



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

iOS PLATFORMUNDA GÖRME ENGELLİLER İÇİN TL TANIMA UYGULAMASI

Özgür ŞAHİN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Haziran-2017

KONYA

Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

iOS PLATFORMUNDA GÖRME ENGELLİLER İÇİN TL TANIMA UYGULAMASI

Özgür ŞAHİN

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Barış KOÇER

2017, 72 Sayfa

Jüri
Yrd. Doç. Dr. Barış KOÇER
Doç. Dr. Mustafa Servet KIRAN
Yrd. Doç. Dr. Mehmet HACIBEYOĞLU

Günümüzde artan kredi kartı kullanımına rağmen kâğıt para kullanımı hala geçerliliğini korumaktadır. Türk Lirasının kupürlerinde görme engelliler için Braille kabartmaları ve boyut farklılıkları mevcut olmasına rağmen kabartmaların zamanla aşınması ve kupür boyut farklılıklarının milimetrik olması, bu kupürlerin ayırt edilmesini zorlaştırmaktadır. Bu soruna bir çözüm sunabilmek için bu çalışmada yapay sinir ağlarından faydalanılmıştır. Bu bağlamda banknotların fotoğraflarıyla veri kümesi oluşturulup bu kümeyle ImageNet kümesi üzerinde eğitilmiş olan ResNet50 modeli tekrar eğitilerek yeni bir yapay sinir ağları modeli elde edilmiştir. Bu model mobil uygulamada kullanılarak, görüntüden banknotun sınıfının belirlenmesi sağlanmıştır. Sonuç olarak ortaya çıkan banknot tipi mobil uygulama kullanıcısına ses veya titreşim ile bildirilmektedir. Eğitilen model, oluşturulan veri kümesi ve geliştirilen mobil uygulama kütüphanesi açık kaynak olarak kullanıma sunulacaktır.

Önerilen yöntemin test sonuçlarının gösterdiğine göre bu sistem gerçek zamanlı olarak TL tanımada yüksek verimlilik sağlamaktadır.

Anahtar kelimeler: Evrişimsel Sinir Ağları, Öğrenme Transferi, Banknot Tanıma

ABSTRACT

MS THESIS

TURKISH BANKNOTE RECOGNITION APPLICATION FOR VISUALLY IMPAIRED

Özgür ŞAHİN

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE OF COMPUTER ENGINEERING

Advisor: Asst. Prof. Dr. Barış KOÇER

2017, 72 Pages

Jury
Asst. Prof. Dr. Barış KOÇER
Assoc. Prof. Mustafa Servet KIRAN
Asst. Prof. Dr. Mehmet HACIBEYOĞLU

Despite the usage of credit cards, using paper currency is still common. Turkish Banknotes have Braille Signs and differs in sizes but these signs disappear with time and size differences are not so significant. These factors make it difficult to classify banknotes for visually impaired. In order to offer a solution to these problem, artificial neural network is used in this research. Pre-trained ResNet50 model which is trained with ImageNet dataset, has been retrained and new model is produced. This model is used in a mobile application and banknote type is identified with the video input of a mobile phone. The user is notified with sound and vibration about the classification result. The retrained model, the dataset and the mobile application will be published as open source.

In consideration of the test result, proposed method offers significant accuracy in recognizing banknote types.

Keywords: Convolutional Neural Networks, Transfer Learning, Banknote Recognition

ÖNSÖZ

Bu çalışmamda, bilgi ve desteğini benden esirgemeyen danışman hocam Yrd. Doç. Barış KOÇER'e ve Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü Öğretim Elemanları'na teşekkürlerimi sunarım.

Özgür ŞAHİN
KONYA-2017

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **viii**

1. GİRİŞ **9**

2. KAYNAK ARAŞTIRMASI **11**

2.1 Görüntü İşleme Teknikleri 11
2.1.1. Görüntünün elde edilmesi 11
2.1.2. Görüntü iyileştirme 11
2.1.3. Morfolojik işlemler 18
2.1.4. Bölütleme (Segmentasyon) 19
2.1.5. Özellik tespiti ve özellik çıkarımı 20
2.1.6. Nesne Tanıma 26
2.2. Yapay Sinir Ağları 28
2.2.1. Yapay sinir ağlarının temelleri 28
2.2.2. Aktivasyon fonksiyonları 31
2.2.3. Maliyet fonksiyonları 33
2.2.4. Optimizasyon fonksiyonları 34
2.2.5. Düzenlileştirme (regularization) 38
2.2.5. Öğrenme yöntemleri 40
2.2.6. Derin sinir ağları 40
2.2.7. YSA kullanan banknot tanıma çalışmaları 48

3. MATERYAL VE YÖNTEM **55**

3.1 Veri Kümesi Oluşturulması ve Ön İşleme 55
3.3 ESA Eğitim Süreci 56
3.3 Sıfırdan Model Eğitimi 57
3.4 Modellerin Yeniden Eğitilerek Karşılaştırılmaları 61
3.5 Model Çıktısını iOS Platformunda Kullanma 64

4. SONUÇLAR VE ÖNERİLER **69**

4.1 Sonuçlar 69
4.2 Öneriler 69

KAYNAKLAR **70**

ÖZGEÇMİŞ **73**

SİMGELER VE KISALTMALAR

Kısaltmalar

CNN: Convolutional Neural Network (Evrişimsel Sinir Ağı)

ESA: Evrişimsel Sinir Ağı

FLANN: Fast Library for Approximate Nearest Neighbors (En Yakın Komşuları Yakınsamak İçin Hızlı Kütüphane)

ILSVRC: ImageNet Large-Scale Visual Recognition Challenge (ImageNet Büyük Ölçekli Görüntü Tanıma Yarışması)

RBSA: Radyal Bazlı Sinir Ağları

RGB: Red, Green, Blue (Kırmızı, Yeşil, Mavi)

ROI: Region of Interest (İlgi Alanı)

SIFT: Scale-Invariant Feature Transform (Ölçekten Bağımsız Özellik Dönüşümü)

SURF: Speeded-Up Robust Features (Hızlandırılmış Gürbüz Öznitelikler)

YSA: Yapay Sinir Ağı

1. GİRİŞ

Bilgisayar bilimleri zaman zaman farklı disiplinlerden ilham almıştır, bu alanlardan birisi de biyolojidir. Özellikle duyu organları ve beyin arasındaki veri aktarımı ve beyindeki veri işleme mekanizmaları bilgisayar bilimine ilham kaynağı olmuştur. Beynin girdi kanalları olarak da adlandırabileceğimiz duyu organları çeşitli formdaki verileri alarak beyne iletir. Görme duyusu da bu kanallardan biridir. Göz, cisimlerden yansıyan fotonları beynin anlayabileceği bir formda iletir. Görüntü işlemenin de temeli; görüntüleri bilgisayarların anlayabileceği forma çevirmeye dayanır. Görüntü işleme sayesinde insandaki görme duyusuna benzer prensiplerle renk, boyut, konum algılanarak sınıflandırma yapılır. Bu süreç ayrıca görüntünün sayısallaştırılması, bölümlere ayrılması, iyileştirilmesi ve döndürülmesi gibi biçimsel işlemleri de kapsamaktadır.

Görüntü işleme günümüzde birçok kullanım alanına yayılmıştır. Trafikte plakaların tespit edilmesi, resimlerin etiketlenmesi, görüntülerden suçluların tespiti gibi çok çeşitli alanlarda görüntü işleme yöntemlerinden yararlanılmaktadır. Bugün popüler resim uygulamaların çoğu, görüntü işleme teknikleri kullanır, bu sayede resimlere filtre uygular, iyileştirme sağlar ve yeniden boyutlandırabilirler.

Bilgisayar bilimlerine ilham veren bir diğer organımız ise beynimizdir. Beyin, verileri nöronlardan oluşan ağlarla işler. Bu nöron yolları alışkanlıklara ve algılanan verilere göre sürekli yeniden şekil alarak öğrenme dediğimiz olguyu ortaya çıkarır. Beyindeki bu ağlardan ilham alınarak benzer modeller bilgisayar bilimlerinde de oluşturulmuştur. Bu yapılara yapay sinir ağları (YSA) adı verilir. YSA sayesinde bilgisayarlar verilerle eğitilerek öğrenebilen modeller oluşturulabilir. Bu modeller sınıflandırma, nesne tanıma ve hatta yeni veriler oluşturmada kullanılabilir. Günümüzde her formda veri (yazı, ses veya görüntü vb.) YSA'lar aracılığıyla işlenebilmektedir. Bu ağlar bir yazarın yazılarıyla eğitilerek yeni yazılar yazabilir veya yeni müzik parçaları besteleyebilirler. Verileri ön işleme tabi tutmadan da işleyebilirler, bu sayede anlamlı veriyi bulabilmek amacıyla algoritma geliştirme yükünden kurtuluruz, YSA verilerden kendi kendine anlam çıkarıp bunu yorumlayabilmektedir.

Bu çalışmanın odaklandığı alan, görme engellilerin banknotları ayırt etmesini kolaylaştırmaktır. Bu problem aslında bir görüntü sınıflandırma problemidir. Literatürdeki görüntü sınıflandırma problemleri hem görüntü işleme teknikleriyle hem de yapay sinir ağları kullanımıyla çözümlenmeye çalışılmıştır. 2012 yılındaki ILSVRC

(ImageNet Large-Scale Visual Recognition Challenge) yarışmasında, görüntü sınıflandırma ve tanıma alanında, AlexNet isimli bir YSA modeli (Alex K., 2012) büyük bir başarı sağlamıştır. Bu yıldan sonra YSA'lar özellikle de evrişimsel sinir ağları (convolutional neural networks) bu alanda gittikçe daha çok kullanılmaya başlanmıştır. Bu çalışmada iki alandaki çalışmalar incelenmiş olup, sınıflandırma için evrişimsel sinir ağı (ESA) kullanılmasına karar verilmiştir. Sıfırdan eğitmek maliyetli olduğu için öğrenim transferi yöntemiyle eğitilmiş ağlar kullanılmıştır. Bu kararın verilmesine neden olan araştırmalar çalışmada paylaşılmıştır. Sonuç olarak veri kümesi oluşturulup, YSA modeli eğitilmiş ve mobil uygulama geliştirilerek görüntü üzerinden tahminleme yapılarak, sesli ve titreşimli geri bildirim verilmesi sağlanmıştır.

2. KAYNAK ARAŞTIRMASI

Bu bölümde literatür taraması yapılarak görüntü sınıflandırma ve banknot sınıflandırma alanındaki çalışmalar araştırılmaktadır. Yapılan araştırmada görüntü sınıflandırma çalışmalarında ağırlıkla görüntü işleme teknikleri ve yapay sinir ağlarından faydalanıldığı görülmektedir.

2.1 Görüntü İşleme Teknikleri

Görüntüdeki verilerin işlenerek bilgisayarın anlayacağı şekle getirilmelerine görüntü işleme adı verilmektedir. Görüntü işleme; görüntünün elde edilmesi, iyileştirilmesi, özellik tespiti ve nesnelerin tanımlanması gibi farklı alt kategorilerde incelenmektedir. Görüntü işlemenin temel adımları ve banknot tanıma çalışmalarında kullanımları bu bölümde detaylı olarak incelenecektir.

2.1.1. Görüntünün elde edilmesi

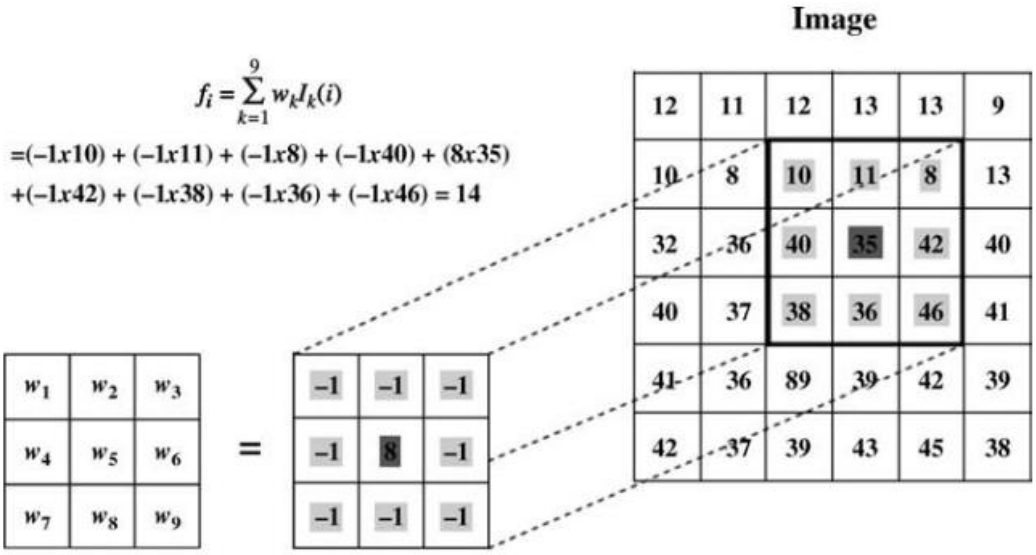
Görüntü işlemenin ilk adımıdır. Görüntüyü elde etmek için kaynaktan yansıyan ışığın sensörler tarafından ölçülmesi gerekmektedir. İstenilen görüntü çeşidine göre (x-ray, 3d, termografik vb.) hangi optik sistemin (lens, sensör) kullanılacağına bu adımda karar verilmektedir. Ayrıca bu süreç görüntüyü ölçeklendirme gibi ön işlemleri de içermektedir.

2.1.2. Görüntü iyileştirme

Görüntü elde edildikten sonraki adım ise görüntü iyileştirme adımıdır. Görüntü iyileştirme, resmin bazı özelliklerini ön plana çıkartarak uygulamalar tarafından daha kolay işlenebilmesini sağlamaktır. Gürültünün yok edilmesi, resim kenarlarının keskinleştirilmesi ve yumuşak odak (blurring) efekti popüler görüntü iyileştirme yöntemleridir.

Günlük hayatta kullandığımız sayısal resimlerin oluşturduğu alana uzaysal alan (spatial domain) denmektedir. Uzaysal alandaki resimlerin pikselleri doğrudan doğruya işlenebilmektedir. İyileştirme işlemleri, uzaysal alan filtrelemesiyle yapılmaktadır. Filtreler resimler üzerindeki piksel değerlerini belirli oranda değiştiren yapılardır. Filtreleme işlemleri genellikle komşuluk ilişkisinde olan piksellerin üzerinde kayan bir pencere prensibiyle işlem yapmaktadır. Resimdeki her piksel, çevresindeki NxN boyutundaki matris üzerinde yapılan operasyona göre işlenir. Lineer uzaysal filtrelerde yeni piksel değeri, pikselin komşu piksel değerlerinin lineer bir kombinasyonuyla hesaplanmaktadır. Komşu piksellerin hangi lineer kombinasyonla alınıp işleneceğini filtre maskesi (kernel) belirlemektedir. Bu maske hedef pikselin belirtilen sınırlardaki her komşusu için bir ağırlık içeren bir dizidir. Filtreleme, bir matrisin kayan bir pencere

gibi tüm ilgi alanlarının üzerinden geçerek altına denk gelen piksellerin değerlerini onlara karşılık gelen ağırlıkla çarparak bunların toplamını alan ve bu toplamı tüm o ilgi alanına kopyalayan bir işlemdir. Maskedeki ağırlıklarla ona karşılık gelen piksellerin çarpımı Şekil 2.1'de gösterilmektedir (Solomon, 2011).



Şekil 2.1 3x3'lük maske ile resmin filtrelenmesi (Solomon, 2011)

Yukarıda gösterilen maske ve resim arasındaki işleme evrişim (konvolüsyon, filtreleme) adı verilmektedir. Bu işlemin formülü 2.1 numaralı denklemde gösterilmektedir.

$$f_i = \sum_{k=1}^{N} W_k I_k(i) \quad (2.1)$$

Bu denklemde $I_k(i)$ i. pikselin komşu piksellerini, k satır veya sütun bazında komşuluk alanında ilerleyen lineer indeksi, W_k maske değerlerini, f_i ise orijinal piksel değeri olan $I_k(i)$ değerinin filtrelenmiş sonucunu temsil etmektedir.

Lineer filtreleme (konvolüsyon) adımları şu şekildedir:

1. Filtre maskesi belirlenir.

2. Maske resim üzerinde kaydırarak ortasındaki pikselin resimdeki her bir hedef piksel ile kesişmesi sağlanır.

3. Maske altında kalan piksellerle, onlara karşılık gelen maskedeki değerlerin (ağırlıklar) çarpımları alınarak toplanır.

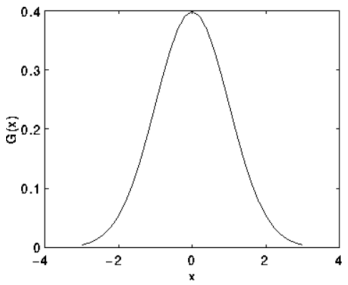
4. Sonuç değeri yeni resimdeki(filtrelenmiş) aynı yerlere kopyalanır. (Solomon, 2011)

Gürültü azaltma işlemi için de filtreler kullanılmaktadır. En çok kullanılan yöntemler; ortalama, ortanca ve Gauss filtreleri olarak sayılabilir.

Ortalama filtreleme yönteminde komşuluk matrisi içine denk gelen komşu piksellerin ortalaması alınarak piksel değerleri bu değer ile değiştirilmektedir. Ortanca filtreleme yönteminde piksel değerleri, komşuluk matrisine denk gelen piksellerin ortanca değeri ile değiştirilmektedir.

Bu yöntemlerden birisi olan Gauss filtresi de resmi flulaştırmak ve gürültüyü azaltmak için kullanılmaktadır. Bu filtrede amaç; komşu pikseller arasındaki dağılımı, Gauss eğrisine yakınlaştırmaktır. Aşağıdaki bir boyutlu Gauss filtresi denklemi ve frekans dağılımı görülmektedir. σ sembolü standart sapmayı göstermektedir.

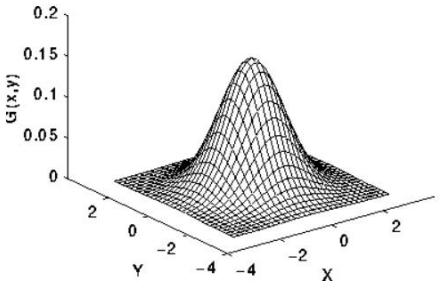
$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{2.2}$$



Şekil 2.2 Bir boyutlu Gauss dağılımı (Fisher, 2000)

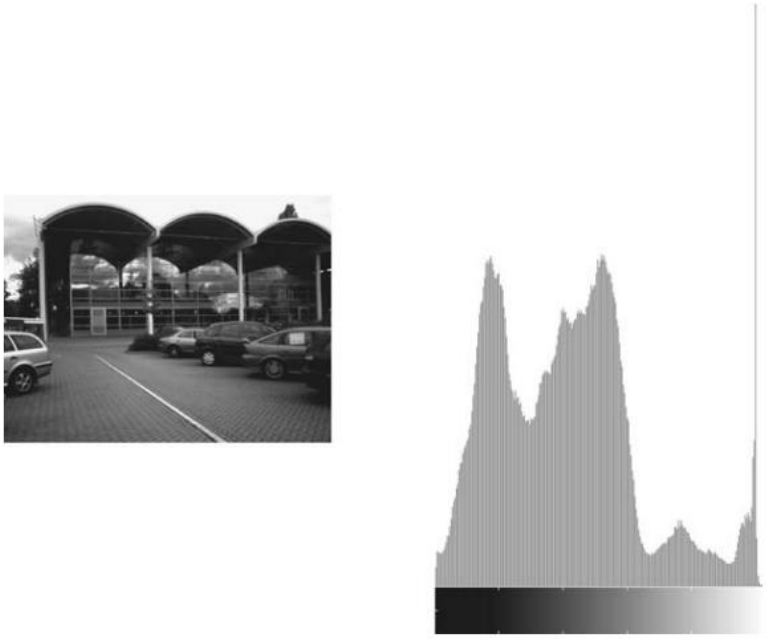
İki boyutlu bir Gauss filtresinde ise denkleme y ekseni de dahil olur. İki boyutlu Gauss denklemi ve frekans dağılım aşağıdaki gibidir.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.3}$$



Şekil 2.3 İki boyutlu Gauss dağılımı (Fisher, 2000)

Görüntü iyileştirmenin diğer yöntemleri ise histograma dayalı yöntemlerdir. Histogram bir görüntüdeki renk değerlerinin sayılarını gösteren grafiktir. Örnek bir resmin histogramı Şekil 2.4'de gösterilmektedir.



Şekil 2.4 Resim ve histogram grafiği (Solomon, 2011)

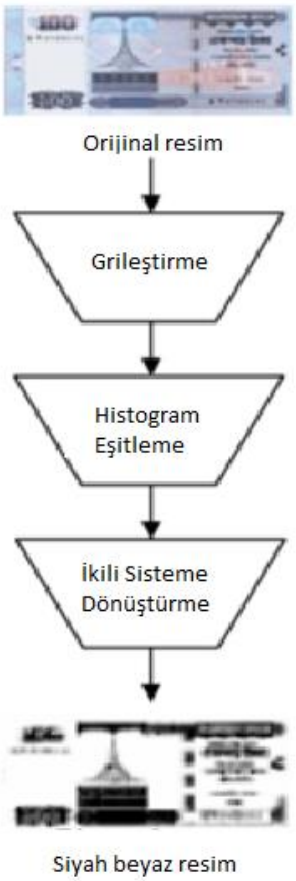
Histograma dayalı görüntü iyileştirme için başlıca iki yöntem vardır; bunlar histogram germe (histogram streching) ve histogram eşitleme (histogram equalization) işlemleridir.

Histogramı dar bir alana yayılmış, kontrastı iyi olmayan görüntüleri iyileştirmek için histogram germe yöntemi kullanılmaktadır. Temel mantığı histogramı geniş bir alana yaymaktır.

Histogram eşitleme ise bir resimdeki renk değerlerinin belli bir yerde gruplanmış olmasından kaynaklanan renk dağılımı bozukluğunu gidermek için kullanılan bir yöntemdir. Histogram eşitleme, fotoğraftaki piksellerin değerlerini 0-255 aralığında dağıtarak piksel değerlerinin aralarının açılmasını sağlamaktadır. Bu sayede görüntüde gözle görülür bir iyileşme olup detaylar ortaya çıkmaktadır. Ayrıca görüntüyü işlerken eşik seviyeleri daha iyi hesaplanıp daha verimli sonuçlar elde edilmektedir. Histogram eşitlemenin siyahlarla beyazlar birbirlerinden uzaklaşarak fotoğrafta bir keskinlik yarattığını gözlemlenmektedir. Bu yöntem görüntünün işlenebilirliğini kolaylaştırdığı için sık kullanılan yöntemlerden biridir.

Kontrast seviyesini artırmak için gri seviyedeki resim histogram eşitleme işleminden geçirilmektedir. Histogram eşitleme ile resmin histogramı dengelenerek yoğunluk frekansları resim genelinde aynı oranlarda dağılmış olur. Bu şekilde karşıtlık artırılmaktadır. Banknot tanıma üzerine yapılan çalışmada önemli alanlar koyu çizgiler, noktalar ve alanlar olduğu için eşitlenen resim daha sonra eşik değer belirlenerek siyah beyaza dönüştürülmektedir. Histogram eşitleme işlemi para tanıma uygulamalarında

sıklıkla kullanılmıştır. Örnek bir çalışmadaki kullanımı Şekil 2.5'de gösterilmektedir (Jahangir, 2007).



Şekil 2.5 Resim grileştirme süreci (Jahangir, 2007)

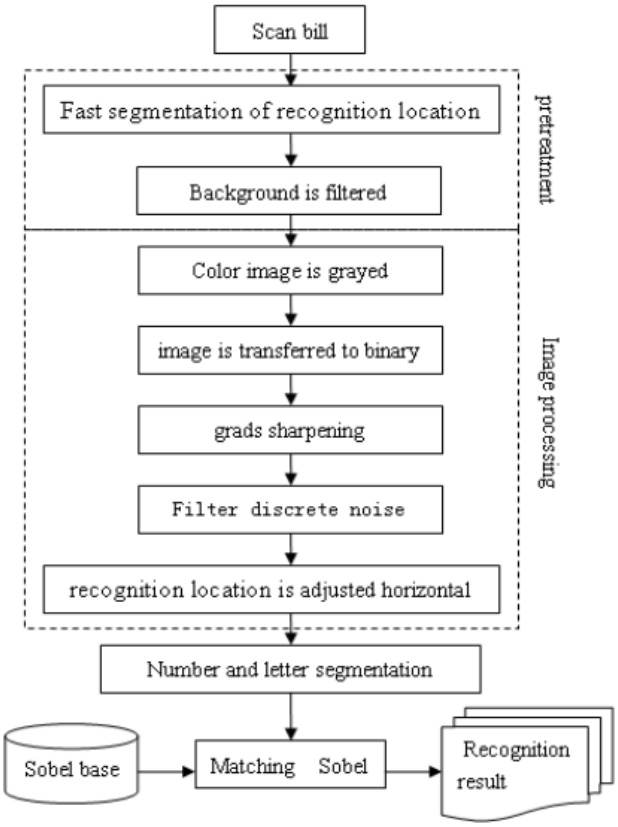
Bazı banknot çalışmalarında ise banknot resmin tamamını kaplamaz ve açısal olarak düzgün değildir. Şekil 2.6'da, dolar üzerinde yapılan banknot sınıflandırma çalışmasında, bu şekildeki bir banknot resmi üzerinde yapılan açısal düzeltme ve kırpma işlemleri gösterilmektedir.



