



T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ZEKA TABANLI ARAÇ KOLTUĞU TANIMA SİSTEMİNİN
GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Ali İhsan BADEM

Akıllı Sistemler Mühendisliği Anabilim Dalı

Nisan 2023

T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ZEKA TABANLI ARAÇ KOLTUĞU TANIMA SİSTEMİNİN
GELŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Ali İhsan BADEM
(19324814039)

Akıllı Sistemler Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. İzzet Fatih ŞENTÜRK

Nisan 2023

BTÜ, Lisansüstü Eğitim Enstitüsü'nün 19324814039 numaralı Yüksek Lisans Öğrencisi Ali İhsan BADEM, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “Yapay Zeka Tabanlı Araç Koltuğu Tanıma Sisteminin Geliştirilmesi ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. İzzet Fatih ŞENTÜRK**
Bursa Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Metin BİLGİN**
Uludağ Üniversitesi

Dr. Öğr. Üyesi Ergün GÜMÜŞ
Bursa Teknik Üniversitesi

Teslim Tarihi :
Savunma Tarihi : 13 Nisan 2023



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, Bursa Teknik Üniversitesi’nin aboneli olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

İNTİHAL BEYANI

Bu tezde görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, tez içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde kaynak göstererek belgelediğimi, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Öğrencinin Adı Soyadı: Ali İhsan BADEM

İmzası :

X X X X

Eşime ve çocuklarıma,



ÖNSÖZ

Tezimde desteđini esirgemeyen, yol gösteren kıymetli sayın danışman hocam Doç. Dr. İzzet Fatih ŞENTÜRK'e teşekkür ve saygılarımı sunarım.

Hayatım boyunca her türlü desteđi ile yanımda olan başta sevgili eşim Tuğba BADEM ve biricik kızım İdil BADEM'e; varlığı ile güç veren, ilham olan kıymetli annem Ayşe BADEM'e sonsuz minnetlerimi ve teşekkürlerimi sunarım.

Nisan 2023

Ali İhsan BADEM
(Endüstri Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	viii
KISALTMALAR	x
SEMBOLLER	xi
ÇİZELGE LİSTESİ.....	xii
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xiv
SUMMARY	xv
1. GİRİŞ	1
1.1 Literatür Araştırması	2
1.1.1 Yapay zeka	2
1.1.2 Makine öğrenmesi	3
1.1.2.1 Denetimli öğrenme.....	3
1.1.2.2 Denetimsiz öğrenme.....	4
1.1.3 Derin öğrenme.....	5
1.1.3.1 Yapay sinir ağları	5
1.1.3.2 Tek katmanlı algılayıcılar	7
1.1.3.3 Çok katmanlı algılayıcılar	7
1.1.3.4 İleriye yayılım	8
1.1.3.5 Maliyet fonksiyonunun belirlenmesi.....	9
1.1.3.6 Geriye yayılım.....	9
1.1.4 Benzer çalışmalar	10
2. EVRİŞİMSEL SİNİR AĞLARI	12
2.1 Evrişim Katmanı	12
2.2 Ortaklama Katmanı	12
2.3 Tam Bağlantı Katmanı	13
2.4 Evrişimsel Sinir Ağı Modelleri	13
2.5 Hipotez	13
3. KOLTUK MODELLERİNİN BELİRLENMESİ.....	15
3.1 Kullanılan Yöntemler.....	15
3.1.1 Anacoda Python ve OpenCV kütüphanesi.....	15
3.1.2 Tensorflow kütüphanesi	15
3.1.3 Numpy kütüphanesi	16
3.1.4 Keras kütüphanesi	16
4. KULLANILAN VERİ SETİ VE YÖNTEMLER.....	17
4.1 Faster RCNN Modelinin Oluşturulması ve Sonuçları	18
4.2 RetinaNet Modelinin Oluşturulması ve Sonuçları	22
4.3 EfficientDet Modelinin Oluşturulması ve Sonuçları	24
4.4 Tüm Modellerin Karşılaştırılması	25
5. SONUÇ VE ÖNERİLER.....	29

5.1 Çalışmanın Uygulama Alanı	30
KAYNAKLAR	31
ÖZGEÇMİŞ.....	33



KISALTMALAR

CNN	: Convolutional Neural Network
CPU	: Central Process Unit
DVM	: Destek Vektör Makinesi
ESA	: Evrişimsel Sinir Ağları
Fast R-CNN	: Fast Region Convolutional Neural Network
Faster R-CNN	: Faster Region Convolutional Neural Network
IoU	: Birleşim üzerinden Kesişim (Intersection over Union)
MÖ	: Makine Öğrenmesi
R-CNN	: Region Convolutional Neural Network
ResNet	: Residual Network
RPN	: Region Proposal Network
SSD	: Single Shot MultiBox Detector
TF	: Tensor Flow
VOC	: Visual Object Classes
YSA	: Yapay Sinir Ağları

SEMBOLLER

b	: Yanlılık değeri
m	: Örnek sayısı
t	: Zaman
w	: Ağırlık



ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 4.1 : Eğitim ve Test sayısı.	17
Çizelge 4.2 : Test modelinin başarı oranları.	22
Çizelge 4.3 : Tensorflow veri seti doğruluk oranı tablosu.	22
Çizelge 4.4 : RetinaNet veri seti doğruluk oranı tablosu.	23
Çizelge 4.5 : EfficientDet veri seti doğruluk oranı tablosu.	25
Çizelge 4.6 : Modellerin başarı oranları	26

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Yapay zeka, makine öğrenmesi derin öğrenme kapsamı.	3
Şekil 1.2 : Denetimli makine öğrenme akışı.....	4
Şekil 1.3 : Sinir hücre yapı şeması.....	6
Şekil 1.4 : Yapay sinir hücresi.....	6
Şekil 1.5 : Tek katmanlı algılayıcılara ait model.....	7
Şekil 1.6 : Çok katmanlı algılayıcılara ait model.....	8
Şekil 1.7 : RetinaNet modeli ile yüz maskesi algılama.....	10
Şekil 1.8 : EfficientDet modeli ile Glokom hastalığı tespit örneği.....	11
Şekil 1.9 : Faster R-CNN modeli ile metal levhalar üzerindeki hata tespiti.....	11
Şekil 2.1 : Evrişim işlemi.....	12
Şekil 2.2 : Maksimum ve ortalama havuzlama katmanı.....	13
Şekil 4.1 : Koltuk etiketleme örnek görüntüsü.....	17
Şekil 4.2 : Test veri setine ait csv örneği.....	18
Şekil 4.3 : Model eğitim adımları.....	19
Şekil 4.4 : Sınıflandırma kayıp grafiği.....	20
Şekil 4.5 : Yerelleştirme kayıp grafiği.....	20
Şekil 4.6 : Düzenleştirme artış grafiği.....	21
Şekil 4.7 : Toplam kayıp grafiği.....	21
Şekil 4.8 : RetinaNet Modelinin Yapısı.....	23
Şekil 4.9 : EfficientDet modelin yapısı.....	24
Şekil 4.10 : Trend AGR koltuğunun genel görünümü.....	26
Şekil 4.11 : Trend AGR koltuk modelinin test sonucu.....	27
Şekil 4.12 : Active, trail ve carpet modellerinin test sonucu genel görünümü.....	27
Şekil 4.13 : Koltuk modellerinin tespit edilebildiği örnek test sonuçları.....	28

YAPAY ZEKA TABANLI ARAÇ KOLTUĞU TANIMA SİSTEMİNİN GELİŞTİRİLMESİ

ÖZET

Motorlu taşıtların tasarlandığı, üretildiği ve pazarlanarak satıldığı iş alanına otomotiv sektörü denilmektedir. Sektörün yaklaşık %70'inde otomobil imalatları yer almaktadır. Otomobili oluşturan birçok parça bulunmaktadır. Motor, tekerlek, metal gövde ve kapılar, koltuk, plastik paneller gibi yaklaşık 5.000 parçadan oluşur. Her parçayı imal eden yan sanayiler de sektörde önemli role sahiptir. Bu çalışmada, koltuk üreten bir yan sanayi firmasındaki bir araç modeline ait koltuk tipleri incelenmiştir.

Otomotiv sektöründe, senelik yaklaşık 300.000 adet sadece bir modele ait araç üretimi gerçekleştirilmektedir. Her bir aracın ön sırasında bir adet sürücü koltuğu ve bir adet yolcu koltuğu olmak üzere, toplam iki adet koltuk bulunmaktadır. Koltukların kumaş tiplerine göre ayırım yapıldığında, altı farklı model bulunmaktadır. Koltuk üreten yan sanayi firmalarında, senelik her model için ortalama 600.000 adet koltuk imalatı gerçekleştirilmektedir. Derin öğrenme temelli bilgisayar görüşü teknolojisinin gelişmesiyle birlikte üretilen koltukların modellerine göre otomatik tanımlanması, stoklanması için uygulama ihtiyacı ortaya çıkmıştır.

Yüksek stok maliyetleri, hatalı üretim ve etiket, otomotiv firmalarının en büyük problemleri arasındadır. Endüstri 4.0 ile birlikte akıllı makineler sayesinde akıllı üretimler ve kontroller yapılabilmektedir. Hızlı, yüksek doğruluk oranı ve düşük maliyetler ile yüksek maliyetli problemlerden uzaklaşmıştır. Yapay zeka teknolojilerinin kullanıldığı akıllı makineler sayesinde, akıllı üretim teknolojileri geliştirilmiştir. Yüksek çözünürlüklü kameralar, hassasiyeti yüksek algılayıcı sensörler, otomasyon taşıyıcı sistemler ile üretilen ürünler daha hızlı ve doğru olmaktadır.

Bu çalışmada, otomobil modeline ait farklı versiyon koltuklarını üreten bir otomotiv yan sanayi firmasında, stok alanı ve üretim öncesi ile sonrası koltuk modellerinin otomatik ayrılması amaçlanmıştır. Bu çalışmada daha önce üretim alanlarında ve stok bölgelerinde konumlandırılmış olan araba koltukları kullanılarak otomatik nesne (koltuk) tespitinin yapılması için yapay zeka yönteminden yararlanılmıştır. Koltuk tiplerine göre eğitim ve test verileri oluşturulmuştur. Her bir koltuğa ait modelin fotoğrafları farklı ortamlarda çekilmiştir. Fotoğraflar model tipine göre tek tek etiketlenerek, test ve eğitim verileri oluşturulmuştur. Faster R-CNN, RetinaNet ve EfficientDet modelleri ile yapılan eğitimler sonucunda, Tensorflow Faster R-CNN modelinde yüksek oranda test sonuçları elde edilmiştir.

Anahtar kelimeler: Model Algılama, RetinaNet, Faster RCNN, EfficientDet, Python

DEVELOPMENT OF ARTIFICIAL INTELLIGENCE BASED VEHICLE SEAT RECOGNITION SYSTEM

SUMMARY

The area of business in which motor vehicles are designed, produced and marketed is called the automotive industry. Automobile manufacturing takes place in approximately 70% of this industry. There are many parts that make up the car. It consists of about 5,000 parts such as engine, wheel, metal body and doors, seat, plastic panels. The sub-industries that manufacture each part also have an important role in the sector. In this study, seat types belonging to a vehicle model in a sub-industry company that produces seats were examined.

In the automotive sector, approximately 300,000 vehicles belonging to only one model are produced annually. There are a total of two seats in the front row of each vehicle, one for the driver and one for the passenger. There are six different models when the seats are differentiated according to the fabric types. In the sub-industry companies that produce seats, an average of 600,000 seats are produced for each model annually. With the development of deep learning-based computer vision technology, the need for an application has emerged for the automatic identification and stocking of the seats produced according to their models.

High inventory costs, faulty production and labels are among the biggest problems of automotive companies. With Industry 4.0, smart productions and controls can be made thanks to smart machines. High cost problems are avoided with fast, high accuracy rate and low costs. Thanks to smart machines in which artificial intelligence technologies are used, smart production technologies have been developed. Products produced with high-resolution cameras, high-sensitivity sensors, automation carrier systems are faster and more accurate.

In this study, it is aimed to automatically separate the stock area and pre-production and post-production seat models in an automotive supplier industry company that produces different version seats of the automobile model. In this study, artificial intelligence method was used for automatic object (seat) detection by using car seats that were previously located in production areas and stock areas. Training and test data were created according to seat types at different environments. Test and training data were created by labeling the photographs one by one according to the model type. As a result of the trainings conducted with Faster R-CNN, RetinaNet and EfficientDet models, high test results were obtained in the Tensorflow Faster R-CNN model.

