

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

**DERİN ÖĞRENME İLE FOTOĞRAFTAN YAŞ VE
CİNSİYET TESPİTİ**

YÜKSEK LİSANS TEZİ

Gül GÜNDÜZ

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ
Tez Danışmanı : Prof. Dr. İsmail Hakkı CEDİMOĞLU

Haziran 2019

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**DERİN ÖĞRENME İLE FOTOĞRAFTAN YAŞ VE
CİNSİYET TESPİTİ**

YÜKSEK LİSANS TEZİ

Gül GÜNDÜZ

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 11.06.2019 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

**Prof. Dr.
İsmail H. CEDİMOĞLU
Jüri Başkanı**

**Dr. Öğr. Üyesi
Mümtaz İPEK
Üye**

**Dr. Öğr. Üyesi
Ahmet KARACA
Üye**



BEYAN

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

Gül GÜNDÜZ

11.06.2019

TEŞEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, yardımını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Prof. Dr. İsmail Hakkı CEDİMOĞLU'na teşekkürlerimi sunarım.

Araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında anlayış ve yardımlarını esirgemeyen Sakarya Üniversitesi Endüstri Mühendisliği Arş. Gör. Dr. M. Raşit CESUR'a teşekkür ederim.

Ayrıca her zaman beni destekleyen aileme ve Sakarya Üniversitesi Endüstri Mühendisliği öğretim elemanlarına da teşekkür ederim. Yaptığım bu tez çalışmasının konuyla ilgilenen lisans, yüksek lisans öğrencilerine ve bu konuda çalışmak isteyenlere bir rehber olacağını umuyorum.

İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	iv
ŞEKİLLER LİSTESİ	v
TABLolar LİSTESİ	vi
ÖZET	vii
SUMMARY	viii
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
LİTERATÜR ARAŞTIRMASI.....	4
2.1. Derin Öğrenme ile İlgili Yapılan Çalışmalar.....	4
2.2. Tezin Kapsamı ve Amaçları.....	5
BÖLÜM 3.	
BİLİMSEL ARKA PLAN.....	6
3.1. Derin Öğrenme Kavramı	6
3.2. Derin Öğrenme Mimarisi.....	7
3.2.1. Konvolüsyonel sinir ağları	8
3.3. Derin Öğrenme Modelleri.....	11
3.4. Derin Öğrenme Kütüphaneleri	14
3.4.1 Caffe derin öğrenme kütüphanesi.	15
3.5. Derin Öğrenme Uygulamalarının Hiperparametreleri.....	18

BÖLÜM 4.

YAPILAN UYGULAMALI ÇALIŞMALAR.....	21
4.1. Derin Öğrenme ile Çalıştırılan Yaş-Cinsiyet Modelleri.....	21
4.2. Yeni Bir Yaş ve Cinsiyet Tahmini Uygulaması.....	26

BÖLÜM 5.

TARTIŞMA VE SONUÇ	31
KAYNAKLAR	34
EKLER	37
ÖZGEÇMİŞ	45

SİMGELER VE KISALTMALAR LİSTESİ

BVLC	: Berkeley Vision and Learning Center
DAE	: Derin Oto-Kodlayıcılar
DBN	: Derin İnanç Ağları
GPU	: Grafik İşlemci Birimi
CNN	: Konvolüsyonel Sinir Ağları
RBM	: Sınırlı Boltzman Makinesi
ReLu	: Düzleştirilmiş Doğrusal Birim Katmanı
RNN	: Tekrarlayan Sinir Ağları

ŞEKİLLER LİSTESİ

Şekil 1.1. Veri setinden kesit ve etiketler.....	2
Şekil 3.1. 2 boyutlu konvolüsyonel sinir ağları gösterimi ve filtreleme	8
Şekil 3.2. Kenar bulma ve kaydırma işlemi gösterimi	9
Şekil 3.3. Alex Net Mimarisi.....	12
Şekil 3.4. Le Net Mimarisi.....	12
Şekil 3.5. ZF Net Mimarisi.....	13
Şekil 3.6. VGG-16 Mimarisi.....	13
Şekil 3.7. GoogLeNet	13
Şekil 3.8. ResNet.....	14
Şekil 4.1. Çalışmada kullanılan DeepID yapısı	22
Şekil 4.2. Çalıştırılan model sonrası elde edilen görüntüler.....	22
Şekil 4.3. Modelin CNN mimarisine göre katman yapısı	24
Şekil 4.4. Modelin çalıştırılması ile elde edilen örnek resim	24
Şekil 4.5. Modelin mimari yapısı.....	25
Şekil 4.6. Önerilen model mimari yapısı	27
Şekil 4.7. Çalıştırılan VGG-16 modelin cinsiyet başarımları ve kayıp grafiği.....	29
Şekil 4.8. Çalıştırılan VGG-16 modelin yaş başarımları ve kayıp grafiği.....	30
Şekil 5.1. Deneme resimleri için predict sonuçları	32

TABLÖLAR LİSTESİ

Tablo 3.1. Derin Öğrenme Kütüphaneleri	15
Tablo 4.1. Batch Boyutuna Göre En İyi Model Performansı	23
Tablo 4.2. Modellerin Mimari Yapılarının Karşılaştırılması	28
Tablo 4.3. Hazırlanan Veri Seti İle Çalıştırılan Model Sonuçları	28



ÖZET

Anahtar kelimeler: Derin öğrenme, görüntü işleme, yaş tahmini, cinsiyet tahmini

Büyük verilerin büyük hızlarla işlendiği çağımızda milyarlarca veriden farklı parametreler çıkararak çeşitli problemlerin çözümüne kolaylık getirmek için derin öğrenme algoritmaları kullanılmaktadır. Bu çalışmada, Mevcut veri setlerinde bulunan kadın, erkek, yaşlı, genç, çocuk, bebek fotoğraflarının derin öğrenme algoritmaları ile yaş ve cinsiyetlerini tespit etmek amaçlanmıştır. Bu tahminleme algoritmasını gerçekleştirmek için çeşitli derin öğrenme kütüphanelerinden faydalanılmış ve derin öğrenme modelleri ile bir model geliştirilerek diğer modellerle kıyaslanması sağlanmıştır.

Tez, derin öğrenme algoritmalarını kullanarak yaş ve cinsiyet tespiti yapmayı ayrıntılı bir şekilde araştırmaktadır. Derin öğrenme ile yaş ve cinsiyet tespiti alanında yapılan ve yapılacak olan literatür çalışmalarına katkı sağlamayı hedeflemektedir.

Çalışmada öncelikli olarak literatürde yaş ve cinsiyet tahminini derin öğrenme kullanarak gerçekleştiren çalışmalar uygulanmıştır. Uygulamada kullanılan veri seti kadın ve erkek fotoğraflarından oluşturulmuştur. Her fotoğraf kişi cinsiyetine ve yaşına göre etiketlenmiştir. Bu veri seti vikipedi görüntülerinden derlenmiştir, 3170 eğitim verisi ve 318 test verisi içermektedir. Çalıştırılan üç farklı model sonuçları karşılaştırılmıştır. Çalışma, derin öğrenme algoritmalarını kullanarak cinsiyet tahmini yapmayı ayrıntılı bir şekilde açıklamaktadır ve yapılacak olan literatür çalışmalarına yol göstermeyi planlamaktadır.

AGE AND GENDER DETECTION FROM IMAGES WITH DEEP LEARNING

SUMMARY

Keywords: Deep Learning, image processing, age estimation, gender estimation

In our age, where large data is processed at great speeds, deep learning algorithms are used to facilitate the solution of various problems by extracting different parameters from billions of data. In this study, it is aimed to determine the age and sex of the female, male, old, young, child and baby photographs in the existing datasets with deep learning algorithms. In order to realize this estimation algorithm, various deep learning libraries were used and a model was developed with deep learning models and compared with other models.

The thesis explains in detail the age and gender determination using deep learning algorithms. The aim of this thesis is to contribute to the literature studies in the field of age and gender determination with deep learning.

In the study, firstly, the studies which have performed age and gender estimation by using deep learning have been applied in the literature. The data set used in the application is composed of male and female photographs. Each photo is labeled according to the gender and age of the person. This data set is compiled from wikipedi images, contains 3170 training data and 318 test data. The results of three different models were compared. The study describes in detail the gender prediction using deep learning algorithms and plans to guide the future studies to be carried out.

BÖLÜM 1. GİRİŞ

Sosyal bir varlık olan insanlar, çevreyle etkileşim içindedir. Yaş ve cinsiyet ise sosyal etkileşimlerde temel rol oynamaktadır. Genç ve yaşlı kişilere hitap şeklinin farklılığı, cinsiyet farklılığına göre yine hitabetin değişmesi, bazı kültürlerde cinsiyet öğelerine göre dilbilgisi kurallarının değişmesi, kişinin yaş aralığına göre selamlama özelliklerinin farklılık göstermesi bu sosyal etkileşimlerden sadece birkaçıdır.

Son yıllarda yaş ve cinsiyet tahmini uygulamaları çeşitli teknolojilerle karşımıza çıkmaktadır. Bu konuyla ilgili çeşitli örnekler mevcuttur. Bu örneklerin en bilinenlerinden biri olan Microsoft'un fotoğraflar üzerinden yaş tahminlemesi üzerine uygulama geliştiren internet sitesi "How Old Do I Look?" oldukça popüler olarak kullanılmaktadır. Bu uygulama sitesi, fotoğraflar üzerinden yüzleri algılayarak, kişilerin cinsiyetini tahmin etmekte ve yaşını söylemekte, eğlence amaçlı bir site olarak kullanılmaktadır[1]. Bir diğer çarpıcı örneği ise; Xiaomi Mi 6'nın önünde 8 megapiksel çözünürlüklü, otomatik yüz güzelleştirme filtreli ve yapay zekâyı kullanan, bu sayede cinsiyet ve yaş tahmini yapan bir kamerasının bulunmasıdır [2]. Bu uygulama örnekleri çoğaltılabilir ve sosyal medya kullanan toplum yapısının artmasıyla yaş ve cinsiyet tahminine ilginin artacağı öngörülebilir.

Bu çalışmada, büyük veri (big data) ve hayatımızı kolaylaştıran derin öğrenme ağ yapısıyla, milyonlarca fotoğraftan yaş ve cinsiyet tespiti yapan modeller kullanılarak elde edilen tespitler analiz edilmiştir. Çalışmanın amacı, derin öğrenme algoritmalarını kullanarak, mağazalarda ürünlerle ilgilenen insanların, bir mağazada alışveriş yapmayı tercih eden insanların, alışveriş merkezlerindeki müşterilerin, sosyal medya platformlarındaki kullanıcıların, siyasilerin mitinglerindeki kişilerin başta

olmak üzere, görüntüleri üzerinden cinsiyet tahmini demografisini tespit etmeye yardımcı olacak bilgiler ve yöntemler sunmaktır. Aynı zamanda veri setinin modele uygunluğunun başarıma olan etkisi vurgulanmaktadır. Bu amaçla çalışmaya uygun bir veri seti hazırlanmıştır. Veri seti kadın(female) ve erkek(male) olmak üzere 2 sınıfa ayrılmış ve her görüntü cinsiyete göre ve yaş bilgisine göre etiketlenmiştir. Veri setinde bulunan, yaşları 34-35 olan erkek resimleri ile yaşları 16-17 olan kadın resimleri Şekil 1.1'deki gibi etiketlenerek data set hazırlanmıştır. Vikipedi veri setinin tamamına “<https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/>” linkinden erişilebilir. Derin öğrenme algoritmalarından VGG-16 kullanarak yapılan bir çalışmada yaş tahminlemek için de Vikipedi veri setinden faydalanılmıştır [3].



Şekil 1.1. Veri Setinden kesit ve etiketler

Çalışmanın hazırlanması için ubuntu linux işletim sistemi üzerinde, python dilinde gerekli kodlar yazılıp derin öğrenmede çeşitli kütüphaneler kullanılarak, yaş-cinsiyet veri setleri üzerinden gerekli tahminlemeler yapılmıştır. Projede python dilinin seçilme nedeni hızlı geliştirme olanağı ve topluluk desteğine sahip olmasıdır. Linux işletim sistemlerinde Python programlama dili default olarak bulunmaktadır. Default terimi açıklanacak olursa; yani kurulu olarak Linux sistemine Python dilinin gelmesi olarak açıklanabilir. Bu çalışmada kullanım için seçilme nedenlerinden birisi de bu default düzenin sağladığı esnekliktir [4]. Yapılan çalışmada, önceden hazırlanan literatürdeki modellerin çalıştırılması ile tez için hazırlanan modeli çalıştırıp karşılaştırılması sağlanarak analizler yapılmıştır. Sonuç olarak tezin diğer modellerden farkı ve literatüre etkisi üzerinde durulmuştur. Tezin organizasyon

yapısından bahsedilirse; Bölüm 1’de teze giriş yapılmış, yaş ve cinsiyet tahmini konusunun neden seçildiği üzerinde durulmuştur. Bölüm 2’de gerekli literatür araştırması yapılarak, seçilen tez konusunun kapsamı ve amacından bahsedilmiştir. Bölüm 3’te tez çalışmasının bilimsel arka planı irdelenmiştir. Bölüm 4’te çalışmadaki veri setleri, kullanılan metotlar ve çalıştırılan modeller kıyaslanarak incelenmiştir. Bölüm 5’te ise sonuçlar verilerek tartışmalar yapılmıştır.



