In this study neural artificial networks were tested perfectly and applied on the study also all characters were separated intensively.

Trained network recognizes the separated characters respectively. Dedicated words is translated by structure of the ottoman language grammar. Such as for sound of a "ا" (elif), for sound of e "ه" (he) for sounds of o,ö,ü,u "و" (vav) for sound of ı, i "ی" (ye) and another mentioned words is translated Arabic and Persian grammar structures.

BÖLÜM 1. GİRİŞ

OCR (Optical Character Recognition) bir araç veya makine yardımıyla ya da el yazısı ile yazılmış dokümanların bilgisayar diline çevrilmesi ve sayısallaştırılmasıdır [1].

Osmanlıca; Türklerin yüzyıllar boyunca geliştirdikleri özgün bir dildir. Millî kültürümüzün temelini oluşturan eserlerimizin hemen hemen tamamı, Osmanlıcayla yazılmıştır. Osmanlı arşivleri, yüzyıllarca hüküm sürmüş bir imparatorluğun milletimize bırakmış olduğu kültür mirası içerisinde hiç şüphesiz ki oldukça önemli bir yer tutmaktadır. Her ne kadar bu arşivler, geçtiğimiz on yıllar boyunca sayısız araştırmacının en temel kaynağı olmuşsa da, arşivlerin içerdiği engin bilgi ve belgelerin bütün potansiyeliyle kullanılabilmesi ve araştırmaların daha hızlı, kapsamlı ve verimli yürütülebilmesi için günümüz teknolojisinin yardımına ihtiyaç vardır [2].

OCR ile ilgili birçok çalışma yapılmıştır. Fakat OCR devamı olabilecek günümüz Türkçe'sini Osmanlıcaya çevirme işlemi gerçekleştirilmemiştir. Karakterlerin Osmanlıcaya çevrilmesi başarılı bir şekilde gerçekleştirilip, eğitim alanına "Osmanlıca Matbu Hat" ile birçok kaynak sağlanması amaçlanmıştır.

Bu çalışmada yapay sinir ağına dayalı bir tanıma işlemi gerçekleştirilmiştir. Karakter tanıma işleminin net ve düzgün olması için alt sistemlerin çalışma sırası (resmin alınması, ön işlemler, resmin satırlara ve sütunlarına ayrıştırılması, özellik çıkarımı ve karakter tanıma) oldukça önemlidir. Burada bahsedilen ön işlemler; metnin ya da resmin gürültüsünün temizlenmesi, boyut farklılıklarının giderilmesi, karakterlerin inceltirilerek iskeletinin ortaya çıkarılması ve karakterlerin ayrıştırılması gibi işlemlerdir. Karakter tanıma işlemi YSA kullanarak tanımlama yapıldıktan sonra karakterler Osmanlıcaya başarılı bir şekilde çevrilmiştir.

BÖLÜM 2. KAYNAK ARAŞTIRMASI

Bu bölümde YSA ile OCR gerçekleştirilip Osmanlıcaya çevrilmesinde kullanılan sistemler tanıtılmıştır.

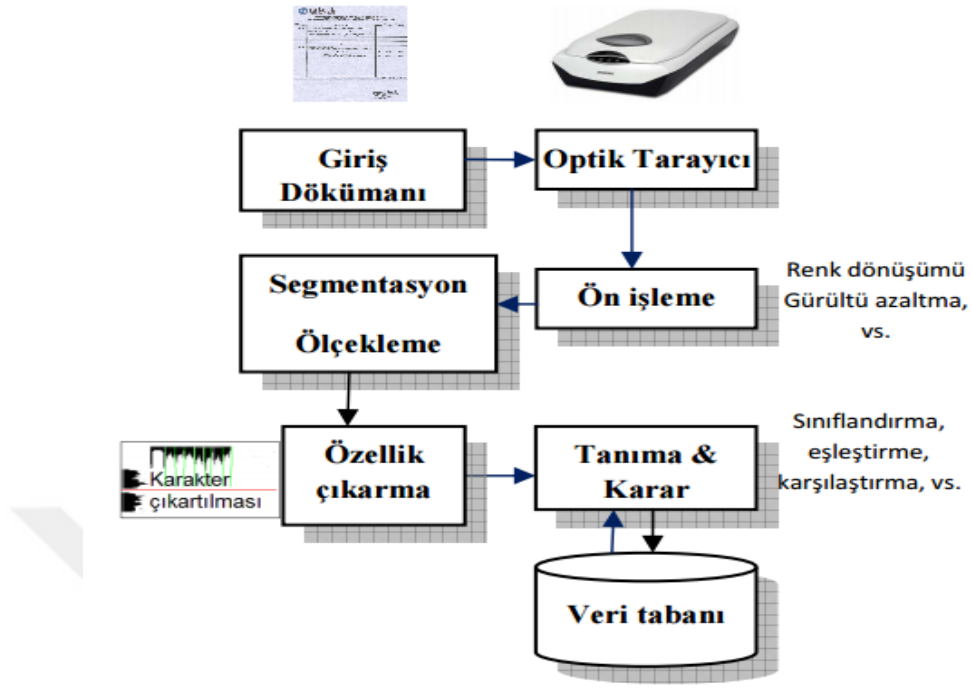
2.1. Optik Karakter Tanıma

OCR; elle ya da elektronik olarak yazılmış görüntüler üzerindeki karakterlerin ya da metin bilgilerinin okunarak ASCII koduna dönüştürülmesi işlemidir. İlk bilgisayar destekli, 23 harfi tanıyabilen OCR sistemi GİSMO, 1950’de David Shepard tarafından Amerika’da geliştirilmiştir. 1950’li yılların ortasından başlayıp günümüze kadar donanım cihazı ve yazılım paketi olarak ticari OCR sistemleri geliştirilmeye devam etmektedir. OCR üzerine yapılan çalışmalar oldukça fazladır [3,4]. OCR ile ilgili çalışmalar günümüzde de devam etmektedir. Karmaşık zeminlerde ve çevre koşullarında, video çerçeveleri üzerindeki yazıların tanınması vb. konularda çalışmalar devam etmektedir. Basılı ve el yazılarının bilgisayar ortamına dönüştürülmesini hızlandırması OCR’ı bilgisayar bilimcileri, mühendisler ve farklı disiplinlerdeki pek çok kişinin ilgisini çeken bir konu haline getirmiştir. Bu tarz yetenekleri sayesinde OCR, insan-bilgisayar etkileşimi, örüntü tanıma, görüntü işleme, metin-konuşma çevrimi, dil işleme, metin madenciliği, yapay zekâ gibi pek çok bilgisayar uygulama alanına yayılmıştır. Ayrıca, OCR ile donatılmış bilgisayar sistemlerinin hızlı veri girişi ve metin işlemeyi izin verme, bazı olası insan hatalarını azaltma ve hızlı bilgi bulmayı mümkün kılma gibi avantajları da vardır. 1999 yılında, Gorski ve arkadaşları, Fransa, İngiltere veya Amerika’da düzenlenen el yazısı veya basılı çekleri işlemek için bir çek tanıma sistemi [5]. Ni ise 2007 yılında, Amerikan posta servislerindeki adresleri sıralamak için posta kodlarını okuyan çok katmanlı geri-beslemeli YSA kullanan bir OCR sistemi geliştirdi [6]. Mani ve Srinivasan 1997 yılında YSA kullanan bir OCR sistemi önerdi [7].

Ahmet 1994 yılında, daha sonra Alshebeili ve arkadaşları ise 1997 yılında Arap alfabesini tanıyan bir OCR sistemi geliştirdi [8]. Leung ve daha sonra Zhang ve arkadaşları 2002 yılında, el yazısı Çin karakterlerini tanıyan çalışmalar geliştirdi [9,10]. Inoue ve arkadaşları 1998 yılında matematiksel formüller içeren Japonca dokümanlara ait taranmış sayfa görüntüsündeki Japonca metinleri ve matematiksel formülleri ayrı ayrı tanıyan bir OCR sistemi [11]. Singh ve arkadaşları YSA yardımı ile Devanagari (Hint) alfabesini tanıyan bir optik karakter tanıma sistemi geliştirdi [12].

Son yıllarda, mobil telefon ve diğer cihazların kameralarından alınan görüntülerdeki kelime ve metinleri okuyan ve farklı dillerde çevirilerini yapan çalışmalar bulunmaktadır. Koga ve arkadaşları mobil bir cihaz kullanılarak resmi çekilmiş Japonca karakterleri tanıyan bir OCR sistemi geliştirdi [13]. Park ve Kwon, masaüstü OCR sistemlerinin mobil platformlar üzerinde çalışabilecek gömülü versiyonunu geliştirdi [14]. Rodriguez ve arkadaşları afiş ve tabelalardaki cep telefonu kamerası kullanılarak çekilmiş JPEG formatındaki resimlerindeki metinleri İngilizceden İspanyolcaya çeviren bir mobil uygulama geliştirdi [15].

OCR işlemi, basılı belgelerin okunarak bilgisayar metinlerine dönüştürülmesi işlemidir. Şekil 2.1.'de tipik bir OCR sisteminin genel yapısı gösterilmektedir. Optik bir cihaz aracılığı ile taranan basılı veya el yazısı dokümana ait karakterler kümesi sayısal görüntüye dönüştürülür. Karakter şekillerinin karmaşıklığı, gürültü, kelime haznesinin kapsamı, sayısallaştırılmış karakterin çözünürlüğüne etki eden matris boyutu, karakter tanımanın hız ve doğruluğunu etkileyen faktörler arasındadır. Basılı veya el yazısı verilerin bulunduğu bölgeyi belirleyerek başlayan tipik bir OCR sistemindeki sonraki işlem adımları sırasıyla ön işleme, segmentasyon ve ölçeklendirme özellik çıkartma ve karakter tanımadır.



Şekil 2.1. OCR genel yapısı

Fotoğraf ve taranmış dokümanlar genellikle RGB renkli resimlerdir. RGB değerlerinin kullanılması, görüntü analizi açısından karmaşıklığın artmasına neden olmaktadır. Bu nedenle renkli resimden gri seviyeye transfer karmaşıklık sorunlarının çözümü ve bir standartlaşma olarak karşımıza çıkmaktadır. Resim gri renk formatına dönüştükten sonra yazının zeminden ayrılması için eşikleme yapılması gerekir. Böylece yazının arka planı çıkarılabilir. Elde edilen ikili görüntüde bir takım gürültüler olması tanıma kalitesini düşürmektedir. Bu olumsuzluğu gidermek için filtreleme ve gürültü azaltıcı algoritmalar uygulanır. Resim üzerindeki satırların bulunması için yatay izdüşümü gerçekleştirilir böylece resim satırları içeren küçük resimlere dönüşür. Daha sonra dikey izdüşümü yapılarak karakterler bölünür. Elde edilen karakterler farklı boyutlarda olabileceğinden dolayı sınıflandırma algoritmasında kullanımı problem teşkil ettiğinden ölçeklendirme yapılır. OCR uygulamalarında, özellik çıkartma önemlidir ve her birinin kendine özgü güçlü ve zayıf yanları bulunan farklı teknikler mevcuttur. Karakterleri doğru bir şekilde hangi sınıfa ait olduğunu belirleyebilen özellikleri çıkartmak en uygun olan yöntemdir. Karakterler doğru parçalarından oluştuğuna göre, karakterleri tanımak için pek çok farklı özellik çıkartılabilir. Son

adımda karakter görüntüsüne ait özellik vektörleri YSA sınıflandırma algoritmaları ile sınıflandırılır.

2.2. Yapay Sinir Ağları

YSA, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri, herhangi bir yardım almadan otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir.

YSA; insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır. Bu nedenledir ki, bu konu üzerindeki çalışmalar ilk olarak beyni oluşturan biyolojik üniteler olan nöronların modellenmesi ve bilgisayar sistemlerinde uygulanması ile başlamış, daha sonraları bilgisayar sistemlerinin gelişimine de paralel olarak birçok alanda kullanılır hale gelmiştir.

İnsan beyninin çalışma prensibini taklit ederek çalışan bu sistemler, her ne kadar bilgisayar teknolojisi hızlı bir gelişim göstermiş, işlem hızları nanosaniyeler mertebesine inmiş olsa da, bırakalım insan beynini, ilkel bir canlı beyninin fonksiyonları dahi temel alındığında, böyle bir organizmanın yanında çok ilkel kalmaktadır. Nanosaniyeler bazındaki işlem hızları ile YSA'lar, milisaniyeler mertebesindeki işlen hızları ile işlem yapan insan beyninin işlevselliğinin henüz çok uzağındadır.

Burada kısa bir hatırlatma yapmak gerekirse; insan beyninde yaklaşık 10^{11} sinir hücresinin varlığından bahsedilmekle birlikte, bu sayının bilgisayar ortamında modellenmesi şu an için mümkün görünmemektedir. Fakat karar hızı açısından insan beyni ile henüz yarışmasalar bile, YSA'lar yapısallıkları ve hassas eşleştirmelerin başarı ile gerçekleştirebilmeleri ile gün geçtikçe daha fazla uygulama alanı bulmaktadır.

YSA'ların Avantajları

YSA'lar, uygulanan ağ modeline göre değişik karakteristik özellikler göstermelerine karşın temel birkaç ortak özelliğe sahiptirler.

