YÜZ PARAMETRELERİ YARDIMI İLE İNSANLARIN ETNİK KÖKENLERİNE GÖRE SINIFLANDIRILMASI

Ufuk Serdoğan
151402126

YÜKSEK LİSANS TEZİ

Bigisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal Güvenoğlu

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Ocak, 2020

YÜZ PARAMETRELERİ YARDIMI İLE İNSANLARIN ETNİK KÖKENLERİNE GÖRE SINIFLANDIRILMASI

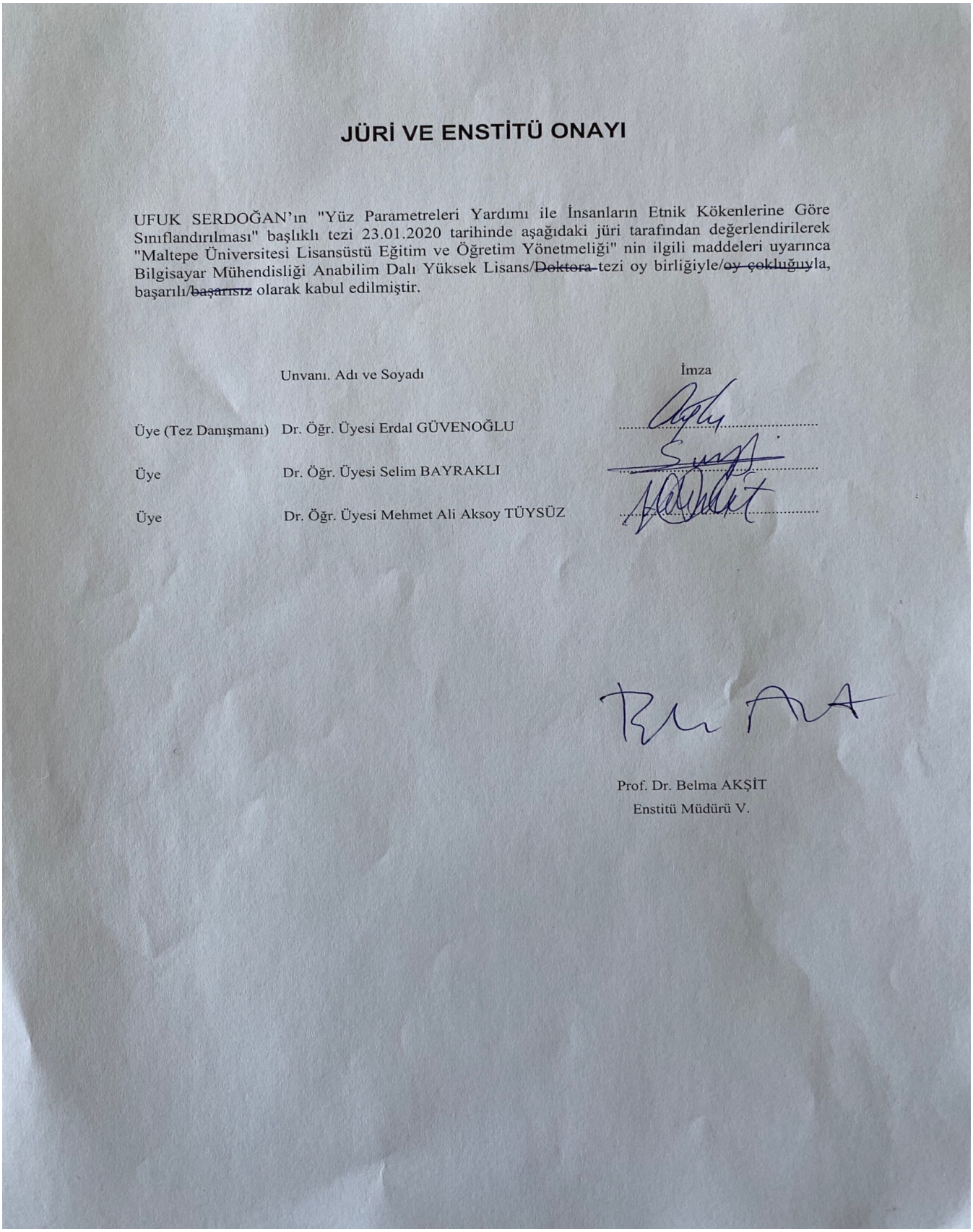
Ufuk Serdoğan
151402126
Orcid: 0000-0001-5251-9419

YÜKSEK LİSANS TEZİ
Bigisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal Güvenoğlu

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Ocak, 2020



JÜRİ VE ENSTİTÜ ONAYI



JÜRİ VE ENSTİTÜ ONAYI

UFUK SERDOĞAN'ın "Yüz Parametreleri Yardımı ile İnsanların Etnik Kökenlerine Göre Sınıflandırılması" başlıklı tezi 23.01.2020 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Maltepe Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği" nin ilgili maddeleri uyarınca Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans/~~Doktora~~ tezi oy birliğiyle/~~oy çokluğuyla~~, başarılı/~~başarısız~~ olarak kabul edilmiştir.

	Unvanı. Adı ve Soyadı	İmza
Üye (Tez Danışmanı)	Dr. Öğr. Üyesi Erdal GÜVENOĞLU	.....
Üye	Dr. Öğr. Üyesi Selim BAYRAKLI	.....
Üye	Dr. Öğr. Üyesi Mehmet Ali Aksoy TÜYSÜZ	.....

Prof. Dr. Belma AKŞİT
Enstitü Müdürü V.

ETİK İLKE VE KURALLARA UYUM BEYANI

maltepe üniversitesi	LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ ETİK İLKE VE KURALLARA UYUM BEYANI	Doküman No	FR-178
		İlk Yayın Tarihi	01.03.2018
		Revizyon Tarihi	23.01.2020
		Revizyon No	01
		Sayfa	1

23/01/2020

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bulguların sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilmeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; çalışmanın Maltepe Üniversitesinde kullanılan "bilimsel intihal tespit programı" ile tarandığını ve öngörülen standartları karşıladığını beyan ederim.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

Ufuk Serdoğan

Hazırlayan: Enstitü Sekreterliği	Onaylayan: Kalite Yönetim Koordinatörlüğü

TEŞEKKÜR

Yüksek lisans eğitimim süresince ve bu tezin hazırlanmasında bana yol gösteren, bilgi, tecrübe ve desteklerini esirgemeyen saygıdeğer danışman hocam Dr. Öğr. Üyesi Erdal Güvenoğlu' na sonsuz teşekkürlerimi sunarım. Ayrıca, tezin hazırlanmasında emeği geçen Hasan Hüseyin Gücüyener, Hüseyin Aslan ve her zaman destekleri ile yanımda olan aileme teşekkürlerimi borç bilirim.

Ufuk Serdoğan
Ocak 2020

ÖZ

YÜZ PARAMETRELERİ YARDIMI İLE İNSANLARIN ETNİK KÖKENLERİNE GÖRE SINIFLANDIRILMASI

Ufuk Serdoğan
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal Güvenoğlu
Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü, 2020

İnsanların yaşam biçimleri, etnik kökenleri, hareketlerini ve tercihlerini etkilemektedir. Teknolojinin gelişmesi ile insanların tanımlanması ve sınıflandırılması güvenlik, pazarlama, satış gibi farklı alanlarda daha çok kullanılmaya başlamıştır. İnsan yüzünün karakteristik özellikleri sınıflandırma için gerekli parametreleri barındırmaktatır. Bu parametreler ile sınıflandırma işlemi farklı teknikler kullanılarak yapılmaktadır.

Bu çalışmada derin öğrenme teknikleri kullanılarak, insanları etnik kökenlere göre sınıflandıran bir sistem yapılmıştır. Sisteme gelen fotoğraf, görüntü işleme ve derin öğrenme teknikleri ile işleme sokulup yüz tanıma, öznitelik çıkarma, hizalama, ve sınıflandırma gibi aşamalardan geçmektedir. Farklı derin öğrenme modelleri eğitilerek karşılaştırmaları yapılmış, veri seti ve model yapılarının öneminden bahsedilmiştir. Sistem etnik kökene göre sınıflandırma yaparak elde edilen sonuçların farklı sistemler ve araştırmalarda kullanılabileceğini göstermektedir.

Anahtar Sözcükler: Derin öğrenme, Etnik Köken, Sınıflandırma Görüntü İşleme, Öznitelik Çıkarma

ABSTRACT

CLASSIFICATION OF HUMANS REGARDING TO THEIR ETHNICAL ROOTS WITH FACE PARAMETERS

Ufuk Serdoğan
Master Thesis
Department of Computer Engineering
Computer Engineering Programme
Advisor: Asst. Prof. Erdal Güvenoğlu
Maltepe University Graduate School, 2020

Ethnic origins affect people's movements, preferences and lifestyles. With the development of technology, identification and classification of people has become more widely used in different fields such as security, marketing and sales. The characteristics of the human face contains the necessary parameters for classification. Classification of these parameters is done by using different techniques.

In this study, a system that categorizes people according to their ethnic background is made by deep learning techniques. The photograph is processed with image processing and deep learning techniques and passed through stages such as face recognition, feature extraction, alignment, and classification. Different deep learning models have been trained and compared, and the importance of data set and model structures has been mentioned. The system shows that the results obtained by classification according to ethnic origin can be used in different systems and researches.

Keywords: Deep Learning, Image Processing, Ethnic Origin, Classification, Feature Extraction, Ethnical Roots

İÇİNDEKİLER

JÜRİ VE ENSTİTÜ ONAYI ii
ETİK İLKE VE KURALLARA UYUM BEYANI iii
TEŞEKKÜR iv
ÖZ v
ABSTRACT vi
İÇİNDEKİLER vii
TABLOLAR LİSTESİ viii
ŞEKİLLER LİSTESİ ix
KISALTMALAR xi
ÖZGEÇMİŞ xii
BÖLÜM 1. GİRİŞ 1
BÖLÜM 2. LİTERATÜR ARAŞTIRMASI 2
2.1 Makine Öğrenmesi 4
2.1.1 Gözetimli Öğrenme 4
2.1.2 Gözetimsiz Öğrenme 5
2.1.3 Takviyeli Öğrenme 5
2.2 Yapay Sinir Ağları 6
2.3 Aktivasyon Fonksiyonları 7
2.3.1 Geri Yayılma Algoritması ve Dereceli Alçalma 7
2.3.2 Popüler Aktivasyon Fonsiyonları 9
2.4 Derin Öğrenme 15
2.4.1 Konvolüsyonel Sinir Ağları 16
2.4.2 Derin Öğrenme Modelleri 20
BÖLÜM 3. MATERYAL VE YÖNTEM 25
3.1 Python Kütüphane ve Modülleri 25
3.1.1 Python Derin Öğrenme Kütüphaneleri 26
3.2 Ön İşlemler 28
3.2.1 Histogram Eşitleme 28
3.2.2 Yüz Tanıma ve Anlamlı Verinin Alınması 30
3.3 Yöntem 31
BÖLÜM 4. SONUÇ 49
KAYNAKÇA 50

TABLOLAR LİSTESİ

Tablo 2. 1 Geri Yayılma Algoritması Değerleri....8

Tablo 3. 1 VGG-16 Model Özeti....35

Tablo 3. 2 VGG-19 Model Özeti....39

Tablo 3. 3 AlexNet Model Özeti....42

Tablo 3. 4 ZfNet Model Özeti....46

Tablo 3. 5 VS1 Kullanılmış Modellerin Karşılaştırılması....47

Tablo 3. 6 VS2 Kullanılmış Modellerin Karşılaştırılması....47

Tablo 3. 7 AlexNet Veri Setlerine Göre Başarı Oranları....48

ŞEKİLLER LİSTESİ

Şekil 2. 1 Sinir Ağları ve Nöron Yapısı 6
Şekil 2. 2 Nöron Karar Mekanizması 6
Şekil 2. 3 Dereceli Alçalma 8
Şekil 2. 4 İkili Adım Fonksiyonu [22] 9
Şekil 2. 5 İkili Adım Fonksiyonu Hata Değeri [22] 10
Şekil 2. 6 Doğrusal Aktivasyon Fonksiyonu [22] 10
Şekil 2. 7 Doğrusal Aktivasyon Fonksiyonu Hata Değeri [22] 11
Şekil 2. 8 Sigmoid Aktivasyon Fonksiyonu [22] 11
Şekil 2. 9 Sigmoid Aktivasyon Fonksiyonu Hata Değeri [22] 12
Şekil 2. 10 ReLu Aktivasyon Fonksiyonu [22] 12
Şekil 2. 11 ReLu Aktivasyon Fonksiyonu Hata Değeri [22] 13
Şekil 2. 12 Leaky ReLu Aktivasyon Fonksiyonu [22] 14
Şekil 2. 13 Leaky ReLu Aktivasyon Fonksiyonu Hata Değeri [22] 14
Şekil 2. 14 Konvolüsyonel Katmanı Filtreleme [19] 16
Şekil 2. 15 Özellik Haritası [18] 17
Şekil 2. 16 Düzleşme Katmanı Veri Yapı Değişimi [20] 18
Şekil 2. 17 Tam Bağlantılı Katman Yapısı [19] 19
Şekil 2. 18 Seyreltme Katmanı Yapısı [21] 20
Şekil 2. 19 AlexNet Model Yapısı 20
Şekil 2. 20 GoogleNet Model Yapısı 21
Şekil 2. 21 VGG Model Yapısı 22
Şekil 2. 22 ResNet Model Yapısı 23
Şekil 2. 23 ZfNet Model Yapısı 23
Şekil 2. 24 LeNet-5 Model Yapısı 24
Şekil 3. 1 TensorFlow Yapı Örneği 26
Şekil 3. 2 Caffe Yapı Örneği 27
Şekil 3. 3 Histogram Eşitleme Uygulanmış Görüntü 29
Şekil 3. 4 Histogram Eşitleme Uygulanmamış Görüntü 29
Şekil 3. 5 Histogram Eşitleme Uygulanmış Görüntü Histogramı 30

Şekil 3. 6 Histogram Eşitleme Uygulanmamış Görüntü Histogramı 30
Şekil 3. 7 Dlib Kütüphanesi İşlemleri [24].....32
Şekil 3. 8 VGG16 VS1 TOP-1 Validasyon Verisi Doğruluk Tahmini 33
Şekil 3. 9 VGG16 VS1 TOP-2 Validasyon Verisi Doğruluk Tahmini 34
Şekil 3. 10 VGG16 VS2 TOP-1 Validasyon Verisi Doğruluk Tahmini 34
Şekil 3. 11 VGG16 VS2 TOP-2 Validasyon Verisi Doğruluk Tahmini 35
Şekil 3. 12 VGG19 VS1 TOP-1 Validasyon Verisi Doğruluk Tahmini 37
Şekil 3. 13 VGG19 VS1 TOP-2 Validasyon Verisi Doğruluk Tahmini 37
Şekil 3. 14 VGG19 VS2 TOP-1 Validasyon Verisi Doğruluk Tahmini 38
Şekil 3. 15 VGG19 VS2 TOP-2 Validasyon Verisi Doğruluk Tahmini 38
Şekil 3. 16 AlexNet VS1 TOP-1 Validasyon Verisi Doğruluk Tahmini 40
Şekil 3. 17 AlexNet VS1 TOP-2 Validasyon Verisi Doğruluk Tahmini 41
Şekil 3. 18 AlexNet VS2 TOP-1 Validasyon Doğruluk Tahmini 41
Şekil 3. 19 AlexNet VS2 TOP-2 Validasyon Doğruluk Tahmini 42
Şekil 3. 20 ZfNet VS1 TOP-1 Validasyon Verisi Doğruluk Tahmini 44
Şekil 3. 21 ZfNet VS1 TOP-2 Validasyon Verisi Doğruluk Tahmini 44
Şekil 3. 22 ZfNet VS2 TOP-1 Validasyon Verisi Doğruluk Tahmini 45
Şekil 3. 23 ZfNet VS2 TOP-2 Validasyon Verisi Doğruluk Tahmini 45
Şekil 3. 24 AlexNet VS1 ve VS2 Karşılaştırması 48

KISALTMALAR

DL : Derin Öğrenme (Deep Learning)

IP : Görüntü İşleme (Image Processing)

AI : Yapay Zeka

ML : Makine Öğrenmesi

CNN : Evrimel Sinir Ağları(Convolutional Neural Network)

ReLu : Düzleştirilmiş Doğrusal Birim Katmanı(ReLu)

GPU : Grafik İşlem Birimi (Graphics Processing Unit)

CPU : Merkezi İşlemci Birimi (Central Proccesing Unit)

VS1 : Veri Seti 1

VS2 : Veri Seti 2

ÖZGEÇMİŞ

Ufuk Serdoğan

Bilgisayar Mühendisliği Anabilim Dalı

Eğitim

Derece	*Yıl*	*Üniversite, Enstitü, Anabilim*
Ls.	2014	Doğuş Üniversitesi, Mühendislik Fakültesi Bilgisayar Mühendisliği
Lise	2005	Anadolu Lisesi, Turhal / Tokat

İş/İstihdam

Yıl	*Görev*
2019 -	Tech Lead, Multinet Inventiv
2017-2019	Yazılım Mühendisi, Multinet Inventiv
2016-2017	Yazılım Mühendisi, SecretCV
2012-2016	Freelance Yazılım Mühendisi

Kişisel Bilgiler

Doğum yeri ve yılı : Turhal/Tokat, 1987 Cinsiyet: E

Yabancı diller : İngilice (İyi)

GSM / e-posta : 0 555 465 65 26 / ufukserdogan@gmail.com

BÖLÜM 1. GİRİŞ

Yüz tanıma sistemleri uzun zamandır teknoloji firmalarının ve akademik çalışmaların odak noktasındadır. Yıllar içerisinde bir çok farklı teknik kullanılmış sistemler geliştirilmiştir. Teknolojinin ilerlemesi, insan tanıma performanslarının artması bu sistemleri daha popüler hale getirmiştir. Örneğin, akıllı telefonlar veya dizüstü bilgisayarlar gibi yaygın aygıtlarda, kimlik doğrulama için yüz tanıma uygulayarak parola, kaydırma deseni gibi geleneksel tanıma teknolojilerinin yerine yeni sistemler entegre edildi. Günümüzde kullanılan gelişmiş güvenlik kamerası sistemlerinde izinli ve izinsiz girişleri takip etmek için yüz tanıma sistemleri kullanılmaktadır. [1] Yüz tanıma uygulamaları kimlik yönetiminin ötesinde, kişisel veri toplama kapasitesi sayesinde bir resimden yaş, ırk, cinsiyet gibi bilgileri toplama fırsatı sunar.
Irk, aynı türe ait bireylerin oluşturduğu topluluklardır. Sadece aynı türe ait olmalarının yanında, bireyleri belli başlı morfolojik ve genetik özellikleri ortak olarak birbirinde barındırabilirler. Örnek vermek gerekirse Afrika'da yaşayan bireylerin ten renklerinin siyah, Asya'da yaşayan bireylerin gözlerinin çekik olması gibi belli başlı ön plana çıkan keskin ortak özellikleri vardır. Yüz tanıma sistemleri topladığı kişisel ve ortak özellikler sayesinde reklam verenlere pazarlama kampanyalarını kişiselleştirme olanağı sağlar. Özellikle cinsiyet ve ırk üzerinden kişiselleştirilmiş bilgiler reklam, pazarlama, güvenlik gibi alanlarda sıkça kullanılmaktadır.

Bu çalışmanın amacı son teknolojik gelişmeler ışığında en yeni teknikler kullanılarak insan fotoğrafları üzerinden cinsiyet ve ırk sınıflandırması yapan bir sistem oluşturmak, kullanılan veri setinin ve modelin sonuca etkisini belirleyerek yapılmış ve ileride yapılacak çalışmalara yardımcı olmaktır.