BÖLÜM 2. LİTERATÜR ARAŞTIRMASI

2.1. Derin Öğrenme İle İlgili Yapılan Çalışmalar

Derin öğrenme, bilgisayarların deneyimden ders almasını ve dünyayı kavramların hiyerarşisi açısından anlamasını sağlayan bir makine öğrenmesi olarak tanımlanmaktadır [5]. Literatürde ve uygulamada derin öğrenmenin kullanıldığı çok sayıda çalışma yapılmıştır. Doğal dil işleme, görüntü ve video işleme, biyomedikal sinyal ve görüntü işleme, nesne tanıma, robotik, kimya, reklam, finans, arama motorları, otonom araç sistemleri gibi konularda derin öğrenme uygulamaları yaygın bir şekilde kullanılmakla birlikte çeşitli uygulamalarla denenerek geliştirilmektedir[6]. Cinsiyet sınıflandırma için kullanılan ilk metotlardan biri, yapay sinir ağlarının az sayıda insan yüzü görüntüsünü içeren veri setiyle eğitilmesine dayanmaktadır [7].Derin öğrenme sistematik olarak verinin temsilinden öğrenmeye dayanmaktadır. Bu tez çalışmasında ele alınan yaş ve cinsiyet tahmini ile ilgili akademik çalışmalar son yıllarda popülerlik kazanmıştır. Bu çalışmaların bazıları detaylı incelenmiş ve yapılan tez çalışmasının farklılıklarının belirlenmesi için yol göstermiştir. Chuan-Hui Ju ve Yun-Hong Wang çalışmasında, yüz görüntüsünün ve regresyonun yerel özelliklerine dayalı otomatik yaş tahminini irdelemiştir. Bu makalede, yaşlanma sürecine duyarlı insan yüzü görüntülerinin bazı bölgelerinden yerel özellikleri kullanarak insan yaşını tahmin etmek için bir çerçeve geliştirilmiştir. Adaboost algoritması, ihtiyaç duyulan bölgeleri bulmak için uygulanır ve LBP özellikleri, yaş tahmini için sağlam bir kuadratik regresyon işlevini modellemek için çıkarılır [8]. Yuan Dong, Yinan Liu ve Shiguo Lian hazırladıkları makalede derin öğrenme algoritmasına dayalı otomatik yaş tahminini çalışmıştır. . Bu çalışmada, yüz görüntüsünün bir yaş aralığını temsil eden bir etikete atanmış olduğu yaş sınıflandırma görevine odaklanılmaktadır. Yaş sınıflandırma görevi için derin bir öğrenme temelli bir çerçeve önerilmiştir. Önerilen algoritmada, Derin Konvolüsyel

Sinir Ağları (Deep CNN), üst düzey karmaşık yaşa bağlı görsel özellikleri elde etmek ve giriş yüz görüntüsünün yaş aralığını tahmin etmek için kullanılır. Yüz imzalı yüz görüntülerinin olmaması nedeniyle, Deep ConvNets'i eğitmek için transfer öğrenme stratejisi kullanılmıştır. Ek olarak, sıralı bir dizi oluşturan etiketler arasındaki ilişkileri tanımlamak için, yaş sınıflandırma görevinin eğitim sürecinde yeni bir kayıp fonksiyonu tanımlar. Deneyler, İnsan Gruplarının yaygın olarak kullanılan bir yaş tahmin veri seti görüntüleri üzerinde gerçekleştirilmiştir. Deneysel sonuçlar, önerilen algoritmamızın son teknoloji yöntemlere karşı mükemmel performansını göstermektedir [9]. Gil Levi ve Tal Hassner konvolüsyonel sinir ağlarını kullanarak yaş ve cinsiyet sınıflaması yaptığı çalışmada önemli ilerlemeler kaydetmiştir. Bu makalede ise, derin-konvolüsyonel sinir ağları kullanılarak, temsilleri öğrenerek, yüz tanıma ile ilişkili performansta önemli bir artış elde edilebileceği gösterilmektedir. Bu amaçla, öğrenme verilerinin miktarı sınırlı olduğunda bile kullanılabilen basit bir kırımlı net mimarisi önerilmiştir. Son zamanlardaki Adience kriterlerine ilişkin yöntemi yaş ve cinsiyet tahmini açısından değerlendiren çalışma güncel yöntemlerden önemli ölçüde daha iyi performans gösterdiğini belirtmektedir [10]. Tarafımdan yazılan tezin ise kapsam ve amaçları belirtilerek literatüre katkısı belirtilmiştir. Filtrelenmemiş yüzlerin yaş ve cinsiyet tahmini yapan çalışmada “vahşi” şartlarda zorlu olarak elde edilen yüz görüntülerinden yaş ve cinsiyet gibi yüz özelliklerinin tahmini ile ilgilidir. Sistemin yüz özellik tahmini için SVM yaklaşımı kullanılmıştır ve derin inanç ağlarıyla öğrenme tekniklerinden esinlenilmiştir [11].

2.2. Tezin Kapsamı ve Amaçları

Tezin kapsamında, yaş ve cinsiyet tahmini modellerinin çalıştırılması ve alışveriş merkezlerine, mağazalara talep gösteren insanların demografisi ile özel ürünlere ilgi gösteren müşteri yaş ve cinsiyet demografisini çıkaracak bilgi ve çalışmalar kazandırılmaya çalışılmıştır. Yaş ve cinsiyet tespit etme amaçlı geliştirilen modellerin analizi incelenmektedir. Bu çalışmadaki amaç ise literatürde yaş ve cinsiyet tahmini ile ilgili çalışılan modelleri irdeleyerek yeni bir veri setiyle hazırlanan modeli bu çalışmalarla kıyaslamaktır.

BÖLÜM 3. BİLİMSEL ARKA PLAN

3.1. Derin Öğrenme Kavramı

Derin öğrenme kavramı, çoklu soyutlama yapısı ile verinin temsil şeklini öğrenmek amacıyla bir araya getirilen çoklu işleme katmanlarından oluşan bir yapıdır [12]. Derin öğrenme teknolojisi; makinelerin dünyayı algılama ve anlamasına yönelik yapay zekâ geliştirmede en popüler yaklaşımlardan biridir. Son yıllarda bu konu üzerine çalışmaların yoğunlaştığı bir alandır. Derin öğrenme, çoklu soyutlama seviyelerine sahip verilerin gösterilmesini öğrenmek için çoklu işleme katmanlarından oluşan hesaplama modellerine izin vermektedir. Derin öğrenme, bir makinenin her katmandaki gösterimi önceki katmandaki gösterimden hesaplamak için kullanılan dahili parametrelerini nasıl değiştirmesi gerektiğini göstermek için geri yayılma algoritmasını kullanarak karmaşık veri kümelerinde karmaşık yapıyı keşfetmektedir [12]. Derin ağlar, tek yöntemlerin (örneğin, metin, görüntü veya ses gibi) denetimsiz özellik öğrenimine başarıyla uygulanmaktadır [13]. Endüstri ve akademik çevrelerdeki veri bilimciler görüntü sınıflandırma, video analizi, konuşma tanıma ve doğal dil öğrenme süreci dâhil, çeşitli uygulamaları yapmak için makine öğrenmesi çalışmalarında GPU'ları (Grafik İşlemci Ünitesi) kullanmaktadır. Büyük miktarlarda etiketlenmiş eğitim verilerinden özellik saptama yapabilen sistemler oluşturmak amacıyla ileri teknoloji, çok seviyeli derin sinir ağlarının kullanılması olarak ifade edilen derin öğrenme, çok önemli sayıda yatırım ve araştırmanın yapıldığı bir saha haline gelmiştir [14]. Önemli araştırma kuruluşları derin öğrenme ve makine öğrenmesi alanında çeşitli araştırma, geliştirme çalışmaları yapmaktadır. Bu kuruluşların yanısıra dünyanın en büyük internet şirketlerinden bazıları araştırma ve ürünlerinde derin öğrenme ile gelişmeler yapabilmek için GPU'ları kullanmaktadır. GPU denilen işlemci ünitesi, çok büyük miktardaki eğitim verisi sayesinde makine öğrenmesinde ve derin öğrenmede yeni bir eğilime yol açmıştır. GPU'lar, çok büyük

eğitim setleri ile derin sinir ağlarını çok daha kısa sürelerde ve çok az veri merkezi altyapısıyla eğitmek için kullanılmaktadır. GPU'lar aynı zamanda, çok daha fazla veri hacmi ve daha az güç ile altyapıyı destekleyerek, veri bulutu içinde sınıflandırma ya da tahminleme için eğitilmiş makineyle öğrenme modellerini çalıştırmak amacıyla kullanılmaktadır [14]. Bu tez çalışmasında derin öğrenme uygulamaları için kullanılan bilgisayarın özellikleri; Intel Core i5 işlemci tipi, ubuntu 18.04.1 işletim sistemi, NVIDIA GeForce GT820M ekran kartı ve 8 GB RAM belleği olmak üzere 'Dell İnsipiron 3542' adlı aygıtla çalışılmıştır.

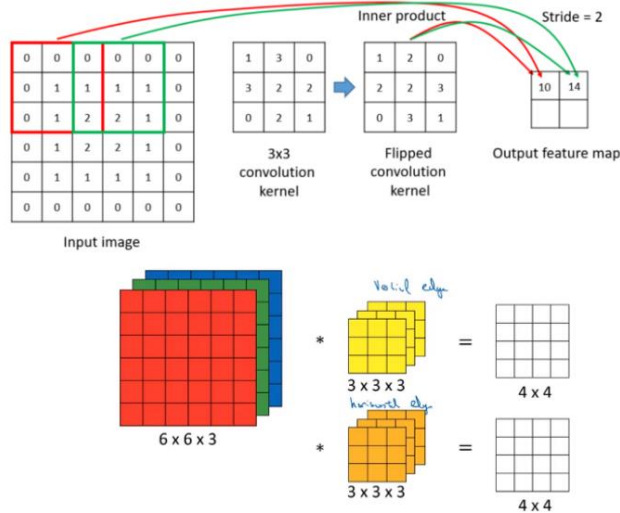
3.2. Derin Öğrenme Mimarisi

Eğer bir derin öğrenme uygulayıcısı, derin öğrenme dünyasına dünyasına girmek istiyorsa, zaten sinir ağları ile tanışmış olabilir. Biyolojik sinir ağlarından ilham alan sinir ağları, karmaşık, çok katmanlı hesaplama problemlerini çözme konusunda oldukça faydalıdır. Derin öğrenme, yüz ve konuşma tanıma, doğal dil işleme, makine çevirisi ve daha fazlasını içeren birçok yüksek profilli araştırma alanında oldukça göze çarpmaktadır [15]. Derin öğrenme mimarisinde yer alan 6 mimariden bahsedilmiş, tezin genelinde kullanılan “Konvolüsyonel Sinir Ağları (CNN)” mimarisi daha detaylı ve ayrı bir başlık altında incelenmiştir. CNN bilgisayarlı görme işlevlerinde etkilidir ve bu tezin modellenmelerinde ayrıntılı kullanılan bir mimaridir. “Tekrarlayan Sinir Ağları” (RNN); birimler arası verilerin yönlendirilmiş bir döngü oluşturduğu yapay sinir ağıdır. RNN görüntü tanımlayıcıları kullanmak için diğer mimarilerle birlikte kullanılmaktadır. “Uzun-Kısa Vadeli Hafıza Ağları”; değerleri rastgele aralıklarla hatırlayan bir tekrarlayan sinir ağı (RNN) mimarisidir. Bu mimari yapısı konuşma ve metin işleme çalışmalarında kullanıldığında iyi sonuçlar vermektedir. “Sınırlı Boltzman Makineleri” (RBM); girdi seti üzerinde olasılık dağılımlarını öğrenebilen 2 katmandan (görünür katman ve gizli katman) oluşan bir yapay sinir ağı yapısıdır. RBM; sınıflandırma, özellik öğrenimi ve modelleme için uygun bir ağıdır. “Derin İnanç Ağları” (DBN) ise RBM'lerin yığını ve oto kodlayıcılarla bileşimi olarak görülür. DBN, görüntü tanıma uygulamalarında kullanılmaktadır. “Derin Oto Kodlayıcılar” (DAE); denetimsiz öğrenme için ve

verinin üretken modellerini öğrenmek için kullanılan, girdi katmanı verilerini, çıktı katmanına kopyalayan bir yapıdır [16].

3.2.1. Konvolüsyonel sinir ağları

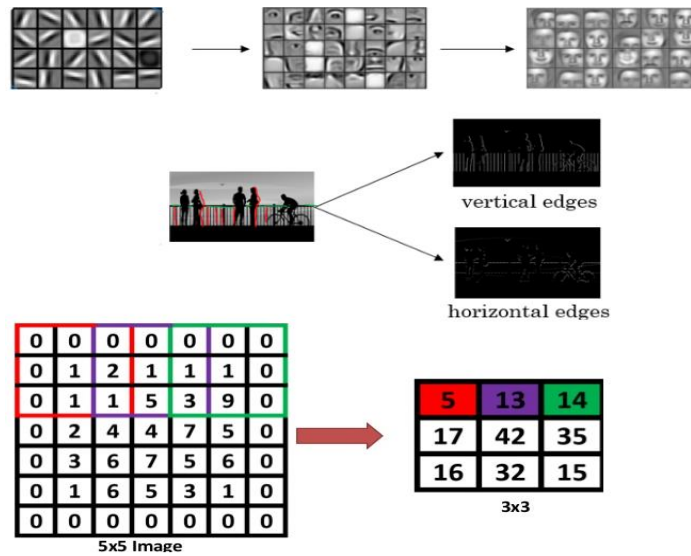
CNN, derin öğrenme kavramına ait temel mimari yapısıdır. İlk CNN 1988 yılında Yann LeCun tarafından oluşturulmuştur. 1998'lere kadar sürekli geliştirilen LeNet ilk CNN mimarisidir [17]. CNN'ler görüntüleri girdi olarak almak için tasarlanmış yapılardır ve bilgisayarlı görmede etkili bir şekilde kullanılmaktadırlar. CNN, bir veya birden çok konvolüsyonel katman ve standart çok katmanlı bir sinir ağı gibi bir veya daha fazla tamamen bağlı katmanlardan oluşmaktadır [18].



