



Derin Öğrenme ile Uydu Fotoğraflarından Askeri Uçak Tiplerinin Tespiti

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Ozan ÖZÇELİK

ORCID 0009-0008-8058-8444

Proje Danışmanı: Dr. Öğretim Üyesi Osman GÖKALP

Ocak 2024

Derin Öğrenme ile Uydu Fotoğraflarından Askeri Uçak Tiplerinin Tespiti

ÖZ

Günümüzde uydu teknolojisinin gelişmesi, askeri uçakların tespit ve takibini savunma ve güvenlik stratejilerinin temel unsurlarından biri haline getirmiştir. Uydu görüntüleri geniş coğrafi alanları kapsadığı ve olayları gözlemlememize olanak sağladığı için bu bağlamda önemli bir rol oynamaktadır. Askeri uçak tespiti, geleneksel yöntemlere yönelik çeşitli zorluklar doğurmaktadır. Bu zorluklar arasında askeri uçaklarla diğer nesnelere ayırt etme zorluğu, gürültülü ve bulanık uydu görüntüleri ve hızlı hareket eden uçaklar yer alıyor.

Derin öğrenme, büyük miktarda veriyi analiz ederek karmaşık kalıpları keşfetme yeteneğiyle bilinir. Bu özelliği sayesinde derin öğrenme algoritmaları, geleneksel askeri uçak tespit yöntemlerine göre daha iyi sonuçlar elde edebiliyor. Bu çalışma, uydu görüntülerini kullanan derin öğrenme algoritmalarını kullanarak askeri uçak tipi tanımadaki son gelişmeleri incelemektedir.

Derin öğrenme modelleri, askeri uçak türlerinin uydu görüntülerinden tanınmasında geleneksel yöntemlere göre daha başarılıdır. Derin öğrenme modeli, farklı askeri uçak türlerini ayırt etmek için yüksek doğruluk ve doğruluk sağlar. Derin öğrenme modelleri, gürültülü ve bulanık uydu görüntüleri gibi zorluklara karşı daha toleranslı olabilir. Bu sonuçlar, uydu teknolojisi ve derin öğrenmenin entegrasyonunun askeri uçak tespitinde büyük potansiyele sahip olduğunu göstermektedir.

Bu teknolojinin geliştirilmesi, hava savunma sistemlerini daha güçlü ve hassas hale getirerek savunma stratejilerine yeni bir boyut kazandırabilir. Derin öğrenme algoritmaları askeri uçak tespiti için geleneksel yöntemlere göre daha yüksek

hassasiyet ve doğruluk sağlayabilir. Bu, hava savunma sistemlerinin daha etkin kullanılmasına olanak sağlayabilir. Farklı uydu görüntüleri ve farklı askeri uçak türleri için özelleştirilebilir. Bu durum, farklı koşullar altında daha verimli kullanılmasını sağlar. Derin öğrenme modellerinin geliştirilmesi ve kullanılması geleneksel yöntemlere göre daha ucuzdur. Bu sayede daha geniş bir alanda kullanıma olanak sağlanabilir. Sonuç olarak derin öğrenme algoritmaları askeri uçakları tespit etme konusunda büyük potansiyele sahiptir. Bu teknolojinin geliştirilmesi, hava savunma sistemlerini daha güçlü ve hassas hale getirerek savunma stratejilerine yeni bir boyut kazandırabilir.

Anahtar Sözcükler: Derin öğrenme, vggnet, resnet, densenet, mobilenet, inception, xception, yapay zeka, uydu görüntüleri, savunma sanayi, nesne tespiti, değerlendirme ölçütleri

Detecting Military Aircraft Types from Satellite Photos with Deep Learning

Abstract

Today, the development of satellite technology has made the detection and tracking of military aircraft a key element of defense and security strategies. Satellite imagery plays an important role in this context as it covers large geographical areas and allows us to observe events. Military aircraft detection poses several challenges to traditional methods. These challenges include the difficulty of distinguishing between military aircraft and other objects, noisy and blurry satellite imagery, and fast-moving aircraft.

Deep learning is known for its ability to discover complex patterns by analyzing large amounts of data. Thanks to this feature, deep learning algorithms can achieve better results than traditional military aircraft detection methods. This study examines recent advances in military aircraft type recognition using deep learning algorithms using satellite imagery.

Deep learning models are more successful than traditional methods in recognizing military aircraft types from satellite imagery. The deep learning model provides high accuracy and precision to distinguish different types of military aircraft. Deep learning models can be more tolerant to challenges such as noisy and blurred satellite images. These results show that the integration of satellite technology and deep learning has great potential in military aircraft detection.

The development of this technology can bring a new dimension to defense strategies by making air defense systems more powerful and precise. Deep learning algorithms can provide higher precision and accuracy than traditional methods for military aircraft detection. This could enable more effective use of air defense systems. It can be customized for different satellite imagery and different types of military aircraft. This allows it to be used more efficiently under different conditions. Deep learning models

are cheaper to develop and use than traditional methods. In this way, it can be used in a wider area. In conclusion, deep learning algorithms have great potential in detecting military aircraft. The development of this technology can bring a new dimension to defense strategies by making air defense systems more powerful and sensitive.

Keywords: Deep learning, vggnet, resnet, densenet, mobilenet, inception, xception, artificial intelligence, satellite imagery, defense industry, object detection, evaluation criteria

Bu alıřmayı bilime gnl vermiř tm insanlara ithaf ediyorum.

Teşekkür

İzmir Kâtip Çelebi Üniversitesi 2023-2024 akademik dönemi Yazılım Mühendisliği Yüksek Lisans Bölümü öğrencisi olarak bu süre zarfı içerisinde bilgilerini ve deneyimlerini paylaşan tüm hocalarıma, beraber keyifle eğitim aldığım tüm arkadaşlarıma ve aileme teşekkürlerimi sunarım.

Dönem bitirme projemde desteklerini esirgemeyen Dr. Öğretim Üyesi Osman Gökalg hocama da ayrıca teşekkür ederim.

İçindekiler

Öz	i
Abstract	iii
Teşekkür	vi
Şekiller Listesi.....	x
Tablolar Listesi.....	xii
Kısaltmalar Listesi	xiv
Semboller Listesi.....	xv
1 Giriş	1
2 Literatür Taraması ve Metodoloji	3
2.1 Derin Öğrenme Nedir	3
2.2 Derin Öğrenmenin Tarihi.....	6
2.3 Derin Öğrenmenin Geleceği	7
2.4 Derin Öğrenme Katmanları.....	8
2.5 Derin Öğrenme Modellerinin Kullanıldığı Alanlar	10
2.6 Derin Öğrenme Modellerinin Avantajları.....	10
2.7 Derin Öğrenme Modellerinin Dezavantajları	10
2.8 Derin Öğrenme Mimarileri	11
2.8.1 Yapay Sinir Ağları.....	12
2.8.2 Evrişimsel Sinir Ağları (CNN'ler)	14
2.8.3 Tekrarlayan Sinir Ağları (RNN'ler)	16
2.8.4 Üretken Çekişmeli Ağlar (GAN'lar)	18
2.8.5 Derin Sinir Ağları (DNN'ler)	19

2.8.6	Doğrusal Olmayan Bağlamlı Mimariler (Transformers).....	22
2.8.7	Varyasyonel Otoenkoder (Variational Autoencoder-VAE)	24
2.8.8	Transfer Öğrenme Mimarileri	26
2.9	Derin Öğrenme Modelleri.....	27
2.9.1	VggNet.....	27
2.9.2	ResNet	29
2.9.3	Inception	31
2.9.4	Xception.....	33
2.9.5	MobileNet.....	35
2.9.6	DenseNet	37
2.10	Derin Öğrenme ile Savunma Sanayi Arasındaki İlişki	38
2.11	Derin Öğrenme Modelleri ile Uydu Görüntülerinden Uçak Tiplerinin Tespiti için Yapılan Çalışmalar.....	41
2.12	Derin Öğrenme Modelleri ile Uydu Görüntülerinden Uçak Tiplerinin Tespiti için Atılacak Adımlar.....	43
2.13	Değerlendirme Ölçütleri	44
2.13.1	Accuracy (Doğruluk).....	44
2.13.2	Precision (Kesinlik)	45
2.13.3	Recall (Geri Çağırma)	46
2.13.4	F1-Score.....	48
2.13.5	Validation Accuracy (Doğrulama Doğruluğu).....	49
2.13.6	Epoch (Tur Sayısı).....	50
2.13.7	Overfitting (Aşırı Uyum).....	50
3	Bulgular	52
3.1	Veri Seti	52
3.2	Vgg16 Modeline göre Veri Setinin Değerlendirilmesi	53
3.2.1	Epoch 5'e göre.....	53
3.2.2	Epoch 8'e göre.....	57

3.2	ResNet50 Modeline göre Veri Setinin Değerlendirilmesi.....	60
3.2.1	Epoch 5'e göre.....	60
3.2.2	Epoch 8'e göre.....	62
3.2	MobileNetV2 Modeline göre Veri Setinin Değerlendirilmesi.....	65
3.2.1	Epoch 5'e göre.....	65
3.2.2	Epoch 8'e göre.....	68
3.2	DenseNet121 Modeline göre Veri Setinin Değerlendirilmesi.....	70
3.2.1	Epoch 5'e göre.....	70
3.2.2	Epoch 8'e göre.....	73
3.2	InceptionV3 Modeline göre Veri Setinin Değerlendirilmesi.....	76
3.2.1	Epoch 5'e göre.....	76
3.2.2	Epoch 8'e göre.....	79
3.2	Xception Modeline göre Veri Setinin Değerlendirilmesi.....	83
3.2.1	Epoch 5'e göre.....	83
3.2.2	Epoch 8'e göre.....	86
4	Sonuçlar.....	90
	Kaynaklar.....	94

Şekiller Listesi

Şekil 2.1	Derin Öğrenme	4
Şekil 2.2	Yapay zeka alt grupları	5
Şekil 2.3	Derin öğrenme katmanları	8
Şekil 2.4	Yapay sinir ağları	13
Şekil 2.5	Evrişimsel sinir ağları yapısı	14
Şekil 2.6	Tekrarlayan sinir ağları yapısı	16
Şekil 2.7	Üretken çekişmeli ağ yapısı	18
Şekil 2.8	Derin sinir ağları-LSTM.....	20
Şekil 2.9	Transformers yapısı.....	23
Şekil 2.10	Varyasyonel otoenkoder yapısı	25
Şekil 2.11	Transfer öğrenme yapısı.....	26
Şekil 2.12	VggNet yapısı.....	28
Şekil 2.13	ResNet yapısı.....	29
Şekil 2.14	Inception yapısı	31
Şekil 2.15	Xception yapısı.....	33
Şekil 2.16	MobileNet yapısı	35
Şekil 2.17	DenseNet yapısı.....	37
Şekil 2.18	Derin öğrenme ile nesne tespiti.....	39
Şekil 2.19	Uçak tespiti.....	41
Şekil 3.1	Vgg16 modeli epoch 5'e göre doğruluk grafiği	55
Şekil 3.2	Tahmin edilen uçak	56
Şekil 3.3	Vgg16 modeli epoch 8'e göre doğruluk grafiği	58
Şekil 3.4	ResNet50 modeli epoch 5'e göre doğruluk grafiği	61
Şekil 3.5	ResNet50 modeli epoch 8'e göre doğruluk grafiği	64
Şekil 3.6	MobileNetV2 modeli epoch 5'e göre doğruluk grafiği.....	67
Şekil 3.7	MobileNetV2 modeli epoch 8'e göre doğruluk grafiği.....	69
Şekil 3.8	DenseNet121 modeli epoch 5'e göre doğruluk grafiği	72

Şekil 3.9	DenseNet121 modeli epoch 8'e göre doğruluk grafiği	74
Şekil 3.10	InceptionV3 modeli epoch 5'e göre doğruluk grafiği.....	77
Şekil 3.11	InceptionV3 modeli epoch 8'e göre doğruluk grafiği.....	81
Şekil 3.12	Xception modeli epoch 5'e göre doğruluk grafiği	84
Şekil 3.13	Xception modeli epoch 8'e göre doğruluk grafiği	88

Tablolar Listesi

Tablo 3.1	Vgg16 modeli epoch 5'e göre değerlendirme ölçütleri.....	53
Tablo 3.2	Vgg16 modeli epoch 5'e göre confusion matrix.....	53
Tablo 3.3	Vgg16 modeli epoch 5'e göre düzenlenmiş confusion matrix.....	53
Tablo 3.4	Vgg16 epoch 5' göre tahmin ve doğruluk oranı	56
Tablo 3.5	Vgg16 modeli epoch 8'e göre değerlendirme ölçütleri.....	57
Tablo 3.6	Vgg16 modeli epoch 8'e göre confusion matrix.....	57
Tablo 3.7	Vgg16 modeli epoch 8'e göre düzenlenmiş confusion matrix.....	57
Tablo 3.8	Vgg16 epoch 8'e göre tahmin ve doğruluk oranı.....	59
Tablo 3.9	ResNet50 modeli epoch 5'e göre değerlendirme ölçütleri.....	60
Tablo 3.10	ResNet50 modeli epoch 5'e göre confusion matrix	60
Tablo 3.11	ResNet50 modeli epoch 5'e göre düzenlenmiş confusion matrix.....	60
Tablo 3.12	ResNet50 epoch 5' göre tahmin ve doğruluk oranı.....	62
Tablo 3.13	ResNet50 modeli epoch 8'e göre değerlendirme ölçütleri.....	62
Tablo 3.14	ResNet50 modeli epoch 8'e göre confusion matrix	63
Tablo 3.15	ResNet50 modeli epoch 8'e göre düzenlenmiş confusion matrix.....	63
Tablo 3.16	ResNet50 epoch 8' göre tahmin ve doğruluk oranı.....	65
Tablo 3.17	MobileNetV2 modeli epoch 5'e göre değerlendirme ölçütleri	65
Tablo 3.18	MobileNetV2 modeli epoch 5'e göre confusion matrix.....	66
Tablo 3.19	MobileNetV2 modeli epoch 5'e göre düzenlenmiş confusion matrix	66
Tablo 3.20	MobileNetV2 epoch 5' göre tahmin ve doğruluk oranı	67
Tablo 3.21	MobileNetV2 modeli epoch 8'e göre değerlendirme ölçütleri	68
Tablo 3.22	MobileNetV2 modeli epoch 8'e göre confusion matrix.....	68
Tablo 3.23	MobileNetV2 modeli epoch 8'e göre düzenlenmiş confusion matrix	68
Tablo 3.24	MobileNetV2 epoch 8' göre tahmin ve doğruluk oranı	70
Tablo 3.25	DenseNet121 modeli epoch 5'e göre değerlendirme ölçütleri.....	70
Tablo 3.26	DenseNet121 modeli epoch 5'e göre confusion matrix	71
Tablo 3.27	DenseNet121 modeli epoch 5'e göre düzenlenmiş confusion matrix.....	71

Tablo 3.28 DenseNet121 epoch 5' göre tahmin ve doğruluk oranı.....	72
Tablo 3.29 DenseNet121 modeli epoch 8'e göre değerlendirme ölçütleri.....	73
Tablo 3.30 DenseNet121 modeli epoch 8'e göre confusion matrix	73
Tablo 3.31 DenseNet121 modeli epoch 8'e göre düzenlenmiş confusion matrix.....	73
Tablo 3.32 DenseNet121 epoch 8' göre tahmin ve doğruluk oranı.....	75
Tablo 3.33 InceptionV3 modeli epoch 5'e göre değerlendirme ölçütleri.....	76
Tablo 3.34 InceptionV3 modeli epoch 5'e göre confusion matrix.....	76
Tablo 3.35 InceptionV3 modeli epoch 5'e göre düzenlenmiş confusion matrix.....	76
Tablo 3.36 InceptionV3 epoch 5'e göre tahmin ve doğruluk oranı.....	79
Tablo 3.37 InceptionV3 modeli epoch 8'e göre değerlendirme ölçütleri.....	79
Tablo 3.38 InceptionV3 modeli epoch 8'e göre confusion matrix.....	79
Tablo 3.39 InceptionV3 modeli epoch 8'e göre düzenlenmiş confusion matrix.....	80
Tablo 3.40 InceptionV3 epoch 8'e göre tahmin ve doğruluk oranı.....	82
Tablo 3.41 Xception modeli epoch 5'e göre değerlendirme ölçütleri.....	83
Tablo 3.42 Xception modeli epoch 5'e göre confusion matrix	83
Tablo 3.43 Xception modeli epoch 5'e göre düzenlenmiş confusion matrix.....	83
Tablo 3.44 Xception epoch 5'e göre tahmin ve doğruluk oranı	85
Tablo 3.45 Xception modeli epoch 8'e göre değerlendirme ölçütleri.....	86
Tablo 3.46 Xception modeli epoch 8'e göre confusion matrix	86
Tablo 3.47 Xception modeli epoch 8'e göre düzenlenmiş confusion matrix.....	86
Tablo 3.48 Xception epoch 8'e göre tahmin ve doğruluk oranı	89
Tablo 4.1 Değerlendirme sonuçları (epoch 5).....	90
Tablo 4.2 Değerlendirme sonuçları (epoch 8).....	91

Kısaltmalar Listesi

AI	Yapay Zeka
BERT	Transformatörlerin Çift Yönlü Kodlayıcı Gösterimleri
CNN	Evrişimli Sinir Ağı
CPU	Merkezi İşlem Birimi
DNN	Derin Sinir Ağları
GAN	Üretken Çekişmeli Ağlar
GPU	Grafik İşlemci Ünitesi
GPT	Generatif Önceden Eğitimli Transformatör
LSTM	Uzun kısa süreli bellek ağları
ML	Makine Öğrenimi
MT	Makine Çevirisi
NLP	Doğal Dil İşleme
RNN	Tekrarlayan Sinir Ağı
VAE	Varyasyonel Otoenkoder

Semboller Listesi

$F(x)$	Katmanın öğrenmesi gereken dönüşümü
x	Giriş verileri

Bölüm 1

Giriş

Günümüzde uydu teknolojisindeki ilerlemeler, stratejik gözetleme ve savunma alanında da büyük değişiklikleri beraberinde getiriyor. Bu değişiklikler, askeri uçakların tespitini ve takibini savunma ve güvenlik stratejilerinin temel taşı haline getirdi. Uydu görüntüleri geniş coğrafi alanları kapsaması ve olayların gözlemlenmesine olanak sağlaması nedeniyle bu alanda önemli bir rol oynamaktadır. Bu bağlamda bu çalışma, uydu görüntülerinden askeri uçak tiplerinin tanınmasında derin öğrenme algoritmalarının etkinliği ve potansiyeline odaklanarak savunma teknolojisindeki son gelişmeleri araştırmayı amaçlamaktadır.

Askeri uçak tespiti, geleneksel yöntemlerin karşılaştığı çeşitli zorlukları beraberinde getirmektedir ve bu bağlamda derin öğrenme modellerinin uygulanması, bu zorluklara etkili çözümler sunma potansiyeline sahiptir. Derin öğrenme, büyük miktarda veriyi analiz ederek karmaşık kalıpları keşfetme yeteneğiyle bilinir. Bu çalışma, özellikle uydu görüntülerini kullanarak askeri uçak türlerini tanımak için derin öğrenme algoritmalarının kullanımını ayrıntılarıyla anlatıyor.

Uydu görüntüleri, yüksek çözünürlüklü ayrıntılara sahip geniş bir veri setini temsil ediyor ve derin öğrenme modelleri, askeri uçakları diğer nesnelere ayırtmak için bu veri setlerini etkili bir şekilde işleyerek benzersiz özellikleri öğrenme yeteneği sağlıyor. Bu çalışmanın temel iddiası, derin öğrenme modellerinin uydu görüntüleri üzerinde başarılı bir şekilde eğitilebileceği ve bu modellerin askeri uçak tespitinde geleneksel yöntemlere göre daha yüksek doğruluk ve hassasiyet sağlayabildiğidir. Araştırmanın amacı, uydu teknolojisindeki son gelişmeler ile derin öğrenmeyi birleştirerek savunma ve güvenlik stratejilerine yeni boyutlar kazandırmaktır.

Teknolojinin hızla gelişmesiyle birlikte askeri uçakların kullanımı ve çeşitleri önemli ölçüde artmış, bu uçakların tespiti ve takibi zorlaşmıştır. Bu bağlamda uydu görüntülerinin kullanılması, geniş bir coğrafi alan üzerindeki hava trafiğinin izlenebilmesi avantajına sahiptir. Bununla birlikte, bu büyük veri kümelerini etkili bir

şekilde analiz etmek ve askeri uçakları diğer nesnelere ayıran belirgin özellikleri belirlemek zor olduğundan, geleneksel yöntemlerin sınırlamaları vardır. Derin öğrenme algoritmalarının özellikle uydu görüntüleri aracılığıyla bu tür zorlukların üstesinden gelme potansiyeli, araştırmanın ana motivasyonunu oluşturuyor. Bu potansiyel, derin öğrenme modellerinin askeri uçak tespitinde etkili bir şekilde kullanılabileceği ve hava savunma sistemlerinin daha güçlü ve hassas hale getirilebileceği yönündeki umutları artırıyor.

Bu araştırma, savunma sanayi alanındaki teknolojik ilerlemeleri desteklemek ve askeri stratejilerin etkinliğini artırmak için uydu görüntüsü analizi ve derin öğrenmenin entegrasyonunu ele almaktadır. Bu araştırmanın geri kalan bölümleri üç ana bölüme ayrılmıştır: literatür taraması ve metodoloji, bulgular ve sonuçlar. Öncelikle literatür taraması ve metodoloji bölümünde benzer çalışmaların göze çarpan sonuçlarını ve bu alanda var olan boşlukları değerlendirdik. Bu değerlendirme, savunma teknolojisi, uydu görüntüsü analizi ve derin öğrenme alanlarındaki önceki araştırmalara dayanarak çalışmanın kavramsal çerçevesini oluşturuyor. Daha sonra derin öğrenme modelinin nasıl eğitildiğini ve uydu görüntüsü verilerinin nasıl kullanıldığını ayrıntılarıyla anlatıyor olacağız. Bu aşama, çalışmanın temel metodolojik yaklaşımını açıklarken veri toplama ve analiz sürecini açıklığa kavuşturarak çalışmanın bilimsel geçerliliğini sağlar.