Şekil 2.6 Açı düzeltme işlemi (You, 2014)

Song'un Çin Banknotları için yaptığı sınıflandırma çalışmasında ise çeşitli görüntü işleme yöntemleri kullanılmıştır. Bunlar sırasıyla arka plan filtreleme,

grileştirme, ikili resme dönüştürme, keskinleştirme, gürültü azaltma, bölütleme gibi yöntemlerdir.



Şekil 2.7 Ön işleme adımları (Song, 2009)

Aşağıda Çin banknotu üzerinde yapılan işlemler adım adım gösterilmektedir. Sağ üstteki siyah yazı ve numara banknottaki kilit bilginin bulunduğu noktadır.



Şekil 2.8 Çin banknotu (Song, 2009)

Her banknotun sağ üstünde seri numarası bulunmaktadır. Bu bölüm yaklaşık olarak genişlikte 1/4, yükseklikte ise 1/6'lık bölüm olarak kırpılmaktadır.



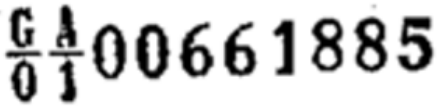
Şekil 2.9 Banknottaki ilgi alanı (Song, 2009)

Kilit bilgi siyah renkli yazıdır, bu yazunun ortalama eşik değerine karşılık piksel değeri diğer renklerden daha düşüktür. Şekil 2.9'daki tüm piksel değerleri eşik değerinden büyükse filtrelenmektedir. Daha sonra piksel değerinin kırmızı, mavi ve yeşil bileşenleri beyaz renge çevrilecektir. Bu şekilde kırmızı ızgara ve mühür Şekil 2.10'da gösterildiği gibi resimden filtrelenecektir.



Şekil 2.10 Grileştirilmiş resim (Song, 2009)

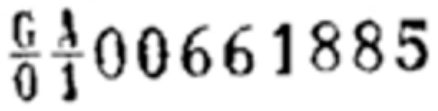
Göründüğü üzere kırmızı mühür ve kırmızı desenler filtrelenmiştir. Daha sonra resmi ikili resme dönüştürmek için renk kanalları üzerinde biçimsel açma ve kapama yaparak, uyarlanmış (adaptive) eşik değerine göre pikseller işlenmiştir. (Solymar, 2011).



Şekil 2.11 Siyah beyaza dönüştürme (Song, 2009)

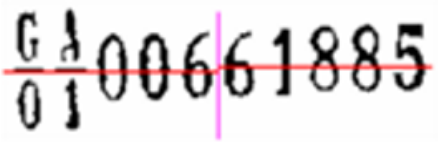
Resmi çevreleyen sınırı bulmak ve yazıların daha belirgin hale gelmesi için gradyan keskinleştirme algoritması kullanılmaktadır.

Her bir piksel bir eşik değeri ile karşılaştırılarak, bu değerden küçük olanlar için kendi değerleri korunurken, büyükler ise beyazla değiştirilmektedir. Bu yöntem ile şekil 2.12'de göründüğü gibi resimdeki gürültü kalıntıları filtrelenip ve noktaların çoğu temizlenebilmektedir.



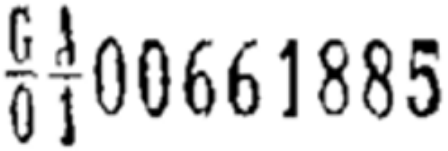
Şekil 2.12 Gürültü filtrelemesi ve resim keskinleştirme (Song, 2009)

Banknotun yatay olarak taranmadığı durumlarda bilgisayarın tanımlama yapabilmesi zorlaşmaktadır. Banknotun uzun kenarı yatay olacak şekilde hizalanmalıdır. Resimdeki anahtar bilgi toplam 10 karakterden oluşmaktadır ve her bir karakter aynı yüksekliğe ve genişliğe sahiptir. Yatay olarak piksel sayısının belirlenmesi ile siyah anahtar bilgiyi tam ortadan bölecek şekilde tek bir kolon çizgisi (pembe çizgi) belirlenebilmektedir. Sağ taraftaki her bir karakterin piksel sayısı ayrı ayrı hesaplanır, her bir karakterin ortalama piksel sayısının bir hat üzerinde bulunduğu düşünülerek Şekil 2.13'deki gibi bir çizgi ortaya çıkartılmaktadır.



Şekil 2.13 Oryantasyon çizgisinin bulunması (Song, 2009)

Resim adım adım döndürülerek, her adımda iki taraftaki çizgilerin eğimleri hesaplanmaktadır. İki tarafında eğimi aynı olduğu anda resim artık istenen yatay koşullara geldiği kabul edilmektedir (Song, 2009).



Şekil 2.14 Yatay pozisyonu ayarlanmış resim (Song, 2009)

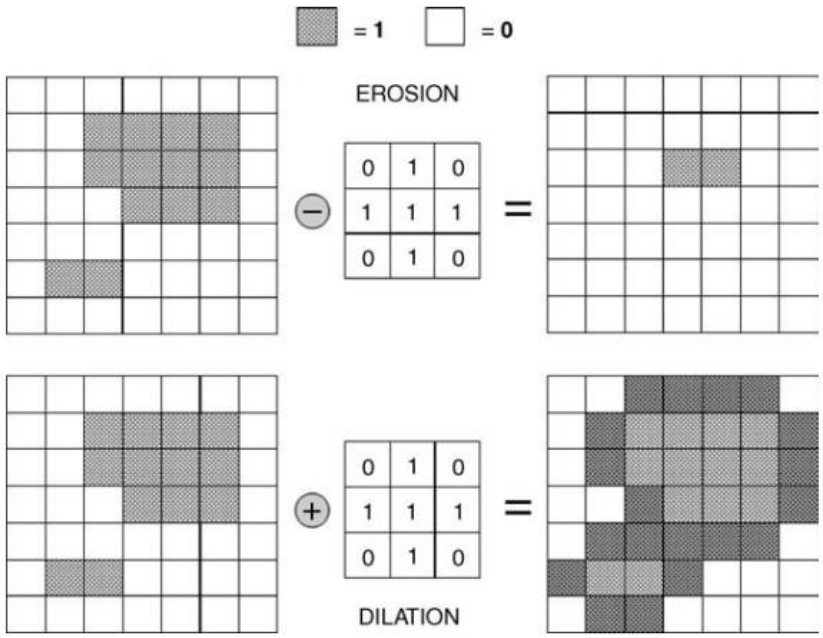
2.1.3. Morfolojik işlemler

Resimlerin siyah beyaza dönüştürüldüğünde resimdeki şekillerde gürültüden kaynaklı bazı bozulmalar olabilmektedir. Morfolojik görüntü işleme, görüntü içindeki formlarda ve yapılarda oluşan bu gibi bozuklukları gidermeyi hedeflemektedir.

Morfolojik görüntü işleme, resim içindeki şekil veya biçimsel özelliklere dayanan lineer olmayan işlemlerden oluşmaktadır. Bu morfolojik işlemler, piksellerin değerlerine göre değil, sıralanmalarına göre çalışır. Piksel değerlerinin bir önemi olmadığı için özellikle siyah beyaz resimleri işlemek için uygundur. En önemli morfolojik operatörler; aşınma (erosion) ve yayma (dilation) operatörleridir. Diğer bütün morfolojik işlemler bu iki operatörü kullanarak tanımlanabilmektedir. (Solomon, 2011)

Şekil 2.15'de aşınma ve yayma işlemleri gösterilmektedir. 1 ve 0'lardan oluşan çekirdek matris tüm resim üzerinde gezdirilerek aşınma işleminde çıkarma, yayma

işleminde ise toplama yapılmaktadır. Aşınma işleminde şekil küçülürken, yayma işleminde ise genişlemektedir.



Şekil 2.15 Aşınma ve yayma işlemleri (Solomon, 2011)

Aşınma ve yayma işlemleri farklı uygulanarak morfolojik açma ve kapama işlemleri yapılmaktadır. Morfolojik açma işlemi, aşınma ve yayma işlemlerinin art arda uygulanmasıdır. Bu işlem sayesinde belirli alandan küçük piksel grupları resimden kaldırılabilmektedir. Morfolojik kapama işleminde ise önce yayma sonra aşınma işlemi uygulanmaktadır. Bu işlem ile görüntüdeki nesne içerisindeki boşluklar giderilebilmektedir.

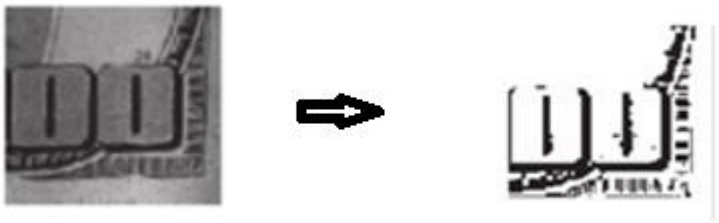
2.1.4. Bölütleme (Segmentasyon)

Bölütleme, görüntüyü birden fazla segmente bölerek nesneleri ve arka planı ayrıştırma işlemidir. Bu işlem genellikle nesneleri ve sınırları tespit etmek için kullanılmaktadır. Bölütleme işleminde genellikle resimdeki renk, doku ve görüntüler arasındaki nesnelerin hareketi gibi faktörler dikkate alınmaktadır. Bölütleme işleminin en basit metodu eşik değeri belirleme metodudur. Bu metotta belirli bir eşik değeri belirlenmekte ve pikseller bu değerle karşılaştırılarak siyah veya beyaza dönüştürülmektedir. Burada eşik değeri manuel veya otomatik belirlenebilir. Eşik değerini otomatik belirlemek için kullanılan popüler yöntemlerden birisi de Otsu metodudur.

Otsu metodunda, algoritma resmin piksellerinin çift durumlu histogram ile iki gruba (ön plan ve arka plan pikselleri) ayrılabileceğini varsaymaktadır. Otsu algoritması bu iki gruba ayırma işlemi için sınıflar arası varyans minimum, sınıf içi varyans ise

maksimum olacak şekilde en iyi eşik değerini hesaplamaktadır. (Otsu, 1979) Bu metodun birden fazla seviyede eşik değeri ayarlamak için geliştirilen versiyonu ise çoklu Otsu yöntemi olarak adlandırılmaktadır.

Otsu metodu para tanıma çalışmalarında sıklıkla kullanılmaktadır. Örnek olarak, 2014 yılında yapılan bir çalışmada banknot üzerindeki desenleri daha belirgin hale getirmek için Otsu metodu uygulanmaktadır (You, 2014).



Şekil 2.16 Otsu algoritması uygulaması (You, 2014)

2.1.5. Özellik tespiti ve özellik çıkarımı

Resimdeki ilgi alanlarının bulunmasına özellik tespiti denir. Bu yöntemde algoritmalar resim hakkındaki bilgileri hesaplar ve buna bağlı olarak her bir piksel için o pikselin belirli bir özelliğe ait olup olmadığına karar verir. Ortaya çıkartılan özellikler genellikle ayrık noktalar, kavisler veya birbirine bağlı bölgeler olarak elde edilmektedir. Resim özellikleri probleme ve uygulamaya göre değişebilmektedir. Kenarlar, köşeler ve bölgeler resim özelliklerine örnek olarak verilebilmektedir.

2.1.5.1 Kenar tespiti

Kenar tespiti görüntü işlemenin en önemli alanlarından biridir. Bir nesnenin tüm kenarları bulunarak sınırları belirlenebilirse nesne resimde kolayca ayrıştırılabilir. Çoğu kenar dedektörü (Prewitt ve Sobel çekirdekleri gibi) gradyan differansiyel filtreleri kullanır. Gradyan resimdeki rengin veya yoğunluğun yönünü gösterir. Bu filtreler kenarların yerini tam olarak bulamasalar da nerede olabileceklerine dair işaretler gösterirler.

Banknot tanıma çalışmalarında sıklıkla kullanılan kenar tespiti yöntemleri Canny kenar dedektörü, Sobel ve Prewitt işleçleridir.

Canny kenar dedektörü 1986 yılında John canny tarafından geliştirilmiştir. (Canny, 1986). Canny kenar tespit algoritmasının adımları basitçe şu şekildedir:

1. Resimdeki gürültü Gauss filtresiyle azaltılır.

2. Kenarın ağırlığı bulunur. Kenar ağırlığı belirlenirken Sobel işleyicileri ve resmin gradyanı kullanılır.

3. Kenarın yönü belirlenir.

4. Kenar yönü sayısallaştırılır.

Canny kenar tespiti ile ortaya çıkan kenar haritası aşağıdaki gibidir.



Şekil 2.17 Canny kenar dedektörü ile kenar tespiti

Prewitt ve Sobel işleçleri 3x3'lük yatay ve düşey yönlü maskelerden oluşmaktadır. Prewit işleci resimdeki her bir pikseldeki yoğunluk için gradyan hesaplar bu şekilde açıktan koyuya doğru giden yönü ve o yöndeki değişiklik oranını tespit edebilmektedir. Bu şekilde resmin o noktasındaki değişimin keskin bir şekilde olup olmadığı dolayısıyla resmin bu bölümünün kenara ait olup olmadığı tespit edilir. Prewitt ile Sobel işleçleri arasındaki temel fark; Sobel filtresinde, differansiyon ile Gauss ortalamasının birbirinden farklı yönlerde uygulanmasıdır. Bu işlem sayesinde kenar bölgesi pürüzsüzleştirilir ve filtrenin çalışmasını bozabilecek gürültülü veya izole piksellerin oluşumu engellenir (Solomon, 2011).

2.1.5.2 Özellik tanımlayıcılar

Özellik tanımlayıcı algoritmalar görüntünün anahtar noktalarını ortaya çıkartmak için kullanılmaktadır. Bu özelliklerin tespit edilmesi için bazı tanımlayıcı algoritmalar kullanılmaktadır. Banknot tanıma için çalışmalarında sıklıkla kullanılan özellik tespit edici algoritmalar; ölçekten bağımsız öznitelik dönüşümü (scale invariant feature transform: SIFT) (Lowe, 2004), hızlandırılmış gürbüz öznitelikler (speeded-up robust features: SURF) (Herbert, 2008) ve yönlü gradyan histogramı (histogram of oriented gradient: HOG) algoritmalarıdır.

D. Lowe yazdığı makalede algoritmayı şu şekilde tanımlamaktadır: SIFT yöntemi; nesnenin ve sahnenin farklı görünümleri arasında güvenilir bir eşleştirme sağlayabilen ayrık ve değişmeyen özelliklerin çıkarımını yapmaktadır. (Lowe, 2004).

Aşağıdaki adımlar ölçekten bağımsız öznitelik dönüşümü yöntemi ile özellik çıkarımı için kullanılan ana adımlardır.

1. Ölçek uzaydaki uç noktaların bulunması: Hesaplamanın ilk adımı olarak, tüm ölçeklerde ve resmin tüm noktalarında arama yapılmaktadır. Bu adımdaki hesaplama, ölçekten ve açıdan bağımsız potansiyel ilgi noktalarının tespit edilmesi için Gauss fonksiyonu kullanılarak gerçekleştirilmektedir.
2. Kilit noktaların konumlarının belirlenmesi: Her bir potansiyel konumda, konumu ve ölçeği belirlemek için detaylı bir model uygulanmaktadır.
3. Yön atama: Lokal resim gradyan yönlerine bağlı olarak her bir kilit nokta için bir veya birden fazla yön tanımlanmaktadır. Bu aşamadan sonraki tüm özellik işlemleri, özelliğe atanan yöne, ölçeğe ve yere göre dönüştürülen resim verisi üzerinde gerçekleştirilir bu sayede bu dönüşümlerden bağımsızlık sağlanmaktadır.
4. Kilit nokta tanımlayıcının bulunması: Yerel resim gradyanları her bir kilit noktanın etrafında belirlenen ölçekte hesaplanmaktadır.

Bu yaklaşım, resim verisini yerel özelliklere göre, ölçekten bağımsız koordinatlara dönüştürdüğü için ölçekten bağımsız öznitelik dönüşümü olarak adlandırılmaktadır. Ayrıca bu algoritmayla tespit edilen özellikler yüksek oranda özgündür, bu sayede bu özellikler büyük bir özellik veri tabanında kolaylıkla eşleştirilebilir (Lowe, 2004).

SURF yöntemi ise hızlandırılmış gürbüz öznitelikler anlamına gelen, özellik tanımlayıcı bir algoritmadır (Herbert, 2008). Bu algoritma resimde bazı biricik kilit noktaları ve tanımlayıcıların belirlenmesini sağlamaktadır. Bu tanımlayıcılar kullanılarak resimler arası benzerlik ölçülebilmektedir. SURF orta düzey resim gösterimi denilen integral resimleri kullanır, bu resimler girdi resimlerden hesaplanarak herhangi bir dikdörtgen alanda hızlandırılmış hesaplar yapılabilmesini sağlamaktadır. Bu hesaplamalar sıfır noktasından başlayarak resmin sonuna kadar x, y koordinatlarındaki piksel değerlerinin toplanmasıyla oluşturulmaktadır. Bu şekilde hesaplama zamanını boyuttan bağımsız hale getirir ve özellikle büyük resimleri işlerken kullanımda fayda sağlamaktadır. Bu yöntem açıdan bağımsız olduğu için nesneler döndürülmüşken dahi tespit edebilmektedir. (Herbert, 2008).

SURF algoritması ile özelliklerin tespiti banknot sınıflandırma çalışmalarında sıklıkla kullanılmaktadır. Hint banknotları üzerinde yapılan bir çalışmada, banknot ortasındaki sayı kısmı kırpılarak, SURF tanımlayıcı ile özellikleri tespit edilmektedir. Bu çalışmada görüldüğü üzere SURF yöntemi resimlerdeki ilgi alanlarını açıdan ve

ölçekten bağımsız olarak tespit edebilmekte ve arka plan gürültülerini göz ardı edebilmektedir (Kamal, 2015).

Etiyopya banknotları üzerinde yapılan başka bir çalışma da ise SURF algoritması belirli bir alanda değil resmin tümünde uygulanarak özellikleri tespit etmesi sağlanmaktadır. Bulunan özelliklerden en yoğunları seçilerek, banknot kategorilerinin özellik tanımlayıcılarıyla karşılaştırılmaktadır. Tespit edilen özellikle aşağıdaki gibidir (Zeggeye, 2016).



Şekil 2.18 SURF noktaları (Zeggeye, 2016)

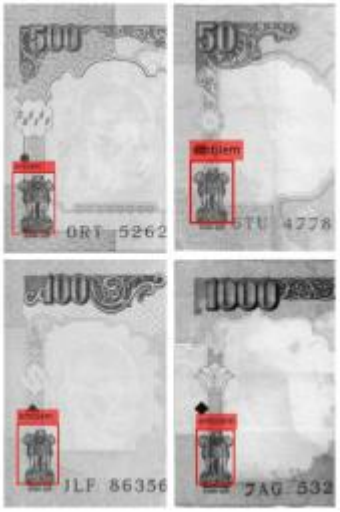
Özelliklerin tespit edilmesi için kullanılan diğer bir yöntem ise HOG algoritmasıdır. Algoritmanın yaygın kullanılmaya başlanmasında, Dalal ve Triggs'in 2005 yılında yaptığı çalışmanın önemi büyüktür. Bu çalışmada araştırmacılar yönlü gradyan histogramı (HOG) tanımlayıcılarını kullanarak popüler bir yaya tanıma algoritması geliştirdiler (Dalal, 2005).

HOG algoritmasında amaç imgeyi yerel histogram grubu olarak tanımlamaktır. Bu histogramlar, imgenin yerel bir bölgesindeki gradyanların yön sayılarını içermektedir. HOG tanımlayıcıları belirlenmesindeki basamaklar şu şekilde sıralanabilir:

1. İmgenin gradyanın alınması
2. Belirlenen lokasyonlar için yön histogramlarının oluşturulması
3. Oluşturulan lokasyon grupları içindeki histogramların normalize edilmesi (Karakaya, 2009).

Kamal'ın Hint banknotları üzerinde yaptığı çalışmada, HOG yönteminden faydalanılmaktadır. Bu banknotların ayırt edici özelliklerinden birisi de sol altta bulunan şekildir. Bu çalışmada banknot resim öncelikle gri ölçeğe çevrilmekte daha sonra resim

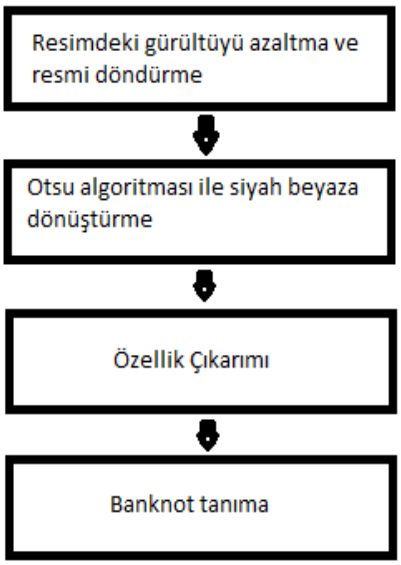
üzerinde kayan bir pencere ile HOG özellikleri tespit edilmeye çalışmaktadır. Resim 8x8 boyutlarında hücrelere bölünerek, her bir hücre için histogram yönleri oluşturulmaktadır. HOG tanımlayıcı, istenen nesnenin şeklini, gradyan yoğunluk dağılımlarına veya kenar yönlerine göre tanımlamaktadır. Tanımlayıcı bu yönlü gradyanların histogramlarının birleşimidir. Dedektörün incelediği bölge aranan nesneyi içeriyorsa nesne işaretlenmektedir (Kamal, 2015).



Şekil 2.19 Hint banknotlarında amblem tespiti (Kamal, 2015)

2.1.5.3 Özellik çıkarımı

Özellik çıkarımı ise görüntünün sınıflandırılabilmesi için kullanılan ilgi noktalarının tespit edilmesi ve bu noktaların özelliklerinin elde edilmesidir. İncelenen çalışmalarda görüldüğü üzere öncelikle resim ön işlemeye tabii tutulmakta daha sonra özellik çıkarımı yapılmaktadır. Örnek bir banknot sınıflandırma sürecinde özellik çıkarımının kullanımı Şekil 2.20'deki gibidir (You, 2014).



Şekil 2.20 Resim işleme aşamaları

Hint banknotlarıyla yapılan bir çalışmada ise merkezdeki sayı, tanımlayıcı işaret, ulusal amblem ve renkli bant gibi ayırt edici özellikler belirlenmektedir. Bu özellikleri tespit etmek için HOG algoritması gibi özellik tespit edici algoritmalar kullanılmaktadır. Banknot üzerindeki ayırt edici işaretler Şekil 2.21'de gösterilmektedir.



Şekil 2.21 Hint banknotlarındaki ayırt edici özellikler (Kamal, 2015)

Bir başka çalışmada ise Macar banknotlarını tanımak için kupüre özel alanlara odaklanılmaktadır. Bu banknotlardaki özel alan ise dokunsal işaretlerin bulunduğu bölgedir. Macar banknotlarında, banknotun arka sağ üst bölümünde farklı semboller bulunmaktadır (Solymar, 2011).

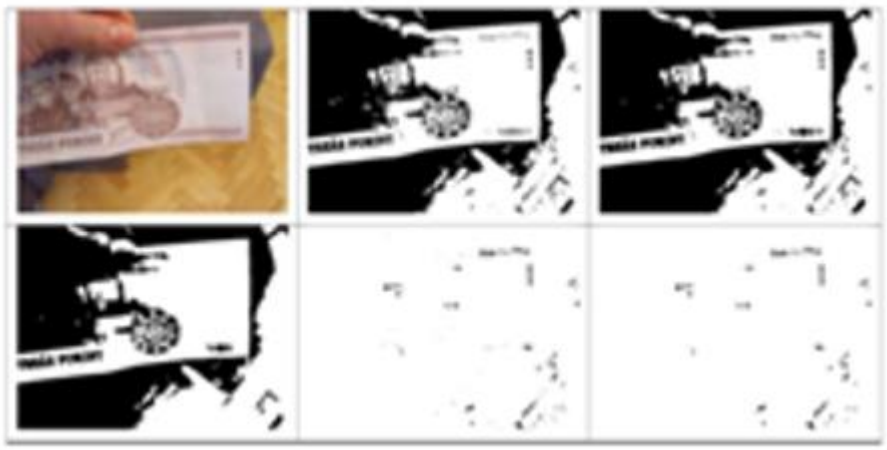


Şekil 2.22 Macar banknotlarındaki dokunsal semboller (Solymar, 2011)

Bu kabartmalı işaretler görme engelliler için tasarlanmaktadır. Bu alanlar birkaç hafta sonra dokunsal özelliğini kaybetse de görsel olarak tespit edilebilmektedirler.