Şekil 3.1. İki boyutlu konvolüsyonel sinir ağları gösterimi ve filtreleme işlemi [20]

İki boyutlu konvolüsyonda Şekil 3.1'deki gibi, İki boyutlu bilgi için filtrenin x ve y eksenine göre simetrisi alınmaktadır. Tüm değerler, matris formatına göre eleman eleman çarpılmaktadır ve tüm değerlerin toplamı çıkış matrisinin ilgili elemanı olarak kaydedilmektedir. Buna çapraz korelasyon ilişkisi de denir. Giriş verisi kanallı iken bu işlem basitçe yapılabilmektedir. Ancak giriş verisi farklı formatlarda ve kanal sayısında olabilmektedir. Renkli görüntüler, (Kırmızı-Yeşil-Mavi) 3 kanaldan meydana geldiği için, konvolüsyon işlemi 3 kanal için yapılmaktadır. Çıkış işaretinin kanal sayısı da uygulanan filtre kanalı sayısı ile eşit olarak hesaplanmaktadır. Kenar bulma işleminde; kenar bilgileri, görüntüden elde edilen öznitelikler içinde en çok

ihtiyaç duyulan bilgilerden birisi olma özelliğini taşır. Bu, giriş bilgisinin yüksek frekanslı bölgelerini temsil etmektedir. Bu öznitelikleri elde etmek için dikey ve yatay olmak üzere iki filtre ayrı olarak kullanılmaktadır. Padding İşlemi; konvolüsyon işleminden sonra giriş işareti ile çıkış işareti arasındaki boyut farkını yönetmek amacıyla kullanılan bir hesaplama adıdır. Kaydırma İşlemi(Stride); konvolüsyon için ağırlık matrisi olan filtreyi görüntü üzerinde birer piksellik adımlarla veya daha büyük adımlarla kaydırılacağına bilgisini vermektedir. Bu durumda doğrudan çıkış boyutunu etkileyen diğer bir parametreyi oluşturmaktadır. Şekil 3.2’de kenar bulma ve kaydırma işlemlerinin gösterimi mevcuttur. Ortaklama ya da Havuzlama (Pooling) İşlemi ise; genellikle maksimum ortaklama ya da havuzlama yöntemi kullanılmaktadır. Ağırlık bu katmanı öğrenilen parametreyi içermemektedir. Giriş matrisinin kanal sayısını sabit tutulur bu sayede yükseklik ve genişlik bilgisi azaltılır. Bu işlem, hesaplama karmaşıklığını azaltmak amacıyla kullanılan bir adım olma özelliğini taşır. Ancak, Hinton’ın kapsül teorisine göre bu işlem verideki önemli sayılabilecek bazı bilgilerin de kaybolmasına neden olduğu için başarımdan ödün vermekte olduğu düşünülmektedir. Şekil 3.2’deki örnekte “2x2 max-ortaklama işlemi” 2 piksel kaydırılarak uygulanmıştır. İlgili 4 elemanın olduğu alandaki en büyük değer çıkışa aktarılmaktadır. Çıkışta 1/4 boyutlu bir veri elde edilmektedir [20].



Şekil 3.2. Kenar bulma ve kaydırma işlemi gösterimi [20]

CNN'yi oluşturan katmanlardan aşağıda daha detaylı bahsedilmiştir. Bu katmanlar; Giriş Katmanı, Konvolüsyon Katmanı, Düzleştirilmiş Doğrusal Birim Katmanı, Havuzlama Katmanı, Tam Bağlantılı Katman, DropOut Katmanı ve Sınıflandırma Katmanı'dır.

-Giriş Katmanı: CNN'nin ilk katmanıdır. Ham olarak ağa gelen veride tasarım için veri boyutu oldukça önemlidir. Bu veri boyutunun yüksek seçilmesi, test süresini uzatabilirken, düşük seçilmesi eğitim süresini kısaltabilmektedir. Fakat bu durum performansı da düşürmektedir. Görüntü analizinin en iyi sonuçlarda yapılması için uygun veri boyutu seçilmesi gerekmektedir.

-Konvolüsyon Katmanı: En temel katman olup dönüşüm işleminin gerçekleştirildiği yerdir. Bu dönüşüm görüntünün belirli filtrelerden geçerek kaydırılmasıyla meydana gelmektedir. Filtreler görsel boyunca kaydırılmakta ve kaydırma sırasında görselin değerleri ile filtredeki değerler çarpılmaktadır, elde edilen değerler toplanarak net sonuç bulunmaktadır. Bu işlem tüm görsel yapısına uygulandığında yeni bir görsel elde edilmiş olunur.

-Düzleştirilmiş Doğrusal Birim Katmanı(ReLU): Doğrultucu birim olarak bilinen ReLu katmanı konvolüsyon katmanından sonra gelmektedir. Bu katman ile ağ hızlı öğrenir.

-Havuzlama Katmanı: ReLu katmanından sonra gelen katmandır. Bu katman aşağı örnekleme işlemi yaparak sonraki katman için giriş boyutunu azaltma işlemi yapmaktadır. Bu katmandaki işlemler, konvolüsyon katmanı sonucu oluşan filtre sayısı kadar gerçekleşmektedir. CNN'de havuzlama katmanı isteğe göredir ve bazı mimarilerde bu katman kullanılmaz. Gelen verideki gereksizlikleri yok etmek için kullanılır. 3 tip havuzlama vardır; max, min ve average bunlara örnektir. Ayrıntıları alırken bazı ayrıntıları kaybedip bilgi kaybına sebep olabilir.

-Tam Bağlantılı Katman: Havuzlama katmanından sonra gelen katmandır. Kendinden önceki bütün katmanlara bağlıdır. Farklı mimarilerde Tam Bağlantılı Katman adedi

değişebilmektedir. Tam bağlantılı katman, konvolüsyonda çıkan öznetelikler arası ilişkiyi sağlar.

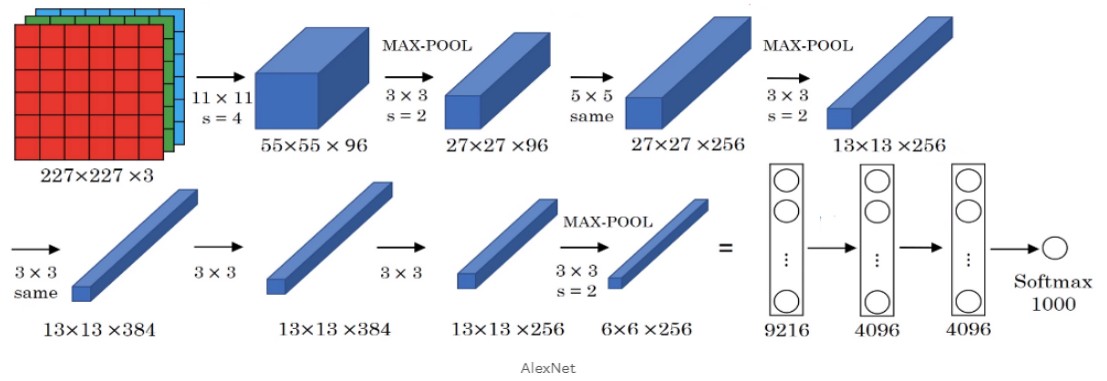
-DropOut Katmanı: Büyük verilerin eğitilmesi sırasında ezberleme yapmamak için kullanılan katmandır. Bu katmanda, ağdaki rastgele seçilen bazı sinirler etkisiz hale getirilmektedir. Sinirlere bağımsızlık kazandırması sayesinde daha iyi verim alınmasını sağlar.

-Sınıflandırma Katmanı: Tam bağlantılı katmandan sonra gelen sınıflandırma işlemlerinin yapıldığı katmandır. Bu katmanda farklı sınıflandırıcıların kullanılması ile birlikte genellikle başarılı sonuçlar veren ‘softmax’ sınıflandırıcısı kullanılır. Sigmuid 2 sınıflı, softmax çok sınıflı derin öğrenme çalışmaları için kullanılan sınıflandırıcılardır. Bu katmandaki çıkış değeri, sınıflandırılacak nesne sayısına eşit şekildedir. Sınıflandırmada, farklı nesneler ‘0-1’ arasında bir değerde çıkış üretmektedir. Ağın tahmin ettiği nesnenin, 1’e yakın sonucu üreten çıkış olduğu anlaşılmaktadır [19].

3.3. Derin Öğrenme Modelleri

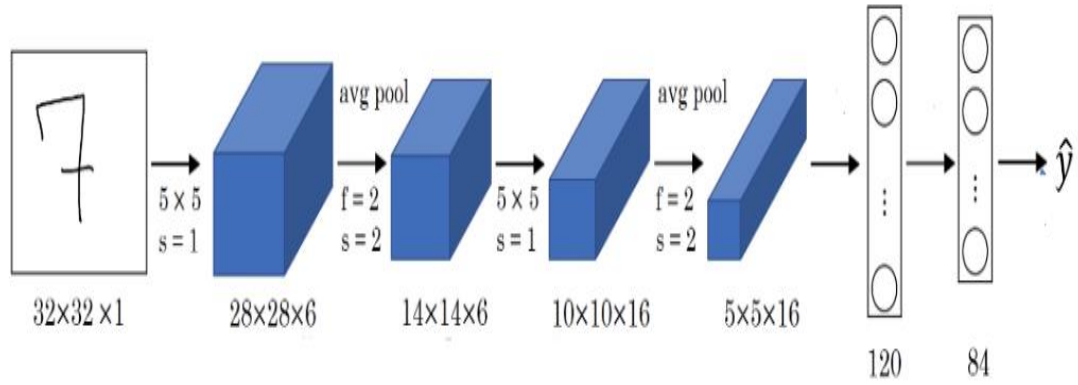
İmageNet model yarışmasını kazanan CNN modelleri; Alex Net, Le Net, ZF Net, VGG-16, GoogLeNet, Microsoft RestNet gibi modeller başta olmak üzere sıralanabilir. Bu modeller derin öğrenmeyi anlamada ve geliştirmede önemli bir yer edinmektedir. Bu sebeple modeller oldukça önemlidir.

Alex Net; dünya çapında derin öğrenmenin duyulmasını sağlayan 2012 yılı ImageNet yarışmasını kazanmış ve Krizhevsky, Sutskever ve Hinton tarafından geliştirilmiş bir modeldir. Birbirini takip eden konvolüsyon katmanları ve ortaklama katmanları mevcuttur. Bilgisayarlı nesne tanımlama doğruluk oranını %10,8 iyileştirerek %83,6 yapmıştır. Şekil 3.1.’de Alex Net mimarisine ait detaylı yapı görülmektedir [20].



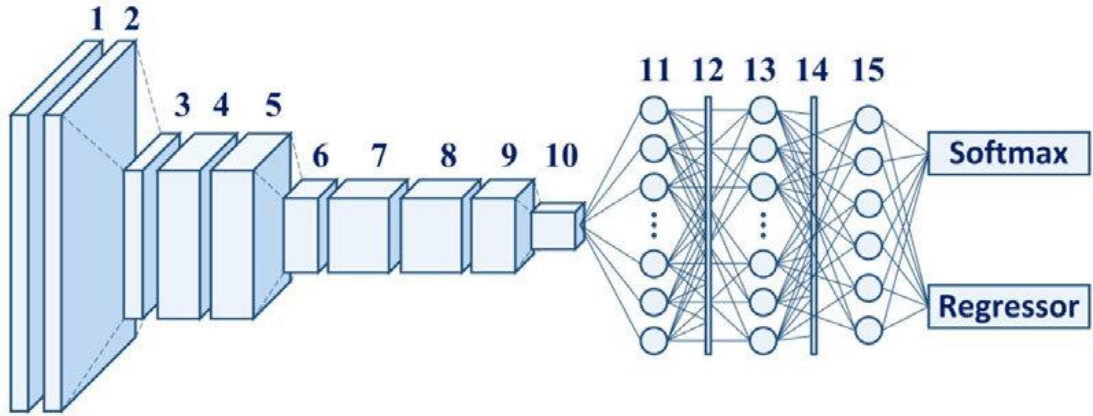
Şekil 3.3. Alex Net mimarisi [20]

Le Net; 1998 yılında yayınlanmış ve ilk başarılı sonucu veren CNN modelidir. Yan LeCun ve ekibi tarafından posta numaraları, banka çekleri üzerindeki sayıların okunması amacıyla hazırlanmıştır. Şekil 3.2.'de Le Net mimari yapısı verilmiştir [20].



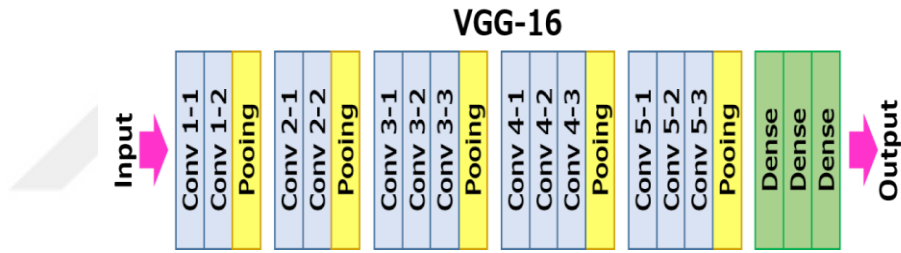
Şekil 3.4. Le Net mimarisi [20]

ZF Net; Bu mimari yapısı, 2013 yılı ImageNet yarışmasını kazanan, Alex Net mimarisinin geliştirilmiş hali olan modeldir. Nesne tanıma hata oranını %4,2 daha iyileştirmiş ve %11,2 ye düşürmüştür. Şekil 3.3.'de ZF Net mimarisine ait yapı ifade edilmiştir [21]. Çalıştırılması için işlemci ve ekran kartı yüksek bilgisayar donanımı gerekmektedir.



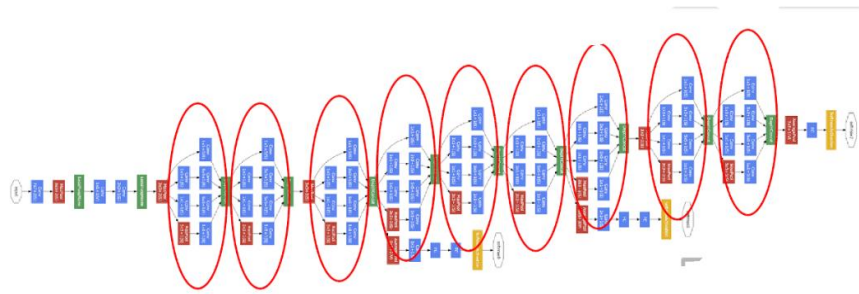
Şekil 3.5. ZF Net mimarisi [21]

VGG-16; basit bir ağ modeli olan VGG-16'yı daha önceki modellerden ayıran en önemli fark, ikili ya da üçlü konvolüsyon katmanlarını ortaklama katmanları izlemesidir. Şekil 3.4.'de VGG-16 mimarisine ait katmanların detayı yapı verilmiştir [22].