1. YSA'lar birçok hücreden meydana gelir ve bu hücreler eş zamanlı olarak çalışarak karmaşık işlevleri yerine getirir. Diğer bir deyişle karmaşık işlevler birçok hücrenin eş zamanlı çalışması ile meydana getirilir. Süreç içerisinde bu hücrelerden her hangi biri işlevini yitirse dahi sistem güvenli bir şekilde çalışmasına devam edebilir.
2. Eğitim esnasında kullanılan nümerik bilgilerden, problemin genel özellikleri elde etmesi ve böylelikle eğitim sırasında kullanılmayan girdiler için de, anlamlı yanıtlar üretebilmesidir.
3. Yapı üzerinde dağılmış belli tipteki doğrusal olmayan alt birimler, doğrusal olmayan problemlerin de çözümünü mümkün kılmaktadır.
4. YSA'lar makine öğrenmesi gerçekleştirebilirler. YSA'nın temel işlevi zaten bilgisayarın öğrenmesini sağlayarak ve olayları öğrenerek benzer olaylar karşısında mantıklı kararlar verebilirler.
5. Bilgi işleme yöntemleri geleneksel programlamadan farklıdır. Bu nedenle geleneksel programlamanın getirdiği birçok olumsuzluk ortadan kaldırılabilir.
6. Bilgiler ağın tamamında saklanır. Geleneksel programlamada olduğu gibi bilgiler veri tabanları ya da dosyalarda belli bir düzende tutulmaz, ağın tamamına yayılarak değerler ile ölçülen ağ bağlantılarında saklanmaktadır. Hücrelerden bazılarının işlevini yitirmesi, anlamlı bilginin kaybolmasına neden olmaz.
7. Dağıtık belleğe sahiptirler. YSA'larda bilgi ağa dağılmış bir şekilde tutulur. Hücrelerin bağlantı ve ağırlık dereceleri, ağın bilgisini gösterir. Bu nedenle tek bir bağlantının kendi başına anlamı yoktur.
8. Örnekleri kullanarak öğrenirler. YSA'nın öğrenebilmesi için örneklerin belirlenmesi, bu örneklerin ağa gösterilerek istenen çıktılara göre ağın eğitilmesi gerekmektedir. Ağın başarısı, seçilen örnekler ile doğru orantılıdır, ağa olay bütün yönleri ile gösterilemezse ağ yanlış çıktılar üretebilir.

9. Daha önce görülmemiş örnekler hakkında bilgi üretebilirler. YSA'lar eğitimleri sırasında kendilerine verilen örneklerden genellemeler çıkarırlar ve bu genellemeler ile yeni örnekler hakkında bilgi üretebilirler.

10. Algılamaya yönelik olaylarda kullanılabilirler. YSA'ların en başarılı oldukları alanlar, algılamaya yönelik uygulama alanlarıdır. Bu alanlarda başarıları kanıtlanmıştır.

11. Örüntü (pattern) ilişkilendirme ve sınıflandırma yapabilirler. YSA'lar kendilerine örnekler halinde verilen örüntüleri kendisi veya diğerleri ile ilişkilendirebilir. Ayrıca kendisine verilen örneklerin kümelenmesi ile bir sonraki verinin hangi kümeye dâhil olacağına karar verilmesi konusunda kullanılabilirler.

12. Örüntü tamamlama yapabilirler. Ağa eksik bilgileri içeren örüntüler verildiğinde eksik bilgilerin tamamlanması konusunda başarılıdırlar.

13. Kendi kendine öğrenebilme ve organize etme yetenekleri vardır. YSA'lar çevrimiçi olarak öğrenebilirler ve kendi kendilerini eğitebilirler.

14. Eksik bilgi ile çalışabilmektedirler. Geleneksel sistemlerin aksine YSA'lar eğitildikten sonra veriler eksik bilgi içerse dahi, çıktı üretebilirler. Bu durum bir performans kaybı yaratmaz, performans kaybı eksik bilginin önemine bağlıdır. Burada bilgilerin önem dereceleri eğitim sırasında öğrenilir.

15. Hata toleransına sahiptirler. YSA'ların eksik bilgilerle çalışabilmeleri ve bazı hücreleri bozulsa dahi çalışabilmeleri, onları hatalara karşı toleranslı yapar.

YSA'ların Dezavantajları

YSA'ların, pek çok avantajın yanında bazı dezavantajları da vardır. Belli başlı dezavantajları;

1. Donanım bağımlıdır. YSA'ların en önemli sorunu donanım bağımlı olmalarıdır. YSA'ların en önemli özellikleri ve var oluş nedenlerinden birisi olan paralel işlem yapabilme yeteneği, paralel çalışan işlemciler ile performans gösterir.

2. Uygun ağ yapısının belirlenmesinde belli bir kural yoktur. YSA'larda probleme uygun ağ yapısının belirlenmesi için geliştirilmiş bir kural yoktur. Uygun ağ yapısı deneyim ve deneme yanılma yolu ile belirlenmektedir.

3. Ağın parametre değerlerinin belirlenmesinde de belli bir kural yoktur. YSA'larda öğrenme katsayısı, hücre sayısı, katman sayısı gibi parametrelerin belirlenmesinde belirli bir kural yoktur. Bu değerlerin belirlenmesi için belirli bir standart olmamakla birlikte her problem için farklı bir yaklaşım söz konusu olabilmektedir.
4. Öğrenilecek problemin ağa gösterimi önemli bir problemdir. YSA'lar nümerik bilgiler ile çalışabilmektedirler. Problemler YSA'lara tanıtılmadan önce nümerik değerlere çevrilmek zorundadırlar. Burada belirlenecek gösterim mekanizması ağın performansını doğrudan etkileyecektir. Bu da kullanıcının yeteneğine bağlıdır.
5. Ağın eğitiminin ne zaman bitirilmesi gerektiğine ilişkin belli bir yöntem yoktur. Ağın örnekler üzerindeki hatasının belirli bir değerin altına indirilmesi eğitimin tamamlandığı anlamına gelmektedir. Burada uygun değer neticeler veren bir mekanizma henüz yoktur ve YSA ile ilgili araştırmaların önemli bir kolunu oluşturmaktadır.
6. Ağın davranışları açıklanamamaktadır. Bu sorun YSA'ların en önemli sorunudur. YSA bir probleme çözüm ürettiği zaman, bunun neden ve nasıl olduğuna ilişkin bir ipucu vermez. Bu durum ağa olan güveni azaltıcı bir unsurdur.

YSA'nın geleneksel algoritmalarına göre avantaj ve dezavantajları bulunmaktadır. Şekil 2.2.'de karşılaştırılmıştır.

Geleneksel Algoritmalar	Yapay Sinir Ağları
Çıktılar, koyulan kurallara girişlerin uygulanması ile elde edilir.	Öğrenme esnasında giriş çıkış bilgileri verilerek, kurallar koyulur.
Hesaplama; merkezi, eş zamanlı ve ardışıldır.	Hesaplama; toplu, eş zamansız ve öğrenmeden sonra paraleldir.
Bellek paketlenmiş ve hazır bilgi depolanmıştır.	Bellek ayrılmış ve ağa yayılmıştır.
Hata toleransı yoktur.	Hata toleransı vardır.
Nispeten hızlıdır.	Yavaş ve donanıma bağımlıdır.
Bilgiler ve algoritmalar kesindir.	Deneyimden yararlanır.

Şekil 2.2. YSA ile geleneksel algoritmaların karşılaştırılması

2.2.1. YSA'ların kullanım alanları

YSA başlıca; Sınıflandırma, Modelleme ve Tahmin uygulamaları olmak üzere, pek çok alanda kullanılmaktadır. Başarılı uygulamalar incelendiğinde, YSA'ların çok boyutlu, gürültülü, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek sensor verilerinin olması ve problemi çözmek için matematiksel modelin ve algoritmaların bulunmadığı, sadece örneklerin var olduğu durumlarda yaygın olarak kullanıldıkları görülmektedir. Bu amaçla geliştirilmiş ağlar genellikle şu fonksiyonları gerçekleştirmektedirler.

Muhtemel fonksiyon kestirimleri, sınıflandırma, ilişkilendirme veya örüntü eşleştirme, zaman serileri analizleri, sinyal filtreleme, veri sıkıştırma, örüntü tanıma, doğrusal olmayan sinyal işleme, doğrusal olmayan sistem modelleme, optimizasyon, kontrol.

YSA'lar pek çok sektörde değişik uygulama alanları bulmuştur. Bunlardan bazıları;

1. Uzay: Uçuş simülasyonları, otomatik pilot uygulamaları, bileşenlerin hata denetimleri vs.
2. Otomotiv: Otomatik yol izleme, rehber, garanti aktivite analizi, yol koşullarına göre sürüş analizi vs.
3. Bankacılık: Kredi uygulamaları geliştirilmesi, müşteri analizi ve kredi müracaat değerlendirilmesi, bütçe yatırım tahminleri vs.
4. Savunma: Silah yönlendirme, hedef seçme, radar, sensor sonar sistemleri, sinyal işleme, görüntü işleme vs.
5. Elektronik: Kod sırası öngörüsü, yonga bozulma analizi, doğrusal olmayan modelleme vs.
6. Eğlence: Animasyonlar, özel efektler, pazarlama öngörüsü vs.
7. Finans: Kıymet biçme, pazar performans analizi, bütçe kestirimi, hedef belirleme vs.
8. Sigortacılık: ürün optimizasyonu, uygulama politikası geliştirme vs.
9. Üretim: üretim işlem kontrolü, ürün dizaynı, makina yıpranmalarının tespiti, dayanıklılık analizi, kalite kontrolü, iş çizelgeleri hazırlanması vs.

10. Sağlık: göğüs kanseri erken teşhis ve tedavisi, Elektrokardiyogram, Manyetik Rezonans, kalite artırımı, ilaç etkileri analizi, kan analizi sınıflandırma, kalp krizi erken teşhis ve tedavisi vs.

11. Petrokimya: arama, verim analizi vs.

12. Robotik: yörengesi kontrol, forklift robotları, görsel sistemler, uzaktan kumandalı sistemler, uygun deęer rota belirleme vs.

13. Dil: sözcük tanıma, yazı ve konuşma çevrimi, dil tercüme vs.

14. Telekomünikasyon: görüntü ve data karşılaştırma, filtreleme, eko ve gürültü önümlendirilmesi, ses ve görüntü işleme, trafik yoğunluęunun kontrolü ve anahtarlama vs.

15. Güvenlik: parmak izi tanıma, kredi kartı hileleri saptama, retina tarama, yüz eşleştirme vs.

Bu örnekler çoęaltılabilir. Görüldüęü gibi YSA'lar günlük hayatımızda farkında olmadığımız pek çok alanda kullanılmaktadır. Gün geçtikçe uygulama alanları genişlemekte ve gelişmektedir.

2.2.2. Yapay sinir ağlarının sınıflandırılması

YSA işleyiş olarak benzer olmalarına rağmen herhangi bir tasarım ve işleyiş standardı bulunmamaktadır. Nöron dizilimlerine, nöronların ağırlıklarının düzenleme için yapılan hesaplamaların türüne ve zamanına göre YSA'yı üç ayrı dalda inceleyebiliriz.

YSA içerdığı nöronların birbirine bağlantı şekline göre ileri ve geri beslemeli olarak ikiye ayrılır.

1. İleri beslemeli ağlar: İleri beslemeli ağlarda nöronlar girişten çıkışa doğru düzenli katmanlar şeklindedir. Bir katmandan sadece kendinden sonraki katmanlara bağ bulunmaktadır. YSA'ya gelen bilgiler giriş katmanına daha sonra sırasıyla ara katmanlardan ve çıkış katmanından işlenerek geçer ve daha sonra dış dünyaya çıkar.

2. Geri beslemeli YSA: Geri beslemeli YSA'da ileri beslemeli olanların aksine bir hücrenin çıktısı sadece kendinden sonra gelen hücrenin katmanına girdi olarak

verilmez. Kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir hücreye de girdi olarak bağlanabilir.

Bu yapısı ile geri beslemeli YSA doğrusal olmayan dinamik bir davranış göstermektedir. Geri besleme özelliğini kazandıran bağlantıların bağlantı şekline göre aynı YSA'yla farklı davranışta ve yapıda geri beslemeli YSA elde edilebilir.

2.2.3. Öğrenme algoritmalarına göre yapay sinir ağları

YSA'nın verilen girdilere göre çıktı üretebilmesinin yolu ağın öğrenbilmesidir. Bu öğrenme işleminin de birden fazla yöntemi vardır. YSA öğrenme algoritmalarına göre danışmanlı, danışmansız ve destekleyici öğrenme olarak üçe ayrılır.

Danışmanlı Öğrenme: Danışmanlı öğrenme sırasında ağa verilen giriş değerleri için çıktı değerleri de verilir. Ağ verilen girdiler için istenen çıkışları oluşturabilmek için kendi ağırlıklarını günceller. Ağın çıktıları ile beklenen çıktılar arasındaki hata hesaplanarak ağın yeni ağırlıkları bu hata payına göre düzenlenir.

Hata payı hesaplanırken ağın bütün çıktıları ile beklenen çıktıları arasındaki fark hesaplanır ve bu farka göre her hücreye düşen hata payı bulunur. Daha sonra her hücrenin kendine doğru gelen ağırlıkları güncellenir.

Danışmansız Öğrenme: Danışmansız öğrenmede ağa öğrenme sırasında sadece örnek girdiler verilmektedir. Herhangi bir beklenen çıktı bilgisi verilmez. Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur. Ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabilecek şekilde düzenleyerek öğrenme işlemini tamamlar.

Destekleyici Öğrenme: Bu öğrenme yaklaşımında ağın her iterasyon sonucunda elde ettiği sonucun iyi veya kötü olup olmadığına dair bir bilgi verilir. Ağ bu bilgilere göre kendini yeniden düzenler. Bu sayede ağ herhangi bir girdi dizisiyle hem öğrenerek hem de sonuç çıkararak işlemeye devam eder.

2.2.4. Öğrenme zamanına göre yapay sinir ağları

YSA öğrenme zamanına göre de statik ve dinamik öğrenme olarak ikiye ayrılır.

Statik Öğrenme: Statik öğrenme kuralıyla çalışan YSA kullanmadan önce eğitilmektedir. Eğitim tamamlandıktan sonra ağı istenilen şekilde kullanılabilir. Ancak bu kullanım sırasında ağın üzerindeki ağırlıklarda herhangi bir değişiklik olmaz.

Dinamik Öğrenme: Dinamik öğrenme kuralı ise YSA'nın çalıştığı süre boyunca öğrenmesini öngörerek tasarlanmıştır. Ağı eğitim aşaması bittikten sonra da daha sonraki kullanımlarında çıkışların onaylanmasına göre ağırlıklarını değiştirerek çalışmaya devam eder.