Banknotun sağ tarafı özellikle üst bölgesi açık bir arka plana ve düşük yoğunluğa sahiptir bu yüzden ayırt etmesi kolaydır. Banknotu tanımak için aşağıdaki biçimsel adımlar gerçekleştirilmektedir:

1. Siyah beyaza çevirme.
2. 4x4 filtre kullanarak biçimsel kapama.
3. 350 piksellik alandan küçük nesnelerin kaldırılması.
4. 2. ve 3. Adımın XOR'unun alınması.
5. 10 pikselden küçük alanı olan nesnelerin kaldırılması.



Şekil 2.23 Dokunsal alanın tespit edilmesi (Solymar, 2011)

Birbirine benzer lekeler 3'lü, 4'lü be 5'li olarak gruplanmaktadır. Bu gruplama kütle merkezleri ve benzerlik oranlarına bakılarak yapılmaktadır. İdeal olarak, ilgi alanına karşılık gelen gruptaki alanlar, aynı çizgi üzerinde bulunan aynı alana ve aynı mesafede kütle merkezine sahiptirler. Bir sonraki adımda bu özelliklere göre minimum sapmaya sahip olan grup seçilir. Karar verildikten sonra, desenler sağ üst köşede tespit edilir. (Solymar, 2011)

2.1.6. Nesne Tanıma

Görüntülerdeki belirli nesneleri ve özellikleri tespit etmek için kullanılan metotlar nesne tanıma alanının çalışma alanını oluşturmaktadır. Nesne tanıma yöntemleri görüntü işleme alanında sıklıkla kullanılmaktadır. İnsanların kolaylıkla yaptığı nesne tanıma ve sınıflandırma işlemlerini bilgisayarlara verimli bir şekilde yaptırabilmek için çalışmalar hala sürmektedir. Nesne tanıma; görünüm tabanlı metotlar ve özellik tabanlı metotlar olarak iki kategori altında incelenmektedir. Görünüm tabanlı metotlarda şablon resimler kullanılır, bu resimler orijinal resimle karşılaştırılarak

benzerlik oranları ölçülerek bu şekilde eşleştirme yapılmaktadır. Özellik tabanlı metotlarda ise görüntünün ve nesnenin özelliklerinden yararlanılarak nesne tespit edilmeye çalışılmaktadır.

Banknot sınıflandırma çalışmaları incelendiğinde, bazı çalışmaların şablon eşleştirme ve özellik çıkarımı yöntemlerinden faydalandığı görülmektedir.

Macar banknotlarıyla yapılan bir çalışmada sayı ve portre bölümünü tespit edilerek banknotlar sınıflandırılmaktadır. Banknotların desenlerini çıkartmak ve diğer nesnelerin görüntüsünü minimize etmek için ilk olarak resmin kenarlarına bağlı olan daha büyük nesneler kaldırılmaktadır. Aynı adımlar renkleri ters çevrilmiş resim üzerinde tekrar uygulanarak, farklı ışıklandırma koşullarında, ön desenler hızlı bir şekilde tespit edilmektedir.



Şekil 2.24 Girdi, tespit edilen desen ve ayrıştırılan portreler (Solymar, 2011)

Bu aşamada en dıştaki nesneler kaldırılarak sadece banknotun desenleri kalmaktadır. İleri analiz ve özellik seçimi için tespit edilen nesneler biçimsel varsayımlar kullanılarak ön sınıflandırmaya tabi tutulmaktadır. (Solymar, 2011)

Başka bir çalışmada ise bir şablon eşleştirme yöntemi olan Tanimoto yöntemi kullanılmaktadır. (Tanimoto, 1958). Bu yöntemde belirlenen şablonlar ile karşılaştırma yapılarak eşleştirme yapılmaktadır. Banknot üzerinde tespit edilen kilit bilgi üzerindeki her bir karakter, 26 harf ve rakamlardan oluşan şablon kümesi ile eşleştirilmektedir. Tanimoto karşılaştırma teorisine göre, ilk olarak siyah beyaz şablon kümesi hazırlanır. Daha sonra her bir çerçeve ile şablon kümesinden karakterler arasındaki benzerlik oranı hesaplanmaktadır (Butina, 1999).

Tanimoto formülü aşağıdaki gibidir:

$$S(x_i + x_j) = \frac{x_i^T x_j}{x_i^T x_i + x_j^T x_i - x_i^T x_j} \quad (2.4)$$

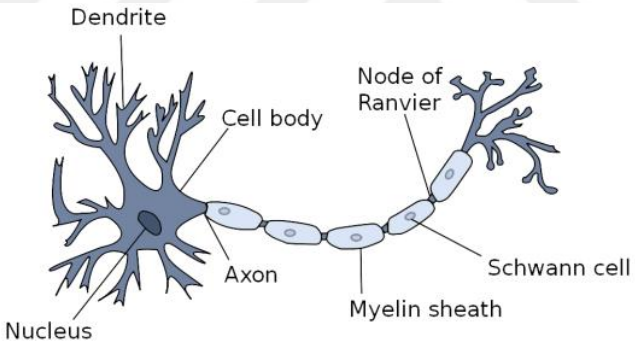
Bu formülde x_i şablonun piksellerini, x_j karşılaştırılmak istenen ikili resmin piksellerini temsil etmektedir. S değeri 0-1 aralığındadır, maksimum değeri şablondaki harfe karşılık gelmektedir. Bu formül kullanılarak görüntü üzerindeki harflerin ve sayıların karşılıkları bulunmaktadır.

2.2. Yapay Sinir Ağları

Yapay sinir ağları (YSA), beyindeki sinir ağlarından ve nöronlardan ilham alınarak oluşturulan bir öğrenme sistemidir. Yapay sinir ağları sayesinde öğrenebilen modeller oluşturmak mümkün hale gelmektedir. Yapay sinir ağları son zamanlarda popüler olmasına karşın temelleri 1940'lı yıllarda oluşturulmuştur.

2.2.1. Yapay sinir ağlarının temelleri

Öğrenme sürecinin temelini, beyindeki sinir hücreleri (nöronlar) arasındaki iletişim oluşturmaktadır. Öğrenme sürecinde beyindeki nöronlar; şekil değiştirmekte, birbirleriyle yeni bağlantılar kurmakta ve aralarındaki iletişimde kullandıkları kimyasalların yoğunluklarını ayarlamaktadırlar. Bir nöron aşağıdaki bölümlerden oluşmaktadır.



Şekil 2.25 Biyolojik nöron (Kriesel, 2007)

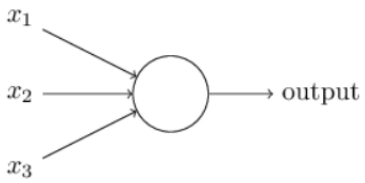
Şekilde görüldüğü gibi biyolojik bir nöron; hücre gövdesi (cell body), akson (axon), dentrit ve çekirdek (nukleus) gibi bölümlerden oluşmaktadır. Aksonların dışı yalıtımı sağlayan miyelin kılıfla kaplıdır, bu kılıfı Schwann hücreleri oluşturur. Aksonlar veriyi ileten kısımlardır. Her bir nöron çok sayıda nöronla bağlantılıdır. Nöronlar arası iletişim sinaps bağlantılarıyla sağlanmaktadır.

Sinirbilimindeki bazı bulgular zihinsel aktivitenin beyindeki nöronlardan oluşan sinir ağlarındaki elektrokimyasal aktiviteden kaynaklandığını göstermektedir. Beyindeki nöron, girdisi ve çıktısı olan bir anahtar gibi çalışmaktadır. Nöron, diğer nöronlar tarafından yeterince uyarıldığında aktif hale gelir ve çıktı olarak diğer nöronlara sinyal gönderir.

Sinirbilimindeki bu sinirsel aktivite sürecinden ilham alan bilgisayar bilimciler, öğrenme sürecini bilgisayarlarda modelleyebilmek için yapay sinir ağları geliştirdiler.

1943 yılında bir nöronun matematiksel modelini oluşturan McCulloch ve Pitts bu alana öncülük eden isimlerdendir. Bu yapay nöron, basitçe girdilerin lineer kombinasyonu belirli bir değeri geçtiğinde tetiklenmektedir.

Bu çalışmadan etkilenen Rosenblatt 1950'lerde algılayıcı (perceptron) adı verilen yapay nöronu geliştirdi. Algılayıcı, girdi olarak birkaç ikili değer alarak tek bir çıktı üretmektedir. Örnek bir algılayıcı aşağıdaki gibidir.



Şekil 2.26 Perceptron (Nielsen, 2015)

Bu örnekte algılayıcı üç girdi değeri almaktadır, bu girdiler daha fazla veya daha az olabilmektedir. Rosenblatt çıktı değerini hesaplamak için ağırlıkları (w1, w2, ...) kullanmaktadır. Bu ağırlıklar her bir girdinin önemini ifade etmektedir. Nöronun çıktısı aşağıdaki formülde gösterildiği gibi girdilerin ağırlıklı toplamı (aktivasyon değeri) belirli bir eşik değerinden büyükse 1, küçükse 0 olmaktadır (Nielsen, 2015).

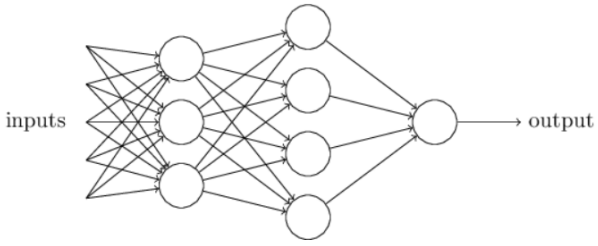
$$\text{çıktı} = \begin{cases} \sum_j w_j x_j \leq \text{eşik değeri ise } 0 \\ \sum_j w_j x_j > \text{eşik değeri ise } 1 \end{cases} \tag{2.5}$$

Denklem 1.4 ile ifade edilen diğer bir gösterim şeklinde ise eşik değeri yerine bias kavramı kullanılmaktadır. Bias değeri, tüm girdiler sıfır olduğunda algılayıcı çıktısının ne olacağını belirlemektedir.

$$\text{çıktı} = \begin{cases} w.x + b \leq 0 \text{ ise } 0 \\ w.x + b > 0 \text{ ise } 1 \end{cases} \tag{2.6}$$

Bias, algılayıcının ne kadar kolaylıkla 1 değerini üretebileceğini göstermektedir. Bias ne kadar büyük olursa algılayıcının 1 çıktısı üretmesi o kadar kolaydır.

Algılayıcı bir karar verme mekanizmasıdır. Girdi olarak verilen parametreleri ağırlıklandırarak bir sonuca ulaşır. Daha hassas kararlar verebilmek için aşağıdaki gibi karmaşık bir algılayıcı ağı kullanılmaktadır. Bu ağa çok katmanlı algılayıcı da (multi-layer perceptron) denmektedir.



Şekil 2.27 Perceptron ağı (Nielsen, 2015)

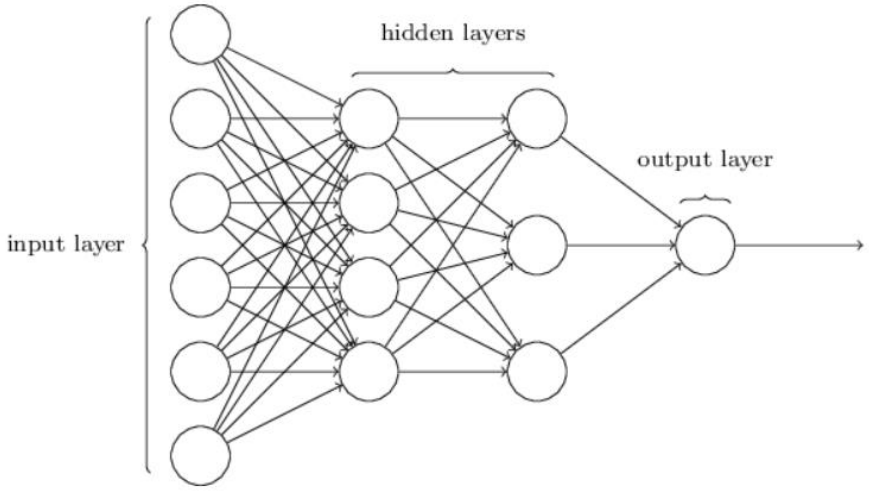
Her bir kolon, sinir ağının bir katmanıdır. İlk katmandaki üç algılayıcı ağırlıkları ölçerek basit kararlar verebilirken ikinci katmandakiler daha soyut ve karmaşık kararlar verebilmektedir.

Bu şekildeki bir yapay sinir ağı basit sınıflandırma problemlerinde kullanılabilmektedir. Algılayıcılardan oluşan ağlar eğitilirken ağırlıklarda yapılan ufak değişiklikler algılayıcı çıktılarını 0'dan 1'e veya tam tersine dönüştürebilmektedir. Bu keskin sonuç değişiklikleri tüm ağın yapısını değiştirerek hassas öğrenmeyi engellemektedir. Bu problemin üstesinden gelebilmek için sigmoid nöronlar geliştirilmiştir. Algılayıcı gibi sigmoid nöronların da girdileri vardır ama bu girdiler algılayıcıda olduğu gibi sadece 0 ve 1 değerleriyle sınırlı değildir. Sigmoid nöronda girdiler 0 ile 1 aralığında herhangi bir değer alabilirler. Algılayıcıda olduğu gibi sigmoid nöronda da her bir girdinin ağırlığı ve bunların bir bias değeri vardır. Sigmoid nöronda çıktı $\sigma(w \cdot x+b)$ formülüyle hesaplanır, σ sigmoid fonksiyonunu temsil etmektedir. Sigmoid fonksiyonu 2.7 no'lu denklemlerde gösterilmektedir. X değerleri girdileri w değerleri ağırlıkları, b ise bias değerini temsil etmektedir (Nielsen, 2015).

$$\sigma(z) \equiv \frac{1}{1+e^{-z}} = \frac{1}{1+\exp(-\sum_j w_j x_j - b)} \tag{2.7}$$

Bu hesaplama sayesinde algılayıcıdan farklı olarak 0 ile 1 arasında çıktı değerleri sunabilmektedir.

Yapay sinir ağı aşağıdaki gibi nöronların birbiriyle bağlanmasıyla oluşmaktadır. İlk katman giriş katmanı, ortadaki katmanlar gizli katmanlar, son katman ise çıkış katmanı olarak adlandırılmaktadır.



Şekil 2.28 Çok katmanlı ağ (Nielsen, 2015)

Yapay sinir ağları veri akışının yönlerine göre ikiye ayrılmaktadır: ileri beslemeli ağlar ve geri beslemeli ağlar. İleri beslemeli ağlarda her bir nörondan çıkan

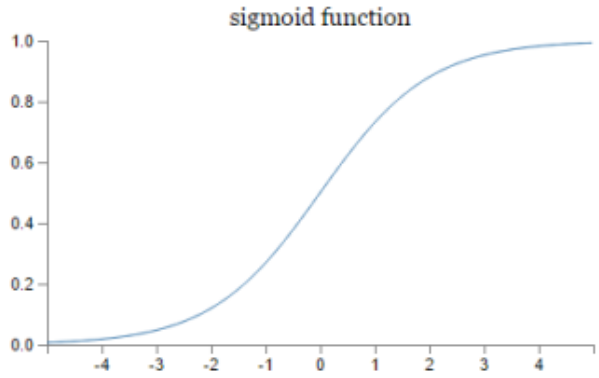
bağlantı bir sonraki katmandaki bir nörona girdi olarak verilerek veri iletimi ileriye doğru yapılmaktadır. Geri beslemeli ağlarda ise nöron çıkışları yine hücrenin kendisine girdi olarak verilebilmektedir, veri iletimi aynı katmanda veya katmanlar arasında yapılabilmektedir.

Yapay sinir ağlarında her bir bağlantının bir ağırlığı mevcuttur. Nöronlar bu bağlantılar sinyalleri alırlar ve bazı fonksiyonlarla dönüştürerek diğer nöronlara iletirler. Nöronlarda yapılan işlemler genel olarak yayılma fonksiyonu, aktivasyon fonksiyonu ve çıktı fonksiyonu olarak gruplandırılmaktadır. Yayılma fonksiyonu bir nörona diğer nöronlardan gelen veriyi girdiye dönüştürür, bu fonksiyon genellikle ağırlıklı toplam fonksiyonudur. Aktivasyon fonksiyonu ise yayılım fonksiyonunun sağladığı girdiyi işleyerek belirli bir eşik değerine göre nöronun vereceği tepkiyi belirler. Aktivasyon fonksiyonu genellikle tüm nöronlar için global olarak tanımlanmaktadır ancak her nöron için aktivasyon eşik değeri farklıdır. Aktivasyon fonksiyonu bazen transfer fonksiyonu olarak da adlandırılmaktadır.

YSA'ların eğitim sürecinde nöronların girdilerini nasıl işleyeceğini ve nasıl tepki vereceklerini belirleyen aktivasyon fonksiyonları kullanılmaktadır. Ayrıca modelin işleyip ürettiği sonuçları gerçek sonuçlarla karşılaştırabilmek için maliyet (cost) fonksiyonu kullanılmaktadır, maliyet fonksiyonu literatürde kayıp (loss) fonksiyonu veya hata (error) fonksiyonu olarak da anılmaktadır. Bu çalışmada maliyet fonksiyonu olarak anılmaktadır. YSA'lar eğitim sürecinde ağırlıklarının ve bias değerlerinin doğru sonuçtan ne kadar saptığını görmek için bu maliyet fonksiyonlarını kullanmaktadır. Ağ eğitilirken bu maliyetin giderek düşürülmesi gerekmektedir. Bu maliyeti düşürme işlemi için de optimizasyon fonksiyonlarından yararlanılmaktadır. YSA öğrenme süreci boyunca ağırlıklarını ve bias değerlerini maliyeti düşürecek şekilde güncelleyerek optimum çözüme ulaşmaya çalışmaktadır.

2.2.2. Aktivasyon fonksiyonları

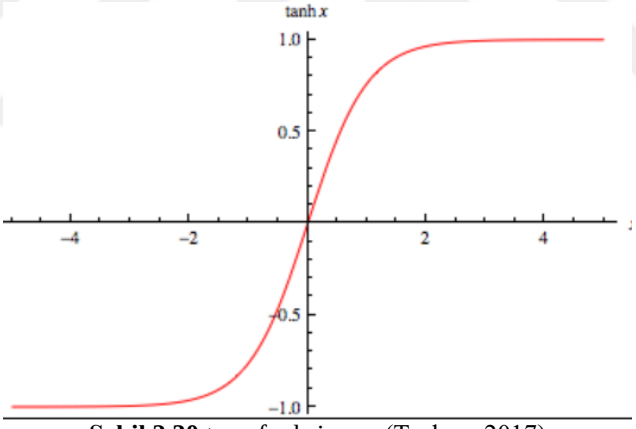
Aktivasyon fonksiyonları nöronların girdileri nasıl işleyeceğini belirlemektedir. Daha önce bahsedilen sigmoid nöronda kullanılan sigmoid fonksiyonu da bir aktivasyon fonksiyonudur. Sigmoid fonksiyonunun kullanılmasının temel amacı 0 ile 1 arasında değerler elde etmektir. Bu nedenle çıktı olarak olasılık tahmin eden modellerde sıklıkla kullanılmaktadır. Fonksiyon grafiği Şekil 2.29'da gösterilmektedir.



Şekil 2.29 Sigmoid fonksiyonu (Taşhan, 2017)

Bu grafikte X eksenindeki değerler nöron girdilerini, Y eksenindeki değerler ise nöron çıktılarını göstermektedir. Grafikte görüldüğü üzere X değeri -2 ile 2 arasındayken Y değerinde belirgin değişiklikler olmaktadır. Bu fonksiyonun dezavantajı ise eğri uçlarında Y değerleri X değerlerine karşı duyarsız kalmaktadır. Bu problem literatürde gradyan yok olması (vanishing gradient) problemi olarak bilinmektedir.

Diğer bir aktivasyon metodu ise hiperbolik tanjant (tanh) fonksiyonudur. Fonksiyon grafiği aşağıdaki gibidir.



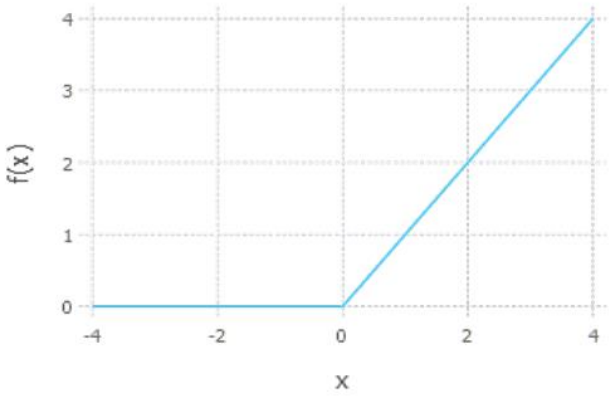
Şekil 2.30 tanx fonksiyonu (Taşhan, 2017)

Tanh fonksiyonu denklemi aşağıdaki gibidir:

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1 \tag{2.8}$$

Tanh fonksiyonu sigmoid fonksiyonuna benzerdir ama tanh fonksiyonunda gradyan daha güçlüdür yani eğrinin eğimi daha keskindir.

En çok kullanılan aktivasyon fonksiyonlarından birisi de Relu (Rectified Linear Unit) fonksiyonudur. Fonksiyonun grafiği Şekil 2.32'de gösterilmektedir.



Şekil 2.31 Relu aktivasyon grafiği (Taşhan, 2017)

Fonksiyonun denklemi ise aşağıdaki gibidir:

$$f(x) = \max(0,x) \tag{2.9}$$

Bu fonksiyon negatif değerlerin sıfırlanmasını sağlamaktadır. Büyük bir sinir ağında tanh veya sigmoid fonksiyonu kullanmak masraflı olabilmektedir, çünkü her bir nöron için hesaplanmaları gerekmektedir. Relu fonksiyonu 0'dan küçük değerler için hesap yapmaya gerek duymadan direk sıfır 0 değeri döndüğü için, nöronlar daha seyrek ve daha verimli bir şekilde aktif hale gelmektedir.

Diğer bir aktivasyon fonksiyonu ise softmax fonksiyonudur. Bu fonksiyon herhangi dağılım üzerindeki kategori bazında olasılığı hesaplamak için kullanılmaktadır. Softmax fonksiyonu, ikili değişken üzerindeki olasılığını hesaplayan sigmoid fonksiyonunun genelleştirilmiş hali olarak düşünülebilir.

Sofmax fonksiyonu genellikle sınıflandırma işlemi yapan YSA'larda çıkış birimi olarak kullanılmaktadır bu sayede n farklı sınıf için olasılık dağılımı gösterilebilmektedir. Bu fonksiyon modelin n seçenekten birini seçmesi gereken durumlarda modelin içinde de nadir olarak kullanılabilmektedir. Aktivasyon fonksiyonları nöronların bireysel olarak nasıl çalışması gerektiğini belirlemektedir.

2.2.3. Maliyet fonksiyonları

Öğrenme sürecinde olası tüm ağırlıklar denenerek en iyi ağırlıklar bulunabilir ama bu yöntem çok uzun sürecektir. En iyi ağırlıkların tahminlenmesi için maliyet fonksiyonları kullanılmaktadır. Bu fonksiyonlara kayıp (loss) fonksiyonları da denmektedir. Maliyet fonksiyonuyla, belirli bir çözüm şeklinin optimum çözüme olan uzaklığı tespit edilebilmektedir.

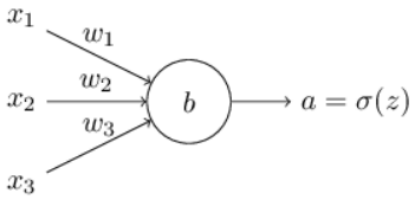
YSA eğitimlerinde karesel maliyet fonksiyonu (quadratic cost) ve çapraz-entropi (cross-entropy) fonksiyonları sıklıkla kullanılmaktadır. Karesel maliyet fonksiyonu aşağıda gösterilmektedir.

$$c(w,b) \equiv \frac{1}{2n}\sum_{x} ||y(x) - a||^2 \quad (2.10)$$

Bu formülde w ağdaki ağırlıkları, b bias değerlerini, n eğitim verisindeki eleman sayısını, a ise x girdi olarak verildiğinde ağın çıktısını göstermektedir. Bu toplama işlemi tüm eğitim verisi üzerinde yapılmaktadır. Bu fonksiyon ortalama karesel hata (mean squared error) olarak da anılmaktadır. Eğitim verisindeki x değerler için y(x) değerleri a çıktısına yakınsadığı zaman bu fonksiyon sıfıra yakınsamaktadır ve ağın iyi öğrendiğini göstermektedir.