Keywords: Model Detection, RetinaNet, Faster RCNN, EfficientDet, Python

1. GİRİŞ

2011 yılında ortaya çıkmaya başlayan Endüstri 4.0 ile sanayide kullanılan makine teknolojileri günümüzde çok daha gelişmiştir. Kronolojik sıralama yapıldığında makineler sayesinde üretimlerin başlangıcı, otomatik konveyör ile birlikte seri üretime geçiş ve bilgisayarlı sistemlerin üretime adaptasyonu sağlanmıştır. Günümüzde ise Endüstri 4.0 ile birlikte, otomasyon sistemler ve ileri teknoloji bilgisayar ve görsel ekipmanların kullanımı ile akıllı üretim sistemlerine geçiş yapılmıştır. Üretim sistemlerinde Endüstri 4.0 teknolojileri kullanılmış fabrikalar artık akıllı sıfatı ile nitelendirilmektedir. Akıllı fabrikalar, kullandıkları ileri teknoloji donanımları sayesinde daha kaliteli, daha hızlı ve çok daha ekonomik ürünler elde edebilmektedir. Üretime kazandırdıkları faydanın yanı sıra pazardaki konumu ve gücü artarak rekabet üstünlüğü elde etmektedir. Birçok faydası olan bu sistem sayesinde işletme performansı da olumlu şekilde etkilenmektedir. Çalışmada, akıllı sistemler başlığı altında geliştirilen üretim sistemlerinin işletmede kullanımı araştırılmıştır. Yapay zeka, makine öğrenmesi, yapay sinir ağları gibi bilgilerle ilgili literatür taraması yapılarak detaylar belirtilmiştir. Çalışmanın hipotez bölümünde, Otomotiv Koltuk Üretimi Fabrikası'nda bir uygulama gerçekleştirilmiştir. Uygulama aşamasında, fabrikada kullanılan koltuklar incelenmiş, daha sonra nesne algılama yöntemi ile otomatik koltuk tiplerinin belirlenmesi sağlanmıştır.

Günümüzde otomotiv sektöründe her yıl farklı markaların farklı modelleri piyasaya çıkmaktadır. Maliyet kriterleri ile birlikte, modellerde farklılıklar oluşmaktadır. Yüksek üretim adetleri, otomotiv yan sanayi firmalarında hızlı üretim, doğru stok, hatasız sevkiyat beklentisini beraberinde getirmiştir. Bu sebeple, doğru koltuk modellerinin üretilip, stoklanması ve sevk edilmesi için, yapılan çalışma ile koltuk modellerinin otomatik algılanması sağlanmıştır.

Belirlenecek koltuk modelleri ile birlikte otomatik stoklama, üretim başında ve sonunda etiket doğrulama, karantina bölgesindeki koltuk tiplerinin ayırımı gibi bir çok alanda uygulama yapılabilecektir. Uygulama daha da genişletilerek, farklı araçlara ait koltuk modellerinin de otomatik algılanması sağlanabilir.

1.1 Literatür Araştırması

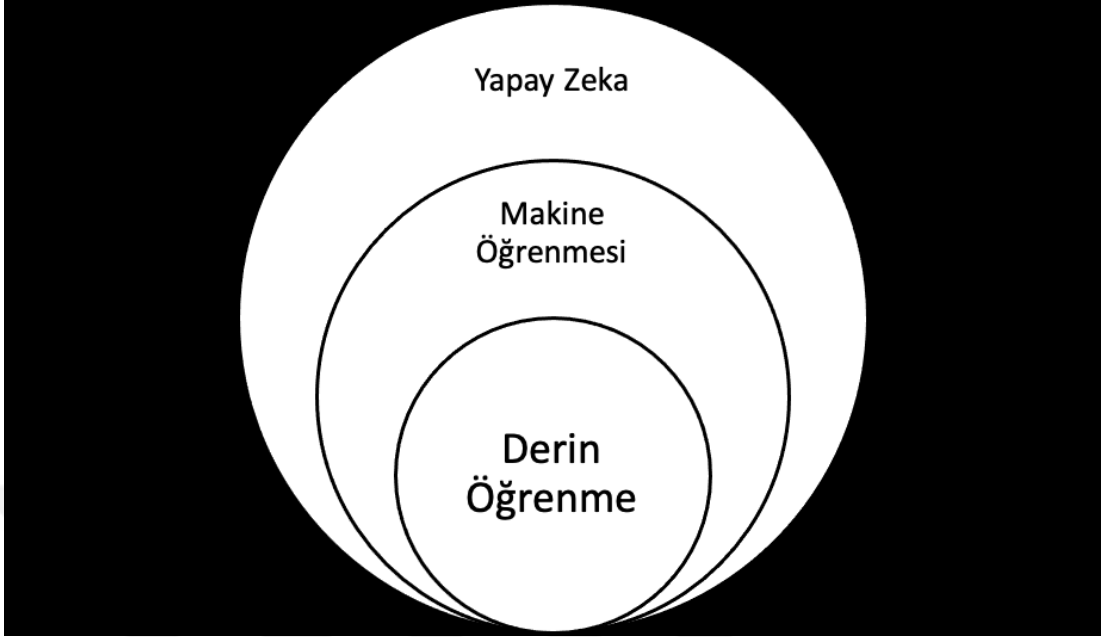
Nesne algılama, bir görüntü veya videoda bulunan nesnelerin yerleştirilmesi ve sınıflandırılmasıyla ilgilenen bilgisayarla görme alanıdır.

Basitçe ifade etmek gerekirse: Nesne algılama, algılanan nesnelerin çevresine, onları belirli bir sahnede (veya içinde nasıl hareket ettiklerini) bulmamıza izin veren sınırlayıcı kutular çizmeye indirgenmesidir.

1.1.1 Yapay zeka

İnsanlar düşünebilen, hareket edebilen, bulunduğu ortamı inceleyen, değerlendirebilen, olabilecekleri öngörebilen canlılardır. Tüm bu yetileri insanlar zekaları sayesinde gerçekleştirmektedir. Yapay zeka, makinelerin akıllı davranabilmesi için insan beynini taklit eden algoritmalarıdır. Bu durumda insanlar, hayvanlar, aynı zamanda makineler de zeki varlıklar olarak nitelendirilebilir. İnsanlar zekaları sayesinde, kendi kendine öğrenebilen, mantıklı düşünebilen, karar verebilen, tepki veren varlıklardır. Hayvanlar ya da akıllı sistemler kullanan makineler ise kültürsüz zeka seviyesindedir [1]. Bilim adamı ve matematikçi Alan Turing, II. Dünya Savaşı'nda "Enigma" adı verilen, karmaşık Alman metinlerini anlayıp çözebilen bir şifre makinesi tasarlayarak, düşünebilen makine fikrini ortaya atmıştır. Böylelikle düşünebilen, algılayabilen, öğrenebilen makinelerin temelini atarak, geleceğe ışık tutmuştur. İnsan zekası sayesinde görme, anlama, karar verme gibi yetileri çok hızlı şekilde gerçekleştirebilirken, karmaşık şekilde verilen matematik modellerini ise daha yavaş ve uzun sürede gerçekleştirebilmektedir. Aynı zamanda, edinilmiş tecrübeler ile davranışlarını kontrol edebilir. Örnek olarak, sobaya yaklaşıp sıcak olduğunu bilmeden dokunan bir kişinin elinin yanması ve acıması sonrasında bir sonraki seferde sobaya yaklaşmayacak ve dokunmayacaktır. Sobanın sıcaklığını ve can yaktığını öğrenen insan beyni bunu tecrübe etmiştir. Bilgisayarlar, çok katmanlı ve zorlu matematik işlemlerini daha hızlı bir şekilde çözebilecek donanımdadırlar. Fakat insanların görme, dokunma gibi çok daha kolay becerileri ise makinelerde yoktur. Yapay zeka, insanların bu becerilerini bilgisayar ve makinelere kazandırmasıdır. İnsan zekası üzerinde yapılan çalışmalar ve makinelerin kabiliyetine uyarlanmasına verilen isim yapay zekadır. Yapay zekanın makine öğrenmesi ve derin öğrenme alt katmanları bulunmaktadır. Şekil 1.1'de kapsam görselleştirilmiştir. İnsanlar gibi makinelerin de

düşünmesini sağlayabilmek için yapılan her çalışma yapay zeka olarak isimlendirilmektedir.



Şekil 1.1 : Yapay zeka, makine öğrenmesi derin öğrenme kapsamı [2].

1.1.2 Makine öğrenmesi

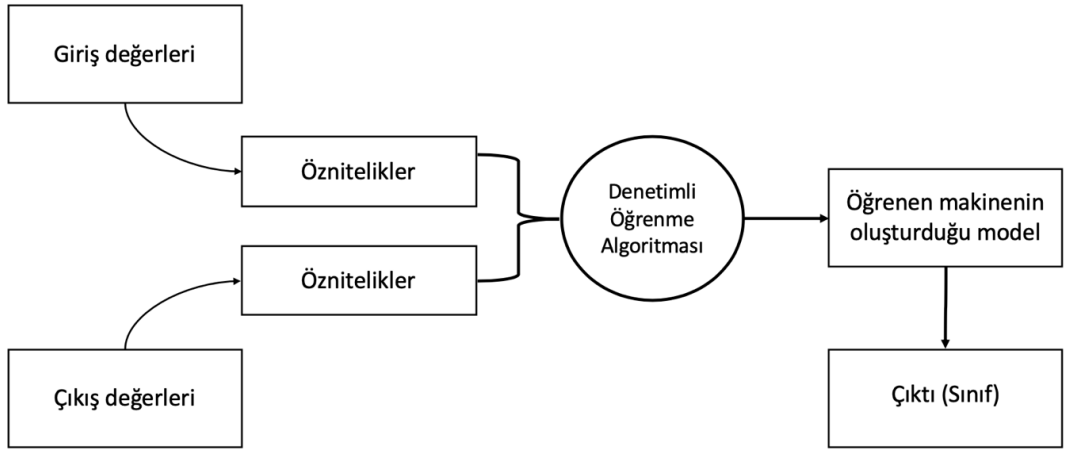
Makine öğrenimi, bilgisayarlara deneyimlerden öğrenmeyi öğreten bir yapay zeka tekniğidir. Makine öğrenimi algoritmaları, bir model olarak önceden belirlenmiş bir denkleme dayanmadan bilgileri doğrudan verilerden öğrenmek için hesaplama yöntemlerini kullanır. Algoritmalar, öğrenme için mevcut örnek sayısı arttıkça performanslarını uyarlamalı olarak geliştirir. Derin öğrenme, makine öğreniminin özel bir biçimidir [3]. Makine öğrenmesindeki algoritmalar, bilgisayar programlamalarındaki gibi sırası ile verilen komutları yerine getirmek yerine, modelin eğitim setinde öğrendiklerini bir genelleme yapması ile bilgileri olması gereken duruma aktarabilmektedir. Modeldeki algoritmalar sayesinde eğitim ile bilgisayara öğrenimin sağlanmasına makine öğrenmesi denilmektedir [4].

1.1.2.1 Denetimli öğrenme

Denetimli öğrenim, bilinen veriler ve girdiler kullanılarak yanıtların alınıp, daha sonra yeni veriler ve girdiler ile makul tahminler oluşturmak üzere bir modeli eğitmesidir. Tahmin edilmesi istenilen çıktılarda bilinen veriler var ise, denetimli öğrenme kullanılabilir. Örneğin; iş yerinden eve gitmek için mesafenin ne kadar süreceğini tahmin edecek makine eğitiminde, hava koşulları, günün saati bilinen girdilerdir. İş

yerinden eve dönmek için geçen süre ise çıktıdır. Yağmur yağdığı zaman eve varış süresinin uzadığı bilinir. Oluşturulması gereken veri seti hava şartları, ev ile iş arasındaki toplam süre gibi bilgilerden oluşacaktır. Makine yağmur miktarı ile varış süresi arasında doğru orantı olduğunu veri setine dayanarak görebilir. Bu durumda yağmur miktarı yükseldikçe, o kadar geç eve varılacağı tespit edilir. Ayrıca işten çıkılan zaman ile varış süresi arasındaki bağlantı da görülebilir. Eğer işten çıkış saatine ne kadar yaklaşıyor ise, eve varışın da aynı oranda gecikileceği bilinir. Makine bu veri sayesinde ilişkiler kurabilir.

Şekil 1.2’de denetimli makine öğrenme akışı gösterilmiştir. Veriler çıktı değerleri veya giriş değerlerinden oluşur. Veriler, öznitelikler ile denetimli öğrenme algoritmasına iletilir. Algoritma ile işlenen veriler makinenin öğrenerek oluşturduğu modeli ile çıktı değerini verir.



Şekil 1.2 : Denetimli makine öğrenme akışı.

1.1.2.2 Denetimsiz öğrenme

Denetimli öğrenmede özniteliklerden iletilen veriler denetimli öğrenme algoritmasına sunulurken, denetimsiz öğrenmede ise denetlemeye gerek kalmadan giriş verilerinin sunulması ile oluşur. Etiketsiz olan çıkış verileri, makine ile birlikte etiketlenecek biçimde sunulmaktadır. Denetimli öğrenmede çıktılar sınıf olarak adlandırılırken, denetimsiz öğrenmede veriler belli bir kategoride olmadan sunulduktan sonra çıktı olarak kategorileşip, bütün olarak elde edilir ve buna kümelene denir.