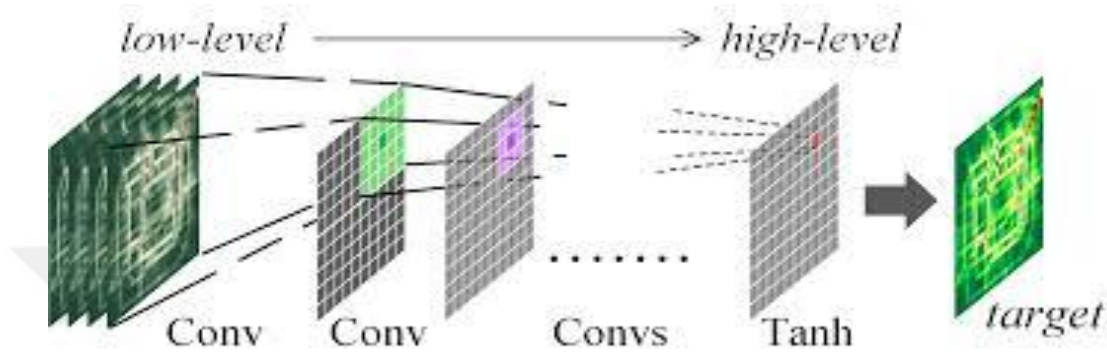
Şekil 3.6. VGG-16 mimarisi [22]

GoogLeNet; ImageNet 2014 yılı yarışmasının kazananı olan ve Inception modüllerinin birleşiminden oluşan model karmaşık bir yapıya sahiptir. %5,7 hata oranına sahip olan model, veri kümelerinde başarılı sonuçlar vermiştir. Şekil 3.5.'de GoogLeNet mimarisi ifade edilmektedir [20].



Şekil 3.7. GoogLeNet [20]

Microsoft ResNet; Tüm mimari yapılardan daha derin olarak tasarlanan model, 2015 yılı ImageNet yarışması kazananı olmuştur. 34 katmanlı ilk ağ olan yapı residual bloklardan oluşmaktadır [23]. Şekil 3.6.'da ResNet mimarisine ait detaylı yapı görülmektedir [24].



Şekil 3.6. ResNet [24]

3.4. Derin Öğrenme Kütüphaneleri

Derin öğrenme yapısında çok sayıda kütüphane bulunduran bir yapıya sahiptir. Python'da bir düzineden fazla derin öğrenme kütüphanesi vardır. Python dili ile kullanabileceğiniz derin öğrenme kütüphanelerinin bazıları hakkında bilgi amaçlı bir tablo oluşturulmuştur. Bu tezin hazırlanmasında temel olarak kullanılan kütüphaneler olan 'Caffe' ve Alex Net modelinde kullanılan Tensorflow ve Keras kütüphaneleri hakkında bilgiler detaylandırılmıştır. Bazı derin öğrenme kütüphaneleri; Theano, TensorFlow, Caffe, Keras, Mxnet, Pylearn2, Lasagne, Torch, Covnetjs, Sci-Kit Learn, Deeplearning4j, Apache Spark bu kütüphanelerden bazılarıdır. Tablo 3.4.te bazı derin öğrenme kütüphanelerinin bilgileri verilmiştir [6]. Bahsedildiği gibi, seçilebilecek çok fazla derin öğrenme kütüphanesi bulunmaktadır. Python Kütüphaneleri ve çerçeve öğretici ile Derin öğrenmede, öğrenmeye ve uygulamaya yapmaya başlamak için çeşitli derin öğrenme kütüphaneleri tanıtılmıştır ve en yararlı kullanım alanları hakkında bilgiler verilmiştir. Her derin öğrenme Python kütüphanesi ve çerçevesinin kendi yararları ve sınırlamaları bulunmaktadır. Tabloda listelenen kütüphaneler en yaygın kullanılan, en popüler, öğrenmeye değer ve uygulama açısından yarar sağlayan kütüphanelerdir.

Tablo 3.1. Derin Öğrenme Kütüphaneleri

Kütüphane Adı	Dili	Geliştiricisi	İşlevi
Theano	Python	MILA Lab	Çok boyutlu diziler dâhil matematik ifadelerini etkili bir şekilde tanımlamayı, en iyilemeyi ve değerlendirmeyi sağlayan bir Python kütüphanesidir.
TensorFlow	Python	Google	Veri akışı grafikleriyle verimli sayısal hesaplamalara olanak tanımaktadır.
Caffe	Python	BVLC	Caffe derin öğrenme yapısı hızlı ve modüler olacak şekilde tasarlanmıştır. Model ve optimizasyonlar kodlama yapılmaksızın ayar dosyası üzerinden yapılabilmektedir.
Keras	Python	Google	Keras kütüphanesi Tensorflow ve Theano'ya üst katman olarak yazılmış, daha kolay model geliştirmeyi sağlayan bir Python kütüphanesidir.
Mxnet	Python	Amazon	Keras gibi yüksek seviye kütüphanedir, bir polyglot (çok dilli) olmasından dolayı, farklı dillerde model paylaşan ekipler için harika çözümler sunmaktadır. Diğer bir avantajı da dağıtılmış bilgi işlemini desteklemesidir.

3.4.1. Caffe, tensorFlow ve keras kütüphaneleri

Caffe: Derin öğrenme; doğal dil işleme, nesne tanıma, yüz tanıma, konuşma tanıma ve bilgisayarlı görme gibi işlemlerin yapıldığı bir makine öğrenmesi kütüphanesidir. Caffe, (Convolution Architecture For Feature Extraction) “Özellik Ekstraksiyonu için Evrişim Mimarisi” açılımına sahip açık kaynak kodlu ve geniş bir camiası olan, en popüler derin öğrenme paketlerinden birisi olarak bilinmektedir. Caffe, derin öğrenme yapısı hızlı olmakla birlikte modüler olacak şekilde Berkeley Vision and Learning Center – BVLC (Berkeley Görüntü ve Öğrenme Merkezi) ve kullanıcı topluluğu tarafından tasarlanarak geliştirilmiştir. Yangqing Jia, Caffe kütüphanesini Berkeley’de doktora döneminde hazırlanmıştır. Caffe BSD 2-Clause license altında kullanıma sunmuştur [25].

Kullanışlı ifade yapısı sayesinde farklı fikirlerin ya da uygulamaların ortaya çıkmasına ilham kaynağı olmaktadır. Kodlama olmaksızın, model ve optimizasyonlar

ayar dosyası üzerinden yapılabilmektedir. GPU üzerinde eğitim işlemini yapmak amacıyla CPU ve GPU değişimi bir etiket ayarı ile gerçekleştirilebilmektedir, bu sayede küme bilgisayarlara ya da mobil cihazlara yayılım sağlanabilmektedir. Genişletilebilir kod yapısı bulunan Caffe, aktif geliştirmeyi destekleyebilmektedir. Caffe'nin ilk kullanım yılında, yaklaşık bin geliştirici tarafından birçok önemli değişiklik için katkıda bulunulmuştur. Caffe araştırma deneyimleri ve endüstri uygulamaları açısından hızlısal olarak mükemmel denilebilecek hale getirilmiştir. Caffe'nin, NVIDIA K40 GPU* işlemciyle “60 milyonun üzerinde resmi bir günde işleyebilme kabiliyetine sahip olduğu” söylenmektedir. Burada 1 ms/resim, öğrenme için 4 ms/resime tekabül eden bu ifade ile Caffe'nin erişilebilir en hızlı convnet uygulamalarından biri olduğu düşünülmektedir. Caffe günümüzde popüler olarak akademik araştırma projelerine, geniş ölçekli görüntü alanındaki endüstriyel uygulamalar ile konuşma ve çoklu medyaya güç kazandırmaktadır.

Caffe dosyaları; Ağ Model Dosyası, Solver Dosyası ve Ağırlık Dosyası olmak üzere 3 kısımdan oluşur. “Ağ Model Dosyası”, ağ yapısını gösteren prototxt formatında dosyalardır. Eğitim ve kullanım için bu dosyalar ayrı ayrı olmaktadır. “Solver Dosyası”, Eğitim parametrelerinin ayarlandığı prototxt formatında dosyalardır. “Ağırlık Dosyası”, Eğitim sırasında ve sonucunda oluşturulan ağırlıklarının tutulduğu ‘.caffemodel’ uzantılı dosyadır [26].

Caffe Model günlük yaşamımızda birçok alanda karşımıza çıkmaktadır. Bir sosyal ağ olan Facebook, ölçekte vizyon için üretimde, yüklenen fotoğraflar Caffe'den geçmektedir, körler için otomatik alt metin, sakıncalı içerik algılama, bugünün anılarını ortaya çıkarmak gibi işlevlerde Caffe'den faydalanmaktadır. Görsel ağırlıklı bir sosyal paylaşım sitesi olan Pinterest’ te yüklenen fotoğrafları Caffe'den geçirmektedir ve görsel arama için derin öğrenmeyi kullanmaktadır. Adobe, Görme ve grafikte araştırma için eğitim ağılarıyla Photoshop dâhil olmak üzere ürünlerde özel çıkarım için Caffe'den yararlanmaktadır. Arama motoru ve e-posta hizmeti olarak çalışan Yahoo ise; tavsiye için kollayıcı haberleri ve restoran fotoğraflarında ve kullanıcı fotoğraf albümleri düzenlemede Caffe Modeli kullanmaktadır.

Bir Caffe Model uygulaması; Problem seçimi, veri seti hazırlama, model seçimi ve düzenleme, eğitim ve test aşamalarından oluşmaktadır. Tezin uygulama kısmı bu aşamalara göre detaylandırılarak incelenmiştir.

TensorFlow: Tensorflow, Google’ın geliştirdiği açık kaynak kodlu bir derin öğrenme kütüphanesidir. Yaygın bir şekilde derin öğrenme uygulamalarında kullanılsa da, Tensorflow daha geniş bir sahayı kapsamaktadır. Tensorflow, veri flow grafları kullanarak numerik hesaplama için geliştirilmiş açık kaynak bir kütüphanedir. Graflardaki düğümler matematiksel işlemleri temsil eder, graf kenarları birbirleri arasında iletilen çok boyutlu arrayleri(tensor) temsil etmektedir. Esnek yapısı sayesinde, tek bir API ile platform farketmeksizin hesaplamaları, bir ya da birden çok CPU, GPU kullanarak deploy etmenize olanak tanır. TensorFlow Hub, makine öğrenim modellerinin yeniden kullanılabilir bölümlerinin yayınlanmasını, keşfedilmesini ve tüketilmesini teşvik eden bir kütüphanedir. İos ve Android desteği ile mobil programlarda kullanılabilir. Özellik algılamayı yeniden kullanarak, daha az eğitim verisiyle ve zamanla, kişisel sınıfları tanıyan modeller oluşturabilmektedir. Aynı görev için modüller ortak bir imza uygulamalıdır, böylece modül tüketicileri bunları kolayca değiştirebilir ve problemleri için en iyi olanı bulabilir. Modern görüntü tanıma modellerinde milyonlarca parametre bulunur. Bunları sıfırdan eğitmek, çok sayıda etiketli eğitim verisi ve çok fazla hesaplama gücü gerektirir (yüzlerce GPU-saat veya daha fazla zaman). Aktarım öğrenimi, ilgili bir görevde eğitilmiş ve yeni bir modelde yeniden kullanabilen bir model parçasını alarak bunun çoğunu kısaltan bir tekniktir [27, 28].

Keras: Keras, Tensorflow veya Theano kütüphaneleri üzerinde çalışmakta olan ve Python dili ile yazılmış yüksek seviyeli bir kütüphanedir. Kullanıcıyı daha düşük seviyedeki diğer kütüphanelerin karmaşıklığından kurtararak, daha basit şekilde yapay sinir ağı modellerinin oluşturulmasına ve eğitilmesine olanak tanımaktadır. Keras temel yapı model olarak, en basit model tipi ise katmanlar yığınınından oluşan “Sequential” modeldir. Yaygın olarak kullanılan yapı, bu yapıdır. Paylaşımlı katmanların, daha karmaşık mimarilerin tasarlanabildiği bir diğer model tipi ise “Functional API” modelidir. Keras’ın içerdiği temel katman çeşitlerinden bazıları;

Dense katmanı, activation katmanı, dropout katmanı, flatten katmanı ve reshape katmanıdır. Dense katmanında, girdideki her bir düğüm çıkıştaki her bir düğüm ile bağlıdır. Activation katmanı, ReLu, tanh, sigmoid gibi aktivasyon fonksiyonları içerir ve çıktı değerini aktivasyon fonksiyonuna göre belirler. Dropout katmanı, eğitim aşamasında *overfitting* i azaltma amacıyla kullanılır. Flatten katman, matris formundaki veriyi düzleştirmek için kullanılır. Reshape katman ise girdi boyutlarını dönüştürmek için kullanılır. Temel katmanların dışındaki bazı katman çeşitleri ise Konvolüsyonel katman, Havuzlama katmanı, Recurrent katman vb. gibi katmanlardır [29, 30].

3.5. Derin Öğrenme Uygulamalarının Hiperparametreleri

Modeli tasarlayan kişiye bırakılan, ne ve nasıl olması gerektiği yine tasarımcı tarafından belirlenen, probleme, veri setine göre değişiklik gösteren parametrelere hiperparametre denilmektedir. Modelin yüksek başarımlı sağlanması için kullanılan birbirinden farklı hiper parametre grupları olabilmektedir. En uygun hiper parametre grubunun seçimi ise önemli problemlerden birini oluşturmaktadır. Bu parametrelerden bazıları aşağıda belirtilmiş olup, problemin çözümü için uygun hiperparametre seçilmesi için gerekli bilgilendirmeler yapılmıştır.