Karesel maliyet fonksiyonu kullanan ağlarda bir süre sonra öğrenmeye yavaşlığı problemi ortaya çıkmaktadır. Bu problemi çözmek için karesel maliyet fonksiyonu yerine çapraz entropi fonksiyonu tercih edilmektedir.

Çapraz entropiyi örnek bir nöron üzerinden incelemek daha açıklayıcı olacaktır. Bu örnek nöron Şekil 3.33'de gösterildiği gibi x1, x2, x3 girdilerinden, w1,w2, w3 ağırlıklarından ve b bias değerinden oluşmaktadır.



Şekil 2.32 Yapay nöron (Nielsen, 2015)

Nöronun çıktısı a=σ(z) ile ifade edilmektedir. Z değeri ise girdilerin ağırlıklı toplamını ifade etmektedir. Bu bağlamda çapraz-entropi fonksiyonu şu şekilde tanımlanmaktadır:

$$C = -\frac{1}{n}\sum_{x}[ylna + (1-y)\ln(1-a)] \quad (2.10)$$

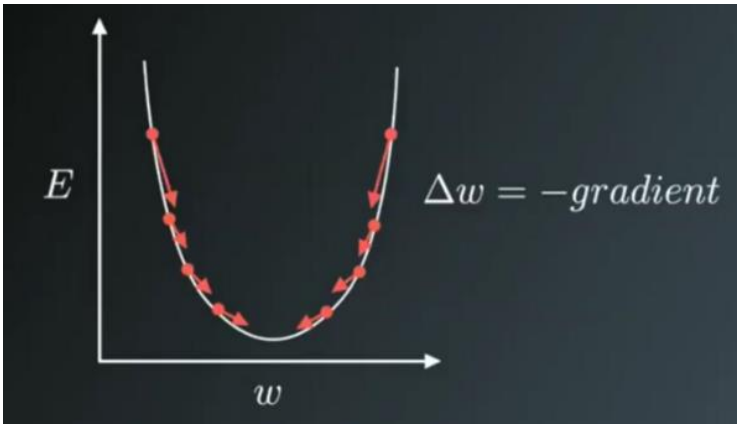
Eğitim verisindeki elemanların toplamı n, tüm eğitim verisi girdilerinin toplamı x, istenilen çıktı da y ile gösterilmektedir. Çapraz-entropi fonksiyonunun sonucu hep pozitiftir. Nöron tüm x girdileri için hedeflenen y değerini hesaplamaya yaklaştıkça çapraz-entropi sıfıra yaklaşmaktadır (Nielsen, 2015).

2.2.4. Optimizasyon fonksiyonları

Maliyet fonksiyonu kullanılarak YSA'da kullanılan belirli bir ağırlık grubunun ne kadar doğru sonuçlar ürettiği ölçülmektedir. Optimizasyon fonksiyonlarının amacı ise maliyet fonksiyonlarıyla ölçülen maliyeti en aza indirgemektir. Bu algoritmalar

YSA'nın eğitim sürecinde, ağın ağırlık (w) ve bias (b) değerlerini güncelleyerek optimum çözüme doğru ilerleme (maliyeti düşürerek) mantığıyla çalışmaktadır. Optimizasyon işlemi YSA'ların eğitim sürecini oluşturan önemli işlemlerden birisidir.

Maliyet fonksiyonun türevi herhangi bir noktadaki fonksiyon teğetinin eğimini vermektedir yani aşağıya doğru giden yönü göstermektedir. Eğim azaltma yönteminde her bir ağırlığa göre maliyetin kısmi türevi, her bir ağırlıktan çıkartılır. Bu yöntemle adım adım en alt noktaya yani en iyi ağırlık değerlerine ulaşılmaktadır. YSA'lar eğim azaltma yöntemini kullanarak ağırlıklarını ve bias değerlerini ayarlayabilmektedir. Maliyet fonksiyonun eğimini hesaplamak için geliştirilen geri-yayılım (back-propagation) algoritması 1970'lerde keşfedilmesine rağmen Rumelhart ve ark. tarafından 1986 yılında yayınlanan makaleyle adını duyurmuştur (Rumelhart, 1986). YSA'ları genellikle geri yayılım algoritmasıyla eğitilmektedir. Bu yöntemde, ilk önce yayılım fonksiyonu kullanılarak girdilerin ilgili ağırlıklarla skaler çarpımları alınmaktadır, daha sonra bu çarpım toplamlarına aktivasyon fonksiyonu uygulanarak girdi sinyalleri çıktıya dönüştürülmektedir. Bu adımdan sonra ağ geri yayılımla hatayı taşımakta ve eğim azaltmayı kullanarak ağırlıkları güncellemektedir. Eğim azaltma yönteminde hata fonksiyonunun (E) ağırlıklara (W) göre eğimi hesaplanarak ağırlıklar maliyet fonksiyonun eğiminin tersi yönünde güncellenmektedir. Bu yöntem için vadiden aşağıya yürüme metaforu da sıklıkla kullanılmaktadır.



Şekil 2.33 Eğimin ters yönünde ağırlık güncelleme işlemi (Walia, 2017)

Yukarıdaki şekilde ağın ağırlıklarına göre hata eğiminin tersi yönünde ağırlıklar güncellenmektedir. Şekilde görüldüğü üzere ağırlıklar çok küçük veya çok büyük olursa yüksek oranda hataya neden olmaktadır. Bu şekilde eğimin tersi yönünde ağırlıklar güncellenerek yerel minimum noktası bulunmaktadır. Burada hata fonksiyonu olarak aşağıdaki formül kullanılmaktadır:

$$E = \frac{1}{2}(y - f(\sum w_i x_i))^2 \tag{2.11}$$

Geri yayılım algoritması maliyet fonksiyonun ağdaki ağırlıklara ve bias değerlerine göre kısmi türevini hesaplamaktadır. Bu şekilde ağırlık ve bias değerleri değiştirildiğinde maliyet fonksiyonun ne kadar hızlı değiştiği gözlemlenebilmektedir. Bu algoritmayla maliyet fonksiyonunun en düşük değerleri bulunmaya çalışılarak ağırlıklar ve bias değerleri ayarlanmaktadır.

Eğim azaltmanın üç yöntemi bulunmaktadır: toplu eğim azaltma (batch gradient descent), rastgele eğim azaltma (stochastic gradient descent) ve parçalı eğim azaltma (mini-batch gradient descent).

Toplu eğim azaltma yönteminde ağın parametreleri aşağıdaki formül kullanılarak güncellenmektedir. Bu yöntem maliyet fonksiyonun parametrelere göre eğimini tüm eğitim veri kümesini kullanarak hesaplamaktadır.

$$\theta=\theta-\eta\cdot\nabla J(\theta) \tag{2.12}$$

Bu formülde; n öğrenme oranını, θ parametreleri, $\nabla J(\theta)$ ise maliyet fonksiyonun eğimini göstermektedir. Öğrenme oranı lokal minimuma doğru gidilerken kaç adım atılması gerektiğini belirlemektedir. Eğim azaltma maliyet fonksiyonunu minimize etmek için kullanılmaktadır (Ruder, 2016). Toplu eğim azaltma işleminde her bir ağırlık güncellemesi için tüm eğitim veri kümesinin eğimi hesaplandığı için bu işlemde eğim azalması yavaş olmaktadır.

Rastgele eğim azaltma yönteminde ise her bir eğitim verisi elemanı için parametre güncellemesi yapılmaktadır. Bu da toplu eğim azaltma yönteminden daha az hesaplama yapılmasına olanak sağlamaktadır. Bu yöntemde kullanılan formül aşağıdaki gibidir.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta;\, x^{(i)};\, y^{(i)}) \tag{2.13}$$

Eğitim verisindeki elemanlar x ile elemanların kategorileri ise y ile gösterilmektedir. Bu sık güncellemeler parametrelerin daha yüksek çeşitliliğe sahip olmasını ve maliyet fonksiyonunun daha fazla dalgalanmasına sebep olmaktadır. Bu dalgalanma da eğim azaltmaların eğrinin dışına çıkarak daha iyi bir yerel minimum noktası bulmasına olanak sağlamaktadır.

Parçalı eğim azaltma yönteminde ise eğitim verisi bölümlere ayırarak her bir bölüm için güncelleme yapılmaktadır. Bu yöntem parametre güncellemelerindeki varyansı azaltarak daha iyi ve daha istikrarlı bir yakınsamaya olanak sağlamaktadır.

Eğim azaltma yöntemini daha da ileriye taşımak için öğrenme oranını ayarlayabilen uyarlanabilir (adaptive) algoritmalar geliştirilmiştir. Bu algoritmaların sık kullanılanları şunlardır: Adagrad, Adadelta, RMSprop, Adam.

Adagrad algoritması parametrelere göre öğrenme oranını ayarlamaktadır. Az ferakanslı parametreler için büyük güncellemeler yaparken sık yüksek frekanslı parametreler için küçük güncellemeler yapmaktadır. Bu yüzden seyrek veriler üzerinde çalışmak için uygundur. Bu yöntem parametre güncellemeleri için aşağıdaki formülü kullanmaktadır (Duchi, 2011).

$$\theta_{t+1,i} = \theta_{t,i} - \frac{n}{\sqrt{G_{t,ii}+\epsilon}} \cdot g_{t,i} \quad (2.14)$$

Bu şekilde her bir zaman aralığında (t) her bir parametre (θ(i)) için geçmişteki o parametre için hesaplanmış olan eğimleri kullanarak farklı bir öğrenme oranı belirlemektedir. Formülde n öğrenme oranını, G önceki eğimlerin kare toplamlarını, g ise amaç fonksiyonunun eğimini göstermektedir. ∈ ise sıfıra bölünmeyi önlemek için eklenmektedir. Adagrad sayesinde öğrenme oranını manuel olarak ayarlama gereksinimi kalmamaktadır. Adagrad algoritmasının dezavantajı ise paydada bulunan eğim karelerinin toplamlarının giderek artmasıdır bu da öğrenme oranının giderek azalmasına neden olmaktadır. Bu yüzden model bir süre sonra öğrenemez hale gelmektedir.

Adadelta ise Adagrad'daki agresif bir şekilde azalan öğrenme oranını azaltmak için geliştirilmiş bir yöntemdir. Bu yöntemde Adagrad yöntemindeki gibi tüm geçmiş eğimleri toplanmamaktadır bunun yerine eğimlerin toplamı tüm geçmiş eğimlerin azalan ortalamasıyla yinelemeli olarak belirlenmektedir. Her bir t anındaki ortalama $E[g^2]_t$, bir önceki ortalamaya (γ oranında) ve o andaki eğime bağlı olarak hesaplanmaktadır (Zeiler, 2012).

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2 \quad (2.15)$$

Bu formüldeki γ momentum terimi olarak da adlandırılmaktadır ve genellikle 0.9 olarak kullanılmaktadır.

Parametre güncellemeleri için ise şu formül kullanılmaktadır.

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t \quad (2.16)$$

Bu formüldeki RMS (root mean square) ortalamaların karekökünün alınmasını ifade etmektedir. $RMS[g]_t$ eğimin hata oranının ortalama karekökünü ifade etmektedir. $RMS[\Delta\theta]_t$ ise parametre güncellemelerinin hatalarının ortalama kareköküdür. Adadelta formülünde öğrenme oranı çıkartıldığı için bu formülde öğrenme oranını belirlemeye gerek yoktur. Bu yöntem sayesinde giderek azalan öğrenme oranı sorunu çözülmektedir.

RMSprop (Root Mean Square Propagation) algoritması ise Geoffrey Hinton'un dersinde önerdiği bir algoritmadır (Tieleman, 2012). Adadelta ve RMSprop birbirinden bağımsız olarak geliştilmiştir ama çözmeyi amaçladıkları sorun aynıdır. İkisi de Adagrad algoritmasının azalan öğrenme oranını çözmeyi amaçlamaktadır. RMSprop algoritmasının ilk güncelleme vektörü Adadelta ile aynıdır. Ortalama formülü ve parametre güncelleme formülü aşağıdaki gibidir.

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \quad (2.17)$$

$$\theta_{t+1} = \theta_t - \frac{n}{\sqrt{E[g^2]_t + \epsilon}} . g_t \quad (2.18)$$

Hinton momentum terimi olarak 0.9, başlangıç öğrenme oranı olarak 0.001 kullanılmasını önermektedir.

Adadelta öğrenme oranını dinamik olarak ayarlarken momentum terimini sabit olarak kullanmaktadır. Momentum terimini de dinamik bir şekilde ayarlamak için Adam (adaptive moment estimation) algoritması geliştirilmiştir (Kingma, 2015). Bu yöntem de her bir parametre güncellemesi için öğrenme oranını ayarlamaktadır. Bu yöntem de Adadelta ve RMSprop algoritmalarındaki gibi önceki eğimlerin karelerinin katlanarak azalan ortalamasını (v_t) kullanmaktadır. Adam algoritması ayrıca önceki eğimlerin katlanarak azalan ortalama (m_t) değerlerini de kullanmaktadır. Bu değer momentum değerine benzer şekilde kullanılmaktadır. Bu değerler aşağıdaki formüllerle hesaplanmaktadır. m_t 1. momenti (ortalama), v_t ise ikinci momenti (varyans) göstermektedir.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.19)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.20)$$

Bu formüldeki β_1 ve β_2 değerleri ise azalma oranlarını göstermektedir. Parametre güncellemeleri için ise şu formülü kullanmaktadır.

$$\theta_{t+1} = \theta_t - \frac{n}{\sqrt{v_t + \epsilon}} . m_t \quad (2.21)$$

Bu formülde θ parametreleri, n öğrenme oranını göstermektedir. Algoritmanın yazarları β_1 için 0.9, β_2 için 0.999, ∈ için de 10^{-8} değerini önermektedir (Ruder, 2016).

2.2.5. Düzenlileştirme (regularization)

Modeller eğitilirken eğitim verisi üzerinde çok iyi sonuçlar verip daha önceden görmediği doğrulama verisi üzerinde iyi sonuçlar veremiyor ise bu modelin aşırı uyum sağladığını (over-fitting) göstermektedir. Aşırı uyum sağlama problemini önlemek için eğitim verisi arttırma yöntemi kullanılabilmektedir, eğer bu mümkün değilse de düzenlileştirme (regularization) yöntemleri kullanılmaktadır.

L2 düzenlileştirme yöntemi, en sık kullanılan düzenlileştirme yaklaşımlarından biridir. Bu yöntem maliyet fonksiyonuna yeni bir parametre (düzenlileştirme terimi) eklemektedir. Düzenlileştirilmiş maliyet fonksiyonu denklemini aşağıdaki gibidir:

$$C = C_0 + \frac{\lambda}{2n}\sum_w w^2 \tag{2.22}$$

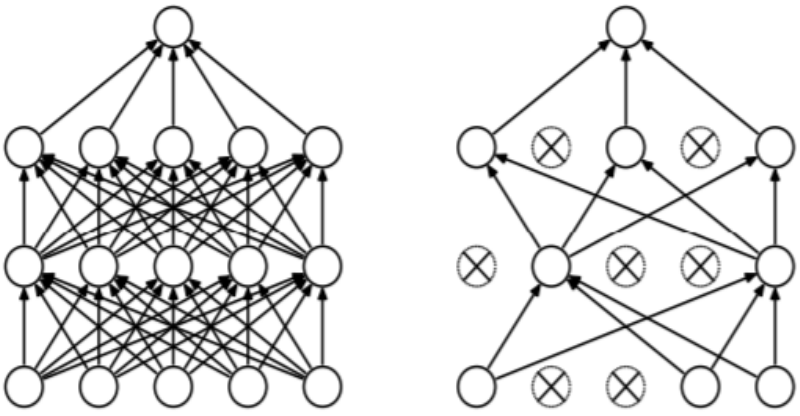
Maliyet fonksiyonuna(C_0), eklenen terim ağdaki ağırlıkların karelerinin toplamıdır. λ düzenlileştirme parametresi, n ise eğitim veri kümesinin eleman sayısıdır. Düzenlileştirme, küçük ağırlık değerleri bulmak ve maliyet fonksiyonunu minimize etmek arasında bir denge kurulmasını sağlamaktadır. İki elemanın hangisinin daha önemli olduğunu λ değeri belirlemektedir. Bu değer küçük olduğunda maliyet fonksiyonu minimize edilmeye çalışır, büyük olduğunda ise küçük ağırlık değerleri tercih edilir.

L1 düzenlileştirme yaklaşımında ise maliyet fonksiyonuna ağırlıkların mutlak değer toplamları eklenmektedir.

$$C = C_0 + \frac{\lambda}{n}\sum_w |w| \tag{2.23}$$

L1 ve L2 düzenlileştirme yaklaşımlarının ikisi de büyük ağırlıkları tercih etmemeye yöneliktir (Nielsen, 2015).

Düzenlileştirme için sık kullanılan diğer bir yöntem ise dropout yöntemidir (Srivastava, 2014). Bu yöntem L1 ve L2 yöntemlerinden farklı olarak maliyet fonksiyonuyla değil ağın yapısıyla ilgilenmektedir. Bu yöntemde ağdaki birimlerden bazıları eğitim sürecinde ağdan çıkartılmaktadır. Rastgele seçilen birimler tüm giriş ve çıkış bağlantılarıyla beraber ağdan çıkartılmaktadır.



Şekil 2.34 Dropout işleminden öncesi ve sonrası (Srivastava, 2014)

Dropout yöntemi eğitim sürecinde ağın tamamı içinde daha küçük ağların örneklenmesi olarak da düşünülebilir. Bu şekilde tek bir ağ değil de, ağ içinde birden fazla ağ eğitilip veriye farklı şekillerde uyum sağlayarak aşırı uyum probleminden kaçınmak mümkündür. Bu yöntem sadece ağ eğitilirken uygulanmaktadır, test sürecinde uygulanmamaktadır.

2.2.5. Öğrenme yöntemleri

YSA, bir veri kümesini girdi olarak alır ve ağırlıklarını ayarlayarak bu veriyi öğrenmeye çalışır. YSA'lar öğrenme yöntemlerine şu sınıflara ayrılmaktadır: gözetimli öğrenme (supervised learning), gözetimsiz öğrenme (unsupervised learning), pekiştirmeli öğrenme (reinforcement learning) ve öğrenim transferi.

Gözetimli öğrenmede eğitim verisindeki sınıflandırmalar hali hazırda belirlidir. YSA girdileri ve onların sınıflarını kullanarak ağırlıklarını ayarlamaktadır. Gözetimli öğrenme yönteminde YSA, girdiler ve çıktılar arasındaki ilişkiyi görmektedir ve bu ilişkiye göre girdilerden çıktıları oluşturacak olan fonksiyonu öğrenmektedir.

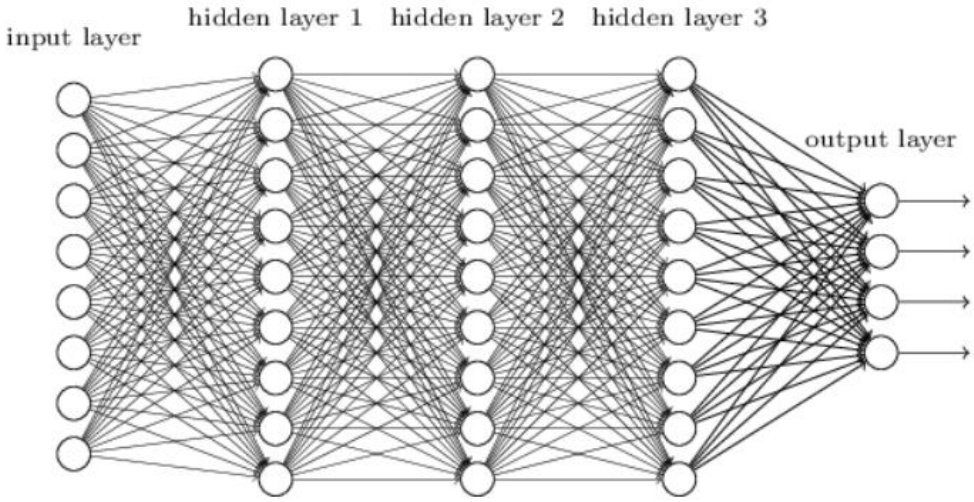
Gözetimsiz öğrenme yönteminde ise eğitim verisi herhangi bir sınıflandırma belirtilmeksizin girdi olarak verilmektedir. YSA, bu veri üzerinde belirli desenler bulup onları sınıflandırmaya çalışmaktadır.

Pekiştirmeli öğrenme yönteminde ise YSA her bir iterasyonda oluşturduğu sonuç hakkında bir geri bildirim alır bu sayede doğru sonuca ne kadar yakınsadığını hesaplayabilmektedir. Bu yöntem bir nevi ödül ve ceza yöntemidir. YSA ağırlıklarını bu geri bildirime göre ayarlayarak maksimum ödüle ulaşmaya çalışmakta bu şekilde öğrenmektedir.

Diğer bir yöntem olan öğrenim transferi ise belirli bir alanda eğitilmiş modelin başka bir işlemde kullanılmak için tekrar eğitilmesi anlamına gelmektedir. Model daha önceden öğrenilen ağırlıkları ve bias değerlerini kullandığı için öğrenme süresi daha kısa sürer ve daha az veriyle eğitmek mümkündür. Bu çalışmada da bu teknikten faydalanılarak, daha önceden farklı veri kümesi üzerinde eğitilmiş modeller yeniden eğitilerek kullanılmaktadır.

2.2.6. Derin sinir ağları

Derin YSA'lar giriş ve çıkış birimlerinin arasındaki gizli birim katmanlarının artırılmasıyla oluşmaktadır. Burada derinlik kavramı YSA'nın birden fazla katmana sahip olması anlamında kullanılmaktadır. Örnek bir derin YSA aşağıdaki gibidir.



Şekil 2.35 Derin YSA (Nielsen, 2015)

Derin katmanlı YSA'larda alt katmanlarda öğrenilen özellikler üst katmanda daha karmaşık fonksiyonlar oluşturulmasına olanak sağlamaktadır. Örneğin bir görüntü işlemek için, ilk katmanda kenarlar tespit edilirken, ikinci katmanda bu kenarlar kullanılarak nesnenin bazı kısımları oluşturulabilmektedir. Daha üst katmanlarda ise bu parçalar birleştirilerek nesne şekilleri ayırt edilebilmektedir (Pascanu, 2014).

Derin sinir ağlarının bir çeşidi olan evrişimsel sinir ağları (convolutional neural networks) görüntü sınıflandırma işlemlerinde büyük bir gelişme sağlamıştır.

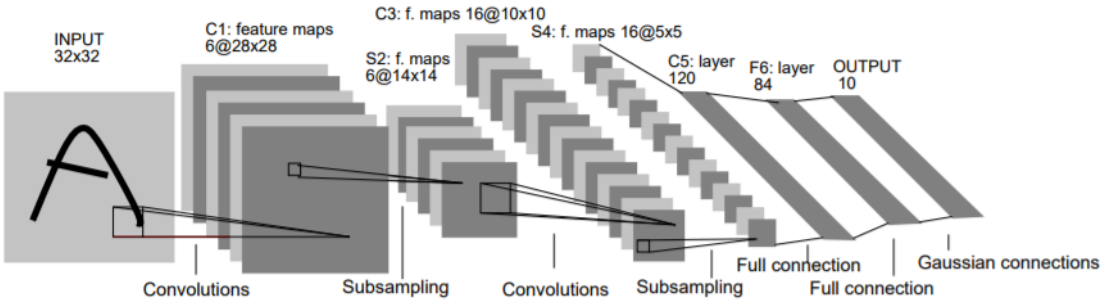
2.2.6.1. Evrişimsel sinir ağları

ESA'lara modern şeklini veren ilk makale 1989 yılındaki yayınlanmıştır (Lecun, 1998). Bu ağların evrişimsel sinir ağı olarak adlandırılmasının sebebi bu yapıdaki ağlarda matematiksel bir işlem olan evrişim işleminin uygulanmasından kaynaklanmaktadır. ESA'lar basitçe matriks çarpımı işleminin yerine evrişim işlemini kullanan yapay sinir ağlarıdır (Goodfellow, 2016).