1.1.3 Derin öğrenme

Yapay zekanın bir alt katmanı makine öğrenmesiydi. Derin öğrenme ise, makine öğrenmenin bir alt katmanı olarak yer almaktadır. Yapay zeka tanımında makinelerin insan zekasını taklit etmesinden bahsedilmiştir. Belirli bir denkleme dayanmadan bilgileri doğrudan veriler ile öğrenebilen algoritmalara makine öğrenmesi denilmektedir. Makine öğrenmesinin bir alt katmanı olan ve insan beyin yapısını simüle etmeye çalışan öğrenme türüne ise derin öğrenme denilmektedir.

İnsan beyin yapısına benzer sonuçların elde edilebilmesi için verilerin mantıksal yapıda derin öğrenme algoritmalar sayesinde analizleri gerçekleştirilir. Doğru sonuçlar ile başarıya ulaşabilmek için, derin öğrenme çok katmanlı algoritma yapısına sahip olan yapay sinir ağları kullanılır.

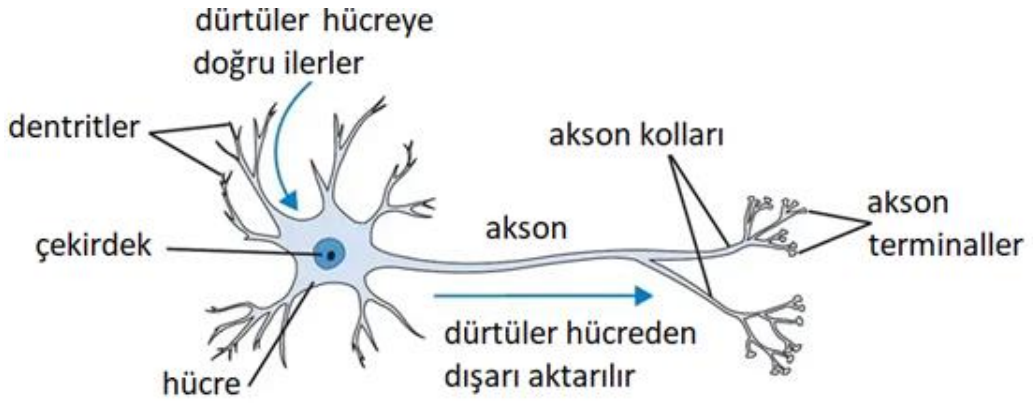
Yapay sinir ağlarının yapısı, insan beynindekiler ile çok benzer yapıdadır. Nesnelerin belirli bir düzen içinde yerleştirilmesi için insanlar beynini kullanırken, yapay sinir ağları ile verilere benzer hedefleri gerçekleştirmesini öğretebiliriz.

Özet olarak, klasik makine öğrenimi algoritmalarının gerçekleştirebildiklerini, sinir ağları da yapabilir (ancak klasik algoritmalar, sinir ağlarıyla aynı görevleri gerçekleştiremez). Başka bir deyişle, yapay sinir ağlarının benzersiz yetenekleri sayesinde çözülmesi imkânsız gözükken görevleri başarıya ulaştırabilmektedir.

1.1.3.1 Yapay sinir ağları

Yapay zekanın bilgisayarlar üzerinde çalışabilme mantığı, insan beyni örnek alınarak sinir sisteminin işleyişi taklit edilmiştir. Yapay sinir ağları, yeni bir örüntü keşfetmek için girdiler ve çıktılar arasında karmaşık bir ilişki gösteren doğrusal olmayan istatistiksel modellerdir. Nesne algılama, dil algılama, ses ve görüntü tanıma gibi farklı fonksiyonlar için yapay sinir ağlarından yararlanır.

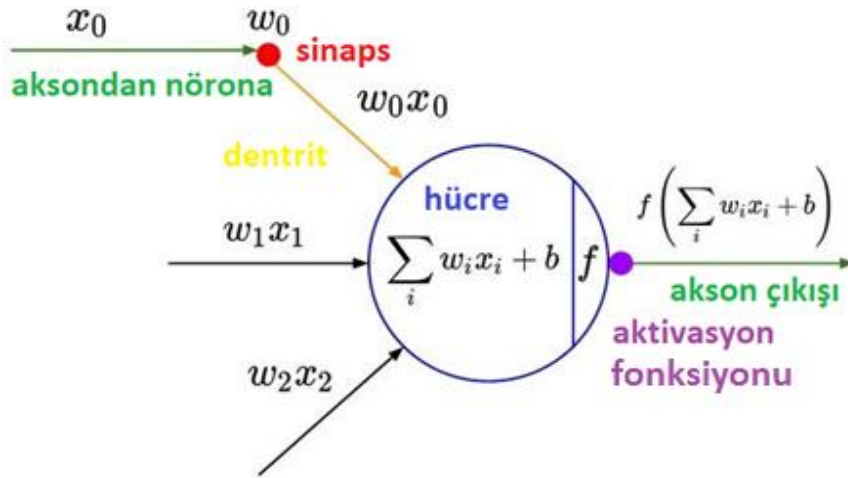
YSA'nın faydalarından biri de örnek veri kümeleri ile öğreniminin gerçekleşebilmesidir. YSA'nın kullanım alanlarının en başında rastgele fonksiyon yaklaşımı gelmektedir. Bu araçlar sayesinde, elverişli maliyetler meydana gelebilir. YSA, çıktılarının elde edilebilmesi için bütün veri kümelerinin yerine örnek verilerin alınması kabiliyetindedir. YSA'ların tahmin yeteneklerinin gelişmiş olması ile verilerin analizlerini geliştirebilir. İnsana ait sinir hücre yapı şeması Şekil 1.3'te gösterilmiştir.



Şekil 1.3 : Sinir hücre yapı şeması [5].

En meşhur makine öğrenme algoritmalarının başında Yapay Sinir Ağları gelmektedir. Yapay Sinir Ağları 1970’li yıllarda icat edilmiştir, son zamanlarda yüksek hesaplama kabiliyeti ile daha da popüler hale gelmiştir ve artık neredeyse her yerde kullanımı olmaktadır [6].

İnsanların sinir sistemi model alınarak oluşturulan ve makinelere, bilgisayarlara uyarlanan yapay sinir hücresi Şekil 1.4’te gösterilmektedir. Diğer sinir hücresinden gelen veriler (x_0) dendrite benzeyen yapı ile (w_0x_0) taşınarak hücre yapısında işleme girer, sonrasında akson benzeri yapı sayesinde başka bir sinir hücresine iletilir. Derin öğrenme modelinde yapay sinir ağları çalışma mantığı bu şekildedir.



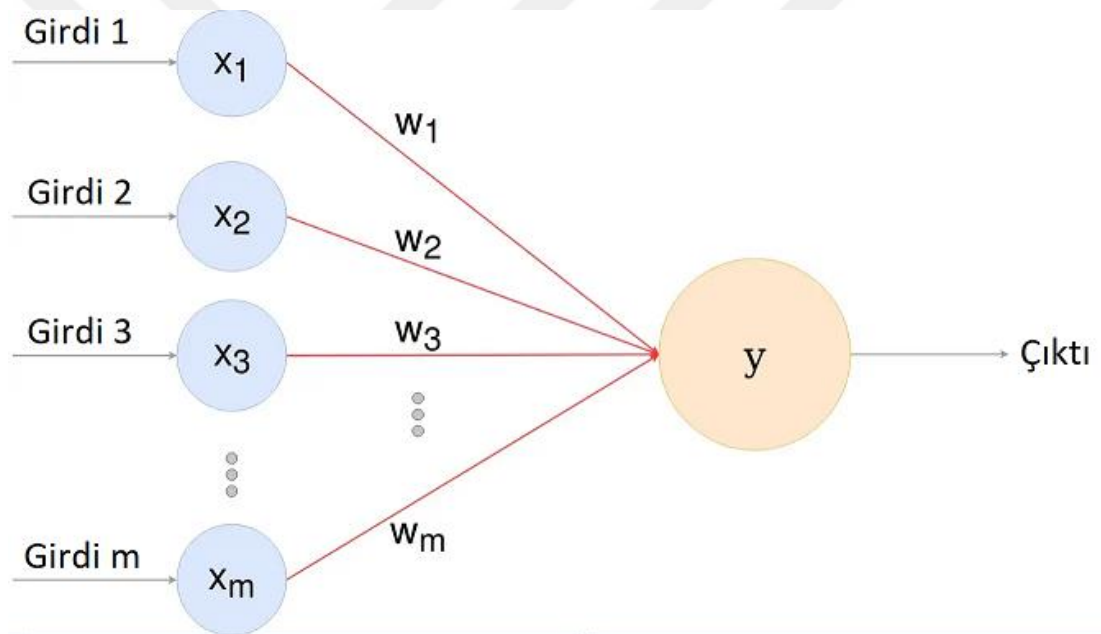
Şekil 1.4 : Yapay sinir hücresi [7].

1.1.3.2 Tek katmanlı algılayıcılar

Tek katmanlı algılayıcıların yapısında girdiler modele gönderilir ve çıktılar elde edilir. Yapay sinir ağı ile yapılan işlemler Denklem 1.1 ile gösterilmiştir. Denklemde x ile gösterilen değerler girdi öznitelikleri, w ile gösterilen değerler ise ağırlık değerlerini göstermektedir. İki değerın çarpım sonucu ile b ile gösterilen yanlılık değeri toplanarak elde edilen sonuç aktivasyon fonksiyonuna tabi tutulur. Çıkan sonuç sıfırdan büyük ise 1, değil ise 0 sonucu üretilir.

$$f(x)=\begin{cases} 1, & w \cdot x + b > 0, \text{ ise} \\ 0, & \text{değilse} \end{cases} \quad (1.1)$$

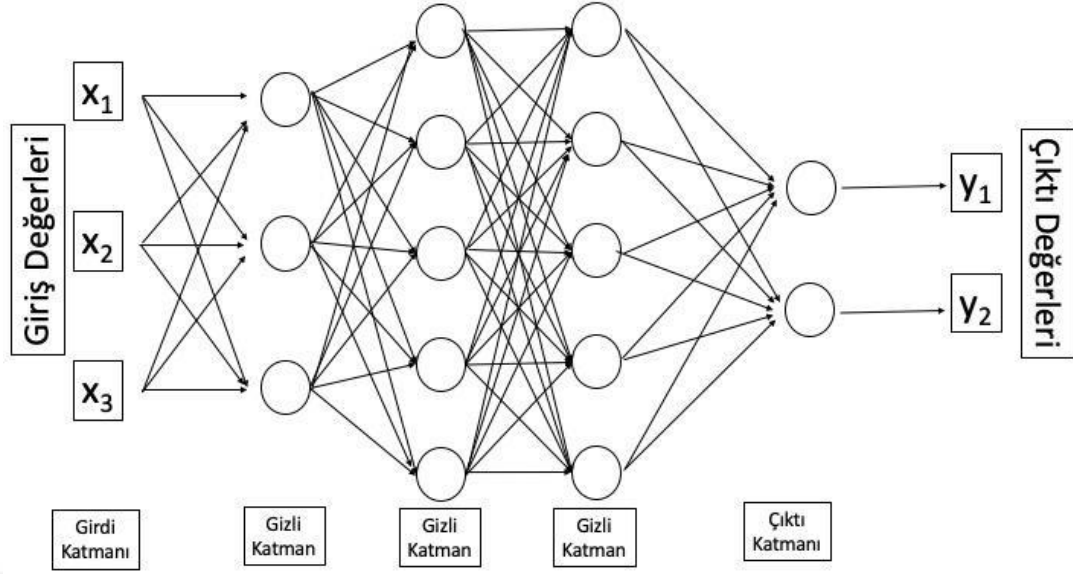
Şekil 1.5'te tek katmanlı algılayıcılara ait modelin şekli gösterilmiştir. Denklem 1.1'de gösterilen formül ile birlikte çıktılar sağlanır.



Şekil 1.5 : Tek katmanlı algılayıcılara ait model [5].

1.1.3.3 Çok katmanlı algılayıcılar

Çok katmanlı sinir ağlar sayesinde daha karmaşık problemler çözülebilmektedir. Girdi katmanında herhangi bir işlem gerçekleşmez. Değerler gizli katmana iletilir. Gizli katmanlarındaki nöronlar, girdi değerleri ile ağırlık değerlerinin çarpılması ile elde edilen sonucu, bir sonraki gizli katmana ileterek girdi sağlanır ve bu işlem mevcut gizli katmanda tekrar edilir. Gizli katmanlardan geçen her bir değer sonucunda çıktılar elde edilir. Şekil 3.6'da çok katmanlı algılayıcılara ait model gösterilmiştir.



Şekil 1.6 : Çok katmanlı algılayıcılara ait model.