-Veri Seti Boyutu; Veri seti büyüklüğü derin öğrenmenin sağlanması için önemlidir. Veri sayısı ne kadar büyük olursa, öğrenme o kadar iyi olur. Bu sebeple öğrenme için en önemli faktördür.

-Mini Batch Boyutu; “Mini-batch” birden fazla girdinin parçalar halinde işlenmesi olarak adlandırılmaktadır. Model tasarlanırken ayarlanan mini batch hiperparametresi, modelin aynı anda kaç veriyi işleyeceğini gösterir.

-Öğrenme Hızı(Learning Rate) ve Momentum Katsayısı; Öğrenme hızı değeri genelde varsayılan değer olarak 0.01 kullanılmakta belli bir iterasyondan sonra 0.001'e düşürülmektedir. Genel olarak kullanılan momentum beta katsayısı değeri 0.9'dur. Uygun parametre aralığı ise 0.8'den 0.99'a kadardır.

-Optimizasyon Algoritması Seçimi; Derin öğrenme uygulamalarında, öğrenme işlemi temel olarak bir optimizasyon problemidir. Doğrusal olmayan problemlerin çözümünde optimum değeri bulmak amacıyla optimizasyon yöntemleri kullanılmaktadır. Bu yöntemler yaygın olarak; stochastic gradient descent, adagrad, adadelta, adam, adamax gibi optimizasyon algoritmalarıdır. Derin öğrenmede yüksek başarımlar için doğru optimizasyon algoritmasının seçilmesi önemlidir.

-Eğitim Tur (Epoch) Sayısı; Tekrarlayan işlem ya da iterasyon sayısına “epoch” denir. İterasyon sayısı arttıkça modelin başarımlarında gözle görülür oranda artma görülebilir.

-Ağırlık (Weight) Başlangıç Değerinin Belirlenmesi; W ağırlık değerleri, model çalışmaya başlatılırken hepsi 0 olacak şekilde veya 0.5 gibi standart sapmaya sahip olacak dağılım şeklinde veya 0.5 ile 0.9 arasında uniform dağılıma sahip olacak şekilde veya daha önceki bir modelin ağırlık değerleri kullanılacak şekilde tanımlanabilir.

-Aktivasyon Fonksiyonu; Çok katmanlı yapay sinir ağlarında, doğrusal olmayan dönüşüm işlemleri için kullanılmaktadır. Modele non-linearite katar. Sigmoid, tanh, ReLU, PreLU fonksiyonlardan bazılarıdır. En kullanışlı olan “ReLU” denilebilir.

-Seyrelme (Dropout) Değeri ve Seyreltme Uygulanacak Katmanlar; Seyreltme değeri genelde 0.5 olarak kullanılmaktadır. Farklı kullanıldığı durumlar bulunmaktadır.

-Katman Sayısı ve Gizli Katmanlardaki Nöron Sayısı; Katman sayısı arttıkça modelin daha iyi öğrenildiği gözlemlenmiştir. Nöron sayısı hafızada tutulan bilgi sayısını göstermekte olup nöron sayısının fazla olması bellek ihtiyacı ve hesaplama zamanını artırmaktadır. GPU olmayan bir ortamda çalışılırsa bu kriterleri dikkate almak gerekmektedir.

-Konvolüsyonel Sinir Ağı (Kernel) Boyutu; Konvolüsyonel sinir ağlarında her bir katmanda matris üzerinde işlem yapan kerneller kullanılmaktadır. Kernel boyutu, ne

kadar genişlikte verinin birbirini etkileyeceğine karar vermektedir. Genelde 3x3, 5x5, 7x7 kerneller kullanılmaktadır.

-Konvolüsyonel Sinir Ağı (CNN) “Havuzlama(Pooling)” Yöntemi: Max, Min, Mean; Konvolüsyonel sinir ağlarında kullanılan kernellerin çıktısı üzerinde “havuzlama” işlemi yapılmaktadır. “Havuzlama” işlemi kernelde bulunan veriler için filtreleme tekniğidir. Sıkça kullanılan havuzlama yöntemleri; kernel içindeki matris’de bulunanların en büyüğünü alma “max pooling”, en küçüğünü alma “min pooling” ve ortalamasını alma “mean pooling”dir.

Derin öğrenme uygulamalarında en sık karşılaşılan hiper parametreler ele alınmıştır. Açıklanan bu hiper parametreler dışında, ek olarak farklı ağ yapılarına özgü başka hiperparametreler de bulunmaktadır [31].

BÖLÜM 4. YAPILAN UYGULAMALI ÇALIŞMALAR

4.1. Derin Öğrenme İle Çalıştırılan Yaş-Cinsiyet Tespiti Modelleri

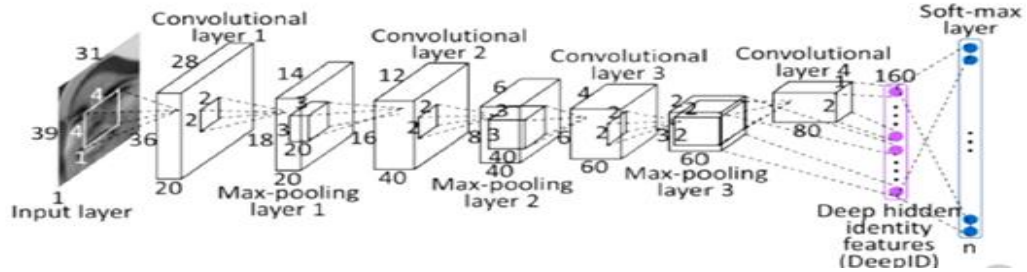
İncelenen Model 1: ‘Derin Öğrenme Algoritmasında Otomatik Yaş ve Cinsiyet tahmini’ yapan çalışmanın modeli ve algoritması çalıştırılarak, yapılan çalışmanın mantığının detaylı bir şekilde anlaşılması amaçlanmıştır [9].

Modelin Amacı: Matlab platformunda derin öğrenme algoritmalarını kullanarak toplu ve tek fotoğraflarda bulunan kişilerin yaş ve cinsiyetlerini tahmin etmek amaçlanmıştır.

Modelin Metodolojisi: Kullanılan veri seti “dataset-images of groups of people” içinde yer alan düğün, aile ve grup fotoğraflarının olduğu 5080 görüntülü 28231 yüz içeren veri setidir. Yüz tanımlayıp cinsiyet tahmin etme ve yaş sınıflandırma olmak üzere 2 farklı görevi eğitmek için 2 dataset kullanılmıştır. Yüz tanımlama görevi için 8025 insan yüzünün her bir 20 tanesi içinden rastgele seçilen 15’i eğitim için, 5’i doğrulama için hazırlanmıştır. Yaş sınıflandırma görevi için insan yaşları 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, 66+ olmak üzere 7 gruba ayrılmıştır. Rasgele 3500 yüz görüntüsü seçilmiş ve her yaş grubundan 500 görüntü seçilecek şekilde ayarlanmıştır. Doğrulama için ise, yaş grubu başına 50 yüz resmi ile 1050 resimli veri kümesi oluşturulmuştur. Çalışmada Deep ConvNets tabanlı yeni bir model üretilmiştir. Deep ConvNets’de multiclass sınıflandırıcı olarak softmax ile eğitim yapılmıştır.

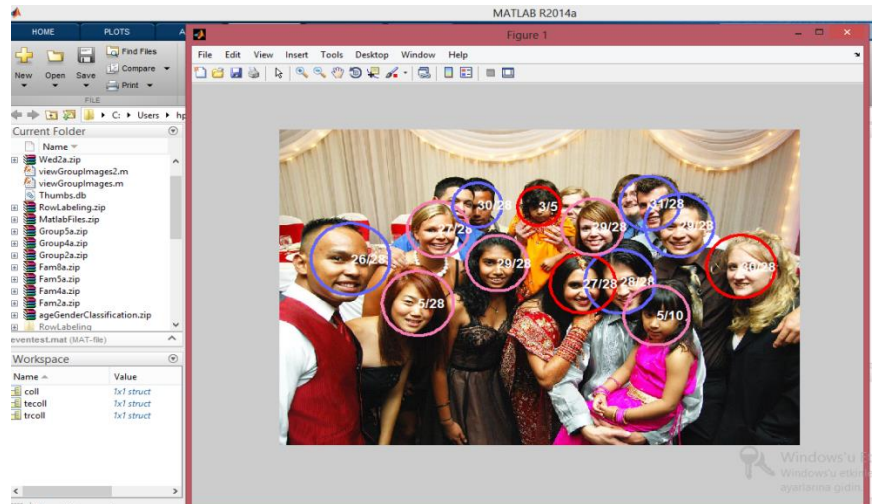
Modelin Mimarisi: Algoritma mimarisi 3 kısımdan oluşmaktadır. İlk olarak, etiketsiz verileri eğitmek için kullanılan ‘transfer learning modeli’, daha sonra çoklu model kısmı ve en son Deep ConvNets’in yapısı olarak algoritma tamamlanmıştır. Modele gelen görüntü 39x31 pixellik ağ ile girer ve bu yapı DeepID’nin yapısına

benzer. Gelen bir görüntü 60 parçaya bölünür ve her bir parça bağımsız bir Deep ConvNets eğitiminde kullanılmaktadır. Her ağda 4 CNN, 3 max-pooling, 1 full connected ve 1 output katmanı bulunmaktadır. Modelin katmanları aşağıdaki şekilde daha ayrıntılı olarak belirtilmiştir(Şekil 4.1.).



Şekil 4.1. Çalışmada kullanılan DeepID yapısı

Modeli Çalıştırma: Çalışmanın dataset adresine gidilerek (The Images of Groups Dataset), düğün, grup, aile resimlerinin bulunduğu veri dosyaları indirilmiştir. Gerekli matlab dosyaları ve yaş-cinsiyet sınıflandırma algoritmaları çalıştırılıp Şekil 4.2. gibi görüntüler elde edilmiştir. Çalıştırılan makinede GPU üzerinde işlem gerekmediği için matlab yüklü olan ‘Hp-G62’ GPU olmayan aygıtı üzerinde çalıştırılmıştır. Model sonucu elde edilen fotoğraflarda pembe halkalar cinsiyet olarak kadını, mavi halkalar da cinsiyet olarak erkeği göstermektedir. Kırmızı halkalar hatalı cinsiyet tahmini yapıldığını belirtmektedir. Yaş tahmininde ise 26/28 örneğinde olduğu gibi 1. Yazılan ‘26’ tahmin edilen yaşı, 2. yazılan ‘28’ gerçek yaşı gösterecek şekilde ayarlanmıştır.



Şekil 4.2. Çalıştırılan model sonrası elde edilen görüntüler

Model Sonucu ve Etkileri: Modelin değerlendirilmesinde kullanılan hiperparametrelere göre; veri seti büyüklüğü ve çeşidi 5080 görüntü ve 28231 yüz içermektedir. Modelin aynı anda kaç veriyi işleyeceğini belirleyen ‘mini batch boyutu’ 128 olarak ayarlanmıştır. Tablo 4.1. ile batch boyutuna göre en iyi model performansını gösteren tablo bilgisi mevcuttur.

Tablo 4.1. Batch boyutuna göre en iyi model performansı

Batch Boyutu	Kesin Eşlemedeki doğruluk(%)	Her bir yaş grubunun hata oranı(%)
16	47	86
32	54	91
64	52	90
128	49	88

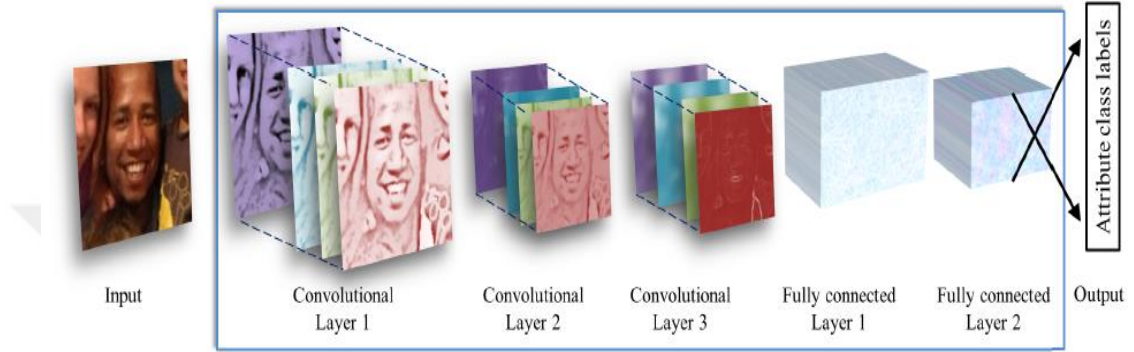
Yaş doğruluk(Gender accuracy); kadınlar için 0,7165 ve erkekler için 0,6571 olarak bulunmuştur. Cinsiyet için doğruluk düşük olmasına rağmen 0-2 yaş ve 66+ yaş aralıklarını diğer aralıklardan daha kolay tahmin ettiği görülmüştür. Öğrenme işlemi 300 iterasyonda 0,8 hata oranından 0,4 hata oranına düşürülmüştür.

İncelenen Model 2: ‘Konvolüsyonel sinir ağlarını kullanarak yaş ve cinsiyet sınıflandırması’ yapan bir diğer derin öğrenme çalışmasında ise caffe kütüphanesi ve caffe model ile gerekli algoritmalar çalıştırılmış, sonuçlar ve model analiz edilerek değerlendirilmiştir [10].

Modelin Amacı: Derin öğrenme algoritmalarını kullanarak, yüz görüntüsünden özellikler çıkarıp yaşı tahmin etmektir.