Bu çalışmada giriş bölümünden sonra bölüm 2 ‘de literatür araştırmalarına yer verilmiş, diğer çalışmalar ve araştırmalardan bahsedilmiştir. Bölüm 3 ‘te çalışmada kullanılan materyaller ve yöntemlerden, bölüm 4 ‘te ise sonuçlar yorumlanmış ve çıktıdan bahsedilmiştir.

BÖLÜM 2. LİTERATÜR ARAŞTIRMASI

Yüz tanıma sistemleri göz, kaş, ağız, burun, dudak, çene şekli gibi karakteristik özellikleri kullanmaktadır. Biçim, konum, birbirleri ile olan mesafeler ayırt edici özelliklerdir. Yaklaşık 50 yıldır yüz tanıma üzerine çalışmalar yapılmaktadır. Bu alanda yapılan ilk çalışma Takeo Kanade tarafında 1973 yılında yapılmıştır. [2]O dönemde donanımların ve yüz veri tabanlarının yetersiz olması gibi sebeplerden dolayı çalışma istenilen seviyelere gelememiştir. İlerleyen yıllarda farklı çalışmalar yapılmış çok sayıda yeni algoritmalar ortaya çıkmıştır.

Geliştirilen ilk sistemlerdeki amaç bilgisayara insan yüzlerini nasıl tanıyacağını öğretmekti. Zamanla insan yüzünü tanımlamanın zorluğunun farkına varıldı. Yüz tanımanın başlıca zorluklarına örnek vermek gerekirse insan yüzünde oluşan yaşlanmaya bağlı kırışıklıklar, kaza veya farklı olaylara bağlı yüzde oluşan izler, fotoğrafın açısı veya renk tonlarının farklı olması gibi faktörler farklı sonuçlar çıkmasına sebep olur.

Arabnia ve Jafri 2009 yılında yüz tanıma teknikleri ve metodolojileri üzerine o yıla kadarki en kapsamlı literatür araştırmasını yayınladılar. [3] 2000'lerin sonlarına kadar, yüz tanıma yöntemlerini bütünsel eşleştirme yöntemleri ve özellik tabanlı yöntemler olarak ayırabiliriz.

Özellik tabanlı yaklaşımlar yoğunluk verilerine dayananır. Giriş görüntüsünün yüzündeki karakteristik yüz özelliklerini belirleyip ölçerek çalışır. [3] Bu yüz özellikleri burun, ağız, göz vb dir. Sonrasında bu özellikler basitleştirilerek aralarındaki geometrik ilişkilere bağlı olarak veri tabanındaki verileri besler. Yüz tanımayla ilgili 1973 yılında Kanade [2] tarafından geliştirilen örnekte 16 yüz parametresinden oluşan bir vektörü çıkarmak için basit görüntü işleme teknikleri kullanmıştır. Bu parametreler mesafelere, alanlara ve açılara bağlı olarak belirlenir. Kırk kişilik bir veri setinde %75 gibi bir başarı oranı elde edilmiştir. Sonrasında farklı çalışmalar ile veri setinin daha fazla olmasına bağlı bu oran %90 ‘a çıkarılmıştır.

Zamanla sofistike özellik tabanlı yöntemler geliştirildi. Bu teknikler arasında Hough dönüşümü [4], Graf'in filtreleme ve morfolojik işlemleri [5] ve Reisfeld'in simetri operatörü [6] bulunmaktadır. Bu yöntemler önceki çalışmalara göre daha performanslı sonuçlar vermiştir.

1996 yılında Cox ve diğerleri daha önceki sistemlere göre büyük bir veri seti kullanan, özellik tabanlı yaklaşıma dayanan bir yüz tanıma sistemi geliştirmiştir. 658 görüntüden bir veri kümesi kullandılar ve geometrik vektörü oluşturmak için 35 yüz özelliğini manuel olarak çıkardılar ve %95 doğruluk elde ettiler. Bu sistemle ilgili sorun, manuel özellik çıkarmasıydı, özellik çıkarma aşaması otomatikleştirilirse doğruluğun azaltılacağı ve sistemin daha az hassas olduğu varsayılmaktadır [7].

İlerleyen yıllarda birçok farklı özellik tabanlı yöntem geliştirildi. Bunlardan öne çıkan Elastik demet grafik eşleştirme yöntemi ile 1997 yılında daha iyi sonuçlar elde edildi. [8] Tüm bunlara rağmen özellik tabanlı yöntemler için temel problemler aynı kalmıştır. Yüz özelliklerinin manuel çıkarılması gerekir ve bu nedenle hangi özelliklerin daha önemli olduğunun tespiti uygulayıcının alması gereken bir aksiyondur. Manuel yapılan bu işlem sonucunda her uygulayıcı farklı özelliklere yoğunlaşabilir ve sonuçlar farklı olacaktır. Sistem performansından önemli ölçüde ödün vermeden özellik çıkarma aşamasını otomatikleştirmek gerçekten zordur.

Bütünsel temelli yaklaşımlar daha önceki çalışmalarda yapılan yüzün yerel özelliklerine odaklanmak yaklaşımı yerine tüm görüntüdeki yüzleri tanımlamaya çalışmak için geliştirilmiştir. İstatistiksel ve yapay zeka yaklaşımları olarak sınıflandırılır. İstatistiksel yaklaşımlar bir görüntüyü 2D yoğunluk değerleri dizisine dönüştürür ve daha sonra bu yoğunluk değerlerini doğrudan veri kümesindeki görüntülerle karşılaştırır. Bu yöntemler kontrollü koşullar altında çok iyi çalışır. [9] Yüksek boyutlarda işlem yapmak gerektiğinde hesaplama maliyeti yüksektir. Ayrıca yüz bakış açışı, arka plan bulanıklığı, renk tonlarında değişim, gürültü gibi farklı etkenler ile test edildiğinde performans düşüklüğü gözlenmiştir. [10]

İstatistiksel yöntemin boyutlarını ekonomik bir seviyeye indirmek için 1987 ’de Sirovich ve Kirby [11] PCA [12] [13] yöntemini kullandılar. PCA ‘nın eigen resimleri

yüzün, eigen resimlerinin bir kombinasyonu, her bir eigen resmi boyunca izdüşümü ile kolayca yeniden oluşturulabileceğini ve koordinat alanı boyunca herhangi bir yüzü etkili bir şekilde temsil edebileceğini gösterdiler.

1991 yılında Turk ve Pentland [14] [15], Sirovich ve Kirby'nin çalışmalarına dayanan bir yüz tanıma sistemi geliştirdi. Yüzlerin resimler boyunca izdüşümleri sınıflandırma özellikleri olarak kullanarak tanınabilmesi idi. Yüz tanıma sistemleri başlangıçta 2.500 görüntü veri kümesi kullanılarak test edildi ve en yüksek performansa ulaşıldı. Sistem farklı ölçeklerde kötü performans göstermiştir. Zaman içerinde daha büyük veri kümesi arttılarak performans da bir iyileşme gördü.

1994 yılında Doğrusal Diskriminant Analizi (LDA) projeksiyonu öncesinde alt uzay projeksiyonuna dayanan Fisherfaces adlı bir yöntem geliştirildi ve bu yöntemin PCA'dan daha iyi olduğunu bildirildi. [16]

Önceki çalışmaların çoğunlukla görüntü işleme(IP) üzerine olmasına karşılık günümüzde ağırlıklı olarak yapay zeka(AI), makine öğrenmesi(ML) ve derin öğrenme(DL) üzerinedir.

2.1 Makine Öğrenmesi

Makine öğrenmesi belli başlı algoritma ve istatistiksel modellerin kullanılarak bilgisayar sistemlerinin kendi başlarına aksiyon almalarını sağlama eylemidir. Bilgisayar sistemi bu aksiyonları bazı çıkarımlar yaparak veya bazı kalıp girdileri tanıyarak eylemlerle eşleştirir.

2.1.1 Gözetimli Öğrenme

Gözetimli öğrenme eğitim verisi olarak adlandırılan bir veri kümesinin üzerine matematiksel model oluşturma işlemidir. Gözetimli öğrenmede eğitim verisi içerisindeki her bir girdiye karşılık gelen, beklentiyi karşılayan bir çıktı bulunur. Burada girdileri çıktılara eşleyen fonksiyon matematiksel modelin algoritma karşılığıdır. Eğitme işleminin yapılacağı makine her bir girdi için çıktıyı alır ve bu çıktıdan doğrulama işlemi

yaparak hata payını hesaplar. Hata payı, hata düzeltme işleminde kullanılacak algoritmaya verilir ve model üzerinde gerekli düzenlemeler yapılır. Bu işlem eğitim verisindeki tüm girdiler için tekrarlanır ve istatiksel modelin başarı ölçüsü kabul edilebilir düzeyde ise model tamamlanmış demektir.

2.1.2 Gözetimsiz Öğrenme

Gözetimsiz öğrenme bilgisayar sisteminin eğitim datası olmadan çıkarımlar yapmasına dayanan bir makine öğrenmesi türüdür. Bu türde model önce birbirine benzeyen girdileri gruplar ve daha sonra bu gruplanmış veri türünün dışına çıkan girdilerde veya bu veri türüne benzer girdilerde aksiyon alır.

Gözetimsiz öğrenmenin zorlukları;

- Gözetimsiz öğrenmede örnek bir veri kümesi ve öğrenme yöntemi olmadığı için büyük veride daha yavaş çalışmaktadır.
- Ortaya çıkan modelin doğru kararlar verdiğini deneme ihtiyacı bulunmakta.

Zorluklara rağmen gözetimsiz öğrenmenin tercih sebepleri;

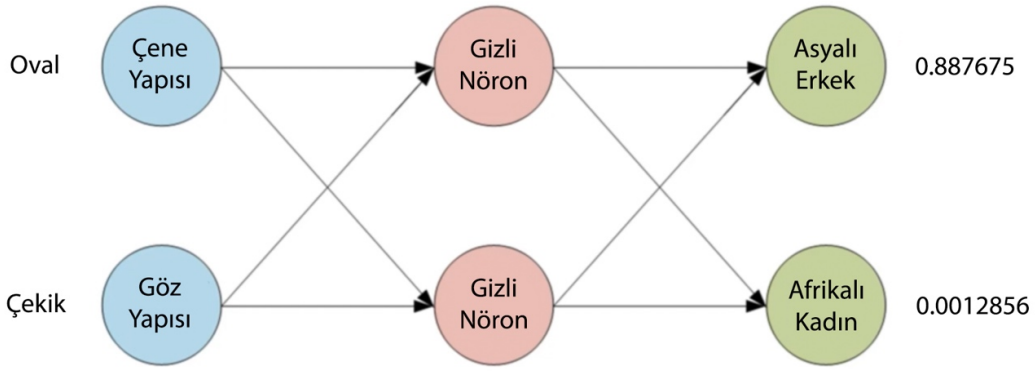
- Büyük veride etiketleme ve gruplama işleri çok maliyetli.
- Öbekleme yöntemiyle kullanılan veriye yeni bakış açıları kazandırmak.
- Üzerinde çalışılan verinin gerçek hayatta kaç farklı çıktı çeşidi verebileceğini kestirememek.

2.1.3 Takviyeli Öğrenme

Takviyeli öğrenme bir bilgisayar sisteminin bir ortamda nasıl davranması gerektiğini kontrol etmektedir. Takviyeli öğrenme ile çalışırken genellikle ödül ve ceza sistemi kullanılır. Bu sistemde öğrenme işlemi yapılırken bilgisayar sistemi tanımlanan aksiyonlardan rastgele denemeler yapar ve aldığı ödül puanının değerine göre kendi matematiksel modelini kurar. Takviyeli öğrenme özellikle oyun teorisi alanında büyük ses getirmiştir.

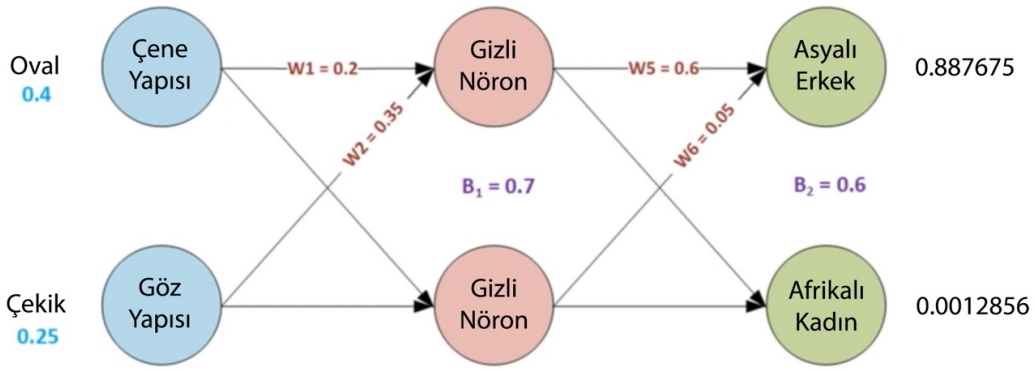
2.2 Yapay Sinir Ağları

Yapay sinir ağları insan beyni baz alınarak oluşturulmuş ufak algoritmalardır. Bir sinir ağı girdi olarak aldığı sayıları mekanik bir sezgi, etiket veya kümeleme yöntemiyle çıktılara dönüştürür. Sinir ağlarının resimleri, sesi ve yazıları tanıyabilmesi için bu verilerin dönüştürülmesi gerekmektedir



Şekil 2. 1 Sinir Ağları ve Nöron Yapısı

Şekil 2.1 bir sinir ağına örnektir. Bu ağ girdi olarak resimleri almaktadır ve mavi ile işaretlenmiş nöronlar çene yapısını ve göz yapısını ayırt edecek şekilde eğitilmiştir. Kırmızı ile gösterilen ara katman nöronları, resimden gelen veri yerine resimde bulunan özellikleri almaktadır. Örneğin bir nöron çene yapısından ve göz yapısından çıkan değerleri alıp belli bir oranda karıştırarak yüz tipi verisi oluşturabilir. Asyalı erkek için oran belirten nöron da ten rengini ve yüz tipi verisini alıp resimdeki kişinin bir asyalı erkek olma ihtimalini ortaya koyar.



Şekil 2. 2 Nöron Karar Mekanizması

Şekil 2.2 ‘de çene yapısı bir gizli nöronu %20 ve göz yapısı bu nöronu %35 etkileyebilir. Bu şekilde gizli nöron kendi kararını verebilir. Kendisine gelen verinin ırk bilgisine karar veren yeşil nöron yine aynı şekilde ağırlıklar ile gizli nöron gibi karar vererek çıktı üretir.

$$GN_{1\text{çıkt}ı} = k_1 w_1 + k_2 w_2 + b_1 \tag{2.1}$$

Örnek bir nöron çıktı hesaplaması denklem 2.1 ‘de verilmiştir. B(bias) değeri nöronun sapma değeridir ve çıktıya eklenir.

$$GN_{1\text{çıkt}ı} = 0.4 * 0.2 + 0.25 * 0.35 + 0.7 = 0.8675 \tag{2.2}$$

Denklem 2.2 ‘de bir nöronun çıktısı hesaplanmıştır.

2.3 Aktivasyon Fonksiyonları

Bir nöronun girdisi ve çıktısı bellidir ve bu veriler arasında doğrusal bir ilişki vardır. Nöronları saf haliyle ele alırsak doğrusal bir denklemle çözülebilen regresyon problemine dönüşürler. Derin öğrenme ise doğrusal olmayan problemlerde uzmanlaşmıştır. Bu haliyle nöronları derin öğrenmede kullanabilmek için aktivasyon problemlerinden geçirerek doğrusallıklarını bozmaktayız. Bu şekilde nöronun kapasitesini arttırıp daha çok bilgiyi saklamasını sağlarız.

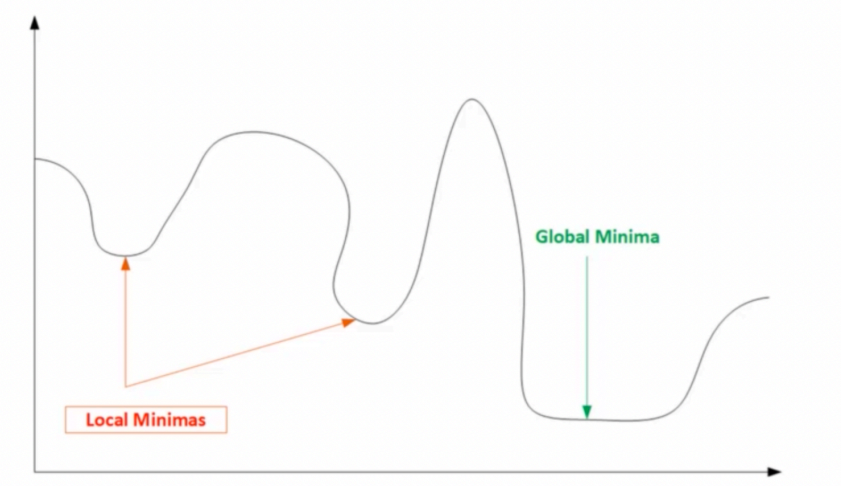
2.3.1 Geri Yayılma Algoritması ve Dereceli Alçalma

Bir sinir ağı eğitilirken ağırlıkların yeni gelen değerlere göre eğitilebilmesi gereklidir. Bu eğitim için geri yayılma algoritması kullanılır. Geri yayılma algoritması hata ölçüm fonksiyonunun en az değerini arar ve bulduğu değerle nöronu optimize eder. Geri yayılma algoritması dereceli alçalma (gradient descent) tekniğini kullanır.

Tablo 2. 1 Geri Yayılma Algoritması Değerleri

Girdi	Çıktı(Ağırlık: 3)	Beklenen	Hata	Çıktı(Ağırlık: 4)	Hata
3	7	5	2	6	1
7	16	12	4	14	2

Tablodaki nöronun eğitiminde önce rastgele olarak ağırlığı 3 veririz. Bu ağırlıktaki hata değerini not edip ağırlığı arttırırız. Hata değeri azaldığı için arttırmaya devam edip daha da düşürebilir miyiz diye bakarız. Eğer hata değeri azalmayıp artsaydı ağırlığı arttırmak yerine düşürme yoluna giderdik. Bu tür ağırlık arttırıp azaltma yönteminde adım miktarı için kullanılan terim "learning rate"tir. Bu değer fazla olursa minimum hata değerini kaçırıp yanlış yöne ilerleyebiliriz. Bu değer az olursa da minimuma ulaşmak için gereğinden fazla eğitim adımı yapmak gerekebilir.



Şekil 2. 3 Dereceli Alçalma

Şekil 2.3 'de görüldüğü gibi minimum değerlerine ulaştığımızda bu değerin lokal minimum mu yoksa global minimum mu olduğunu anlayabilmemiz için elimizdeki değerlere bakmak için kullandığımız yöntem dereceli alçalma tekniğinde gizlidir.

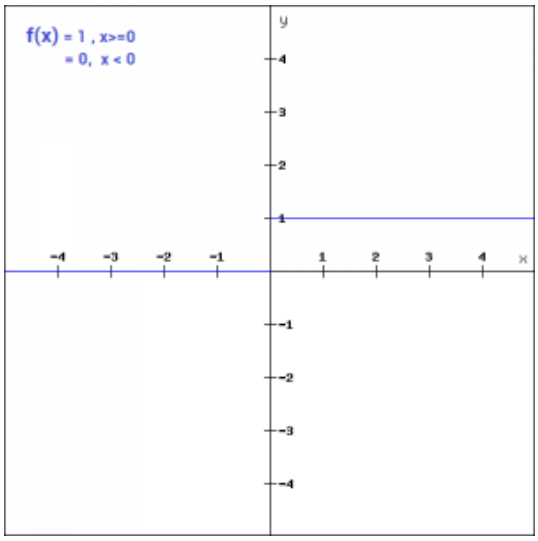
Dereceli alçalma için 3 ana yöntem vardır:

1. **Saf Dereceli Alçalma (Naive Gradient Descent)** : Bu yöntemde en düşük hata değerini bulabilmek için bütün veri setinin üzerinden geçilir. Küçük veriler için uygun olsa da büyük veriler için aşırı pahalı bir yöntem olabilir.