1.1.3.4 İleriye yayılım

Fotoğraf üzerindeki nesnelerin algılanması, piksellerin sayısal ifade edilmesi ile sağlanmaktadır. Her bir öznetelik ağırlık değerleri ile ya da rastgele hesaplamaya girer. Gizli katmanlar ile gelen sonuçlar kendisinden sonra gelen katmana ileterek ileri yönde hesap yapılır. Sonuçlar aktivasyon fonksiyonu ile tekrar hesaplamaya girer ve anlamlı hale gelir. Hesaplamalar sonucunda negatif piksel değerleri önemsizdir ve sıfır değeri alır. Böylelikle gizli katmandaki hesapların bir sonraki katmana aktarılmasına ileri yönde yayılım denir.

Denklem 1.2 ile gösterilen işlemler, çok katmanlı yapay sinir ağlarında kullanılmaktadır. $A^{[0]}$ girdi olarak denklemde kullanılır. Z ise çıktı değeridir. Girdi değerleri ağırlık değeri (W) ile çarpılır, yanlılık değeri (b) ile toplanarak çıktı elde edilir. Denklem 1.3'te bir önceki hesaplama ile iletilen $A^{[1]}$ değeri ile birlikte sonraki gizli katmandaki (G) işleme tabi tutulur.

$$\begin{aligned} Z^{[1]} &= W^{[1]} * A^{[0]} + b^{[1]} \\ A^{[1]} &= G^{[1]}(Z^{[1]}) \end{aligned} \quad (1.2)[8]$$

$$\begin{aligned} Z^{[2]} &= W^{[2]} * A^{[1]} + b^{[2]} \\ A^{[2]} &= G^{[2]}(Z^{[2]}) \end{aligned} \quad (1.3)[8]$$

1.1.3.5 Maliyet fonksiyonunun belirlenmesi

İleri yönde yayılım hesaplamaları sonucunda bir çıktı elde edilir. Elde edilen bu tahminin asıl ölçüte olan uzaklığını belirleyerek, doğruluk değerinin hesaplanması gerekmektedir. Modeldeki tahmin parametreleri saptanır. Maliyet fonksiyonu tahmin ile gerçek değeri hesaplayarak sayısal olarak ifade eder [9].

Maliyet fonksiyonu iki farklı yöntem ile hesaplanabilir. Bunlar ortalama mutlak hata ve ortalama kare hatası olarak hesaplanmaktadır. Ortalama mutlak hata mutlak hataların toplamını alır. Ortalama mutlak hata Denklem 1.4 ile gösterilmiştir. Örnek sayısı m ile ifade edilmiştir. \hat{y} tahmin değeridir.

$$\text{Ortalama mutlak hata} = \frac{1}{m} \sum_{i=1}^m |\hat{y}^{(1)} - y^{(1)}| \quad (1.4)$$

Ortalama kare hatası, maliyet fonksiyonunun hesaplanması için farklı bir yöntemdir. Burada doğruluk değeri sonuçlarının gerçek değerden ne kadar farklı olduğunu bir mutlak sayı ile ifade eder. Ortalama kare hatası Denklem 1.5 ile gösterilmiştir.

$$\text{Ortalama kare hatası} = \frac{1}{2m} \sum_{i=1}^m (\hat{y}^{(1)} - y^{(1)})^2 \quad (1.5)$$

1.1.3.6 Geriye yayılım

Geri yönde yayılım yapay sinir ağlarının eğitiminde önemli bir rol alır. Yapay sinir ağlarının önceki dönemleri ile gelen hata oranlarının istenilen doğrultuda iyileşmesi gerekmektedir. Düğümdeki ve her birimdeki kayıpların geri yönde yayılım sürecinde bulunması gerekir. Her kayıp sayısal bir değere sahiptir. Düğümlerdeki kayıpların sebebini bulmak için, ağırlık değerleri daha küçük hale getirilerek modelin optimizasyonu sağlanır [10].

Derin öğrenme yapısında girdi vektörü olarak x kullanılması ile ileri yayımlı sinir ağı oluşturularak, y çıktısının meydana gelmesi için sinir ağına ileri yönde bir akış oluşur. Yapıdaki girdiler ile ilk bilgi elde edilir ve gizli katmanlar ile çıktı sağlanır. İleri yayılım işlemi maliyet fonksiyonu elde edilene dek devam eder. Maliyet fonksiyonu elde edildikten sonra geriye yayılım basamağına geçilir. Değerlerin geriye akışı sağlanır ve eğim bulunur.

Geri yayılım denkleminde ağırlık (w) ile yanlılık (b) birimlerinin hesaplanması için Denklem 1.6’da gösterilen formül kullanılmaktadır. Öğrenim oranı a değeri ile ifade edilir.

$$\begin{aligned} w &= a \cdot \frac{\partial j(w, b)}{\partial w} \\ b &= b - a \cdot \frac{\partial j(w, b)}{\partial w} \end{aligned} \quad (1.6)[8]$$

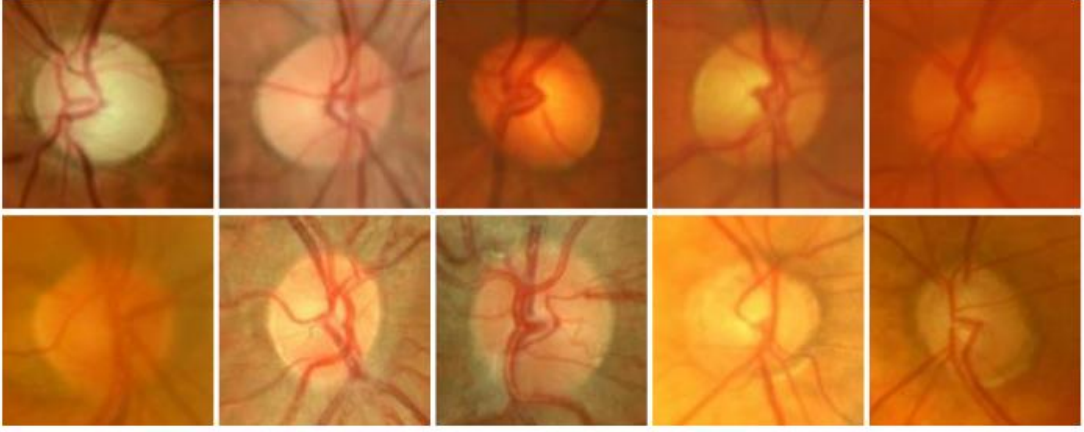
1.1.4 Benzer çalışmalar

Derin öğrenme modellerinin günümüzde popüler olması ile birlikte birçok alanda da kullanılmaya başlanmıştır. Sağlık sektörü, inşaat sektörü, otomasyon araçlar gibi birçok alanda kullanılan yapay zeka sayesinde, nesneler hızlı şekilde tespit edilebilmekte ve yüksek oranda doğru sonuçlar alınmaktadır. Pandemi dönemi ile hayatımızın bir parçası haline gelen maske kullanımı, özellikle kapalı alanlara girişlerde zorunlu kılınmıştı. Bu sebeple de kapalı alanlara girişlerde kişilerin maske takıp takmadığını otomatik olarak algılayarak girişe izin verilmesi için RetinaNet modeli ile yüz maske algılanması modeli oluşturulmuştur. Şekil 1.7’de modelin örnek görüntüsü görülmektedir.



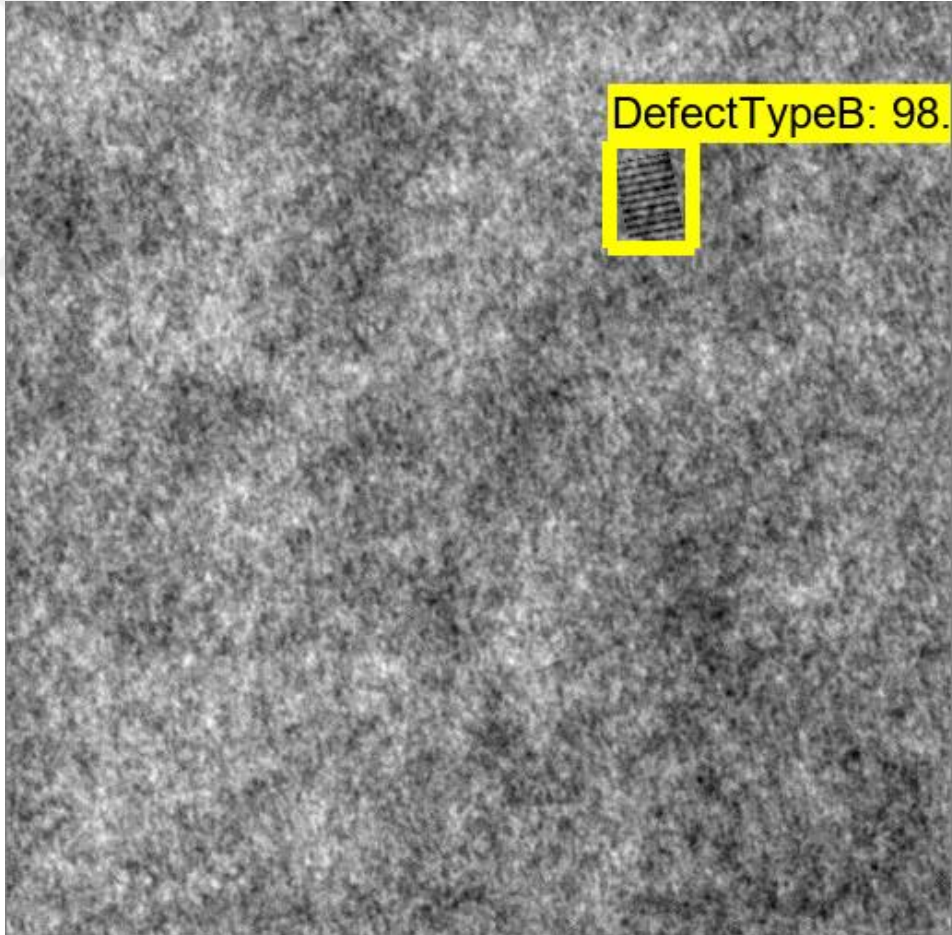
Şekil 1.7 : RetinaNet modeli ile yüz maskesi algılama [11].

Sağlık sektöründe de kullanılan yapay zeka teknolojisinin bir diğer örneği de görme kaybına neden olan Glokom hastalığının tespiti ile ilgilidir. Hastalığın tespiti ve erken teşhisi için EfficientDet modeli kullanılmıştır. Sağlıklı ya da Glokom hastası tespitinde yüksek doğruluk oranı ile (0,9775) başarıya ulaşılmıştır. Şekil 1.8’de modelin örnek görüntüsü görülmektedir.



Şekil 1.8 : EfficientDet modeli ile Glokom hastalığı tespit örneği [12].

Sanayi sektöründe de yapay zekanın kullanıldığı alanlar vardır. Metal levhalar üzerindeki hataların hızlı tespiti için Tensorflow Faster R-CNN modeli kullanılmıştır. Şekil 1.9’da modelin örnek görüntüsü görülmektedir.



Şekil 1.9 : Faster R-CNN modeli ile metal levhalar üzerindeki hata tespiti [13].

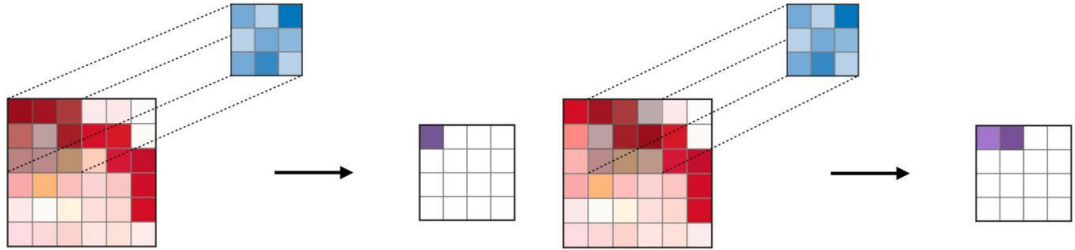
2. EVRİŞİMSEL SİNİR AĞLARI

Evrışimsel sinir ağı, girdi görüntüsündeki nesnelere algılayabilen bir derin öğrenme algoritmasıdır. Temel olarak görüntüleri sınıflandırır, kümelendirir. Nesnelere tanıma için kullanılan derin yapay sinir ağlarıdır.

Yapay sinir ağları ile evrişimel sinir ağları arasında önemli bir fark vardır. Evrişimsel sinir ağları, görüntülerin işlenmesinde örüntü tanıma alanlarında kullanılmaktadır. Evrişimsel sinir ağları evrişim, ortaklama ve tam bağlantı katmanı olmak üzere toplamda üç adet katmandan oluşur [14].