2.3. Günümüz Osmanlı Türkçesi

Osmanlı Türkçesi ya da Osmanlıca, Osmanlı İmparatorluğu'nun ilk anayasası olan Esasi'de geçtiği hâliyle Türkçe (Osmanlı Türkçesi: لسان تورکی Lisani-ı Türki; تورکی Türki; تورکجه Türkçe; لسان عثمانی, Lisān-ı Osmānī), 13 ile 20. yüzyıllar arasında Anadolu'da ve Osmanlı Devleti'nin yayıldığı bütün ülkelerde kullanılmış olan, Arapça ve Farsçadan etkilenmiş Türk dilidir. Alfabe olarak Arap alfabesinin Farsça ve Türkçe için uyarlanmış bir biçimi kullanılmıştır. (Şekil 2.3.) Halk arasında bazen yanlış kullanım olarak bu dil dönemi için “Eski Türkçe” tabiri de kullanılmaktadır.

XVI. yüzyıldan başlayarak klasik gelişimini sürdüren Türk dili, Doğu Türkçesi veya Çağatay Türkçesi ile Batı Türkçesi veya Osmanlı Türkçesi diye adlandırılan iki büyük yazı dili hâlinde Türk dünyasında varlığını sürdürmektedir [16].

Osmanlı Türkçesinin “lisan-ı Osmani”, “Osmanlı lisanı” diye adlandırılmasına ünlü sözlükçü, yazar Şemseddin Sami şu sözlerle karşı çıkmış ve tıpkı Kaşgarlı Mahmud,

Yusuf Ulug Has Hacib, Ali Şir Nevai gibi dilin adının “Türkçe” olduğunu ifade etmiştir [17].

Şemseddin Sami, Osmanlı lisanı, Çağatay lisanı gibi adlandırmaların yakışsız olduğunu, Çağatay adının da Türk kavimlerinden birinin adı olması dolayısıyla dil adı olmayacağını belirtmiştir. Sami'nin çok doğru ifadesiyle Osmanlı İmparatorluğu'nda kullanılan yazı dili Batı Türkçesi, uzak ülkelerde Türkistan coğrafyasında kullanılan yazı dili ise Doğu Türkçesidir[19].

Klasik devirde "Osmanlı Türkçesi" ayrı bir dil olarak algılanmamış, üç dilden oluşan bir karışım olarak görülmüştü. "Türkçe" ise, evde, sokakta ve köyde konuşulan basit dile verilen addır[20]. Ancak 19. yüzyılda standart bir yazı dili ihtiyacının belirmesiyle birlikte Osmanlı dili tartışmaları yoğunlaştı. Bu dilin belkemiğini oluşturan Türkçenin güçlendirilmesi ve yazı dilinin Türkçe konuşma diline yaklaştırılmasına ilişkin talepler Şinasi, Ali Suavi, Ahmet Vefik Paşa gibi yazarlarca dile getirildi. 19. yüzyıl sonlarında doğan Türkçülük akımı, Osmanlı yazı dilinin esasen Türkçe olduğu ve "Türkçe" diye adlandırılması gerektiğini vurguladı [21].

Cumhuriyet döneminde ise "Osmanlı Türkçesi" deyimini genellikle olumsuz bir anlam kazandı. Dil Devrimi'ni izleyen kültürel ortamda, "Osmanlı Türkçesi", Türkçeden ayrı ve yoz bir dil olarak görüldü. Türk Dil Kurumu'nda 1983'e dek bu görüş egemendi. Buna karşılık Osmanlı kültürüne yakınlık duyan muhafazakâr kesim, Osmanlı yazı dilinin de Türkçenin bir lehçesi olduğunu vurgulamak amacıyla "Osmanlı Türkçesi" deyimini tercih etmiştir [21].

Öte yandan, Osmanlı yazı diline "Osmanlı Türkçesi" adı verildiği zaman, bundan çok farklı bir dil olan Osmanlı dönemi konuşma Türkçesine ne ad verileceği konusu, çözülmemiş bir sorun olarak kalmaktadır [21].

23 Aralık 1876'da ilan edilen Osmanlı İmparatorluğu ilk anayasası olan Kanun-i Esasi'nin 18. maddesinde devletin resmî dilinin "Türkçe" olduğu belirtilmiş ve Türkçe bilmeyenlerin devlet memuriyetine alınmayacağı ifade edilmiştir. Bu devirde Eski

Anadolu Türkçesinin söz varlığı kadar açık bir Türkçe söz varlığı yoktur. Ancak, dilin yapısındaki yabancı sözlerin kullanımı metinden metine, muhitten muhite değişebilmiştir. Örneğin, sanat yapmak kaygısıyla saray muhitinde yazılan ve sadece dar aristokrat kesime hitap etmesi amaçlanan şiir ve nesir örneği eserlerin dili oldukça ağırdır. Halk arasında “Osmanlıca” denince algılanan o Türkçenin dışında farklı dil düşüncesi bu gibi kullanımların sonucu olmuştur. Sanatsal kaygı ve dar kesime hitap durumlarının dışında kalan, bu muhit dışında yazılan dönem eserlerinin dili Türkçenin bir döneminde olabilecek normallikte yabancı öge içermiştir. Türkçe yazı diline Arapça ve Farsça sözcüklerin girişi İslamiyet’in kabulüyle başlar. Türkiye Türkçesinde 13. yüzyıla ait en eski metinlerde toplam kelime hazinesinin üçte biri ile yarısı Arapça ve Farsça alıntılardan oluşur. Ancak 15. yüzyıl ortalarına dek kullanılan yazı Türkçesi, günümüz konuşma dilinden yapıca çok uzak değildir. Dönemin şiir ve düzyazı örneklerinden birçoğu, konuşma Türkçesine yakın yapıdadır.

Korunmuş	Bitiş	Orta	Başlangıç	Adı	ALA-LC Harf Çevirisi	Güncel Türkçesi
ا	ا	—	—	elif	a, â	a, e, â
ء	—	—	—	hemze	ʔ	ʔ, a, e, i, u, ü
ب	ب	ب	ب	be	b, p	b
پ	پ	پ	پ	pe	p	p
ت	ت	ت	ت	te	t	t
س	س	س	س	se	s	s
چ	چ	چ	چ	cim	c, ç	c
چ	چ	چ	چ	çim	ç	ç
ح	ح	ح	ح	ha	h	h
خ	خ	خ	خ	hı	h, x	h
د	د	—	—	dal	d	d
ذ	ذ	—	—	zel	z	z
ر	ر	—	—	re	r	r
ز	ز	—	—	ze	z	z
ج	ج	—	—	je	j	j
س	س	س	س	sin	s	s
ش	ش	ش	ش	şin	ş	ş
ص	ص	ص	ص	sad	s	s
ض	ض	ض	ض	dad	z, đ	d, z
ط	ط	ط	ط	tı	t	t
ظ	ظ	ظ	ظ	zı	z	z
ع	ع	ع	ع	ayın	ʔ	ʔ, h
غ	غ	غ	غ	gayın	g	g, ğ
ف	ف	ف	ف	fe	f	f
ق	ق	ق	ق	kaf	k, q	k
ك	ك	ك	ك	kef	k, g, ñ	k, g, ğ, n
گ	گ	گ	گ	gaf ¹	g	g, ğ
گ	گ	گ	گ	nef, sağır kef	ñ	n
ل	ل	ل	ل	lam	l	l
م	م	م	م	mim	m	m
ن	ن	ن	ن	nun	n	n
و	و	—	—	vav	v, w, o, ô, ö, u, û, ü	v, o, ô, u, û, ü
ه	ه	ه	ه	he	h, e, a	h, e, a
لا	لا	—	—	lamelif	lâ	la
ی	ی	ی	ی	ye	y, i, î, î	y, i, î, î

Şekil 2.3. Osmanlı Türkçesi alfabesi

BÖLÜM 3. MATERYAL VE YÖNTEM

3.1. Optik Karakter Tanıma

OCR yazılımları kâğıt vb. üzerindeki siyah noktalar topluluğunu yani harfleri, rakamları, sembolleri vb. algılayabilecek ve okuyabilecek bir yapıya sahiptirler. Her kelimeyi ayrı tutarak ve sayfayı satırlara bölerek analiz eder. Analiz sırasında harflerin yapısal karakterleri, yükseklikleri ve genişlikleri değerlendirilir. Programın kendi bilgi deposu, her bir karakter için tanımlanmış çeşitli parametrelerle doludur. Algılanan pikseller bu parametrelerden birisine uygun ise, harfin ortaya çıkması mümkün hale gelir. Resim dosyalarının ve tarayıcının kaliteli olması OCR yazılımlarının metin tarama hatalarını çok düşük seviyeye indirir. Son yıllarda piyasaya sunulan OCR yazılımları hemen hemen her türlü karakter yakalayabilecek seviyelere ulaşmıştır. Optik karakter tanıma işleminde yapılan aşamalar aşağı da sırasıyla anlatılmaktadır.

3.1.1. OCR ön işleme

Fotoğraf ve taranmış dokümanlar genellikle RGB renkli resimlerdir. RGB değerlerinin kullanılması, görüntü analizi açısından karmaşıklığın artmasına neden olmaktadır. Bu nedenle renkli resimden gri seviyeye transfer karmaşıklık sorunlarının çözümü ve bir standartlaşma olarak karşımıza çıkmaktadır. Resim gri renk formatına dönüştükten sonra yazının zeminden ayrılması için eşikleme yapılması gerekir. Böylece yazının arka planı çıkarılabilir. Elde edilen ikili görüntüde bir takım gürültüler olması tanıma kalitesini düşürmektedir. Bu olumsuzluğu gidermek için filtreleme ve gürültü azaltıcı algoritmalar uygulanır. Görüntü işleme birçok filtreleme bulunmaktadır. Gaussian filtreleme uygulanmıştır. Gaussian filtrelemesi; Ortalama filtrenin Gaussian dağılımını

kullanarak biraz daha değiştirilmiş hali Gaussian filtresi olarak bilinir. Gaussian filtreleme aynı zamanda bir fourier dönüşümüdür. Gauss filtre ile sonsuz bir transfer fonksiyonuna karşılık mekânsal alanda sonlu bir pencerede (tarama penceresi) filtreleme yapılabilir. Bu da filtrelemenin temel problemini daha kolay çözülebilir hale getirir. Gauss Yumuşatmasının Avantajları: Filtreleme önce yatay ardından çıkan sonuçla düşey ekseninde gerçekleştirilebilir. Bu çalışmada uygulanan ön işlemler Şekil 3.1.' de gösterilmiştir.

```

Mat SplitOCR::pre_process(Mat src){
    Mat gray, thr_img;
    cvtColor(src, gray, CV_BGR2GRAY);
    threshold(gray, thr_img, 10, 255, THRESH_BINARY);
    cv::GaussianBlur(thr_img, thr_img, cv::Size(3,3), 0, 0, 1);
    return thr_img;
}

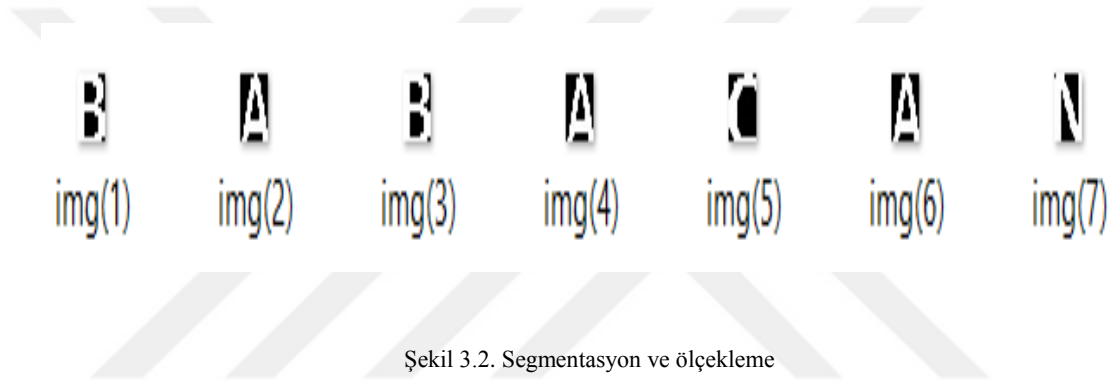
```

Şekil 3.1. Her resmin ön işlemeden geçirilmesi

3.1.2. Segmentasyon ve ölçekleme

Segmentasyon genellikle görüntü analizinin ilk aşamasıdır. Görüntü segmentasyonu, bir görüntüyü her biri içerisinde farklı özelliklerin tutulduğu anlamlı bölgelere ayırmak olarak tarif edilebilir. Örneğin, görüntü içerisindeki benzer parlaklıklar olabilir ve bu parlaklıklar ilgili görüntünün farklı bölgelerindeki nesnelere temsil edebilir. Uygulamaya bağlı olarak değişebilen bu segmentlere (bölütler-elemanlar) başka bir örnek olarak; hava-yer fotoğrafında, yolda hareket eden araçları ve çevreyi yoldan ayırt edebilmek için bir segmentleme yapılabilir. Unutulmamalıdır ki, tüm görüntülere uygulanabilecek genel (universal) bir segmentasyon yöntemi yoktur ve hiçbir segmentasyon yöntemi mükemmel değildir. Başka bir deyişle, görüntü iyileştirme ve onarma problemlerinde olduğu gibi görüntü segmentasyonu için tasarlanan yöntemler ve bu yöntemlerin başarımları, görüntüden görüntüye ve uygulamaya dayalı olarak değişiklik arz eder. Otomatik görüntü segmentasyonu görüntü işleminin en zor işlemlerinden biridir. Gri seviyeli görüntülerde segmentasyonu algoritmaları gri seviye (parlaklık) değerlerinin süreksizlik ve benzerlik özelliğine dayandırılır. Süreksizlik

tabanlı segmentasyon algoritmaları; İzole nokta, ince çizgi veya resim kenarları gibi (gri seviye değerleri birden değişen) süreksizlikleri, düşük ve yüksek filtrelemedeki gibi benzer maskeler kullanarak tespit edebilmeye dayanır. Benzerlik tabanlı segmentasyon algoritmaları ya eşiklenen bölgede büyüme, ya da bölge bölme ve birleştirmeye dayanmaktadır. Bu uygulama da ise yatay ve dikey kestirim yöntemleri kullanarak segmentasyon işlemi gerçekleştirilmiştir. Elde edilen segmentler farklı boyutlarda olabileceğinden dolayı sınıflandırma algoritmasında kullanımı problem teşkil ettiğinden ölçeklendirme yapılır. Bu çalışmadaki ölçeklendirme 16x16 piksel şeklinde yapılmıştır ve Şekil 3.2.'de gösterilmiştir.