Lecun, evrişimsel ağları iki katmanlı olarak oluşturmaktadır, bu katmanlar evrişim katmanları ve alt-örneklem (sub-sampling) katmanlarıdır. Her bir katmanın topografik bir yapısı vardır yani her bir nöron girdi resmin üzerinde bir yere denk gelen sabit bir pozisyonla ve algıladığı bölümle (receptive field) (nöronun tepkisi belirleyen resmin belirli bir bölümü) ilişkilendirilmektedir. Bu ağda her bir katmanda farklı girdi ağırlıklarına sahip nöronlar bulunmaktadır. Her nöron kendinden önceki katmandaki nöronların bir bölümüyle ilişkilendirilmektedir (Bengio, 2009).

ESA resimdeki nesnelerde kayma, büyüme veya küçülme, bükülme olsa dahi bu gibi problemlerden etkilenmemek için üç yapısal özelliği birleştirmektedir. Bu özellikler; lokal algı alanlar (local receptive fields), müşterek ağırlıklar (shared weights)

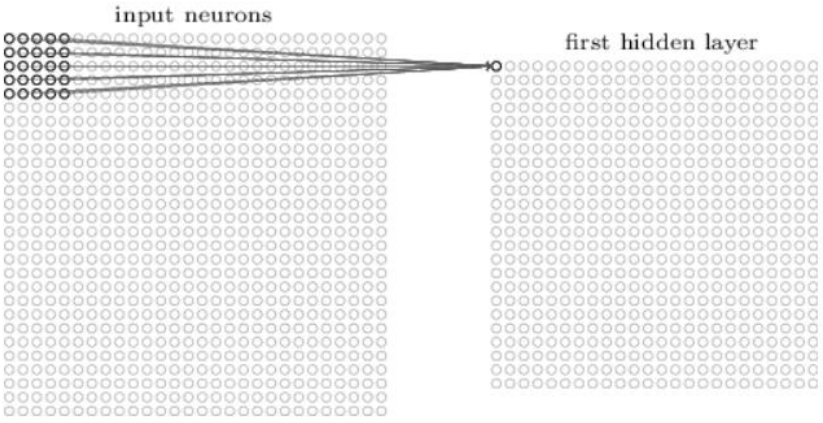
ve alt örneklemedir (sub-sampling/pooling). Örnek bir ESA aşağıdaki şekildeki gibidir. Bu ağ girdi olarak ortalanmış harf resimleri almaktadır. Katmanlardaki her bir birim bir önceki katmandaki yakın komşuluk alanındaki birimlerden girdileri almaktadır (Lecun, 1998).



Şekil 2.36 LeNet-5, evrişimsel sinir ağı (Lecun, 1998)

Katmanlardaki birimler aynı ağırlıkları paylaşacak şekilde düzlemsel olarak organize edilmektedir. Her birimin çıktı kümesi özellik haritası (feature map) olarak adlandırılmaktadır. Bu özellik haritasındaki birimler aynı işlemi resmin farklı bölgelerine uygulamaktadır. İdeal evrişim katmanı birden fazla özellik haritasından (farklı ağırlık vektörleri olan) oluşmaktadır bu sayede her pozisyonda farklı özelliklerin çıkarımını yapabilmektedir.

LeNet5'in ilk katmanı her biri özellik haritası olan altı düzlemden oluşmaktadır. Bu ağda girdi resmin her pikseli bir birime bağlanmak yerine resim bölgeleri birimlere bağlanmaktadır. Özellik haritasındaki bir birimin girdideki 5x5'lik alana bağlı 25 girdisi vardır, bu alana algı alanı denmektedir (Lecun, 1998).

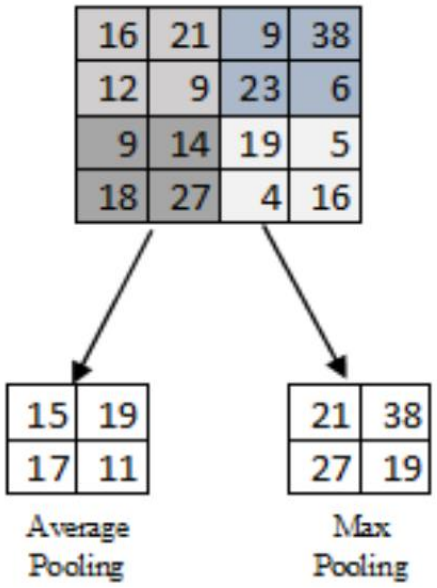


Şekil 2.37 Giriş nöronlarında algı alanı ve ilk gizli katman (Nielsen, 2015)

Daha sonra bu bölge piksel piksel kaydırılarak tüm resim taranmakta ve bir sonraki katmandaki ilgili birime bağlanmaktadır. Bu işleme kaydırma adımı (stride) denmektedir. Kaydırma adımı bu matrisin kaç piksel atlayarak tarama yapacağını belirlemektedir. Özellik haritasındaki birimler aynı ağırlık kümesini ve bias değerini kullanırlar bundan dolayı aynı özelliği girdi üzerinde olası tüm yerlerde tespit

etmektedirler. Bu yöntem sayesinde ESA'lar resimdeki nesneler kaydırılsa dahi yine de özelliklerini tespit edebilmektedir. Özellik haritasında aynı ağırlıkların ve bias değerinin kullanılmasından dolayı bu ağırlıkların ve bias değeri müşterek değerler olarak adlandırılmaktadır. Bu müşterek ağırlıklar ve bias değerleri de çekirdek (kernel) veya filtre denilen matrisleri oluşturmaktadır.

Evrişimsel ağlarda ayrıca pooling katmanları da bulunmaktadır. Bu katmanların görevi ağın evrişim katmanlarından çıkışlarındaki veri boyutu azaltmak ve dolayısıyla hesaplama süresini kısaltmaktır. Bu katmanlar genellikle evrişim katmanlarından sonra kullanılmaktadır. Her bir özellik haritasına ayrı ayrı pooling uygulanmaktadır. En çok kullanılan pooling yöntemleri maksimum ve ortalama pooling yöntemleridir. Bu yöntemlerde verilen iki boyutlu girdi, eşit boyutta matrislere bölünür daha sonra bu alt matrisler içindeki değerler, yönteme göre maksimumu veya ortalaması alınarak, daha küçük bir matrisle gösterilmektedir. LeNet5 ağında ortalama pooling kullanılmaktadır.



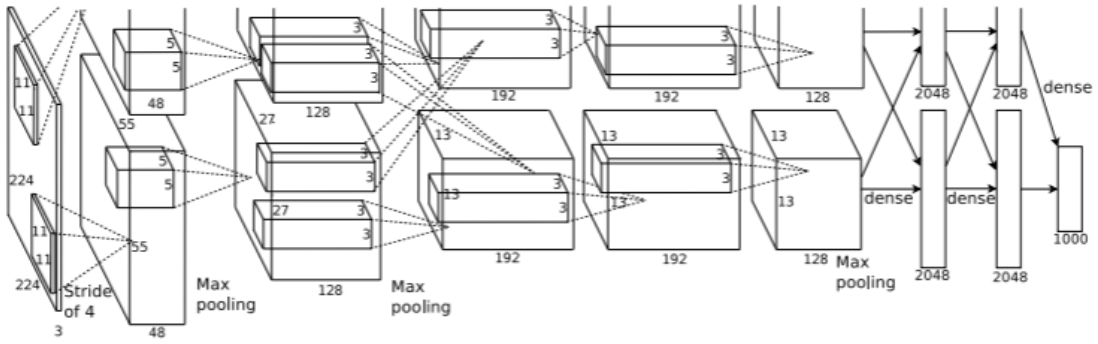
Şekil 2.38 Pooling yöntemleri (Taşhan, 2017)

LeNet5 ağının sonunda ise tam bağlantı (fully-connected) katmanı yer almaktadır. Bu katman, pooling katmanındaki her bir birimi bir sonraki katmandaki bir birime bağlamaktadır. Ağın en sonunda ise 10 sınıflı bir softmax katmanı bulunmaktadır. Bu katman sayesinde LeNet5 ağına girdi olarak verilen rakam resminin hangi rakama ne kadar olasılıkla benzediğini gösteren olasılık dağılımı hesaplanmaktadır.

Gelişmelerin gösterdiğine göre ESA'lar görüntü sınıflandırma alanında umut verici oranda başarılıdır. Güncel araştırmalar dikkate alındığında, görüntü sınıflandırmak için görüntü işleme tekniklerindense ESA kullanmak daha verimli görünmektedir.

Derin sinir ağları ve ESA'lar 2012 yılında AlexNet modeliyle tekrar popüler hale gelmiştir. Bu model 2012 yılındaki ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) yarışmasında, dikkat çeken bir başarı elde etmiştir. 2010'dan beri düzenlenen bu görüntü tanıma yarışması her sene düzenlenerek bilgisayarla görü alanında en iyi algoritmaların bulunmasını sağlamaktadır. Bu yarışmada ImageNet adı verilen veri kümesi kullanılmaktadır. ImageNet kümesinde 22.000 kategoriye ait 15 milyondan fazla resim bulunmaktadır. ILSVRC yarışmasında bu kümeden seçilen her kategoride 1000 resim olmak üzere 1000 kategorilik bir veri kümesi kullanılmaktadır. Yaklaşık 1.2 milyon eğitim resmi, 50.000 doğrulama resmi ve 150.000 test resmi kullanılmaktadır. ImageNet verisi üzerinde genellikle iki hata oranına bakılarak karşılaştırma yapılmaktadır. Bu oranlar ilk-1 hata oranı ve ilk-5 hata oranıdır. İlk-5 hata ortanı test resimlerine ait olası 5 kategoriyi tahminlemede ne kadar hata yapıldığını göstermektedir. AlexNet modeli ilk 5 tahminde test hatası oranında %15,3'i elde ederken, en yakın rakibi ise %26,2 oranında kalmıştır (Alex K., 2012).

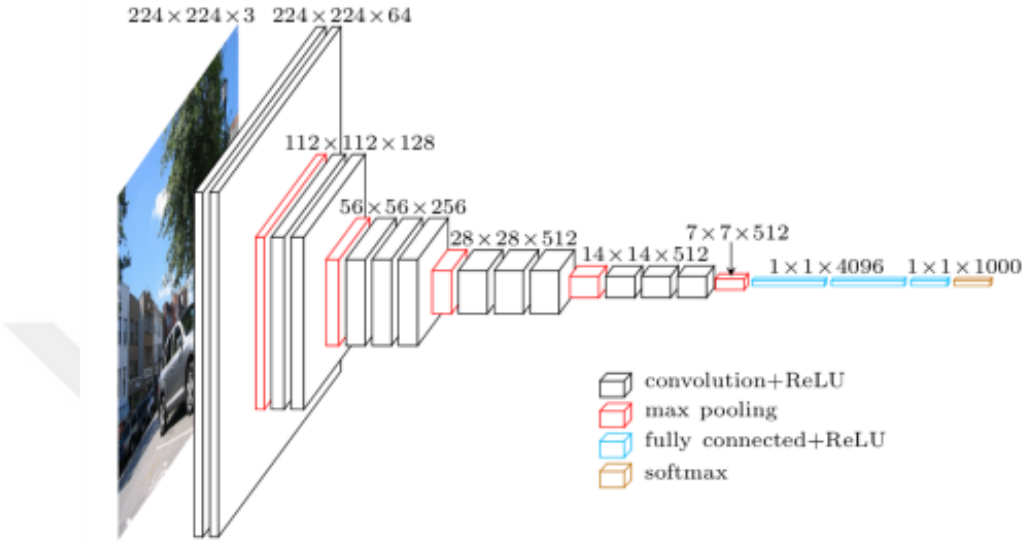
AlexNet modeli sekiz katmanlı bir ESA modelidir. Giriş katmanı, 224x224 boyutundaki renkli bir resmin kırmızı, yeşil, mavi değerlerini de temsil edecek şekilde 3x224x224 nörona sahiptir. İlk beş katman evrişimsel katmanlar, sonraki üç katman ise tam bağlantı katmanlarıdır. Ağın çıkışında ise tam bağlantı katmanı 1000 sınıflı bir softmax katmanıyla bu sınıflar için olasılık dağılımını oluşturmaktadır. Bu ağda da evrişim katmanlarından sonra pooling katmanları kullanılmaktadır. Bu model yaklaşık 60 milyon parametre kullanmaktadır. AlexNet modelinin mimarisi aşağıdaki gibidir.



Şekil 2.39 AlexNet modeli

2014 yılındaki ILSVRC yarışmasında ise Derin ESA modeli olan VGG16 modeli lokalizasyonda birinci, sınıflandırmada ise ikinci olmuştur. ImageNet verisi üzerinde eğitilerek test edilen bu model ilk 5 sınıflandırmada %92.7 tutarlılığa ulaşmıştır (Simonyan, 2014).

16 katmandan oluşan VGG16 modelinin mimarisi aşağıdaki gibidir:

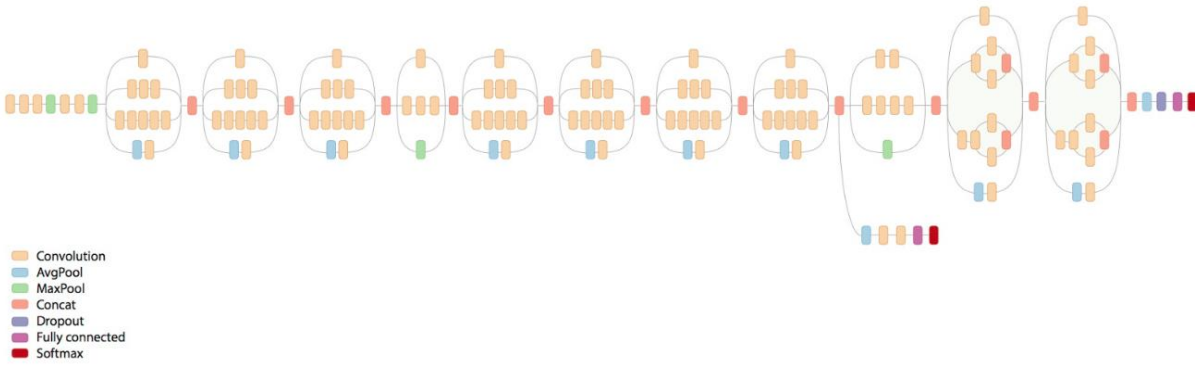


Şekil 2.40 VGG16 Mimarisi (Frossard, 2016)

Bu modelde evrişim katmanlarından sonra Relu aktivasyonu sonrasında da maksimum pooling kullanılmaktadır. AlexNet modelinde ilk katmanlarda daha büyük algı alanları (11x11, 4 kaydırma adımlı) kullanılmaktadır. Vgg16'da ise daha küçük (3x3, 1 kaydırma adımlı) algı alanları kullanılmaktadır. Ayrıca Vgg16 modeli evrişim katmanlarını 2'li ve 3'lü gruplar halinde art arda kullanmaktadır. Bu model yaklaşık olarak 138 milyon parametre kullanmaktadır.

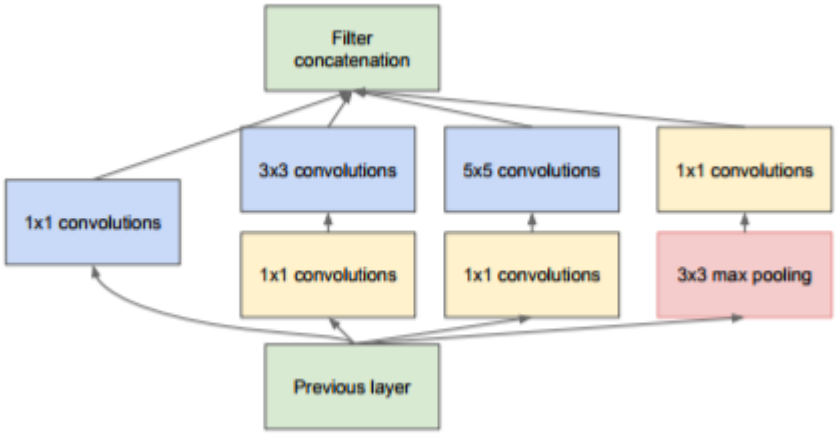
ESA'larda katmanlarına yeni bir bakış açısı getiren ağlardan birisi de Inception modelidir. Inception ağlarının ilk versiyonu olan 22 katmanlı bir ESA olan GoogleNet 2014 yılındaki ILSVRC yarışmasında, %6.7'lik ilk beş hata oranıyla sınıflandırma alanında birinci olmuştur.

GoogleNet ağı sadece parametreli katmanlar sayıldığında 22 katmanlıdır (pooling katmanları da dahil edilir ise 27 katman). 9 modül içeren bu ağda toplam 100 katman (bağımsız bloklar sayıldığında) bulunmaktadır (Szegedy, 2015a). Ağın görünümü aşağıdaki gibidir.



Şekil 2.41 GoogleNet modeli (Deshpande, 2016)

Bu ağ ESA'ların genel yöntemi olan evrişim ve pooling katmanlarını art arda sıralama yönteminin dışına çıkmaktadır. Inception ağında aşağıdaki gibi 1x1, 3x3 ve 5x5 boyutlarında paralel evrişim katmanları ve pooling katmanları kullanılmaktadır.

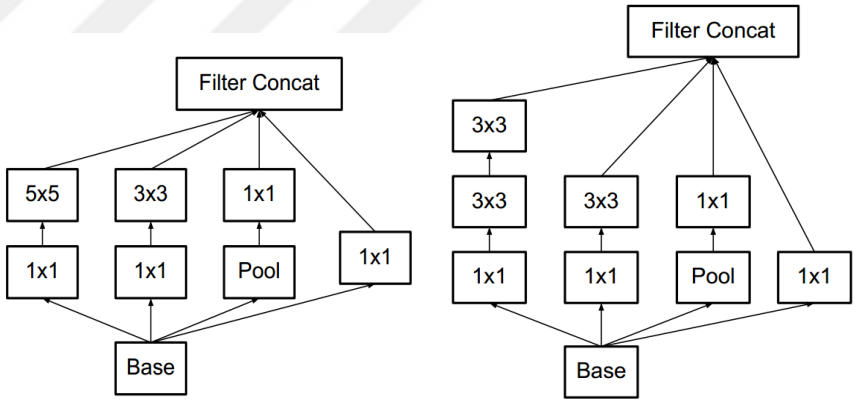


Şekil 2.42 Inception modülü (Szegedy, 2015a)

Bu ağın ana amaçlarından birisi de hesaplama maliyetini düşürmektir. Bundan dolayı maliyetli evrişim katmanlarından önce 1x1'lik evirişim katmanları kullanılarak boyut düşürülmektedir. Bu işlem darboğaz (bottleneck) olarak adlandırılmaktadır. Örneğin; 100x100x60 boyutundaki bir katman çıktısına 1x1'lik 20 filtre uygulanırsa, boyut 100x100x20 olmaktadır (Deshpande, 2016). Bu işlemin hesaplama sayısını düşürmesine rağmen başarılı olmasının nedeni ise girdi olarak verilen özelliklerin birbiriyle ilişkili olmasından kaynaklanmaktadır. Bu özelliklerin 1x1'lik evrişimlerle birleştirilmesi fazlalık yaratan veriyi ortadan kaldırmaktadır. Daha az özellikli evrişim katmanından sonra tekrar evrişim katmanıyla genişletme yapılarak sonraki katmana anlamlı bilginin iletilmesi sağlanmaktadır (Culurciello, 2017). GoogleNet modelindeki kazanımların çoğu evrişim katmanlarını verimli bir şekilde ayırabilmesinden kaynaklanmaktadır.

GoogleNet modeli daha sonra geliştirilerek Inception-v3 modeli oluşturulmuştur. Bu modelin hesaplama maliyeti Vgg16'dan daha düşüktür. Inception-v3 modeli ILSVRC 2012 doğrulama verisi üzerinde belirgin bir başarı göstermektedir. Her bir tahmin için 5 milyar çarpma-toplama ve 25 milyondan az parametre kullanarak, ilk kategoride %21,2 hata payına, ilk beş kategoride ise %5,6 hata payına ulaşmaktadır (Szegedy, 2015b).

Bu modelde evrişim katmanlarının daha verimli şekilde ayrılabileceği gösterilmektedir. Inception ağları tamamıyla evrişimsel ağlar oldukları için ağdaki her bir ağırlık, her aktivasyon işlemi için bir çarpma işlemi demektir. Bu modelde uygun bir evrişim ayırma yöntemiyle daha bağımsız parametreler elde edilerek ağın daha hızlı eğitilmesi sağlanmaktadır. Evrişim filtrelerinin büyütülmesi hesaplama maliyetini arttırmaktadır. 5x5'lik bir filtre 3x3'lük bir filtreye göre 25/9=2.78 kat daha masraflıdır ama daha büyük filtreler ilk katmanlarda birimler arası aktivasyon sinyallerini daha geniş ölçüde yakalamaktadır. GoogleNet inception modüllerindeki 5x5'lik evrişimler yerine Inception-v3 modülünde iki adet 3x3'lük evrişim katmanı kullanılmaktadır. Bu yöntemlerle hesaplama maliyetinde %28 oranında bir tasarruf sağlanmaktadır (Szegedy, 2015b). Bu modüller aşağıda gösterilmektedir.

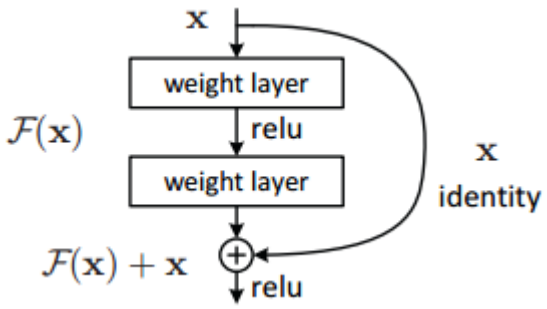


Şekil 2.43 GoogleNet ve Inception-v3 modülleri (Szegedy, 2015b)

Inception-v3 modelinde geliştirilen diğer bir unsur ise asimetrik evrişim katmanlarıdır. Asimetrik evrişim, nxn'lik bir katman kullanmak yerine nx1 ve 1xn'lik iki katmanı art arda kullanmaktır. 3x3'lük bir evrişim katmanı yerine 3x1 ve 1x3'lük iki katman kullanıldığında %33'lük bir tasarruf sağlanmaktadır. 3x3'lük yerine 2x2'lik bir katman kullanıldığında ise bu tasarruf oranı sadece %11'dir (Szegedy, 2015b).

Derin ağlarda önemli bir gelişme sağlayan diğer bir model ise Inception-v3 ile aynı zamanda çıkan ResNet (Residiual Network) modelidir. Bu ağın 152 katmanlı versiyonu %3.57'lik ilk-beş hata oranıyla 2015 yılındaki ILSVRC yarışmasında sınıflandırma alanında birinci olmuştur.

Bu ağ, YSA'ların derinleşmesiyle ortaya çıkan degradasyon problemi çözmeye odaklanmıştır. Bu problem sebebiyle ağın verimliği doygunluğa ulaştıktan sonra daha fazla katman eklemek ağın veriminin azaltmaktadır. ResNet modeli bu sorunu çözmek yeni bir teknik sunmaktadır. ResNet modeli aşağıdaki gibi bloklardan oluşmaktadır. Bu bloklarda iki katman öncesinden gelen artık değer blok sonuna eklenmektedir. Bu yöntem katmanlar arasında referans sağlayarak optimizasyonu kolaylaştırmaktadır (He, 2015). Örnek bir blok aşağıdaki gibidir:



Şekil 2.44 Artık (residual) blok

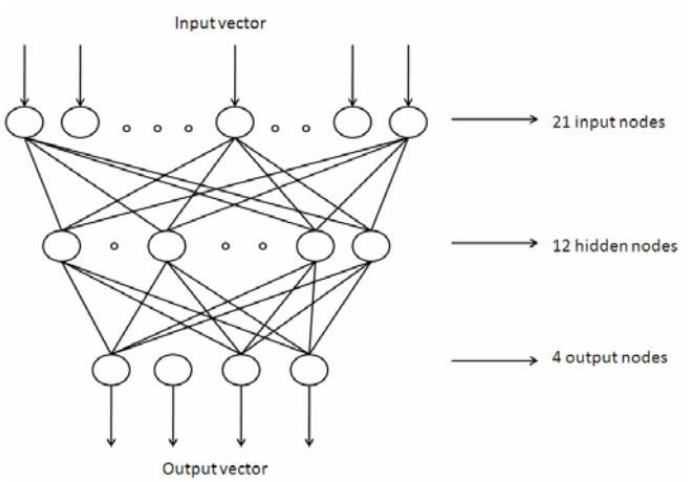
Bu bloktaki F(x)+ x ileri beslemeli bir sinir ağdaki kısayol bağlantısı olarak ifade edilmektedir. Kısayollar bir veya iki katmanı atlayabilmektedir. ResNet'de kısayol bağlantıları özdeşlik gönderimi (identity mapping) için kullanılmakta ve bu bağlantıların çıktıları katmanların çıktısına eklenmektedir.