Sonuçlar bölümü, elde edilen sonuçları ve bunların araştırma sorusuna veya hipoteze nasıl cevap verdiğini açıklar. Bu bölüm, askeri uçak tespitinde uydu görüntüsü analizinin ve derin öğrenme modellerinin etkinliğine ilişkin verileri sunmakta ve sonuçların savunma stratejisi açısından önemini vurgulamaktadır. Son olarak “Sonuçlar” bölümünde çalışmanın genel katkısı, sınırlılıkları ve gelecek araştırmalara yönelik öneriler sunulmaktadır. Bu bölüm, çalışmanın savunma teknolojisi ve stratejisi üzerindeki etkisini değerlendirmekte, bulgularının genel önemini vurgulamakta ve gelecekte benzer araştırmalara yol gösterecek öneriler sunmaktadır.

Bölüm 2

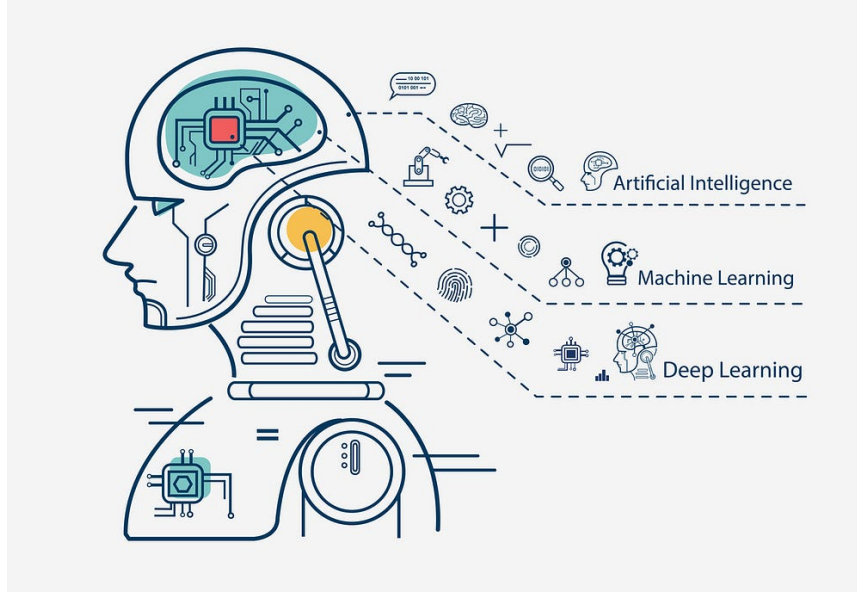
Literatür Taraması ve Metodoloji

2.1 Derin Öğrenme Nedir

Derin öğrenme, özellikle yapay sinir ağlarının katmanları aracılığıyla insan beyninin çalışma prensiplerine benzeyen algoritmaları birleştirerek, büyük veri kümelerinden öğrenme yeteneği ile tanımlanan makine öğreniminin bir alt kümesidir (Maraş ve Sarıyıldız 2023). Bu yaklaşım, büyük miktarda veri üzerinde eğitilmiş sinir ağlarını kullanarak nöronların yapılandırılması sürecine dayanmaktadır. Eğitim sonrasında elde edilen model, yeni verileri işleyebilme yeteneğine sahip derin öğrenme modeli oluşturur. Derin öğrenme, öğrenme süreci sırasında hiyerarşik özellikleri keşfetmeye odaklanarak karmaşık veri kümelerindeki kalıpları daha etkili bir şekilde tespit etme yeteneğiyle bilinir. Bu yetenek, uydu görüntüleri gibi büyük ve karmaşık veri kümelerinde askeri uçakların tespiti gibi özellikle zorlu görevlere başarıyla uygulanabilir (Doğan ve Türkoğlu 2018).

Derin öğrenme modelleri, çeşitli veri kaynaklarından bilgi alıp, bu verileri insan müdahalesine gerek kalmadan gerçek zamanlı olarak analiz edebiliyor. Bu modellerin eğitimi sırasında GPU gibi grafik işlem birimleri kullanılmaktadır. Bunun nedeni, bu birimlerin aynı anda birden fazla hesaplama işlemini gerçekleştirme yeteneğine sahip olması ve derin öğrenme modellerinin karmaşık yapılarını eğitmek için optimize edilmiş olmasıdır. Bu da karmaşık derin öğrenme modellerinin eğitim sürecinde yüksek performans ve etkili sonuçlara ulaşmada büyük avantajlar getiriyor (Doğan ve Türkoğlu 2018).

Derin öğrenme, çeşitli yapay zeka teknolojilerini yönlendirebilen ve otomasyon yoluyla analitik görevleri iyileştirebilen bir kavramdır. Günümüzde insanlar günlük hayatlarında internette gezinirken ya da cep telefonlarını kullanırken derin öğrenmeyi kullanan pek çok yapay zeka uygulamasıyla karşılaşılıyorlar. Bu, derin öğrenmenin yaygınlaştığı ve teknolojinin günlük hayatımıza entegre olduğu çağın önemli bir göstergesidir.



Şekil 2.1: Derin Öğrenme

Makine öğrenimi, bilgisayarları insan beyninden ilham alan verileri işleyecek şekilde eğitmeyi amaçlayan, yapay zeka (AI) alanında kullanılan bir yöntemdir. Belirli bir veri kümesindeki kalıpları tanımlamak için özel algoritmalar kullanılır. Algoritmalar genellikle büyük miktarda veri üzerinde eğitilir ve bu eğitim süreci, veri analizi yoluyla yeni verileri işleyebilen eğitilmiş modellerin oluşturulmasına olanak sağlar (Tütüncü 2022).

Makine öğrenimi modelleri; resimler, metinler ve ses dahil birçok veri türündeki karmaşık kalıpları tanıyabilir. Bu özellik, doğru tahminler ve tahminler yapmanızı sağlar. Ek olarak makine öğrenimi, genellikle insan zekası gerektiren görevleri otomatikleştirmek için derin öğrenme tekniklerinden yararlanabilir (Tütüncü 2022). Örneğin görselleri görüntüleme veya ses dosyalarını metne dönüştürme gibi görevleri başarıyla gerçekleştirebilirsiniz.

Makine öğrenimi, otomasyon ve analitik görevlerin geliştirilmesinde önemli rol oynayan bir yapay zeka kavramıdır. Bu yaklaşım, teknolojik gelişimin önünü açmak ve yapay zeka teknolojilerine yön vermek amacıyla çeşitli uygulama alanlarında kullanılmaktadır (Tütüncü 2022).



Şekil 2.2: Yapay zeka alt grupları

Yapay Zeka, insan beyninden ilham alan ve bilgisayarların veri işlemesine olanak tanıyan bir yapay zeka (AI) yöntemidir. İdeal bir perspektiften bakıldığında bu yaklaşım, insan zekasının gelişmiş bilişsel yeteneklerini ve otonom davranışını sergileyen yapay bir işletim sistemini ifade etmektedir. Yapay zeka, bir bilgisayarın veya bilgisayar kontrollü robotun normalde akıllı varlıklarla ilişkili görevleri başarıyla yerine getirme yeteneğidir (Doğan ve Türkoğlu 2018).

Bu bağlamda yapay zeka, öğrenme ve problem çözme yetenekleri, bilgisayar sistemlerinin insan benzeri bilişsel işlevleri taklit edebilme yeteneği anlamına gelmektedir. Aynı zamanda biliş, öğrenme, farklı kavramları ilişkilendirme, düşünme, fikirleri ifade etme, problem çözme, iletişim, karar verme gibi çeşitli becerileri de içermelidir.

Yapay zeka, otomasyonu ve analitik görevleri iyileştirebilen birçok yapay zeka (AI) teknolojisinin temel konseptini oluşturur. Bu bağlamda yapay zeka, teknolojik gelişmenin ve yeniliğin temel itici güçlerinden biridir. Bilgisayar sistemlerine çeşitli uygulamalarda kullanılarak insan benzeri zeka ve işlevsellik kazandırmayı amaçlayan bir disiplindir (Kırca 2021).

2.2 Derin Öğrenmenin Tarihi

Derin öğrenme, yapay zekanın (AI) bir alt alanı olan makine öğreniminin (ML) bir alt alanıdır. Makine öğrenimi, bilgisayarların veri analizine dayalı olarak öğrenmesini sağlayan bir disiplindir. Derin öğrenme, sinir ağları olarak adlandırılan yapay sinir sistemleriyle çalışır. Sinir ağları, insan beynindeki sinir hücrelerinin işlevini taklit eder.

Derin öğrenmenin tarihi 1943'e kadar uzanıyor. Bu yıl Walter Pitts ve Warren McCulloch sinir ağının ilk matematiksel modelini geliştirdiler. Bu model sinir ağlarının temelini oluşturmuş ve derin öğrenmenin gelişmesine yol açmıştır (Burgaz, 2020).

1950'lerde Alan Turing makine öğrenimi hakkında bir tahminde bulundu. Bu tahmin, derin öğrenmenin potansiyelini ilk kez ortaya koyan önemli bir çalışma olarak tarihe geçti.

1957'de Frank Rosenblatt algılayıcı adı verilen bir model geliştirdi. Bu model derin sinir ağlarının temelini attı. Perceptron basit bir model olmasına rağmen derin öğrenmenin temel kavramlarını içeriyordu (Burgaz, 2020).

1960'larda sinir ağları konusundaki araştırmalar durakladı. Bunun nedeni, sinir ağlarının sınırlarını anlamadaki zorluklar ve derin öğrenmenin potansiyelinin tam olarak anlaşılabilmesiydi (Burgaz, 2020).

1980'lerde sinir ağları üzerine araştırmalar yeniden başladı. Bu süre zarfında evrimsel öğrenme adı verilen yeni bir teknik geliştirildi. Evrimsel öğrenme, özellikle görüntü tanıma gibi uygulamalarda derin öğrenmenin verimliliğini ve doğruluğunu önemli ölçüde artıran bir gelişme olarak ortaya çıkmıştır, (Burgaz 2020).

1990'lı yıllarda denetimsiz derin öğrenme adı verilen yeni bir yaklaşım geliştirildi. Bu metodoloji, etiketlenmemiş verileri kullanarak sinir ağlarını eğitmenize olanak tanıyan bir yaklaşım sağlar (Burgaz, 2020).

2000'li yıllarda derin öğrenme araştırmaları hızla ilerledi. Bu süre zarfında AlexNet, VGGNet ve ResNet gibi önemli derin öğrenme modelleri geliştirildi. Bu modeller

görüntü tanıma, doğal dil işleme (NLP), makine çevirisi gibi çeşitli alanlarda büyük başarılar elde etmiştir (Burgaz, 2020).

Derin öğrenme, 2010'lu yıllarda yapay zeka alanında en popüler araştırma konularından biri haline geldi. Bu süre zarfında oyun, sürücüsüz arabalar ve sağlık hizmetleri gibi yeni uygulamalarda derin öğrenme kullanılmaya başlandı (Burgaz, 2020).

Şu anda derin öğrenme, yapay zeka alanındaki en güçlü ve etkili araçlardan biri olarak kabul ediliyor. Bu teknik, görüntü tanıma, doğal dil işleme, makine çevirisi, oyun, sürücüsüz arabalar ve sağlık hizmetleri dahil olmak üzere birçok alanda devrim yarattı.

2.3 Derin Öğrenmenin Geleceği

Derin öğrenmenin geleceği çok parlak. Yapay zekanın en hızlı büyüyen alanlarından biri olan derin öğrenme, gelecekte aşağıdaki alanlarda daha da fazla uygulama alanı bulacaktır.

Derin öğrenme, görüntü tanıma alanında şimdiden büyük başarı elde etti. Gelecekte derin öğrenmenin görüntü tanıma da daha da hassas ve güvenilir hale gelmesi bekleniyor.

Doğal dil işleme alanında derin öğrenme şu anda büyük bir başarı elde ediyor. Gelecekte, derin öğrenmenin doğal dil işlemeyi ilerletmesi ve insan düzeyinde dil anlayışına ulaşması bekleniyor.

Makine çevirisinde derin öğrenme şu ana kadar büyük bir başarı elde etti. Makine çevirisinde derin öğrenmenin gelecekte daha verimli ve doğru hale gelmesi bekleniyor.

Oyunlarda derin öğrenme şu ana kadar büyük bir başarı elde etti. Oyunlarda derin öğrenmenin gelecekte gelişmesi ve insan düzeyindeki oyun yeteneklerine ulaşması bekleniyor.

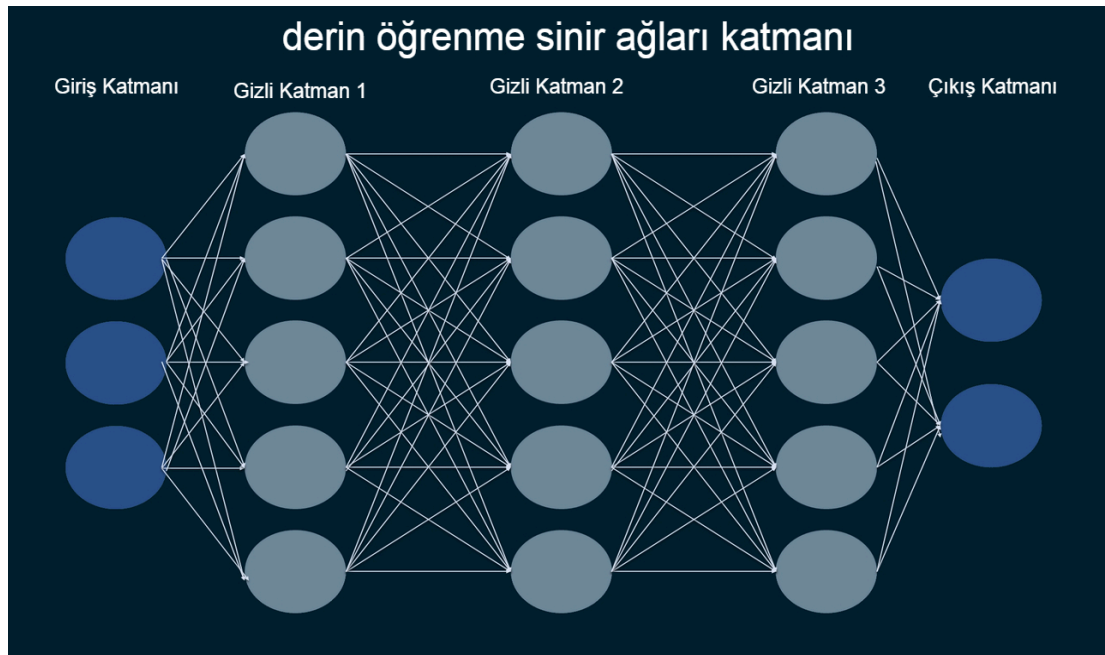
Otonom araç sürüşünde derin öğrenme halihazırda büyük bir başarı elde etti. Derin öğrenme, muhtemelen gelecekte otonom sürüşü daha güvenli ve daha verimli hale getirecek.

Derin öğrenme sağlık alanında giderek daha fazla kullanılıyor. Derin öğrenmenin gelecekte tıp alanında da yaygın olarak kullanılacağına, hastalıkların erken teşhisine, tedavilerin iyileştirilmesine ve yeni tedavilerin geliştirilmesine katkı sağlayacağına inanılıyor.

Yapay zeka alanındaki en öne çıkan ve önemli gelişmelerden biri olan derin öğrenmenin, bu yenilikçi teknolojinin gelecekte insan yaşamının birçok alanına etki etmesi bekleniyor.

2.4 Derin Öğrenme Katmanları

Derin öğrenme, çok katmanlı sinir ağlarını içeren bir makine öğrenme yöntemidir. Bu yöntemler genellikle farklı öğrenme türlerini gerçekleştiren birden fazla katman içerir. Tipik bir derin öğrenme modelinin temel katmanları şunlardır:



Şekil 2.3: Derin öğrenme katmanları

Giriş katmanı: Veriyi alır ve ağa sunar. Tipik olarak verilerin vektör veya matris formatında temsil edilmesi gerekir (İnik ve Ülker, 2017).

Gizli Katmanlar: Derin öğrenme modellerinde giriş katmanını takip eden ve çeşitli öğrenme işlemlerini gerçekleştiren genellikle katman bulunur. Her bir gizli katman, içerdiği nöron sayısına bağlı olarak birbirine bağlı ağırlıklar ve aktivasyon

fonksiyonları ile karakterize edilir. Çoğu derin öğrenme modeli birden fazla gizli katman içerir (Tan, 2019).

Ağırlık katmanı: Her bağlantı arasındaki ağırlıkları içeren katmanı, giriş verilerini ve çeşitli öğrenme süreçlerini birleştirerek çıktıyı üreten temel bileşendir. Bu ağırlıklar derin öğrenme modelinin öğrenme yeteneğini belirleyen önemli parametrelerdir (Kırca, 2021).

Aktivasyon katmanı: Her gizli katmanda kullanılan aktivasyon fonksiyonu, nöronların çıktısını belirleyerek modelin öğrenme yeteneğini geliştiren önemli bir bileşendir. Bu işlevler, derin öğrenme modelinin her bir gizli katmanındaki nöronların aktivasyon düzeyini ayarlamak, model karmaşıklığını ve öğrenme esnekliğini sağlamak için kullanılır (Kutlu, 2019).

Çıkış Katmanı: modelinin nihai çıktısını üreten katman, çeşitli görevlere yönelik tahminler sağlayan önemli bir bileşendir. Çıktı katmanının yapısı, modelin çözmeye çalıştığı spesifik göreve bağlıdır. Örneğin sınıflandırma veya regresyon. Bu katman, derin öğrenme modelinin eğitim sürecini tamamlar ve giriş verilerinden elde edilen bilgileri belirli bir formatta temsil eder (İnik ve Ülker, 2017).

Normalizasyon Katmanı: Model öğrenme sürecini stabilize etmek ve giriş verilerini standartlaştırmak için kullanılan katmanına düzenleme tekniği denir. Örneğin toplu normalleştirme gibi teknikler, derin öğrenme modellerinin daha hızlı ve daha güvenilir şekilde eğitilmesine yardımcı olabilir. Bu teknikler, ağıdaki katmanların gradyan kaybını azaltarak ve daha tutarlı bir eğitim sürecine izin vererek model performansını artırabilir (Doğan ve Türkoğlu, 2018).

Bu katmanlar, derin öğrenme modellerinin karmaşık öğrenme süreçlerini gerçekleştirmesini sağlayan önemli bileşenlerdir. Farklı derin öğrenme mimarileri farklı sayıda ve türde katman içerebilir. Ek olarak, transfer öğrenme, yinelemeli öğrenme ve takviyeli öğrenme gibi teknikler, bu temel katmanları daha da iyileştirmek ve model performansını iyileştirmek için kullanılabilir.

2.5 Derin Öğrenme Modellerinin Kullanıldığı Alanlar

Derin öğrenme modelleri çeşitli görevler için kullanılabilir. Görüntü tanıma alanında derin öğrenme modelleri dikkate değer bir başarı elde etti. Bu modeller nesne tanıma, yüz tanıma, metin tanıma gibi görevlerde etkin bir şekilde kullanılabilir. Derin öğrenme modelleri, doğal dil işleme alanında büyük başarı elde etti. Bu modeller metin çevirme, soruları yanıtlama, e-postaları sınıflandırma gibi görevlerde oldukça etkili bir şekilde kullanılabilir. Derin öğrenme modelleri, makine çevirisi alanında büyük başarı elde etti. Bu modeller, metni bir dilden diğerine çevirirken çok etkili bir şekilde kullanılabilir. Derin öğrenme modelleri, otonom araç sürüşü alanında giderek daha popüler hale geliyor. Bu modeller, araç ortamını etkili bir şekilde yakalamak ve güvenli bir sürüş deneyimi sağlamak için tasarlanmıştır. Derin öğrenme sağlık alanında giderek daha fazla kullanılıyor. Bu modeller hastalıkların doğru teşhisi, tedavilerin geliştirilmesi, yeni ilaçların keşfi gibi önemli alanlarda kullanılabilir (Yılmaz ve Kavzoğlu, 2021).

2.6 Derin Öğrenme Modellerinin Avantajları

Derin öğrenme modelleri, karmaşık ilişkileri öğrenme yeteneğine sahiptir; bu da onları özellikle görüntü tanıma, doğal dil işleme ve makine çevirisi gibi uygulamalarda etkili kılar.

Derin öğrenme modelleri genellikle doğruluk oranlarına göre tanımlanır. Bu önemli bir özelliktir ve özellikle güvenlik, finans ve sağlık gibi kritik uygulama alanlarında değerlidir.

Derin öğrenme modelleri, yeni veri kümeleriyle etkileşime girerek sürekli bir öğrenme sürecine girebilmektedir. Bu özellikler, değişen koşullara özellikle etkili bir şekilde uyum sağlamalarına olanak tanır.

2.7 Derin Öğrenme Modellerinin Dezavantajları

Derin öğrenme modelleri genellikle etkili bir şekilde çalışmak için büyük miktarda veri gerektirir. Bu durum, eğitim sürecini pahalı ve zaman alıcı hale getirebilir.

Derin öğrenme modelleri aşırı uyum riski oluşturabilir. Aşırı uyum, bir modelin eğitim verilerine aşırı uyum sağladığı bir durumu ifade eder. Bu durum, modelin gerçek verileri etkili bir şekilde işleme yeteneğini olumsuz yönde etkileyebilir.

Genel olarak derin öğrenme modellerini açıklamak zor olabilir. Bu durumda, model kararlarını anlamak ve açıklamak için ek çaba gerektirebilir.