Şekil 3.2. Segmentasyon ve ölçekleme

3.1.3. Özellik çıkartma

Herhangi bir problemin makine öğrenmesi yöntemleriyle çözülebilmesi için sistemin uygun şekilde temsil edilmesi gerekir. Çözülmesi istenen problem, makine öğrenmesi yöntemlerine doğrudan verilebilecek niteliklere her zaman sahip olmayabilir. Özellikle zaman serileri ve görüntüler gibi işaretlerden uygun özelliklerin çıkartılması gerekir. Zaman serilerine odaklanan bilim dalı “işaret işleme” ve fotoğraf-video gibi görüntülerle uğraşan bilim dalı ise “görüntü işleme” adıyla anılır. “Örüntü tanıma” adlı alan ise zaman serileri ve görüntüleri de içeren her tür işaretten özellik çıkartmayı amaçlar. Literatürde önerilmiş birçok özellik çıkartma yöntemi vardır. Fakat bir probleme ait veride ne kadar çok nitelik varsa makine öğrenmesi yöntemlerinin maliyeti de o kadar artar. Bu istenmeyen bir durumdur. Bu yüzden eldeki problemi en uygun şekilde temsil etmek üzere minimum sayıda özellikten yararlanmak gerekir. Bu

tez çalışmasında ölçeklenme yapılan imgeler 16X16'lık piksel binarize edilerek (Şekil 3.3.) 256 bitlik bir vektör haline getirilmiştir.

1,1,1,1,1,0,1,1,1,1,1,0,1,1,	0,1,1,1,1,1,1,1,1,1,0,0,0,0,
0,0,0,1,1,0,0,0,1,1,1,0,0,0,1,	1,1,1,1,1,1,1,1,1,1,1,1,0,0,
0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,	1,1,1,1,1,1,1,1,1,1,1,1,1,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,1,1,1,1,1,0,0,
0,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,
0,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,
0,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,
1,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,
1,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,
1,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,
1,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,

Şekil 3.3. Binarize edilmiş imge

3.2.1. Yapay sinir hücresinin yapısı

Yapay sinir hücreleri de biyolojik sinir hücrelerine benzer yapıdadır. Yapay nöronlar da aralarında bağ kurarak yapay sinir ağlarını oluştururlar. Aynı biyolojik nöronlarda olduğu gibi yapay nöronların da giriş sinyallerini aldıkları, bu sinyalleri toplayıp işledikleri ve çıktılarını ilettikleri bölümleri bulunmaktadır.

Bir yapay sinir hücresi beş bölümden oluşmaktadır;

1. Girdiler
2. Ağırlıklar
3. Toplama Fonksiyonu (Birleştirme Fonksiyonu)
4. Aktivasyon fonksiyonu
5. Çıktılar

Girdiler: Girdiler nöronlara gelen verilerdir. Girdiler yapay sinir hücresine bir diğer hücreden gelebileceği gibi direk olarak dış dünyadan da gelebilir. Bu girdilerden gelen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nöron çekirdeğine gönderilir. Bu tez çalışmasında ise giriş katmanına, imgeler 16X16 ölçeklendirildiği için giriş katmanı da 256 giriş seçilmiştir (Şekil 3.4.).

```
cv::Mat layers(3,1,CV_32S);
layers.at<int>(0,0) = ATTRIBUTES;//giriş katmanı #define ATTRIBUTES 256 //Pixel sayısı 16x16
layers.at<int>(1,0)=32;//gizli katman
layers.at<int>(2,0) =CLASSES;//çıkış katmanı
```

Şekil 3.4. YSA'nın giriş katmanı

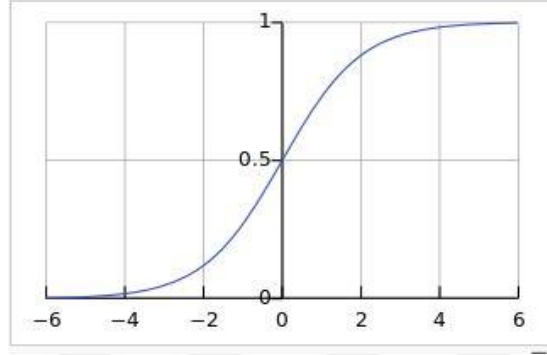
Ağırlıklar: Yapay sinir hücresine gelen bilgiler girdiler üzerinden çekirdeğe ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılarak çekirdeğe iletilir. Bu sayede girdilerin üretilecek çıktı üzerindeki etkisi ayarlanabilmektedir. Bu ağırlıkların değerleri pozitif, negatif veya sıfır olabilir. Ağırlığı sıfır olan girdilerin çıktı üzerinde herhangi bir etkisi olmamaktadır.

Toplama Fonksiyonu (Birleştirme Fonksiyonu): Toplama fonksiyonu bir yapay sinir hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur.

Bazı durumlarda gelen girdilerin değeri dikkate alınırken bazı durumlarda ise gelen girdilerin sayısı önemli olabilmektedir. Bir problem için en uygun toplama fonksiyonu belirlenirken geliştirilmiş bir yöntem yoktur. Genellikle deneme yanılma yoluyla toplama fonksiyonu belirlenmektedir. Bazen her hücrenin toplama fonksiyonunun aynı olması gerekmez. Bu konulara karar vermek tasarımcıya aittir.

Aktivasyon Fonksiyonu: Bu fonksiyon hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon olarak seçilir. YSA'nın bir özelliği olan "doğrusal olmama" aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir. Aktivasyon fonksiyonu seçilirken dikkat edilmesi gereken bir diğer nokta ise fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Geri beslemeli ağlarda aktivasyon

fonksiyonunun türevi de kullanıldığı için hesaplamaların yavaşlamaması için türevi kolay hesaplanır bir fonksiyon seçilir. Günümüzde en yaygın olarak kullanılan “Çok katmanlı algılayıcı”(ÇKA) modelinde genel olarak aktivasyon fonksiyonu olarak “Sigmoid fonksiyonu” kullanılır (Şekil 3.5.).



Şekil 3.5. Sigmoid fonksiyonu

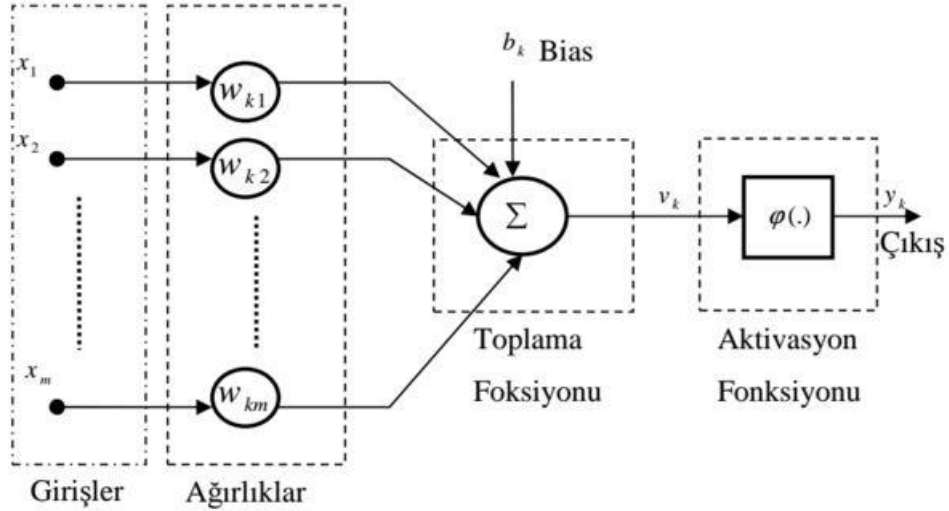
Bu tez çalışmasında sigmoid aktivasyon fonksiyonu tercih edilmiştir. Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur [31]. Doğrusal olmayışından dolayı YSA uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında değer üretir.

Hücrenin Çıktısı: Aktivasyon fonksiyonundan çıkan değer hücrenin çıktı değeri olmaktadır. Bu değer ister YSA'nın çıktısı olarak dış dünyaya verilir isterse tekrardan ağıın içinde kullanılabilir. Her hücrenin birden fazla girdisi olmasına rağmen bir tek çıktısı olmaktadır. Bu çıktı istenilen sayıda hücreye bağlanabilir.

3.2.2. Matematiksel yapay sinir hücresi modeli

YSA'yı oluşturan yapay sinir hücreleri (nöronlar) tek başlarına ele alındıklarında çok basit işlevleri olan işlemcilerdir. Bu küçük işlemci birimini üç temel bölüme ayırabiliriz. Şekil 3.6.'de bir yapay sinir hücresi gösterilmektedir. Diğer hücrelerden ya da dış ortamlardan hücreye giren bilgiler (yapay sinir girdileri) sinaptik bağlantılar üzerindeki ağırlıklarla çarpılarak bir birleştirme fonksiyonuna uygulanmaktadır. Elde edilen toplam, yapay sinir hücresi aktivasyon fonksiyonundan geçirilerek çıkışlar elde

edilir. Birleştirme fonksiyonu, bir hücreye gelen net girdiyi hesaplayan bir fonksiyondur ve genellikle net girdi, girişlerin ilgili ağırlıklarla çarpımlarının toplamıdır. Birleştirme fonksiyonu, ağ yapısına göre maksimum alan, minimum alan, çarpım ya da toplam fonksiyonu olabilir. Aktivasyon fonksiyonu ise birleştirme fonksiyonundan elde edilen net girdiyi bir işlemde geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur [22].



Şekil 3.6. Yapay sinir ağı matematiksel gösterimi

Şekilde yola çıkarak Yapay sinir hücresi için denklem yazılırsa;

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1.1)$$

$$v_k = u_k + b_k \quad (1.2)$$

$$y_k = \varphi(v_k) \quad (1.3)$$

w_{kj} =ağ ağırlıkları, x_j =girdi değerleri, u_k =girdilerin ve ağ ağırlıklarının toplamı, b_k =eşik değeri, y_k =çıkış değeri, $\varphi(0)$ ise aktivasyon fonksiyonu göstermektedir. Her bir girdideki değişim, YSH çıkışında bir değişime neden olmakta ve bu değişimin genliği, girişin etki derecesini belirleyen ağırlığa, toplayıcının eşik değerine ve aktivasyon fonksiyonuna bağlıdır. Yukarıdaki modelden de gözleneceği üzere eşik

değeri girdilerden bağımsız olmasından dolayı bütün girişler sıfır olsa bile çıkışta bir $\varphi(0)$ değeri gözlenir [23].

3.2.3. Geri yayılım algoritması

Bu tez çalışmasında geri yayılma algoritması kullanılmıştır. Geri yayılma algoritması, basitliği ve uygulamadaki görüş açısı gibi başarılarından dolayı ağ eğitimi için en popüler algoritmalarından biridir [24]. Bu algoritma; hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Geri yayımlı öğrenme kuralı ağ çıkışındaki mevcut hata düzeyine göre her bir tabakadaki ağırlıkları yeniden hesaplamak için kullanılmaktadır. Bir geri yayımlı ağ modelinde giriş, gizli ve çıkış olmak üzere 3 katman bulunmakla birlikte, problemin özelliklerine göre gizli katman sayısını artırabilmek mümkündür. Geri yayılım çok katmanlı ağlarda kullanılan delta kuralı için geliştirilmiştir bir algoritmadır. Bu algoritma çok katlı ağlarda hesap işlerini öğrenmede kullanılabilir. Geri yayılım ağında hatalar, ileri besleme aktarım işlevinin türevi tarafından, ileri besleme mekanizması içinde kullanılan aynı bağlantılar aracılığıyla, geriye doğru yayılmaktadır. Öğrenme işlemi, bu ağda basit çift yönlü hafıza birleştirmeye dayanmaktadır [25].

Geri beslemeli YSA'da ileri beslemeli olanların aksine bir hücrenin çıktısı sadece kendinden sonra gelen hücrenin katmanına girdi olarak verilmez. Kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir hücreye de girdi olarak bağlanabilir.

Bu yapısı ile geri beslemeli YSA doğrusal olmayan dinamik bir davranış göstermektedir. Geri besleme özelliğini kazandıran bağlantıların bağlantı şekline göre geri aynı YSA'yla farklı davranışta ve yapıda geri beslemeli YSA elde edilebilir.

Çok katmanlı ağlar (ÇKA) üç ayırt edici özelliğe sahiptir. Bunlar aşağıda verilmiştir.

1. Her bir nöronun modeli bir nonlinear aktivasyon modeli içerir (Denklem 3.1).

$$y_i = \frac{1}{1+e^{-v_i}} \quad (3.1)$$

Denklem 3.1’de geçen ifadelerin aşağıda tanımlanmıştır.

y_i : Nöronun çıkışı

i : Nöron

v_i : Uyarılmış yerel alan (bütün sinaptik girişlerin toplamı olan ağırlıklar)

2. Ağ bir veya birden fazla katman içerir.
3. Ağın sinapsları tarafından belirlenen güçlü bağlantılar içerir. Geri yayılım algoritması denklemini ise aşağıda verilmiştir.

Q katmanlı geri yayılım algoritması;

$q = 1, 2, 3, \dots, Q$ katman numarası,

$p H_i$: q 'inci katmandaki i biriminin girdisi,

$q i y$: q 'inci katmandaki i biriminin çıktısı,

$q w_{ij}$: $(q-1)$ 'inci katmandaki i birimini, q 'ncü katmandaki j birimine bağlayan ağırlık olmak üzere;

Adım 1: w 'ye reel değerli küçük rastlantısal sayıları başlangıç değeri olarak atanır.