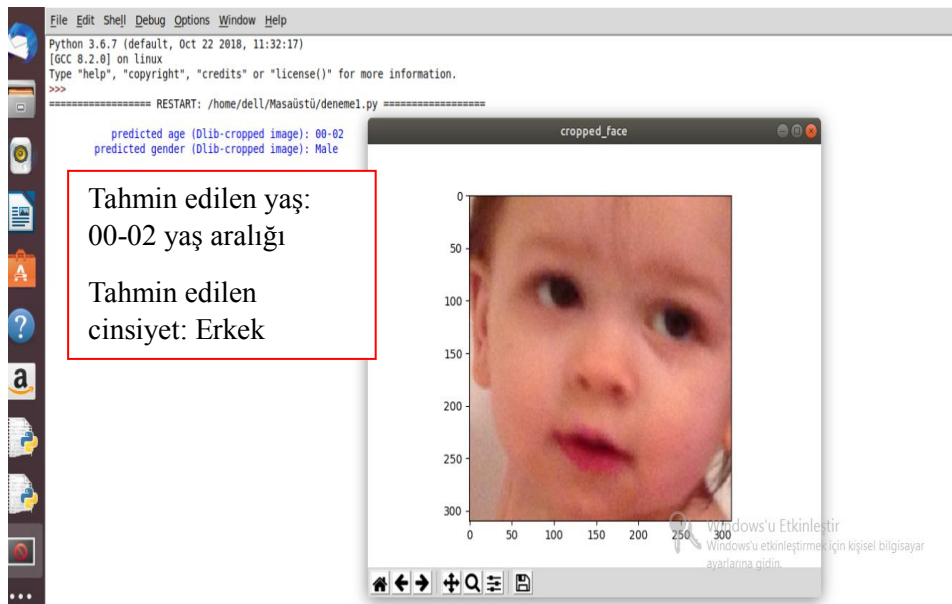
Modelin Metodolojisi: Kullanılan veri seti ‘Aidience Benchmark Dataset’ olup, 8192 erkek ve 9411 kadın olmak üzere toplam 19487 görüntü içermektedir. Cinsiyet sınıflandırma dışında yaş sınıflandırma için ise; 0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60+ olmak üzere 8 sınıfa ayrılmıştır. Caffe açık kaynaklı çerçeve kullanılmıştır.

Modelin Mimarisi: Model, 3 konvolüsyonel katman ve 2 tam bağlantılı katmandan oluşmaktadır. İlk konvolüsyonel katmanı 7x7 pixel 96 filtre içermektedir. İkinci konvolüsyonel katmanı 5x5 piksel 256 filtre, üçüncü ve son konvolüsyonel katmanı ise 3x3 piksel 384 filtre içerir. Her biri 512 nöron içeren 2 adet tam bağlı katman bulunmaktadır. Şekil 4.3. te modelin CNN mimarisinin katmanları gösterilmektedir.



Şekil 4.3. Modelin CNN mimarisine göre katman yapısı

Modelin Çalıştırılması: Caffe kütüphanesi içindeki ‘.prototxt’ ve ‘.caffé’ uzantılı dosyalar bilgisayara indirilmiştir. Caffe kütüphanesi ile çalıştırılan modelden elde edilen sonuç şekil 4.4. te görüldüğü gibidir. Gerçek yaşı 1 olan ve cinsiyeti erkek olan örnek resimdeki bebek, program tarafından ‘0-2 yaş aralığında erkek’ olarak tahmin edilmiştir.



Şekil 4.4. Modelin çalıştırılması ile elde edilen örnek resim

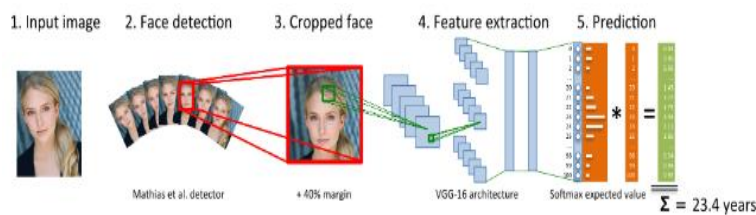
Modelin Sonucu ve Etkileri: Model Adience Benchmark ile cinsiyet için yaklaşık %87 doğru tahminleme, yaş için ise yaklaşık %85 tahminleme yapmıştır. Modelin çalışmasından iki önemli sonuç çıkarılabilir. İlk olarak, CNN, yaş ve cinsiyet için etiketlenmiş daha küçük boyutta çağdaş sınırsız görüntü kümeleri göz önüne alınsa bile, yaş ve cinsiyet sınıflandırma sonuçlarını iyileştirmek için kullanılabilir. İkincisi, modelin sadeliği, daha fazla eğitim verisi kullanan daha ayrıntılı sistemlerin, burada bildirilenlerin ötesinde sonuçları büyük ölçüde iyileştirebileceğini göstermektedir.

İncelenen Model 3: ‘DEX: Tek bir görüntüden gözle görülür yaştan derin beklentisi’ üzerine çalışan makalede VGG-16 derin öğrenme modelinden faydalanarak yarım milyona yakın IMDB ve Vikipedi resimleri üzerine çalışılmıştır [3].

Modelin Amacı: Derin öğrenme algoritmalarından VGG-16 modelini kullanarak, yüz görüntüsünden özellikler çıkarıp yaşı tahmin etmektir.

Modelin Metodolojisi: Kullanılan veri seti IMDB-WIKI datasettir. Bu veri seti; yarım milyona yakın veri içermektedir. Eğitim için bu veri setinin yaklaşık %50’si kullanılmıştır. Bu, bugüne kadarki yaş tahmini için en büyük veri kümesidir. Yaş regresyon problemini derin bir sınıflandırma problemi olarak takip etmektedir, ardından softmax ile beklenen bir değer iyileştirmesi yapılmaktadır. CNN’lerin doğrudan regresyon eğitimi konusundaki iyileştirmeleri gösterilmektedir. Önerilen yöntem ile, görünür yaştaki Derin Beklenti (DEX), önce test görüntüsündeki yüzü algılar ve daha sonra kırılmış yüzdeki 20 ağ grubundan CNN tahminlerini çıkarır.

Modelin Mimarisi: Model tipik bir 41 katmanlı VGG-16 modelidir. Bu model, 13 konvolüsyon katmanı, 5 havuzlama katmanı ve 14 ReLu katmanı içermektedir. Modelin mimari yapısı Şekil 4.5. ile gösterilmiştir [3, 32].



Şekil 4.5. Modelin mimari yapısı[32]

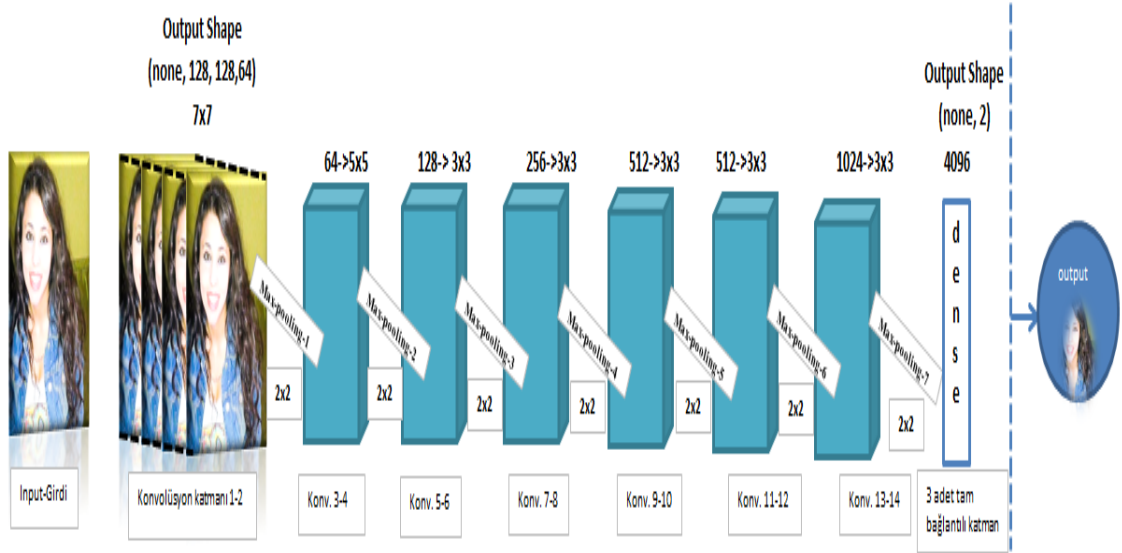
Modelin Çalıştırılması: PyCharm üzerinden indirilen veri seti ile VGG-16 modeli çalıştırılmak istense de donanımsal yetersizlikle karşılaşılmıştır.

Modelin Sonucu ve Etkileri: Görünen görüntülerde, görünen yaş tahmini konusu ele alınmıştır. Önerilen Derin Beklenti (DEX) yöntemi, ImageNet'te önceden tanımlanmış VGG-16 mimarisine sahip konvolüsyonel sinir ağlarını (CNN'ler) kullanmaktadır. Buna ek olarak, bugüne kadar bilinen en büyük veri setini oluşturmak ve CNN'leri sorgulamak için mevcut yaşta Internetteki ünlülere ait yüz görüntüleri taranmıştır. Ayrıca, CNN'ler sayesinde, yaş etiketli görünen yüz görüntüleri üzerinde ince ayarlanmalar yapılmıştır. Yaş regresyon problemi derin bir sınıflandırma problemi olarak belirlenmiştir, ardından softmax ile beklenen değerde bir inceltme yapılmıştır ve CNN'lerin doğrudan regresyon eğitimi konusundaki iyileştirmeleri gösterilmiştir. DEX kırılmış yüz görüntüsünde 20 ağın tahminini yapmaktadır. DEX açıkça yüz işaretlerini kullanmamaktadır. Önerilen yöntem, ChaLearn LAP 2015'in görünürde yaş tahmini konusunda meydan okumasını kazanmıştır [33]. İnsan referansından önemli ölçüde daha iyi bir performans sergilemiştir.

4.2. Yeni Bir Yaş Ve Cinsiyet Tahmini Uygulaması

Fotoğraflardan derin öğrenme ile yaş ve cinsiyet tahmini yapabilmek için veri seti olarak Vikipedi fotoğraflarından derlenen 3170 görüntülük dosya oluşturulmuştur. Veri seti içindeki resimler cinsiyetlerine ve yaşlarına göre etiketlendirilmiştir. Kadın ve erkek olmak üzere 2 sınıf oluşturulmuştur. Yaş için 15-25, 25-35, 35-50, 50-65, 65+ yaşlara sahip kişilerin görüntülerinin bulunduğu 5 sınıf oluşturulmuştur. Cinsiyet tahminleme için eğitim veri seti için 3170 görüntü, test veri seti için ise toplam görüntü içinden rasgele seçilen 318 görüntü kullanılmıştır. Yaş tahminleme için test veri setinde her bir yaş grubu için yaklaşık 55 resim bulunmaktadır. Uygulama, Ubuntu işletim sistemi üzerinde Python diliyle yazılmıştır ve sınıflandırma için derin öğrenme modellerinden Alex Net ve VGG-16 gibi derin öğrenme algoritmaları kullanılmıştır. Yeni geliştirilen bir derin öğrenme modeli ile Alex Net ve VGG-16 parametrelerinin kıyaslaması yapılmıştır. Yeni geliştirilen model 14 konvolüsyon katmanı ve 7 havuzlama katmanından ve 3 tam bağlantılı katmandan oluşmaktadır.

Toplam parametre sayısı 46.197.314'tür. Şekil 4.6.'da, modelin katman yapısı detaylı olarak ifade edilmiştir.



Şekil 4.6. Önerilen model mimari yapısı

Yeni modelin, karşılaştırma yapılan algoritmalarından farkını gösteren bilgiler tablo 4.2'de detaylı olarak gösterilmiştir. Tablodan yapılan incelemeler sonucu görüldüğü gibi yeni model, Alex Net ve VGG-16 modellerinden farklı bir mimariye sahip olmakla birlikte, başarımlar değerlendirilmesinde de Alex Net ve VGG-16 modellerinin arasında bir sonuç vermektedir. İşlem süresi bir kıyaslama kriteri olarak alındığında en uzun süreli çalışan modelin yeni geliştirilen model olduğu gözlemlenmiştir. Yüz tanımda kenarlık çıkarma işlevi için havuzlama katmanları kullanılmaktadır. Havuzlama katmanının, ayrıntıları alırken bilgi kaybına sebep olması ihtimali vardır. Yeni modelin VGG-16'dan farklı olarak, Alex Net gibi Dropout katmanına sahip olmadığı da görülmektedir. Dropout katmanı, ağı içindeki bazı bağlantıların kaldırılmasıyla eğitim performansını artırmaktadır. Öğrenememe ve ezberleme problemlerine çözüm için kullanılmaktadır. Relu aktivasyon fonksiyonu da aynı problemlere çözüm için modellerde bulunmaktadır [34]. Uygulama bölümündeki bütün modeller için keras, tensorflow ve matplotlib kütüphaneleri kullanılmıştır.