2.8 Derin Öğrenme Mimarileri

Derin öğrenme modelleri, istenilen işi gerçekleştirmek için yapay sinir sistemlerini kullanır. Bu sistemlere genellikle sinir ağları denir. Sinir ağı, insan beynindeki nöronların işlevini taklit eden bir yapıdır. Derin öğrenme modelleri, karmaşık görevleri gerçekleştirmek için tasarlanmış birden fazla katmandan oluşur. Bu katmanlar girdi verilerini işler ve çıktı verilerini üretir (Kunt, 2021). Derin öğrenme modelleri farklı türlere göre sınıflandırılabilir.

Yapay sinir ağları: Derin öğrenme modellerinin temelini oluşturan yapay sinir ağları, insan beynindeki nöronların işlevini taklit eden yapılardır (Kunt, 2021).

Evrişimli Sinir Ağı (CNN), özellikle görüntü verileri içindeki yerel ilişkileri tespit etmek için kullanılan bir tür sinir ağıdır. Bu ağlar, görüntü tanıma ve doğal dil işleme gibi çeşitli uygulamalarda yaygın olarak kullanılmaktadır (Bayram., 2019).

Tekrarlayan sinir ağı (RNN) zaman serisi verilerindeki ilişkileri ortaya çıkarmak için özel olarak tasarlanmış bir sinir ağı türüdür. RNN'ler konuşma tanıma ve doğal dil işleme gibi çeşitli uygulamalarda yaygın olarak kullanılmaktadır (Burgaz, 2020).

Üretken Rekabet Ağı (GAN), gerçekçi veriler üretmeyi amaçlayan bir sinir ağı türüdür. GAN'lar, özellikle görüntü oluşturma, sanat oluşturma ve metin oluşturma gibi çeşitli uygulamalarda yaygın olarak kullanılmaktadır (Burgaz, 2020).

Derin Sinir Ağı (DNN), genellikle birden fazla gizli katmandan oluşan bu temel derin öğrenme modeli, büyük veri kümeleri üzerinde sınıflandırma, regresyon ve örüntü tanıma gibi görevler için kullanılır (Şeker vd., 2017).

Uzun Kısa Süreli Bellek Ağları (LSTM), geliştirilmiş bir RNN türüdür. Gradyan kayıplarını daha etkili bir şekilde yönetebilen ve uzun vadeli bağımlılıkları daha etkin bir şekilde yönetebilen bir mimariye sahiptir (Şeker vd., 2017).

Doğrusal Olmayan Bağlam Modeli (Transformer), doğal dil işleme görevleri için özel olarak tasarlanmış dikkat mekanizmasına sahip bir modeldir. Bu kategoriye giren yaygın modeller arasında GPT (Generatif Önceden Eğitimli Transformatör) ve BERT (Transformatörlerin Çift Yönlü Kodlayıcı Gösterimleri) bulunur (Kutlu,2019).

Varyasyonel otomatik kodlayıcı (VAE), veri kümeleri üzerinde eğitim ve öz değerlendirme için kullanılan bir öğrenme modeli türüdür. Bu model özellikle verinin temsilini öğrenmek ve bu temsilde değişiklik yapmak için kullanılır (Kunt,2021).

Öğrenme transfer modelleri, bir birincil görevde öğrenilen bilgilerin başka bir görevde kullanılmasına olanak tanıyan modellerdir. Genel olarak öğrenilen özellikler geneldir ve bu özellikler genellikle farklı kavramlar arasında aktarılabilir (Tan, 2019).

Bu modeller, farklı endüstrilerde kullanılacak farklı görevleri çözmek için tasarlanmıştır. Ancak hangi modeli kullanacağınız veri kümenizin özelliklerine ve çözmeniz gereken spesifik göreve bağlıdır.

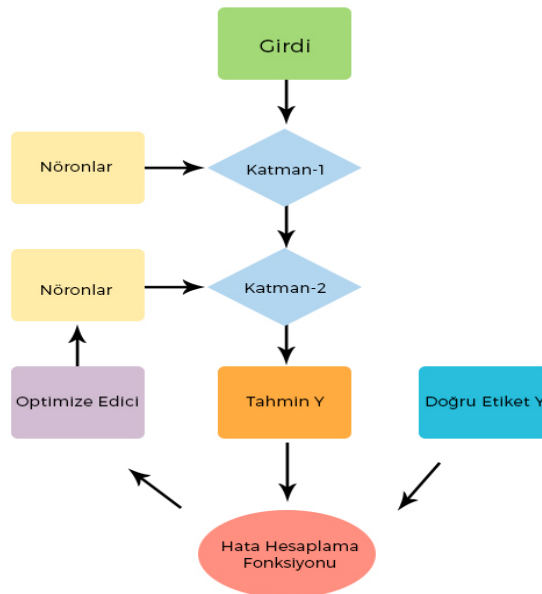
2.8.1 Yapay Sinir Ağları

Yapay sinir ağları, insan beynindeki nöronların işlevini taklit etmek için matematiksel modeller kullanan yapılardır. Bu modeller büyük miktardaki verilerden öğrenme yeteneğine sahiptir ve bu da onları özellikle bilgisayarlı görme, doğal dil işleme ve makine çevirisi gibi alanlarda çeşitli uygulamalarda son derece değerli kılmaktadır. Bu yapay zeka teknikleri, biyolojik sinir ağlarının temel prensiplerini kullanarak bilgisayar sistemlerini geliştirmeyi amaçlamaktadır. Bu bağlamda yapay sinir ağları, örüntüleri tanıma, karmaşık görevleri yerine getirme ve büyük veri kümelerini analiz ederek yeni bilgiler öğrenme yetenekleriyle öne çıkıyor. Özellikle bilgisayarlı görme, doğal dil işleme ve makine çevirisi gibi uygulama alanlarında yapay sinir ağları, büyük miktarda veriyi etkin bir şekilde işleyerek yüksek performans ve başarı elde etme potansiyeline sahiptir (Mutludoğan, 2020).

Yapay Sinir Ağları, genellikle katmanlar olarak adlandırılan bir dizi bağlantılı düğümünden oluşur. Her düğüm, diğer düğümlerden gelen girdileri işleyerek çıktı üretme yeteneğine sahiptir. Bu işlevler, parametreler adı verilen ağırlıklar ve eşikler tarafından kontrol edilir. Katmanlar arasındaki bu bağlantılar, sinir ağının genel yapısını ve işlevselliğini belirler. Her katman, belirli görevleri gerçekleştirmek için özelleştirilebilir işlevler ve öğrenilebilir parametrelerle tanımlanır. Bu yapı, yapay sinir ağlarının karmaşık görevleri öğrenme ve çeşitli uygulamalarda başarılı sonuçlar elde etme yeteneğini desteklemektedir (Aktürk ve Serbest, 2022).

Yapay sinir ağı, eğitim verilerindeki kalıpları tanımak için özel olarak tasarlanmış bir matematiksel modeldir. Eğitim süreci sırasında ağırlıklar ve eşikler, modelin giriş verilerine en iyi uyacak şekilde ayarlanır ve geliştirilir. Bu süreç, ağın belirli görevlerdeki performansını optimize etmek ve genel olarak yeni veri örneklerine daha duyarlı hale getirmek için tekrarlanır (Aktürk ve Serbest, 2022).

Yapay sinir ağlarının birçok avantajı vardır. Büyük miktarlardaki verilerden öğrenme yeteneği, onu karmaşık kalıpları ve ilişkileri tanımlamak için ideal kılar. Ayrıca genelleme becerileri de yüksektir. Yani yeni veri örnekleriyle karşılaşıldığında genellikle iyi performans gösterir.



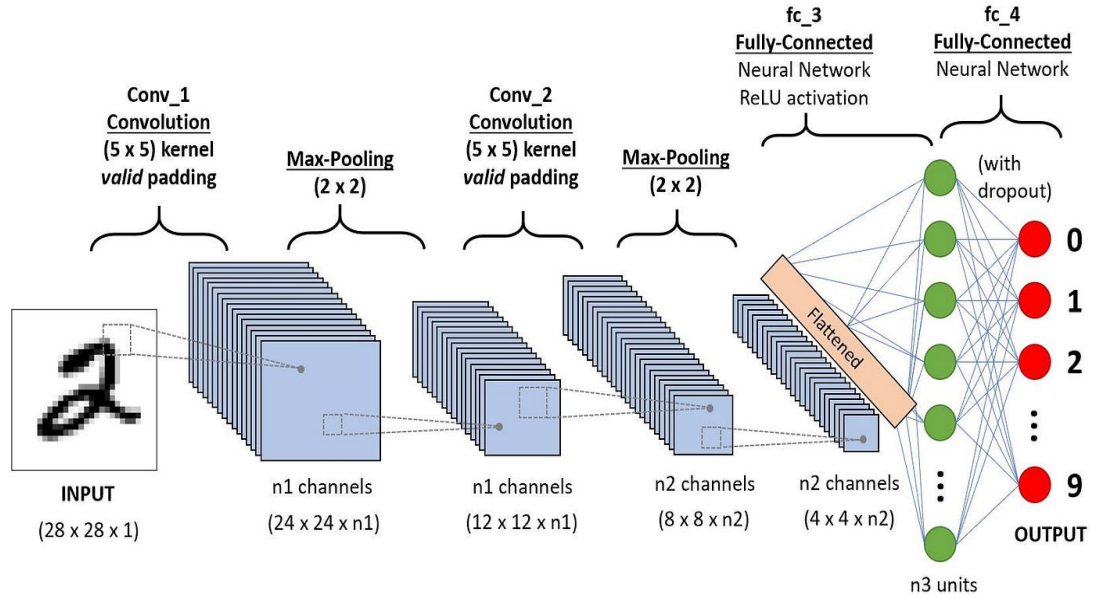
Şekil 2.4: Yapay sinir ağı yapısı

Yapay sinir ağlarının birçok dezavantajı vardır. Eğitim süreci çoğu zaman zaman alıcı ve karmaşık olabilir. Ayrıca model performansı, kullanılan eğitim verilerinin kalitesine ve miktarına bağlı olarak değişiklik gösterebilir.

Yapay sinir ağlarının geniş bir uygulama alanı vardır. Bilgisayarlı görme alanında, görüntülerdeki nesnelerin tanınması ve sınıflandırılması amacıyla yapay sinir ağlarından yararlanılmaktadır. Doğal dili işlerken, örneğin metin çevirisi, soru-cevap, konuşma tanıma ve daha fazlası için yapay sinir ağları kullanılabilir. Makine çevirisi, metni bir dilden diğerine çevirmek için yapay sinir ağlarını etkili bir şekilde kullanır. Bu teknolojinin potansiyel uygulamalarının yanı sıra tıp, finans, üretim ve daha birçok alanda önemli faydalar sunması beklenmektedir.

2.8.2 Evrişimsel Sinir Ağları (CNN'ler)

Evrişimli sinir ağları (CNN'ler), görüntü verileri içindeki yerel ilişkileri tespit etmek için tasarlanmış bir tür yapay sinir ağıdır. Bu ağlar başta görüntü tanıma, doğal dil işleme ve makine çevirisi olmak üzere çeşitli uygulamalarda etkin bir şekilde kullanılmaktadır.



Şekil 2.5: Evrişimsel sinir ağları yapısı

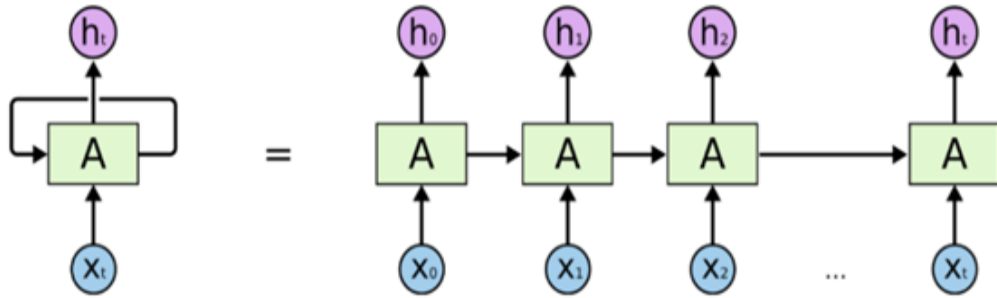
CNN birkaç temel bileşenden oluşur:

- Evrişimli katman: Evrişimli katman, görüntü verilerinden yerel özellikleri çıkarmak için tasarlanmış bir katmandır. Bu katmanlar, diğer şeylerin yanı sıra filtre matrislerini kullanarak görüntü verilerini işler. Bir filtre matrisi, belirli desenleri tanımak ve yerel özellikleri vurgulamak için görüntü verilerini tarar. Evrişimsel katmanlar görüntü tanıma, nesne tanıma ve diğer birçok uygulama alanında etkin bir şekilde kullanılmaktadır (Somuncu, 2022).
- Maksimum havuzlama katmanı: Maksimum havuzlama katmanı, evrişimsel katmanın çıktısını azaltmak ve özetlemek için entegre edilen bir katmandır. Bu katmanlar görüntü verilerinin en belirgin özelliklerini koruma eğilimindedir. Maksimum havuzlama, bölgesel özellikleri vurgulayarak ve her bölge için maksimum değeri seçerek genel bir genel bakış sağlar. Bu teknik, özellik haritasının boyutunu azaltarak hesaplama miktarını azaltır ve aynı zamanda özelliklerin dönüşümlere (hareketlere) ve küçük değişikliklere karşı daha dayanıklı olmasına yardımcı olur (Albayrak,2021).
- Tam bağlı katmanlar: Tam bağlı katmanlar, evrişimli katmanların çıktısından genel özellikleri çıkarmak ve bir sinir ağının karar verme sürecini yönlendirmek için kullanılır. Evrişimli katmanlar, bir görüntü veya veri içindeki yerel özellikleri vurgulamak için kullanılır. Bununla birlikte, tam bağlı katmanlar, bu evrişimli katmanların çıktılarından elde edilen geniş yelpazedeki özellikleri anlamak ve bunlardan yararlanmak üzere tasarlanmıştır. Tam bağlı bir katman, tüm giriş özelliklerini birbirine bağlayarak önceki evrişimli katmanlardaki özellikleri entegre eder. Bu entegrasyon süreci, sinir ağının genel bir anlayış oluşturmaya ve nihai çıktıyı üretmesine yardımcı olur. Ek olarak, tam bağlı katmanlar genellikle sınıflandırma ve regresyon gibi görevler için kullanılır çünkü genel özellikleri anlamak ve bu özelliklere dayalı olarak tahminler yapmak önemlidir (Albayrak, 2021).
- Transfer öğrenimi: Evrişimli sinir ağları (CNN'ler), genellikle büyük veri setleri üzerinde önceden eğitilmiş model ağırlıklarını kullanarak küçük veri setlerini ve diğer görevleri başarıyla tamamlamak için kullanılır. Transfer öğrenme uygulamalarında yaygın olarak tercih edilir. Transfer öğrenme, bir görevde kazanılan bilgiyi başka bir görevde kullanma fırsatı sağlar. CNN'ler, büyük veri kümeleri üzerinde eğitilen modellerden öğrenilen özellikler

kullanılarak daha sınırlı veri kümelerine ve diğer görevlere uyarlanabilir. Bunun özellikle görsel tanıma görevlerinde başarılı olduğu kanıtlanmıştır. Bu, model performansını artırır ve öğrenme sürecini daha verimli hale getirir (Tan, 2019).

2.8.3 Tekrarlayan Sinir Ağları (RNN'ler)

Tekrarlayan sinir ağları (RNN'ler), zaman serisi verilerindeki ilişkileri keşfetmek için tasarlanmış bir tür yapay sinir ağıdır. RNN'ler, diğerlerinin yanı sıra konuşma tanıma, doğal dil işleme ve makine çevirisi gibi uygulamalarda yaygın olarak kullanılmaktadır. Bu ağlar, veriler arasındaki zamansal bağımlılıkları yönetebildikleri için zaman içindeki kalıpları tanıma ve anlama yetenekleriyle bilinir. Bu özellikler ses ve zaman serisi veri görevlerinde etkin bir şekilde kullanılmasına olanak sağlar (Şeker vd., 2017).



Şekil 2.6: Tekrarlayan sinir ağları yapısı

Tekrarlayan sinir ağları (RNN'ler), sıralı verileri işlemek için özel olarak tasarlanmış derin öğrenme modelleridir. Standart ileri beslemeli sinir ağları her veri örneği üzerinde bağımsız olarak çalışırken, RNN'ler önceki adımlarda elde edilen bilgileri içeren gizli durumları koruyarak sıralı veri yapılarını daha etkili bir şekilde işleyebilir (Şeker vd., 2017).

RNN birkaç önemli bileşenden oluşur:

- Gizli katmanlar: Tekrarlayan sinir ağları (RNN'ler) tarafından işlenen zaman serisi verileri için kullanılan gizli katmanlar, daha önce öğrenilen bilgilere dayanarak sonraki zaman adımlarında tahminler yapan anahtar bileşenleri içerir. Bu gizli katmanlar, RNN'lerin zaman serisi analizi görevlerinde,

özellikle de geçmiş verilerdeki kalıpları anlama ve gelecekteki durumları tahmin etme yeteneklerinde etkili olmasını sağlar (Kutlu, 2019).

- Çıkış katmanı: Sonuç katmanı, gizli katmanın çıktısına göre kararlar verir. Bu katman özellikle konuşma tanıma gibi uygulamalarda sesleri, doğal dil işleme gibi uygulamalarda ise kelimeleri algılamak için kullanılır ve gizli katmanlardan gelen bilgileri işleyerek nihai çıktıyı üretir (Kutlu, 2019).

RNN'nin ana özellikleri şunlardır:

- Yinelemeli Mekanizma: Tekrarlayan bir sinir ağı (RNN), zamanın belirli bir noktasındaki çıktıyı ve iç durumu alarak ve zamanın bir sonraki noktasında çıktıda iç durumu üreterek bir döngüde çalışır. Geçmiş bilgiler dahili bir durumda depolandığından ve gelecekteki çıktıyı etkilediğinden, bu bir tür depolama mekanizmasını temsil eder (Kutlu, 2019).
- Parametre paylaşımı: Her zaman adımında ağırlıkların aynı olması, tekrarlayan sinir ağlarının (RNN'ler) farklı uzunluklardaki sürekli verileri işlemesine olanak tanır (Kutlu, 2019).
- Sıralı Veri İşleme: Tekrarlayan sinir ağları (RNN'ler), metin, zaman serisi ve ses gibi doğal sıralı veri türlerini işlemek için özel olarak tasarlanmıştır. Bu özellikleri dil modelleme, metin üretme, çeviri, konuşma tanıma gibi uygulamalarda etkin bir şekilde kullanılmasına olanak sağlar (Kutlu, 2019).

Geleneksel RNN'lerin çeşitli sınırlamaları vardır:

- Kaybolan gradyan sorunu: Uzun dizili verilerle uğraşırken, gradyanlar zamanla küçülebilir, bu da öğrenmeyi engelleyebilir ve RNN'lerin eğitilmesini zorlaştırabilir.
- Gradyan patlaması sorunu: Buna karşılık, gradyanlar zaman adımlarında çok büyük hale gelebilir ve bu da eğitim sırasında istenmeyen davranışlara yol açabilir.

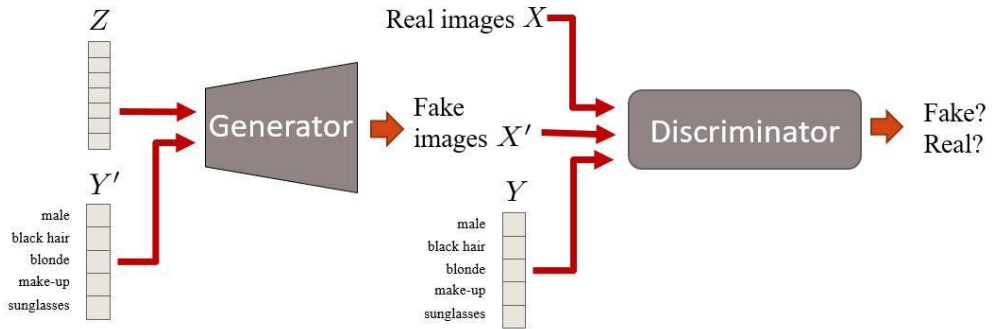
Bu bağlamda, bu sorunlara çözüm sağlamak amacıyla LSTM (Uzun Kısa Süreli Bellek) ve GRU (Gated Recurrent Unit) başta olmak üzere RNN'lerin geliştirilmiş türleri geliştirilmiştir. LSTM ve GRU gibi hücre yapıları, RNN'lerin uzun vadeli bağlamı daha etkili bir şekilde öğrenmesine yardımcı olur (Kutlu, 2019).

2.8.4 Üretken Çekişmeli Ağlar (GAN'lar)

Üretken Çekişmeli Ağlar (GAN), 2014 yılında Ian Goodfellow ve ekibi tarafından tanımlanan bir tür derin öğrenme modelidir. GAN, iki temel bileşeni içeren yenilikçi bir yapıya sahiptir. Bunlar jeneratör ve ayırıcıdır (Çelik ve Talu, 2020).

- Jeneratörler: Jeneratörler, rastgele gürültüden gerçekçi veriler üretmeye çalışan sinir ağlarıdır. GAN'lar özellikle bir jeneratör kullanarak rastgele piksellerden gerçekçi görüntüler oluşturmaya çalıştıkları görüntü oluşturma görevlerinde kullanılır (Çelik ve Talu, 2020).
- Ayırıcı: Ayırıcı, bir jeneratör tarafından üretilen gerçek ve sahte veriler arasındaki farkı belirlemeye çalışan bir sinir ağıdır. Bu ağ, gerçek ve sahte veriler arasında ayırım yapmayı öğrenerek GAN'ın merkezindeki bu rekabetçi dinamikleri etkinleştirir (Çelik ve Talu, 2020).