2.1 Evrişim Katmanı

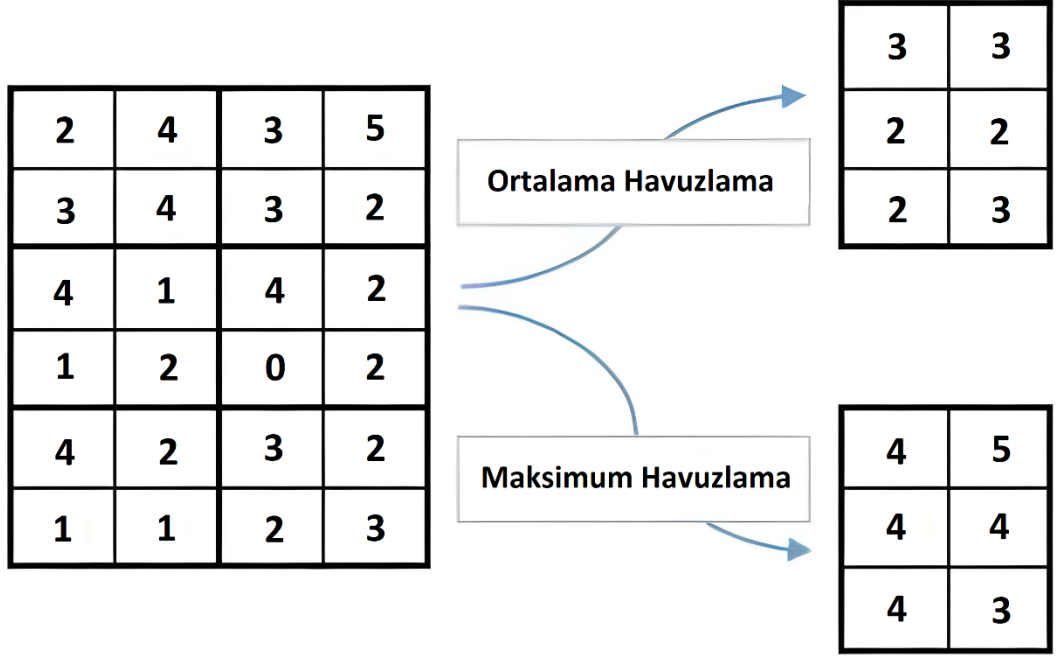
Evrışimsel sinir ağı algoritmalarında ilk görüntü alınan katman evrişim katmanıdır. Görüntü işlenecek her bir görsel aslında piksellerden oluşan birer matrislerdir. Bu matrisler üzerinde daha ufak boyutlardaki filtreler gezerek belirli özellikleri yakalamaya çalışır [15]. Şekil 2.1’de evrişim işlemindeki iki adım gösterilmiştir. Filtre, görüntünün sol üst tarafından başlayıp, adım adım ilerlemektedir.



Şekil 2.1 : Evrişim işlemi [16].

2.2 Ortaklama Katmanı

Ortaklama katmanında da amaç evrişim katmanındaki gibi boyut azaltmaktır. Böylelikle iş gücü azalır ve önemli özelliklere odaklanmaya çalışılır. Evrişim katmanındaki gibi filtreler matris üzerinde gezer. Eğer maksimum ortaklama kullanılıyorsa en büyük değer, ortalama ortaklama kullanılıyorsa değerlerin ortalaması alınır. Böylelikle boyut azalmış olur ve önemli özelliklere odaklanılır. Şekil 2.2’de maksimum ve ortalama havuzlama katmanına ait örnek gösterilmektedir [17].



Şekil 2.2 : Maksimum ve ortalama havuzlama katmanı [18].

2.3 Tam Bağlantı Katmanı

Tam bağlantı katmanının görevi evrişimsel sinir ağını ileri yönde beslemektir. Ortalama katmanındaki çıktılar, bir adım gerideki evrişim ile vektöre dönüşür. Oluşan bu vektör ağırlık ve yanlılık değerleri ile birlikte hesaplamaya girer [19].

2.4 Evrişimsel Sinir Ağı Modelleri

Evrişimsel sinir ağları, görüntü doğrulama becerisine sahiptir. Bu beceri sayesinde görüntüleri tanıması ve tanınan görüntülerin işlenmesi için kullanılan bir yapay sinir ağı türüdür. Evrişimsel sinir ağlarındaki eğitim modellerinde çok fazla etiketlenmiş veriye ihtiyaç duyulmaktadır. Evrişimsel sinir ağları ile yararlı olacak kadar hızlı sonuç üretilecekse, yüksek güçlü işlemcilerle eğitilmelidir.

2.5 Hipotez

Çalışma için R-CNN, RetinaNet ve EfficientDet modelleri kullanılmıştır. Nesne algılama modelinde Tensorflow kütüphanesi kullanılmıştır. Çalışma için otomotiv yan sanayide üretilen araba koltuğu nesne olarak belirlenmiştir. Toplam 787 adet fotoğraf LabelImg programı ile etiketlenerek 9 farklı modele ayrılmıştır. Her bir model

bilgisayara öğretilerek, kamera ve fotoğraf yardımı ile üretim hattı sonunda, ambar ve istifleme alanlarındaki her bir koltuğun modelini otomatik algılayarak etiketlenerek hızlı ayırım yapılması amaçlanmıştır.



3. KOLTUK MODELLERİNİN BELİRLENMESİ

3.1 Kullanılan Yöntemler

Bu çalışmadaki verilerin işlenmesi ve derin öğrenme için Python kullanılmıştır. Derin öğrenme ile dışarıdan gelen verilerin işlenmesini, bilgisayar tarafından öğrenmesini sağlayacak model için Faster RCNN, RetinaNet ve EfficientDet kullanılmıştır. Veri setinde kullanılan koltuk fotoğraflarının işlenip, nesne algılanması için OpenCV kütüphanesi kullanılmıştır.

Modeli desteklemek için kullanılan diğer kütüphaneler ise; tensorflow, numpy, imageio, keras, pandas, matplotlib gibi başlıca python kütüphanelerdir.

3.1.1 Anacoda Python ve OpenCV kütüphanesi

Python, bir web uygulamasının arka ucunda, ön ucunda veya tüm yığnında kullanabileceğiniz çok yönlü bir dildir. Bu uygulamada Python 3.9.16 versiyonu kullanılmıştır. OpenCV (Açık Kaynak Bilgisayarlı Görüntü İşleme Kütüphanesi), bilgisayarlı görüntü işleme ve makine öğrenimi uygulamaları için açık kaynaklı bir kütüphanedir. Bilgisayarlı görüntü işleme alanında en popüler kütüphanelerden biridir ve birçok endüstriyel, akademik ve araştırma projelerinde kullanılmaktadır.

3.1.2 Tensorflow kütüphanesi

Makine öğrenimi karmaşık bir disiplindir ancak makine öğrenimi modellerini uygulamak, Google'ın TensorFlow gibi veri edinme, modelleri eğitme, tahminleri sunma ve gelecekteki sonuçları iyileştirme sürecini kolaylaştıran makine öğrenimi çerçeveleri sayesinde eskisinden çok daha iyi seviyededir.

TensorFlow 2015 yılında Google tarafından geliştirilmiştir ve yaygınlaşmıştır. Herkesin ulaşabileceği açık kaynaklı bir kütüphanedir. TensorFlow kullanılarak birçok makine öğrenmesi ve yapay zeka modelleri geliştirilebilir ve eğitim gerçekleştirilebilir. Tensorflow, Python ile kullanılan bir kütüphanedir. Aynı zamanda farklı bilgisayar dillerini de destekleme özelliğine sahiptir.

TensorFlow, hesaplama yaparken daha genel yapıya sahip bir veri akışı mimarisi kullandığından, sinir ağlarının yaygın bir örneği olduğu çok büyük ölçekli paralel işleme uygulamalarına özellikle açıktır [20].

3.1.3 Numpy kütüphanesi

NumPy (1.23.5 sürümü), Python ile kullanılabilen bir kütüphanedir. Matematiksel işlemlerin rahatlıkla yapılabildiği çok popüler bir araçtır. Numpy sayesinde verimli ve hızlı işlem yapılabilir. Numerical ve Python kelimelerinin ilk hecelerinin birleşmesi ile isimlendirilen bu kütüphane, büyük boyutlardaki matrisler ile işlem yapılmasını sağlar.

3.1.4 Keras kütüphanesi

Keras, derin öğrenme modellerinin eğitilmesinde kullanılan ve Google tarafından geliştirilmiş bir kütüphanedir. Python ile kullanılabilen bu kütüphane sayesinde, basit bir şekilde derin öğrenme modeli oluşturulabilir ve eğitimi gerçekleştirilebilir.

Keras'ın kullanımı ve öğrenilmesi kolaydır. Diğer kütüphanelere nispeten daha yavaştır ama yeni başlayanlar için kullanıcı dostudur. Ayrıca en büyük avantajlarından birisi de, çok fazla kaynağa sahip olması sayesinde karşılaşılan hataların çözümlerine ulaşabilmek çok kolaydır [21].

4. KULLANILAN VERİ SETİ VE YÖNTEMLER

Tezin bu aşamasında veri setlerinin oluşturulabilmesi için koltuk fotoğraflarının toplanması gerekmektedir. Toplam 787 adet koltuk görüntüsü veri setinde kullanılmıştır. %80'i eğitim verisi (632 adet) için, %20'si test verisi (155 adet) için ayrılmıştır. Modele ait resim sayıları Çizelge 4.1'de gösterilmiştir. Her bir koltuk resminin modellerini belirlemek için etiketleme yapılması gerekmektedir. Etiketleme işlemi için LabelImg programı kullanılmıştır. Limited modeli gri renk kumaşlı, Active logolu, Sport kalın çizgi desenli, Map Pocket arka cepli, Trendagr iki parçadan oluşan oturak modelidir ve ayırt edilmesi kolaydır. Tit-X siyah deri, Base siyah kumaş, Carpet ise siyah halı tipi kumaşlı modellerdir ve ayırt edilmesi zordur.

Çizelge 4.1 : Eğitim ve Test sayısı.

Koltuk Modelleri	Eğitim Resim Sayısı	Test Resim Sayısı
Trendagr	71	18
Vinyl	71	18
Sport	70	17
Active	70	17
Limited	70	17
Tit-X	70	17
Base	70	17
Map Pocket	70	17
Carpet	70	17

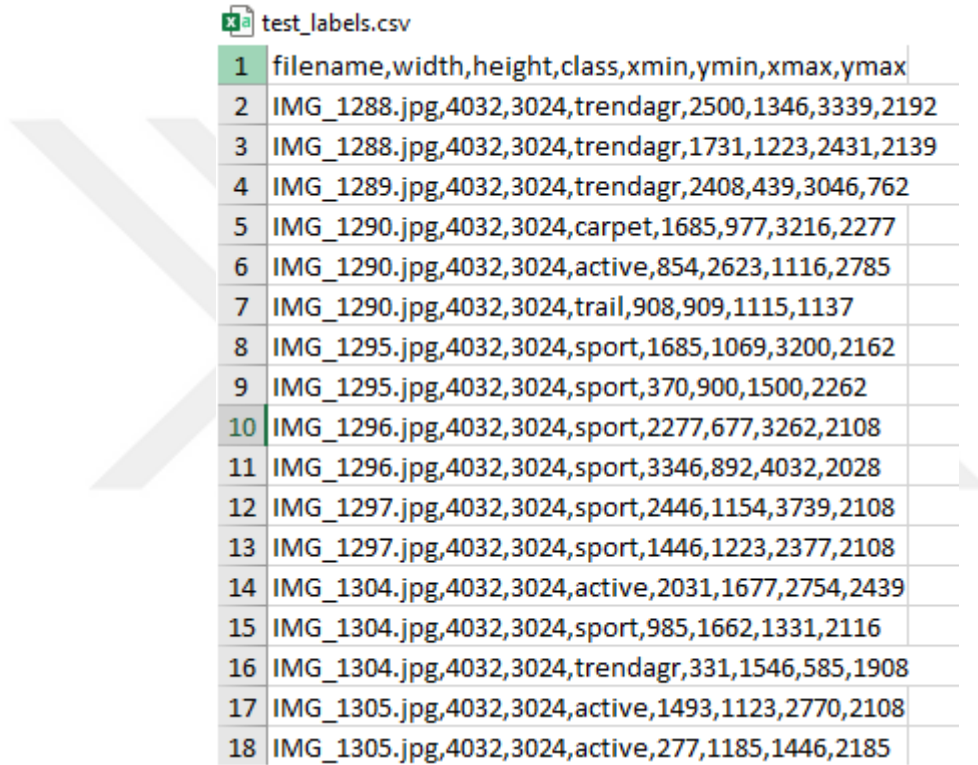
Birçok resim, aynı anda farklı modelleri de içerebilmesi dolayısı ile, eğitilen resim ve test resim sayıları da artmaktadır.



Şekil 4.1 : Koltuk etiketleme örnek görüntüsü.

Koltuğun farklı alanlarda ve açılarda çekilmiş bir örneği Şekil 4.1 ile gösterilmiştir. Farklı açılar, arka plan ve ışık gibi etkenler, eğitim sonundaki başarı oranına olumlu katkı sağlamıştır. Etiketlenen her bir görüntü .xml uzantısı ile kaydedilir ve sayısal bir vektör haline getirilir. Etiketlenen görüntülerde dosya ismi, genişliği, yüksekliği, sınıfı, x ve y koordinatlarına ait çerçeve değerleri bulunmaktadır.