ResNet ağlarında genellikle 3x3'lük filtreler kullanılmaktadır. Kaydırma adımı sayısı ise 2'dir. Ağın sonunda global ortalama pooling ve 1000 sınıflı bir softmax kullanılmaktadır.

2.2.7. YSA kullanan banknot tanıma çalışmaları

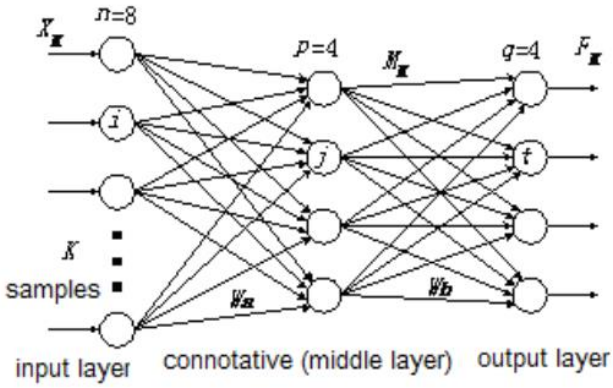
Literatürdeki YSA kullanarak görüntü sınıflandırma yapan çalışmalar incelenmiştir. Araştırmada mümkün olduğunca banknot sınıflandırmaya yönelik çalışmalara ağırlık verilmiştir.

Gunaratna ve ark. yaptığı çalışmada (2008) Sri Lanka banknotlarını sınıflandırmak için YSA kullanılmaktadır. Resimleri eğitmek için 3 katmanlı geri yayılımlı YSA kullanılmaktadır. Bu çalışmada sırasıyla şu adımlar uygulanmaktadır: gri ölçeğe çevirme, doğrusal dönüşüm, Canny algoritmasıyla kenarlar tespiti, özellik çıkarımı. Çıkarılan özellikler daha sonra eğitim verisi olarak kullanılarak YSA eğitimi yapılmaktadır. Bu çalışmadaki YSA'nın mimarisi aşağıdaki gibidir. Giriş katmanında 21 birim, gizli katmanda 12 birim, çıkış katmanında ise 4 birim bulunmaktadır. Gizli katmanlarda ve çıkış katmanlarında sigmoid fonksiyonu kullanılmaktadır.



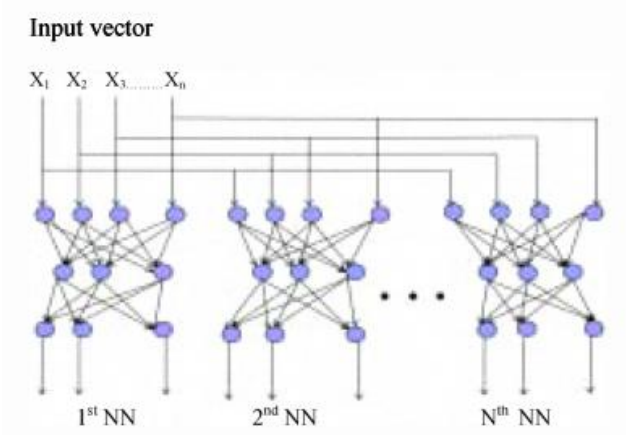
Şekil 2.45 YSA mimarisi (Gunaratna, 2008)

Wu ve ark. 2009 yılında yaptığı çalışmada ise Çin banknotlarının yönünü tespit etmek için, çok fazla pikseli tek nörona yüklemek yerine, resim 8 bölgeye bölünerek ayrı ayrı girdi olarak verilmektedir. Bu çalışmada 3 katmanlı geri yayılımlı ağ kullanılmaktadır. Bu ağ 10000 devirli bir eğitimle banknotların ön yüz yönünü %99 oranla tespit edebilmektedir (Wu, 2009).



Şekil 2.46 Geri yayılımlı ağ (Wu, 2009)

Bangladeş banknotlarını sınıflandırmak için yapılan bir çalışmada ise, ağ boyutunu küçültmek için görüntüler gri tonlamaya çevrilmekte ve daha sonra 125x80 boyutuna küçültülmektedir. Bu verilerle, her bir ağı 3 katmanlı ileri beslemeli ağdan oluşan birleştirilmiş YSA'lar eğitilip karşılaştırılmaktadır. Her bir ağ 7 çıktı nöronuna ve 70 gizli birime ve 10000 (125*80=10000) girdiye sahiptir. Girdi olarak hem eski hem yeni banknotlar kullanılmıştır. Eğitme tekniği olarak negatif korelasyonlu öğrenme kullanılmaktadır. Bu yöntem sayesinde birleştirilmiş ağ içindeki ağlar arasındaki hatalar ilişkilendirilerek tüm ağın daha verimli bir şekilde öğrenmesi amaçlanmaktadır (Debnath, 2010).



Şekil 2.47 Birleştirilmiş ağ (Debnath, 2010)

Cao ve Liu'un yaptığı çalışmada (2010) ise banknotları sınıflandırmak için geri yayılımlı bir sinir ağı genetik algoritmaya optimize edilmektedir. Genetik algoritma belirli bir problemi çözmek için bazı başlangıç çözümlerini kullanmaktadır. Daha sonra çözümlere sürekli olarak çaprazlama ve mutasyon uygulayarak daha iyi çözümler üretmektedir. Bu yeni üretilen çözümlerin hayatta kalmaları, problem tanımına ne kadar uyumlu olduklarına bağlıdır. Her bir nesille beraber daha iyi çözümler hayatta kalır ve bu şekilde daha iyi çözümler üretilebilmektedir.

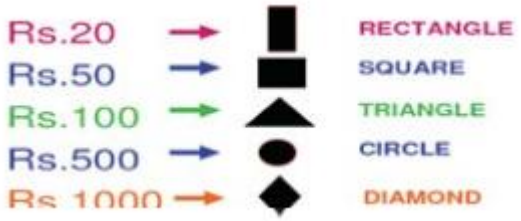
Bu yöntemde, ağın gerçek çıktısı ile hedeflenen çıktısı arasındaki hata farkına dayalı bir uyumluluk fonksiyonu (fitness function) oluşturularak gen algoritmasıyla ağın ağırlıkları değiştirilerek en verimli ağırlıklar bulunmaya çalışılmaktadır. Bu çalışmada genetik algoritmayla geri yayılımlı ağın eğitim süresini kısaltıp daha iyi sınıflandırma yapılabildiği gösterilmektedir. Geri yayılımlı ağ bu sınıflandırma çalışmasında %91.25 başarı sağlarken genetik algoritmayla geliştirilmiş versiyonu ise %94.37 başarı sağlamaktadır (X., 2010).

Bir başka çalışmada ise radyal bazlı fonksiyon ağları (RBFA) kullanılmaktadır. RBFA'lar aktivasyon fonksiyonu olarak radyal bazlı fonksiyon kullanan yapay sinir ağlarıdır. Radyal bazlı fonksiyonlar fonksiyonlara yakınsamak için kullanılmaktadır. RBFA modelinde, üç katman vardır: girdi katmanı, gizli katman ve çıktı katmanı. Girdi katmanı ara değerleme (interpolating) yapılacak veriden oluşmaktadır. Gizli katman doğrusal olmayan dönüşüm ile girdi katmanı değerlerini gizli katman değerlerine dönüştürmektedir. Bu çalışmada RBFA eğitimi için 110 Suudi Arabistan banknotu kullanmaktadır. Bu resimlerin 10'u 15 dereceden az açıyla eğik resimlerdir, geri kalanlar ise gürültülü ve normal resimlerden oluşmaktadır. Resimler ağa iletilmeden önce ön işlemeye tabi tutulmaktadır. Resimler önce gri tonlamaya daha sonra siyah beyaza çevrilmiştir. Resim kenarları Prewitt metoduyla filtrelenmekte ve Canny Kenar

algoritmasıyla da kenarları tespit edilmektedir. Daha sonra resmin boyutu, Prewitt ve Canny uygulamasından sonraki resim alanları, Euler numarası gibi özellikler ile RBFA ağı eğitilmektedir. Euler numarası bir resimdeki toplam nesne sayısından bu nesnelerdeki delik sayısını çıkartılarak hesaplanmaktadır.

Bu özelliklerle eğitilen RBFA ağının ortalama doğru tanıma oranı %91,51'dir. Tanıma oranı normal resimler için %95,37, gürültülü resimler için %91.65, eğik resimler için %87.5'dir (Sarfraz, 2015).

Hint banknotları üzerine yapılan bir sınıflandırma çalışmasında ise ayırt edici işaretlere odaklanılmaktadır. Hint banknotlarında farklı kupürlerde farklı şekiller bulunmaktadır.



Şekil 2.48 Kupür simgesi (Gogoi, 2015)

Gogoi ve arkadaşlarının yaptığı çalışmada (2015) bu tür banknotların renk, boyut ve kupür simgesiyle ayırt edilebileceği gösterilmektedir. İlk olarak resimdeki baskın renk elde edilmekte, daha sonra en boy oranı tespit edilerek bu özellikler çıkartılmaktadır. Ayrıca kupür simgesinin olduğu bölüm bölütleme ile ayrıştırılıp, bu bölümün özellikleri Fourier tanımlayıcıları kullanılarak çıkartılmaktadır. YSA farklı banknot tipleri için elde edilen Fourier tanımlayıcılarla eğitilmektedir. Kullanılan YSA, 3 katmanlı bir ileri beslemeli bir ağ olup 6 giriş birimi, 45 gizli birim ve 5 çıkış biriminden oluşmaktadır. Bu çalışmada YSA ile kupür simgesi sınıflandırılmakta ve genel algoritma içinde yardımcı bir adım olarak kullanılmaktadır. En boy oranı, baskın renk ve YSA sonucu bir algoritmayla değerlendirilerek banknot sınıfı tespit edilmektedir. Bu simgelerle banknot ayırımı için YSA eğiterek %97 başarı oranına ulaşmaktadırlar.

Bahreyn banknotlarıyla yapılan bir çalışmada (Althafiri, 2012) ise iki sınıflandırıcı kullanılmaktadır. Öncelikle banknot resimlerinin belirli özellikleri çıkartılmaktadır. Resim boyutu, kenar tespiti, Euler sayısı ve korelasyon katsayısı gibi faktörler kullanılmaktadır. İki sınıflandırıcı oluşturup karşılaştırılmaktadır, bunlar Öklid uzaklığına dayalı tanıma ve YSA ile tanımadır.

(x,y) koordinatındaki P ve (u,v) koordinatındaki Q noktaları arasındaki Öklid uzaklığı şu şekilde hesaplanmaktadır:

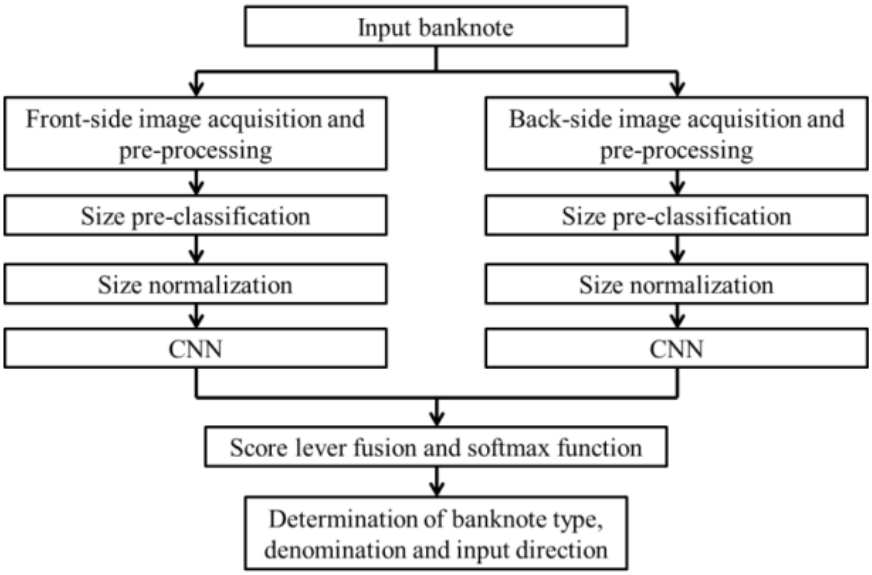
$$d(P,Q) = \sqrt{(x-u)^2 + (y-v)^2} \quad (2.24)$$

Öklid uzaklığına dayalı sınıflandırmada girdi verisiyle, hedef sınıflar arasındaki ağırlıklı Öklid uzaklığı hesaplanmakta ve en düşük uzaklığa sahip olan sınıf, bu verinin sınıfı olarak kabul edilmektedir. Bu yöntemde resim boyutu, kenarlar, Euler sayısı ve korelasyon katsayısı gibi özellikler kullanılarak, ağırlıklı Öklid uzaklığını hesaplanarak resimlerin benzerliği ölçülmektedir.

Bu çalışmada kullanılan YSA iki katmanlı ileri beslemeli bir ağdır. Ağın gizli katmanında 20 birim bulunmaktadır. Öklid uzaklığı ile tanıma sistemi %96,4 oranında başarı sağlarken, iki katmanlı ileri beslemeli YSA kullanan sistem %95,5 tanıma oranına ulaşmaktadır.

Bir diğer çalışmada sahte paraları ayırt edebilmek için YSA eğitilmektedir. Bu çalışmada UCI makine öğrenimi veri tabanında bulunan banknot veri kümesi kullanılarak gerçekleştirilmektedir. Bu veri kümesinde, dalgacık dönüşümüne uğramış banknot resimlerinin varyansı, çarpıklık katsayısı gibi bilgiler bulunmaktadır. Bu çalışmada kullanılan YSA 3 katmanlı olup, giriş katmanında 5 birim, gizli birimde 10 birim, çıkış biriminde ise 1 birim bulunmaktadır. Elde edilen, 1372 veri örneğinin 960 adedi eğitim verisi, 206 adedi test verisi, diğer 206 adedi ise doğrulama verisi olarak ayrılmaktadır. Yapılan sınıflandırma çalışmasında eğitim regresyonu 0,99914'ü test regresyonu 0,99786, validasyon regresyonu ise 0,9953 olarak hesaplanarak yüksek oranda başarılı bir sınıflandırma yaptığı tespit edilmektedir (Kaya, 2016).

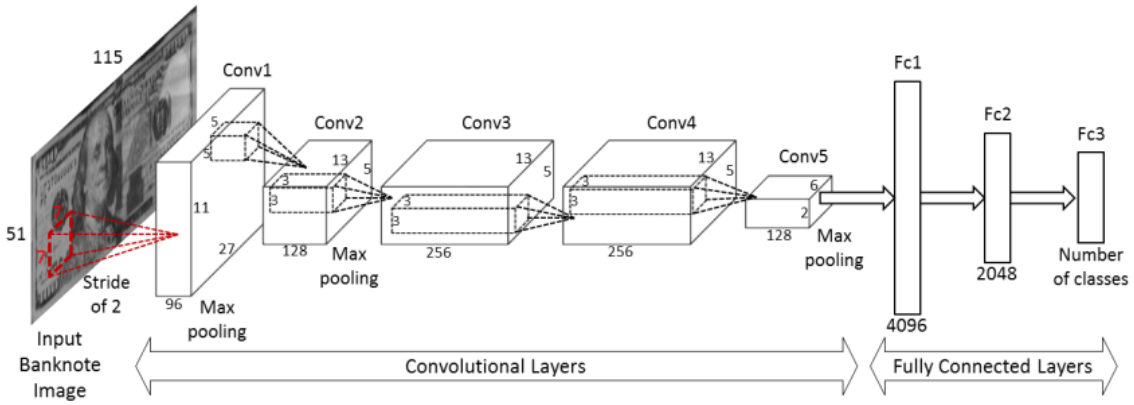
Yakın zamanda yapılan çalışmalardan de ise 6 farklı ülkeye ait 62 farklı banknot tipini sınıflandırmak için bir ESA eğitilmektedir. Bu çalışmada para sayma makinelerinden alınan banknot görüntüleri kullanılmaktadır. Sunulan sistemin mimarisi şu şekildedir:



Şekil 2.49 Banknot tanıma sistemi

Bu sistemde para sayma makinesinden elde edilen görüntü, paranın ön ve arka görüntüsü olarak iki farklı yol izlemektedir. Kenar tespit etme algoritmasıyla banknot görüntüsü arka plandan ayırt edilmektedir. Resimler daha sonra ESA'ya (CNN) girdi olarak verileceği için aynı boyuta (115 × 51) indirgenmektedir.

Çalışmada kullanılan ESA modeli AlexNet mimarisinden esinlenerek tasarlandığı belirtilmektedir. Bu ESA'da 5 adet evrişim katmanı (3'ü tam bağlantılı) bulunmaktadır. Tasarlanan ESA modeli aşağıda gösterilmektedir.



Şekil 2.50 ESA mimarisi (Pham, 2017)

Bu modelde Relu aktivasyon fonksiyonu kullanılmaktadır. Evrişim katmanları arasında ise maksimum pooling katmanları kullanılmaktadır. Aşırı uyum sağlama probleminden kaçınmak için fc2 olarak belirtilen tam bağlantılı katmanda ise dropout katmanı kullanılmaktadır. Eğitim metodu olarak rastgele eğim azaltma (stochastic gradient descent) kullanılmaktadır. Bu ESA modeli, 64.668 adet resim kullanılarak 100 devirde eğitilmektedir. Bu model %100 tanıma sağlamaktadır (Pham, 2017).

Literatür taramasında görüldüğü üzere, çalışmalarda kullanılan YSA'lar sınıflandırma konusunda genel olarak başarılıdır. Bazı çalışmalarda resimlerden özellik çıkarımı yapılarak bu veriler ile model eğitilirken, bazılarında ise ham resim verisi direkt girdi olarak kullanılmaktadır. YSA'ların son zamanlarda gelişmesiyle ve derinleşmesiyle beraber artık resim sınıflandırma işlemi için görüntü işleme teknikleri kullanılmadan da başarı sağlanabilmektedir. Özellikle evrişimsel sinir ağları resimlerin kategorisel özellikleri kendi kendine öğrenebilmektedir. Bu çalışmada da resimlerin ham verisi kullanılarak evrişimsel sinir ağlarından faydalanılacaktır.

3. MATERYAL VE YÖNTEM

3.1 Veri Kümesi Oluşturulması ve Ön İşleme

Veri kümesi oluşturmak için banknotların resimleri farklı ışıklandırmalar altında farklı açılardan, cep telefonu kamerası ile çekilmiştir. Banknotların arka ve ön yüzleri veri setinde aynı oranda tutulmaya çalışılmıştır. Bu şekilde altı farklı Türk banknotunun 1540 adet resmi derlenmiştir.



Şekil 3.1 Veri kümesinden örnekler

Tüm resimler 200x200 boyutuna getirilmiştir. Resimler banknot tiplerine göre gruplandırılarak ayrıştırılmıştır. Veri kümesinde her banknot tipi için aynı sayıda resim bulunmaktadır. Bu resimlerin %10'u doğrulama için kullanılmak üzere ayrılmıştır. Eğitimin sonuçlarının sağlıklı ölçülebilmesi için doğrulama kümesi, modelin daha önceden görmediği; yani eğitim kümesinde bulunmayan resimlerden seçilmiştir. Bu veri kümelerinin farklı resimlerden oluşması, YSA eğitimlerinde bilinen bir problem olan aşırı uyum sağlamanın (overfitting) üstesinden gelinmesi için çok önemlidir. Aşırı uyum sağlama problemi; modelin eğitim verisi üstünde çok iyi sonuçlar verip, daha önceden görmediği veriler karşısında başarısız olmasıdır.

Veri kümesinin büyüklüğü öğrenmeyi büyük ölçüde etkilemektedir. Veri sayısını ve çeşitliliğini yapay yöntemlerle arttırmak için veri artırma (data augmentation) işleminden faydalanılmaktadır. Bu işlem ile resimler bazı bozulmalara uğratılmaktadır. Resim üzerinde açısal döndürme, perspektif değiştirme, kaydırma, yakınlaştırma gibi işlemler yapılarak resimlerin çeşitliliği arttırılmaya çalışılmaktadır. Bu çalışmada da veri arttırma tekniklerinden faydalanılarak resim sayısı 1540'dan 2640'a çıkartılmaktadır. Bu çalışmada 1540 resimden oluşan veri kümesi küçük veri kümesi, 2640 resimden oluşan veri kümesi ise büyük veri kümesi olarak anılacaktır. Resim içindeki banknotların farklı pozisyonlarını elde edebilmek için, 90 derecelik

açısal döndürme, kaydırma ve yakınlaştırma işlemleri kullanılmaktadır. Elde edilen yeni veri kümesinin 2400 adedi eğitim için 240 adedi ise doğrulama için kullanılmaktadır.

Çalışmalar hem küçük veri kümesiyle hem de büyük veri kümesiyle yapılarak, modellerin karşılaştırılması sağlanmaktadır.

3.3 ESA Eğitim Süreci

Model eğitmenin iki yolu vardır: sıfırdan eğitme ve öğrenim transferi. Öğrenim transferiyle karşılaştırıldığında sıfırdan eğitme işlemi Inception gibi derin sinir ağları modellerinde, eğer veri kümesi de büyükse çok daha fazla zaman almaktadır. Ayrıca küçük veri kümelerinde öğrenim transferi yöntemi tercih edilmektedir.

Öğrenim transferinde birbiriyle yarışan bazı modeller bulunmaktadır. Bu modeller ve yeni çıkan modeller her sene ortak bir veri kümesi üzerinde test edilip birbirleriyle karşılaştırılmaktadırlar. ImageNet kümesi üzerinde eğitilen modeller daha sonra görmedikleri resimler ile test edilerek verimlilikleri ve hata oranları ölçülmektedir. Bu modellerden özellikle VGG16, InceptionV3 ve ResNet50 modelleri ön plana çıkmaktadır. Bu çalışmada öğrenim transferi yöntemiyle bu modeller yeniden eğitilerek banknot tanıma işlemi için karşılaştırılmaktadır. Ayrıca referans olması için sıfırdan yeni ESA modelleri oluşturularak bu modellerle karşılaştırılmaktadır.

Eğitim için Keras kütüphanesi kullanılmaktadır. Keras kütüphanesi Tensorflow üzerine geliştirilen YSA eğitim kütüphanesidir. Tensorflow kütüphanesi veri akışı grafiklerini kullanarak numerik hesaplama yapılabilen açık kaynaklı bir kütüphanedir. Google tarafından geliştirilen bu kütüphane, makine öğrenimi ve derin sinir ağları araştırmaları için hazır metotlar sunmaktadır. Tensorflow sitesinde performans ölçümünün sağlıklı yapılması için şu öneriler bulunmaktadır: "Ağın gerçek performansını ölçmek için gerçek ölçü; eğitim verisinde bulunmayan veriler üzerinde test etmektir, bu sayede doğrulama (validation) tutarlılığı ölçülür. Eğitim verisi üzerinde tutarlığı yüksek olmasına rağmen doğrulama tutarlılığı düşükse, bu ağın aşırı uyum sağladığını ve eğitim resimlerindeki genelde işe yaramayan gereksiz özellikleri hatırladığını gösterir. Çapraz entropi, öğrenme sürecinin nasıl ilerlediğini görmemize sağlayan bir kayıp fonksiyonudur." (2016).

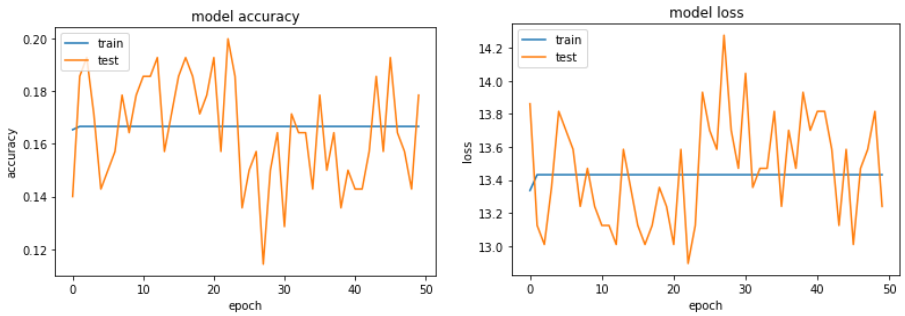
Bu çalışmada eğitim sürecinde öncelikle banknot sınıflandırma işlemi için sıfırdan bir model oluşturulmaktadır ve bu modelle ilgili gerekli ölçümler yapılmaktadır. Bu model oluşturulduktan sonra ise önceden başka veri kümesi üzerinde eğitilmiş olan üç başarılı model (VGG16, InceptionV3 ve ResNet50) banknot tanıma

işlemi için yeniden eğitilmektedir. Yeni model bu modellerle karşılaştırılarak, en başarılı model bulunmakta ve bu çalışmada kullanılmaktadır.