GAN, bu iki bileşen arasında oyun teorisine dayalı bir derin öğrenme modelidir. Jeneratörler ve ayırıcılar birbiriyle yarışıyor. Jeneratör, ayırıcıyı yanıltmak için yeterli gerçekçi veri üretmeye çalışırken, ayırıcı gerçek ve sahte veriler arasındaki farkı en iyi şekilde anlatmaya çalışır. Bu yarışma, GAN'ın normal çalışmasını sağlamak için hem jeneratörlerin hem de ayırıcıların geliştirilmesine katkıda bulunur (Çelik ve Talu, 2020).



Şekil 2.7: Üretken çekişmeli ağ yapısı

GAN'ın temel amacı gerçekçi ve özgün veriler üretmektir. Bu modeller özellikle aşağıdaki alanlarda başarıyla kullanılmıştır:

- Görüntü sentezi: GAN, yüksek kaliteli görüntüler oluşturmak için kullanılan bir derin öğrenme modelidir. Bunlar özellikle deepfake teknolojisi ve sanat

üretimi gibi alanlarda yaygın olarak kullanılıyor. Deepfake teknolojisi, gerçek gibi görünen sahte içerik oluşturmak amacıyla videoları ve görüntüleri değiştirmek için GAN'ları kullanır. Sanat üretimi bağlamında GAN'lar resim, müzik ve diğer sanat eserleri yaratarak yaratıcılığı teşvik etmek için kullanılabilir (Çelik ve Talu, 2020).

- Veri Arttırma: GAN'lar, sınırlı veri kümeleriyle örnek çeşitliliğini artırmak için kullanılabilir. Bu modeller, gerçekçi veri örnekleri oluşturarak, özellikle eğitim verileri sınırlı olduğunda öğrenme performansını iyileştirebilir (Çelik ve Talu, 2020).
- Stil Transferi: GAN, bir görüntünün stilini diğerine aktarmak için kullanılabilir. Bu modeller, özellikle sanat ve görüntü işleme uygulamalarında, bir görüntünün estetik özelliklerini diğerine başarılı bir şekilde entegre edebilme yetenekleriyle öne çıkıyor (Çelik ve Talu, 2020).
- Süper çözünürlük: GAN, düşük çözünürlüklü görüntüleri yüksek çözünürlüklü görüntülere dönüştürmek için kullanılabilir. Bu modeller, özellikle görüntü iyileştirme ve restorasyon uygulamalarında düşük kaliteli görüntüleri daha kaliteli ve daha ayrıntılı hale getirme yetenekleriyle dikkat çekiyor (Çelik ve Talu, 2020).
- Yaratıcı İçerik Üretimi: GAN'ları sanat ve tasarımda kullanmak benzersiz içerik oluşturmanıza yardımcı olabilir. Bu modeller, özellikle yaratıcı ve estetik ifade için kullanıldığında orijinal sanat eserleri ve tasarımlar yaratmada etkilidir (Çelik ve Talu, 2020).

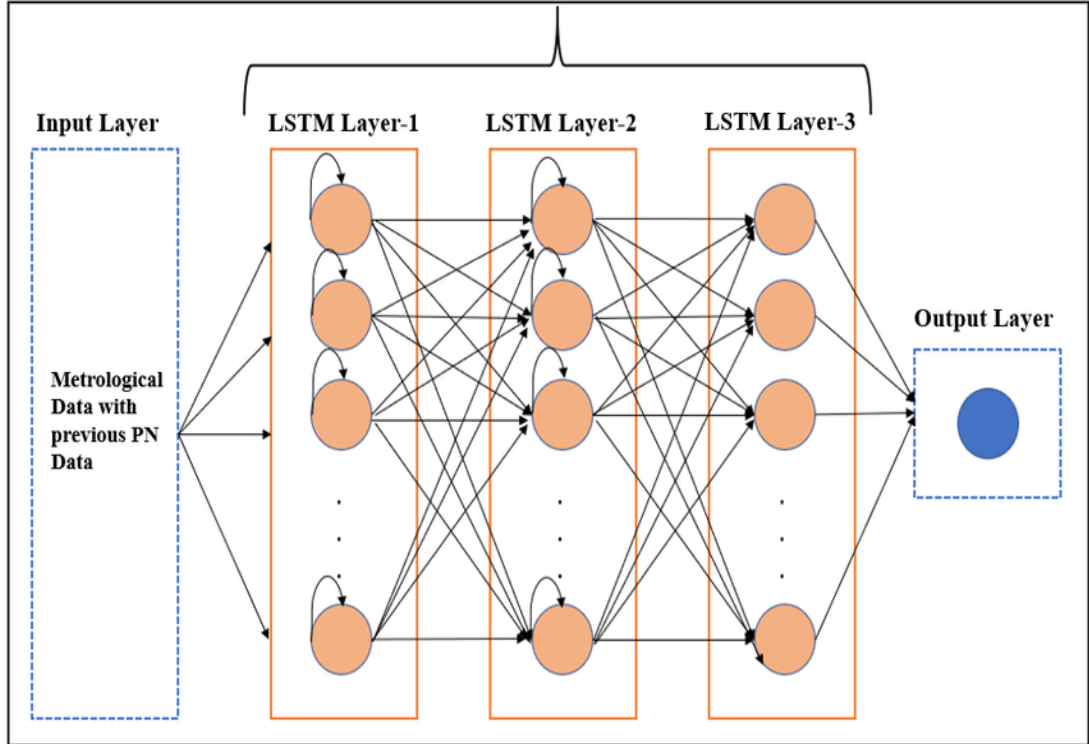
GAN'lar karmaşık ve dengeli bir eğitim sürecine sahiptir ve dikkatli bir şekilde uygulanmalıdır. Bu modellerin başarılı bir şekilde eğitilmesi, oluşturucu ve ayırıcı arasındaki rekabeti optimum seviyede tutmak için birçok parametrenin dengelenmesini gerektirir.

2.8.5 Derin Sinir Ağları (DNN'ler)

Uzun kısa süreli bellek ağları (LSTM'ler), tekrarlayan sinir ağları (RNN'ler) kategorisine ait özel bir yapay sinir ağı türüdür. RNN'ler zaman serileri, doğal dil işleme ve makine çevirisi gibi uygulamalarda yaygın olarak kullanılan yapay sinir ağlarıdır ve bu uygulamalarda zamana bağlı verilerin işlenmesinde etkilidir. LSTM,

uzun vadeli bağlamı daha etkili bir şekilde öğrenebilen hücresel bir yapıya sahiptir ve bu özellik, geleneksel RNN'lerin karşılaştığı sorunlara çözüm sağlar (Ser ve Bati, 2019).

LSTM, her hücrenin üç temel kapı içerdiği hücrelerden oluşur: unutmaya kapısı, giriş kapısı ve çıkış kapısı. Bu kapılar, hücrenin önceki durumunu unutmak, yeni bilgi eklemek ve güncellenmiş bir durum oluşturmak için etkileşime girer. Bu mekanizma özellikle uzun vadeli bağlantıları sürdürme ve öğrenme yeteneğiyle karakterize edilir. LSTM, geleneksel RNN'lerin zamanla azalan gradyan problemini ele alarak ve uzun vadeli bağlamları daha etkili bir şekilde ele alarak birçok uygulama alanında başarıyla kullanılmıştır. Tekrarlayan Sinir Ağı (RNN) ailesinin bir alt kategorisi olan LSTM, RNN'lerdeki kısa süreli hafıza sorunlarına etkili bir çözüm sağlamak üzere özel olarak tasarlanmıştır. Standart RNN'ler zamanla değişen verileri işlerken önceki girdileri unutmaya eğilimindedir, bu nedenle uzun vadeli bağlamı düzgün bir şekilde sürdürmek zor olabilir (Kutlu, 2019).



Şekil 2.8: Derin sinir ağları-LSTM

LSTM, bu kısa süreli hafıza sorunlarını gidermek için özel hücre yapıları içerir. Bu hücrelerde unutmaya kapıları, giriş kapıları ve çıkış kapıları adı verilen üç temel kapı bulunur. Her kapı, hücrenin önceki durumunu unutmaya, yeni bilgiler ekleme ve

güncellenmiş durumu üretme görevini etkin bir şekilde yerine getirir. Bu şekilde LSTM önceki girdileri daha etkili bir şekilde hatırlayabilir ve uzun vadeli bağlamı başarılı bir şekilde koruyabilir (Kutlu, 2019).

LSTM bu sorunu üç kapı kullanarak çözer:

- Giriş kapısı: Giriş kapısı, yeni girişlerin bellek hücresine girmesine izin verir.
- Unutma Kapısı: Unutma Kapısı, hafıza hücrelerindeki mevcut bilgileri silmenizi sağlar.
- Çıkış Kapısı: çıkış kapısı, bellek hücresinin çıkışını kontrol eder.

Bu kapılar LSTM'nin önceki girdileri hatırlamasına ve uzun vadeli kalıpları tanımasına olanak tanır. Bu durum, onu özellikle doğal dil işleme ve makine çevirisi gibi uygulamalar için değerli kılar (Kutlu, 2019).

LSTM, unutma ve giriş/çıkış kapıları aracılığıyla önceki girişleri hatırlayabilir, böylece uzun vadeli bağlamı başarıyla koruyabilir. Bu kapılar hücrelerin önceki durumlarını güncelleyerek geçmiş bilgileri hatırlamasını sağlar. Bu özellikler LSTM'yi doğal dil işleme ve makine çevirisi gibi uygulamalar için son derece değerli kılmaktadır. Bu uygulamalarda dilin yapısının uzun vadeli bağlamını anlamak ve manipüle etmek önemlidir ve LSTM bu ihtiyaçları karşılamak için etkili bir çözüm sunar (Salouhou, 2019).

LSTM'nin bazı uygulamaları şunlardır:

LSTM, çeşitli doğal dil işleme görevlerinde etkili bir şekilde kullanılabilir. Metin sınıflandırma, metin özetleme, dil çevirisi ve soruların yanıtlanması. Bu özel sinir ağı türü, dilin yapısını ve uzun vadeli ilişkilerini başarılı bir şekilde anlama yeteneği ile karakterize edilir. Belirli kategorilere ait metinleri tanımlamak ve sınıflandırmak için metin sınıflandırma görevlerinde kullanılır. Uzun metinleri işledikten sonra orijinal özetler oluşturmak için metin özetleme görevlerinde kullanılabilir. Ek olarak, dil çevirisi görevleri, metni bir dilden diğerine çevirmek için doğal dili etkili bir şekilde kullanabilir ve soru yanıtlama görevleri, sorulara uygun yanıtları bulmak için doğal dili etkili bir şekilde kullanabilir (Salouhou, 2019).

Makine çevirisi alanında, özellikle uzun ve karmaşık metinlerin çevirisinde LSTM etkin bir şekilde kullanılabilir. Bu özel sinir ağı türü, bir dildeki metni anlama ve anlamını koruyarak hedef dilde eşdeğer metin üretme yeteneği ile karakterize edilir. Bu sayede farklı dillerdeki metinler arasında anlamsal tutarlılık sağlanarak yüksek kalitede çeviriler elde etmek mümkündür. Bu özellik, dil çevirisi zorluklarının üstesinden gelme ve içeriği farklı dillerde başarılı bir şekilde iletme potansiyeli sunar (Salouhou, 2019).

Zaman serisi analizi bağlamında LSTM özellikle finansal, hava durumu veya sosyal medya verileri gibi dinamik zaman serileri analiz edilirken önerilir. Bu özel sinir ağı türü, geçmiş ve şimdiki zaman adımlarından gelen bilgileri dikkate alarak gelecekteki değerleri tahmin etme yeteneğine sahiptir. LSTM' nin finansal verilerdeki gelecekteki eğilimleri tahmin etmek, hava durumu değişikliklerini tahmin etmek ve sosyal medya verilerindeki eğilimleri analiz etmek gibi uygulamalarda zaman serisi analizi için etkili bir araç olduğu kanıtlanmıştır (Salouhou, 2019). Potansiyel uygulamaları tıp, üretim ve diğer birçok alanda önemli faydalar sağlayabilir.

2.8.6 Doğrusal Olmayan Bağlılı Mimariler (Transformers)

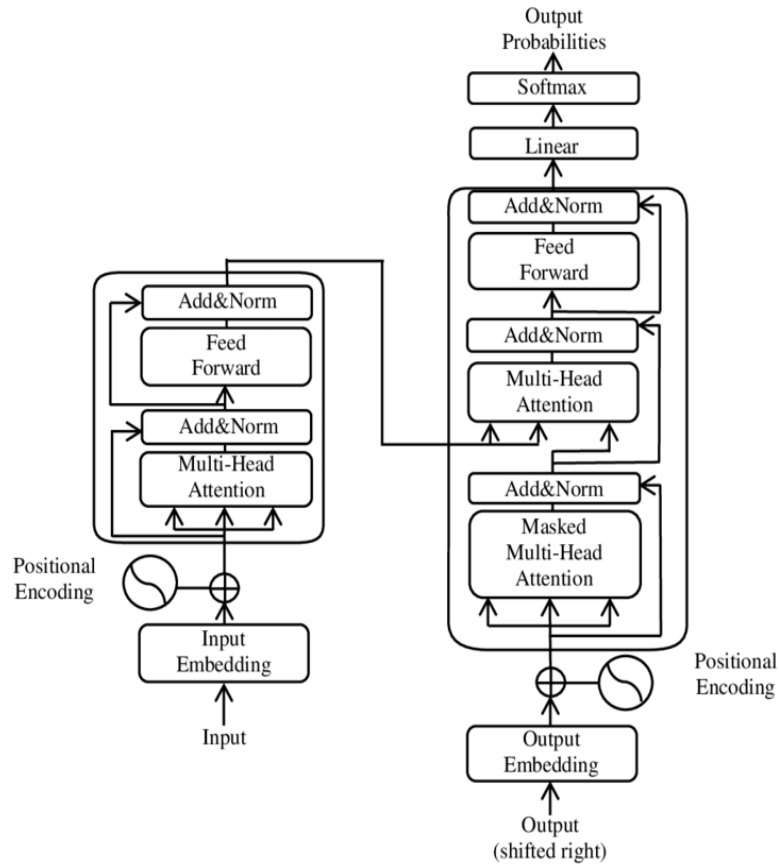
Doğal dil işleme (NLP) ve makine çevirisi (MT) gibi alanlarda kullanılan doğrusal olmayan bağlam modelleri (transformatörler), geleneksel tekrarlayan sinir ağlarından (RNN'ler) veya dikkat modellerinden farklı bir yapay sinir ağı türünü temsil eder. Transformatörler, giriş ve çıkış arasında doğrudan bir bağlantı oluşturarak uzun mesafeli bağlantıları daha iyi anlayabilir. Bu özellik, özellikle dil anlama ve çeviri görevlerinde daha yüksek performans ve daha zengin bağlam anlayışı sağlar (Şeker vd., 2017).

Transformatör iki ana bileşenden oluşur. Bu iki ana bileşen eğrileştirmeli dikkat ve kişisel dikkattir.

Eğrileştirmeli dikkat, bileşen, giriş ve çıkış arasında doğrudan bağlantı oluşturan bir dikkat mekanizmasıdır. Bu özellik, transformer mimarisinin uzun vadeli korelasyonları daha etkili bir şekilde anlamasını sağlar. Bu mekanizma, dil anlama, makine çevirisi ve benzeri görevlerde daha zengin, bağlama daha duyarlı performans sağlamayı amaçlamaktadır (Şeker vd., 2017).

Kişisel dikkat, bir girdi dizisindeki öğeler arasındaki ilişkileri belirlemek için kullanılan bir dikkat mekanizmasıdır. Bu bileşen, Transformer mimarisinin giriş dizisi içindeki anlamsal ilişkileri daha iyi anlamasını sağlar. Bu mekanizma, dil işleme, makine çevirisi ve benzeri görevlerde daha zengin, bağlama daha duyarlı performans sağlamayı amaçlamaktadır(Şeker vd., 2017).

Transformers, geleneksel tekrarlayan sinir ağlarına (RNN'ler) kıyasla doğal dil işleme (NLP) ve makine çevirisi (MT) görevlerinde kullanılır ve dikkat modellerinden daha etkilidir. Bunun nedeni transformatörlerin uzun mesafeli bağlantıları daha iyi anlayabilmesidir. Bu mekanizma, özellikle dile dayalı görevlerde daha geniş bir bağlam anlayışı sağlayarak, geleneksel yöntemlere kıyasla daha iyi sonuçlar elde etmemizi sağlar (Şeker vd., 2017).



Şekil 2.9: Transformers yapısı

Transformers uygulamaları önemli alanlarda görülmektedir. Doğal dil işleme alanında Transformers, metin sınıflandırma, metin özetleme, dil çevirisi ve soru cevaplama gibi çeşitli görevlerde etkin bir şekilde kullanılabilir. Bu özel yapay sinir ağı modeli, metin

verileri içindeki karmaşık ilişkileri anlama ve çeşitli dil işleme görevlerini başarıyla gerçekleştirme becerisiyle karakterize edilir. Makine çevirisi alanında, bir dilde yazılan metni anlamak ve başka bir dile doğru ve anlamlı bir şekilde çevirmek için transformatörler etkin bir şekilde kullanılabilir. Geleneksel yöntemlerden farklı olarak bu özel yapay sinir ağı modeli, önceki metnin bağlamını dikkate alarak ve uzak mesafe bağlantılarını daha etkin bir şekilde anlayarak çeviri kalitesini artırabilmektedir. Konuşma dili işleme alanında dönüştürücüler, konuşma tanıma ve anlamsal yorumlama gibi görevler için etkin bir şekilde kullanılabilir. Bu yapay sinir ağı modeli özellikle konuşulan dili anlamada, ses içeriğini çıkarmada ve doğal dil işleme görevlerini gerçekleştirmede iyi performans gösteriyor. Bu gelişmiş sinir ağı modelinin potansiyel uygulamaları sağlık, finans, üretim ve diğer birçok sektöre büyük fayda sağlayabilir.

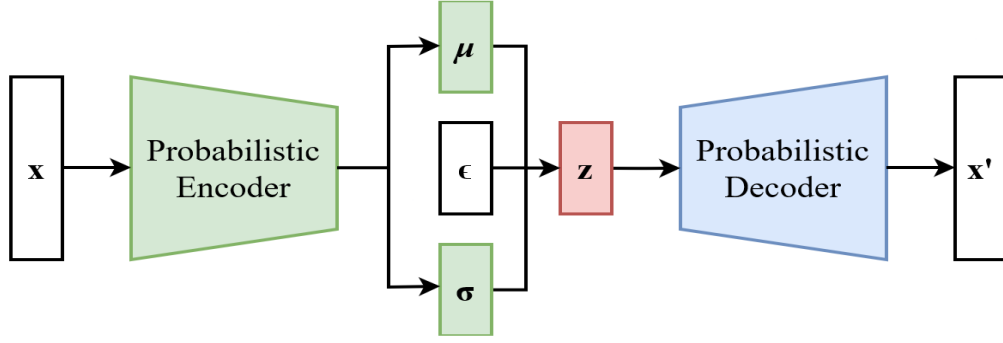
Transformers bazı avantajlara sahiptir. Bunlar, uzun menzilli korelasyonların daha iyi anlaşılması, daha hızlı ve daha verimli eğitim, daha az sayıda parametre olarak söyleyebiliriz.

Transformers dezavantajları ise daha karmaşık mimari yapısının olması ve daha fazla bilgi işlem kaynağı gerektirmesi olarak söylenebilir.

Transformers'ın geleceği, doğal dil işleme (NLP) ve makine çevirisi (MT) gibi alanlarda hızla standart haline geliyor. Bu sinir ağı modelinin daha da geliştirilmesi, bu alanlarda daha yüksek performans düzeylerine ulaşmak için büyük bir potansiyel sunabilir.

2.8.7 Varyasyonel Otoenkoder (Variational Autoencoder-VAE)

Değişken otomatik kodlayıcı (VAE), bir veri kümesini etkili bir şekilde sıkıştırmak ve bu sıkıştırılmış temsili orijinal verileri yeniden oluşturmak için kullanmak üzere tasarlanmış bir yapay sinir ağı modelidir. VAE, doğal dil işleme, bilgisayarlı görme ve makine öğrenmesi gibi uygulamalar başta olmak üzere çeşitli alanlarda yaygın olarak kullanılmaktadır (Bölük ve Cingiz, 2022).



Şekil 2.10: Varyasyonel otoenkoder yapısı

VAE iki ana bileşenden oluşur. Bunlar enkodeliyici ve kod çözücüdür. Enkodeliyici, giriş verilerini gizli bir gösterime sıkıştırarak temsil eder. Bu gizli temsil, verilerin önemli özelliklerini sayısal olarak temsil eden bir bilgi topluluğudur. Kod çözücü, orijinal verileri yeniden oluşturmak için gizli gösterimleri kullanır (Bölük ve Cingiz, 2022).

VAE, gizli gösterimler oluşturmak için varyasyonel Bayes modellerini kullanır. Varyasyonel Bayes modelleri, gizli temsillerin olasılık dağılımını tanımlamak için bir dizi parametre kullanır. Bu parametreler eğitim süreci sırasında veri seti içerisinde öğrenilir (Bölük ve Cingiz, 2022).

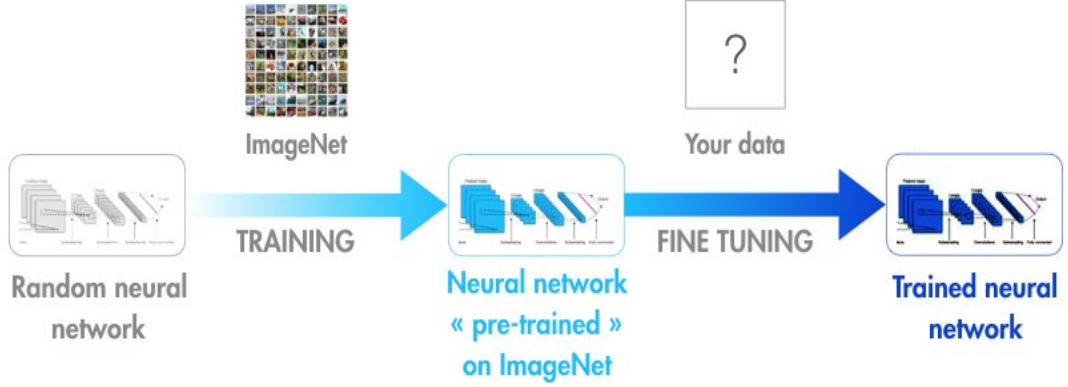
VAE'nin avantajları; VAE, resim, ses ve metin gibi farklı veri türlerini işlemek için kullanılabilir. Verilerdeki gizli kalıpları ortaya çıkarmak için kullanılabilir. Yeni veri oluşturmak için kullanılabilir.