Adım 2: Rasgele bir (giriş-hedef) çalışma modeli seçilir ve q katmanındaki her bir j birimi için ileri yönde 'çıkış' değerleri hesaplanır. Böylece çıkış, Denklem 3.2' deki gibi olur.

$$y_i^q = f(\sum_i y_i^{q-1} w_{ij}^q) \quad (3.2)$$

Adım 3: Çıkış birimleri için hata terimleri hesaplanır (Denklem 3.3).

$$\delta_i^q = (y_i^q - y_i^p) f'(h_i^q) \quad (3.3)$$

Adım 4: $q = Q, Q-1, \dots, 2, 1$ katmanlarındaki tüm i birimleri için geriye yayılımla deltaları yani gizli katman birimleri için hata terimleri hesaplanır (Denklem 3.5).

$$\delta_i^{q-1} = f(h_i^{q-1}) \sum_j y_j^q w_{ij}^q \quad (3.5)$$

Adım 5: Bütün bu ağırlıklar w_{ij} 'leri kullanılarak güncellenir (Denklem 3.6).

$$w_{ij}^{yeni} = w_{ij}^{eski} + \Delta w_{ij}^q$$

$$\Delta w_{ij}^q = n \delta_i^q y_j^{q-1} \quad (3.6)$$

Adım 6: Adım 2'ye dönüp, toplam hata kabul edilebilir bir düzeye gelene kadar her bir p modeli için işlemler tekrarlanır.

Geri yayılım algoritmasının amacı uygunluk fonksiyonunu minimum yapmaktır. Uygunluk fonksiyonu YSA'nın ağırlık değerlerine bağlı olduğundan, algoritma YSA ağırlıklarının en uygun biçimde değiştirilmesi işlemlerinden oluşmaktadır [26].

3.2.4. YSA'da geri yayılım çalışma prosedürü

YSA'nın çalışma prosedürünü maddeler halinde aşağıda açıklanmaktadır.

1. Örneklerin toplanması; Ağın çözmesi istenilen olay için daha önce gerçekleşmiş örneklerin bulunması adımıdır. Ağın eğitilmesi için örnekler toplandığı gibi (eğitim seti), ağın test edilmesi içinde örneklerin (test seti) toplanması gerekmektedir. Ağın eğitilmesi sırasında test seti hiç ağa gönderilmez. Eğitim setindeki örnekler tek tek gösterilerek ağın olayı öğrenmesi sağlanır.
2. Ağın topolojik yapısının belirlenmesi; Öğrenilmesi istenen olay için oluşturulacak olan ağın yapısı belirlenir. Kaç tane girdi ünitesi, kaç tane ara katman, her ara katmanda kaç tane hücre elemanı ve kaç tane çıktı elemanı olması gerektiği belirlenmektedir.

3. Öğrenme parametrelerinin belirlenmesi; Ağın öğrenme katsayısı, süreç elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.
4. Ağın başlangıç değerlerinin atanması; Hücre elemanlarını birbirine bağlayan ağırlık değerlerinin ve eşik değere başlangıç değerinin atanması yapılır.
5. Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi; Ağın öğrenmeye başlaması, öğrenme kuralına uygun olarak ağırlıkların değiştirilmesi için ağa örnekler (girdi/çıkı değerleri) belirli bir düzeneğe göre gösterilir.
6. Öğrenme sırasında ileri hesaplamaların yapılması; Verilen girdi için ağın çıkı değerinin hesaplanır.
7. Gerçekleşen çıkının beklenen çıkı ile karşılaştırılması; Ağın ürettiği hata değerlerinin hesaplanır.
8. Ağırlıkların değiştirilmesi; Geri hesaplama yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değiştirilmesi yapılır.
9. Öğrenmenin tamamlanması: İleri beslemeli sinir ağı öğrenmeyi tamamlayıncaya, yani gerçekleşen ile beklenen çıkı arasındaki hatalar kabul edilir düzeye ininceye kadar devam eder.

Ağın kendisine gösterilen girdi örneği için beklenen çıkıyı üretmesini sağlayacak ağırlık değerleri başlangıçta rastgele atanmakta ve ağa örnekler gösterildikçe ağırlıklar değiştirilerek istenen değerlere ulaşması sağlanmaktadır. İstenen ağırlık değerlerinin ne olduğu bilinmemektedir. Bu nedenle YSA'nın davranışlarını yorumlamak ve açıklamak mümkün olmaz.

Bazı durumlarda ağın takıldığı yer hata düzeyinin üstünde kalabilir. Bu durumda ağın olayı öğrenmesi için bazı değişiklikler yapılarak yeniden eğitilmesi gerekir. Bunlar; Başlangıç değerlerinde, ağ topolojisinde, ağın parametrelerinde değişiklikler yapılabilir. Ağ sunulan verilerin gösterimi ve örneklerin formülasyonu değiştirilerek yeni örnek seti oluşturulabilir. Öğrenme setindeki örneklerin sayısı artırılabilir veya azaltılabilir.

İleri beslemeli sinir ağının yerel sonuçlara takılıp kalmaması için momentum katsayısı geliştirilmiştir. Ağların eğitilmesinde diğer önemli bir sorun ise öğrenme süresinin çok uzun olmasıdır. Ağırlık değerleri başlangıçta büyük değerler olması durumunda ağın yerel sonuçlara düşmesi ve bir yerel sonuçtan diğerine sıçramasına sebep olmaktadır. Eğer ağırlıklar küçük aralıkta seçilirse o zaman da ağırlıkların doğru değerleri bulması uzun zamanlar almaktadır. Bazı problemlerin çözümü sadece 200 iterasyon sürerken bazıları 5-10 milyon iterasyon sürmektedir.

Ağın öğrenmesinin gösterilmesinin en güzel yolu hata grafiğini çizmektir. Her iterasyonda oluşan hatanın grafiği çizilirse hatanın zaman içinde düştüğü gözlenebilir. Belirli bir iterasyondan sonra hatanın daha fazla azalmayacağı görülür. Bu ağın öğrenmesinin durduğu ve daha iyi bir sonuç bulunamayacağı anlamına gelir. Eğer elde edilen çözüm kabul edilemez ise o zaman ağ yerel bir çözüme takılmış demektir.

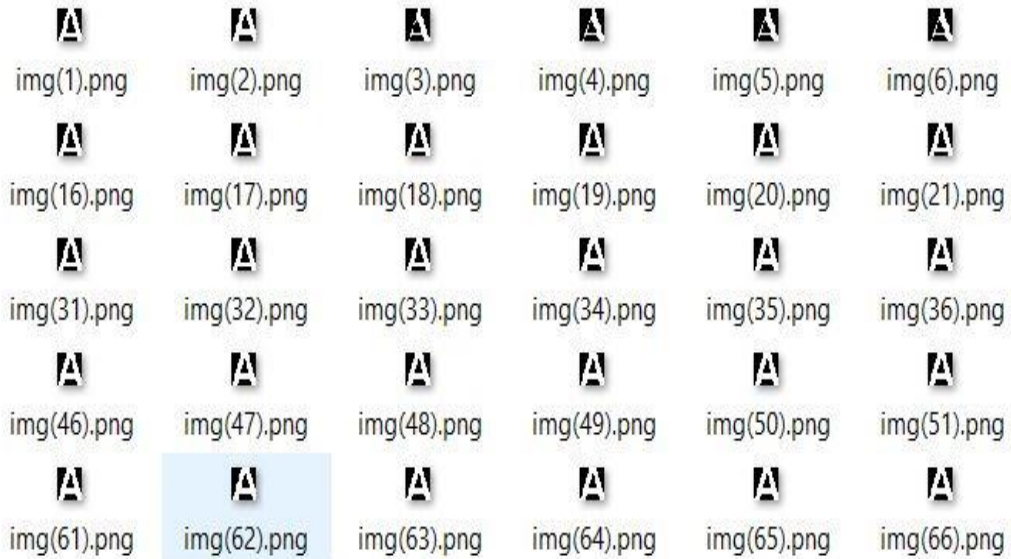
3.2.5. Yapay sinir ağlarının eğitilmesi

İnsan beyni doğumdan sonraki gelişme sürecinde çevresinden duyu organlarıyla algıladığı davranışları yorumlar ve bu bilgileri diğer davranışlarında kullanır. Yaşadıkça beyin gelişir ve tecrübelenir. Artık olaylar karşısında nasıl tepki göstereceğini çoğu zaman bilmektedir. Fakat hiç karşılaşmadığı bir olay karşısında yine tecrübesiz kalabilir. YSA'nın öğrenme sürecinde de dış ortamdan girişler alınır, aktivasyon fonksiyonundan geçirilerek bir tepki çıkışı üretilir. Bu çıkış yine tecrübeyle verilen çıkışla karşılaştırılarak hata bulunur. Çeşitli öğrenme algoritmalarıyla hata azaltılıp gerçek çıkışa yaklaşılmaya çalışılır. Bu çalışma süresince yenilenen YSA'nın ağırlıklarıdır. Ağırlıklar her bir çevrimde yenilenerek amaca ulaşılmaya çalışılır. Amaca ulaşmanın veya yaklaşmanın ölçüsü de yine dışarıdan verilen bir değerdir. Eğer YSA'ya verilen giriş-çıkış çiftleriyle amaca ulaşılmış ise ağırlık değerleri saklanır. Ağırlıkların sürekli yenilenerek istenilen sonuca ulaşılan kadar geçen zamana öğrenme adı verilir. YSA öğrendikten sonra daha önce verilmeyen girişler verilip, sinir ağı çıkışıyla gerçek çıkış yaklaşımı incelenir. Eğer yeni verilen örneklere de doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir. Sinir ağına verilen örnek sayısı optimum değerden fazla ise sinir ağı işi öğrenmemiş, ezberlemiş demektir. Genelde eldeki

örneklerin %80'i ağı verilip ağı eğitilir. Daha sonra kalan %20'lik kısım verilip ağı davranışı incelenir. Böylece ağı testi yapılmış olur.

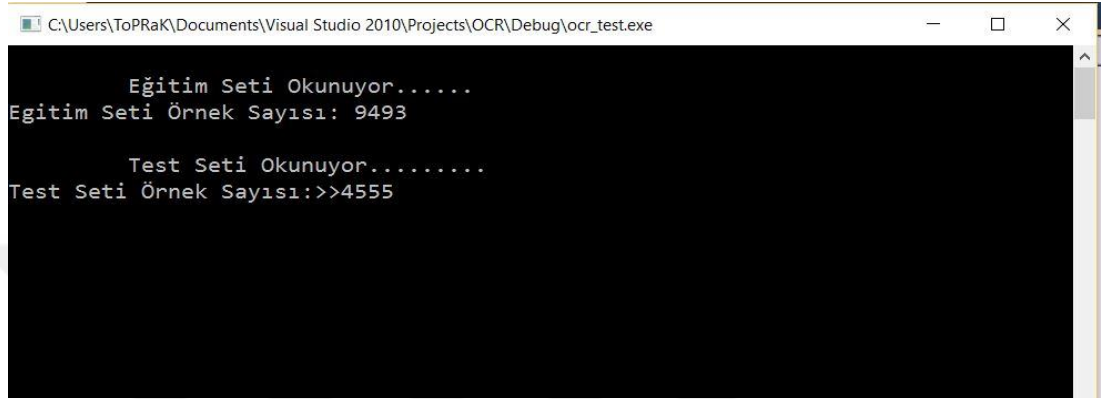
Bu tez çalışmasında yapay zekâ tekniklerinden olan YSA ile farklı disiplinler bir araya getirilmeye çalışılmıştır. İlk uygulama aşamasında YSA oluşturulurken, OCR sistemleri imgeler üstündeki karakterlerin tanınması ve Günümüz Türkçesinin Osmanlı Türkçesine kendi ait kurallarıyla birlikte çevrilmesi işlemi yapılmıştır.

Başlangıçta, Türkçe kelimelerinde kullanılan büyük ve küçük harfler elde edilmiştir. Bu harfler temel görüntü işleme işlemlerinde geçirildikten sonra her harf 16X16'lık pikseller halinde siyah-beyaz imge formatında kaydedilmiştir (Şekil 3.7.). Bu harfler sistem tarafından taranarak her harf beyaz piksellere 1 siyah piksellere 0 atayarak eğitim seti oluşturulmuştur (Şekil 3.8.).



Şekil 3.7. Eğitim seti resim örnekleri

Günümüz Türkçesinde toplam 58 harf vardır. Eğitim setinde her harf için 200 örnek kullanılmıştır. Eğitim setinin örnekleri fontu Times New Roman, Arial ve Black Arial olup 18 puntoluk seçilmiştir. Şekil 3.9.'de gösterildiği gibi, toplam eğitim seti için örneklerin %60'ı (9493 örnek) test seti için de örneklerin %40'ı (4555 örnek) kullanılmıştır.



```
C:\Users\ToPRaK\Documents\Visual Studio 2010\Projects\OCR\Debug\ocr_test.exe

Eğitim Seti Okunuyor.....
Eğitim Seti Örnek Sayısı: 9493

Test Seti Okunuyor.....
Test Seti Örnek Sayısı:>>4555
```

Şekil 3.9. Eğitim ve test setlerin hazırlanması

YSA eğitilmesi için eğitim seti ve test seti hazırlanmıştır. YSA eğitim setini sınıflandırma işlemi ise şu şekilde yapılmaktadır. Günümüz Türkçe alfabesinde 58 tane harf vardır. Her harf için ayrı bir sınıf oluşturulmuştur. Büyük A için 0 sınıfı büyük B için 1 sınıfı vb. her bir sınıf içinde 200 örnek vardır. Eğitim seti oluşturulduğu her imgenin sayısal değerinin sonuna o sınıfın değeri ilave edilmektedir (Şekil 3.10.).