Her etiketlenen görüntüye ait .xml dosyaların tek bir veri halinde toplanması için .csv uzantılı dosya oluşturulmalıdır. Test ve eğitim setleri için iki ayrı .csv oluşturulmalıdır. Şekil 4.2’de test veri setine ait bir .csv uzantılı dosyanın örneği gösterilmiştir.



1	filename,width,height,class,xmin,ymin,xmax,ymax
2	IMG_1288.jpg,4032,3024,trendagr,2500,1346,3339,2192
3	IMG_1288.jpg,4032,3024,trendagr,1731,1223,2431,2139
4	IMG_1289.jpg,4032,3024,trendagr,2408,439,3046,762
5	IMG_1290.jpg,4032,3024,carpet,1685,977,3216,2277
6	IMG_1290.jpg,4032,3024,active,854,2623,1116,2785
7	IMG_1290.jpg,4032,3024,trail,908,909,1115,1137
8	IMG_1295.jpg,4032,3024,sport,1685,1069,3200,2162
9	IMG_1295.jpg,4032,3024,sport,370,900,1500,2262
10	IMG_1296.jpg,4032,3024,sport,2277,677,3262,2108
11	IMG_1296.jpg,4032,3024,sport,3346,892,4032,2028
12	IMG_1297.jpg,4032,3024,sport,2446,1154,3739,2108
13	IMG_1297.jpg,4032,3024,sport,1446,1223,2377,2108
14	IMG_1304.jpg,4032,3024,active,2031,1677,2754,2439
15	IMG_1304.jpg,4032,3024,sport,985,1662,1331,2116
16	IMG_1304.jpg,4032,3024,trendagr,331,1546,585,1908
17	IMG_1305.jpg,4032,3024,active,1493,1123,2770,2108
18	IMG_1305.jpg,4032,3024,active,277,1185,1446,2185

Şekil 4.2 : Test veri setine ait csv örneği.

4.1 Faster RCNN Modelinin Oluşturulması ve Sonuçları

Tensorflow kütüphanesinin son versiyonu (2.11.0) kullanılmıştır ve protos dosyalarının derlenmesi gerekmektedir. Bunun için protobuf kütüphanesinden protoc (3.4.0) kullanılmıştır. Zip içerisindeki protoc.exe dosyası, modelde bulunan research klasörünün içerisine kopyalanmalıdır. Kütüphane içerisindeki her bir dosyanın derlenmesi için protoc.exe çalıştırılmalıdır.

Verilerin eğitimi ilgili kodlar ile gerçekleştirilir. Şekil 4.3'te eğitilen modele ait adımlar gösterilmiştir. Her adımda öğrenme değerleri ve kayıp değerleri adım adım çıkmaktadır.

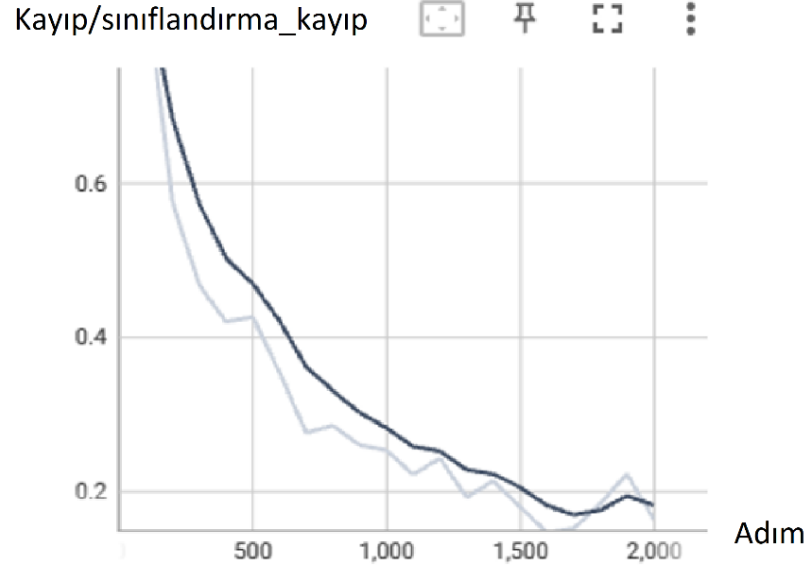
Model eğitimi 2.000 adım tekrar ederek tamamlanmıştır. Eğitim toplam 7 saat 44 dakika sürmüştür.

```
'loss/total_loss': 0.49864027,
'learning_rate': 0.0799474}
I1218 02:18:57.942490 7592 model_lib_v2.py:708] {'loss/classification_loss': 0.18415615,
'loss/localization_loss': 0.15399876,
'loss/regularization_loss': 0.16048536,
'loss/total_loss': 0.49864027,
'learning_rate': 0.0799474}
INFO:tensorflow:Step 1900 per-step time 8.371s
I1218 02:32:54.997494 7592 model_lib_v2.py:705] Step 1900 per-step time 8.371s
INFO:tensorflow:({'loss/classification_loss': 0.22264358,
'loss/localization_loss': 0.16788624,
'loss/regularization_loss': 0.16040282,
'loss/total_loss': 0.55093265,
'learning_rate': 0.07993342}
I1218 02:32:55.007490 7592 model_lib_v2.py:708] {'loss/classification_loss': 0.22264358,
'loss/localization_loss': 0.16788624,
'loss/regularization_loss': 0.16040282,
'loss/total_loss': 0.55093265,
'learning_rate': 0.07993342}
INFO:tensorflow:Step 2000 per-step time 8.284s
I1218 02:46:43.377488 7592 model_lib_v2.py:705] Step 2000 per-step time 8.284s
INFO:tensorflow:({'loss/classification_loss': 0.16380948,
'loss/localization_loss': 0.13284342,
'loss/regularization_loss': 0.16024075,
'loss/total_loss': 0.45689365,
'learning_rate': 0.07991781}
I1218 02:46:43.390489 7592 model_lib_v2.py:708] {'loss/classification_loss': 0.16380948,
'loss/localization_loss': 0.13284342,
'loss/regularization_loss': 0.16024075,
'loss/total_loss': 0.45689365,
```

Şekil 4.3 : Model eğitim adımları.

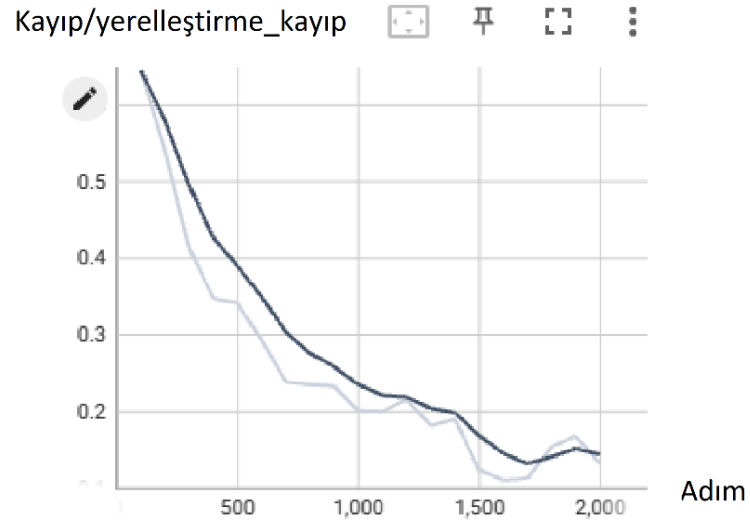
Eğitim modeli sonrası test verisindeki resim çıktısı ile koltuk modeli doğrulanmıştır. Burada amaç, yüksek doğruluk oranı ile doğru koltuk modelini elde edebilmektir.

Modelin eğitim verileri grafikler halinde gösterilmiştir. Grafiklerde, modelin öğrenimi ve hata kayıplarının azaldığı görülmüştür. Şekil 4.4'te modelin sınıflandırma verilerindeki kayıplarının azaldığı görülmüştür.



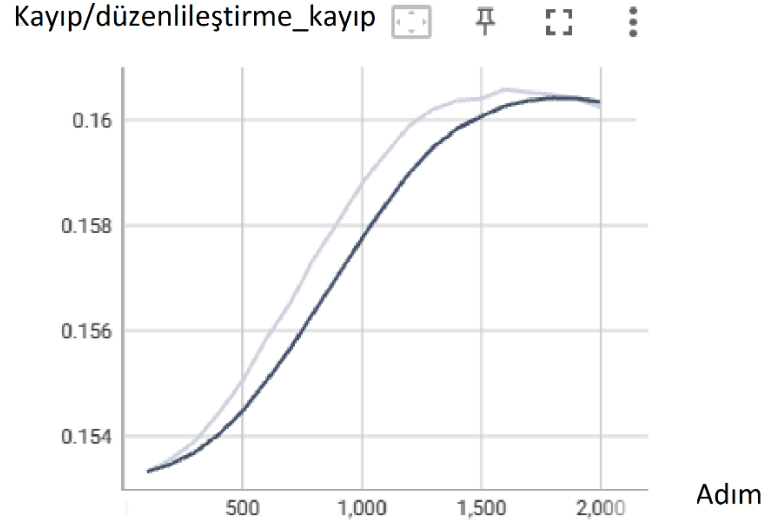
Şekil 4.4 : Sınıflandırma kayıp grafiği.

Şekil 4.5'te yerleştirme kayıplarının azaldığı görülmüştür.



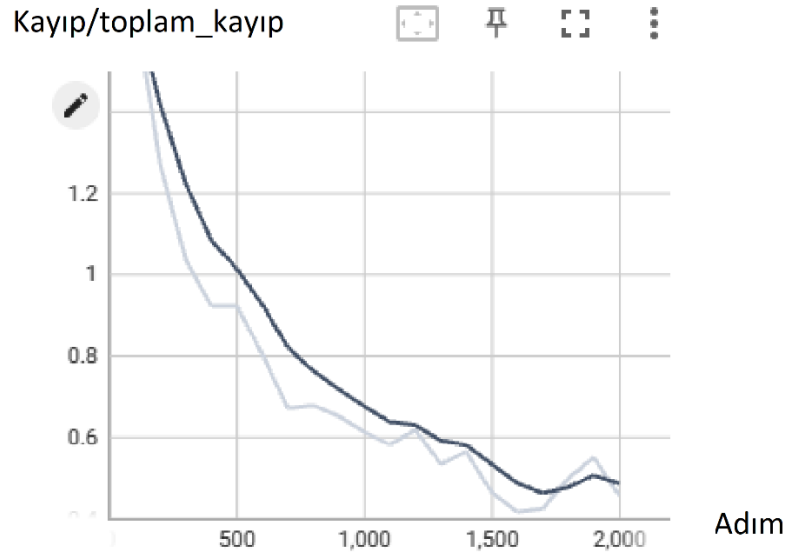
Şekil 4.5 :Yerleştirme kayıp grafiği.

Şekil 4.6'da düzenlileştirmenin arttığı görülmüştür.



Şekil 4.6 : Düzenleştirme artış grafiği.

Şekil 4.7’de toplam kayıpların azaldığı grafikler ile gösterilmiştir.



Şekil 4.7 : Toplam kayıp grafiği.

Özet olarak, model kayıpları en aza indirmeye zorlanmış ve doğru çözümler üretilmeye çalışılmıştır.

Çizelge 4.2’de 0,5 ile 0,95 arasındaki ortalama hassasiyeti görebilirsiniz. Hassasiyet değerlerinin en büyüğü 0,797 olarak ölçülmüştür. Değerler 0,95’e ne kadar yakın ise, modelin test sonuçlarının da o kadar iyi olduğu görülmektedir. Çizelge 4.2’de Modelin başarı oranları gösterilmiştir.

Çizelge 4.2 : Test modelinin başarı oranları.

Ortalama Birim	Hassasiyet	Alan	Maksimum Algı	Algı Değeri
Ortalama Hassasiyet	0.50:0.95	Hepsi	100	0.688
Ortalama Hassasiyet	0.50	Hepsi	100	1.000
Ortalama Hassasiyet	0.75	Hepsi	100	0.813
Ortalama Hassasiyet	0.50:0.95	Küçük	100	-1.000
Ortalama Hassasiyet	0.50:0.95	Orta	100	-1.000
Ortalama Hassasiyet	0.50:0.95	Geniş	100	0.688
Ortalama Hatırlama	0.50:0.95	Hepsi	1	0.797
Ortalama Hatırlama	0.50:0.95	Hepsi	10	0.797
Ortalama Hatırlama	0.50:0.95	Hepsi	100	0.797
Ortalama Hatırlama	0.50:0.95	Küçük	100	-1.000
Ortalama Hatırlama	0.50:0.95	Orta	100	-1.000
Ortalama Hatırlama	0.50:0.95	Geniş	100	0.700

Tensorflow ile yapılan eğitim ile birlikte test edilen resimlerin doğruluk oranları Çizelge 4.3’te gösterilmiştir.