Tablo 4.2 Modellerin Mimari Yapılarının Karşılaştırılması

Modeller ve Mimari Bilgileri	Konvolüsyon Katmanı sayısı	Havuzlama Katmanı sayısı	Relu Katmanı Sayısı	Dense Katmanı Sayısı	Parametre Sayısı	DroupOut Katmanı Durumu
Alex Net	5	3	7	3	26.004.894	Yok
Yeni Model	14	7	16	3	46.197.314	Yok
VGG-16	13	5	14	2	23.129.945	Var

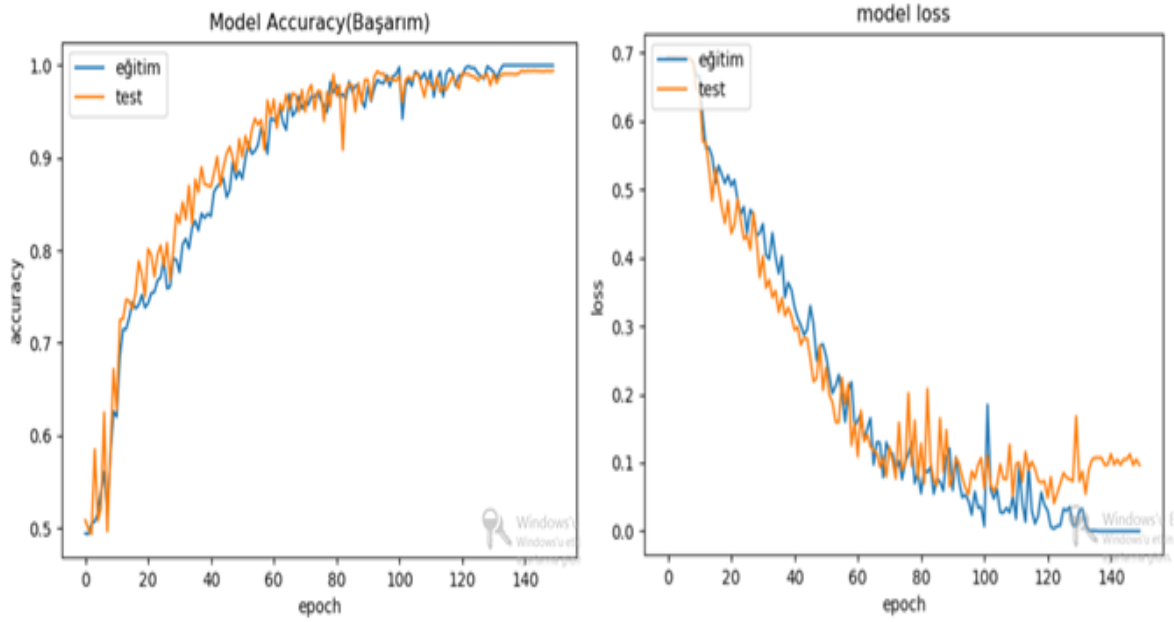
Hazırlanan veri setinin üç farklı modelde çalıştırılması ile ortaya çıkan sonuçlar tablo 4.3’de gösterilmiştir. Veri setinin uygulama içinde en uyumlu olduğu modeli seçmek üzere tablo incelenmiştir. Model tasarlanırken mini-batch parametresi olarak belirlenen değer; modelin aynı anda kaç veriyi işleyeceğini ifade etmektedir [35]. Epoch sayısı ise modelin kaç iterasyon çalışacağını saptayan bir hiper parametredir. Bütün modeller için aynı mini batch sayısı ve epoch sayısı belirlenmesinin nedeni modellerin başarımlarının karşılaştırmasının, model algoritmasına göre yapılmasını sağlamaktır. Mini batch ve epoch sayısını artırma ya da azaltma gibi işlemlerin başarımlarını değiştirebileceği düşünülmektedir.

Tablo 4.3 Hazırlanan Veri Seti ile Çalıştırılan Model Sonuçları

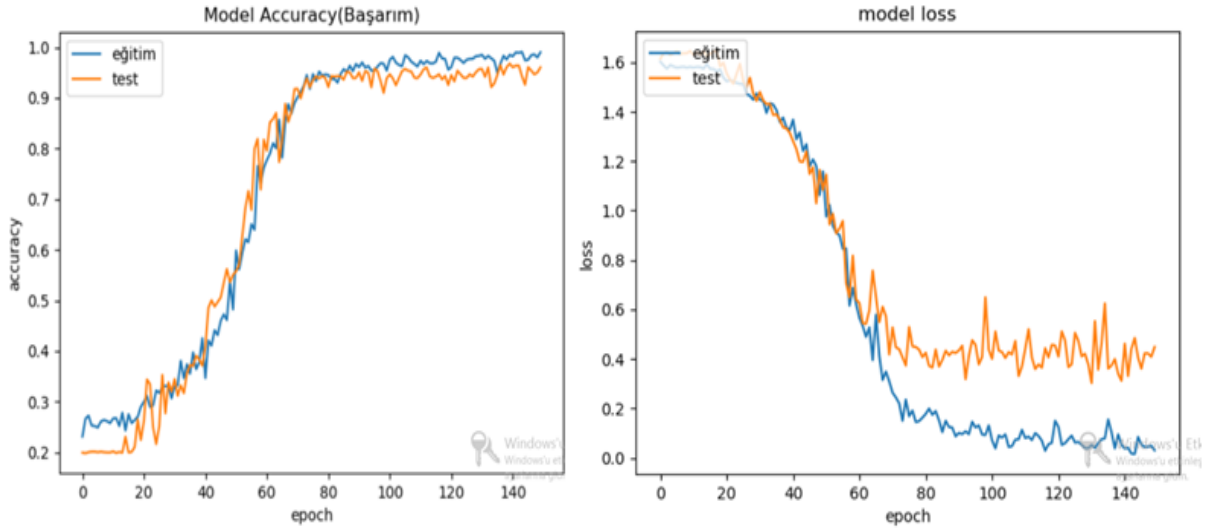
YAŞ	Model Türü	Mini Batch Sayısı	Katman Sayısı	Epoch Sayısı	Optimizer	En iyi Başarım %' si
	Alex Net	32	25	150	SGD	20,80%
	Önerilen Model	32	41	150	SGD	26,08%
	VGG-16	32	41	150	SGD	96,70%
CİNSİYET	Model Türü	Mini Batch Sayısı	Katman Sayısı	Epoch Sayısı	Optimizer	En iyi Başarım %' si
	Alex Net	32	25	150	SGD	65,63%
	Önerilen Model	32	41	150	SGD	72,20%
	VGG-16	32	41	150	SGD	99,41%

Sonuçlardan görüldüğü üzere en iyi doğruluk yüzdesini, sentetik veri setinin kullanıldığı VGG-16 modelinde vermektedir. VGG-16 modelinin 150 iterasyon çalıştırılması sonucu ortaya çıkan başarımlar ve kayıp değer grafikleri sonuç kısmındaki şekilde gösterilmiştir.

CPU ile çalışmanın uzun sürmesi nedeniyle, eğitimler için GPU destekli bilgisayarlar tercih edilmiştir. Minibatch sayısı artırmanın ve epoch sayısı artırmanın doğruluk yüzdesini artırırken süreyi uzattığı tespit edilmiştir. Derin öğrenme algoritmalarını kullanarak cinsiyet tahmini yapan bir uygulama örneği olarak yapılan çalışmada, derlenen bir veri setinin modellerdeki performansı kıyaslanmıştır. Bu kıyaslamada en iyi performansı VGG-16 modelinin verdiği gözlemlenmiştir. Geliştirilen modelin ise veri setine uyumunun, Alex Net modelinden daha iyi ve VGG-16 modelinden ise daha kötü bir sonuç verdiği belirlenmiştir. Bu yüzden, modelin daha fazla denemelerle geliştirilmesi gerektiği saptanmıştır. Veri çeşitliliği, veri sayısı artırımı, minibatch artırımı, optimizier farklılaştırma, katmanları artırma gibi işlemlerin denenmesi devam edilerek daha iyi sonuçlar bulunması hedeflenmektedir. Şekil 4.7.'de en iyi sonucu veren modelin başarımları ve kayıp grafikleri gösterilmektedir.



Şekil 4.7. Çalıştırılan VGG-16 modelin cinsiyet başarımları ve kayıp grafiği



Şekil 4.8. Çalıştırılan VGG-16 modelin yaş başarımları ve kayıp grafiği

VGG-16 modeli ile uyumlu olduğu görülen veri setine göre, 150 iterasyon ile çalışan modelin %99,41 cinsiyet başarımları elde ettiği tespit edilmiştir. 150 iterasyon ile çalışan VGG- 16 yaş modelinin %96,70 yaş başarımları elde ettiği tespit edilmiştir. 140 iterasyondan sonra öğrenmenin doygunluğa ulaştığı ve %99’da ve %96 civarında kaldığı, grafikler incelendiği zaman fark edilmektedir. Bu uygulama sonucunda elde edilen VGG-16 başarımları ve kayıp grafiklerindeki eğriler analiz edildiğinde, amaçlanan öğrenmenin gerçekleştirildiği görülmektedir. 150 iterasyon ile çalışan VGG- 16 yaş modelinin %96,70 yaş başarımları elde ettiği tespit edilmiştir. Derin öğrenme çalışmalarının mimari modellerinden bazıları Matlab’de hazır toolbox içerisinde mevcuttur. Bu tez çalışması ise pycharm üzerinde Python diliyle geliştirilmiş kodlamaları içermektedir. Yaş ve cinsiyet tespiti çalışması için kullanılan derin öğrenme modellerinden Alex Net kodlama, önerilen model ve VGG-16 kodlama yapısı ekteki gibidir. Keras kütüphanesi kullanılan, sequential model ve sgd aktivasyon fonksiyonu içeren kod yapısı ek belgelerde detaylandırılarak sunulmuştur. Her katmanın kodlaması ifade edilmiştir [35, 36].

BÖLÜM 5. TARTIŞMA VE SONUÇ

Bu çalışmada 3 farklı veri seti kullanılarak yaş ve cinsiyet tahmini uygulamaları yapılmıştır. Bu veri setlerinden kullanılan ilk 2 veri seti literatürde bu çalışmayı başarılı bir şekilde gerçekleştiren derin öğrenme çalışmalarının veri setleridir. Kullanılan ilk veri seti “dataset-images of groups of people” içinde yer alan düğün, aile ve grup fotoğraflarının olduğu verileri içermektedir. Bu çalışmada Deep ConvNets tabanlı yeni bir model üretilmiş ve çalıştırılmıştır.

Kullanılan veri seti ‘Aidience Benchmark Dataset’ olup, derin öğrenme algoritmalarından ‘Caffe Model’den faydalanılmıştır. Caffe model yaş ve cinsiyet dosyaları ile tahminlemeler yapılmıştır.

Her iki çalışmanın veri setleri ve modellerin çalıştırılması, analiz edilmesi ve test aşamaları yapılarak konunun daha anlaşılır bir biçimde ifade edilmesi sağlanmıştır. Yapılan çalışmada, önceden hazırlanan literatürdeki modellerin çalıştırılması ile tez için hazırlanan modellerin çalıştırıp karşılaştırılması sağlanarak literatüre farklı bir açıdan katkıda bulunmak istenmiştir. Vikipedi görüntülerinden derlenen veri seti ile 3 farklı modele göre yaş-cinsiyet tahmini için karşılaştırmalar yapılmıştır. Bu modellerden Alexnet ve VGG-16 dışında bir model geliştirilerek performans test edilmiştir. Çalıştırılan modellerin sonuçları değerlendirildiğinde yaş ve cinsiyet tahmini için hazırlanan veri setine en uygun modelin VGG-16 modeli olduğu görülmüştür. Yaş tahmini için %96 civarı ve cinsiyet tahmini için %99 civarı başarı elde edilmiştir. Başarılı sayılabilecek bu yüzdelerin daha gelişmiş modellerle daha fazla başarı elde edebileceği tahmin edilmektedir. Donanımsal kısıtlardan dolayı; ekran kartı ve işlemci yetersizliği gibi kısıtlar sebebiyle daha gelişmiş modellerin çalıştırmasına imkân bulunamamıştır.

Çalışma için eğitilen veri setinin doğru çalışıp çalışmadığını kontrol etmek için 2 farklı resim ile tahminleme testi yapılmıştır. VGG-16'dan elde edilen tahminleme sonucu aşağıdaki şekilde görünmektedir.



Şekil 5.1. Deneme resimleri için predict sonuçları

Gerçekte 25 yaşında erkek olan resim, VGG-16 predict için 15-25 yaş aralığında erkek olarak doğru tahmin edilmiştir. Gerçekte 21 yaşında olan kadın resmi de 15-25 yaş aralığı kadın olarak yine doğru tahmin edilmiştir.

Tez geliştirilip, gelecekte gerçek zamanlı ve videolardan görüntü alan bir derin öğrenme çalışması haline getirilebilir. Bu tez çalışmasının, revize edilen veri setiyle model uyumu göz önüne alındığında, diğer çalışmalara kıyasla daha yüksek yüzde ile başarımlar sağlamış bir yaş-cinsiyet tespiti çalışması olduğu söylenebilir. Çalışmanın esas özgün yönü ise 3 farklı modelle kıyaslanmış sonuçların değerlendirilmesi ve var olan modellere ek bir model önerilmesidir. Aynı veri setinin farklı modellerdeki performansları değerlendirildiğinde, veri seti-model uyumunun önemini açıklamayı amaçlayan ve bu durumun başarıma etkisini vurgulayan çalışma ile literatüre farklı bakış açıları kazandırmak hedeflenmiştir.

Sonuç olarak çalışmanın aşağıda belirtilen durumlara fayda sağlayacağı düşünülmektedir:

- a. Alışveriş merkezlerinde, mağazalarda ya da marketlerde belirlenen herhangi bir ürün ile ilgilenen kişilerin yaş ve cinsiyet demografisini çıkararak ürüne talep duyulan cinsiyet türünü ve yaş aralığını belirleyebilir. Bu bilgi ile hedef kitleye uygun reklamlar yapılabilir veya elde edilen bilgiler ışığındaki taleplere göre ürünün üretim planlaması yapılabilir.
- b. Bir siyasi mitingde, konferansta ya da kongredeki katılımcı görüntülerinden hitap edilen hedef kitlenin yaş ve cinsiyet bilgisi elde edilebilir. Bu sayede hedef kitlenin isteklerine uygun konuşma içerikleri düzenlenebilir, vaatler verilebilir, örnekler bu yaş grubu üzerine yoğunlaşabilir. Yoğunluğun az görüldüğü yaş ve cinsiyet grubunun dikkati çekilerek konu ve durum için cazip durumlar sağlanabilir.
- c. Birçok teknoloji devi şirketin de yaptığı gibi eğlence amaçlı kullanılabilir. Yeni model akıllı telefonlarda bir uygulama olarak geliştirilen ya da internet ortamında yaşı ve cinsiyeti tahmin eden diğer uygulamalar kullanıcılar tarafından büyük ilgi görmektedir.
- d. Sosyal medya uygulamaları için kullanışlı bir uygulama olduğu bilinmekle birlikte, hala daha iyi doğruluk başarımı veren uygulamalar için çalışmalar yapılmaktadır. Facebook ve Google firmalarının Caffé model kullanarak yaş ve cinsiyet tespiti yaptığı çalışmalara bir öneri çalışma olarak sunulabilecek şekilde hazırlanmış tez, bu alandaki açıklıklar içinde katkıda bulunmayı hedeflemektedir.