VAE'nin dezavantajları ise eğitim karmaşık ve zaman alıcı olabilir. Model performansı eğitim verilerinin kalitesine bağlıdır.

VAE çeşitli uygulamalarda kullanılır. Doğal dil işleme alanında metni özetlemek, dilleri tercüme etmek ve soruları yanıtlamak için kullanılabilir. Bilgisayarla görme alanında görüntü sınıflandırma, nesne tanıma ve yüz tanıma için kullanılabilir. Makine öğrenimi yeni veriler oluşturmak, verilerdeki gizli kalıpları keşfetmek ve modelleri eğitmek için kullanılabilir.

2.8.8 Transfer Öğrenme Mimarileri

Transfer öğrenimi, bir modelin bir görevde öğrendiği becerileri benzer görevlerde daha az veri ve daha az eğitim süresiyle yeniden kullanmasıdır. Bu yaklaşım, modelin ilk görevde edindiği genelleme becerilerini kullanmayı amaçlamaktadır (Foto, 2022).



Şekil 2.11: Transfer öğrenme mimarisi

Öğrenmeyi transfer etmek için iki ana yaklaşım vardır:

- **Özellik çıkarma:** Bu yöntemde, önceden eğitilmiş bir modelin ağırlıkları kaynak görevden hedef göreve aktarılır. Bu, yeni modelin kaynak görev için önemli olan özellikleri baştan öğrenmesini sağlayarak hedef görevde daha etkili performansa olanak tanır (Foto, 2022).
- **İnce ayar:** Bu yöntemde, önceden eğitilmiş bir model, hedef görev için yeniden eğitilir. Böylece, yeni modelin yalnızca kaynak görevdeki bilgileri değil aynı zamanda hedef görevdeki yeni bilgileri de öğrenmesine olanak tanır (Foto, 2022).

Transfer öğreniminin birçok avantajı vardır. Transfer öğrenimi, yeni görevler için potansiyel olarak daha az veri gerektirebilir; bu, özellikle veri toplamanın zor veya pahalı olduğu durumlarda faydalıdır. Transfer öğrenimi, yeni görevler için eğitim süresini azaltabilir. Bu özellikle eğitim süresinin sınırlı olduğu durumlarda faydalıdır. Transfer öğrenimi, yeni görevlerdeki performansı artırır. Bu yaklaşım, özellikle hedef görev kaynak görevden farklı olduğunda daha da etkilidir.

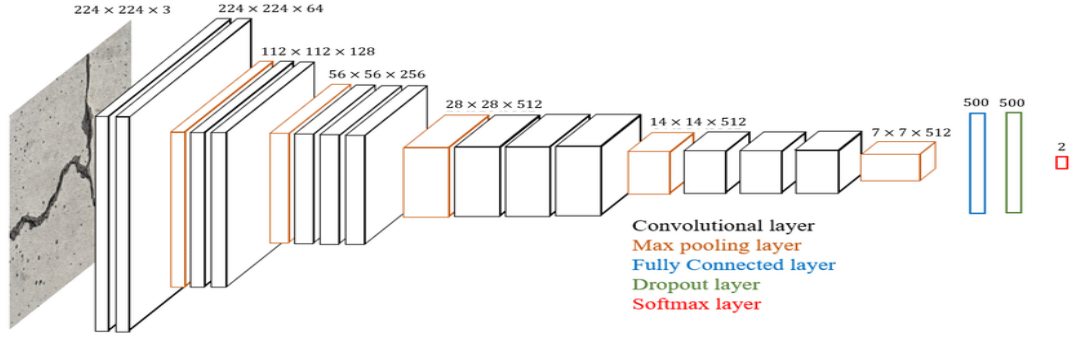
Transfer öğrenimi çeşitli uygulamalarda kullanılır. Bilgisayarlı görme alanında görüntü sınıflandırma, nesne tanıma ve yüz tanıma gibi görevlerde transfer öğrenmeden yararlanır. Doğal dil işleme bağlamında transfer öğrenme, metni sınıflandırma, dilleri çevirme ve soruları yanıtlama gibi görevlerde etkilidir. Makine çevirisi alanında, metni bir dilden diğerine çevirmek için transfer öğrenimi kullanılır. Tıp alanında transfer öğrenme, örneğin hastalıkları teşhis etmek ve tedavi geliştirmek için kullanılabilir. +Finansta transfer öğrenimi, riskleri tahmin etmek ve yatırım fırsatlarını belirlemek için etkili bir yol olarak kullanılıyor.

2.9 Derin Öğrenme Modelleri

2.9.1 VGGNet

VGGNet, Oxford Visual Geometry Group (VGG) tarafından 2014 yılında geliştirilen ve Imagenet görüntü sınıflandırma yarışmasında birincilik ve ikincilik kazanan bir derin öğrenme modelidir. Modelin mimarisi, ardışık evrişim katmanlarına dayanan basitliği ile karakterize edilir. Basit yapısına rağmen VGGNet'in benzer modern modellere göre yüksek performansı, derin öğrenme alanında önemli bir kilometre taşı olarak kabul ediliyor (Aktürk ve Serbest, 2022).

VGGNet derin bir mimariye sahiptir. Genellikle 16 veya 19 katmandan oluşan derin bir mimariye sahiptir. Bu katmanların çoğu evrişimli katmanlardan oluşur. VGGNet, evrişimli katmanlarda genellikle 3x3 filtre boyutunu tercih eder. Bu seçim, modelin ayrıntıları etkili bir şekilde yakalamasına ve hesaplama verimliliğini artırmasına olanak tanır. VGGNet tipik olarak bir evrişimsel katman ve ardından bir maksimum havuzlama katmanı kullanır. Bu yaklaşım, modelin mekansal değişime karşı daha dayanıklı olmasına yardımcı olur. Tamamen bağlantılı katmanlar: Derin evrişimli katmanlardan sonra model, sonuçları sınıflandırmak için bir dizi tamamen bağlantılı katman kullanır (Kırca, 2021).



Şekil 2.12: VggNet yapısı

VGGNet'in bazı avantajları vardır. Imagenet yarışmasında birincilik ve ikincilik kazanarak görüntü sınıflandırmadaki yüksek performansını tescillemiştir. VGGNet'in mimarisi basit ve anlaşılması kolaydır, modelin yorumlanmasını ve hata ayıklamasını kolaylaştırır. Önceden eğitilmiş bir model olarak VGGNet, çeşitli görüntü sınıflandırma ve hedef tespit görevleri için kullanılabilir.

VGGNet'in dezavantajları ise karmaşık mimarisi, eğitim ve tahmin aşamaları sırasında hesaplama karmaşıklığını artıran çok sayıda katman ve parametreye sahiptir. VGGNet, son teknoloji ürünü, daha karmaşık derin öğrenme modellerinden daha iyi performans gösteriyor. Mimarinin karmaşıklığından dolayı, VGGNet kararlarının anlaşılması ve yorumlanması zor olabilir.

VGGNet, basit mimarisine rağmen yüksek performans elde eden ve derin öğrenme araştırmalarına önemli katkı sağlayan, derin öğrenme alanında önemli bir mirastır. Şu anda performans açısından zirvede olmasa da derin öğrenme gelişiminde modellerin rolünün ve basitlik ile performans arasındaki ilişkinin vurgulanması açısından önemlidir (Kırca, 2021).

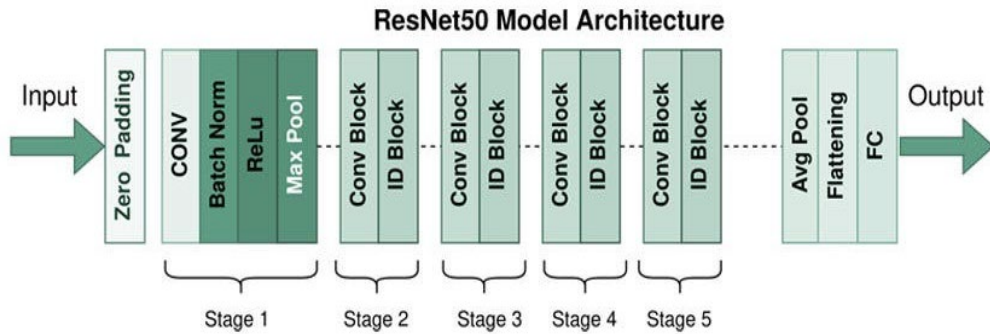
VGGNet, derin öğrenmede yaygın olarak kullanılan bir model olmaya devam etmekte ve mimarisi birçok yeni modele ilham kaynağı olmaktadır. Sadeliği ve anlaşılabilirliği, onu yeni derin öğrenme tekniklerini öğrenmek için mükemmel bir başlangıç noktası haline getirmektedir (Kırca, 2021).

VGG16, görüntü tanımayla yönelik klasik bir derin öğrenme mimarisidir. VGG16, Oxford Visual Geometry Group (VGG) tarafından 2014 yılında geliştirilmiş ve ImageNet yarışmasında birincilik ve ikincilik kazanarak büyük bir başarı elde etmiştir.

Mimarisi, ardışık evrişim katmanlarına dayanan sadeliğiyle dikkat çekiyor. Sadeliğine rağmen VGG16, benzer modern modellere kıyasla yüksek performans sunuyordu. Dolayısıyla bu, derin öğrenme alanında önemli bir kilometre taşı olarak değerlendiriliyor (Doğan ve Türkoğlu, 2018).

2.9.2 ResNet

ResNet (Artık Ağ), derin sinir ağlarının eğitimini kolaylaştıran ve performansını artıran bir derin öğrenme modelidir. Microsoft Research tarafından 2015 yılında Kaiming He ve diğerleri tarafından geliştirilmiştir. ResNet'in önemli bir özelliği, ağ derinleştikçe ortaya çıkan eğitim zorluklarını ele alan bir yapı olan "artık bloklar" veya "artık birimler" kullanmasıdır. Artık birimler, ağın önceki katmanlarından gelen bilgilerin doğrudan çıktı katmanına eklenmesine olanak tanıyarak ağın daha derin ve daha etkili bir şekilde öğrenmesine olanak tanır (Tan, 2019).



Şekil 2.13: ResNet50 yapısı

Standart evrişimli sinir ağları, her katmanın çıktısını doğrudan bir sonraki katmana ileterek çalışır. Ancak ResNet'in benzersiz bir özelliği, ağı derinleştirmek için artık blokları kullanmanıza olanak tanır. Geriye kalan bloklar, önceki katmandaki bilgilerin doğrudan bu katmanın çıktısına eklenmesini içerir. Bu mekanizma, orta katmanda karşılaşılan zorlukların üstesinden gelerek ağın daha derin ve etkili öğrenmesini sağlar. Artık blok, giriş verilerini çıkışa ekleyen bir "artık" (fark) bağlantısı içerir. Bu bağlantı, ağın yalnızca belirli bir katmanın çıktısını öğrenmesine değil aynı zamanda bu katmanın önceki bilgilerden nasıl farklı olduğunu da öğrenmesine olanak tanır. Bu yapı ağın derinleşmesini kolaylaştırır ve öğrenmeyi stabilize edici etkiye sahiptir.

Matematiksel olarak $F(x)+x$ şeklinde ifade edilebilir. Burada $F(x)$ katmanın öğrenmesi gereken dönüşümü, x ise giriş verilerini temsil eder (Tan, 2019).

ResNet temel olarak iki farklı blok türünden oluşur. Bu bloklar temel blok ve darboğaz bloktur. Şişirilmiş bloklar genellikle daha fazla parametreye sahiptir ve daha derin ağ yapıları için uygundur (Mutludoğan, 2020).

ResNet, çok derin ağ yapıları oluşturmak için özel olarak geliştirilmiştir. ResNet modelleri genellikle farklı derinliklere sahiptir: Bunlar 50, 101, 152 katmanlıdır (Mutludoğan, 2020).

Artık bağlantı, eğitim sürecini hızlandırır ve daha derin ağların etkili eğitimine olanak tanır. Artık bağlantılar, ağ performansını artırmak ve aşırı uyumu azaltmak için kullanılır ve ağların daha etkili bir şekilde optimize edilmesine olanak tanır (Mutludoğan, 2020).

ResNet 'in bazı avantajları sahiptir. ResNet, artık bağlantılar sayesinde daha derin bir mimariye olanak tanıyarak daha fazla bilgi öğrenme ve yüksek performans elde etme yeteneğini artırır. Artık bağlantılar daha verimli degrade akışı sağlayarak model eğitim sürecini hızlandırır. ResNet ve Imagenet gibi büyük veri kümelerinde yüksek doğruluk elde etmektedir (Mutludoğan, 2020).

ResNet 'in dezavantajları ise ResNet hâlâ hesaplama açısından yoğun bir modeldir ve etkili eğitim için yüksek donanım gereksinimleri gerektirebilir. Model karmaşıklığı: ResNet' in kalan bağlantıları, model mimarisine biraz karmaşıklık katabilir.

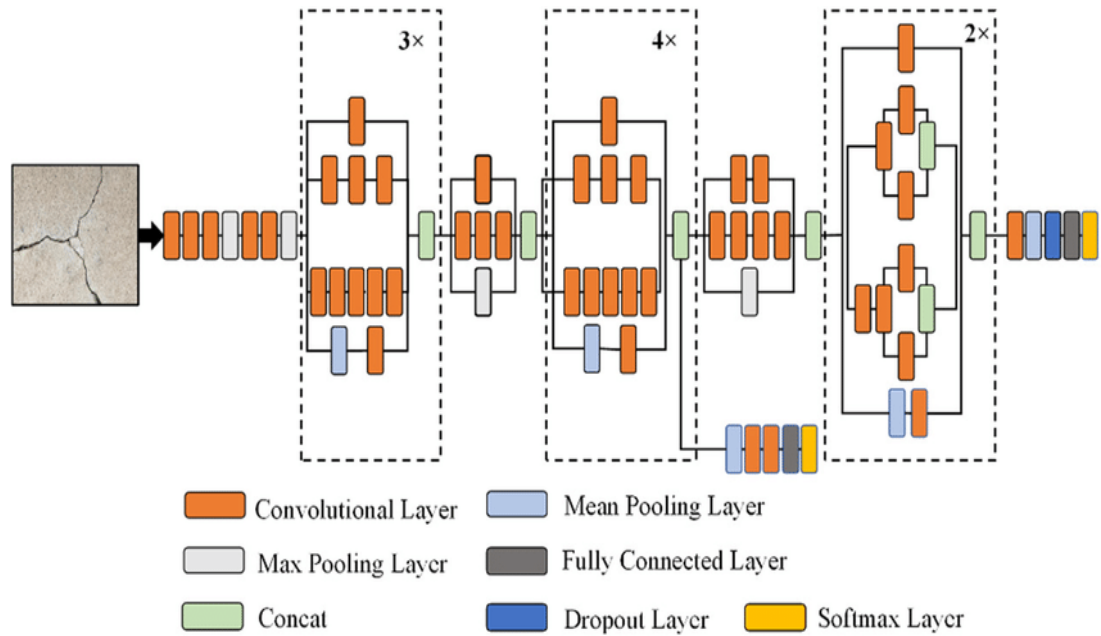
ResNet, derin öğrenme alanında çığır açan bir modeldir. Artık bağlantılar ve darboğaz blokları gibi yenilikler, daha derin ve daha güçlü modellerin geliştirilmesine öncülük etti. ResNet' in etkisi bugün hala birçok derin öğrenme modelinde görülebilmektedir.

ResNet birçok alanda kullanılır. Özellikle görüntü sınıflandırma görevleri için yaygın olarak tercih edilen bir derin öğrenme modelidir. ResNet hedef tespit görevlerinde de başarıyla kullanılabilir. ResNet, görüntülerdeki nesnelere ve bölgelere segmentlere ayırmak için etkili bir şekilde kullanılabilir.

Genel olarak ResNet, derin öğrenme tarihinde önemli bir yere sahiptir. Artık bağlantı gibi yenilikler, daha derin ve daha güçlü modellerin geliştirilmesine yol açarak derin öğrenme alanında devrim yarattı.

2.9.3 Inception

"Inception" Google tarafından geliştirilen bir derin öğrenme modelidir. Bu evrişimli sinir ağı (CNN) mimarisi, görüntü sınıflandırma ve tanıma görevlerinde kullanılmak üzere tasarlanmıştır ve resmi olarak "GoogLeNet" olarak bilinir ancak genellikle Başlangıç olarak bilinir. Bu model, 2014 ImageNet Büyük Ölçekli Görsel Tanıma Yarışmasında büyük bir başarı elde etti (İnik ve Ülker, 2017).



Şekil 2.14: Inception yapısı

Inception'ın ana özelliklerini incelediğimizde başlangıç modülü, paralel işlemler, 1x1 evrişimsel katman ve ek bağlantılar olarak dörde ayrılır.

Başlangıç modülünde başlangıç olarak, paralel çalışan birçok alt ağ içeren bir "Başlangıç Modülü" kullanır. Bu model, geleneksel evrişimli sinir ağı modellerinden farklı bir yaklaşımdır. Bu modüller farklı filtre boyutları ve özellik haritalarıyla çalışarak ağın özellikleri farklı ölçeklerde daha etkili bir şekilde öğrenmesine olanak tanır. Inception modülü, farklı filtre (çekirdek) boyutlarına sahip evrişimli katmanlar içerir. Bu modül, her filtre boyutunun belirli bir ölçekteki bilgileri daha etkili bir

şekilde işleyebileceğinden, farklı ölçeklerdeki işlevleri öğrenmenize olanak tanır (Kunt 2021).

Paralel evrişim işlemleri inception modülünde gerçekleştirilir. Bu modüller, ağın farklı filtre boyutlarını kullanarak farklı özellikleri ve ölçekleri öğrenmesine yardımcı olur (Kırca, 2021).

Başlangıç modülü ayrıca 1x1 boyutlu filtrelerle oluşturulmuş bir evrişimsel katmana sahiptir. Bu katmanlar özellik haritasının boyutunu küçültmek ve hesaplama verimliliğini artırmak için kullanılır (Kırca, 2021).

Ek bağlantılarda ise başlangıç modülü, alt ağlardan gelen çıktıları toplar ve birleştirir. Bu yöntem, daha zengin özellikleri öğrenmek için ağın farklı özelliklerini birleştirmeye yardımcı olur (Kunt, 2021).

Inception'ın bazı avantajları vardır. Inception, görüntü sınıflandırma görevlerinde şimdiye kadar elde edilen en yüksek doğruluğu sağladı. Yardımcı dallar, hesaplama miktarını azaltır ve modellerin daha az kaynakla çalışmasına olanak tanır. Inception'ın modüler yapısı diğer modellere uyum sağlamayı kolaylaştırır.

Inception'ın dezavantajları ise inception'ın mimarisi diğer modellere göre daha karmaşıktır. Bu durum, modelin yorumlanmasını ve hata ayıklamasını zorlaştırabilir. Inception hâlâ yoğun bilgi işlem içeren bir modeldir ve eğitim için güçlü donanım gerektirebilir.

Inception, derin öğrenme mimarilerinde önemli bir yer tutar. Inception'ın başarısı, boyutsal filtreleme ve modül karma gibi tekniklerin, derin öğrenme modellerinin performansını daha da artırmak için kullanılabileceğini gösteriyor. Inception'ın mimarisi günümüzde birçok derin öğrenme modeli için hala geçerlidir.

Inception'ın kullanım alanları incelendiğinde görüntü sınıflandırma görevleri için yaygın olarak kullanılan bir modeldir. Hedef tespit görevleri için de başarıyla kullanılabilir. Bir görüntünün stillerini diğerine aktarmak için Inception'ı kullanabilirsiniz.

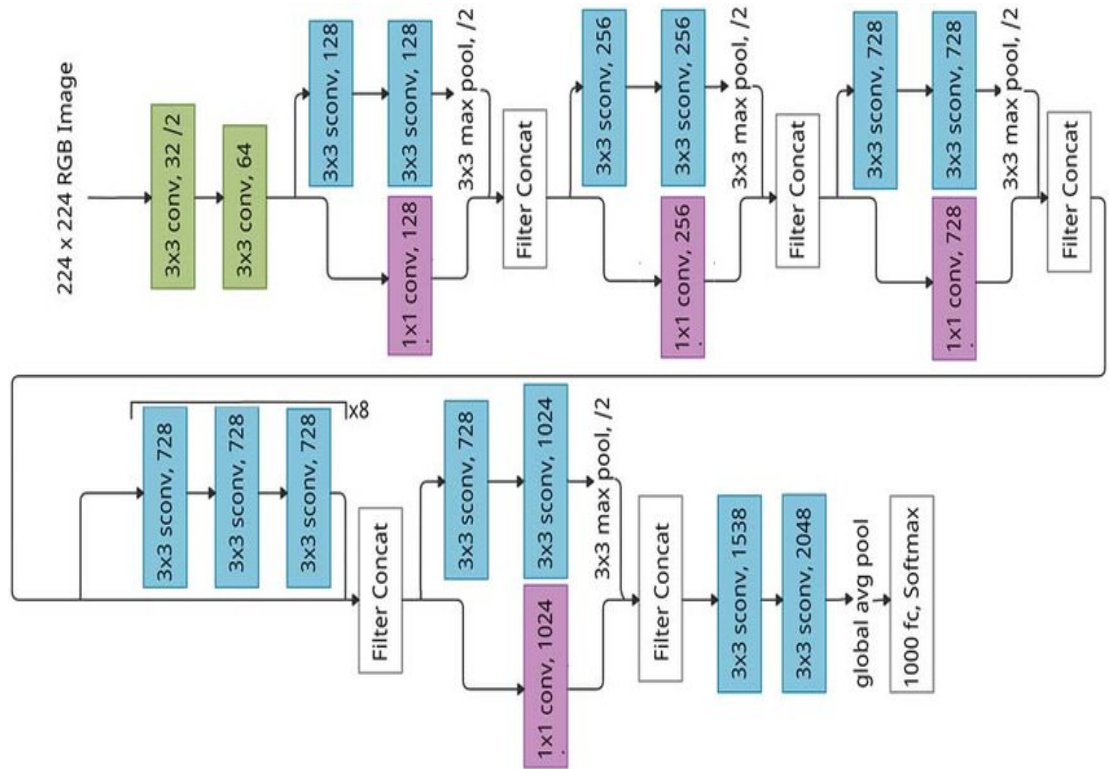
Sonuç olarak Inception, derin sinir ağlarının basit ve etkili bir şekilde çalışmasını amaçlayan bir tasarımla ortaya çıktı. Bu model, karmaşık görsel verileri etkili bir

şekilde işleyebilir ve özellikle bilgisayarlı görme ve derin öğrenme uygulamalarında olumlu performans sergileyebilir.