Çizelge 4.3 : Tensorflow veri seti doğruluk oranı tablosu.

Koltuk Modelleri	Eğitim Resim Sayısı	Test Resim Sayısı	Ortalama Doğruluk Oranı
Trendagr	71	18	0,840
Vinyl	71	18	0,771
Sport	70	17	0,865
Active	70	17	0,804
Limited	70	17	0,792
Tit-X	70	17	0,667
Base	70	17	0,758
Map Pocket	70	17	0,889
Carpet	70	17	0,794

4.2 RetinaNet Modelinin Oluşturulması ve Sonuçları

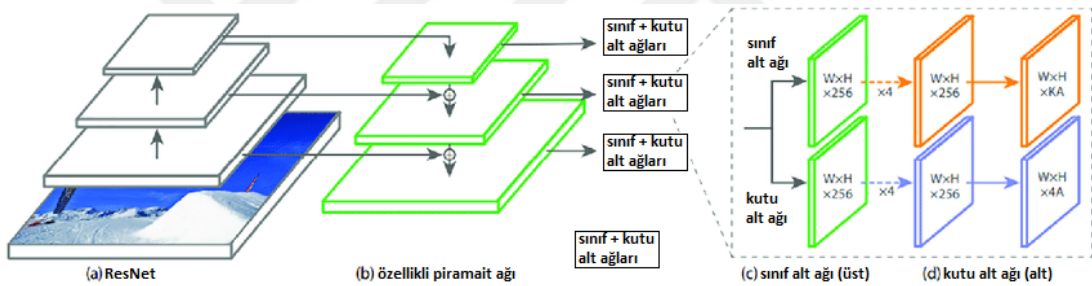
RetinaNet, nesne tanıma alanında kullanılan bir derin öğrenme modelidir. Diğer nesne tanıma modellerinden farklı olarak, çok ölçekli nesne algılama için tasarlanmıştır. Bu özelliği, özellikle küçük nesnelerin tespiti için faydalıdır.

RetinaNet, iki ana bileşenden oluşur: önceden eğitilmiş bir evrişimli sinir ağı (CNN) ve bir nesne algılama başlangıcı (detection head). CNN, girdi görüntülerinde öznitelikleri çıkarmak için kullanılır ve detection head, öznitelikleri alır ve görüntüdeki nesnelere tespit etmek için kullanılır.

RetinaNet'in en önemli özelliği, "focal loss" adı verilen bir kayıp fonksiyonu kullanmasıdır. Bu kayıp fonksiyonu, nadir sınıfların (örneğin küçük nesnelere) sınıflandırmasını iyileştirir ve modelin daha yüksek doğruluk oranları elde etmesine yardımcı olur.

RetinaNet, zorlu nesne algılama görevlerinde diğer modellere göre daha iyi sonuçlar vermiştir. RetinaNet modeli, açık kaynaklı olarak TensorFlow kütüphanesinde bulunmaktadır ve çeşitli uygulamalarda kullanılmaktadır.

Şekil 4.8'de RetinaNet modelinin yapısı gösterilmiştir.



Şekil 4.8 : RetinaNet Modelinin Yapısı [22].

RetinaNet modeli veri setinde toplam 787 adet fotoğraf kullanılmıştır. %80'i eğitim verisi için (632 adet fotoğraf), %20'si test verisi için (155 adet fotoğraf) kullanılmıştır. RetinaNet modeli ile eğitim toplam 6 saat 33 dakika sürmüştür. Çizelge 4.4'te doğruluk oranları gösterilmiştir. Her modele ait doğruluk oranları da ayrıca gösterilmiştir ve ortalama başarı oranı %60,8 olarak çıkmıştır.

Çizelge 4.4 : RetinaNet veri seti doğruluk oranı tablosu.

Koltuk Modelleri	Eğitim Resim Sayısı	Test Resim Sayısı	Ortalama Doğruluk Oranı
Trendagr	71	18	0,587
Vinyl	71	18	0,506
Sport	70	17	0,744
Active	70	17	0,752
Limited	70	17	0,789

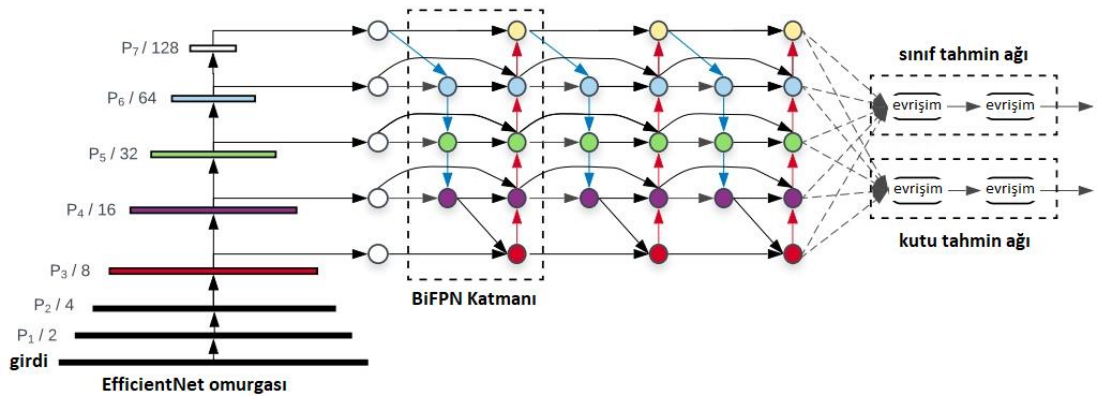
Tit-X	70	17	0,325
Base	70	17	0,599
Map Pocket	70	17	0,632
Carpet	70	17	0,546

4.3 EfficientDet Modelinin Oluşturulması ve Sonuçları

EfficientDet, Google tarafından geliştirilen bir nesne algılama modelidir. Model, önceki nesne algılama modellerinden daha hızlı ve daha iyi performans gösterir.

EfficientDet modeli, EfficientNet adlı bir önceden eğitilmiş modelin özelliklerini kullanarak oluşturulmuştur. Bu özellikler, farklı ölçeklerdeki özellik haritalarının birleştirilmesiyle elde edilir. Daha sonra, özellik haritaları bir çift boyutlu evrişim katmanı ve bir çift boyutlu evrişim katmanı ve daha sonra çıktıları hesaplamak için bir çift boyutlu evrişim katmanı kullanılarak birer birer sıkıştırılır. Model, ölçeklendirilebilir bir yapıya sahip olduğundan, farklı nesne boyutlarını algılamak için farklı özellik haritaları kullanarak çalışabilir.

EfficientDet, diğer nesne algılama modellerinden farklı olarak, daha hızlı ve daha düşük bir hesaplama maliyeti ile daha iyi bir performans gösterir. Bu nedenle, gerçek zamanlı uygulamalar için daha uygundur. Ayrıca, daha önceki modellerin birleştirilmesiyle oluşturulduğundan, daha önceki modellerden daha küçük bir model boyutuna sahiptir. Şekil 4.9'da EfficientDet model yapısı gösterilmiştir.



Şekil 4.9 : EfficientDet modelin yapısı [23].

EfficientDet modeli veri setinde toplam 787 adet fotoğraf veri setinde kullanılmıştır. %80'i eğitim verisi için (632 adet fotoğraf), %20'si test verisi için (155 adet fotoğraf) kullanılmıştır. EfficientDet modeli ile eğitim toplam 5 saat 12 dakika sürmüştür.

Çizelge 4.5'te doğruluk oranları gösterilmiştir. Her modele ait doğruluk oranları da ayrıca gösterilmiştir ve ortalama başarı oranı %50,7 olarak çıkmıştır.

Çizelge 4.5 : EfficientDet veri seti doğruluk oranı tablosu.

Koltuk Modelleri	Eğitim Resim Sayısı	Test Resim Sayısı	Ortalama Doğruluk Oranı
Trendagr	71	18	0,476
Vinyl	71	18	0,412
Sport	70	17	0,612
Active	70	17	0,665
Limited	70	17	0,632
Tit-X	70	17	0,288
Base	70	17	0,544
Map Pocket	70	17	0,484
Carpet	70	17	0,459

4.4 Tüm Modellerin Karşılaştırılması

Toplam 787 adet fotoğraf %80 eğitim setinde, %20 test testinde kullanılmıştır ve üç ayrı model üzerinde çalıştırılmıştır. Tensorflow Faster R-CNN modelinde başarı oranı %79,7, RetinaNet modelinde başarı oranı %60,8 ve EfficientDet modelinde başarı oranı %50,7 çıkmıştır. Çizelge 4.6'da her modele ait başarı oranları tablo halinde gösterilmiştir.

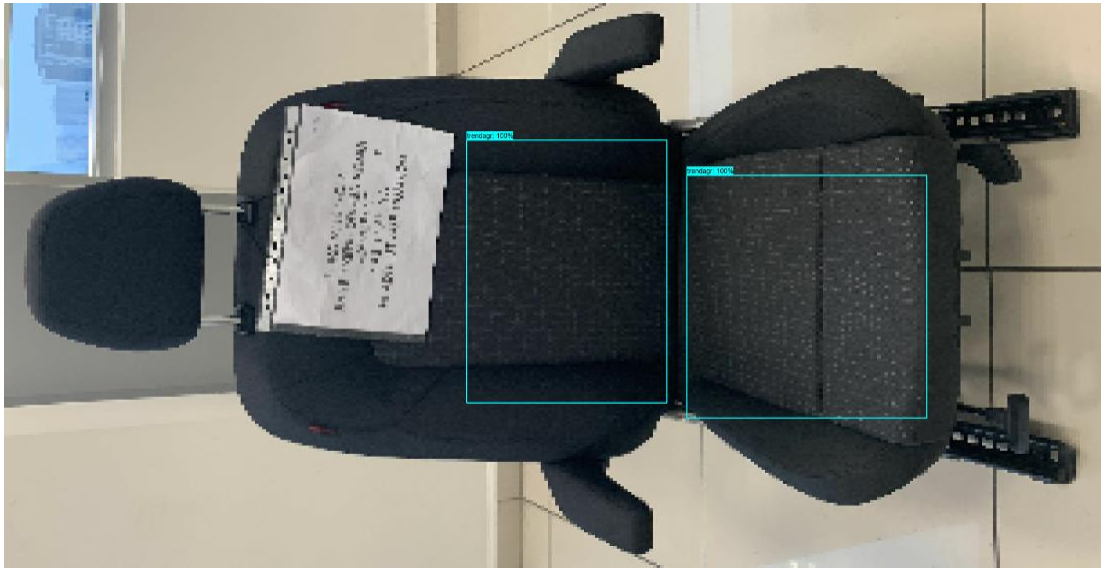
Tensorflow modelinde Sport, Active, Limited ve Trendagr sınıflarında yüksek başarı oranları elde ettiği görülmüştür. Tit-X ve base sınıflarında diğerlerine göre daha düşük başarı oranı elde edildiği görülmüştür. RetinaNet modelinde Limited sınıfı yüksek başarı oranına sahipken, Tit-X sınıfında çok düşük başarı oranı elde edilmiştir. EfficientDet modelinde Active sınıfı en yüksek başarı oranı elde edilirken, diğer modellerdeki gibi Tit-X sınıfında düşük başarı görülmüştür.

Limited sınıfına ait koltuklarda, ayırt edici özellik olarak koltuk kumaşının açık renkte olması, her üç modelde doğruluk oranının yüksek olmasını sağladığı görülmüştür. Active ve Sport sınıfına ait koltuklarda da, kumaş üzerindeki logolar ayırt edici olmuştur ve üç modelde de başarı oranları yüksektir. Tit-X sınıfı koltuklar sadece deri olması ve farklı ışıklı ortamlarda çekilmesi sebebi ile Base sınıfı koltukların renklerine yakın olduğu görülmüştür. Bu sebeple de başarı oranları düşük çıkmıştır. Tensorflow Faster R-CNN modelinin eğitim yöntemi, diğer modeller karşısında yüksek başarı oranı göstermiştir.