KAYNAKLAR

- [1] <https://www.log.com.tr/tek-fotograftan-yas-tahmini-yapan-siteye-buyuk-ilgi/>, Eriřim Tarihi: 05.03.2019.
- [2] <https://www.teknoblog.com/xiaomi-mi-6-on-kamera-tahmin/>, Eriřim Tarihi: 05.03.2019.
- [3] Rasmus Rothe, Radu Timofte, Luc Van Gool; “DEX: Deep EXpectation of Apparent Age From a single image”, ICCV, 2015.
- [4] <https://www.python.tc/django-framework-kurulumu/>, Eriřim Tarihi: 05.03.2019.
- [5] Bariř, C., Kwang Gi Kim, Deep Learning, PhD, Biomedical Engineering Branch, Division of Precision Medicine and Cancer Informatics, National Cancer Center, Goyang, Korea, 2016.
- [6] Abdulkadir řeker, Banu Diri, Hasan Hüseyyin Balık, Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme, Gazi Mühendislik Bilimleri Dergisi, 2017.
- [7] B.A. Golomb, D.T. Lawrance and T.J.Sejnowski; Sexnet: A Neural Network Identifies Sex From Human Faces, In Neural Inform Processing System, 1991.
- [8] Chuan-Hui Ju ; Yun-Hong Wang, Automatic age estimation based on local feature of face image and regresion, Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009.
- [9] Yuan Dong, Yinan Liu, Shiguo Lian, Automatic age estimation based on deep learning algorithm, Neurocomputing, 2016.
- [10] Gil Levi and Tal Hassner, Age and Gender Classification using Convolutional Neural Networks, Computer Vision Foundation 2015.
- [11] Eran Eiding, Roe Enbar, Tal Hassner; Age and Gender Estimation of Unfiltered Faces, IEEE, 2013.

- [12] Y. LeCun, Y. Bengio, and G. Hinton, Deep Learning, *Nature*, vol. 521, pp. 436–444, 2015.
- [13] Jiguan Ngiam et al, Multimodal Deep Learning, Appearing in Proceedings of the 28 th International Conference on Machine Learning, Bellevue, WA, USA, 2011.
- [14] <http://www.derinogrenme.com/2015/07/21/derin-ogrenme-deep-learning-nedir/> , Eriřim Tarihi: 05.03.2019.
- [15] <https://hub.packtpub.com/top-5-deep-learning-architectures/>, Eriřim Tarihi: 15.04.2019.
- [16] Abdulkadir řeker, Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme, Doktora Semineri, Bilgisayar Müh. Bölümü, İstanbul, 2017.
- [17] Y. Le Cun *et al.*, Handwritten digit recognition: applications of neural network chips and automatic learning, *IEEE Commun. Mag.*, vol. 27, no. 11, pp. 41–46, Nov. 1989.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.
- [19] Özkan İnik, Erhan Ülker; “Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri”, *Gaziosmanpasa Journal of Scientific Research*, Cilt/Volume: 6 Sayı/Number: 3 Sayı/Pages: 85-104, 2017.
- [20] <https://medium.com/deep-learning-turkiye/deri%CC%87ne-dahaderi%CC%87ne-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-2813a2c8b2a9>, Eriřim Tarihi: 13 Nisan 2019
- [21] Young Jin Cha et al, “Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types” , *Computer Aided Civil And Insfractructure Engineering*, DOI: 10.1111, Nov.2017.
- [22] <https://neurohive.io/en/popular-networks/vgg16/> , Eriřim Tarihi: 13 Nisan 2019.
- [23] Merve Ayyüce Kızrak, Bülent Bolat, Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Arařtırma, *Literatür Makalesi/Review Article*, *Biliřim Teknolojileri Dergisi*, Cilt: 11, Sayı: 3, Temmuz 2018.

- [24] Junbo Zhang et al, “Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction”, Proceeding of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) | November 2016.
- [25] <http://www.derinogrenme.com/caffe/> , Eriřim Tarihi: 31 Mart 2019.
- [26] <https://www.slideshare.net/bluekid/caffe-klavuzu> , Eriřim Tarihi: 10 Nisan 2019.
- [27] <https://www.tensorflow.org/> , Eriřim Tarihi: 10 Nisan 2019.
- [28] <https://medium.com/@mccemilg/tensorflowu-anlamak-fb84e1a5c4bd> , Eriřim Tarihi: 10 Nisan 2019.
- [29] <https://keras.io/> , Eriřim Tarihi: 10 Nisan 2019.
- [30] <https://devhunteryz.wordpress.com/2019/01/15/keras-ile-derin-ogrenme-temel-bilgiler/> , Eriřim Tarihi: 10 Nisan 2019.
- [31] <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4> , Eriřim Tarihi: 10 Nisan 2019.
- [32] S. Escalera, J. Fabian, P. Pardo, X. Baro, J. Gonzalez, H. J. Escalante, and I. Guyon, Chalearn Looking at People 2015: Apparent Age and Cultural Event Recognition Datasets and Results. In ICCV, ChaLearn Looking at People workshop, 2015.
- [33] <http://derindelimavi.blogspot.com/2015/11/derin-ogrenme-2.html>, Eriřim Tarihi: 4 Mart 2019.
- [34] <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4>, Eriřim Tarihi: 4 Mart 2019.
- [35] <https://github.com/eweill/kerasdeepcv/blob/master/models/classification/alexnet.py> , Eriřim Tarihi: 14 Nisan 2019.
- [36] <https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3>, Eriřim Tarihi: 14 Nisan 2019.

EKLER

EK A : Alex Net Kod Yapısı

```
"""
    "AlexNet Keras Implementation

    BibTeX Citation:
    @inproceedings{krizhevsky2012imagenet,
        title={Imagenet classification with deep convolutional
        neural networks},
        author={Krizhevsky, Alex and Sutskever, Ilya and
        Hinton, Geoffrey E},
        booktitle={Advances in neural information processing
        systems},
        pages={1097--1105},
        year={2012}
    }
    """

    # Import necessary packages
    import argparse

    # Import necessary components to build LeNet
    from keras.models import Sequential
    from keras.layers.core import Dense, Dropout, Activation,
    Flatten
    from keras.layers.convolutional import Conv2D,
    MaxPooling2D, ZeroPadding2D
    from keras.layers.normalization import BatchNormalization
    from keras.regularizers import l2

    def alexnet_model(img_shape=(224, 224, 3), n_classes=10,
        l2_reg=0.,
        weights=None):
        # Initialize model
        alexnet = Sequential()

        # Layer 1
        alexnet.add(Conv2D(96, (11, 11),
            input_shape=img_shape,
            padding='same',
            kernel_regularizer=l2(l2_reg)))
        alexnet.add(BatchNormalization())
        alexnet.add(Activation('relu'))
```



```

alexnet.add(MaxPooling2D(pool_size=(2, 2)))

# Layer 2
alexnet.add(Conv2D(256, (5, 5), padding='same'))
alexnet.add(BatchNormalization())
alexnet.add(Activation('relu'))
alexnet.add(MaxPooling2D(pool_size=(2, 2)))

# Layer 3
alexnet.add(ZeroPadding2D((1, 1)))
alexnet.add(Conv2D(512, (3, 3), padding='same'))
alexnet.add(BatchNormalization())
alexnet.add(Activation('relu'))
alexnet.add(MaxPooling2D(pool_size=(2, 2)))

# Layer 4
alexnet.add(ZeroPadding2D((1, 1)))
alexnet.add(Conv2D(1024, (3, 3), padding='same'))
alexnet.add(BatchNormalization())
alexnet.add(Activation('relu'))

# Layer 5
alexnet.add(ZeroPadding2D((1, 1)))
alexnet.add(Conv2D(1024, (3, 3), padding='same'))
alexnet.add(BatchNormalization())
alexnet.add(Activation('relu'))
alexnet.add(MaxPooling2D(pool_size=(2, 2)))

# Layer 6
alexnet.add(Flatten())
alexnet.add(Dense(3072))
alexnet.add(BatchNormalization())
alexnet.add(Activation('relu'))
alexnet.add(Dropout(0.5))

# Layer 7
alexnet.add(Dense(4096))
alexnet.add(BatchNormalization())
alexnet.add(Activation('relu'))
alexnet.add(Dropout(0.5))

# Layer 8
alexnet.add(Dense(n_classes))
alexnet.add(BatchNormalization())
alexnet.add(Activation('softmax'))

if weights is not None:
    alexnet.load_weights(weights)

return alexnet

```

```

def parse_args():
    """
    Parse command line arguments.

    Parameters:
        None

    Returns:
        parser arguments
    """

    parser =
argparse.ArgumentParser(description='AlexNet model')
    optional = parser._action_groups.pop()
    required = parser.add_argument_group('required
arguments')
    optional.add_argument('--print_model',
                           dest='print_model',
                           help='Print AlexNet model',
                           action='store_true')
    parser._action_groups.append(optional)
    return parser.parse_args()

if __name__ == "__main__":
    # Command line parameters
    args = parse_args()
    # Create AlexNet model
    model = alexnet_model()
    # Print
    if args.print_model:"

```

EK B:Önerilen Model Kod Yapısı

```

img_width=128
img_height=128
train_data_dir='/home/gulgunduz/Downloads/train'

test_data_dir='/home/gulgunduz/Downloads/test'

train_samples=3170 #eğitilen veri sayısı
test_samples=318 #test veri sayısı
nb_epochs=150 #iterasyon sayısı
nbbatch_size=32 #minibatch sayısı
num_class=2 #cinsiyet için 2 sınıf, yaş için 5 sınıf
sgd = optimizers.SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)

model = models.Sequential() #başlangıç seviyesi modeller için
#Öznitelik çıkarımı, filtre matrisleri ve katmanlar
model.add(layers.Conv2D(64, (7, 7), activation='relu', padding='same',
input_shape=(img_width,img_height, 3)))
model.add(layers.Conv2D(64, (5, 5), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))

model.add(layers.Conv2D(128, (5, 5), activation='relu', padding='same'))
model.add(layers.Conv2D(128, (5, 5), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))

model.add(layers.Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(layers.Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))
model.add(layers.Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(layers.Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))

model.add(layers.Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(layers.Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))
model.add(layers.Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(layers.Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))

model.add(layers.Conv2D(1024, (3, 3), activation='relu', padding='same'))
model.add(layers.Conv2D(1024, (3, 3), activation='relu', padding='same'))

```

```
model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))
```

```
model.add(layers.Dense(4096, activation='relu'))
```

```
model.add(layers.Dense(4096, activation='relu'))
```

```
model.add(layers.Flatten())
```

```
model.add(layers.Dense(2, activation='softmax'))
```

```
model.compile(optimizer=sgd,  
              loss='categorical_crossentropy',  
              metrics=['accuracy'])
```

```
model.summary()
```

```
print("Starting to fit the model")
```

```
train_datagen = ImageDataGenerator(rescale=1./255)
```

```
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory(  
    train_data_dir,  
    target_size=(img_width, img_width),  
    batch_size=nbbatch_size,  
    class_mode='categorical')
```

```
validation_generator = test_datagen.flow_from_directory(  
    test_data_dir,  
    target_size=(img_width, img_width),  
    batch_size=nbbatch_size,  
    class_mode='categorical')
```

EK C:VGG-16 Kod Yapısı

```

“from
keras.models
import
Sequential

from keras.layers.core import Flatten, Dense, Dropout
from keras.layers.convolutional import Convolution2D, MaxPooling2D,
ZeroPadding2D
from keras.optimizers import SGD
import cv2, numpy as np
def VGG_16(weights_path=None):
    model = Sequential()
    model.add(ZeroPadding2D((1,1),input_shape=(3,224,224)))
    model.add(Convolution2D(64, 3, 3, activation='relu'))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(64, 3, 3, activation='relu'))
    model.add(MaxPooling2D((2,2), strides=(2,2)))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(128, 3, 3, activation='relu'))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(128, 3, 3, activation='relu'))
    model.add(MaxPooling2D((2,2), strides=(2,2)))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(256, 3, 3, activation='relu'))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(256, 3, 3, activation='relu'))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(256, 3, 3, activation='relu'))
    model.add(MaxPooling2D((2,2), strides=(2,2)))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(512, 3, 3, activation='relu'))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(512, 3, 3, activation='relu'))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(512, 3, 3, activation='relu'))
    model.add(MaxPooling2D((2,2), strides=(2,2)))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(512, 3, 3, activation='relu'))
    model.add(ZeroPadding2D((1,1)))
    model.add(Convolution2D(512, 3, 3, activation='relu'))
    model.add(ZeroPadding2D((1,1)))

```

```

model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))
model.add(Flatten())
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1000, activation='softmax'))
if weights_path:
    model.load_weights(weights_path)
return model
if __name__ == "__main__":
    im = cv2.resize(cv2.imread('cat.jpg'), (224,
224)).astype(np.float32)
    im[:, :, 0] -= 103.939
    im[:, :, 1] -= 116.779
    im[:, :, 2] -= 123.68
    im = im.transpose((2,0,1))
    im = np.expand_dims(im, axis=0)
    # Test pretrained model
    model = VGG_16('vgg16_weights.h5')
    sgd = SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)
    model.compile(optimizer=sgd, loss='categorical_crossentropy')
    out = model.predict(im)
    print np.argmax(out)

```

EK D:Tahmin için Kullanılan Veri Seti Harici Orijinal Test Resimleri

ÖZGEÇMİŞ

Gül Gündüz, 17.04.1992’de Elazığ’da doğdu. İlkokulu Kırşehir’de Cumhuriyet İlköğretim Okulu’nda, ortaokulu ve lise eğitimini Elazığ’da tamamladı. 2010 yılında Çubuk Bey Anadolu Lisesi’nden mezun oldu. 2011 yılında başladığı Sakarya Üniversitesi Endüstri Mühendisliği Bölümü’nü 2016 yılında bitirdi ve 2013 yılında başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü 2018 yılında bitirdi. 2017 yılında Sakarya Üniversitesi Endüstri Mühendisliği Bölümü’nde yüksek lisans eğitimine başladı. 2019 Yılı Nisan ayında Abdullah Gül Üniversitesi, Endüstri Mühendisliği Bölümü/Araştırma Görevlisi olarak başladığı görevine hala devam etmektedir.