2. **Stokastik Dereceli Alçalma :** Bu yöntem ağırlık değerlerini her veriden sonra değiştirir.
3. **Küçük Gruplu Dereceli Alçalma :** Bu yöntemde ağırlık değerleri her bir grup işlendikten sonra değiştirilir. Gruptaki veri sayısı değiştirilebilir bir değerdir.

2.3.2 Popüler Aktivasyon Fonsiyonları

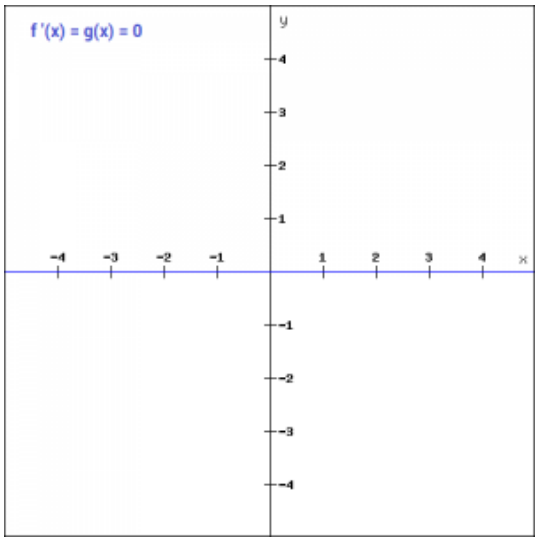
İkili Adım Fonksiyonu : Basit bir şekilde aktivasyon fonksiyonu oluşturmak için akla gelen ilk isimdir. Belli bir aralığın üstünü pozitif, altını ise negatif olarak ele alır.



Şekil 2. 4 İkili Adım Fonksiyonu [23]

Bu aktivasyon fonksiyonu kümelendirme işlemlerinde kullanılabilmekte. Örneğin bir soruya cevap vermek istiyor olabiliriz. Evet, hayır, doğru veya yanlış tarzı cevapların aktivasyon fonksiyonu ikili adım fonksiyonu olabilir.

İkili adım fonksiyonunun alçalma fonksiyonu bir sayının türevi 0 olduğundan sabit bir sayı olan 0'dır.



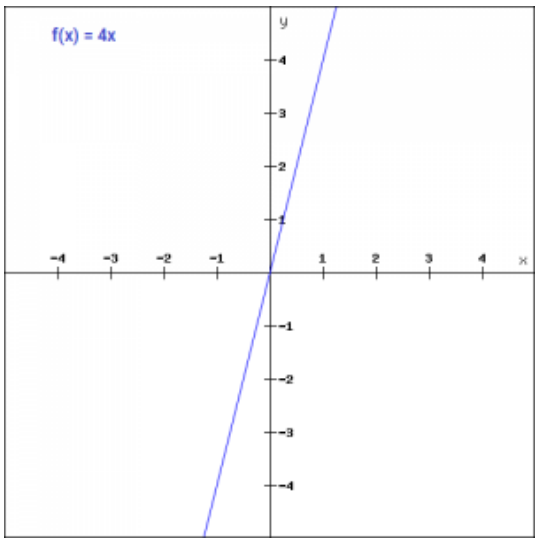
Şekil 2. 5 İkili Adım Fonksiyonu Hata Değeri [23]

Şekil 2.5'de görülebileceği üzere bir nöron eğitilirken ikili adım fonksiyonunu kullandığımızda hata değeri her zaman 0 olacağından eğitme işlemi yapılmayacaktır.

Doğrusal (Lineer) Aktivasyon Fonksiyonu : İkili adım fonksiyonunun yetmediği yerleri çözmek için doğrusal aktivasyon fonksiyonu kullanılmıştır.

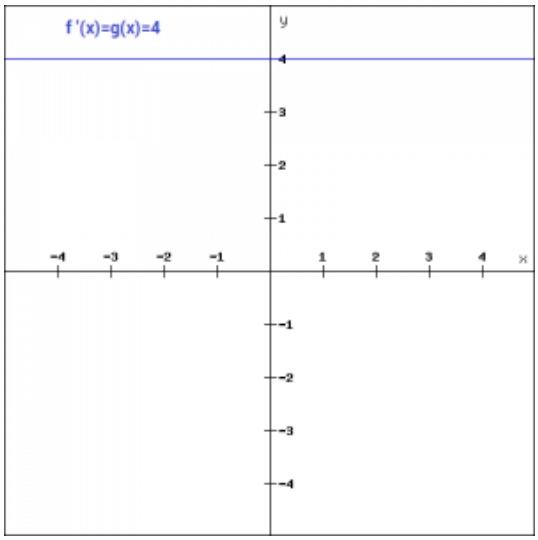
$$f(x) = ax + b \tag{2.3}$$

Bu fonksiyonda nöronlar denklem 2.3 'de görüldüğü gibi bir formülle aktive edilirler.



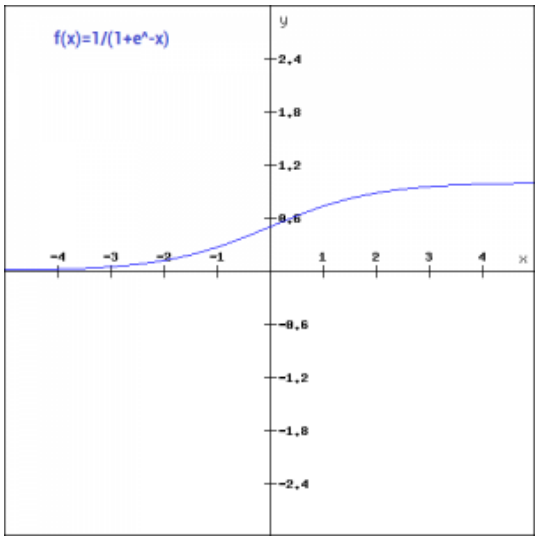
Şekil 2. 6 Doğrusal Aktivasyon Fonksiyonu [23]

Doğrusal fonksiyonların hata fonksiyonu, türevi alındığında sabit bir sayı çıktığı için nöron eğitmede kullanılamaz. Hata sabit bir değer çıkacağından dolayı doğru nöronlar da hatalı gözükecektir.



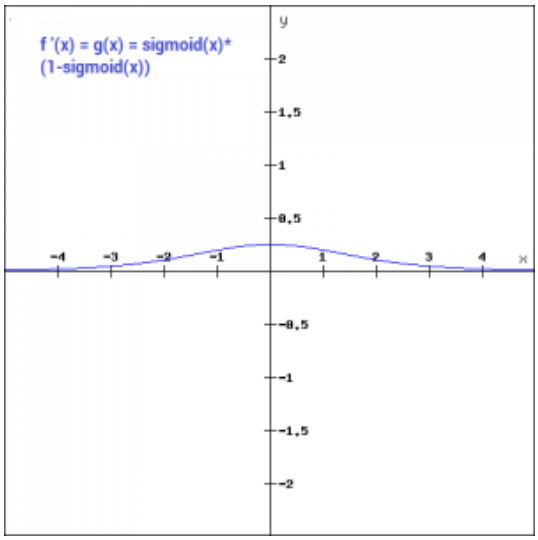
Şekil 2. 7 Doğrusal Aktivasyon Fonksiyonu Hata Değeri [23]

Sigmoid Aktivasyon Fonksiyonu : Doğrusal aktivasyon fonksiyonu geri yayılma için uygun olmadığından yerine daha sonra Sigmoid fonksiyonu gelmiştir. Sigmoid fonksiyonunun görünüşü aşağıdaki gibidir



Şekil 2. 8 Sigmoid Aktivasyon Fonksiyonu [23]

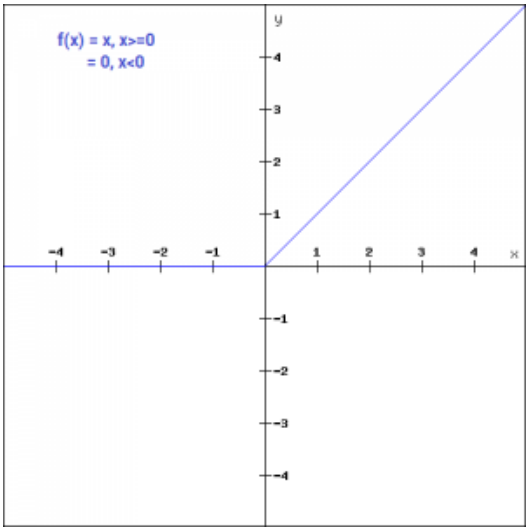
Bu fonksiyonun en büyük avantajı doğrusal olmamasıdır. Derin öğrenme için doğrusal sonraki en uygun fonksiyondur. Günümüzde yaygın olarak kullanılmasına rağmen Sigmoid'in düzeltilmesi gereken pürüzleri bulunmaktadır. Özellikle +3 ve -3 değerlerinin sonrasının etkisiz olması bu pürüzlerden bir örnek olarak verilebilir.



Şekil 2. 9 Sigmoid Aktivasyon Fonksiyonu Hata Değeri [23]

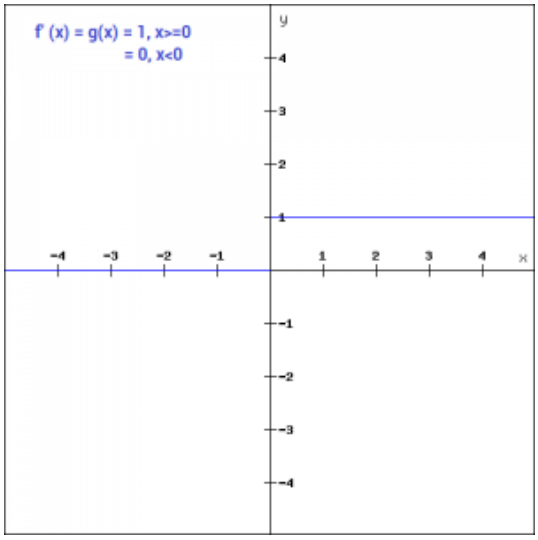
Sigmoid fonksiyonunun hata fonksiyonuna baktığımızda -3 ve +3 aralığının dışındaki değerler için sıfıra yaklaştığını ve öğrenme işleminin gerçekleşmeyeceğini söyleyebiliriz.

ReLu Aktivasyon Fonksiyonu : Relu fonksiyonu f(x) = max(0, x) formülünü barındırır. Grafiksel gösterimi de aşağıdaki gibidir:



Şekil 2. 10 ReLu Aktivasyon Fonksiyonu [23]

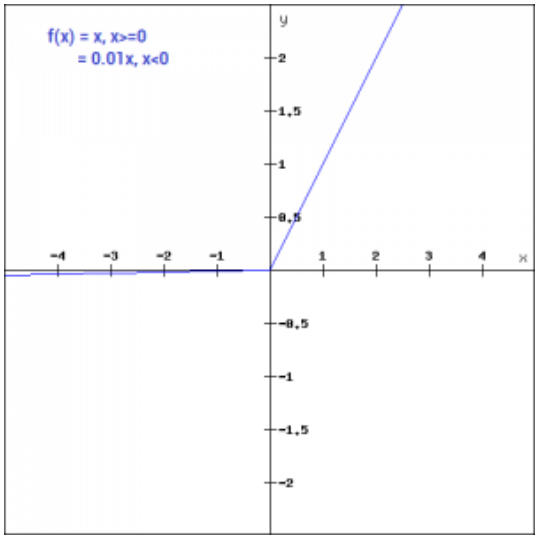
ReLU günümüzde en çok kullanılan aktivasyon fonksiyonudur. En önemli artısı nöronun çıktısının negatif değerlere düştüğünde aktifleştirilmemesidir. Böylece sinir ağındaki nöronların bir miktarı aktif olarak çıktıya etki eder ve daha başarılı bir sinir ağına sahip oluruz.



Şekil 2. 11 ReLu Aktivasyon Fonksiyonu Hata Değeri [23]

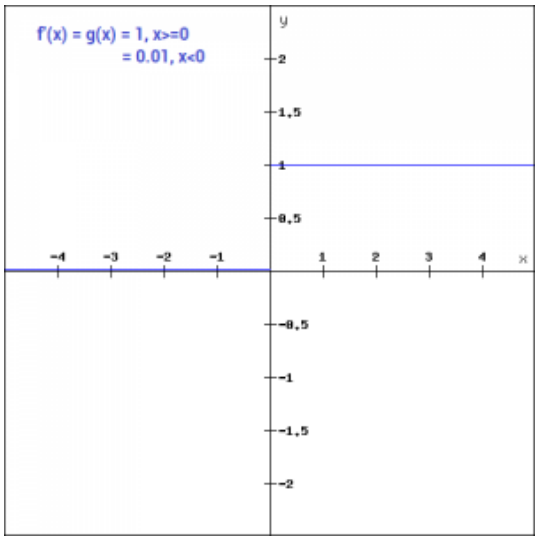
Hata fonksiyonuna baktığımızda ise ReLU kullanıldığında 0 ve aşağısındaki değerler için nöronların öğrenme işlemini yapamadıklarını görmekteyiz

Leaky ReLu Aktivasyon Fonksiyonu : Leaky ReLU normal ReLU fonksiyonunun geliştirilmiş halidir. ReLU'da gördüğümüz üzere 0 altındaki değerler için ReLU'da eğitim gerçekleşmemektedir. Leaky ReLU ile 0'dan küçük değerler için çıktı sıfırlanmak yerine 0,01 gibi bir değerle çarpılır.



Şekil 2. 12 Leaky ReLu Aktivasyon Fonksiyonu [23]

Negatif değerler için düz 0 değerler alınmayarak hata fonksiyonunun işlevselliği arttırılmakta.



Şekil 2. 13 Leaky ReLu Aktivasyon Fonksiyonu Hata Değeri [23]

Leaky ReLU ile çözüme ulaşılamadığı durumlarda parametreli ReLU fonksiyonu kullanılabilir. Bu fonksiyonda 0,01 değeri de eğitim verisiyle değişip daha fazla olasılığa ve daha yüksek başarı oranlarına imkan vermektedir.

SoftMax Aktivasyon Fonksiyonu : Softmax kümeleme işlemlerinde kullanılmaktadır. Daha önce incelediğimiz sınıflandırma aktivasyon fonksiyonu ikili sonuç veren bir fonksiyondu. Daha fazla durumu içeren sınıflandırmalar için ise Softmax fonksiyonu kullanılmaktadır.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K. \tag{2.4}$$

Softmax fonksiyonu tüm çıktıları toplamlarına bölerek normalleştirme işlemini de yapmaktadır. Örneğin, bir softmax fonksiyonuyla 3 kategori çıktısı alabiliriz ve bu değerlerin (1.5, 1, 2.5) olduğu durumu düşünürsek, çıktımız (0.3, 0.2, 0.5) olacaktır. Aktivasyon fonksiyonu seçilirken uyulması gereken katı kurallar bulunmamaktadır. Ancak bazı inceliklere dikkat edilirse daha iyi sonuçlar elde edilebilir.

1. ReLU fonksiyonu genel bir aktivasyon fonksiyonudur ve en iyi başarıyı gizli katmanlarda vermektedir
2. Eğer sinir ağında pasif nöronlarla karşılaşırsak Leaky ReLU kullanmak daha iyi performans verecektir.

2.4 Derin Öğrenme

Derin öğrenme yapay zekanın bir alt dalı olup, makine öğrenmesinin yöntemlerinden biridir. Yapay sinir ağları yöntemiyle desteklenip daha doğru sonuçlar üretilebilir. Genel olarak derin öğrenmede ayırt edici özellikler kullanılarak katmanlı bir yapı oluşturulup eğitim süresince modellerin kendi kendine oluşması sağlanır.Genel olarak derin öğrenmede kullanılan veriler düzenlenmemiş veriler, makine öğrenmesinde ise düzenlenmiş veriler kullanılır. İlk olarak 1956 yılında John McCarthy tarafından yapay zeka kavramı ortaya çıkmasına rağmen temelleri 1950 yılında Alan Turing tarafından ortaya atılmıştır. Yapay sinir ağları beynin çalışma mantığı göz önüne alınarak ortaya çıkmıştır. İlk olarak 1957 yılında Frank Rosenblatt tarafından ortaya atılmıştır.

Derin öğrenme 1986 yılında Rina Dechter tarafından kullanılmıştır. Fakat o yıllardaki donanım tam olarak ihtiyacı karşılamadığı için başarılı sonuçlar elde edilememiştir. Günümüzdeki teknolojinin ilerlemesi ve veri miktarındaki artışa bağlı

olarak derin öğrenme artık önemli bir konuma gelmiştir. Görüntü işleme(IP) gibi geleneksel yöntemler ile derin öğrenmeyi karşılaştıracak olursak;

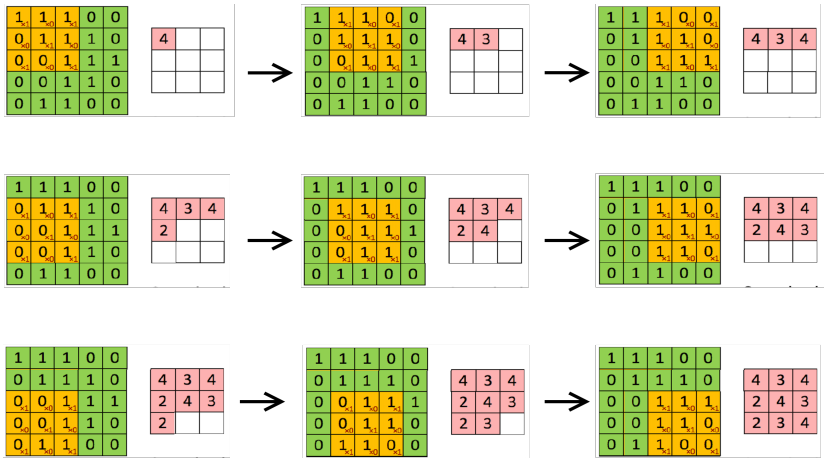
Veri miktarının az olduğu durumlarda geleneksel yöntemlerin daha başarılı olduğu gözlemlenmiştir. Fakat veri miktarı arttıkça derin öğrenmenin performansının geleneksel yöntemlerin performansına göre oldukça hızlı arttığı gözlemlenmiştir.Bu artış donanımdaki gelişmeler ile doğru orantılıdır.

Derin öğrenme günümüzde verilerin çokluğu göz önüne alındığında en popüler yaklaşımlardan biri haline gelmiştir. Nesne ve yüz tanımlama, otonom gelişmeler, doğal dil işleme, konuşma ve ses tanıma gibi alanlarda sıklıkla karşımıza çıkmaktadır.

2.4.1 Konvolüsyonel Sinir Ağları

Derin öğrenmede kullanılan sinir ağları insan vücüdun da bulunan sinir ağlarının çalışma şeklini taklit etmektedir. Yapay bir sinir ağı yapısı vardır. CNN sınıflandırma için standart sinir ağı kullanır, özellik tespit etmek, belirleme yapmak gibi işlemler için farklı katmanlar kullanır.