Çizelge 4.6 : Modellerin başarı oranları

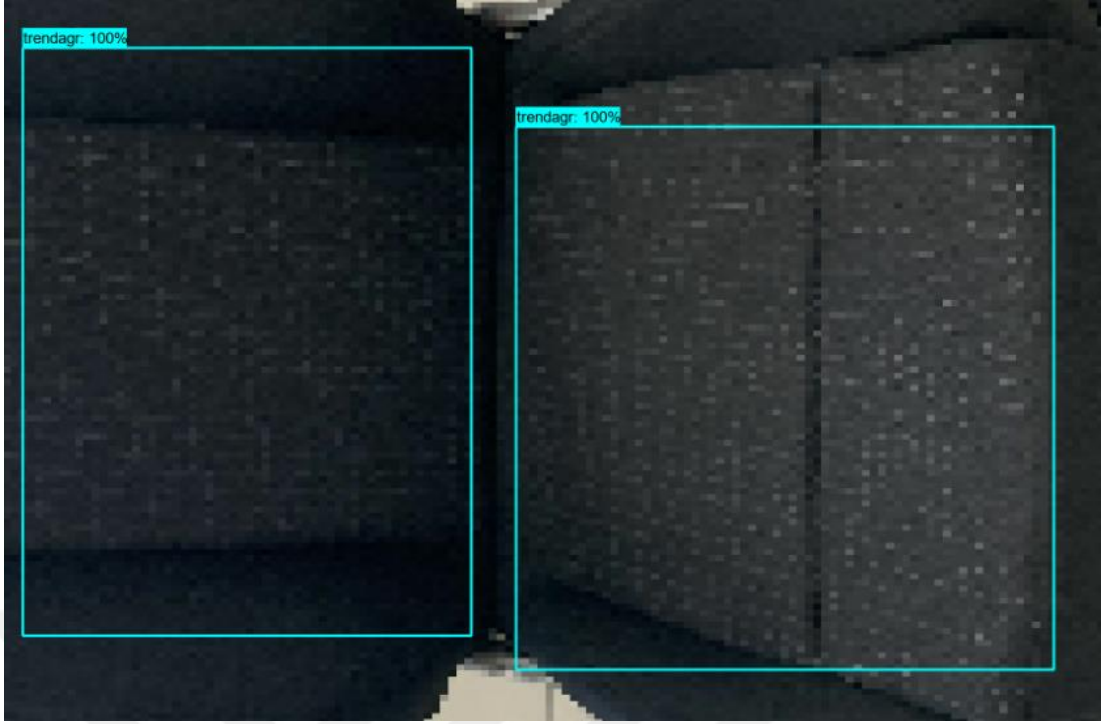
Model	Eğitim Süresi	Doğruluk Oranı
Tensorflow Faster R-CNN	7 saat 44 dakika	79,7
RetinaNet	6 saat 33 dakika	60,8
EfficientDet	5 saat 12 dakika	50,7

Şekil 4.10’da Trend AGR koltuk modelinin genel görüntüsü ile test sonucu gösterilmiştir. Şekil 4.11’de ise, test sonucundaki koltuğun modelini ayırt eden bölgenin işaretlenmiş ve doğruluğuna emin olduğu yüzde değeri görülmektedir.



Şekil 4.10 : Trend AGR koltuğunun genel görünümü.

Şekil 4.12’de Active, Trail ve Carpet koltuk modellerinin tek bir kare içerisindeki genel görüntüsü ile test sonuçları görülmektedir. Genel görüntüde görüldüğü üzere, tek bir görüntü üzerinden aynı anda üç adet şekil otomatik olarak algılanmıştır. Şekil 4.13’te ise test sonuçlarının detayları gösterilmiştir.



Şekil 4.11 : Trend AGR koltuk modelinin test sonucu.



Şekil 4.12 : Active, trail ve carpet modellerinin test sonucu genel görünümü.



Şekil 4.13 : Koltuk modellerinin tespit edilebildiği örnek test sonuçları.

5. SONUÇ VE ÖNERİLER

Günümüzde kontrol edilemeyen çevresel faktörlerin etkisiyle ortaya çıkan değişimler, tarım öncesinden bu yana insanların içinde bulunduğu örgütleri her yeni teknoloji dalgasıyla birlikte değişime ayak uydurmaya zorlamakta ve örgütleri bu şekilde etkilemektedir. Gelişen sanayi, hızla artan popülasyon ve şehirleşme, taleplerin yükselmesi ve arz dengesinin kurulması değişimi beraberinde getirmektedir. Böylelikle yenilikçi bir ortamın oluşmasında büyük rol almaktadır. Dünyada gelişmiş ülkelerdeki birçok kuruluş, rekabetçi fiyat politikalarının bir gereği olarak üretime yönelik birçok operasyonda ve üretim maliyetlerini düşük seviyede tutabilmek için işgücünün ucuz olduğu ülkeleri tercih etmektedir. Hedef kitle konumundaki tüketici, tedarik zincirinin tekilliğinden doğabilecek teslimat riskleri ile marka imajının korunması için tedarik zincirlerini çeşitlendirmeyi tercih etmektedir. Endüstri 4.0 teknolojileri, gelişmiş ülkelerin işgücü maliyetlerini olumlu yönde etkileyerek, kendi ülkelerinde bazı üretim operasyonlarını kısmen de olsa mümkün kılacaktır.

Nesne algılama sayesinde birçok teknolojik gelişmeler elde edilmiştir. Cep telefonlarının kameralarının yüz algılaması, otonom sürüş teknolojisinde kullanılan kameralar ile insan, trafik ışıkları, araç, plaka, şerit gibi birçok nesnenin algılanması sağlanabiliyorken, yine de insanlar kadar başarılı olunmasa da son dönemde çok ilerleme sağlanmıştır. İnsansız araçların ve robotların yaygınlaştığı günümüzde, nesne algılama sistemlerine ihtiyaç çok daha fazla önemli hale geliyor.

Bu uygulamada, otomotiv yan sanayi firmasında üretilen koltukların modellerini otomatik algılayabilen YSA yardımı ile bir model oluşturulmuştur. Bir otomobil markasına ait farklı koltuk modelleri incelenmiştir. Toplam 787 adet koltuk görseli kullanılarak, model eğitilmiştir. Eğitilen model sonrasında otomatik olarak koltuk modellerinin algılanıp sistem tarafından doğru model çıktısı alınması amaçlanmıştır.

Test edilen 155 adet görsel sonucunda, Faster R-CNN modeli ile %79,7, RetinaNet modeli ile %60,8 ve EfficientDet modeli ile %50,7 başarı oranları elde edilmiştir. Koltuk modellerinin tespiti için en doğru modelin Faster R-CNN olduğu görülmüştür. Elde edilen verilerin doğruluk oranlarının artırılması için sisteme daha çok veriler eklenerek, üretim sonrasında ve stok alanındaki doğru model ayırımı yapılabilecek

seviyeye ulařılabilir. Ayrıca, üretim alanındaki model belirleme kabiliyetini arttırmak için, sabit ışıklı ortam sağlanabilir.

5.1 Çalışmanın Uygulama Alanı

Araçlara ait koltuk modellerinin, üretim hattı sonunda model bazında ayırım yapılarak, doğru stoklama ve hızlı ayırım yapılması hedeflenmiştir. Her üretilen koltuk modelleri, stok alanında model bazında ayrılıp, sevkiyatın doğru şekilde yapıldığını doğrulanması beklenmektedir. Tüm bu sonuçlar, YSA alt yapısı sayesinde üretim sonunda ve stok alanında yerleştirilecek olan kameralar ile otomatik nesne algılama yapılacaktır.

Yüksek üretim ve sonucunda ortaya çıkan hatalı üretimlerin de hızlı ve doğru ayıklanması için de nesne algılama yöntemi kullanılarak, yeni giren personellerin de en az tecrübeli personeller gibi koltuk modellerini doğru tespit edebilecek ve doğru alana yönlendirme yapabilecektir.

KAYNAKLAR

- [1] Nilsson, N. J. (2010). Yapay Zeka Geçmişi ve Geleceği, *Boğaziçi Üniversitesi Yayinevi*
- [2] Delipetrev, B. , Tsinaraki, C. , Kostic, U. (2020). AI Watch Historical Evolution of Artificial Intelligence, *European Commission* ,Erişim tarihi 17.11.2022 , doi:10.2760/801580
- [3] Url-1 <<https://onlinedegrees.sandiego.edu/machine-learning-engineer-career/>>, Erişim tarihi 17.11.2022.
- [4] Yılmaz, A. (2018). Bilgisayarlı Öğrenme ve Makine Öğrenmesi, *Kodlab Yayın*
- [5] Arc. (2018). The Perceptron, Erişim: 12 Aralık 2022, <https://towardsdatascience.com/the-perceptron-3af34c84838c>
- [6] Tüminçin, F. , Aytekin, A. , Ayaz, A. (2019). İnsan Beyni Modellemesi ve Yapay Sinir Ağları
- [7] Kızrak, A. (2018). ŞU KARA KUTUYU AÇALIM: Yapay Sinir Ağları, Erişim: 12 Aralık 2022, <https://ayyucekizrak.medium.com/%C5%9Fu-kara-kutuyu-a%C3%A7alim-yapay-sinir-a%C4%9Flar%C4%B1-7b65c6a5264a>
- [8] Murat, S. (2021). İNSANSIZ HAVA ARACI GÖRÜNTÜLERİNDEN DERİN ÖĞRENME YÖNTEMLERİYLE NESNE TANIMA, *Maltepe Üniversitesi*, Erişim: 12 Aralık 2022, <https://openaccess.maltepe.edu.tr/xmlui/bitstream/handle/20.500.12415/7379/10382825.pdf?sequence=1&isAllowed=y>
- [9] Url-2 <<https://akillifabrikalar.com.tr/derin-ogrenme-nedir-nasil-calisir-2/>>, erişim tarihi 17.11.2022.
- [10] Krzyk, K. (2018). Coding Deep Learning for Beginners, Linear Regression Cost Function, Erişim: 13 Aralık 2022 , <https://towardsdatascience.com/coding-deep-learning-for-beginners-linear-regression-part-2-cost-function-49545303d29f>
- [11] Url-5 <<https://www.analyticsvidhya.com/blog/2020/08/how-to-build-a-face-mask-detector-using-retinanet-model/>>, Erişim: 30 Aralık 2022.
- [12] Toptaş, B. , Hanbay, D. (2022). The Separation of glaucoma and non-glaucoma fundus images using EfficientNet-B0, *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi* Erişim: 03 Ocak 2023, <https://dergipark.org.tr/en/download/article-file/2647487>

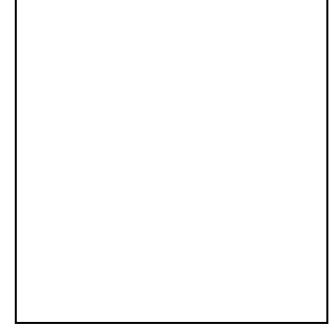
- [13] **Bektaş, M.** (2020). Tensorflow Ortamında Nesne Tanıma Uygulaması (Defect Type Detection using Faster R-CNN), Erişim: 07 Ocak 2023, <https://medium.com/@mbektas/tensorflow-ortam%C4%B1nda-nesne-tan%C4%B1ma-uygulamas%C4%B1-defect-type-detection-using-faster-r-cnn-1be1e1f59c8c>
- [14] **O'Shea, K. , Nash, R.** (2015). An Introduction to Convolutional Neural Networks, Erişim: 30 Kasım 2022 , <https://arxiv.org/abs/1511.08458>
- [15] **Amidi, A. , Amidi, S.** (2019). Derin Öğrenme, *Stanford University*
- [16] **Amidi, A. , Amidi, S.** (2018). VIP Cheatsheet: Convolutional Neural Networks, *Stanford Uni.*, Erişim: 12 Aralık 2022, <https://github.com/afshinea/stanford-cs-230-deep-learning/blob/master/en/cheatsheet-convolutional-neural-networks.pdf>
- [17] **Stanford University,** (2020). Convolutional Neural Networks for Visual Recognition, *Stanford University*, Erişim: 10 Aralık 2022 <https://cs231n.github.io/convolutional-networks/#pool>
- [18] **Savaş, S.** (2020). Derin Öğrenme Algoritmaları, Erişim: 12 Aralık 2022, <https://medium.com/yapay-zeka-makine-%C3%B6%C4%9Frenmesi-derin-%C3%B6%C4%9Frenme/derin-%C3%B6%C4%9Frenme-algoritmalar%C4%B1-8dea99d9b41f>
- [19] **Arunava,** (2018). Convolutional Neural Network, Erişim: 12 Aralık 2022 <https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05>
- [20] **Url-3** <<http://www.tensorflow.org/>>, erişim tarihi 17.05.2022.
- [21] **Url-4** <<https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras>>, Erişim tarihi 10.05.2032.
- [22] **Goyal, R. , Girshik, K.** (2017). Focus Loss for Dense Object Detection, *Facebook AI Research (FAIR)*
- [23] **Yu, A. , Tan, M.** (2020). EfficientDet : Towards Scalable and Efficient Object Detection, Erişim: 17 Aralık 2022, <https://ai.googleblog.com/2020/04/efficientdet-towards-scalable-and.html>

ÖZGEÇMİŞ

Ad-Soyad : Ali İhsan BADEM

Doğum Tarihi ve Yeri :

E-posta :



ÖĞRENİM DURUMU:

- **Lisans** : 2010, Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

- Marturfompak / Program Manager (2021 – Devam ediyor)
- İnova Otomotiv / Project Coordinator (2017-2021)
- Aunde Teknik / Project Leader (2015-2017)
- Alta Metal / Project Engineer (2011-2015)