Inception, derin öğrenme tarihinde önemli bir yere sahiptir. Çok boyutlu filtrelerin kullanımı ve modüler karmaşıklık başta olmak üzere çeşitli yenilikler, derin öğrenme mimarilerinde yeni bir çağın kapısını açmıştır. Bu sayede yüksek doğruluk ve verimlilik elde edilerek Inception'ın katkısı derin öğrenme alanında önemli bir dönüm noktası haline geldi.

2.9.4 Xception

Xception (Genişletilmiş Algi), François Chollet tarafından önerilen derin öğrenme modelini ifade eder. Bu model, bilgisayarla görme görevlerinde kullanılmak üzere özel olarak tasarlanmış bir evrişimli sinir ağı (CNN) mimarisidir. Xception, Google'ın Başlangıç mimarisini temel alır ancak önemli değişiklikler içerir. Modelin adı "Extreme Inception" teriminin kısaltmasıdır ve François Chollet tarafından önerilen bu derin öğrenme modeli, özellikle bilgisayarlı görme uygulamalarında etkin performans sağlamak üzere tasarlanmıştır (Öner, 2023).



Şekil 2.15: Xception yapısı

Xception, derin evriřimli bir sinir ađı modelidir. Ancak Inception modelinden farklı olan önemli bir özellik, Xception'ın evriřimli katmanlarının daha derin ve daha basit bir yapıya sahip olacak şekilde tasarlanmış olmasıdır. Xception modelleri iç bağlantılara (derinlikle ayrılabilen kıvrımlara) dayalı bir yapıya sahiptir. Bu yaklaşım, özellik haritasının her kanalına önce bir evriřim işleminin uygulanması, ardından kanallar arasındaki etkileşimlerin 1x1 evriřim ile gerçekleştirilmesi anlamına gelir. Bu tasarım daha az parametre kullanır ve özellik öğrenme yeteneđini geliştirir Xception modelleri büyük veri kümeleri üzerinde eğitilir ve çeşitli veri kaynakları üzerinde güçlü performans gösterir. Xception modelleri özel olarak ortak görme görevleri için tasarlanmıştır ve sınıflandırma, nesne algılama ve segmentasyon gibi çeşitli görevleri etkili bir şekilde gerçekleştirebilir (Öner, 2023).

Xception'ın bazı avantajları vardır. Xception modeli yüksek doğruluk elde ederek ImageNet yarışmasında üçüncü sırada yer aldı. Bu başarı, modelin karmaşık görsel veri kümelerini etkili bir şekilde işleme yeteneđini vurguluyor. Derinliđiyle ayrılabilen evriřimlerin ve uzaysal derinliđiyle ayrılabilen evriřimlerin kullanılması, hesaplama maliyetlerini azaltır ve daha sınırlı kaynaklarla verimli çalışma yeteneđi sağlar. Xception modelinin mimarisi Inception'a göre basittir. Bu durum, kullanıcılara daha fazla anlayış ve kontrol sağlar ve modellerin yorumlanmasını ve hata ayıklamasını kolaylaştırır (Öner, 2023).

Xception'ın dezavantajları ise Xception modelleri performans açısından daha yeni, daha karmaşık modellerin gerisinde kalma eğilimindedir. Ancak belirli uygulama senaryolarında veya kaynakların kısıtlı olduđu durumlarda avantajlı olabilir. Xception modelleri hâlâ hesaplama açısından yođundur ve etkili eğitim için güçlü donanım gerektirebilir (Öner, 2023).

Xception modelleri, derin öğrenme alanında önemli bir mirası temsil ediyor ve derinlikle ayrılabilir evriřimler ve uzaysal derinlikle ayrılabilir evriřimler gibi yenilikler sayesinde hesaplama verimliliđinde dikkate deđer kazanımlar elde etti. İlerleme kaydediyoruz. Xception'ın mimarisi günümüzde birçok derin öğrenme modelinde hala ön plandadır (Öner, 2023).

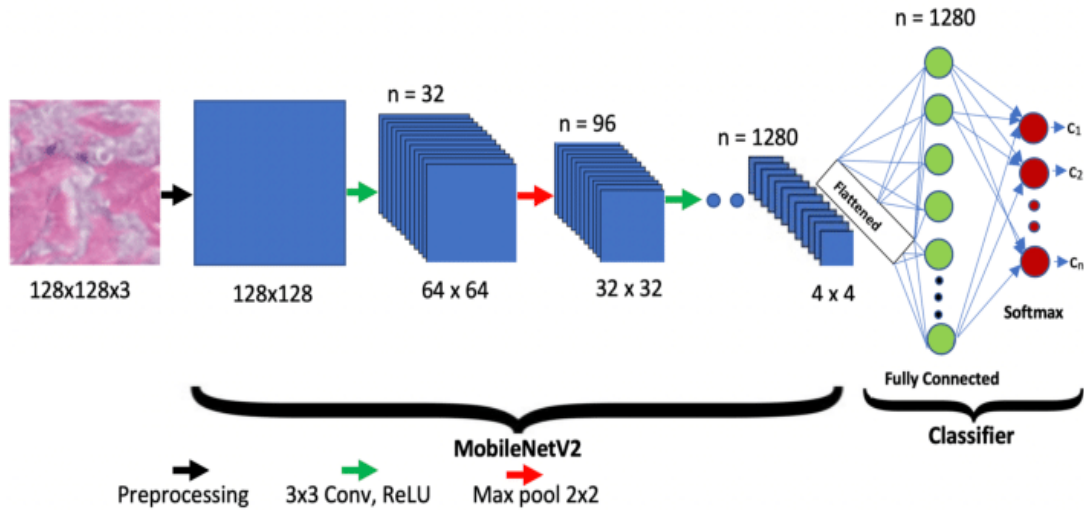
Xception'ın çeşitli kullanım alanları vardır. Xception modeli yaygın olarak kullanılan bir modeldir ve görüntü sınıflandırma görevlerinde etkili performans gösterir. Hedef

tespit görevlerinde başarıyla kullanılabilir bir modeldir. Bir görüntünün stillerini diğerine aktarmak için Xception'ı kullanabilirsiniz. Xception modelleri çeşitli derin öğrenme görevlerinde başarıyla kullanılabilir.

Bu model, karmaşık fonksiyonları etkili bir şekilde öğrenebildiği ve hesaplama maliyetini azaltabildiği için önerilmektedir.

2.9.5 MobileNet

Google tarafından geliştirilen MobileNet, hafif ve uygun maliyetli bir derin öğrenme modelidir. MobileNet, cep telefonları gibi yerel bilgi işlem gücü sınırlı olan cihazlar için özel olarak tasarlanmış olup, gömülü sistemlerde etkin bir şekilde kullanılması amaçlanmaktadır. Genellikle bilgisayarlı görme görevlerinde tercih edilen evrişimli sinir ağı (CNN) mimarilerinde önemli bir yer tutar (Çetiner, 2022).



Şekil 2.16: MobileNet yapısı

MobileNet modelleri, geleneksel evrişimli katmanlar yerine derinliğiyle ayrılabilir evrişimli katmanları tercih eder. Bu yaklaşım, evrişim süreci sırasında kanallar arasındaki bağlantıları azaltarak parametre kullanımını ve hesaplama maliyetini azaltır. MobileNet genel olarak küçük ve hafif modelleri ile dikkat çekmektedir. Bu, daha az bellek kullandığı ve daha hızlı çıktı üretebileceği anlamına gelir. MobileNet, ihtiyaçlarınıza göre daha fazla doğrulama ve ayarlama parametresi ile yapılandırılmıştır. Bu, farklı uygulamalara uyum sağlamasına ve farklı kaynak kısıtlamalarını ele almasına olanak tanır. MobileNet modelleri, transfer öğrenme

uygulamalarında kullanılmak üzere önceden eğitilmiş ağırlıklarla donatılabilir. Bu özellik, küçük veri kümeleri ve özel görevlerde etkili performans elde etmenize yardımcı olur (Somuncu, 2022).

MobileNet' in bazı avantajlara sahiptir. Mobil cihazlarda etkin kullanım için özel olarak geliştirildi. Küçük boyutu ve sınırlı sayıda parametresi hızlı sonuçlara olanak sağlar. Mobil cihazlar için optimize edilmiş bir model olmasına rağmen MobileNet, ImageNet yarışmasında yüksek doğruluk elde ederek iyi bir performans sergilemiştir. Görüntü sınıflandırma, nesne algılama ve yüz tanıma gibi çeşitli derin öğrenme görevlerinde etkili bir şekilde kullanılabilir (Eryılmaz ve Karacan, 2021).

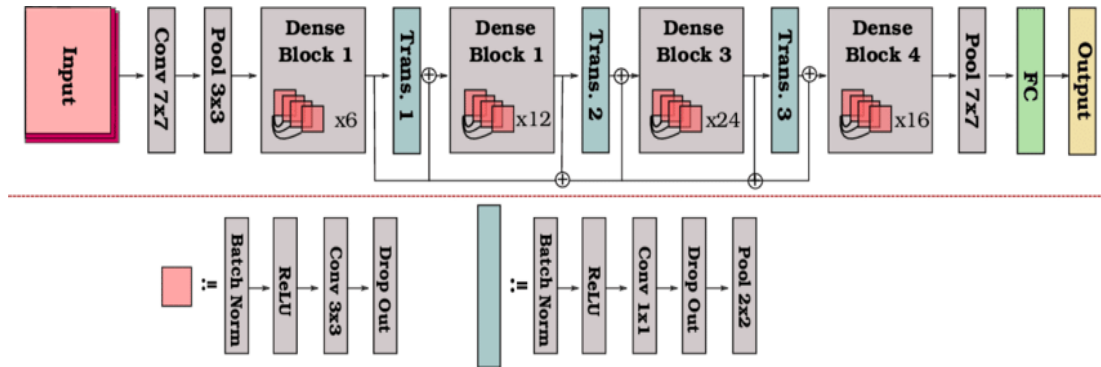
MobileNet'in dezavantajları ise daha yeni ve daha karmaşık modeller, MobileNet'e göre daha yüksek doğruluk elde edebilir. Bu durum, MobileNet'in performansının düşme eğiliminde olduğunu gösterir. MobileNet'in mimarisi diğer modellere göre daha karmaşık olabilir. Bu, modelin yorumlanmasını ve hata ayıklamasını zorlaştırabilir (Çetiner 2022).

MobileNet'in mobil cihazlarda derin öğrenme uygulamalarının geliştirilmesinde önemli bir etkisi vardır. Hafif ve hızlı yapısı, akıllı telefon ve akıllı kamera gibi cihazlarda gerçek zamanlı görüntü işleme uygulamalarına olanak tanıyarak mobil derin öğrenme uygulamalarının gelişmesine katkı sağlıyor. MobileNet kullanım alanları, akıllı telefonlarda nesne tanımlama ve yiyecek görüntüsü tanıma gibi uygulamalarda kullanılır. Artırılmış gerçeklik uygulamalarında ve otomatik odaklama sistemlerinde kullanılan MobileNet, akıllı telefon güvenlik uygulamalarında ve kimlik doğrulama sistemlerinde de kullanılır. Ayrıca tıbbi görüntü analizi ve sesli komut tanıma gibi çeşitli mobil derin öğrenme görevlerinde de kullanılmaktadır. MobileNet modelleri genellikle nesne tespiti, sınıflandırma ve segmentasyon gibi görevlerde etkin bir şekilde kullanılır. Oluşturulması kolay ve hızlı olduğundan özellikle mobil uygulamalar ve gömülü sistemler için popülerdir (Somuncu, 2022).

2.9.6 DenseNet

DenseNet (Densely Connected Convolutional Networks-Yoğun Bağlantılı Evrişimli Ağlar), derin öğrenme alanında kullanılan bir evrişimli sinir ağı (CNN) mimarisini temsil eder. Bu mimari, Gao Huang, Zhuang Liu ve Laurens van der Maaten tarafından 2017'de yayınlanan "Yoğun Bağlantılı Evrişimli Ağlar" başlıklı bir makalede ayrıntılı olarak açıklanmıştır (Foto, 2022).

DenseNet'in ana özelliği, her katmanın çıktısının doğrudan bir sonraki katmanın girişine eklenmesidir. Bu tasarım, her katmanın önceki tüm katmanlara sıkı bir şekilde bağlı olduğu anlamına gelir. Bağlantıların yüksek yoğunluğu bilgi akışını optimize eder ve öğrenme süreci sırasında oluşabilecek değişim kayıplarını etkili bir şekilde azaltır (Foto, 2022).



Şekil 2.17: DenseNet yapısı

DenseNet iki ana bileşenden oluşur. Bunlar yoğun geçişler ve birbirine sıkı bağlanmadır. Yoğun geçişler, her katman, önceki tüm katmanların çıktılarını girdi olarak alarak DenseNet mimarisinde yoğun bağlantılar oluşturur. Bu sayede her katman bir önceki katmandan gelen bilgiyi doğrudan kullanma olanağına sahip olur. Birbirine sıkı bağlanmada ise tüm katmanlar birbirine sıkı bağlanmıştır. Bu, DenseNet mimarisinde daha fazla bağlantıya ve bilgi aktarımına olanak tanır (Serin, 2020).

DenseNet bazı avantajlara sahiptir. DenseNet, daha güçlü ve geliştirilebilir modeller oluşturur. Daha az parametreyle etkin çalışır. Eğitim süreci daha hızlı tamamlanır. DenseNet'in dezavantajları ise eğitim süreci karmaşık bir yapıya sahip olabilir. Daha fazla bilgi işlem kaynağı gerekiyor (Serin, 2020).

DenseNet çeşitli uygulamalarda kullanılır. Genellikle görüntü sınıflandırma görevleri için önerilen modeldir. DenseNet, özellikle ImageNet gibi büyük veri kümeleri üzerinde eğitildiğinde insan düzeyinde performansa ulaşabilir. DenseNet modelleri nesne algılama görevlerinde de etkin bir şekilde kullanılmaktadır. Bu modeller görüntülerdeki nesnelerin tanımlanmasında başarıyla uygulanabilmektedir. DenseNet modeli, zaman serisi analizine başarıyla uygulandı. Bu modeller finansal, hava durumu ve sosyal medya verileri gibi zaman serisi verilerini analiz etmek için etkili araçlardır (Foto, 2022).

DenseNet, sınırlı veri kümeleri veya kaynaklara sahip uygulamalarda özellikle etkili bir şekilde kullanılabilir. Yoğun bağlantı yoluyla daha derin ve daha verimli öğrenme fırsatları sağlayarak önceki mimarileri aşma potansiyeline sahiptir.

2.10 Derin Öğrenme ile Savunma Sanayi Arasındaki İlişki

Derin öğrenme savunma sanayinin birçok alanında kullanılmaktadır. Bu alanlar şunlardır:

- **Güvenlik:** Derin öğrenme, güvenlik görevlerinde, özellikle de yüz tanıma, nesne tanıma ve şüpheli davranış tespiti gibi görevlerde yaygın olarak kullanılmaktadır. Örneğin havalimanlarında yolcuları tanımak, yasa dışı sınır geçişlerini tespit etmek, terör saldırılarını önlemek gibi amaçlarla derin öğrenme modelleri kullanılabilir.
- **Silah Sistemleri:** Silah sistemlerinde hedef tanıma, otomatik silah kontrolü, ateş destek sistemleri gibi görevlerde derin öğrenme etkin bir şekilde kullanılmaktadır. Örneğin savaş uçakları tarafından hedef tanıma, zırhlı araçlar için otomatik silah kontrolü, topçu desteği gibi amaçlarla derin öğrenme modelleri kullanılabilir.
- **Savunma Stratejileri:** Derin öğrenme, tehdit değerlendirmesi, risk analizi ve karar verme gibi savunma stratejilerini ve görevlerini geliştirmek için etkin bir şekilde kullanılır. Derin öğrenme modelleri, örneğin düşman kuvvetlerini değerlendirmek, saldırı olasılığını tahmin etmek ve savunma planlarını formüle etmek için kullanılabilir.



Şekil 2.18: Derin öğrenme ile nesne tespiti

Savunma sanayinde derin öğrenmenin spesifik uygulamaları şunları içerir:

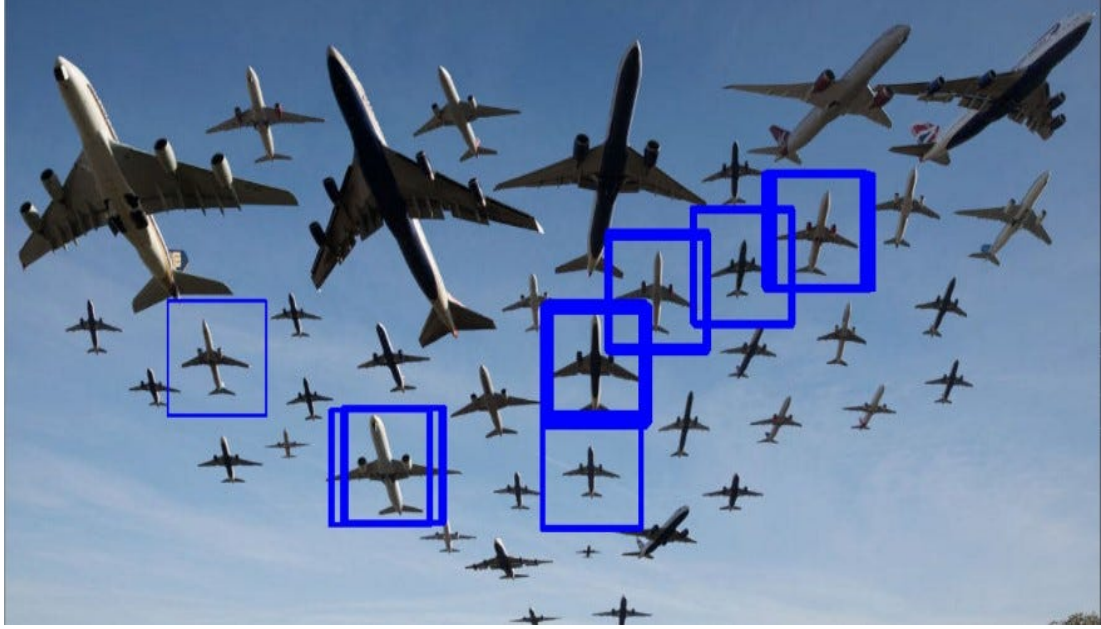
- **Yüz Tanıma:** Derin öğrenme modelleri, yüz tanıma görevlerinde büyük başarı elde etti. Bu modeller insanlar arasındaki benzerlikleri ve farklılıkları büyük bir doğrulukla tanıma yeteneğine sahiptir. Bu özellikler sayesinde derin öğrenme modelleri şunları yapabilir: Örneğin havalimanlarındaki yolcuları tespit etmek, yasadışı sınır geçişlerini tespit etmek, terör saldırılarını önlemek vb. amaçlarla etkin bir şekilde kullanılabilir.
- **Nesne algılama:** Derin öğrenme modelleri, nesne algılama görevlerinde de büyük başarı elde etti. Bu modeller görüntü ve videolardaki nesnelere tanıyıp sınıflandırabiliyor. Bu özellikler sayesinde derin öğrenme modelleri şunları yapabilir: Savaş alanındaki düşman kuvvetlerinin tespiti, trafik güvenliğinin artırılması, tıbbi görüntü analizinde kanser hücrelerinin tespit edilmesi gibi çeşitli alanlarda etkin bir şekilde kullanılabilir.
- **Şüpheli davranış tespiti:** Derin öğrenme modelleri, şüpheli davranış tespit görevleri için de etkili bir şekilde kullanılabilir. Bu modeller, insanların hareketlerini ve eylemlerini analiz ederek potansiyel tehditleri tespit edebiliyor. Bu bağlamda derin öğrenme modelleri havalimanlarındaki şüpheli yolcuları tespit etmek, yasa dışı sınır geçişlerini önlemek ve terör saldırılarını önlemek için kullanılabilir.
- **Hedef tespit:** Derin öğrenme modelleri, hedef tespit görevlerinde de büyük başarı elde etti. Bu modeller, görüntü ve videolardaki hedefleri başarılı bir şekilde tespit edip sınıflandırabilmektedir. Bu özellikler sayesinde derin

öğrenme modelleri şunları yapabilir: Savaş alanındaki düşman hedeflerini tespit etmek, topçu ateşini desteklemek ve silah sistemlerini otomatikleştirmek gibi çeşitli görevlerde etkin bir şekilde kullanılabilir.