Konvolüsyonel Katman (Convolutional Layer) : CNN 'nin ana katmanıdır, özelliklerin algılanmasından sorumludur. Katmanda, görüntüdeki yüksek ve düşük seviyeli özellikleri çıkarmak için resme çeşitli filtreler uygulanır. Filtreler genelde çok boyutludur ve piksel değerleri içerirler.
5x5 boyutunda bir görüntüye 3x3 filtre uygulanacak olursa ;



Şekil 2. 14 Konvolüsyonel Katmanı Filtreleme [20]

Filtre görüntünün sol üst köşesine konumlandırılır ve iki matris arasında indisler birbirleri ile çarpılır ve sonuçlar toplanır, çıkan sonuç çıktı matrisine yazılır. Daha sonra filtre 1 piksel sağa haraket ettirerek işlem tekrarlanır. Satır bitiminde alt satıra geçilir ve tüm görüntü üzerinde filtre gezdirilir. Bu şekilde çıktı 3x3 çıktı matrisi oluşturulur. Matris boyutunun 3x3 olmasının sebebi 5x5 görüntü matrisinin üzerinde filtre olarak 3x3 boyutunda matris yatayda ve dikeyde 3 kez hareket etmesinden kaynaklanır. Resmin boyutu ve uygulanan filtrenin boyutu çıktı matrisini etkiler.

Çıktı matrisine özellik haritası (Feature Map) denir. Filtre görüntü üzerinde gezdirilerek elde edilen sonuçlar ile ana görüntü üzerinden özellikler tespit etmiş olur. Görüntü üzerinde birden çok özellik bulunduğu için bu özellikleri tespit edebilmek adına birden fazla filtre kullanılır, yani bir CNN ‘de birden fazla katman bulunur.



Şekil 2. 15 Özellik Haritası [19]

Ana görsel üzerinde bir filtre uygulandığında, bir özellik haritası oluşturuluyor ve bu şekilde bir özellik tespit edilmiş oluyor. Daha sonra, ikinci bir filtre uygulanarak başka bir özellik haritası oluşturulur. Bu şekilde devam ederek CNN de oluşturulan katmanlara göre yeni özellik haritaları ortaya çıkar.

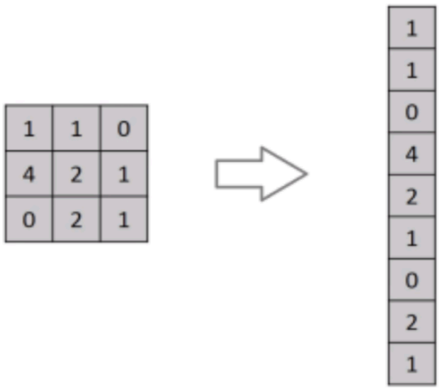
Düzleştirilmiş Doğrusal Birim Katmanı (ReLu) : Konvolüsyonel katmanlardan sonra genellikle ReLu katmanı gelir. Aktivasyon katmanı olarakta adlandırılır çünki ReLu bir aktivasyon fonksiyonudur. Sigmoid ve tahn gibi aktivasyon fonksiyonları geçmişte

kullanılmış olsada derin öğrenme eğitiminin hızı konusunda en iyi sonucu ReLu fonksiyonu verdiği için artık bu fonksiyon kullanılmaktadır.

ReLu [0, + ∞] aralığında değer alması , negatif eksende sıfır değer alması demek olur buda daha hızlı çalışmasını sağlar. Hesaplama maliyetinin hiperbolik tanjant veya sigmoid fonksiyonlarına göre az olması çok katmanlı ağlarda daha tercih edilmesine sebep olmuştur.

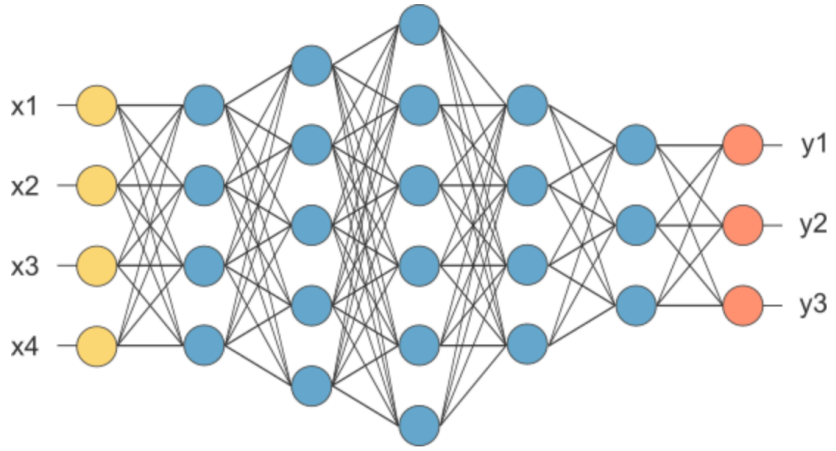
Havuzlama Katmanı (Pooling Layer) : Havuzlama katmanı CNN 'de isteğe bağlıdır, konvolüsyonel katmanlar arasına sıklıkla eklenip kullanılsada, bazı mimarilerde kullanılmaz. Hesaplama sayısını azaltmak, gösterimin kayma boyutunu ve ağ içindeki parametreleri ayarlamak için bu katman kullanılır.Bu sayede ağdaki uyumsuzluk kontrol edilmiş olur. Maksimum, minimum ve ortalama olmak üzere 3 tip havuzlama vardır.

Düzleşme Katmanı (Flattening Layer) : Bu katmanın görevi, CNN 'nin en önemli katmanlarından biri olan tam bağlantılı katmanın girişindeki verileri hazırlamaktır.Genel olarak, sinir ağları, girdi verilerini tek boyutlu diziden alır. Bu sinir ağındaki veriler ise konvolüsyonel ve havuzlama katmanından gelen matrislerin tek boyutlu diziye çevirilmiş halidir.



Şekil 2. 16 Düzleşme Katmanı Veri Yapı Değişimi [21]

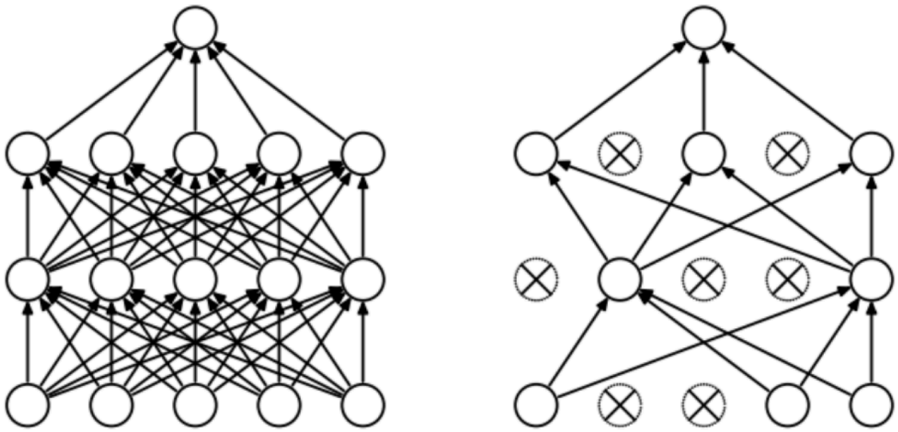
Tam Bağlantılı Katman (Fully Connected Layer) : Bu katman CNN ‘in en önemli katmanlarındandır. Farklı mimarilerde tam bağlantılıtı katman adedi değişebilir. Verileri düzleşme katmanından alır ve sinir ağı yoluyla öğrenme işlemini gerçekleştirir. Konvolüsyonel katmandan çıkan öznitelikler arası ilişkiyi sağlar. Veriler gizli katmanlardan, çıkış katmanına aktarılır. Maliyet değeri sıfıra yaklaşıncaya veya sıfır olana kadar öğrenmeye devam ettirilir.



Şekil 2. 17 Tam Bağlantılı Katman Yapısı [20]

Şekil 2.17 ‘de, x1,x2,x3 özellik haritasının matris vektörüne cevirilmiş halleridir. Tam bağlantılı katmanlarla, bir model oluşturmak için bu özellikler bir araya getirilir. Son olarak, çıktıları asyalı, afrikalı vb. gibi sınıflandırmak için kullanılır.

Seyreltme/Bırakma Katmanı (DropOut) : Veriler üzerinden modelin eğitilmesi sırasında aşırı öğrenmeyi(overfitting) engellemek için kullanılan katmandır.

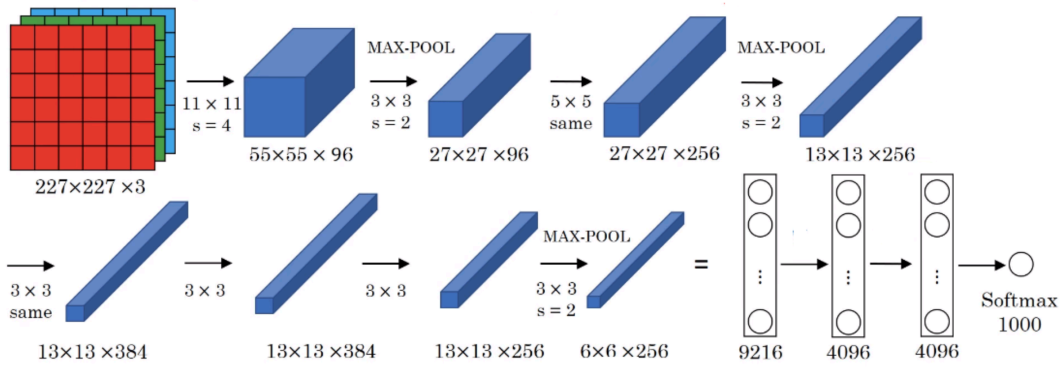


Şekil 2. 18 Seyreltme Katmanı Yapısı [22]

Çok uzun süre eğitim yapılıyorsa veya veri sayısı çok az ise aşırı öğrenme riski vardır. Ezber olarak nitelendirilebilecek aşırı öğrenmeyi engellemek için bu katman ile bazı nöronları unutmak gerekir.

2.4.2 Derin Öğrenme Modelleri

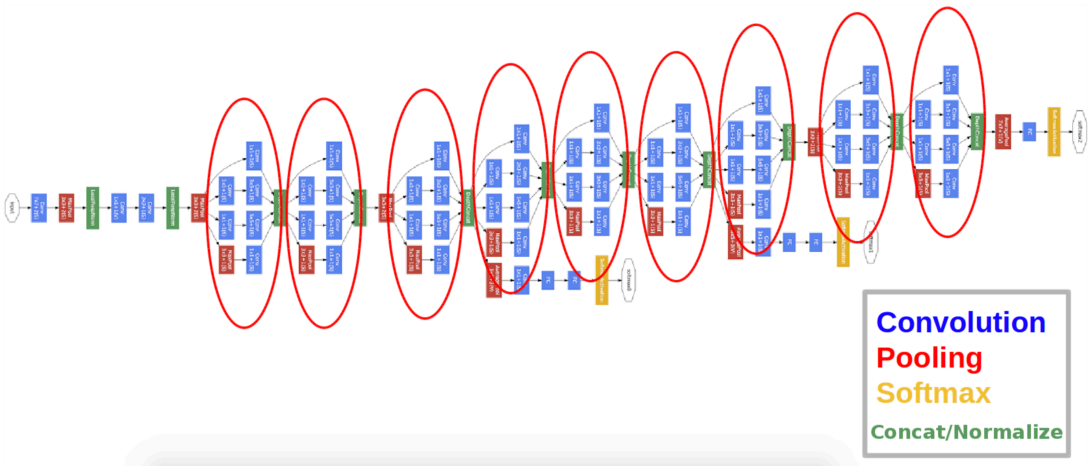
AlexNet : Derin öğrenmenin ve konvolüsyonel sinir ağ modellerinin 2000'li yıllarda tekrar popüler hale gelmesini sağlayan ilk çalışmadır. Alex Krizhevsky, Geoffrey Hinton ve Ilya Sutskever tarafından geliştirilmiştir.Takip halindeki konvolüsyonel ve havuzlama(pooling) katmanları bulunur. Aktivasyon fonksiyonu olarak ReLu, havuzlama katmanlarında ise maksimum-havuzlama kullanılır.



Şekil 2. 19 AlexNet Model Yapısı

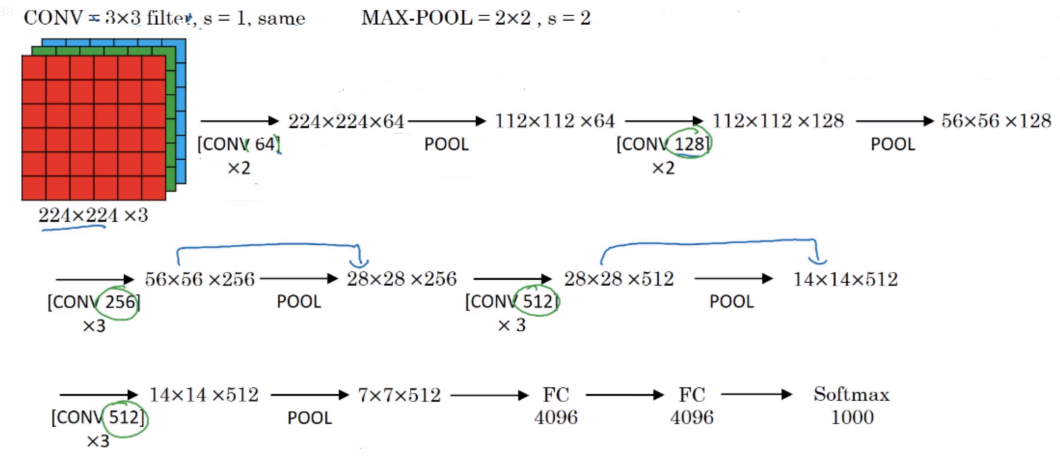
Büyük ve derin olan bu ağ modeli paralel çift GPU (Grafik İşlem Birimi) üzerinde iki parçalı bir modeldir. 60 milyona yakın parametre hesaplanmaktadır. ImageNet yarışmasında sınıflandırma doğruluk oranını %74,3'ten %83,6'ya çıkararak bir kırılma meydana getirmiştir.

GoogLeNet : Anlaşılması zordur ancak hesaplama karmaşıklığına ve büyüklüğüne bulduğu çözüm beraberinde hızı ve başarıyı getirmektedir. AlexNet'ten 12 kat daha az parametre içermektedir. 27 katman derinliğe sahiptir. Konvolüsyon, Maks-Havuzlama, softmaxlayer, tam bağlantılı katman, reLu katmanı, giriş katmanı ve çıkış katmanı gibi 144 katmandan oluşmaktadır. ImageNet yarışmasında %93 doğruluk oranında başarım elde etmiştir. Kullanılan herbir modüle inception(başlangıç) adı verilmektedir. Toplamda 9 inception bloğundan oluşur, bu işlere GoogLeNet adı verilir.Modelin kendisi genişleyebilmektedir.



Şekil 2. 20 GoogleNet Model Yapısı

VGG : Basit bir ağ modelidir. En önemli farkı konvolüsyonel katmalarının 2'li ya da 3'lü olmasıdır. Tam bağlantılı katmanında 7x7x512=4096 nöronlu bir öznitelik vektörüne dönüştürülür. İki tam bağlantılı katmanı çıkışında 1000 sınıflı softmax başarım hesaplanır.138 milyona parametre hesabı yapılmaktadır. Girişten çıkışa doğru matrislerin yükseklik ve genişlik boyutları azalırken derinlik değeri (kanal sayısı) artmaktadır.

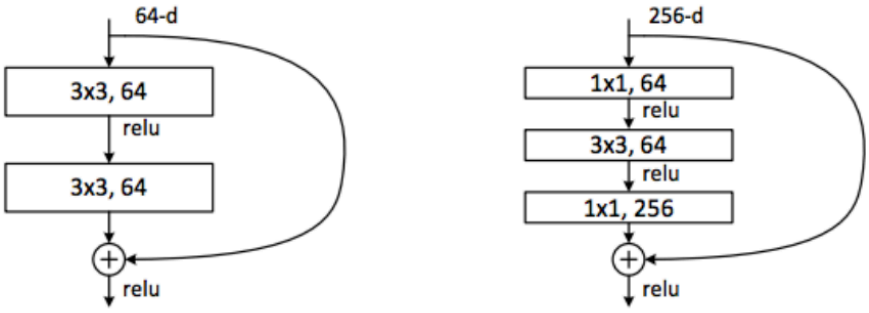


Şekil 2. 21 VGG Model Yapısı

VGG 16 : ILSVRC2014 yarışmasında kullanılan 13 konvolüsyon 3 tam bağlı katmanından oluşan bir ağdır. Maks-havuzlama, tam bağlantılı katman, ReLu katmanı, seyreltme katmanı ve Softmax katmanlarıyla birlikte toplamda 41 katman yer almaktadır. Girdi katmanında yer alacak görüntü 224x224x3 boyutundadır. Son katman ise sınıflandırma katmanıdır. ImageNet yarışmasında %89 doğruluk yakalamıştır.

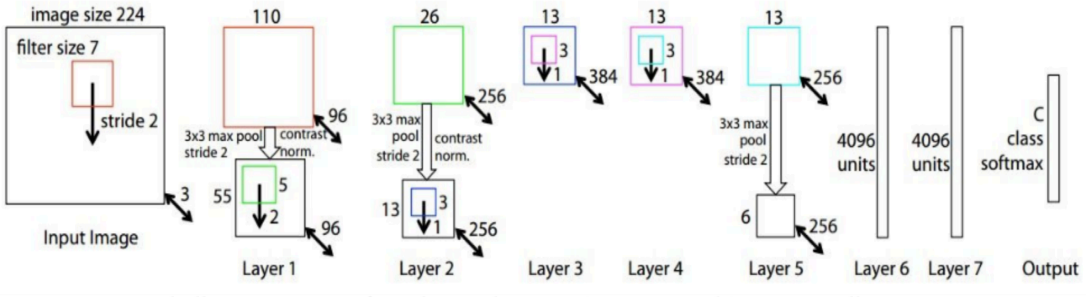
VGG 19 : Vgg16 algoritmasının akabinde oluşturulan 16 konvolüsyon 3 tam bağlı katmanlı bir ağdır. Maks-havuzlama, tam bağlantılı katman, reLu katmanı, seyreltme katmanı ve Softmax katmanlarıyla birlikte toplamda 47 katman yer almaktadır. Giriş katmanı 224x224x3 boyutundadır. Son katman ise sınıflandırma katmanıdır. ImageNet yarışmasında %88 doğruluk yakalamıştır.

ResNet : ResNet mikro mimari modüllü bir yapıya sahiptir. Artık değerlerin, sonraki katmanları besleyen blokların modele eklenmesiyle oluşmaktadır. ResNet bu özelliği sebebiyle klasik bir model değildir. 177 katmandan oluşan bir ağ yer almaktadır. Giriş katmanı 224x224x3 boyutundadır. Şekil 2.22 ‘de Resnet bağlantı örneği görülmektedir.



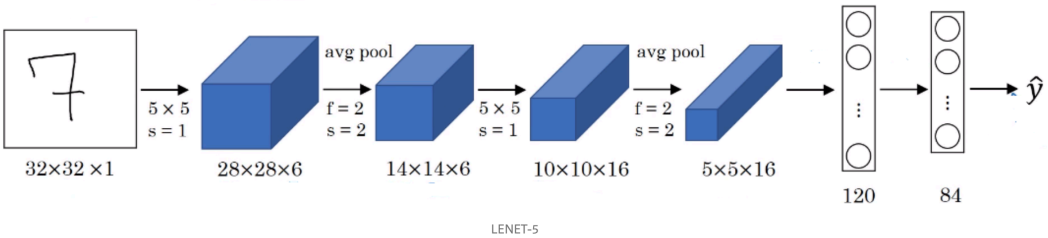
Şekil 2. 22 ResNet Model Yapısı

ZfNet : ILSVRC 2013 kazananı da ZFNet olarak bilinen bir CNN idi. ZFNet Dr. Rob Fergus ve PhD öğrencisi Dr. Matthew D. Zeiler tarafından bulunmuştur ve ismi de soyadlarından esinlenme bir isimdir. ZFNet'in farkı standart Convolüsyon > Aktivasyon > Havuzlama ve bu sistemin tekrar şemasından uzaklaşıp geri havuzlama ve geri konvolüsyon adımlarını tanıtmasıdır.