3.3 Sıfırdan Model Eğitimi

Bu çalışmada sıfırdan model eğitimi için öncelikle az sayıda katmanı olan bir ESA modeliyle çalışmalara başlanmaktadır. Modelde evrişim çıktı boyutları, katman sayıları değiştirilerek yüksek doğruluk ve minimum kayıp oranına ulaşılmaya çalışılmaktadır. Ayrıca maxpooling katmanları ve tam bağlantı katmanları da kullanılmaktadır. Kayıp fonksiyonu olarak çapraz-entropi fonksiyonu kullanılmaktadır. Optimizasyon fonksiyonu olarak dinamik bir şekilde öğrenme oranını ayarlayabilen Adadelta algoritması kullanılmaktadır. Tüm ağlar 50 devirde eğitilmektedir. Hesaplama maliyetini düşürmek için aktivasyon metodu olarak relu fonksiyonu tercih edilmektedir. Bu çalışmada kaydırma adımı olarak 2'nin 1'e göre daha iyi sonuçlar verdiği görüldüğünden 2 tercih edilmektedir. Tüm eğitimler Tesla K80 ekran kartı kullanılarak gerçekleştirilmektedir.

İlk olarak 3 evrişim katmanı, 1 tam bağlantı katmanı ve softmax katmanına sahip 5 katmanlı bir ağ genişletilmiş veri kümesi üzerinde eğitilmektedir. Evrişim katmanları ve tam bağlantı katmanının her biri 64 çıktıya sahiptir. Evrişim katmanları arasında 2x2 matrisli maxpooling katmanları kullanılmaktadır. En son katman olarak altı çıktılı softmax katmanı kullanılmaktadır. Evrişim katmanının matris boyutu 3x3'dür. Bu 4 katmanlı ağın eğitim sürecindeki doğruluk ve kayıp grafiği aşağıdaki gibidir.

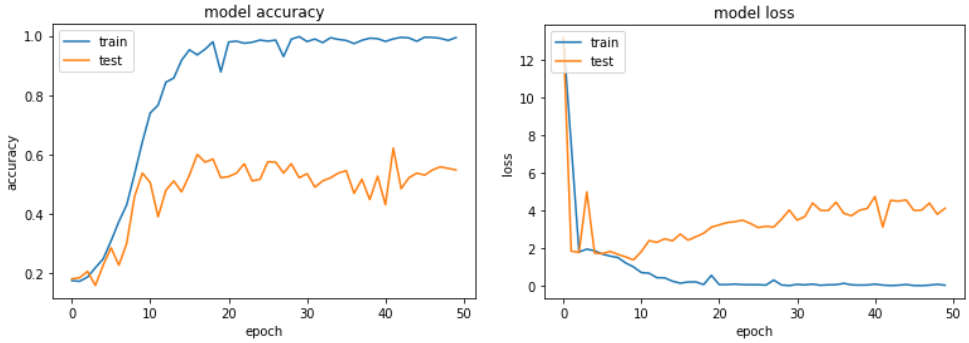


Şekil 3.2 5 katmanlı ağ eğitim grafiği

Bu ağın eğitim sonucunda ulaştığı değerler şu şekildedir: eğitim verisi için çapraz-entropi: 13.43, doğruluk: 0.166, doğrulama verisi çapraz-entropi: 13.23, doğrulama verisi doğruluğu 0.17. Bu ağın doğruluğu düşük, kaybı ise yüksektir henüz yeterince öğrenememektedir.

Öğrenmenin artırılması için Vgg16 ağından ilham alınarak katman sayısı ve katmanların çıktıları giderek artırılmaktadır. Bu amaçla 8 katmanlı (6 evrişim, 1 tam bağlantı, 1 softmax) bir ağ oluşturulmaktadır. Evrişim katmanları çıktı sayıları

64,64,128,128, 256 ve 256'dir. En son tam bağlantı katmanının çıktısı ise 128'dir. Bu 8 katmanlı ESA'nın doğruluk ve kayıp grafikleri aşağıda gösterilmektedir.

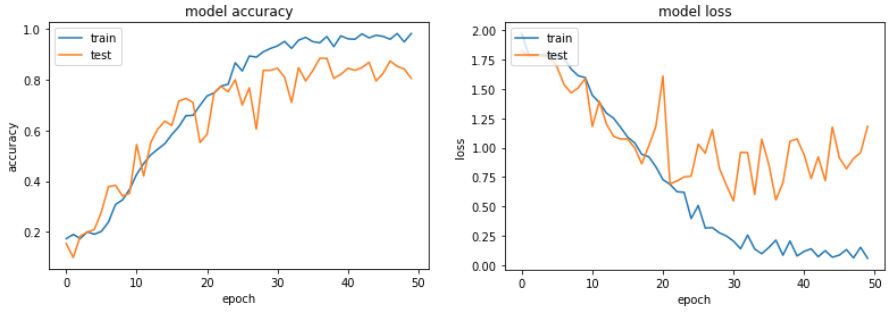


Şekil 3.3 8 katmanlı ağın doğruluk ve kayıp grafiği

Bu ağın eğitimi sonucunda ulaşılan değerler şu şekildedir: eğitim verisi için çapraz-entropi 0.034, doğruluk 0.993, doğrulama verisi için çapraz-entropi 4.13, doğruluk 0.547.

Ağın katman sayısını artırmak grafikte de göründüğü gibi eğitim verisi üzerindeki doğruluğunu arttırmaktadır. Doğrulama verisi üzerinde ise yeteri doğruluğa henüz ulaşamamaktadır. Bu durum ağın katmanlarının yetersiz olduğundan veya aşırı uyum sağladığından kaynaklanabilmektedir. Kayıp fonksiyonu da eğitim verisi için 0'a yakınsarken, doğrulama verisi için 2'nin altına düşmemektedir.

Bu aşamada öğrenimi artırmak için katman sayısı artırılarak 11 katmanlı bir ağ oluşturulmuştur. Bu ağ; 9 adet 64 çıktılı evrişim katmanı, 1 adet 64 çıktılı tam bağlantı katmanı ve son katman olarak da 6 çıktılı bir softmax katmanına sahiptir. Ayrıca aşırı uyum problemine önlemek için tam bağlantı katmanından sonra 0.5'lik bir dropout katmanı eklenmiştir. Ağın eğitim süresince ölçülen doğruluk ve kayıp fonksiyon grafikleri aşağıda gösterilmektedir.

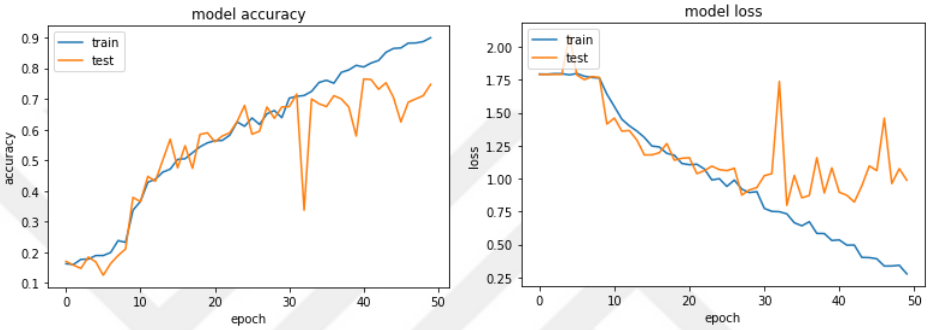


Şekil 3.4 11 katmanlı ağın doğruluk ve kayıp grafiği

11 katmanlı ağ, evrişim katmanlarının çıktıları daha küçük olmasına rağmen grafiklerde de göründüğü gibi, 8 katmanlı ağa göre daha iyi öğrenmektedir. Doğruluk grafiğinde görüldüğü üzere eğitim ve doğrulama verisi arasındaki fark kapanmıştır. Kayıp grafiğinde de eğitim verisi için 0'a yakınsamakta, doğrulama verisi için de zaman

zaman 1'in altına düşmektedir. Bu ağın eğitimi sonucunda ulaşılan değerler şu şekildedir: eğitim verisi için çapraz-entropi: 0.057, doğruluk 0.981, doğrulama verisi için çapraz-entropi 1.18, doğruluk 0.805.

Modelde katman sayısı artırılarak denemelere devam edilmektedir. Bir sonraki aşamada katman sayısı 17 adede çıkartılarak deneme yapılmaktadır. Bu ağın evirişim ve tam bağlantı katmanı 64 çıktılı katmanlardan oluştuğunda şu sonuçlar ölçülmektedir: Eğitim verisi için çapraz-entropi: 0.276, doğruluk: 0.899, doğrulama verisi için çapraz-entropi 0.987, doğruluk 0.747. Bu ağın eğitim sürecince ölçülen değerleri aşağıdaki grafikte gösterilmektedir.



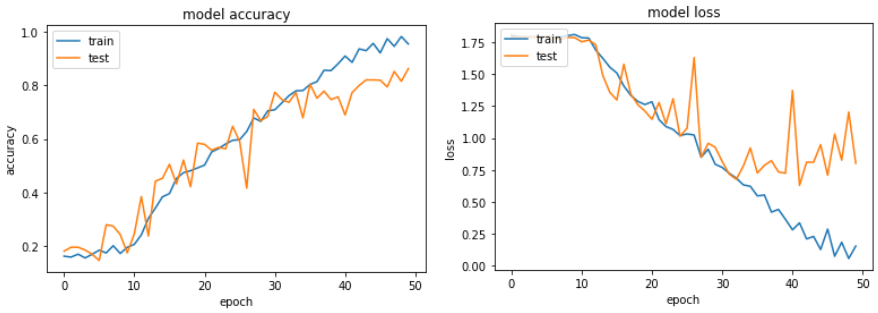
Şekil 3.5 17 katmanlı (64'lük çıktılı) ağ doğruluk ve kayıp grafiği

17 katmanlı bu ağın öğrenme yeteneklerini daha da artırmak için bazı düzenlemeler yapılmaktadır. Denemelerde en iyi değere ulaşmak için ağın Vgg16 ağında olduğu gibi giderek artan sayıda çıktılara sahip evrişim katmanları kullanılması gerektiği görülmektedir. Bu ağda sırasıyla 64, 128 ve 256 çıkışlı evirişim katmanları gruplar halinde kullanılmaktadır. Bu ağın modeli aşağıda gösterilmektedir.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 200, 200, 64)	1792
activation_1 (Activation)	(None, 200, 200, 64)	0
conv2d_2 (Conv2D)	(None, 200, 200, 64)	36928
max_pooling2d_1 (MaxPooling2	(None, 100, 100, 64)	0
conv2d_3 (Conv2D)	(None, 100, 100, 64)	36928
conv2d_4 (Conv2D)	(None, 100, 100, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 50, 50, 64)	0
conv2d_5 (Conv2D)	(None, 50, 50, 64)	36928
conv2d_6 (Conv2D)	(None, 50, 50, 64)	36928
max_pooling2d_3 (MaxPooling2	(None, 25, 25, 64)	0
conv2d_7 (Conv2D)	(None, 25, 25, 128)	73856
conv2d_8 (Conv2D)	(None, 25, 25, 128)	147584
conv2d_9 (Conv2D)	(None, 25, 25, 128)	147584
max_pooling2d_4 (MaxPooling2	(None, 12, 12, 128)	0
conv2d_10 (Conv2D)	(None, 12, 12, 256)	295168
conv2d_11 (Conv2D)	(None, 12, 12, 256)	590080
conv2d_12 (Conv2D)	(None, 12, 12, 256)	590080
max_pooling2d_5 (MaxPooling2	(None, 6, 6, 256)	0
conv2d_13 (Conv2D)	(None, 6, 6, 512)	1180160
conv2d_14 (Conv2D)	(None, 6, 6, 512)	2359808
conv2d_15 (Conv2D)	(None, 6, 6, 512)	2359808
max_pooling2d_6 (MaxPooling2	(None, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 64)	294976
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 6)	390

Şekil 3.6 17 katmanlı ağ modeli

Bu ağ eğitildiğinde sıfırdan eğitilen modeller arasında en iyi değerlere ulaştığı görülmektedir. Eğitim verisi üzerinde çapraz-entropi 0.153, doğruluğu 0.955; doğrulama verisi üzerinde çapraz-entropi 0.802, doğruluğu ise 0.862 olarak ölçülmektedir. Bu model 8.225.926 adet parametreye sahiptir.



Şekil 3.7 17 katmanlı ağ ölçümleri

Bu aşamadan sonra katman sayısını artırmak metrikleri daha iyi hale getirmemektedir.

Bu çalışmada banknot tanıma için genişletilmiş veri kümesi üzerinde sıfırdan eğitilen modellerin karşılaştırma sonuçları aşağıdaki çizelgede gösterilmektedir.

Çizelge 3.1 Sıfırdan eğitilen ESA'ların karşılaştırılması

Model	Eğitim verisi-doğruluk	Doğrulama verisi-doğruluk	Eğitim verisi-çapraz-entropi	Doğrulama verisi-çapraz-entropi
5 katmanlı	0.166	0.17	13.43	13.23
8 katmanlı	0.993	0.547	0.034	4.13
11 katmanlı	0.981	0.805	0.057	1.18
17 katmanlı (64 çıkışlı)	0.899	0.747	0.276	0.987
17 katmanlı (artan çıkışlı)	0.955	0.862	0.153	0.802

Bu ölçümler incelendiğinde en iyi sonucu veren modelin 17 katmanlı artan çıkışlı evrişim katmanlarına sahip olan model olduğu görülmektedir. Bu model 8 katmanlı ağa göre eğitim verisi doğruluğunda biraz geride kalsa dahi doğrulama verisi üzerindeki düşük çapraz-entropi oranı ve yüksek doğruluk oranıyla daha iyi öğrenme gerçekleştirmektedir.

3.4 Modellerin Yeniden Eğitilerek Karşılaştırılmaları

Bu bölümde görüntü sınıflandırma konusunda başarılarını kanıtlamış üç model (Inception-V3, Vgg16 ve ResNet50) banknot sınıflandırma işlemi için yeniden eğitilmektedir. Eğitilen modeller önceki bölümde sıfırdan oluşturulan 17 katmanlı ESA modelinin sonuçlarıyla karşılaştırılmaktadır. Tüm eğitimler Tesla K80 ekran kartı kullanılarak gerçekleştirilmektedir.

Araştırma için seçilen üç modeli karşılaştırmak için son katmanları çıkartılarak bunnun yerine max-pooling, Relu aktivasyonlu 1024 çıktılı dense ve softmax aktivasyonlu 7 çıktılı dense katmanları eklenmektedir. Bu ağların önceki katmanları dondurularak eğitim süresinde sadece son eklenen bölüm eğitilmektedir. Relu aktivasyonu ağın daha hızlı öğrenmesi için eklenmiştir. Softmax fonksiyonu ise banknot sınıflarının olasılıklarını hesaplamak için en sona eklenmiştir.

Sıfırdan eğitme işleminin öğrenim transferi modelleriyle karşılaştırılması için de önceki bölümde geliştirilen 17 katmanlı ESA kullanılmaktadır.

Bu önceden eğitilmiş modeller ve yeni oluşturulan basit ESA modeli, oluşturulan veri seti üzerinde 50 devir (epoch) ile eğitilerek süreleri, çapraz entropileri, eğitim verisi üzerinde ve validasyon verisi üzerindeki tahminleme tutarlılıkları ölçülmüştür. Optimizasyon için Adadelta, kayıp fonksiyonu olarak ise çapraz entropi seçilmiştir. Zeiler'in 2012 yılında yayınladığı Adadelta yaygın olarak kullanılan bir öğrenme oranı hesaplayıcı metottur. Bu şekilde tüm faktörler sabit tutularak sadece modeller değiştirilerek karşılaştırma yapılmaktadır. Karşılaştırmalar 1540 adet resimden oluşan küçük veri kümesi ve 2640 adet resimden oluşan büyük veri kümesi üzerinde yapılmaktadır.

Küçük veri kümesi üzerinde yapılan karşılaştırmaların sonuçları aşağıdaki çizelgedeki gibidir:

Çizelge 3.2 Küçük veri kümesi üzerinde modellerin sonuçları

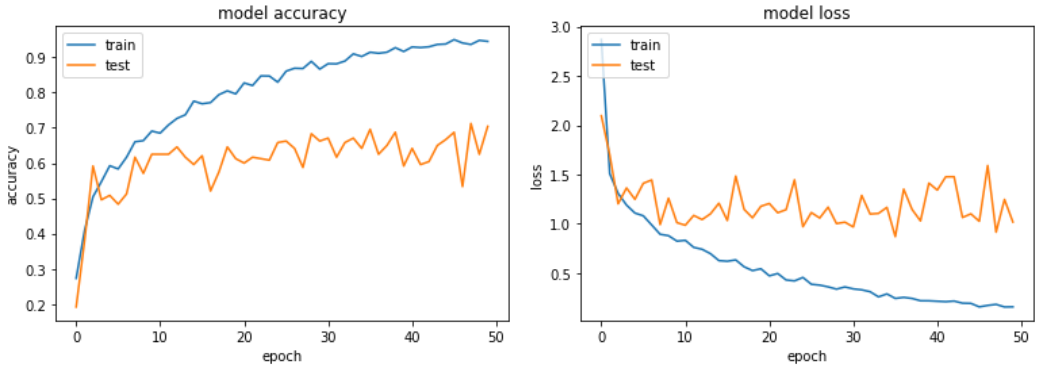
Model	Eğitim verisi - çapraz entropi	Eğitim verisi-doğruluk	Doğrulama verisi- çapraz entropi	Doğrulama verisi-doğruluk	Eğitim Süresi
InceptionV3	0.0205	0.9986	0.5333	0.84	10 dk
VGG16	0.1213	0.9616	0.3828	0.91	27 dk
ResNet50	1.4735e-04	1.0	0.0455	0.98	14 dk
17 katmanlı ESA	0.346	0.9929	1.1379	0.81	31 dk

Çapraz entropi; her bir devir ile sıfıra yakınsayarak modelin doğru tahmin etmeye ne kadar yakınsadığını gösterir. Doğruluk ise modelin daha önceden görmüş olduğu eğitim verisi üzerinde ne kadar doğru tahminleme yaptığını ölçer, 1'e yaklaşması doğru yönde gidildiğini gösterir ama bir yandan da eğitim verisi üzerinde çok iyi tahminleme yapmak aşırı uyum sağlamanın da habercisidir. Eğitim verisine aşırı uyum sağlayan bir model o veriler için çok iyi sonuçlar üretse de farklı bir veri gördüğünde bu kadar başarılı olamayabilir. Doğrulama verisi-çapraz entropi ise doğrulama verisi üzerindeki entropiyi göstermektedir. Doğrulama verisi üzerindeki doğruluk, validasyon verisi üzerinde yani modelin daha önceden görmediği veriler üzerinde ne kadar başarılı olduğunu gösterir. Bu araştırmada modelleri karşılaştırmak için doğrulama verisi doğruluğu en önemli metrik olmuştur. Eğitim süresi ise modelin banknot sınıflandırma işlemi yapmak üzere yeniden eğitilmesi için geçen süreyi göstermektedir.

Küçük veri kümesi üzerinde ResNet50 yüksek doğrulama verisi doğruluğu ve düşük entropisiyle bu modeller arasında banknot sınıflandırma için en iyi modeldir. İkinci model ise yüksek validasyon tutarlılığı ise Vgg16 olmuştur.

Aynı modeller veri artıma yöntemiyle oluşturulan büyük veri kümesi üzerinde de karşılaştırılmaktadır. Her bir model 50 devirde eğitilmektedir. Büyük veri kümesi 2400 adet eğitim resmi ve 240 adet doğrulama resminden oluşmaktadır.

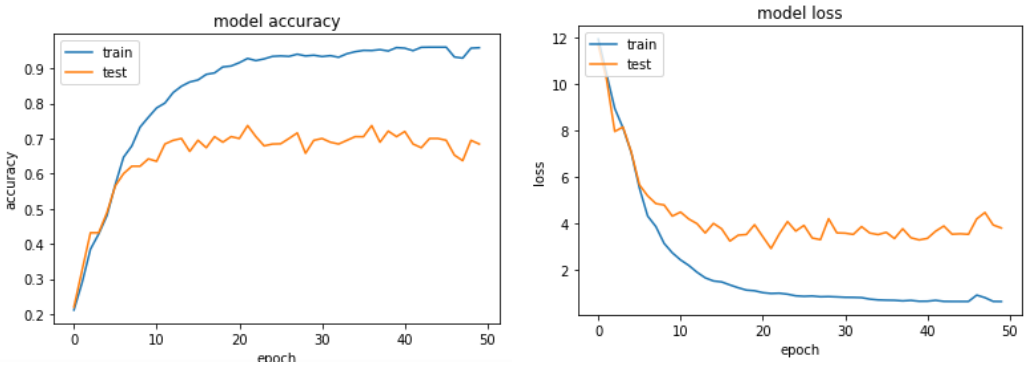
Inception-v3 modeli öğrenme yöntemiyle büyük veri kümesi üzerinde yeniden eğitilmektedir. Bu yeniden eğitilen model ile eğitim verisi üzerinde yapılan testlerde 0.159 çapraz entropi değerine, 0.945 doğruluk değerine ulaşılmaktadır. Doğrulama verisi üzerinde ise çapraz entropi değeri 1.019, doğruluk değeri ise 0.704 olarak ölçülmektedir. Modelin eğitim sürecindeki doğruluk ve kayıp oranını gösteren grafikler aşağıda gösterilmektedir.



Şekil 3.8 Inception-V3 doğruluk ve çapraz entropi grafiği

Inception-v3 modeli eğitim veri kümesi için %90'ın üzerinde başarılı tahminleme yaparken, daha önceden görmediği doğrulama veri kümesi üzerinde doğruluğu %70'i geçememektedir. Kayıp fonksiyonunu gösteren çapraz entropi değerleri de eğitim verisi için 0'a yakınsarken, doğrulama verisi için 1'in altına inememektedir.

Vgg16 modeli de büyük veri kümesiyle tekrar eğitilerek tutarlığı ve çapraz-entropi değerleri ölçülmektedir. Eğitim verisi üzerinde çapraz entropi 0.390, doğruluk 0.851 olarak ölçülmektedir. Doğrulama verisi üzerinde ise çapraz entropi 1.752, doğruluk 0.625 olarak ölçülmektedir. Bu model eğitilirken ölçülen doğruluk ve çapraz entropi grafikleri aşağıda gösterilmektedir.

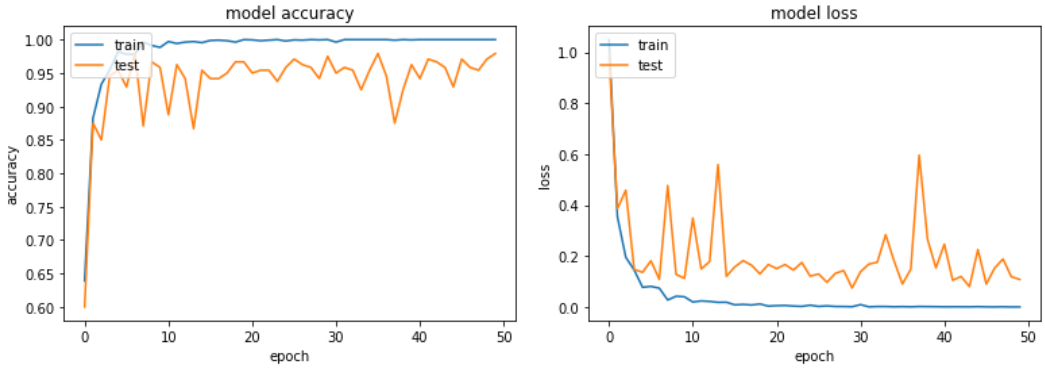


Şekil 3.9 Vgg16 doğruluk ve çapraz entropi grafiği

Vgg16 modeli de eğitim verisi üzerinde doğruluk oranlarında başarılı sonuçlar verirken doğrulama verisi üzerinde %70' i geçememektedir. Çapraz entropi de eğitim verisi üzerinde 1'in altına inerken doğrulama verisi üzerinde 3'ün altına inememektedir. Model eğitim verisini görece iyi sınıflandırabilirken doğrulama verisi üzerinde yeterince başarılı olamamaktadır.

ResNet50 modelinin tekrar eğitilmesiyle yapılan çalışmada ise şu sonuçlara ulaşılmaktadır. Eğitim verisi üzerinde çapraz-entropi 3.9096e-04, doğruluk 1.0, doğrulama verisi üzerinde ise çapraz-entropi 0.108, doğruluk 0.979 olarak

ölçülmektedir. Eğitim sürecinde eğitim ve test verisi üzerinde ölçülen değerler aşağıdaki grafiklerde gösterilmektedir.