.1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,0,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,0	0
.0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,1	1
.0,0,0,0,0,0,1,1,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0	2
.1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0	3
.0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0	4
.0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1	5
.0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0	6
.0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0	7
.0,0,1,1,1,1,1,0,0,1,0	8
.0,0,1,1,1,1,1,1,0,0,1,0	9
.1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,10	10
.1,11	11
.1,1,1,1,1,1,1,1,0,0,1,0	12
.0,13	13
.1,14	14
.1,1,0,0,1,1,1,1,1,1,1,1,1,0,0,0,15	15
.0,16	16
.0,1,1,1,1,1,1,0,0,0,17	17
.0,0,1,1,1,1,1,1,0,0,18	18
.0,0,1,1,1,1,1,0,0,0,0,19	19
.0,0,0,0,1,1,1,1,1,1,1,1,0,20	20
.1,1,0,21	21
.1,1,1,0,22	22
.0,0,1,1,1,1,1,1,1,0,1,23	23
.0,0,0,0,0,0,1,1,0,1,24	24
.1,0,25	25
.0,0,0,0,0,0,0,0,0,1,26	26
.1,1,0,27	27
.0,0,0,0,0,0,0,0,1,28	28
.1,1,1,1,1,1,1,1,0,0,1,29	29
.0,0,0,0,1,1,1,1,0,0,1,30	30
.0,0,1,1,1,1,1,1,0,0,1,31	31
.1,1,0,32	32
.0,0,1,1,1,1,1,1,0,0,1,33	33
.0,0,1,1,1,1,1,1,0,0,1,34	34
.1,1,1,0,0,0,0,0,0,0,0,1,35	35
.0,0,0,0,1,1,1,1,0,0,1,36	36
.0,0,0,0,1,1,1,1,0,0,1,37	37
.0,0,0,1,38	38
.1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,39	39
.1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,40	40
.1,1,1,1,1,1,0,0,0,1,41	41
.0,0,0,0,0,0,0,0,0,1,42	42

Şekil 3.9. Eğitim seti örnekler ve sınıflandırma şekli

İleri beslemeli sinir ağında bağlantıların ağırlık değerlerinin başlangıçta belirlenmesi ağın performansını yakından ilgilendirmektedir. Genel olarak ağırlıklar belirli aralıklarda atanmaktadır. Bu aralık eğer büyük tutulursa yerel çözümler arasında sürekli dolaştığı küçük olması durumunda ise öğrenmenin geç gerçekleştiği görülmüştür. Bu değerlerin atanması için henüz standart bir yöntem yoktur. Tecrübeler 0.1 ile 1.0 arasındaki değerlerin başarılı sonuçlar ürettiğini göstermektedir. Fakat bu tamamen öğrenilmesi istenen problemin niteliğine bağlıdır. Başlangıç değerleri kadar öğrenme ve momentum katsayılarının belirlenmesi de ağın öğrenme performansı ile yakından ilgilidir. Öğrenme katsayısı ağırlıkların değişim miktarını belirlemektedir. Eğer büyük değerler seçilirse o zaman yerel çözümler arasında ağın dolaşması ve osilasyon yaşaması söz konusu olmaktadır. Küçük değerler seçilmesi ise öğrenme zamanını artırmaktadır. Tecrübeler genellikle 0.2-0.4 arasındaki değerlerin kullanıldığını göstermiştir. Bazı uygulamalar öğrenme katsayısının 0.6 değerini aldığı zaman en başarılı sonuçları verdiğini göstermiştir. Bu durum tamamen probleme bağlıdır. Benzer şekilde momentum katsayısı da öğrenmenin performansını etkiler. Momentum katsayısı bir önceki iterasyondaki değişimin belirli bir oranının yeni değişim miktarına eklenmesidir. Bu özellikle yerel çözümlere takılan ağların sıçrama ile daha iyi sonuçlar bulmasını sağlamak amacıyla önerilmiştir. Bu değerlerin küçük olması yerel çözümlerden kurtulmayı zorlaştırabilir. Çok büyük değerler ise çözüme ulaşmada sorunlar yaşatabilir. Tecrübeler momentum katsayısının 0.6-0.8 arasında seçilmesinin uygun olacağını göstermiştir. Problemin niteliğine göre kullanıcının almasında fayda vardır.

Bu tez çalışmasında YSA'nın eğitilmesi için OpenCV kütüphanesi kullanılmıştır. OpenCV görüntü işlemede kullanılan açık kaynak kodlu bir kütüphanedir. OpenCV kütüphanesinde görüntü analizi, histogram, renk uzayları, temel dönüşümler, filtreler, eşleştirme, kenar algılama, köşe algılama, şekil algılama, nesne algılama, nesne tanıma gibi genel görüntü işleme konularına yer verilmiştir. Ayrıca OpenCV algoritmaları kullanılarak, Yüz tanıma, İşaret Dili Tanıma, Hareket Yakalama, Algılama ve Takibi, Hareketli Robot Teknolojileri gibi alanlarda uygulamalar geliştirilir. Basitçe bir resim ya da bir video üzerinde birçok işlem ve gelişmiş uygulamalar yapma olanağı sağlayan bu kütüphane robotbilim, otomotiv, tıp, tüketim, güvenlik, üretim ve araştırma alanları

gibi birçok alanda kullanılmaktadır. Kütüphane C dilinde yazılmıştır. C++ ara yüzü de eklenmiştir. Daha çok C ve C++ dilleri kullanılarak geliştirilmeye açık bir kütüphanedir. Genel olarak çoğu işletim sistemlerinde çalışmaktadır (Windows, Linux, OS, Android, iOS vb.). Ayrıca Opencv birçok programlama dilini de desteklemektedir (C/C++, Java, PHP, MATLAB, NET, Python, Delphi).

OpenCV’de sistemin eğitilmesi için ilk önce bir YSA oluşturulur. Bu ağın katman sayısı (input, hidden, output-L1,L2,L3), aktivasyon fonksiyonu, öğrenme katsayısı, eşik değeri belirlenir. Bu çalışmada birçok deneme yapıldıktan sonra en uygun ağ yapısı Şekil 3.11.’da gösterilen yapı olduğu tespit edilmiştir.

```
// http://docs.opencv.org/modules/ml/doc/neural\_networks.html

CvANN_MLP nnetwork(layers, CvANN_MLP::SIGMOID_SYM,0.2,1);
CvANN_MLP_TrainParams params(

    cvTermCriteria(CV_TERMCRIT_ITER+CV_TERMCRIT_EPS, 1000, 0.000001),

    CvANN_MLP_TrainParams::BACKPROP,
    // http://docs.opencv.org/modules/ml/doc/neural\_networks.html#cvann-mlp-trainparams
    0.1,
    0.1);
```

Şekil 3.11. Tercih edilen ağ yapısı

Ağ yapısı tercih edildikten sonra YSA’nın parametre değerleri seçilir. İterasyon sayısı, nöron ağırlıkları, geri yayılım algoritması, momentum değeri dq_scale değeri seçilir. Ağ yapısı ve eğitim parametreleri seçildikten sonra ağ eğitimi başlar. Ağ eğitimi bittikten sonra elde edilen sınıflandırma sonuçları *.xml dosyası şeklinde kaydedilir.

3.2.6. Kullanılan YSA'nın test edilmesi

Eğitim işlemi tamamlandıktan sonra eğitilmiş YSA'nın test edilmesi işlemine geçilir. Bazı test sonuçları Şekilde 3.12.'de gösterilmiştir. YSA için aynı parametreler ile farklı eğitim setleri denendiğinde eğitim setinin örnek sayısının YSA üzerinde etkili olduğu görülmüştür.

```

C:\Users\ToPRaK\Documents\Visual Studio 2010\Projects\OCR\Debug\OCR_ANN.exe
Testing Sample 3433 -> class result (class 23)
Testing Sample 3434 -> class result (class 36)
Testing Sample 3435 -> class result (class 23)
Testing Sample 3436 -> class result (class 36)
Testing Sample 3437 -> class result (class 23)
Testing Sample 3438 -> class result (class 36)
Testing Sample 3439 -> class result (class 23)
Testing Sample 3440 -> class result (class 15)
Testing Sample 3441 -> class result (class 23)
Testing Sample 3442 -> class result (class 36)
Testing Sample 3443 -> class result (class 23)
Testing Sample 3444 -> class result (class 36)
Testing Sample 3445 -> class result (class 23)
Testing Sample 3446 -> class result (class 15)
Testing Sample 3447 -> class result (class 15)
Testing Sample 3448 -> class result (class 15)
Testing Sample 3449 -> class result (class 36)
Testing Sample 3450 -> class result (class 15)
Testing Sample 3451 -> class result (class 15)
Testing Sample 3452 -> class result (class 15)

Test Sonucu
Dogru Siniflama: 3064 (88.7344%)
Yanlis Siniflama: 389 (11.2656%)

```

Şekil 3.12. YSA'nın test edilmesi

Eğitilen YSA'nın doğru sonuçlar verip vermediğini kontrol etmek için test edilmesi gerekir. Bu çalışmada örneklerin %40'ı test seti için kullanılmıştır. YSA'nın farklı parametrelere göre farklı sonuçlar verdiği görülmüştür. Bu sonuçlar Tablo 3.1.'de gösterilmiştir.

Tablo 3.1. YSA'nın farklı parametreleri için başarımları

No	Eğitim Örneği	Test Örneği	Fonksiyon Eşik Değeri	Ara Katman	İterasyon Sayısı	Başarımları Oranı
1	6628	3453	0.2	32	1000	95%
2	6628	3453	0.6	64	1000	99%
3	6628	3453	0.2	40	1000	99%
4	6628	3453	0.2	28	1000	93%

3.3. Osmanlı Türkçesi

Osmanlıca bin senelik Türk İslâm tarihinin, kültürünün, medeniyetinin, biliminin, felsefesinin oluşturduğu, Arap harflerini alfabe olarak kabul etmiş bir milletin konuşma ve yazı dilidir. Osmanlıca muhteva ve şekil itibariyle birbirinden ayrılmaz üç unsurdan oluşmaktadır. Osmanlıcanın elbisesi Arap harfleri, vücudu kelimeler, ruhu kelimelerin manalarıdır [22].

Pratikte kısaca Osmanlıca diye de adlandırılan -her ne kadar bu adlandırma tehlikeli de olsa- Osmanlı Türkçesi, 15. yüzyılın sonlarından 20. yüzyılın başlarına kadar Osmanlı devletinin sınırları içinde kullanılan yazı dilidir.

Bu dönemin en belirgin özelliği, Arapça, Farsça gibi yabancı dillerden oldukça fazla kelimenin Türkçeye girmiş olmasıdır. Klâsik bir edebiyat oluşturma ve sanat yapma anlayışıyla Türk yazı dili âdetâ Arapça, Farsça ve Türkçe kelimelerden oluşan üçüz bir dil hâline getirilmiştir. Konuşma diliyle yazı dili arasındaki farklar her geçen gün artarken bir tarafta konuşulan fakat yazılmayan bir dil; diğer tarafta yazılan fakat konuşulmayan bir dil ortaya çıkmıştır. Halka, halkın diliyle seslenen halk şairlerinin yalın Türkçesi yanında sanat yapma endişesiyle sadece belli bir zümrenin anlayabildiği, halkın anlamadığı, konuşmadığı unsurlar divan şairleri aracılığıyla dile girmiştir. Bu durum 17. yüzyılda doruğa çıkmıştır. Dilde ortaya çıkan bu ikilikten kaynaklanan anlaşılma sorunu, 17. yüzyılda mahallîleşme hareketiyle yavaş yavaş çözülmeye başlamıştır. Bu çözüme 18. yüzyıl boyunca ve Tanzimat'a kadar devam ettiyse de Türkçe, yabancı kelimelerle yüklü ağır bir dil olarak varlığını Batı Türkçesinin üçüncü dönemini oluşturan Türkiye Türkçesine kadar sürdürmüştür [23].

3.3.1. Osmanlı Türkçesi genel özellikleri

Osmanlı Türkçesinde kullanılan harfler Arap harfleridir. 28 harften oluşan Arap alfabesinde olmayan “ p, ç, j, g ve nazal n (پ چ ج گ ن) ” harfleri, Farsça ve Türkçe sözcüklerdeki sesleri karşılamak üzere ilave edilmiştir. Türkçe olmayan sözcüklerdeki

sesleri doğru yazmak ve sesletmek için kimi harflerin özelliklerinin bilinmesi gerekmektedir.

1. ح ذ ص ض ط ظ ع ve hemze(ء) yalnızca Arapça sözcüklerde geçen harflerdir.
2. ژ harfi Farsçaya özgüdür.
3. ح ذ ص ض ط ظ ع ق harfleri Farsçada yoktur.
4. ح ذ ص ط harfleri çok az sözcükte kullanılmıştır.
5. ح ذ ص ض ط ظ ع ve hemze gerek ses ve gerekse bunları karşılayan harf olarak Türkçede yoktur.
6. د ذ ر ز و ا harfleri kendilerinden sonraki harfle birleşmez; bu nedenle ortada bu harflerden birini barındıran sözcükler bölünerek yazılır, sözcükte bütünlük olmamaktadır.
7. ث harfi, peltek çıkarılan bir sestir; dil ucu üst dişlere dokundurularak çıkarılır.
اثبات - isbât, حادثه-hâdise, تمثال – timsâl, تثبیت – tesbît
8. ص harfi ince bir “s” sesiyken ص harfi, kalın sesli sözcüklerde özellikle de başta kullanılan fakat sesletim olarak “s” den farklı olmayan bir sestir. تصور – saf, صف – tasavvur, صميم – Samim, صدا – seda, صورت - suret
9. ه (güzel h) “h” sesini karşılayan diğer iki harfe (ح خ) göre daha ince bir sestir.
10. ح harfi, boğazımızın en arkasından çıkarılan bir sestir. حبس – habs (haps), حمام – hamam, حج – hac, حضرت – hazret
11. خ harfi, ح’den farklı olmamakla birlikte daha hırıltılı bir sestir. Azeri lehçesine yakın olan Doğu bölgesi ağızlarında (çoh-çok, bahtih-baktık) k/h arası bir sestir. Türkçede, k’den gelen h sesi dışında (kangı-hangi, kanı-hani) “h” harfi yoktur. خم – ham, خراب harab, مخصوص mahsus, خس – has

12. ز harfi, Türkiye Türkçesinde kullanılan “z” sesidir. **زيرا** – zira
13. ذ harfi, dil ucunun dişlere değmesiyle çıkarılan “d,y,z” sesleri arasında peltek bir “z”dir. **ذكر** - zikr, **اذييت** - eziyet, **اذان** - ezan, **لذيذ** - leziz, **ذكا** - zekâ, **ذكي** – zeki
14. ض harfi, “d” ile “z” arasında bir sestir. Bu sesi içeren sözcükler kimi zaman “d” kimi zaman “z” olarak seslendirilir. **قاضي** – kadı, **قضا** – kaza, **ضربه** – darbe, **فضا** – feza, **بياض** – beyaz
15. ظ harfi, orta kalınlıkta bir “z” sesidir. Dilin ortasının üst damağa ve üst-ön dişlere değmesiyle çıkarılır. **نظر** – nazar, **حافظ** – hafız, **ظلمت** – zulmet
16. ط harfi, kalın sesli sözcüklerde özellikle de başta kullanılan fakat sesletim olarak “t” den farklı olmayan bir sestir. **طرف** – taraf, **مضبوط** – mazbut, **اطراف** – etraf
17. ع harfi, ses çıkarabileceğimiz en derin bölgeden çıkan bir sestir. Türkçede ya hiç okunmaz ya da uzunluğa veya kesmeye yol açar. **عسكر** - asker, **تعريف** - ta’rif, **الوداع** elveda’, **علم** - ilim [24].