Şekil 2. 23 ZfNet Model Yapısı

LeNet-5 : 1998 yılında ilk başarılı sonucu veren konvolüsyonel sinir ağı modelidir. Yann LeCun ve ekibi tarafından geliştirilmiştir. Posta numaraları, banka çekleri üzerindeki sayıların okunması için geliştirilmiştir. MNIST (Modified National Institute of Standards and Technology) veri seti üzerinde deneyler gösterilmiştir. Bu modelde sonradan geliştirilecek maks-havuzlama yerine ortalama havuzlama işlemi yapılmaktadır. Aktivasyon fonksiyonu olarak ta hiperbolik ve sigmoid tanjant kullanılır.



Şekil 2. 24 LeNet-5 Model Yapısı

Yukaridaki örnek için tam bağlantılı katmanına gelen parametre sayısı 5x5x16=400 ve y çıkışında 10 sınıflı softmax bulunmaktadır. 60 bin parametre hesaplamaktadır. Matrisin yükseklik ve genişlik bilgisi azalırken derinlik (kanal sayısı) değeri artmaktadır.

BÖLÜM 3. MATERYAL VE YÖNTEM

Derin öğrenme çalışmalarında kullanılan farklı programlama dilleri, bu dillere bağlı farklı kütüphaneler bulunmaktadır. En çok tercih edilen programlama dilleri Python ve Matlab olmasına rağmen farklı seçeneklerde mevcuttur. Bu çalışmada Python programlama dili tercih edilmiştir. Python ‘ın modern bir programlama dili olması, çok sayıda doküman ve desteğin olması, stabil ve hızlı çalışan birçok kütüphane barındırması tercih için önemli etkenler olmuştur.

Çalışmada donanım olarak ilk başta 16 GB 2400 MHz DDR4 RAM, 2,6 GHz 6-Core Intel Core i7 CPU, Intel UHD Graphics 630 ve Radeon Pro 560X GPU özelliklerinde Macbook Pro kullanılmıştır. Çalışmanın ilerleyen aşamalarında bu donanımın yetersiz kalmasından dolayı www.paperspace.com, www.scaleway.com ve Google Cloud Platform gibi siteler üzerinden yüksek işlem gücüne sahip ekran kartı barındıran sunucular kiralanarak çalışmaya devam edilmiştir.

3.1 Python Kütüphane ve Modülleri

Python dili tercih edildikten sonra ortam kurulumu için kullanılan kullanılan modüllerden en öenmlileri ;

Conda : Conda bir açık kaynaklı paket ve ortam yönetme sistemidir. Windows, Linux ve macOS üzerinde çalışabilme imkanına sahiptir. Conda ile istenilen paketler bağımlı olduğu kütüphaneler ile birlikte kolay bir şekilde kurulabilir. Conda ile Python ortamları basit bir şekilde oluşturulabilir ve bu ortamlarda değişiklikler yapılıp silinebilir. Ortaya çıkışındaki amaç Python’a destek vermek olsa da, Conda ile herhangi bir dil için herhangi bir program yüklenebilir. Eğer istenilen paket yüklü olan Python’dan farklı bir sürüm gerektiriyorsa farklı bir Python’la ortam oluşturarak yükleme işlemini tamamlanabilir.

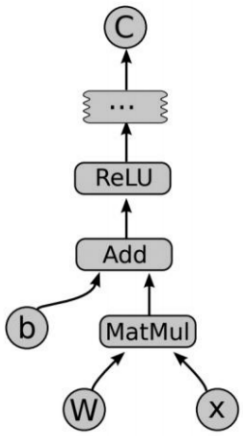
Anaconda : Anaconda Python ve R dillerinde veri bilimi uygulamaları geliştirmek için hazırlanan bir platformdur. Dünya genelindeki 15 milyon kullanıcısı ve içeriğindeki 1500’den fazla veri bilimi paketi ile endüstri standardı haline gelmiştir. İçerisinde endüstri liderleri tarafından desteklenen 1500’den fazla veri bilimi paketi bulunmaktadır.

İçerisinde Conda'yı barındırır. Anaconda ile istediğimiz yapay zeka motorunda geliştirme yapabiliriz. TensorFlow, Theano ve scikit-learn Anaconda paketleri tarafından desteklenmektedir. Veri bilimi için endüstri standardı haline gelmiş kütüphaneleri barındırmaktadır. NumPy, pandas, Dask ve Numba bu kütüphanelerden birkaçıdır. Veri sonuçlarını gösterebileceğimiz Matplotlib, Bokeh, Datashader ve Holoviews gibi grafik kütüphanelerini barındırmaktadır.

3.1.1 Python Derin Öğrenme Kütüphaneleri

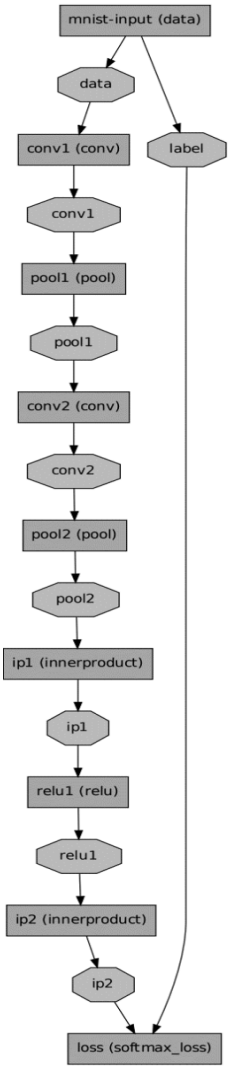
Derin öğrenme üzerine geçtiğimiz yıllarda artan ilgi, bu konuda yayınlanan projelerde de kendini göstermektedir. Yıllardır farklı gruplar tarafından geliştirilen kütüphaneler yaygınlıklarını sürdürürken, diğer taraftan bu konuyu şirket hedefi haline getirmiş büyük şirketlerde kendileri için geliştirdikleri kütüphaneleri araştırmacıların kullanımına sundu. En yaygın kullanılan kütüphaneler ;

TensorFlow : TensorFlow 2011 yılında Google araştırma ekibi(Google Brain) tarafından geliştirilmiştir. Bu grubun oluşturduğu ilk makine sistemi DistBelief'tir. DistBelief'in kullandığı alanlar dil çevrimi, öğretmensiz öğrenme, nesne tespiti ve görüntü sınıflandırması, video sınıflandırması, konuşma tanıma, dizi tahmini, yaya belirleme, takviyeli öğrenme gibi alanlardır. DistBelieften sonra makine öğrenimini daha etkin hale getirebilecek TensorFlow'u geliştirmişlerdir.Temel olarak bir dizi hesaplamalardan oluşan veri, akış grafiklerinden oluşmaktadır. Akış grafikleri, güncellemek için dallanma ve döngü kontrolüne izin veren ve düğümlerin durumunu korumak bir veri akışı hesaplamasını sunar. Şekil 3.1 de örnek yapı gösterilmiştir.



Şekil 3. 1 TensorFlow Yapı Örneği

Caffe : Caffe (Convolution Architecture For Feature Extraction) Berkley üniversitesinde geliştirilmiş, açık kaynak kodlu bir kütüphanedir. GPU kullanımını destekler. Açık kaynak kodlu olması sebebiyle genellikle ticari amaçla kullanılır. CaffeNet konvülasyon, havuzlama, iç-gizli katmanlar, relu, normalizasyon, elemanlı işlemler, softmax ve hinge gibi katmanlardan oluşur. Şekil 3.2 'de veritabanı sınıflandırmasını yapan bir cafe modeli CafeNet örneği gösterilmektedir.



Şekil 3. 2 Caffe Yapı Örneği

Theano : LISA laboratuvarlarında geliştirilmiştir. Kullanım amacı makine öğrenim algoritmalarını hızlı ve verimli bir şekilde kullanmaktır. İsmini Theano isimli bir yunan matematikçiden almıştır. Çok boyutlu matrisleri barındıran matematiksel ifadelerin

optimize edilmesi ve verimli şekilde kullanılmasını sağlar. Açık kaynak kodludur. Sınıf oluşturmayı gösteren derin öğrenme öğreticisi mevcuttur. Yapay sinir ağları kullanılmaz. Theano genelde en hızlı çalışan derin öğrenme kütüphanesidir.

Torch : Açık kaynak kodlu bir kütüphanedir. Hızlı ve kolay şekilde derin öğrenme yapısı oluşturulabilir. Sayısal optimizasyon yapılabilir. Açık kaynak kodlu olduğu için bir çok şirket ticari amaçla kullanmıştır. Ayrıca sıklıkla üniversiteler tarafından da kullanılmaktadır.

Keras : Theano ve TensorFlow için kullanılan Phyton dilinde yazılmış bir kütüphanedir. Google'da çalışan mühendis Francois Chollet tarafından geliştirilmiştir. Keras'ın asıl yaptığı iş; algoritmaları optimize etmek, normalleştirme sağlamak ve aktivasyon fonksiyonlarını en iyi seçim olarak seçmektir. Özetle TensorFlow ve Theano tarafından desteklensede giriş verilerinin boyutlarındaki farklılıklar yüzünden her iki kütüphane içinde çalışabilmesi için iyi bir şekilde tasarlanması gerekir. Bu çalışmada, kullanım kolaylığı ve sağladığı hız göz önüne alınarak keras tercih edilmiştir.

3.2 Ön İşlemler

Ön işlemler, derin öğrenme ile özellik çıkarımından önce veri üzerinde yapılan görüntü işleme, görüntü iyileştirme çalışmalarıdır. Bu çalışmalar sayesinde veri özellik çıkarımı için daha uygun hale gelir ve uygulama performansı artar. Ön işlemlere örnek olarak histogram eşitleme, bulanıklık giderme, segmentasyon, kenar belirleme verilebilir.

3.2.1 Histogram Eşitleme

Bir görüntüdeki renk değerlerinin sayılarını gösteren grafiğe histogram denir. Histogram eşitleme ise bir resimdeki renk değerlerinin belli bir yerde kümelenmiş olmasından kaynaklanan renk dağılımı bozukluğunu düzelmek için kullanılan bir yöntemdir.

$$p_r\,(r_k) = \frac{n_k}{n} \qquad k = 0,\ 1,\ 2\ ,\ 3,\ \ldots..,\ L-1 \tag{3.1}$$

p(r_k), k. ton değerinin görüntü içinde hangi oranda temsil edildiğini gösterir.

n_k, k. tondan görüntü içinde kaç adet olduğunu gösterir

n, toplam piksel sayısıdır.

Önce kümülatif olasılık fonksiyonu sk hesaplanır.

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(j_j) \qquad (3.2)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0,1,2,3, \ldots\ldots, L - 1$$

Daha sonra ters dönüşümü yapılarak hangi renk tonu yerine ne geleceği hesaplanır.

$$r_k = T^{-1}(s_k), \qquad 0 << s_k << 1 \qquad (3.3)$$

$$T^{-1}(s_k) = (L - 1) * T(r_k), \qquad k = 0,1,2,3\ldots\ldots L - 1$$

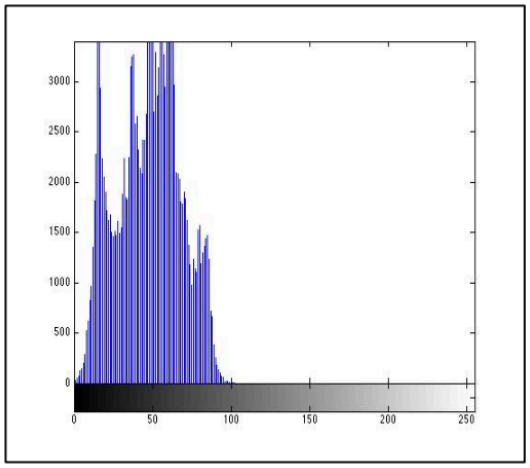
İşlemler sonra histogram eşitlenme tamamlanmış olur.



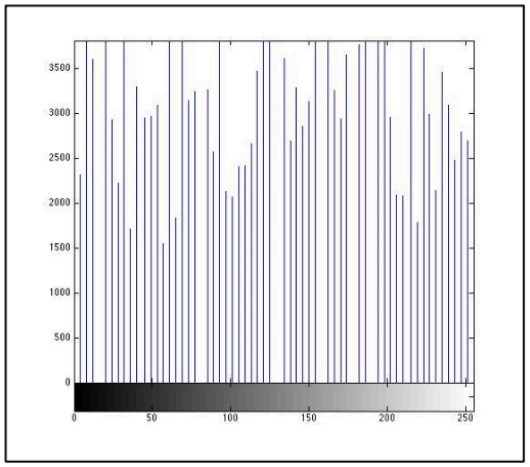
Şekil 3. 3 Histogram Eşitleme Uygulanmamış Görüntü



Şekil 3. 4 Histogram Eşitleme Uygulanmış Görüntü



Şekil 3. 5 Histogram Eşitleme Uygulanmamış Görüntü Histogramı



Şekil 3. 6 Histogram Eşitleme Uygulanmış Görüntü Histogramı

Histogram eşitlemenin görüntü üzerinde oluşturduğu değişiklik Şekil 3.3'te, görüntünün histogramında meydana getirdiği değişiklik ise Şekil 3.5'te görülmektedir. Histogram eşitleme genelde görüntünün kontrast oranını artırır, bu sayede yoğunluklar histogramda daha iyi dağıtılabilir.

3.2.2 Yüz Tanıma ve Anlamlı Verinin Alınması

Veri derin öğrenme modeline gitmeden önce yapılacak en önemli işlemlerden birisi yüz tanıma ve düzenlemedir. Bu işlem için farklı seçenekler olmasının yanında en çok tercih edilen görüntü işleme yöntemi ve kütüphane olarak OpenCV 'dir.

Veri üzerinde ilk olarak yüz tanıma, sonra yüz hatlarına göre açı düzeltme ve kesme işlemleri uygulanır. OpenCV içerisinde Haar Cascade, DNN, HoG, Dlib gibi farklı seçenekler mevcut.

Haar Cascade : Haar Cascade tabanlı yüz dedektörü, Viola ve Jones tarafından tanıtıldığı 2001 yılından bu yana uzun yıllar yüz algılamada son teknoloji ürünü olarak kullanılmıştır. OpenCV üzerinden Haar tabanlı birçok yeni model mevcuttur. CPU üzerinden hızlı çalışır, basit bir mimariye sahiptir, farklı boyutlarda yüzleri tanıyabilir. Ancak ön açıdan olmayan görüntülerde yüz tanıyamaz ve tahmin kısmında çok sayıda hata barındırır.

DNN : Dnn modeli OpenCV'ye 3.3 sürümünde dahil olmuştur. ResNet-10 mimarisi kullanılmıştır. Web üzerinden oluşturulan veri seti ile eğitilmiştir. Caffe ve tensorflow destekleri ile 2 sürümü vardır. CPU üzerinde gerçek zamanlı çalışır. Farklı yüz yönleri için doğru sonuçlar verir. Çeşitli ölçeklerdeki yüzleri algılar. Dnn, Haar tarafından sağlanan herhangi bir faydadan ödün vermeden Haar kaskad tabanlı dedektörün tüm dezavantajlarının üstesinden gelir.

HoG : HoG yüz dedektörü Dlib üzerinde çalışan, yaygın olarak kullanılan bir modeldir. CPU üzerinde çalışan en hızlı modeldir. Ön açıdan iyi, yan açılardan ise kısmen çalışır. Diğer modellere göre daha hafif daha kolay eğitilen bir modeldir. En büyük dezavantajı 80x80 yüz boyutu için eğitildiğinden dolayı küçük yüzleri algılayamaz.

Dlib : Dlib CNN dedektörü CNN tabanlı özelliklere sahip bir Maksimum Marj Nesne Dedektörü(MMOD) kullanır. Bu modeli eğitmek için büyük miktarda veriye ihtiyaç yoktur, küçük miktarda veri ile basitçe eğitilebilir. ImageNet, PASCAL, VOC, VGG, WIDER, Face Scrub gibi çeşitli veri setlerinden görüntüler ile oluşan , özel olarak etiketlenmiş veri seti ile eğitilmiştir. Farklı yüz açılarından yüksek performans ile çalışır. GPU üzerinden çok hızlıdır, buna rağmeen CPU da çok yavaştır. Minimum yüz boyutu 80x80 için eğitildiği için küçük yüzleri algılayamaz.

Bahsedilen modellerden projeye en uygun olanı tercih edilerek ön işlemler tamamlanmalıdır. Sırasıyla ilk olarak görüntü üzerinden yüz tespit edilir daha sonra yüzün açısında düzeltme uygulanır ve son olarak anlamlı yüz verisi kesilerek gereksiz kısımdan ayrıştırılır.

3.3 Yöntem

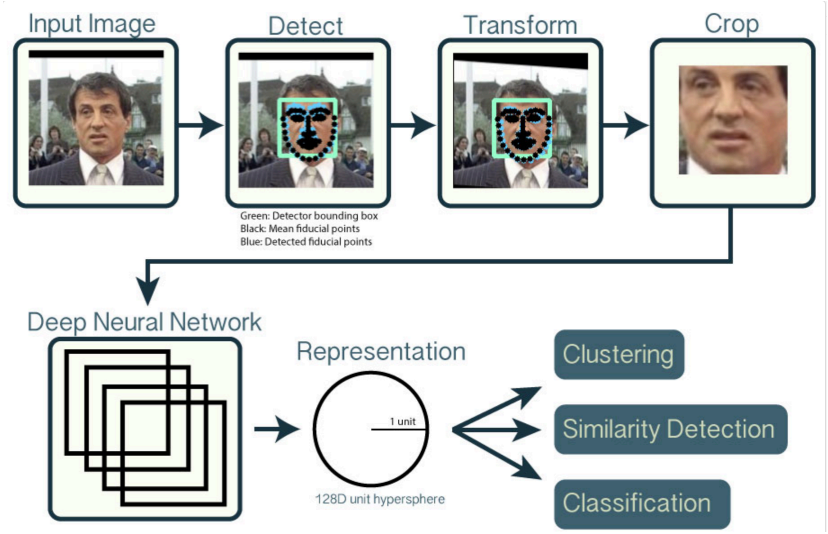
Derin öğrenme çalışmalarında doğru sonuçlar alabilmenin yolu, modelin çalışmaya uygun şekilde kurgulanması ve modelin iyi eğitilmesinden geçmektedir. Modelin eğitilmesi için kullanılan veri setinin boyutu, kalitesi eğitimin başarılı olabilmesi için gerekli en önemli etkendir.

Akademik çalışmalar için ücretsiz kullanıma açılmış UTKFace veri seti kullanılmıştır [17]. Veri seti içerisinde farklı etnik kökenden ve yaşlardan kadın, erkek görüntüleri bulunmaktadır. Asya, Afrika, Hindistan etnik kökenli 2000 adet kadın ve erkek görüntüsü kullanılmıştır. Daha sonra veri seti zenginleştirilmiş ve 6000 adet görüntü kullanılmıştır. Veri setinin sonuca etkisinin gözlemlenebilmesi için 2000 adet görüntüden oluşan veri seti 1 (VS1) ve 6000 görüntüden oluşan veri seti 2 (VS2) ile modeller ayrı ayrı eğitilmişlerdir.