- Otomatik silah kontrolü: Derin öğrenme modelleri otomatik silah kontrolü alanında da etkin bir şekilde kullanılabilir. Bu modeller, görüntü ve videolardaki hedefleri başarılı bir şekilde tespit ederek silah sistemlerini otomatik olarak kontrol edebilir. Bu bağlamda derin öğrenme modelleri, savaş alanındaki düşman hedeflerini tanımlamanıza ve onlara daha hızlı ve daha doğru bir şekilde saldırmanıza olanak tanır.
- Ateş destek sistemleri: Derin öğrenme modelleri, ateş destek sistemlerinde başarıyla kullanılabilir. Bu modeller, hedef konumları doğru şekilde tahmin ederek ateş destek sistemlerini otomatik olarak kontrol edebilir. Bu bağlamda derin öğrenme modelleri, Savaş alanında düşman hedeflerine daha etkili bir şekilde saldırmak için kullanılabilir.

Savunma sanayinde derin öğrenme uygulamaları hızla yaygınlaşıyor. Bu modeller savunma sistemlerinin verimliliğini ve etkinliğini önemli ölçüde artırma potansiyeline sahiptir.

2.11 Derin Öğrenme Modelleri ile Uydu Görüntülerinden Uçak Tiplerinin Tespiti için Yapılan Çalışmalar



Şekil 2.19: Uçak tespiti

Uydu görüntülerinden uçak türlerini tanımak için derin öğrenme teknikleri kullanılarak bazı çalışmalar yapılmıştır. Özellikle Google, uydu görüntülerini uçak türü gibi kategorilere ayırmak için derin öğrenme modelleri kullanılarak geliştirilmiş bir sistem kullanır. Sistem, uydu görüntülerini uçak tipi gibi kategorilere göre %100'ün üzerinde doğrulukla sınıflandırabiliyor. Facebook, uydu görüntülerindeki uçak hareketlerini takip etmek için derin öğrenme modelleri kullanılarak geliştirilmiş bir sistem kullanıyor. Sistem, uçak hareketlerini uydu görüntülerinde 500 m'nin üzerinde doğrulukla başarıyla takip etti. NASA, uydu görüntülerinden uçak türlerini tespit etmek için derin öğrenme modellerini kullanan bir sistem geliştirdi. Sistem, uydu görüntülerinden uçak tiplerini %50'nin üzerinde doğrulukla tespit edebiliyor.

Uydu görüntülerinden uçak tipinin tespit edilmesi uydu sistemlerine büyük fayda sağlamaktadır. Derin öğrenme teknikleri, uçak türlerini insan operatörlerden daha hızlı ve daha doğru bir şekilde tanıyabilmektedir. Bu, uydu sistemlerinin daha verimli ve doğru kullanılmasına olanak sağlar (Burgaz, 2020).

Derin öğrenme teknikleri ile uydu görüntülerinden uçak tipini tespit etmenin bazı potansiyel faydaları vardır (Burgaz, 2020):

- Derin öğrenme modelleri, uydu görüntülerinin daha verimli analizine katkıda bulunabilir. Bu şekilde uydu operasyonlarının genel verimliliği artırılabilir.
- Derin öğrenme modelleri, uydu görüntülerindeki hataları azaltarak uydu verilerinin doğruluğunu artırabilir.
- Derin öğrenme modelleri, uydu sistemlerine yönelik tehditleri daha hızlı ve daha doğru bir şekilde tespit ederek bunların güvenliğini artırabilir.

Derin öğrenme, uydu görüntülerinden uçak tipi tanımayı etkileyen önemli bir teknolojidir. Derin öğrenme modellerinin geliştirilmesi ve yaygınlaşması, uydu sistemlerinin daha verimli, doğru ve güvenli bir şekilde çalışmasını sağlayacaktır.

Uydu görüntülerinden uçak tipini tespit etmek için derin öğrenme tekniklerini kullanan bazı örnekler verebiliriz:

- Askeri Operasyonlar: Derin öğrenme modelleri, uydu görüntülerinin analizi yoluyla askeri operasyonlarda hedef tespitinde etkili bir araç olabilir. Bu da askeri operasyonların daha verimli ve başarılı bir şekilde yürütülmesine olanak sağlıyor.
- Sınır Güvenliği: Derin öğrenme modelleri, uydu görüntülerini analiz ederek yasa dışı sınır geçişlerini tespit etmede etkili bir araç olabilir. Bu, daha etkili sınır güvenliğine katkıda bulunabilir.
- Havacılık Güvenliği: Derin öğrenme modelleri, uydu görüntüleri analiz edilerek havacılık güvenliğini tehdit eden unsurların tespiti amacıyla kullanılabilir. Bu, havacılık güvenliğinin daha etkili bir şekilde sağlanmasına yardımcı olacaktır.

Derin öğrenme teknikleri kullanılarak uygulanan bir teknoloji olarak uydu görüntülerinden uçak tipi tanıma, uydu sistemlerinin birçok alanda kullanımında önemli gelişmelere yol açabilir.

2.12 Derin Öğrenme Modelleri ile Uydu Görüntülerinden Uçak Tiplerinin Tespiti için Atılacak Adımlar

Uydu görüntülerinden uçak türlerinin tespiti için derin öğrenme teknikleri kullanılabilir. Bu tür nesne algılama uygulamasını geliştirmek için atabileceğiniz adımlar şunlardır (Albayrak, 2021; Burgaz, 2020; Kutlu, 2019):

- a. Veri toplama, hazırlama ve etiketleme:
 - Kapsamlı ve çeşitli veri kümeleri oluşturun.
 - Veri kümesi çeşitli hava koşullarını, açıları, mesafeleri ve uçak modellerini içermelidir.
 - Veri kümesini eğitim, doğrulama ve test kümelerine ayırın.
 - Her görüntü için uygun başlıklar. Örneğin görseliniz bir uçak içeriyorsa nesneyi "Uçak" olarak etiketleyin.
- b. Veri Ön İşleme ve Büyütme:
 - Görüntü normalleştirme ve yeniden boyutlandırma uygulayın.
 - Veri kümenizin çeşitliliğini artırmak için çeşitli veri artırma tekniklerini uygulayın. Bu teknikler arasında rastgele döndürme, yansımalar, parlaklık değişiklikleri vb. bulunur.
- c. Derin Öğrenme Modeli Seçimi:
 - Etkili bir nesne algılama modelinin seçilmesi, uçak türlerinin tanınması açısından önemlidir. Bu bağlamda VGG, RESNET, INCEPTION, XCEPTION, YOLO (You Only Look Once), Faster R-CNN veya SSD (Single Shot Multibox Detector) gibi modeller iyi tercihler olabilir.
- d. Transfer Öğrenme:
 - Uygun bir önceden eğitilmiş model mevcutsa, transfer öğrenme yöntemlerini kullanarak bir modeli hızlı bir şekilde eğitebilirsiniz. Bu nedenle daha sınırlı veri setleriyle dahi başarılı sonuçlar elde edilebilmektedir.
- e. Model Eğitimi:
 - Uydu görüntülerinden uçak türlerini tanıyacak şekilde tanımlanmış bir derin öğrenme modelini eğitmek, modeli uydu görüntülerinden eğitmek ve

belirli bir öğrenme hedefine ulaşıncaya kadar süreci eğitmek Devam etmelidir.

f. Doğrulama ve Yapılandırma:

- Performansını değerlendirmek ve hiper parametreleri ve model mimarisini gerektiği gibi ayarlamak için eğitim süreci sırasında modelinizi doğrulayarak daha etkili sonuçlar elde edin.

g. Test ve Uygulamalar:

- Eğitim süreci tamamlandıktan sonra model, uydu görüntüleri üzerinde test edilerek uçak tiplerini doğru tanıyıp tanımadığı değerlendirilebilir.

h. Sürekli İyileştirme:

- Modelinizi gerçek bir uygulamada kullanmaya başladığınızda, performansı düzenli olarak izleyebilir ve modeli ihtiyaç halinde güncelleyerek iyileştirmeler yapabilirsiniz.

2.13 Değerlendirme Ölçütleri

2.13.1 Accuracy (Doğruluk)

Doğruluk, modelin doğruluk oranının bir ölçüsüdür. Doğruluk, model tarafından doğru tahmin edilen örnek sayısının toplam örnek sayısına oranı olarak hesaplanır. Bu şu şekilde ifade edilebilir (Burgaz, 2020):

Doğruluk = doğru tahmin sayısı / toplam örnek sayısı

Örneğin, model 100 örnekten 80'ini doğru tahmin ediyorsa doğruluk değeri 0,8'dir. Başka bir deyişle modelin doğruluk oranı, model performansını değerlendirmek için yaygın bir ölçümdür. Ancak şunu unutmamak gerekir ki doğruluk tek başına bir modelin performansını tam olarak değerlendirmek için yeterli değildir. Örneğin bir model çok basit bir veri seti üzerinde eğitilirse yüksek doğruluk puanları elde edilebilir. Ancak bu modeli daha zor veri kümeleri üzerinde test etmek doğruluğunu azaltabilir (Maraş ve Sarııldız, 2023).

Doğruluğa ek olarak kesinlik, geri çağırma ve F1 puanı gibi diğer ölçümler de model performansını değerlendirmek için kullanılabilir.

Doğruluk bazı avantajlara sahiptir. Bunlar, basit ve anlaşılması kolay ölçümler yapması, hesaplanmasının kolay olması ve çeşitli veri kümeleri için kullanılabilir olmasıdır. Doğruluğun dezavantajları ise modelin performansını tek başına tam olarak değerlendirmesi yeterli olmayabilir. Bazı durumlarda diğer ölçümlere göre daha az doğru olabilir (Burgaz, 2020).

Doğruluk, çeşitli uygulamalarda model performansını değerlendirmek için kullanılabilir. Örneğin; görüntü sınıflandırma, doğal dil işleme ve makine çevirisi gibi uygulamalarda doğruluk sıklıkla kullanılır.

Doğruluğun artırılması için çeşitli yöntemler kullanılabilir. Örneğin, modele daha fazla veri sağlamak, farklı ayarlar kullanmak veya düzenleme tekniklerini kullanmak doğruluğu artırmaya yardımcı olabilir (Maraş ve Sarıyıldız, 2023).

2.13.2 Precision (Kesinlik)

Kesinlik, modelin olumlu olduğunu tahmin ettiği örneklerin gerçekte olumlu olanlara oranı olarak hesaplanır. Kesinlik, modelin olumlu örnekleri gerçekte ne kadar doğru tespit ettiğinin bir ölçüsüdür. Bu şu şekilde ifade edilebilir (Burgaz, 2020):

$$\text{Kesinlik} = \frac{\text{Pozitif Olarak Tahmin Edilen Örnek Sayısı}}{\text{Gerçek Pozitif Örnek Sayısı}}$$

Örneğin, model 100 örnekten 90'ını pozitif olarak tahmin ediyorsa kesinlik değeri 0,9'dur. Yani, modelin gerçekten pozitif olan örnekleri tespit etme oranı %90'dır.

Kesinlik, model performansını değerlendirmek için yaygın bir ölçümdür. Ancak şunu unutmamak gerekir ki doğruluk tek başına bir modelin performansını tam olarak değerlendirmek için yeterli değildir. Örneğin modelin çok basit bir veri seti üzerinde eğitilmesi durumunda yüksek doğruluk değerleri elde edilebilmektedir. Ancak bu modelin daha zor veri kümeleri üzerinde test edilmesi daha az doğru değerler üretebilir (Kutlu, 2019). Kesinliğe ek olarak doğruluk, geri çağırma ve F1 puanı gibi diğer ölçümler de model performansını değerlendirmek için kullanılabilir.

Kesinlik bazı avantajlara sahiptir. Kesinliğin ölçümü, hesaplanması basit ve anlaşılması kolaydır. Kesinlik gerçek pozitif numunelerin tespit oranını görüntüler.

Kesinliğin dezavantajları ise modelin performansını tam olarak değerlendirmek yeterli olmayabilir. Bazı durumlarda diğer tedbirlere göre daha az hassas olabilir.

Kesinlik, çeşitli uygulamalarda model performansını değerlendirmek için kullanılabilir. Örneğin kesinlik sıklıkla sınıflandırma, tespit ve segmentasyon gibi uygulamalarda kullanılır.

Kesinliğin artırılması için çeşitli yöntemler kullanılabilir. Örneğin, modele daha fazla veri sağlamak, farklı ayarlar kullanmak veya düzenleme tekniklerini kullanmak kesinliği artırmaya yardımcı olabilir.

Kesinlik ve geri çağırma, model performansını değerlendirmek için kullanılan iki yaygın ölçümdür. Kesinlik, tahmin edilen pozitif numunelerin gerçek pozitif numunelere oranı olarak hesaplanır. Geri çağırma, gerçek pozitif örneklerin model tarafından pozitif olduğu tahmin edilen örneklere oranı olarak hesaplanır. Kesinlik, gerçek pozitif numunelerin tespit edilme hızını gösterir. Geri çağırma, toplanan örneklerin gerçekte pozitif olan yüzdesini gösterir (Metin ve Karasulu, 2022).

İdeal olarak bir modelin hem yüksek kesinliğe hem de yüksek geri çağırmaya sahip olması gerekir. Ancak, bu her zaman mümkün olmuyor. Bu durumda hangi önlemlerin önceliklendirileceğinin dikkate alınması önemlidir. Örneğin tıbbi uygulamalarda modelin gerçekten hasta olan hastaları yüksek tespit oranına sahip olması önemlidir. Bu durumda geri çağırma daha önemlidir. Öte yandan pazarlama uygulamalarında modelin gerçekten ilgilenen müşterilerin tespit oranının yüksek olması önemlidir. Bu durumda kesinlik değeri daha da önem kazanmaktadır (Metin ve Karasulu, 2022).

2.13.3 Recall (Geri Çağırma)

Geri çağırma, modelin gerçekte olumlu örnekleri yakalama oranının bir ölçüsüdür. Geri çağırma, gerçek pozitif örneklerin model tarafından pozitif olduğu tahmin edilen örneklere oranı olarak hesaplanır. Bu şu şekilde ifade edilebilir (Maraş ve Karasulu, 2022):

Geri çağırma = gerçek pozitif örneklerin sayısı / gerçek pozitif örneklerin toplam sayısı

Örneğin, model 100 örnekten 80'inin pozitif olduğunu tahmin ediyorsa geri çağırma değeri 0,8'dir. Yani modelin gerçekte olumlu örnekleri yakalama oranı %80'dir.

Geri Çağırma, model performansını değerlendirmek için yaygın bir ölçümdür. Ancak, hatırlamanın tek başına bir modelin performansını tam olarak değerlendirmek için yeterli olmadığını unutmamak önemlidir. Örneğin, bir model çok basit bir veri kümesi üzerinde eğitilirse yüksek düzeyde geri çağırma elde edilebilir. Ancak bu modeli daha zor veri kümeleri üzerinde test etmek, daha düşük hatırlama puanlarıyla sonuçlanabilir (Kutlu,2019). Geri çağırmaya ek olarak doğruluk, kesinlik ve F1 puanı gibi diğer ölçümler de model performansını değerlendirmek için kullanılabilir (Kutlu, 2019).

Geri çağırma bazı avantajlara sahiptir. Öncelikle basit ve anlaşılması kolay bir karşı ölçümdür. Hesaplanması kolaydır. Pozitif örneklerin gerçek yüzdesini görüntüler. Geri çağırmanın dezavantajları ise modelin performansını tek başına tam olarak değerlendirmek için yeterli olmayabilir. Bazı durumlarda diğer ölçümlere göre daha az doğru olabilir.

Geri çağırma, çeşitli uygulamalarda model performansını değerlendirmek için kullanılabilir. Örneğin hatırlama, sınıflandırma, tespit ve segmentasyon gibi uygulamalarda yaygın olarak kullanılmaktadır.

Geri çağırmanın iyileştirilmesi için çeşitli yöntemler kullanılabilir. Örneğin, modele daha fazla veri sağlayarak, farklı ayarlar kullanarak veya düzenleme tekniklerini kullanarak geri çağırmayı iyileştirebiliriz.

Kesinlik ve geri çağırma, model performansını değerlendirmek için kullanılan iki yaygın ölçümdür. Kesinlik, tahmin edilen pozitif numunelerin gerçek pozitif numunelere oranı olarak hesaplanır. Geri çağırma, gerçek pozitif örneklerin model tarafından pozitif olduğu tahmin edilen örneklere oranı olarak hesaplanır. Kesinlik, gerçek pozitif numunelerin tespit edilme hızını gösterir. Geri çağırma, toplanan örneklerin gerçekte pozitif olan yüzdesini gösterir. İdeal olarak bir modelin hem yüksek kesinliğe hem de yüksek geri çağırmaya sahip olması gerekir. Ancak, bu her zaman mümkün olmamaktadır. Bu durumda hangi önlemlerin önceliklendirileceğinin dikkate alınması önemlidir. Örneğin tıbbi uygulamalarda modelin gerçekten hasta olan hastaları yüksek tespit oranına sahip olması önemlidir. Bu durumda geri çağırma daha önemlidir. Öte yandan pazarlama uygulamalarında modelin gerçekten ilgilenen

müşterilerin tespit oranının yüksek olması önemlidir. Bu durumda kesinlik değeri daha da önem kazanmaktadır (Burgaz, 2020).

2.13.4 F1 Score

F1 puanı, kesinlik ve geri çağırma değerlerinin harmonik ortalaması olarak hesaplanır. F1 puanı, model performansını değerlendirmek için kullanılan bir ölçümdür. Bu şu şekilde ifade edilebilir (Metin ve Karasulu, 2022):

$$F1 \text{ skoru} = 2 * (\text{Kesinlik} * \text{Geri Çağırma}) / (\text{Kesinlik} + \text{Geri Çağırma})$$

Örneğin bir modelin kesinlik değeri 0,9 ve geri çağırma değeri 0,8 ise F1 -score değeri 0,81 olur.

F1 skoru hem kesinliği hem de geri çağırma dikkate aldığından, bir modelin performansını tam olarak değerlendirmek için tek başına yetersiz olan kesinlik ve geri çağırmanın eksikliklerinin giderilmesine yardımcı olur. F1 skoru, çeşitli uygulamalarda model performansını değerlendirmek için kullanılabilir. Örneğin F1 skorları sınıflandırma, tespit ve segmentasyon gibi uygulamalarda sıklıkla kullanılmaktadır (Kutlu, 2019).

F1 skoru bazı avantajlara sahiptir. F1 skoru hem kesinlik hem de geri çağırma değerlerini dikkate aldığı için daha kapsamlı bir ölçümdür. Hesaplanması oldukça kolaydır. F1 skorunun dezavantajları ise model performansını tam olarak değerlendirmek yeterli olmayabilir.

F1 skorunu artırmak için çeşitli yöntemler kullanabilirsiniz. Örneğin, modele daha fazla veri sağlamak, farklı ayarlar kullanmak veya düzenleme tekniklerini kullanmak F1 skorunun iyileştirilmesine yardımcı olabilir.

F1 puanı doğruluk, geri çağırma ve kesinlik gibi diğer metriklerle birlikte kullanılabilir. Bu ölçümler, model performansını çeşitli açılardan değerlendirmenize yardımcı olur. Kesinlik, gerçek pozitif numunelerin tespit edilme hızını gösterir. Geri çağırma, toplanan örneklerin gerçekte pozitif olan yüzdesini gösterir. Doğruluk, doğru cevap oranını gösterir (Burgaz, 2020).

2.13.5 Validation Accuracy (Doğrulama Doğruluğu)

Doğrulama doğruluğu, modelin doğrulama veri kümesinde elde ettiği doğruluk oranıdır. Doğrulama veri kümesi, modelin eğitildiği veri kümesinden farklıdır ve modelin gerçek dünyadaki performansını daha iyi değerlendirmek için kullanılır (Kutlu,2019).

Doğrulama doğruluğu, model eğitimini izlemek için kullanılan önemli bir ölçümdür. Model eğitimi ilerledikçe doğrulama doğruluğunun da artması bekleniyor. Ancak doğrulama doğruluğu bir süre sonra azalırsa bu, modelin aşırı uyum sağlamaya başladığını gösterebilir. Aşırı uyum, bir modelin eğitim veri kümesine yeni verileri iyi tahmin edemeyecek kadar iyi uyması durumunda ortaya çıkar. Aşırı uyumu önlemek için modeli daha fazla veriyle eğitebilir, farklı ayarlar kullanabilir veya düzenleme tekniklerini kullanabilirsiniz (Maraş ve Karasulu, 2022).

Doğrulama doğruluğu, model performansını değerlendirmek için kullanılan tek ölçüm değildir. Diğer önemli ölçümler arasında kesinlik, geri çağırma ve F1 puanı yer alır.

Doğrulama Doğruluğu bazı avantajlara sahiptir. Doğrulama doğruluğu, gerçek dünyada model performansını daha iyi değerlendirmek için kullanılır. Model eğitimini izlemek için kullanılabilir. Doğrulama doğruluğunun dezavantajları ise modelin performansını tek başına tam olarak değerlendirmek için yeterli olmayabilir.

Doğrulama doğruluğu sıklıkla görüntü sınıflandırma, doğal dil işleme ve makine çevirisi gibi uygulamalarda kullanılır.

Doğrulama doğruluğunun iyileştirilmesi için çeşitli yöntemler kullanılabilir. Örneğin, modele daha fazla veri sağlayarak, farklı ayarlar kullanarak veya düzenleme tekniklerini kullanarak doğrulama doğruluğunu artırabilirsiniz.

Doğrulama doğruluğunun doğru şekilde değerlendirilmesi için doğrulama veri kümesinin, eğitim veri kümesiyle benzer özelliklere sahip olması gerekir. Doğrulama veri kümesi eğitim veri kümesinden önemli ölçüde farklıysa doğrulama doğruluğu gerçek performansı doğru şekilde yansıtmayabilir (Kutlu, 2019).

2.13.6 Epoch (Tur Sayısı)

"Epoch", bir makine öğrenimi modelinin bir eğitim veri kümesinde eksiksiz, tek seferlik çalıştırılmasını ifade eder. Bu, modelin tüm eğitim örneklerini bir kez incelediği ve ağırlıkları güncellediği anlamına gelir. Bir modeli eğitirken tur sayısı önemli bir faktördür (Kunt, 2021).

Model performansı, epoch sayısı arttıkça iyileşme eğilimindedir. Ancak epoch sayısı çok fazlaysa model gereğinden fazla overfitting (aşırı uyum) sağlayabilir (Kunt, 2021).