3.3.2. Kelime çeviri algoritması

Osmanlı Türkçesinde, Türkçe asıllı kelimelerin dışında Arapça ve Farsça asıllı yaklaşık 8 bin kadar kelime vardır. Arapça ve Farsça kelimeler kendi gramer yapılarını korumaktadır. Türkçe asıllı kelimeler ise, Osmanlı Türkçesindeki imla kaidelerine göre çevrilmektedir. Osmanlı Türkçesi okunuşu bazı okutucu harfler vasıtasıyla olmaktadır. Osmanlı Türkçesinde 4 tane okutucu harf vardır. Bu okutucu harfler Günümüz Türkçesinin ünlü seslilerine karşılık gelmektedir. Örneğin; Elif harfi “a”, He harfi “e-a”, Ye harfi “i-ı”, Vav harfi “o-ö-u-ü” seslerini karşılamaktadır [30]. Bu

Tablo 3.2.'de gösterilmiştir. Osmanlı Türkçesinin Okutucu harfler ve imla kaideleri;

Tablo 3.2. Osmanlı Türkçesi ile günümüz Türkçesi'nin karşılaştırılması

ا	o	و	ی
باب باب A B A B	o ~ o ~ E N E N	بو U B	ای لی I K I
بابا B A B A	ننه N E N E	بو B U	ایلی I K I

“Elif” harfi ve “a” sesi;

1. Okutucu “elif” harfleri, “a” sesinin bulunduğu bütün hecelerde gösterilir.
2. İlk iki hecedeki “a” seslerini gösteren okutucu “elif”lerin birincisi, bazen terk edilebilir.
 - a. Ancak, ince harfle veya kapalı hece ile başlayan kelimelerin bu hecelerindeki okutucuların terkedilmemesi daha uygundur.
3. Sona gelen “a” sesleri, ekseriyetle “elif” yerine “he” harfi ile gösterilir.
 - a. İsimlerin yalın hâli ile eklerin sonundaki “a” seslerinin hemen hemen tamamı, “he” okutucusu ile yazılır.
 - b. Sonu “a” sesi ile biten emir kipleri ile az da olsa bazı kelimeler, “elif” okutucusu ile yazılır.
4. “Elif” harfinin üzerine med işareti konduğunda, “a” sesiyle okunur. Bu kurallara göre sisteme verilen bir resim ilk önce resim üzerindeki karakterler ayrıştırılmıştır. Ayrıştırılan karakterler Osmanlı Türkçesine çevrilmiştir (Şekil 3.13.).

“He” harfi ve “e - a” sesleri

1. “He” harfi, ilk hecede okutucu olarak gelmez.
2. “He” harfi; orta hecelerde, bunun bir açık hece olması hâlinde okutucu olarak kullanılır, kapalı hece olması hâlinde ise kullanılmaz.
3. Sondaki “e” sesleri, mutlaka “he” harfi ile gösterilir.

4. İsimlerin yalın hâli ile eklerin sonlarına gelen “a” sesleri için çoğunlukla, “he” harfi kullanılır.
5. Okutucu olan “he” harfi, kendinden sonra gelen harfle birleşmez.
6. Kelime başlarındaki “e” sesleri, okutucu kullanmaksızın “elif” harfi ile gösterilir. Okunuşta kolaylık olmak üzere, bu harf üzerine “hemze” işareti de konabilir.

Bu kurallara göre sisteme verilen bir resim ilk önce resim üzerindeki karakterler ayrıştırılmıştır. Ayrıştırılan karakterler Osmanlı Türkçesine çevrilmiştir (Şekil 3.14.).

“Vav” harfi ve “ u - ü - o - ö ” sesleri

Okutucu “vav” harfleri, “u, ü, o, ö” seslerinin bulunduğu bütün hecelerde gösterilir.

1. İlk iki hecede okutucu olarak gelen “vav” harflerinin birincisi, bazen terk edilebilir. Bunlar terk edilmeden de ayrı ayrı yazılırlarsa, yanlış olmaz.
 - a. İkinci hece “cim veya çe” harfi ile başlıyorsa, ikinci okutucu “vav” ın terk edildiği kelimeler de vardır.
2. Çok az da olsa bazı “ö ve ü” sesleri “vav” harfi beraberinde hemze ile yazılır.
 - a. Birkaç kelime dışında pek yerleşmemiştir. Ancak karışıklığa sebep olabilecek yerlerde, daha önceden klişeleşmiş bir yapı yoksa kullanımı uygundur.
3. Farsça bazı kelimelerde “hı” harfinden sonra gelen “vav” harfleri, yazıldığı hâlde okunmazlar. (vâv-ı ma’dûle)
4. Kelime başlarındaki “u, ü, o, ö” sesleri, “elif” harfinden sonra “vav” okutucusu yazılarak gösterilir. Bir kolaylık olmak üzere “ü, ö” sesleri için, “elif” harfi üzerine hemze işareti konabilir.

“Ye” harfi ve “ ı - i ” sesleri

1. Okutucu “ye” harfi, “ ı, i “ seslerinin bulunduğu bütün hecelerde gösterilir.
2. Klişeleşmiş bazı kelime ve eklerde ise, okutucu “ye” harfi hiç gelmeyebilir.
3. Eklerdeki “u” ve “ü” sesleri, umumiyetle “ye” okutucusu ile gösterilir.
 - a. İyice yerleşmiş olanlar hariç, kelimelerin asli yapısı sonundaki “u ve ü” sesleri “ye” ile yazılmaz.
4. “Ye” harfi okutucu olduğunda, kendinden sonra “ye” harfi ile başlayan eklere birleşmez.

- a. Fiil eklerinde ise böyle değildir, “ye” harfleri bitişik yazılabilir.
- 5. Bazı kelimelerin ilk hecelerindeki “e” sesleri, “ye” okutucusu ile gösterilir.
- 6. Kelime başlarındaki “ı, i” sesleri, “elif” harfinden sonra “ye” okutucusu yazılarak gösterilir.

Günümüz türkçesinin Osmanlı türkçesine çevrilme işlemi yapılırken, Osmanlı türkçesindeki okutucu harflerin kurallarına göre çevrilmektedir.

3.3.2. Osmanlı türkçesindeki kelime özellikleri

Her dilin kendine özgü kuralları olduğu gibi Türkçenin de kendine özgü kuralları vardır. Bu bölümde Türkçe sözcüklerin özelliklerine ait kurallardan bahsetmek istiyorum. Bu bilgiler ışığında yabancı sözcükleri daha kolay bir şekilde ayırt edebiliriz.

1. Büyük ünlü uyumu kuralı (Kalınlık-incelik uyumu) Türkçedeki ünlülerin yarısı kalın yarısı ince ünlülerden oluşmaktadır.

Kalın ünlüler: a, ı, o, u

İnce ünlüler: e, i, ö, ü

2. Bir kelimedeki ünlülerin tamamı ya kalın ünlü olacak ya da ince ünlü olacak. Şayet hem kalın ünlü hem de ince ünlü barındıran bir kelime varsa o kelime Türkçe değildir.(elma, anne, kardeş, hani vb. kelimeler zamanla değişikliğe uğradıkları için bu kurala uymazlar.) orman, ağaç, çiçek, böcek kelimelerindeki ünlüler ya hep kalın ünlü ya da hep ince ünlüdür. Badem, karanfil, fidan, ticaret, kitap kelimelerindeki ünsüzlere bakıldığında zaman kalın ve ince ünlüler aynı kelimedede mevcut. Tek heceli ve birleşik kelimelerde bu kural aranmaz.

3. Küçük ünlü uyumu kuralı (düzlük-yuvarlaklık uyumu) Küçük ünlü uyumu kuralı ile ilgili esaslar;

Düz ünlüler: a, e, ı, i

Yuvarlak ünlüler: o, ö, u, ü,

- a. Bir kelime düz ünlü ile başladığı zaman düz ünlü ile devam etmesi gerekir. Yani a ünlüsü ile başladı ise a veya ı ünlüsü ile devam etmesi gerekir. Ya da e ile başlamışsa kelime e veya i ile devam etmesi gerekir.
- b. Yuvarlak bir ünsüzle başlamışsa düz genişle devam etmesi gerekir. o,ö,u,ü ile başlamışsa kelimemiz a veya e ile devam etmesi gerekir.
- c. Yuvarlak ile başlamışsa dar yuvarlak ile bitmesi gerekiyor. o,ö,u,ü ile başlarsa u,ü ile bitmesi gerekir. Küçük ünlü uyumu kuralı incelendiğinde ilk hecenin dışındaki kelimelerde o ve ö ünlüsünün bulunmadığını görebiliriz. Çünkü: o ve ö ünlüleri yuvarlak geniş ünlülerdir.

Koro (İtalyanca), şato (Fransızca), doktor (Fransızca), foto (Fransızca), daktilo (Fransızca), Aktör (Fransızca), armatör (Fransızca), hoparlör (Fransızca), formatör (Fransızca) aynı kurala göre düz ünlü ile başlayıp yuvarlak ünlü ile biten kelimeler de bu kurala uymazlar. a ile başlayıp u ile biten kelimeler bu kurala uymazlar. Kavun, karpuz, yağmur, hamur, çamur, havuç.

Türkçede sapkalı a (â) sesi bulunmaz. Bu harfi barındıran sözcükler Türkçe değildir.

Kâğıt (Farsça), kâtip (Arapça), fedâ (Arapça), rüzgâr (Farsça)

4. Türkçede iki ünlü yan yana gelmez. Şair, şiir, dair, ait, aile vb. (Örnek kelimeler Arapça kökenlidir)

5. Türkçe sözcüklerin içinde aynı ünsüz yan yana gelmez. Millet, hürriyet, cadde, iddia, zerre, bakkal vb.(Örnek kelimeler Arapça kökenlidir.)(anne, elli kelime istisna)

6. Türkçe kelimeler b,c,d,g ünsüzleriyle bitmez. Ab (Farsça), hac (Arapça), kod (Fransızca), lig (Fransızca) (ad, sac kelimeler istisna)

7. Türkçe kelimeler çift ünsüzle başlamaz. Trafik, tren, fren, kral, grup.. (bu kelimeler dilimize Fransızcadan girmiştir)

8. Türkçe sözcüklerde f, h, j sesi yoktur. Bu kelimeler Türkçe değildir. jilet, jandarma (İtalyanca), müjde (Farsça), makyaj (Fransızca), baraj (Fransızca), fakir (Arapça), filiz (Rumca), fiyat (Arapça), fuar (Fransızca), tayfa (Arapça), kafa (Arapça), hazan (Farsça), hayat (Arapça), hol (İngilizce), sabah (Arapça), huzur (Arapça), cerrah (Arapça)

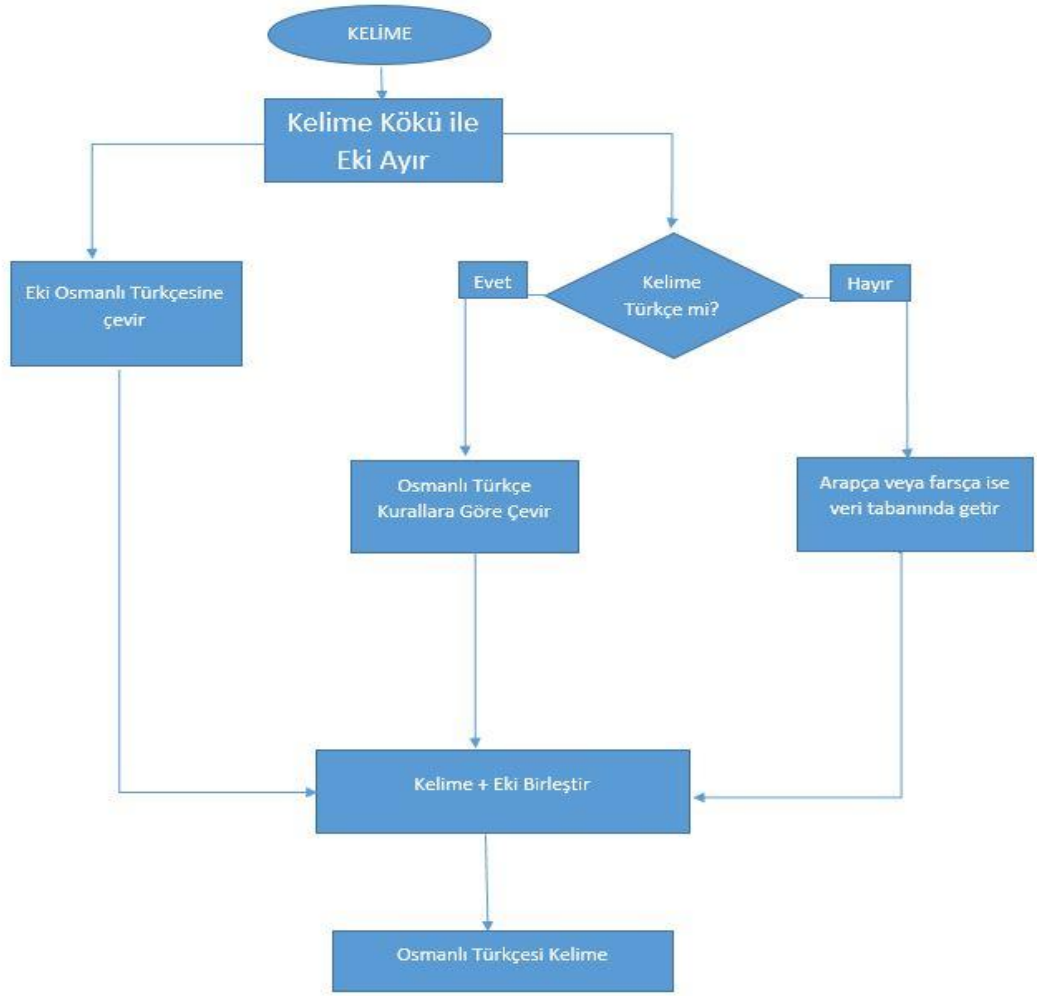
9. Türkçe sözcüklerde “b” ünsüzünden önce “n” harfi olmaz. Böyle durumlarda “n” ünsüzü “m” ye çevrilerek kullanılır. çenber-çember, kanber-kamber, penbe-pembe, lanba-lamba,canbaz-cambaz (istisna durumlar: onbaşı, binbaşı, Safranbolu vb.)
10. Türkçede “c, g, l, m, n, r, v, j, f” sesleri kelimelerin başında bulunmaz. Ancak yansıma (ses taklitleri) sözcükler bu kurala dahil değildir.