Sistemin performansını arttırmak için modeli eğitmeye başlamadan önce görüntüler üzerinde iyileştirmeler yapılmıştır. İlk olarak histogram eşitleme yaparak görüntülerin kontrast oranı arttırılmış, görüntü kalitesi iyileştirilmiştir.

İkinci adım olarak OpenCV kütüphanesi ile Dlib kütüphanesi kullanılmış ve yüz tanıma, hizalama, kesme işlemleri yapılmıştır. Bu işlemlerin amacı model eğitimi için kullanılacak verilerden gereksiz bilgilerin temizlenmesidir.

Görüntü üzerinde insan yüzü tespit edilir, sonra landmark noktaları yöntemi kullanılarak çalışma için gerekli özniteliklerin bulunduğu alanlar işaretlenir. Görüntülerde bakış açıları farklı olabileceği için burada bir düzenlemeye ihtiyaç bulunmaktadır. Tüm görüntülerde insan yüzü aynı şekilde hizalandığı takdirde eğitim de bakış açısından kaynaklanacak farklılıklar ortadan kaldırılmış olur.

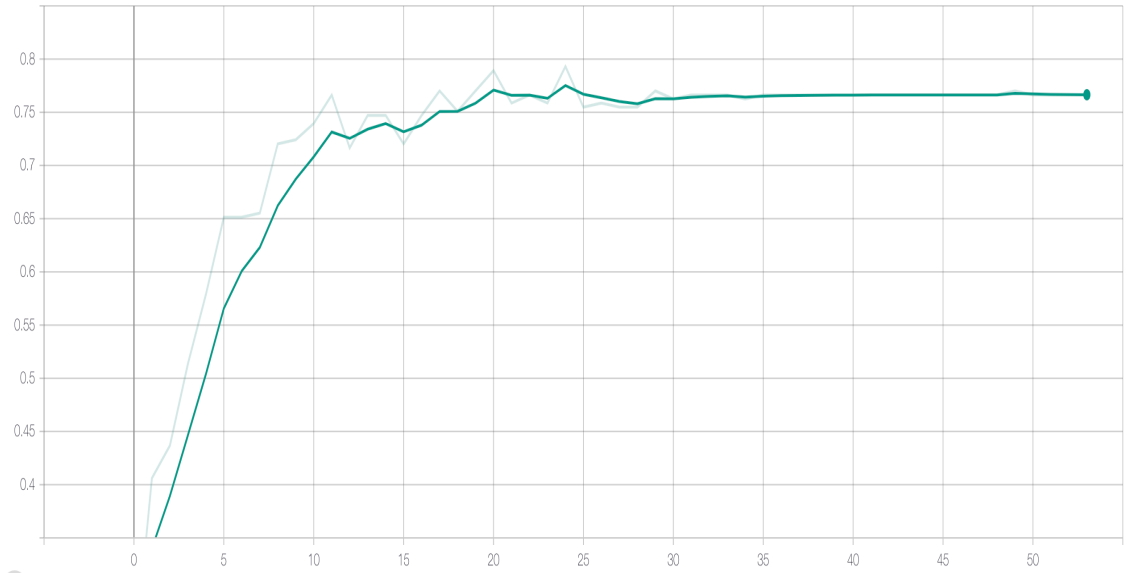


Şekil 3. 7 Dlib Kütüphanesi İşlemleri [24]

Landmark noktaları ile işaretlendikten sonra insan gözleri tespit edilir, gözler odak noktası alınarak resmin açısı düzeltilir ve gözler yatay düzleme paralel hale getirilir. [18] Daha sonra yüz cevresi kesilerek sadece karakteristik özelliklerin bulunduğu alan eğitime gönderilir. Şekil 3.7 ‘de akış anlatılmıştır.

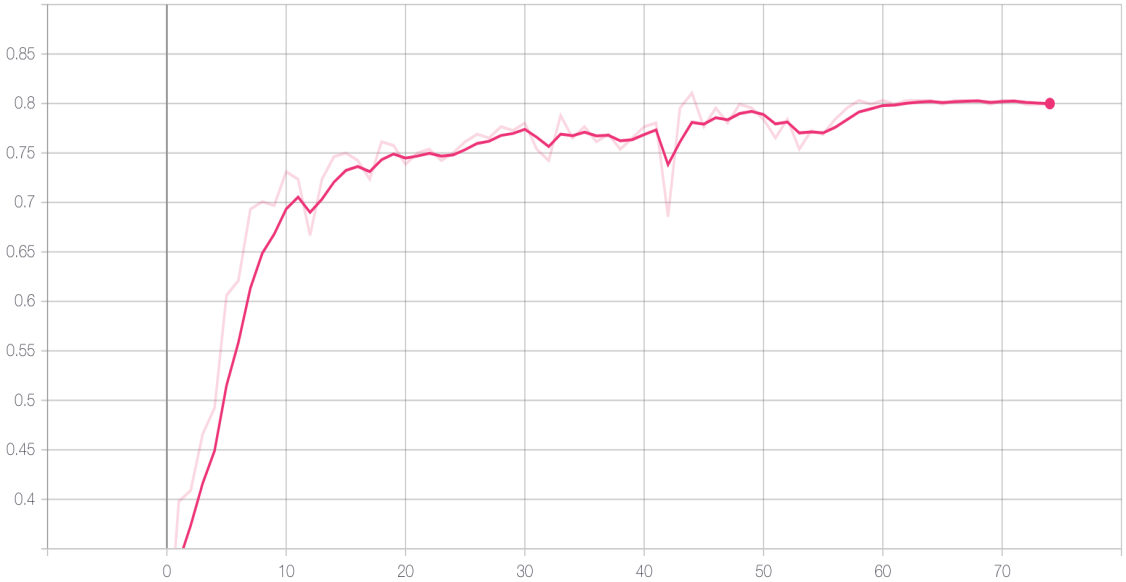
Derin öğrenme modeli eğitiminde birden fazla model kullanılmış ve ideal sistem kurulmaya çalışılmıştır. Akademik çalışmalarda en çok tercih edilen modellerden VGG-16, VGG-19, AlexNet ve ZfNet modelleri ile çalışılmış, hazırlanan veri seti ile eğitilmişlerdir. Tüm modeller VS1 ve VS2 için 5’er kez çalıştırılmış ve her seferinde 250 devir(Epoch) uygulanmıştır. Bu döngülerden elde edilen çıktılar incelenmiş ve modeller aşırı öğrenmeye girmeden yakaladıkları en uygun değerler üzerinden detaylı incelenmiştir. Sonuçlar incelenirken en yüksek oranlı tahminin doğru olduğu farz edilen TOP-1 ve en yüksek ilk iki orandan birisinin doğru olduğu farz edilen TOP-2 yaklaşımları ayrı ayrı değerlendirilmiştir.

VGG-16 : VGG-16 modeli baz alınarak kurulan model, 13 konvolüsyon 3 tam bağlı olmak üzere 16 katmandan oluşmaktadır. Adam optimizasyon fonksiyonu kullanılmıştır. Girdi katmanında yer alan görüntü 224x224x3 boyutundadır.

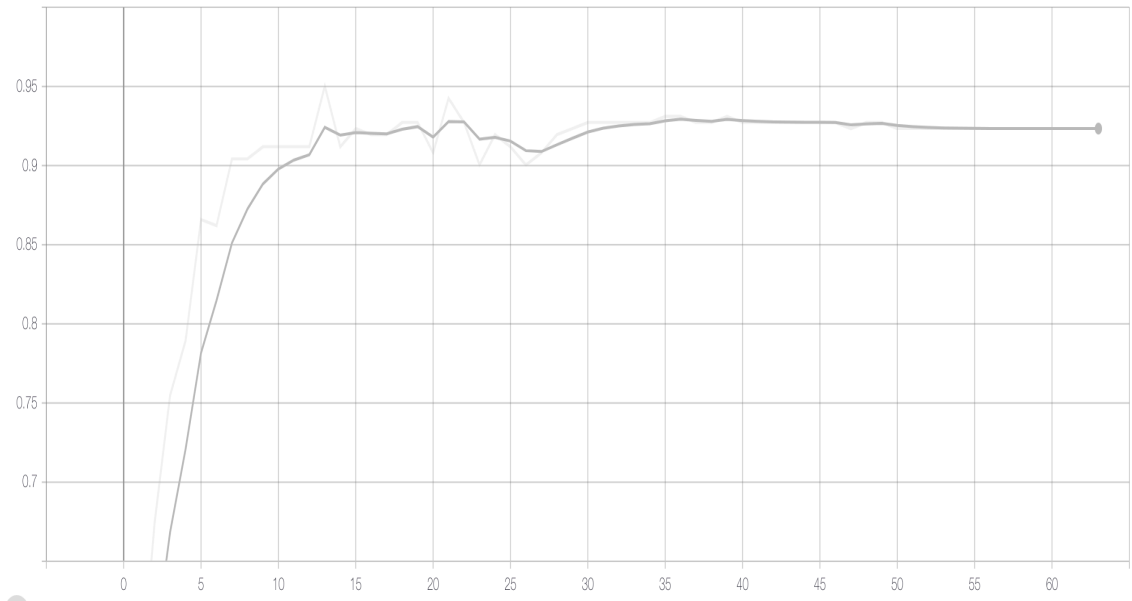


Şekil 3. 8 VGG16 VS1 TOP-1 Validasyon Verisi Doğruluk Tahmini

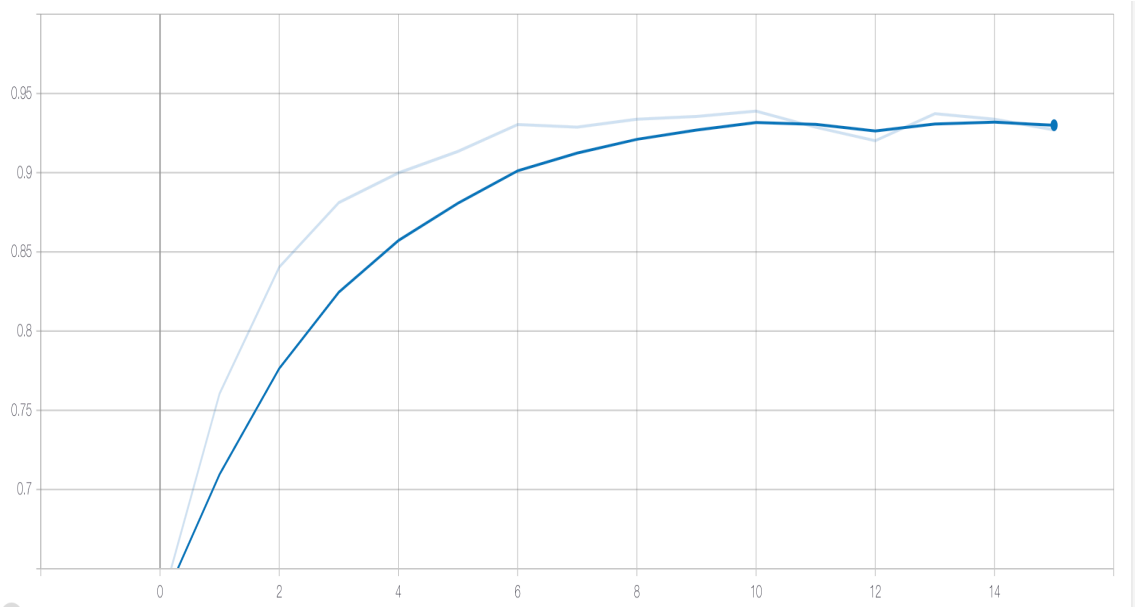
Şekil 3.8 ve Şekil 3.9 ‘da VGG16 modeli için veri seti 1 (VS1) ile yakalanan TOP-1 ve TOP-2 değerlerinin grafikleri gösterilmektedir.



Şekil 3. 10 VGG16 VS2 TOP-1 Validasyon Verisi Doğruluk Tahmini



Şekil 3. 9 VGG16 VS1 TOP-2 Validasyon Verisi Doğruluk Tahmini



Şekil 3. 11 VGG16 VS2 TOP-2 Validasyon Verisi Doğruluk Tahmini

Şekil 3.10 ve şekil 3.11 'de VGG16 modeli için veri seti 2 (VS2) ile yakalanan TOP-1 ve TOP-2 değerlerinin grafikleri gösterilmektedir.

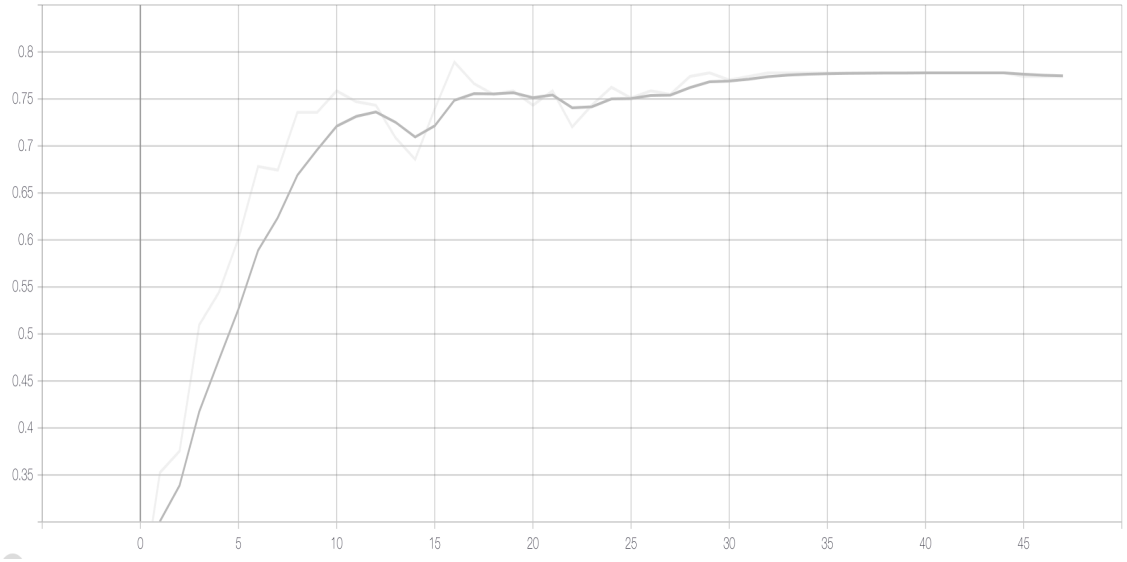
Tablo 3. 1 VGG-16 Model Özeti

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
conv2d_1 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_2 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_3 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_1 (MaxPooling2	(None, 56, 56, 128)	0
conv2d_4 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_5 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_2 (MaxPooling2	(None, 28, 28, 256)	0
conv2d_7 (Conv2D)	(None, 28, 28, 512)	1180160

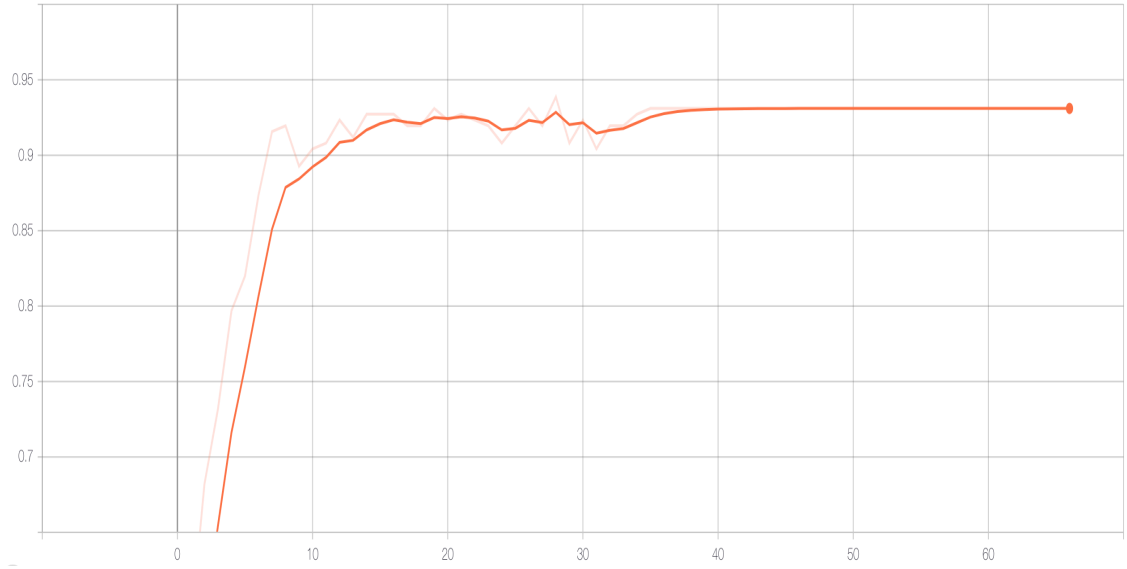
```
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 28, 28, 512)       2359808
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 28, 28, 512)       2359808
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 512)       0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 4096)              102764544
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 24582
=================================================================
Total params: 134,285,126
Trainable params: 134,285,126
Non-trainable params: 0
```

Tablo 3.1 ‘de görüldüğü üzere model 134 milyon 285 bin 126 adet parametre ile eğitilmiştir. Tüm devirler incelendiğinde, model aşırı öğrenmeye girmeden yakalanan en iyi kayıp/başarı oranları baz alınarak doğruluk tahmin değerleri çıkarılmıştır. VS1 ile eğitildiğinde TOP-1 doğruluk tahmin oranı validasyon verisi için %76,63 , eğitim verisi için %89,57, TOP-2 doğruluk tahmin oranı validasyon verisi için %91,19, eğitim verisi için %95,23 çıkmıştır. VS2 ile eğitildiğinde ise bu oranlar TOP-1’de validasyon verisi için %80,48, eğitim verisi için %85,04 , TOP-2’de validasyon verisi için %93,89 , eğitim verisi için %96,85 olmuştur.

VGG-19 : VGG-19 baz alınarak kurulan model, 16 konvolüsyon 3 tam bağlı katmandan oluşmaktadır. Adam optimizasyon fonksiyonu kullanılmıştır. Girdi katmanında yer alan görüntü 224x224x3 boyutundadır.