Şekil 3.10 ResNet50 doğruluk ve çapraz-entropi grafiği

ResNet50 modeli doğruluk grafiğinde görüldüğü üzere, diğer modellere göre daha erken doğruluğa ve düşük çapraz entropiye ulaşmaktadır. Doğruluk oranı olarak hem eğitim hem de doğrulama verisi üzerinde yüzde 95'in üzerine çıkmaktadır. Ayrıca çapraz entropi değeri de eğitim verisi üzerinde 0'a yakınsamakta, doğrulama verisi üzerinde de 0.1'e ulaşmaktadır.

Çizelge 3.3 Büyük veri kümesi üzerinde modellerin sonuçları

Model	Eğitim verisi-çapraz entropi	Eğitim verisi - doğruluk	Doğrulama verisi-çapraz entropi	Doğrulama verisi-doğruluk	Eğitim Süresi
InceptionV3	0.159	0.945	1.019	0.704	15 dk
VGG16	0.390	0.851	1.752	0.625	35 dk
ResNet50	3.9096e-04	1.0	0.108	0.979	20 dk
17 katmanlı ESA	0.153	0.955	0.802	0.862	40 dk

Çizelge 3.3'de görüldüğü üzere ResNet50 modeli tüm metriklerde diğer modellerden daha başarılı çıkmaktadır. Ondan sonraki en iyi model ise doğrulama verisi üzerindeki yüksek doğruluğu ve düşük çapraz entropisi oranıyla 17 katmanlı ESA modelidir. Bu değerlendirmeler sonucu mobil uygulamada ResNet50 modelinin tercih edilmesi uygun görülmektedir.

3.5 Model Çıktısını iOS Platformunda Kullanma

Keras aracı kullanılarak tekrar eğitilen modellerin grafiği hiyerarşik bir veri formatı olan HDF5 dosyası olarak kaydedilmiştir (Group, 2016). Kaydedilen bu dosyalar Apple'ın makine öğrenimi dosya formatı olan CoreML dosyalarına dönüştürülmüştür (Apple, 2017b). Bu dönüşüm Coremltools isimli Python kütüphanesi kullanılarak gerçekleştirilmiştir (Apple, 2017a).

Oluşturulan CoreML model dosyalarının büyüklükleri Çizelge 3.3'de gösterilmiştir:

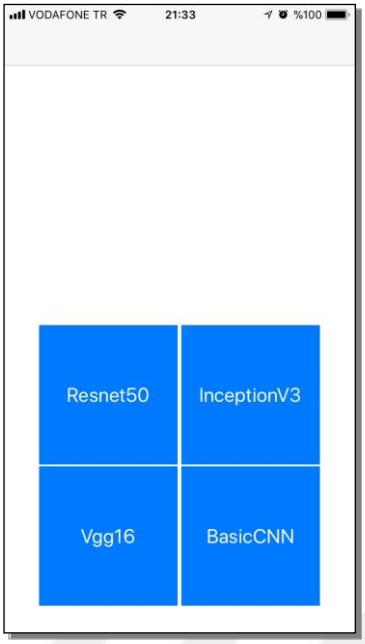
Çizelge 3.3 Modellerin CoreML dosya büyüklükleri

Model	Boyut(mb)
Vgg16	61
Resnet50	102
InceptionV3	96
17 katmanlı ESA	94

En büyük boyutlu olan ResNet50 modelidir ancak günümüz mobil cihazları hafıza genişlikleri düşünüldüğünde bu aralıklardaki dosya boyut farkları ihmal edilebilir.

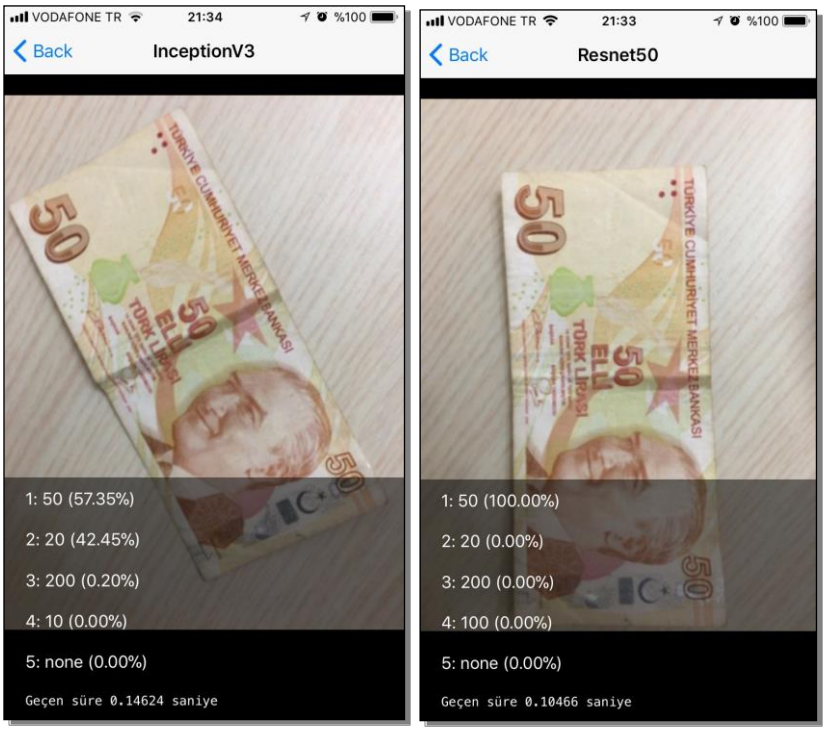
Banknot kategorileri ise yazı dosyası olarak kaydedilmiş ve dönüştürme işleminde CoreML modeline dahil edilmiştir. iOS uygulaması Apple'ın makine öğrenimi kütüphaneleri kullanılarak kaydedilen bu modelleri işleyecek şekilde geliştirilmiştir. Uygulama anlık video girdisini işler, bu girdiyi 200x200 piksel boyutuna dönüştürür ve bu veriyle modeli besler. Çalıştırılan model bu veri üzerine bir tahminde bulunur ve her bir sınıf için bir olasılık üretir. Model bu şekilde en yüksek oranlı sınıfı alarak değerlendirir, belirlenen eşik değerinden (0.8) büyükse bu sınıflandırmayı kabul eder ve sesli olarak banknot sınıfını okur. Bu eşik değeri denemeler sonucu bulunmuştur. Uygulamanın ilk ekranı farklı modelleri deneyebilmek için dört adet büyük butonla seçim ekranı olarak tasarlanmıştır.

Nihai üründe ise en iyi sonuçları veren model Resnet50 modeli seçim yapılmaksızın direkt olarak kullanılmaktadır. Ayrıca uygulama Apple'ın akıllı asistanı Siri'ye entegre olarak da kullanılabilmektedir. Bu sayede kullanıcı uygulamayı açmadan sadece "bu kaç para?" gibi bir seslendirmeyle uygulamayı açıp para tanımayı başlatabilmektedir. Modelleri karşılaştırmak için geliştirilen test uygulamasında farklı modelleri seçebilmek için ekran aşağıdaki gibi tasarlanmıştır:



Şekil 3.5 Para Tanıma Uygulaması Seçim Ekranı

Bu seçim yapıldıktan sonra seçilen evrişimsel sinir ağı modeli yüklenir ve kamera girdisini sürekli olarak işler, anlık olarak tüm sınıfları ve modelin işletilmesi sonucu oluşan tahmini yüzde olarak her model için gösterir. Uygulamanın son halinde bu ekran bulunmamaktadır, görme engelli kullanıcı uygulamayı açar açmaz kamera girdisi işlenmeye başlayıp ekrana dokunduğu anda sesli geri bildirim ile tanınan banknot tipi bildirilmektedir. Ekranda olasılıklar banknot tipi bazında gösterilmektedir. Ayrıca en altta modelin işletilip sonuç üretilmesi için geçen süre de gösterilmektedir:



Şekil 3.6 Para Tanıma Uygulaması

Şekil 3.6'daki ekran aktifken görme engelli şahıslar için ister sesli ister titreşim olarak geri bildirim verilmektedir. Geri bildirim için herhangi bir tuşa basmak gerekmez kullanıcının ekrana dokunması yeterlidir, uygulama direk olarak video girdisini işleyerek en yüksek oranlı gördüğü kategoriyi yani banknot tipini tespit eder ve geri bildirimde bulunur. Geri bildirim için iki seçenek mevcuttur sesli ve titreşimli geri bildirim. Sesli geri bildirimde görüntüdeki banknotun kategorisi tanımlandığında, uygulama tarafından banknotun tipi sesli olarak okunur ('Elli TL' gibi). Titreşimli bildirimde ise 5TL için bir titreşim, 10 TL için 2 titreşim şeklinde artan sayıda titreşim verilerek kullanıcının para tipini anlaması sağlanır.

En verimli modelin bulunabilmesi için mobil uygulama üzerinde de testler yapılarak 100 resim için ortalama kategori tespit süresi hesaplanmıştır ve farklı açılarda ve ışıkta, farklı tip banknotlar gösterilerek doğru tahminleme oranları geliştirilen mobil uygulama ile iPhone6s model bir telefon üzerinde denenmiştir. Son yapılan yüz tespit mobil uygulama içine kaydedilmiş ve yüz işlem için modelin kamera girdini işleyerek kategori sınıfını tespit etme sürelerinin ortalamaları aşağıdaki gibi ölçülmüştür:

Çizelge 3.4 Modellerin çalışma süreleri

Model	Süre(saniye)
Vgg16	0.53
Resnet50	0.22
InceptionV3	0.15
17 katmanlı ESA	0.52

Model çalışma süreleri karşılaştırıldığında en hızlı Inception-v3 modeli çıkmıştır. Vgg16 16 katmana, Resnet50 ise 50 katmana sahiptir. Bu derinlik farklarına rağmen tüm modeller görece hızlı sonuç vererek hepsi bir saniyenin altında sonuç üretmektedir. Modellerin bu şekilde hızlı çalışmaları uygulamalarda rahatça kullanılabileceklerini göstermektedir.

Mobil uygulamada doğru tespit oranlarını da ölçmek için son yüz işlemdeki tespit oranları ve resim kayıtları tutulmaktadır. Bu şekilde tüm banknot tipleri farklı ışıklarda ve açılarda görüntülenerek sonuçlar kayıt altına alınmıştır. Mobil uygulama ile ölçülen doğruluk oranları aşağıdaki gibidir:

Çizelge 3.5 Mobil uygulama üzerinde doğruluk oranları

Model	Doğruluk oranı (%)
Vgg16	39
Resnet50	92
InceptionV3	80
17 katmanlı ESA	85

Mobil uygulama ile cep telefonu üzerinde yapılan testlere göre yüzde 92 doğruluk oranı Resnet50 modeli diğerlerinden çok daha iyi sonuçlar vermektedir. Bu modelin doğruluk oranına en yakın model ise yüzde 85 oranıyla sıfırdan eğitilen 17 katmanlı ESA modelidir. Mobil uygulama ile yapılan testler direkt video girdisi üzerinde gerçek ortam testleri olduğu için sonuçlar diğer testlere göre daha önemlidir.

Bu veriler daha önceden bilgisayar üzerinde iki farklı veri kümesiyle yapılan karşılaştırmadaki sonuçlara benzerlik göstermektedir. Doğrulama kümesi modelin daha önceden görmediği resimlerden oluşmaktadır. Banknot tanıma için yeniden eğitilen modeller ile bilgisayar üzerinde yapılan testlerde doğrulama resimleri üzerinde aşağıdaki sonuçlar gözlemlenmektedir:

Çizelge 3.6 Bilgisayar üzerinde doğruluk oranları

Model	Doğruluk oranı (%)
Vgg16	41
Resnet50	99
InceptionV3	90
17 katmanlı ESA	93

Bu karşılaştırmaların gösterdiği üzere Resnet50 modeli hem cep telefonu ile hem de bilgisayar üzerinde yapılan testlerde doğru tespit oranları incelendiğinde diğerlerinden daha iyi sonuçlar vermektedir.

Bu modelde diğer modellerden farklı olarak alt alta bağlantılar haricinde, n. katmanla (n+x) katmanı arasında da ekstra bağlantılar mevcuttur. Bu bağlantılar sayesinde ağ derinleşmesiyle birlikte gelen degradasyon sorunuyla daha iyi mücadele edilebilmektedir. Degradasyon sorunu; ağın katmanları arttıkça ortaya çıkan doygunluk seviyesinden sonra ağın veriminin gittikçe azalmasıdır. Bu sorun ağa daha fazla katman eklemeyi engellemektedir. ResNet50 farklı seviyedeki katmanlar arasına ekstra kısayollar ekleyerek bu sorunu başarılı bir şekilde çözmektedir (He, 2015).

Bu karşılaştırmalar transfer öğrenimi metodu ile yani daha önceden eğitilmiş modellerin tekrar yeni bir iş için eğitilmesi yoluyla yapılmıştır. Bu sonuçlar göz önüne alınarak geliştirilen nihai mobil uygulamada kullanılacak model olarak Resnet50 modeli seçilmiştir. Uygulama açılır açılmaz kamera girdisini işleyerek bu modeli işleterek banknot tipini tahmin ederek geri bildirim verecek şekilde tasarlanmıştır.

4. SONUÇLAR VE ÖNERİLER

4.1 Sonuçlar

Bu çalışmada YSA tipindeki dört model çapraz entropi, doğruluk gibi ölçütlerle karşılaştırılmaktadır. Veri seti olarak 200x200 boyutunda 1540 adet banknot resminden oluşan küçük küme ve 2640 banknot resminden oluşan genişletilmiş küme ayrı ayrı kullanılmaktadır. 50 devir sonrasında ResNet50 modelinin eğitim verisi üzerinde tutarlılığı %100'dür. Bu performans ölçümü aşırı uyum sağlamayı göz ardı edebilir bundan dolayı doğrulama verisi üzerindeki doğruluk oranı ve çapraz entropi değeri ölçülmektedir. İki farklı veri kümesiyle yapılan karşılaştırmalar sonucunda banknotları sınıflandırmak için ResNet50 modelinin 17 katmanlı ESA, Inceptionv3 ve VGG16 modellerinden daha iyi sonuçlar verdiği gözlemlenmiştir. Yüksek GPU'lara gerek duymadan kişisel bilgisayarda kısa sürede eğitilerek yüksek oranda verim sağlanması isteniyorsa öğrenim transferi metodu kullanılabilir. Öğrenim transferi metoduyla görüldüğü üzere ilişkisi olmayan bir veri seti üzerinde eğitilmiş olan modeller farklı bir sınıflandırma işi için sadece son katmanı değiştirilerek kullanılabilmektedir. 20 dakikada tekrar eğitilmesine rağmen ResNet50 modeli doğrulama verisi üzerinde 0.97 doğruluk oranı sağlamaktadır. Bu sonuçlar göz önüne alındığında ResNet50 modeli banknot sınıflandırma için sıfırdan eğitilen 17 katmanlı ESA ve yeniden eğitilen InceptionV3 ve VGG16 modellerinden daha iyi bir öğrenme ve sınıflandırma sağlamaktadır.

4.2 Öneriler

Makine öğrenimi artık otomatik tercümelerden, sürücüsüz araçlara kadar geniş yelpazede kullanıma başlanmıştır. Giderek daha fazla alanda kullanılmaya ve günlük hayatımızın bir parçası olmaya başlayacaktır.

Bu çalışmada makine öğreniminin alt sınıfı olan yapay sinir ağları araştırılmış ve az veriyle hızlı şekilde eğitme yöntemi olan öğrenim transferi yöntemi kullanılmıştır. Aynı zamanda banknot tanıma konusunda gelecek araştırmalar için bir model oluşturulmuştur. Bu eğitim yöntemi farklı tip banknotlar ve sınıflar için kullanılabilir. Bu modelin verimliği veri kümesindeki resim çeşitliğini artırarak geliştirilebilir.

Bu konuyu bir adım öteye taşımak için "sıfır vuruş öğrenme" (zero shot learning) denilen daha önceden banknot verisi görmemiş bir model kullanılarak sınıflandırma yapma çalışmaları yapılabilir veya tek bir banknot resmiyle eğitilebilen, diğer olasılıkları (resim açısı, nesne ölçüsü gibi) kendi oluşturup eğitim verisini kendi oluşturan bir yöntem geliştirilebilir.

KAYNAKLAR

2016, How to Retrain Inception's Final Layer for New Categories https://www.tensorflow.org/tutorials/image_retraining:

Alex K., I. S. a. G. E., 2012, ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems 25*, 1097-1105.

Althafiri, E., Sarfraz, M. and Alfarras, M., 2012, Bahraini Paper Currency Recognition, *2012*, 2 (2).

Apple, 2017a, Community Tools for CoreML, https://pypi.python.org/pypi/coremltools:

Apple, 2017b, Core ML, https://developer.apple.com/documentation/coreml:

Bengio, Y., 2009, Learning Deep Architectures for AI, Foundations and Trends® in Machine Learning, p. 1-127.

Butina, D., 1999, Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets, *Journal of Chemical Information and Computer Sciences*, 39 (4), 747-750.

Canny, J., 1986, A Computational Approach to Edge Detection, *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*.

Culurciello, E., 2017, Neural Network Architectures, https://towardsdatascience.com/neural-network-architectures-156e5bad51ba: [29/11/2017].

Dalal, N., ve Triggs, B., 2005, Histograms of Oriented Gradients for Human Detection, *International Conference on Computer Vision and Pattern Recognition*, 2, 886–893.

Debnath, K. K., Ahmed, S. U, ve Murase, K, 2010, A paper currency recognition system using negatively correlated neural network ensemble, *Journal of Multimedia*, 5 (6), 560-567.

Deshpande, A., 2016, The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3), https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html: [29/11/2017].

Duchi, J., Hazan, E. ve Singer, Y., 2011, Adaptive Subgradient Methods for Online Learning
and Stochastic Optimization, *Journal of Machine Learning Research*, 12, 2121–2159.

Fisher, R., Perkins, S., Walker, A. ve Wolfart, E, 2000, Gaussian Smoothing, http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm: [1 Aralık 2017].

Frossard, D., 2016, VGG in TensorFlow, https://www.cs.toronto.edu/~frossard/post/vgg16/:

Gogoi, M., Ali, S. E. ve Mukherjee, S., 2015, Automatic Indian currency denomination recognition system based on artificial neural network, *2nd International Conference on Signal Processing and Integrated Networks, SPIN 2015*, 553-558.

Goodfellow, I., Bengio, Y. ve Courville, A., 2016, Deep Learning, MIT Press, p.

Group, T. H., 2016, What Is HDF5?, https://support.hdfgroup.org/HDF5/doc/H5.intro.html#Intro-WhatIs: [10/17/2017].

Gunaratna, D. A. K. S., Kodikara, N. D. ve Premaratne, H. L., 2008, ANN Based Currency Recognition System
using Compressed Gray Scale and Application

for Sri Lankan Currency Notes - SLCRec, *World Academy of Science, Engineering and Technology*

International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2.

He, K., Zhang, X., Ren, S., ve Sun, J., 2015, Deep Residual Learning for Image Recognition, *CoRR*, abs/1512.03385.

Herbert, B., Andreas E.,Tinne T. ve Luc V., 2008, Speeded-Up Robust Features (SURF) *Computer Vision and Image Understanding*, 346 - 359.

Jahangir, N. v. C., A. R. , 2007, Bangladeshi banknote recognition by neural network with axis symmetrical masks, *2007 10th international conference on computer and information technology*, 1-5.

Kamal, S., Chawla, S., Goel, N. ve Raman, B., 2015, Feature Extraction and Identification of Indian Currency Notes, *Conference: 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*.

Karakaya, F. M., Altun, H. ve Çavuşlu, M., 2009, Gerçek Zamanlı Nesne Tanıma Uygulamaları için HOG Algoritmasının FPGA Tabanlı Gömülü Sistem Uyarlaması, *Signal Processing and Communications Applications Conference, 2009. SIU 2009. IEEE 17th*.

Kaya, E., Yasar, A. ve Saritas, İ., 2016, Banknote Classification Using Artificial Neural Network Approach, *2016*, 4 (1), 4.

Kingma, D. v. B., J., 2015, Adam: a Method for Stochastic Optimization, *International Conference on Learning Representations*, 1–13.

Kriesel, D., 2007, A Brief Introduction to Neural Networks

Lecun, Y., Bottou, L., Bengio, Y. ve Haffner, P., 1998, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 2278-2324.

Lowe, G., 2004, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, 91-110.

Nielsen, M., 2015, Neural Networks and Deep Learning, Determination Press, p.

Otsu, N., 1979, A threshold selection method from gray-level histograms, *IEEE Trans. Sys., Man., Cyber*.

Pascanu, R., Montufar, G. ve Bengio, Y., 2014, On the number of linear regions of deep neural networks. Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. Montreal, Canada, MIT Press**:** 2924-2932.

Pham, T., Lee, D. ve Park, K., 2017, Multi-National Banknote Classification Based on Visible-light Line Sensor and Convolutional Neural Network, *Sensors (Basel)*.

Ruder, S., 2016, An overview of gradient descent optimization

algorithms, *CoRR*, abs/1609.04747.

Rumelhart, D., Hinton, G. ve Williams, R., 1986, Learning representations by back-propagating errors, *Nature*, 323, 533.

Sarfraz, M., 2015, An Intelligent Paper Currency Recognition System, *Procedia Computer Science*, 538-545.

Simonyan, K. v. Z., A., 2014, Very Deep Convolutional Networks for Large-Scale Image Recognition, *CoRR*, abs/1409.1556.

Solomon, C. v. B., T., 2011, Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab, Wiley, p.

Solymar, Z., Stubendek, A., Radvanyi, M. ve Karacs, K., 2011, Banknote recognition for visually impaired. ECCTD, IEEE**:** 841-844.

Song, W. v. D., S., 2009, Bank Bill Recognition Based on an Image Processing, *2009 Third International Conference on Genetic and Evolutionary Computing*, 569-573.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. ve Salakhutdinov, R., 2014, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research 15*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P.,Reed, S.,Anguelov, D., Erhan, D., Vanhoucke,V. ve Rabinovich A. , 2015a, Going Deeper with Convolutions, *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, 1–9.

Szegedy, C., Vanhoucke, V.,Ioffe, S.,Shlens, J. ve Wojna, Z., 2015b, Rethinking the Inception Architecture for Computer Vision, *CoRR*.

Tanimoto, T., 1958, An Elementary Mathematical theory of Classification and Prediction.

Taşhan, B., 2017, Road Lane Detection System With Convolutional Neural Network, *Bahçeşehir University*.

Tieleman, T. v. H., G. , 2012, Coursera: Neural Networks for Machine Learning. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude.

Walia, A., 2017, Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent, https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f: [11/11/2017].

Wu, Q. H., Zhang, Y.,Ma, Z. M.,Wang, Z. M. ve Jin, B., 2009, A Banknote Orientation Recognition Method with BP Network, *Proceedings of the 2009 Wri Global Congress on Intelligent Systems, Vol Iv*, 3-+.

X., C. B. Q. v. L. J., 2010, Currency Recognition Modeling Research Based on BP Neural Network Improved by Gene Algorithm, *2010 Second International Conference on Computer Modeling and Simulation*, 246-250.

You, X., Hu, Q.,Xu, D.,Fu, X. ve Q. Sun, 2014, Dollar bill denomination recognition algorithm based on local texture feature, *Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, 269-272.

Zeggeye, J. v. A., Y., 2016, Automatic Recognition and Counterfeit Detection of Ethiopian Paper Currency, *I.J. Image, Graphics and Signal Processing*, 2, 28-36.

Zeiler, M., 2012, ADADELTA: AN ADAPTIVE LEARNING RATE METHOD, *CoRR*.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Özgür Şahin
Uyruğu	**:**	T.C
Doğum Yeri ve Tarihi	**:**	Çorum 05/07/1989
Telefon	**:**	05542602447
Faks	**:**	
e-mail	**:**	Ozgur.sahin@aol.com

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Meram Anadolu Lisesi, Meram, Konya	2007
Üniversite	:	Yaşar Üniversitesi, Bornova, İzmir	2012
Yüksek Lisans	:	Selçuk Üniversitesi, Selçuklu, Konya	
Doktora	:		

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2012	Elpo Otomasyon	Yazılım Mühendisi
2013	Kuveyt Türk Katılım Bankası A.Ş.	Yazılım Mühendisi
2016	KT Bank AG	Uygulama Yöneticisi
2017	Kuveyt Türk Katılım Bankası A.Ş.	Kıdemli Yazılım Uz.

UZMANLIK ALANI

Mobil uygulama geliştirme ve altyapı çalışmaları

YABANCI DİLLER

İngilizce

YAYINLAR

Application of a Data Compression Algorithm in Banking and Financial Sector
ICEECAT 2014

Analysis of Parametric Methods for Effort Estimation In Financial Software Projects
AB 2014

Comparison of Data Compression Algorithms
UYMS 2014

Turkish Banknote Recognition Application For Visually Impaired
ICRES 2017