Günümüz türkçesinin kelime yapısını bilmek Günümüz türkçesinin Osmanlı türkçene çevrilmesinde kolaylık sağlayacaktır. Çünkü Türkçe asıllı kelimeler osmanlı türkçesinin imla kaidelerine göre çevrilmiştir. Arapça ve farsça kelimeler ise kendi gramer yapılarını korumaktadır. Bu kelimelerin osmanlı türkçesine çevrildiğinde yanlış olduğu görülmektedir. Bu duruma örnek olarak bazı kelimeler Tablo 3.3 gösterilmiştir. Aşağıdaki algoritma Günümüz Türkçesinin Osmanlı Türkçesine çevrilmesini göstermektedir (Şekil 3.14).

Günümüz Türkçesini Osmanlı Türkçesine çevrilme işlemi yapılırken başarı oranı artırmak için kelimenin ilk başta ekinden ayrılması gerekmektedir. Ayrılan kelime ile türkçe asıllı bir kelime mi yoksa arapça ve farsça asıllı bir kelime mi tespit edildikten sonra çeviri işlemlerinden geçirilerek kelime osmanlı türkçesine çevrilir.

Tablo 3.3. Arapça ve Farsça kelimelerin gramer yapıları ve yazılışları

KELİME	YANLIŞ	DOĞRU
Millet	میللت	ملت
Türkiye	تورکییه	ترکیه
Cumhuriyet	جومهورییت	جمهوریت
Aile	ایله	عائله
Mimar	میمار	معمار
Daire	دائره	دائرة
Eser	اسر	اثر
Fikir	فیکیر	فکر
Hikâye	هیکایه	حکایه
Hoca	حوجا	خوجا



Şekil 3.13. Osmanlı Türkçesi çeviri algoritması

Şekil 3.12.'de gösterilen algoritmanın çıktısı aşağıdaki Şekil 3.14.'te gösterilmiştir.

Tarihçe

Kendisine çağdaş uygarlığın gerektirdiği her türlü donanıma sahip bireyler yetiştirmeyi amaç edinmiş olan Sakarya Üniversitesi'nin çekirdeğini 1970 yılında açılan Sakarya Mühendislik ve Mimarlık Yüksek okulu oluşturmuştur.

Bu okul 1971 yılında Sakarya Devlet Mimarlık ve Mühendislik Akademisi'ne dönüşmüş,

1982-1992 yılları arasında İstanbul Teknik Üniversitesi'ne bağlı bir Fakülte olarak öğretim vermiştir.

3 Temmuz 1992 tarih ve 3837 sayılı kanun ile Sakarya Üniversitesi kurulmuştur

تاریخچه کنديسنه چاغطاش اويگارلغك گركتيرديگي هر تورلو طونانها

صاحب بيريلر يتيشتميمي اماچ ادينمش اولان ساقاريه

اونيورسيتسي'نك چكيرديگيني ۱۹۷۰ ييلينده اچيلان ساقاريه

موهنديسلك وه معمارلق يوكسك اوقولو اولوشطورمشدر. بو اوقول

۱۹۷۱ ييلنده ساقاريه دولت معمارلق و موهنديسلك اقاده ميسنه

دونوشمش ۱۹۸۲-۱۹۹۲ ييللري الاسنده استانبول تكنيك

Şekil 3.14. Çeviri algoritmasının çıktısı

Elif okutucusu için sistemin çıktısı Şekil 3.15.'de gösterilmiştir.

E:\Tez Defteri\6.JPG

BABACAN AYAKTA AĞLAMAZ
PAŞA BABAM ATMACA YAKALAR
YAŞAYAN PARÇAYLA YAMA YAPMA
ÇAKAL KAÇAR KAPLAN YAKALAR

BABACAN AYAKTA AĞLAMAZ
 PAŞA BABAM ATMACA YAKALAR
 YAŞAYAN PARÇAYLA YAMA YAPMA
 ÇAKAL KAÇAR KAPLAN YAKALAR

باباجان ليقدہ اغلاماز
 پاشا بابام اطہاجا ياقالار
 ياشايان پارچايلا ياما ياپما
 چاقال قاچار کاپلان ياقالار

Günümüz Türkçesi

Osmanlı Türkçesi

Şekil 3.15. Elif okutucusu için sistem çıktısı

He okutucusu için sistemin çıktısı Şekil 3.16.'de gösterilmiştir.

C:\Users\ToPRaK\Documents\Visual Studio 2010\Projects\YeterArtik\YeterArtik7.JPG

DEREDEN TEKNE GEÇMEZ
EMEL EVDE SERÇE BESLER
TEMBEL BEBEK DEDEMENDEN EKMEK BEKLER
DEDEM SERÇEYE EKMEK ATAR

DEREDEN TEKNE GEÇMEZ
EMEL EVDE SERÇE BESLER
TEMBEL BEBEK DEDEMENDEN EKMEK BEKLER
DEDEM SERÇEYE EKMEK ATAR

دردن تکنه گچمز
اهل اوده سرچه بسار
تمبل بیک ددمندن اکک بکلر
ددم سرچیه اکک اطار

Günümüz Türkçesi

Osmanlı Türkçesi

Şekil 3.16. He okutucusu için sistem çıktısı

BÖLÜM 4. TARTIŞMA VE SONUÇ

YSA insan beyni gibi düşünüp uzman kişi gibi karar vermektedir. Bu çalışmada, YSA ile karakter tanıma işlemi kullanılarak günümüz Türkçesi'nin Osmanlıcaya çevrilmesi işlemi gerçekleştirilmiştir. Bu çalışma iki aşamadan oluşmaktadır. Birinci aşama imgenin üzerindeki karakterin başarılı bir şekilde eğitilmiş ağ tarafından tanınmasıdır. İkinci aşama ise tanınan karakterlerin başarılı bir şekilde günümüz Türkçesinin Osmanlı Türkçesine çevrilmesi işlemidir. Birinci aşamada, YSA'yı eğitmek için eğitim seti ve test setleri görüntü işleme tekniklerinden faydalanılarak hazırlanmıştır. Ayrıca eğitim setinin örnek sayısı YSA'nın başarısını etkilediği gözlemlenmiştir. YSA eğitilirken farklı ağırlıkların kullanılmasının, aktivasyon fonksiyonunun eşik değerinin ve ara katman sayısının artırılıp veya azaltılmasının başarı oranını etkilediği tespit edilmiştir. Aynı zamanda görüntü işleme tekniklerinde görüntünün filtreleme yapılıp yapılmaması, görüntünün eşik değerinin ayarlanması ve görüntünün açma ve kapama işlemlerinin de başarı oranını etkilediği gözlemlenmiştir. Bunlar göz önünde bulundurularak sistemin maksimum performans ağ yapısının 256 tane giriş, 32 tane gizli ve 58 tane çıkış katmanı içermesi gerektiği görülmüştür. YSA sınıflandırıcı girişine uygulanmış test örneklerinden %99 oranında doğru tanıma başarımı elde edildiği görülmüştür. İkinci aşamada ise, Osmanlı Türkçesinin kendi kelime hazinesi ile birlikte Arapça ve Farsça kelimeler de bulunmaktadır. Sistemin ikinci aşamasında Osmanlı Türkçesine ait kelimeler başarılı bir şekilde çevrilmektedir. Fakat Osmanlı Türkçesinde arapça ve farsça kelimeler kendi gramer yapılarını korumaktadırlar. Bundan dolayı sistemin ikinci aşamasında arapça ve farsça kelimelerin Osmanlı Türkçesine çevrilmesi sistemin başarısını etkilemektedir. Bu arapça ve farsça kelimeler gramer yapılarına göre çevrilme işlemine başarılı bir şekilde eklenirse, sistemin başarısı artacaktır.

KAYNAKLAR

- [1] Yanıkođlu,B.,Kholmatov,A., "Türkçe İçin Geniş Sözcük Dađarcıklı Doküman Tanıma Sistemi", Turkish-siu2003, Sabancı Üniversitesi, İstanbul, 2003.
- [2] <http://www.osmanlicadergi.com/neden-osmanlica-ogrenmeliyim/>
ErişimTarihi:12.04.2016.
- [3] Mikael Laine,Olli S. Nevalainen "A Standalone OCR System For Mobile Cameraphones", IEEE(2006).
- [4] Jaehwa Park, Young-Bin Kwon, "An embedded OCR: A Practical Case Study of Code Porting for a Mobile Platform", IEEE(2009).
- [5] Gorski, N., Anisimov, V. , Augustin, E., Baret, O., Price, D., Simon, J.C. (1999). "A2iA Check Reader: A Family of Bank Check Recognition Systems", Proc. 5th Int. Conf. Document Analysis and Recognition, pp.523-526.
- [6] Ni, D.X. (May 2007). Application of Neural Networks to Character Recognition, Proceedings of Students/Faculty Research Day.
- [7] Mani, N., Srinivasan, B. (October 1997). "Application of Artificial Neural Network Model for Optical Character Recognition", IEEE International Conference, pp.12-15.
- [8] Ahmed, O. A.-W. (June 1994). Application of Artificial Neural Networks to Optical Character Recognition, Master's Thesis, King Fahd University Of Petroleum & Minerals, Saudi Arabia.
- [9] Leung, C.H. (1997). Feature Selection in the Recognition of Handwritten Chinese Characters, Engng Applic. Artif. Intell., v.10, n.5, pp.495-502.
- [10] Zhang, L.X., Zhao, Y.N., Wang, J.X. (November 2002). "Feature Selection In Recognition Of Handwritten Chinese Characters", Proceedings of the 1st International Conference on Machine Laming and Cybernetics.
- [11] Inoue, K., Miyazaki, R., Suzuki, M. (1998) "Optical Recognition of Printed Mathematical Documents", Proceedings of the 3th Asian Technology Conference in Mathematics, Springer-Verlag, pp. 280–289.

- [12] Singh, R., Yadav, C.S., Verma, P., Yadav, V. (January-June 2010). Optical Character Recognition (OCR) for Printed Devnagari Script Using Artificial Neural Network, International Journal of Computer Science & Communication, v.1, n.1, pp.91-95.
- [13] Koga, M., Mine, R., Kameyama, T., Takahashi, T. (2005) Camera-based Kanji OCR for Mobile-phones: Practical Issues, Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition, ICDAR'05.
- [14] Park, J., Kwon, Y.-B. (2009). An Embedded OCR: A Practical Case Study of Code Porting for a Mobile Platform, Chinese Conference on Pattern Recognition, CCPR '09.
- [15] Rodríguez, A., Kim, S., Kim, J.H., B.- Fernández, Y. (2009). "English to Spanish Translation of Signboard Images from Mobile Phone Camera", IEEE SOUTHEASTCON '09.
- [16] Osmanlı Türkçesi TDK Büyük Türkçe Sözlük.
- [17] Şükrü Halûk Akalın, "Türk Dünyasında Dil", Yeni Türkiye, Temmuz-Ağustos 2013, Yıl 9, Sayı 53-54, Ankara, s. 362-363.
- [18] Şemseddin Sami, "Lisan-ı Türkî (Osmanî)", Hafta (mecmuası), C. 1, S. 12, 10 Zilhicce 1298 (3 Kasım 1881), s. 177-178.
- [19] Prof.Dr. Muharrem Ergin, Osmanlı Türkçesi.
- [20] Orhan Koloğlu, Avrupa Kışkacında Abdülhamid, İstanbul, 1998, s. 406.
- [21] Mütercim Asım Efendi, Burhan-ı Katı Tercemesi, TDK Yayınları, Ankara 2000.
- [22] Öztemel, E. 2003. Yapay Sinir Ağları, Papatya Yayıncılık, İstanbul.
- [23] I. Haykin, S. 1999. Neural Networks A Comprehensive Foundation. 2nd edition.
- [24] Gonzalez R. C. and Woods R. E. "Digital Image Processing", AddisonWesley, 1992.
- [25] Elmas Ç. Yapay Sinir Ağları, Seçkin Yayıncılık, Ankara (2003).
- [26] Lin C. Lee G. (1996). Neural Fuzzy Systems, Prentice Hall, 236-240, 242, 445-448.
- [27] <http://dergi.irfanmektebi.com/2008/02/osmanlica-nedir/> Erişim Tarihi: 15.04.2016.

- [28] <http://www.anadilim.org/osmanli-turkcesi.html> Eriřim Tarihi 09.05.2016.
- [29] Millî Eđitim Bakanlıđı, Osmanlı Trkęesi ders kitabı ISBN 978-975-11-3729-6.
- [30] Osmanlıca İmla Mfredatı Hayrat Neřriyat M.Ali Ensari ISBN 975-7245-10-0.
- [31] Han, Jun; Morag, Claudio (1995). "The influence of the sigmoid function parameters on the speed of backpropagation learning". In Mira, Jos ; Sandoval, Francisco. From Natural to Artificial Neural Computation. pp. 195–201.



ÖZGEÇMİŞ

İshak Dölek, 01.04.1989'da Van'da doğdu. İlk, orta ve lise eğitimini Van'da tamamladı. 2007 yılında Cumhuriyet Lisesi'nden mezun oldu. 2008 yılında başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nü 2013 yılında bitirdi. 2013 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nde yüksek lisans eğitimine başladı. Yüksek lisans eğitimine devam etmektedir. 2013 yılında serbest çalışmaya başladı. 2016 yılında Teknoarge Teknoloji LTD. ŞTİ' de yazılımcı olarak çalışmaya başladı. Halen devam etmektedir.