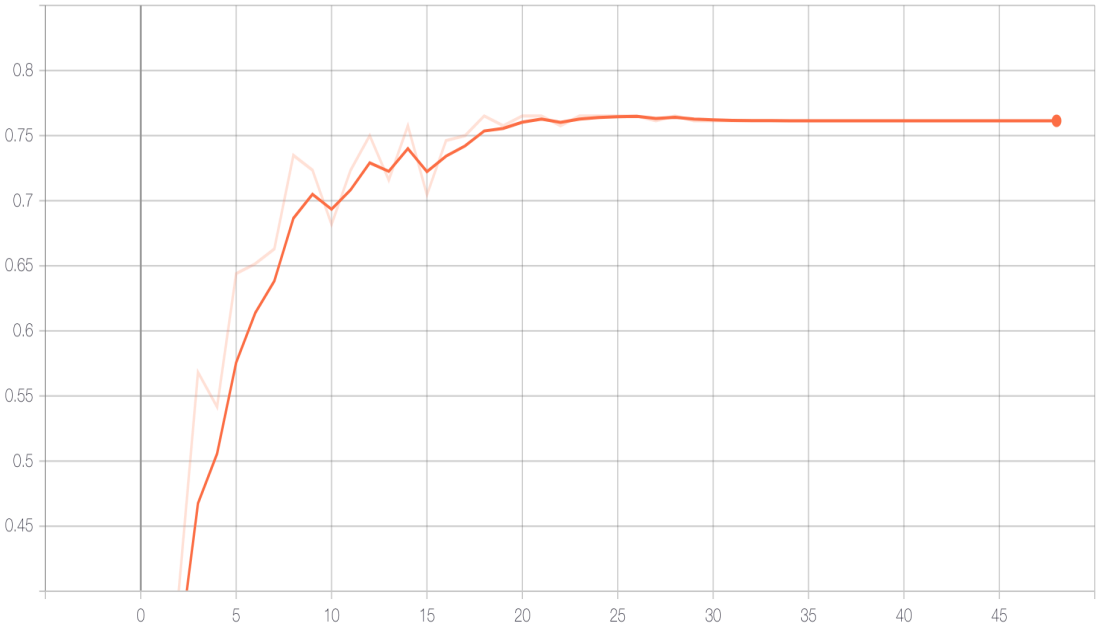


Şekil 3. 12 VGG19 VS1 TOP-1 Validasyon Verisi Doğruluk Tahmini

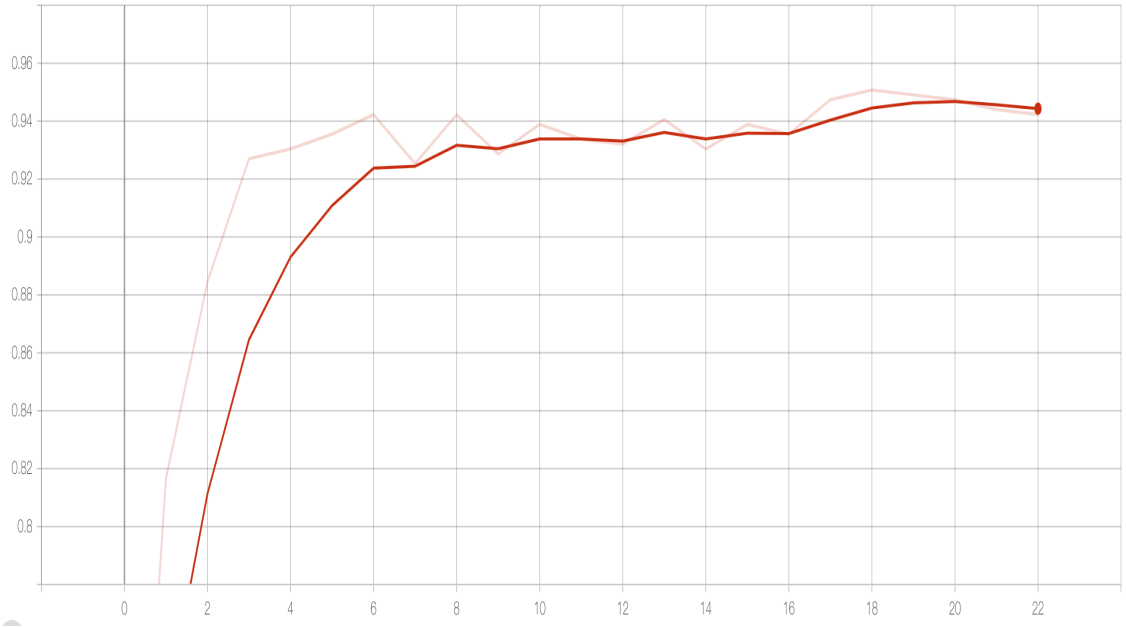


Şekil 3. 13 VGG19 VS1 TOP-2 Validasyon Verisi Doğruluk Tahmini

Şekil 3.12 ve Şekil 3.13 ‘de VGG19 modeli için veri seti 1 (VS1) ile yakalanan TOP-1 ve TOP-2 değerlerinin grafikleri gösterilmektedir.



Şekil 3. 14 VGG19 VS2 TOP-1 Validasyon Verisi Doğruluk Tahmini



Şekil 3. 15 VGG19 VS2 TOP-2 Validasyon Verisi Doğruluk Tahmini

Şekil 3.14 ve şekil 3.15 ‘de VGG19 modeli için veri seti 2 (VS2) ile yakalanan TOP-1 ve TOP-2 değerlerinin grafikleri gösterilmektedir.

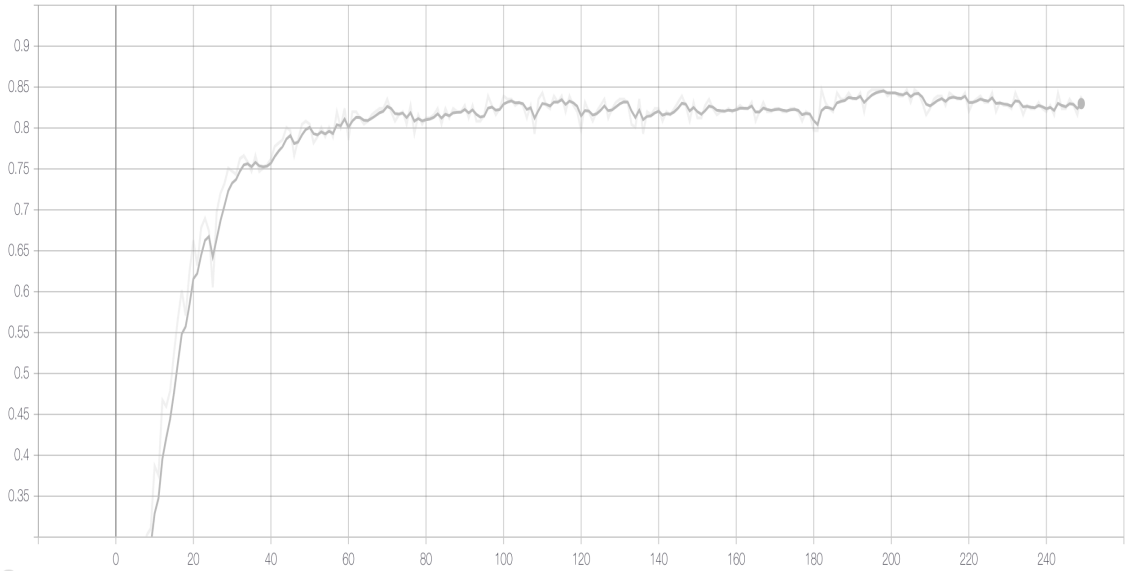
Tablo 3. 2 VGG-19 Model Özeti

Layer (type)	Output Shape	Param #
conv2d_16 (Conv2D)	(None, 224, 224, 64)	1792
conv2d_17 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d_5 (MaxPooling2	(None, 112, 112, 64)	0
conv2d_18 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_19 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_6 (MaxPooling2	(None, 56, 56, 128)	0
conv2d_20 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_21 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_22 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_23 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_7 (MaxPooling2	(None, 28, 28, 256)	0
conv2d_24 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_25 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_26 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_27 (Conv2D)	(None, 28, 28, 512)	2359808
max_pooling2d_8 (MaxPooling2	(None, 14, 14, 512)	0
conv2d_28 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_29 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_30 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_31 (Conv2D)	(None, 14, 14, 512)	2359808
max_pooling2d_9 (MaxPooling2	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_3 (Dense)	(None, 4096)	102764544
dense_4 (Dense)	(None, 4096)	16781312
dense_5 (Dense)	(None, 6)	24582

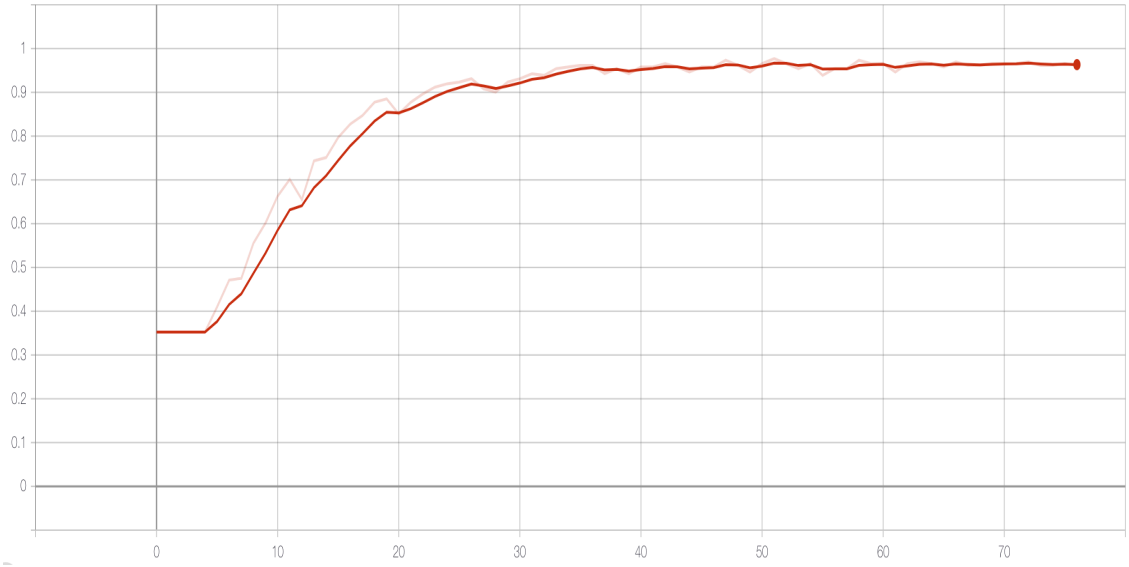
Total params: 139,594,822
Trainable params: 139,594,822

Tablo 3.2 de VGG-19 Modelinin 139 milyon 594 bin 822 adet parametre ile eğitilği görülmektedir. Tüm devirler incelendiğinde, model aşırı öğrenmeye girmeden yakalanan en iyi kayıp/başarı oranları baz alınarak doğruluk tahmin değerleri çıkarılmıştır. VS1 ile eğitildiğinde TOP-1 doğruluk tahmin oranı validasyon verisi için %75,86, eğitim verisi için %93,36, TOP-2 doğruluk tahmin oranı validasyon verisi için %91,95 , eğitim verisi için %92,04 çıkmıştır. VS2 ile eğitildiğinde ise bu oranlar TOP-1'de validasyon verisi için %76.52, eğitim verisi için %92,03, TOP-2'de validasyon verisi için %94,23, eğitim verisi için %97,58 olmuştur.

AlexNet : AlexNet baz alınarak kurulan model, 9 katmandan oluşmaktadır. Adam optimizasyon fonksiyonu kullanılmıştır. Girdi katmanında yer alan görüntü 224x224x3 boyutundadır.

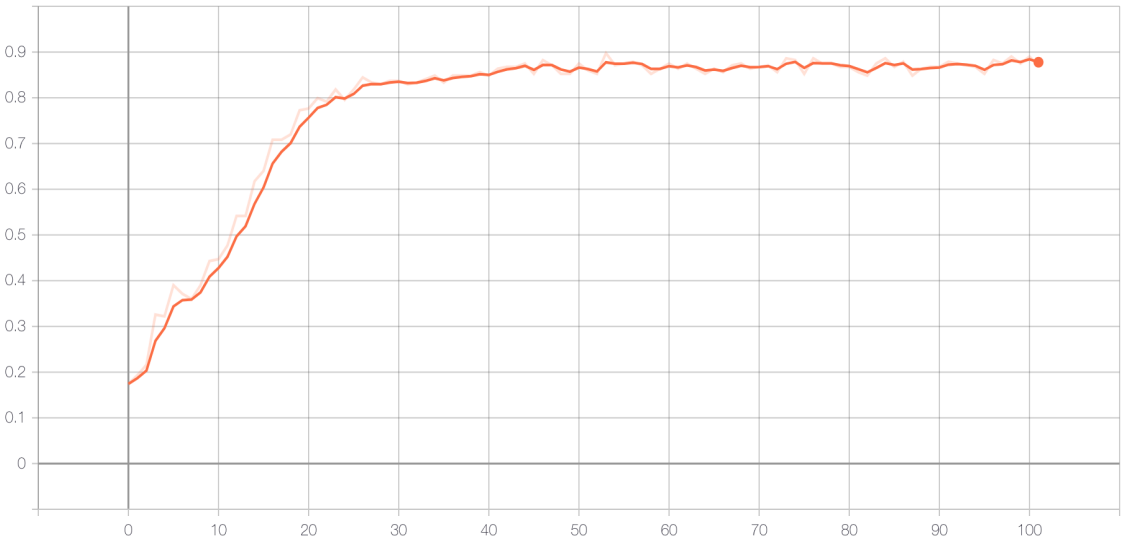


Şekil 3. 16 AlexNet VS1 TOP-1 Validasyon Verisi Doğruluk Tahmini

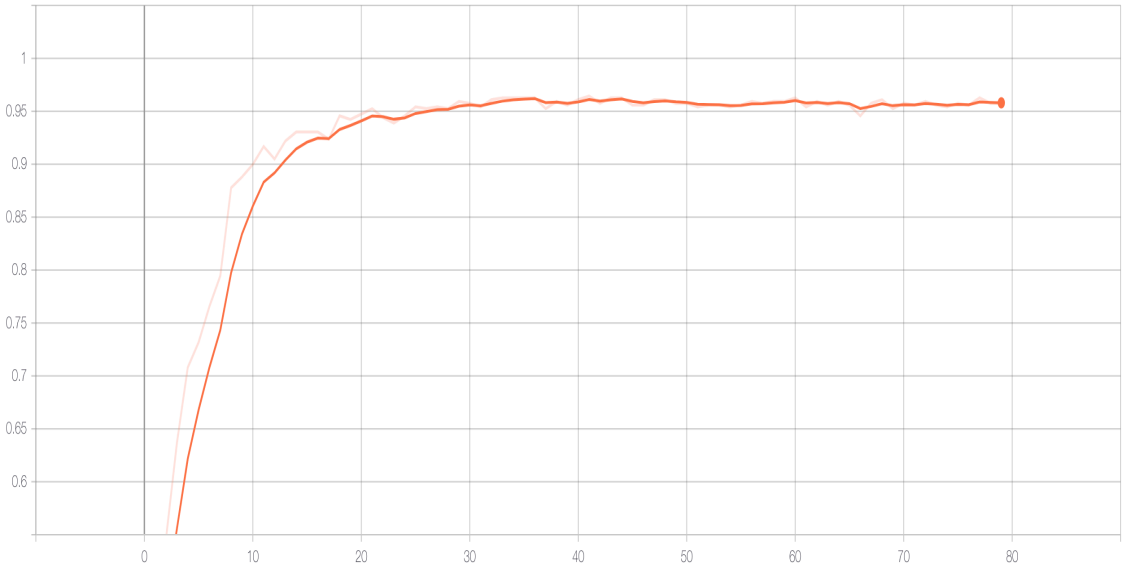


Şekil 3. 17 AlexNet VS1 TOP-2 Validasyon Verisi Doğruluk Tahmini

Şekil 3.16 ve Şekil 3.17 'de AlexNet modeli için veri seti 1 (VS1) ile yakalanan TOP-1 ve TOP-2 değerlerinin grafikleri gösterilmektedir.



Şekil 3. 18 AlexNet VS2 TOP-1 Validasyon Doğruluk Tahmini



Şekil 3. 19 AlexNet VS2 TOP-2 Validasyon Doğruluk Tahmini

Şekil 3.18 ve şekil 3.19 'da AlexNet modeli için veri seti 2 (VS2) ile yakalanan TOP-1 ve TOP-2 değerlerinin grafikleri gösterilmektedir.

Tablo 3. 3 AlexNet Model Özeti

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 54, 54, 96)	34944
activation_9 (Activation)	(None, 54, 54, 96)	0
max_pooling2d_3 (MaxPooling2	(None, 27, 27, 96)	0
conv2d_6 (Conv2D)	(None, 17, 17, 256)	2973952
activation_10 (Activation)	(None, 17, 17, 256)	0
max_pooling2d_4 (MaxPooling2	(None, 8, 8, 256)	0
conv2d_7 (Conv2D)	(None, 6, 6, 384)	885120
activation_11 (Activation)	(None, 6, 6, 384)	0
conv2d_8 (Conv2D)	(None, 4, 4, 384)	1327488
activation_12 (Activation)	(None, 4, 4, 384)	0
conv2d_9 (Conv2D)	(None, 2, 2, 256)	884992

activation_13 (Activation) (None, 2, 2, 256) 0

max_pooling2d_5 (MaxPooling2 (None, 1, 1, 256) 0

flatten_1 (Flatten) (None, 256) 0

dense_4 (Dense) (None, 4096) 1052672

activation_14 (Activation) (None, 4096) 0

dropout_3 (Dropout) (None, 4096) 0

dense_5 (Dense) (None, 4096) 16781312

activation_15 (Activation) (None, 4096) 0

dropout_4 (Dropout) (None, 4096) 0

dense_6 (Dense) (None, 1000) 4097000

activation_16 (Activation) (None, 1000) 0

dropout_5 (Dropout) (None, 1000) 0

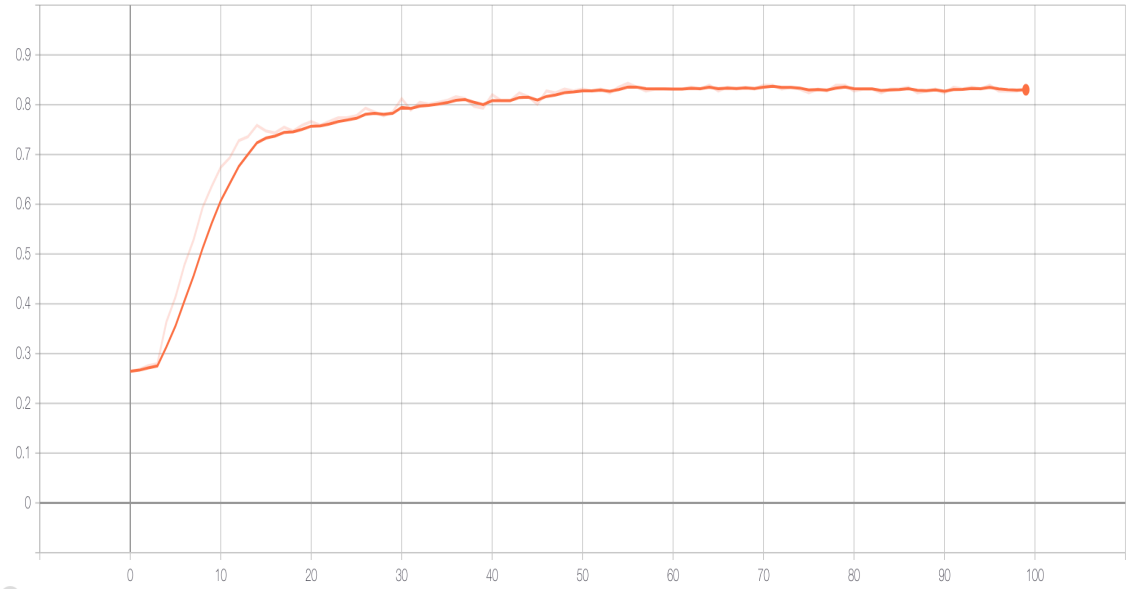
dense_7 (Dense) (None, 6) 6006

activation_17 (Activation) (None, 6) 0

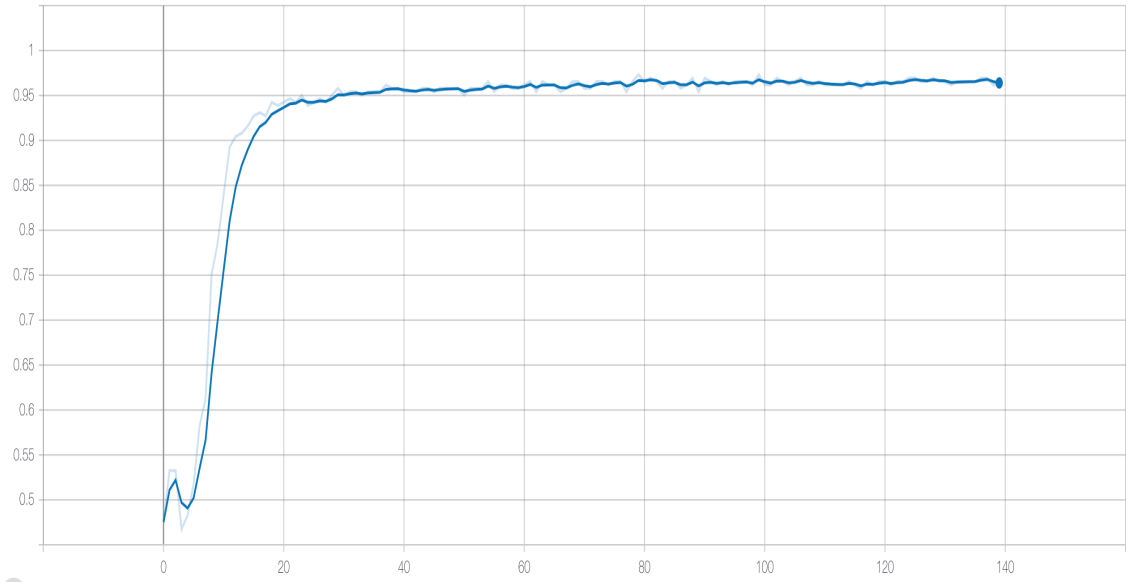
Total params: 28,043,486
Trainable params: 28,043,486
Non-trainable params: 0

Tablo 3.3 ‘de AlexNet Modelinin 28 milyon 43 bin 486 adet parametre ile eğit ilği görülmektedir. Tüm devirler incelendiğinde, model aşırı öğrenmeye girmeden yakal anan en iyi kayıp/başarı oranları baz alınarak doğruluk tahmin değerleri çıkarılmıştır. VS 1 ile eğitildiğinde TOP-1 doğruluk tahmin oranı validasyon verisi için %83,52, eğitim v erisi için %98,08, TOP-2 doğruluk tahmin oranı validasyon verisi için %97,70, eğitim v erisi için %99,02 çıkmıştır. VS2 ile eğitildiğinde ise bu oranlar TOP-1’de validasyon ver isi için %87.50, eğitim verisi için %97,01, TOP-2’de validasyon verisi için %97,26, eğit im verisi için %99,53 olmuştur.

ZfNet : ZfNet baz alınarak kurulan model, 8 katmandan oluşmaktadır. Adam optimiza syon fonksiyonu kullanılmıştır. Girdi katmanında yer alan görüntü 224x224x3 boyutund adır.

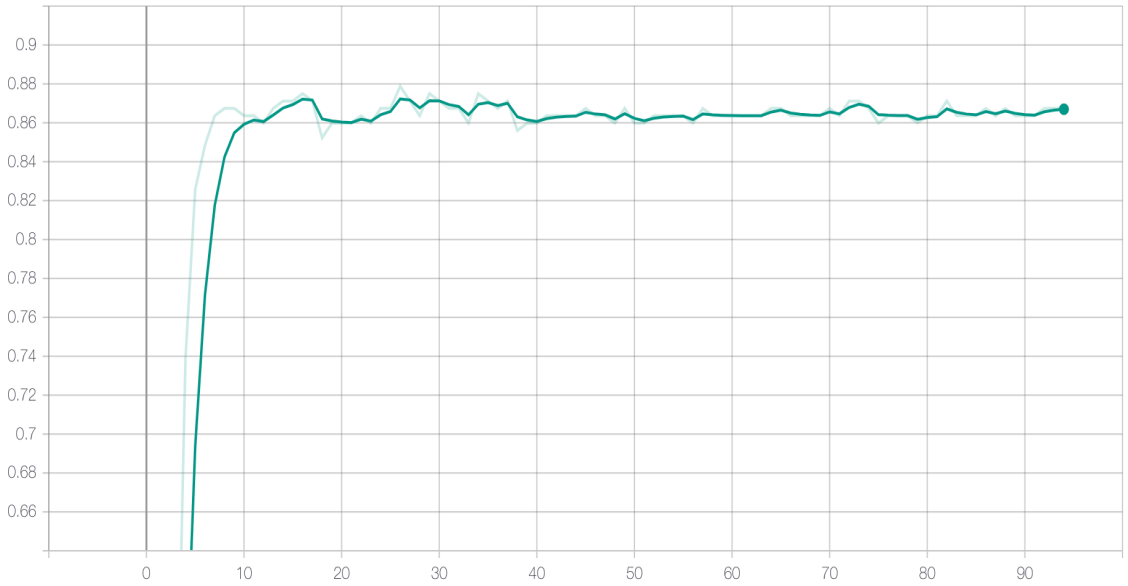


Şekil 3. 20 ZfNet VS1 TOP-1 Validasyon Verisi Doğruluk Tahmini

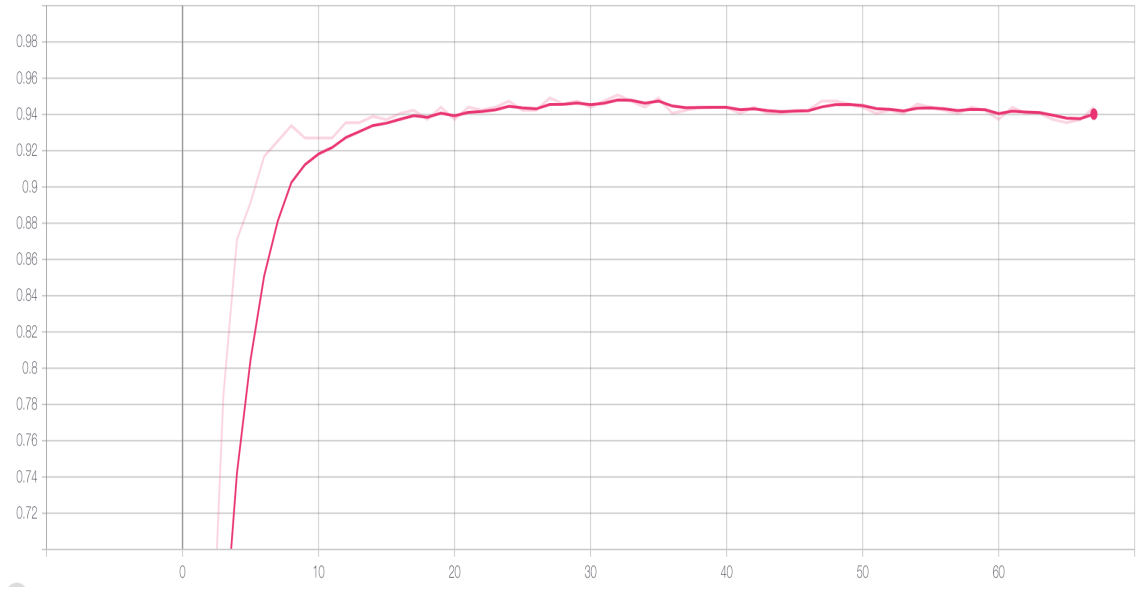


Şekil 3. 21 ZfNet VS1 TOP-2 Validasyon Verisi Doğruluk Tahmini

Şekil 3.20 ve Şekil 3.21 ‘de ZfNet modeli için veri seti 1 (VS1) ile yakalanan TOP-1 ve TOP-2 değerlerinin grafikleri gösterilmektedir.



Şekil 3. 22 ZfNet VS2 TOP-1 Validasyon Verisi Doğruluk Tahmini



Şekil 3. 23 ZfNet VS2 TOP-2 Validasyon Verisi Doğruluk Tahmini

Şekil 3.22 ve şekil 3.23 'da ZfNet modeli için veri seti 2 (VS2) ile yakalanan TOP-1 ve TOP-2 değerlerinin grafikleri gösterilmektedir.

Tablo 3. 4 ZfNet Model Özeti

Layer (type)	Output Shape	Param #
conv2d_19 (Conv2D)	(None, 109, 109, 96)	14208
max_pooling2d_11 (MaxPooling	(None, 55, 55, 96)	0
batch normalization 8 (Batch	(None, 55, 55, 96)	384
conv2d 20 (Conv2D)	(None, 13, 13, 256)	614656
max_pooling2d_12 (MaxPooling	(None, 7, 7, 256)	0
batch_normalization_9 (Batch	(None, 7, 7, 256)	1024
conv2d_21 (Conv2D)	(None, 5, 5, 512)	1180160
conv2d 22 (Conv2D)	(None, 3, 3, 1024)	4719616
conv2d_23 (Conv2D)	(None, 1, 1, 512)	4719104
max_pooling2d_13 (MaxPooling	(None, 1, 1, 512)	0
flatten 1 (Flatten)	(None, 512)	0
dense_9 (Dense)	(None, 4096)	2101248
dense 10 (Dense)	(None, 4096)	16781312
dense_11 (Dense)	(None, 6)	24582

Total params: 30,156,294
Trainable params: 30,155,590
Non-trainable params: 704

Tablo 3.4 ‘de ZfNet Modelinin 30 milyon 156 bin 294 adet parametre ile eğitil ği görülmektedir. Tüm devirler incelendiğinde, model aşırı öğrenmeye girmeden yakala nan en iyi kayıp/başarı oranları baz alınarak doğruluk tahmin değerleri çıkarılmıştır. VS 1 ile eğitildiğinde TOP-1 doğruluk tahmin oranı validasyon verisi için %81,99, eğitim v erisi için %95,53, TOP-2 doğruluk tahmin oranı validasyon verisi için %96,55, eğitim v erisi için %99,79 çıkmıştır. VS2 ile eğitildiğinde ise bu oranlar TOP-1’de validasyon ver isi için %86,74, eğitim verisi için %97,03, TOP-2’de validasyon verisi için %94,74, eğit im verisi için %98,92 olmuştur.

Tablo 3. 5 VS1 Kullanılmış Modellerin Karşılaştırılması

Model	Resim Boyutu	Model Boyutu	TOP-1 Validasyon Başarısı	TOP-2 Validasyon Başarısı	Katman Sayısı	Parametre Sayısı
VGG-16	224x224	1.61 GB	%76,63	%91,19	16	134,285,126
VGG-19	224x224	1.68 GB	%75,86	%91,95	19	139,594,822
AlexNet	224x224	337 MB	%83,52	%97,70	9	28,043,486
ZfNet	224x224	337 MB	%81,89	%96,55	8	30,156,294

Tablo 3.5'de VS1 ile eğitilmiş modellerin karşılaştırılması yapılmıştır. Bu karşılaştırma ile TOP-1 ve TOP-2 için en yüksek doğruluk oranının AlexNet ile yakalanıldığı görülmektedir.

Tablo 3. 6 VS2 Kullanılmış Modellerin Karşılaştırılması

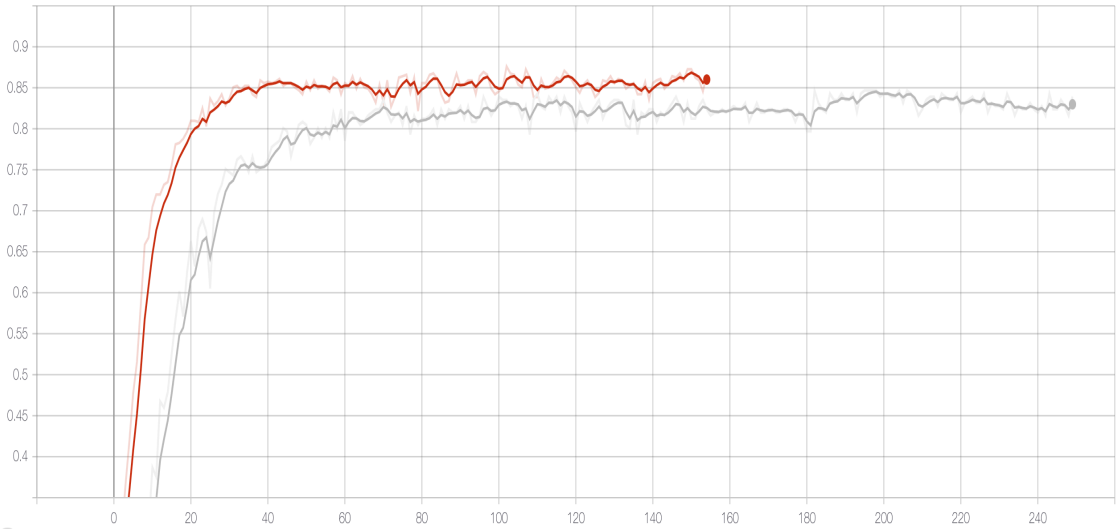
Model	Resim Boyutu	Model Boyutu	TOP-1 Validasyon Başarısı	TOP-2 Validasyon Başarısı	Katman Sayısı	Parametre Sayısı
VGG-16	224x224	1.62 GB	%80,48	%93,89	16	134,285,126
VGG-19	224x224	1.70 GB	%76,52	%94,23	19	139,594,822
AlexNet	224x224	350 MB	%87,50	%97,26	9	28,043,486
ZfNet	224x224	345 MB	%86,74	%94,74	8	30,156,294

Tablo 3.6'da VS2 ile eğitilmiş modellerin karşılaştırılması yapılmıştır. Bu karşılaştırmada TOP-1 ve TOP-2 için en yüksek doğruluk oranının AlexNet ile yakalanıldığı görülmektedir.

Tablo 3. 7 AlexNet Veri Setlerine Göre Başarı Oranları

Veri Seti	TOP-1 Validasyon Verisi Başarısı	TOP-1 Eğitim Verisi Başarısı	TOP-2 Validasyon Verisi Başarısı	TOP-2 Eğitim Verisi Başarısı
VS 1	%83,52	%98,08	%97,70	%99,02
VS 2	%87,50	%97,01	%97,26	%99,53

Veri Seti 1'den (VS 1) veri seti 2'ye geçildiğinde, model eğitimde kullanılan görüntü sayısının artmasının, modellerin başarı oranına pozitif etkisi olduğu gözlemlenmektedir. Tablo 3.7 ve şekil 3.24'de AlexNet modelinde VS1'den VS2'ye geçildiğinde tahmin doğruluk oranın arttığı görülmektedir.



Şekil 3. 24 AlexNet VS1 ve VS2 Karşılaştırması

BÖLÜM 4. SONUÇ

Bu çalışmada insanları, cinsiyet ve etnik kökenlerine göre sınıflandırma yapan bir sistem oluşturmak amacı ile 4 farklı derin öğrenme modeli kullanılmıştır. Tüm modeller UTKFace veri setinden elde edilen 2000 adet görüntüye sahip veri seti 1 (VS 1) ve 6000 adet görüntüye sahip veri seti 2 (VS 2) ile eğitilmiş, sonucunda Asya, Afrika, Hindistan etnik kökenlerinden kadın ve erkekleri sınıflandıracak şekilde validasyondan geçmiştir.

Çalışmada kullanılan Alexnet, VGG-16, VGG-19, ZfNet modelleri farklı performanslar göstermiş olup iki farklı veri seti ve TOP-1, TOP-2 tahmin doğruluk oranlarının karşılaştırmaları yöntem kısmında yapılmıştır. Model eğitiminde kullanılan görüntü sayısının artmasının, tüm modeller için başarı oranını arttırdığı görülmüştür. AlexNet'in, kayıp/başarı oranı optimal olup, başarı oranının en yüksek olduğu noktada kayıp değeri sıfıra en çok yakınsayan model olduğu ve kullanılan her iki veri setinde TOP-1 ve TOP-2 değerleri en yüksek olan model olduğu gözlemlenmiştir. Bu çalışmadan alınan çıktılara göre düşük ve orta boyutlu veri setlerinde, VGG-16,VGG-19 ve ZfNet 'e göre başarı oranı daha yüksek olan AlexNet modeli ile kurulan sistem tavsiye edilen yöntemdir.

KAYNAKÇA

[1] A. Acien, A. Morales, R. Vera-Rodriguez, I. Bartolome ve J. Fierrez, «Measuring the Gender and Ethnicity Bias in Deep Models for Face Recognition,» Biometrics and Data Pattern Analytics (BiDA) Lab - ATVS, EPS, Universidad Autonoma de Madrid C/ Francisco Tomas y Valiente 11, 28049 Madrid, Spain.

[2] T. Kanade, «Picture Processing System by Computer Complex and Recognition of Human Faces. Ph.D Thesis, Department of Information Science,» Kyoto University, 1973.

[3] R. J. ve H. R., «A survey of face recognition techniques,» *JIPS,* pp. 41-68, 2009.

[4] N. M., «Eye spacing measurement for facial recognition,» %1 içinde *SPIE*, 1985.

[5] H. P. Graf, T. Chen, E. Petajan ve E. Cosatto, «Locating faces and facial parts, in International Workshop on Automatic Face and Gesture-Recognition,» pp. 41 - 46.

[6] R. D., «Generalized symmetry transforms: attentional mechanisms and face recognition,» el-Aviv University, PhD. Thesis, technical report, 1994.

[7] G. J. ,. Y. P. N. Cox I. J., «Feature-based face recognition using mixture-distance,» %1 içinde *IEEE Conference*, 1996.

[8] F. J.-M. ,. K. N. ,. C. v. d. M. Wiskott L., «Face Recognition by Elastic Bunch Graph Matching,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 775 - 779, 1997.

[9] B. R. J., «Mechanisms of Human Facial Recognition,» *International Journal of Man-Machine Studies,* cilt 15, pp. 137 - 178.

[10] H. J., «Detection Strategies for face recognition using learning and evolution,» George Mason University, Fairfax, Virginia, Ph. D. Dissertation., 1998.

[11] K. M. Sirovich L., «Low-dimensional Procedure for the Characterization of Human Faces, Journal of the Optical Society of America,» *Optics, Image Science, and Vision,* cilt 4, pp. 519 - 524, 1987.

[12] D. R. C. Jain A. K., «Algorithms for Clustering Data,» New Jersey: Prentice-Hall, 1988.

[13] F. K., «Introduction to Statistical Pattern Recognition,» second ed. Boston, MA: Academic Press, 1990.

[14] P. A. Turk M., «Face Recognition Using Eigenfaces,» %1 içinde *IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[15] P. A. Turk M., «Eigenfaces For Recognition,» *Journal of Cognitive Neuroscience,* cilt 3, pp. 71 - 86, 1991.

[16] A. Y. U. S. Moses Y., «Face recognition: the problem of compensating for changes in illumination direction,» %1 içinde *European Conf. Computer Vision, 286-296*.

[17] Susanqq, «UTKFace,» [Çevrimiçi]. Available: https://susanqq.github.io/UTKFace/.

[18] A. Rosebrock, «pyimagesearch.com,» 3 4 2017. [Çevrimiçi]. Available: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/. [Erişildi: 5 10 2019].

[19] J. Brownlee, «machinelearningmastery.com,» 5 7 2019. [Çevrimiçi]. Available: https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/. [Erişildi: 20 10 2019].

[20] Prabhu, «medium.com,» 4 3 2018. [Çevrimiçi]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148. [Erişildi: 10 9 2019].

[21] T. Ergin, «medium.com,» 2 9 2018. [Çevrimiçi]. Available: https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad. [Erişildi: 10 9 2019].

[22] A. Budhiraja, «medium.com,» 15 12 2016. [Çevrimiçi]. Available: https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5. [Erişildi: 13 9 2019].

[23] D. Gupta, «analyticsvidhya.com,» 23 9 2017. [Çevrimiçi]. Available: https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/. [Erişildi: 15 10 2019].

[24] A. Rosebrock, «pyimagesearch.com,» 24 9 2018. [Çevrimiçi]. Available: https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/. [Erişildi: 15 10 2019].