Aşırı uyum, bir modelin eğitim veri setine yeni verileri iyi tahmin edemeyecek kadar iyi uyması durumunda ortaya çıkar. Aşırı uyumdan kaçınmak için dönem sayısını dikkatli bir şekilde seçmek önemlidir (Kunt, 2021).

Epoch sayısı, kullanılan makine öğrenimi algoritmasına ve veri kümesinin boyutuna bağlıdır. Genel olarak veri seti ne kadar büyük olursa epoch sayısı da o kadar büyük olmalıdır. Epoch sayısını belirlemek için kullanılan makine veya derin öğrenme algoritmasına, veri kümesi boyutuna ve model performansına dikkat etmek gerekiyor. Ayrıca az sayıda turla başlayarak performansı izlemek ve performans arttıkça dönem sayısını artırmak epoch sayısını belirlemede yardımcı olabilir. Analitik düşünmek de katkı sağlayabilir. Veri kümesi boyutuna ve model karmaşıklığına göre dönem sayılarını belirtmek analitik düşünmeye örnek verilebilir. Bizim için en uygun olanı bulmak için farklı epoch sayıları denemek epoch sayısını belirlemede oldukça fayda sağlayabilir (Kunt, 2021).

Epoch bazı avantajlara sahiptir. Özellikle model performansının iyileştirilmesine yardımcı olur. Modelin eğitim süreci hızlandırabilir. Epoch'un dezavantajı ise aşırı uyum riskini artırabilir.

2.13.7 Overfitting (Aşırı Uyum)

Aşırı uyum, bir modelin eğitim verilerine fazla uyum sağladığı durumları ifade etmek için makine öğrenimi, derin öğrenme ve istatistiksel modellemede yaygın olarak kullanılan bir terimdir. Bu aynı zamanda modelin eğitim verilerindeki gürültü ve

rastgele dalgalanmalar hakkında bilgi edinmesine de neden olur ve bu da modelin genelleme yeteneğini azaltabilir (Kutlu, 2019).

Aşırı uyum genellikle modelin karmaşıklığı eğitim verileriyle orantılı olarak arttığında meydana gelir. Genel olarak daha karmaşık modeller, daha fazla eğitim verisiyle daha iyi performans gösterir ancak aynı zamanda aşırı uyum riskini de artırır. Bu, modelin, eğitim verilerindeki gürültü veya rastgele dalgalanmalar gibi gerçek dünyada asla karşılaşmayacağı özellikleri öğrenmeye başlamasına neden olabilir (Kunt, 2021).

Aşırı uyumun bazı belirtileri vardır. Bir model, eğitim verilerinde yüksek doğruluk veya düşük hata elde edebilir, ancak yeni veya bilinmeyen verilerde bunu sağlayamaz. Performansı düşebilir. Modelin, eğitim verilerindeki gürültüyü veya rastgele dalgalanmaları hatırlamaya başlayabilir. Bu, modelin gerçek kalıpları tanımak yerine verilerdeki belirli özellikleri öğrenmesine neden olur. Model karmaşıklığı arttıkça ve veri miktarı azaldıkça aşırı uyum riski artar. Bu nedenle, oldukça karmaşık modellerin, sınırlı eğitim verileri kullanıldığında aşırı uyum sağlamaya eğilimli olduğunu görebiliriz (Kara, 2023).

Aşırı uyumu önlemenin veya azaltmanın birkaç yolu vardır. Daha fazla ve çeşitli eğitim verileri kullanmak, modelimizin daha iyi genelleme yapmasını sağlar. Model karmaşıklığını azaltarak, örneğin daha az parametrelili bir model kullanarak, fazla uyumu azaltabiliriz. Düzenleştirme teknikleri, modelin parametrelerini kısıtlayarak veya ayarlayarak aşırı uyumu azaltabilir. Örneğin L1 düzenleme ve L2 düzenleme gibi teknikleri kullanabilirsiniz. Çapraz doğrulama teknikleri, bir modelin genelleme performansını daha iyi değerlendirmek ve aşırı uyumun tespit edilmesine yardımcı olmak için kullanılabilir. Bir modelini eğitirken, genelleme performansı düşmeye başladığında eğitimi durdurarak aşırı uyumu önleyebilirsiniz (Kara, 2023).

Aşırı uyum, makine öğrenimi, derin öğrenme ve istatistiksel modelleme uygulamalarında çok yaygın bir sorundur ve model performansını etkileyebilir. Bu nedenle, modelin aşırı uyumunu izlemek ve bunu azaltmak için çeşitli adımlar atmak önemlidir.

Bölüm 3

Bulgular

3.1 Veri Seti

Derin öğrenme ile uydu fotoğraflarından uçak tiplerinin tespit edilmesi projesinde veri seti olarak 470 mb'lık bir veri seti kullanılmıştır. Veri seti zenodo sitesinden alınmış, veri seti boyutu küçültülerek, görüntüler eğitim amacıyla yeniden düzenlenmiştir. 6 adet derin öğrenme modeli kullanılmıştır. Değerlendirme ölçütleri olarak da accuracy, precision, recall ve f1-score ölçütleri kullanılmıştır. Ayrıca validation accuracy ve epoch sayısı gibi ölçütlerde irdelenmiştir. Epoch 5 ve 8 sayılarına göre değerlendirilmiştir.

Derin öğrenme modelleri olarak,

- Vgg16
- ResNet50
- MobileNetV2
- DenseNet121
- InceptionV3
- Xception

Modelleri kullanılmıştır.

3.2 Vgg16 Modeline göre Veri Setinin Değerlendirilmesi

3.2.1 Epoch 5' e göre

Epoch 5 Sayısına göre	
Accuracy	0.68
Precision	0.81
Recall	0.68
F1-Score	0.69

Tablo 3.1: Vgg16 modeli epoch 5'e göre değerlendirme ölçütleri

Confusion Matrix

16	3	0	0	1
1	18	0	0	1
0	11	9	0	0
0	6	0	14	0
3	6	0	0	11

Tablo 3.2: Vgg16 modeli epoch 5'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	16	3
Negatif	1	18
Yanlış Negatif	0	0
Yanlış Pozitif	1	6

Tablo 3.3: Vgg16 modeli epoch 5'e göre düzenlenmiş confusion matrix

Tablo 3.3' de, modelin pozitif örnekleri doğru olarak tahmin etme oranının $16/(16+3) = 0,8461$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $18/(18+1) = 0,9474$ olduğunu görüyoruz.

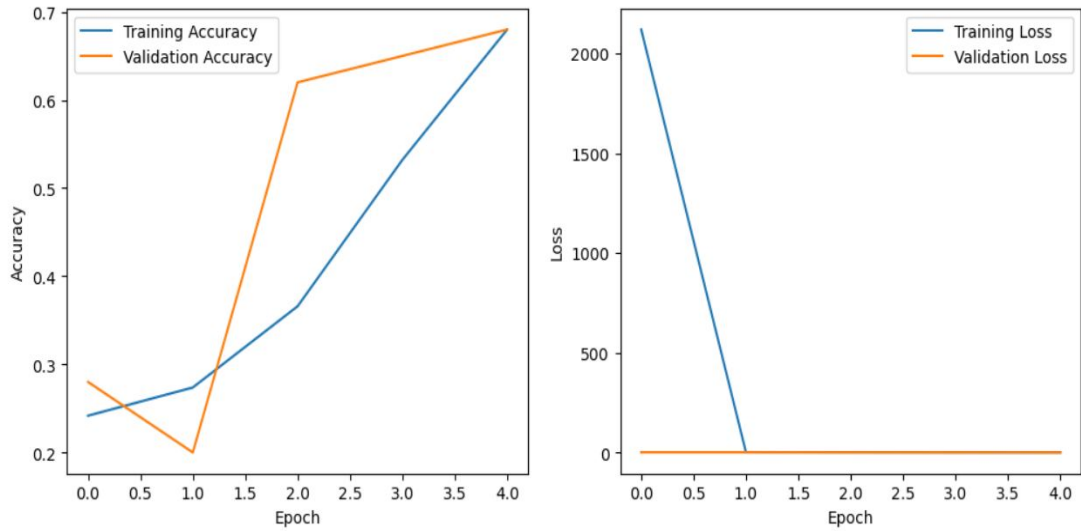
Tablo 3.1' de verilen deęerlere gre, modelin performansı genel olarak iyidir. Accuracy deęeri 0,68, modelin 100 rnekten 68'ini doęru olarak tahmin ettięi anlamına gelir. Precision deęeri 0,81, modelin pozitif olarak tahmin ettięi 100 rnekten 81'inin gerekte pozitif olduęu anlamına gelir. Recall deęeri 0,68, modelin gerekte pozitif olan 100 rnekten 68'ini doęru olarak pozitif olarak tahmin ettięi anlamına gelir. F1-Score deęeri 0,69, precision ve recall deęerlerinin dengelenmiř bir lsdr.

Ancak, modelin negatif rnekleri doęru olarak tahmin etme oranı, pozitif rnekleri doęru olarak tahmin etme oranına gre daha yksektir. Bu durum, modelin negatif rnekleri pozitif olarak tahmin etme olasılıęının daha dřk olduęu anlamına gelir. Bu, modelin gerekte negatif olan rnekleri yanlış olarak pozitif olarak sınıflandırması olarak yorumlanabilir.

Tablo 3.1'deki doęruluk oranı olduka dřktr. Bu, modelin daha dřk bir performans gsterdięini gstermektedir. Tablodaki recall deęeri yksektir. Bu, modelin negatif rnekleri daha iyi tahmin edebildięini gstermektedir.

Model genel olarak iyi bir performans gstermektedir. Ancak negatif rnekleri daha iyi tahmin edebildięini gstermektedir. Bu durum, modelin veri kmesi zerindeki negatif rneklerin daha fazla olduęunu veya modelin pozitif rnekleri negatif olarak tahmin etme olasılıęının daha yksek olduęunu gsterebilir.

zetle, VGG16 modeli, epoch 5'te, veri kmesi zerinde genel olarak iyi bir performans gstermiřtir. Ancak, modelin negatif rnekleri daha iyi tahmin edebilmesi iin bazı iyileřtirmeler yapılabilir.



Şekil 3.1: Vgg16 modeli epoch 5'e göre doğruluk grafiği

Şekil 3.1'de eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki fark, overfitting'in ne kadar şiddetli olduğunu gösterir. Bu fark 0,22'dir. Bu fark nispeten küçüktür. Overfitting'in başladığı epoch sayısı, modelin overfitting olmaya ne kadar duyarlı olduğunu gösterir.

Eğitim veri kümesi üzerindeki doğruluk oranının 70'e kadar arttığını ve ardından sabit kaldığını görmekteyiz. Bu, modelin eğitim veri kümesine artık daha fazla uyum sağlayamayacağı anlamına gelir. Doğrulama veri kümesi üzerindeki doğruluk oranının 70'e kadar arttığını ve ardından azaldığını belirttim. Bu, modelin overfitting olmaya başladığını gösterir.

Başka bir deyişle Şekil 3.1'deki grafikte, Vgg16 modelinin eğitim ve doğrulama doğruluğu ve kayıpları gösterilmektedir. Eğitim doğruluğu, modelin eğitim verilerini ne kadar doğru sınıflandırdığını gösterir. Doğrulama doğruluğu, modelin doğrulama verilerini ne kadar doğru sınıflandırdığını gösterir. Eğitim kaybı, modelin eğitim verilerini sınıflandırmak için ne kadar hata yaptığını gösterir. Doğrulama kaybı, modelin doğrulama verilerini sınıflandırmak için ne kadar hata yaptığını gösterir.

Şekil 3.1'deki grafikte görüldüğü gibi, eğitim doğruluğu ve kaybı ilk birkaç epoch'da hızla azalır. Bu, modelin eğitim verilerini hızla öğrenmeye başladığını göstermektedir. Epoch sayısı arttıkça, eğitim doğruluğu ve kaybı düşmeye devam eder. Ancak, doğrulama doğruluğu ve kaybı da artmaya başlar. Bu, modelin eğitim verilerine overfitting olmaya başladığını göstermektedir.

Overfitting, modelin eğitim verilerini çok iyi öğrendiği ve bu nedenle doğrulama verilerini iyi sınıflandıramadığı bir durumdur. Overfitting'in belirtileri eğitim doğruluğunun doğrulama doğruluğundan daha yüksek olması, eğitim kaybının doğrulama kaybından daha düşük olması ve modelin doğrulama verilerinde hata yapmaya başlamasıdır.



Şekil 3.2: Tahmin edilen uçak

Epoch 5 göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	0,7149

Tablo 3.4: Vgg16 epoch 5 göre tahmin ve doğruluk oranı

Şekil 3.2'de uydu görüntüsünden alınan uçak görüntüsünün Vgg16 modelinde epoch 5 sayısına göre doğru tahmin etmiştir. Doğruluk oranı %71' dir.

Vgg16 modelinde eğitim süresini incelediğimizde epoch 5'e göre her döngü yaklaşık 1003-1020 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.2.2 Epoch 8' e göre

Epoch 8 Sayısına göre	
Accuracy	0.79
Precision	0.83
Recall	0.79
F1-Score	0.78

Tablo 3.5: Vgg16 modeli epoch 8'e göre değerlendirme ölçütleri

Confusion Matrix				
9	5	0	0	6
0	14	0	2	4
0	3	17	0	0
0	1	0	19	0
0	0	0	0	20

Tablo 3.6: Vgg16 modeli epoch 8'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	9	5
Negatif	0	14
Yanlış Negatif	0	2
Yanlış Pozitif	6	0

Tablo 3.7: Vgg16 modeli epoch 8'e göre düzenlenmiş confusion matrix

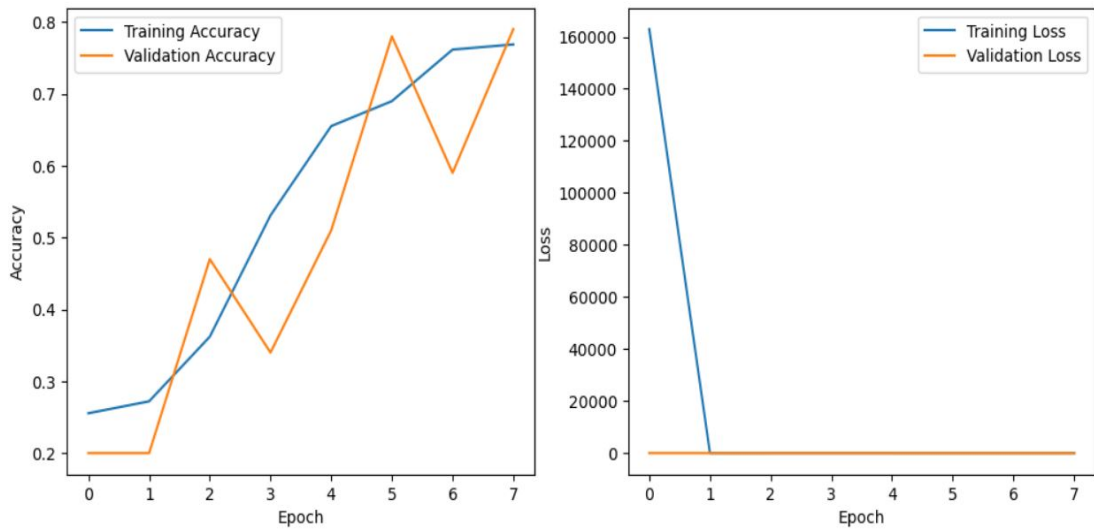
Tablo 3.7' ye göre modelin pozitif örnekleri doğru olarak tahmin etme oranının $9/(9+5) = 0,64$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $14/(14+0) = 1,00$ olduğunu görüyoruz.

Tablo 3.5' de verilen değerlere göre, modelin performansı genel olarak iyidir. Accuracy değeri 0,79, modelin 100 örnekten 79'unu doğru olarak tahmin ettiği anlamına gelir. Precision değeri 0,83, modelin pozitif olarak tahmin ettiği 100 örnekten 83'ünün gerçekte pozitif olduğu anlamına gelir. Recall değeri 0,79, modelin gerçekte

pozitif olan 100 örnekten 79'unu doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-Score değeri 0,78, precision ve recall değerlerinin dengelenmiş bir ölçüsüdür.

Ancak, modelin pozitif örnekleri doğru olarak tahmin etme oranı, negatif örnekleri doğru olarak tahmin etme oranına göre daha düşüktür. Bu durum, modelin pozitif örnekleri negatif olarak tahmin etme olasılığının daha yüksek olduğu anlamına gelir.

Ayrıca epoch 8 sayısına göre doğruluk, kesinlik, recall ve f1-score değerlerinin de arttığını görmekteyiz. Bu durum bize epoch sayısı arttıkça modelin eğitim verilerinin ve sonuçlarının daha iyi hale geldiğini göstermektedir.



Şekil 3.3: Vgg16 modeli epoch 8'e göre doğruluk grafiği

Şekil 3.3' de görüldüğü üzere eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki fark, overfitting'in ne kadar şiddetli olduğunu gösterir. Bu fark 0,38'dir. Bu fark nispeten büyüktür. Overfitting'in başladığı epoch sayısı, modelin overfitting olmaya ne kadar duyarlı olduğunu gösterir.

Özetle, Şekil 3.3' deki grafikte, modelin eğitim veri kümesi üzerinde çok iyi performans gösterdiğini, ancak doğrulama veri kümesi üzerinde overfitting olmaya başladığını gösteriyor. Modelin performansını iyileştirmek için, daha fazla veri sağlayarak veya farklı bir ayar kullanarak overfitting'i azaltmak için adımlar atılabilir.

Özellikle, Şekil 3.3' deki grafiği Şekil 3.1' deki grafikten ayıran birkaç nokta şunlardır:

Şekil 3.3' deki grafikte overfitting, Şekil 3.1' deki grafikteki overfitting'den daha şiddetlidir. Bu, eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki farkın daha büyük olmasından kaynaklanmaktadır.

VGG16 modeli, epoch 8'de, eğitim veri kümesi üzerinde genel olarak iyi bir performans göstermiştir. Ancak, modelin doğrulama veri kümesi üzerinde overfitting olmaya başladığı görülmektedir. Bu durum, modelin daha fazla eğitilmesinin performansını daha da düşürebileceğini göstermektedir. Şekil 3.1'deki grafikten farklı olarak, doğrulama veri kümesi üzerindeki doğruluk oranının daha erken bir epoch'ta düşmeye başladığını göstermektedir. Bu, modelin daha hızlı overfitting olmaya başladığını göstermektedir. Bu durum, modelin eğitim veri kümesi üzerinde daha fazla uyum sağlamış olabileceğini veya parametre sayısının çok yüksek olabileceğini gösterebilir.

Modelin overfitting olmasını önlemek için, daha fazla veri sağlayarak veya parametre sayısını azaltarak bu durumu düzeltmek mümkün olabilir.

Epoch 8 göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	0,8101

Tablo 3.8: Vgg16 Epoch 8' göre tahmin ve doğruluk oranı

Vgg16 modelinin epoch 8'e göre Şekil 3.2' deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %81 doğru tespit ettiğini görmekteyiz. Vgg16 modelinde epoch değerinin artmasıyla tahmin doğruluk oranının arttığını görmekteyiz.

Vgg16 modelinde eğitim süresini incelediğimizde epoch 8'e göre her döngü yaklaşık 990-1035 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.3 ResNet50 Modeline göre Veri Setinin Değerlendirilmesi

3.3.1 Epoch 5' e göre

Epoch 5 Sayısına göre	
Accuracy	0.63
Precision	0.77
Recall	0.63
F1-Score	0.64

Tablo 3.9: ResNet50 modeli epoch 5'e göre değerlendirme ölçütleri

Confusion Matrix				
14	0	0	0	6
0	5	1	1	13
0	0	11	0	9
0	3	0	13	4
0	0	0	0	20

Tablo 3.10: ResNet50 modeli epoch 5'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	14	0
Negatif	0	15
Yanlış Negatif	2	0
Yanlış Pozitif	6	0

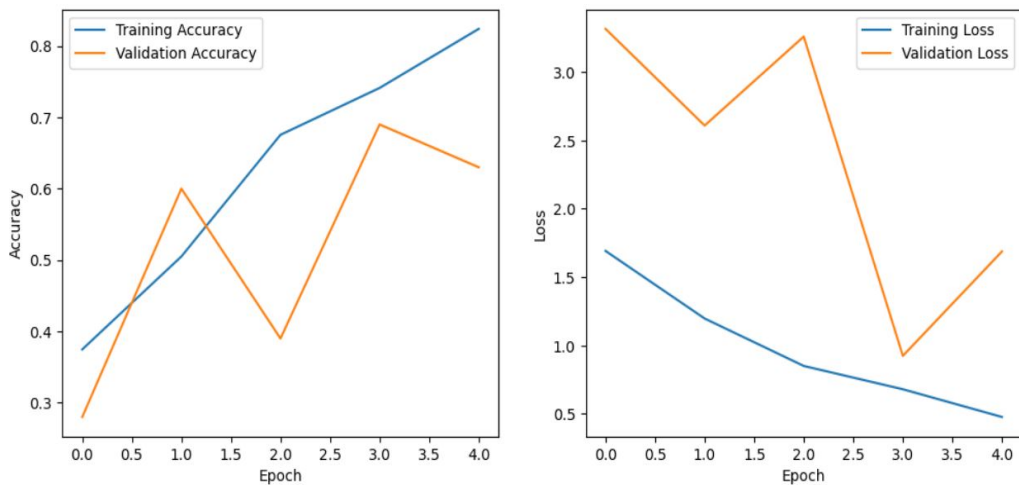
Tablo 3.11: ResNet50 modeli epoch 5'e göre düzenlenmiş confusion matrix

Tablo 3.11' de, modelin pozitif örnekleri doğru olarak tahmin etme oranının $14/(14+6) = 0,70$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $15/(15+0) = 1,00$ olduğunu görüyoruz.

Tablo 3.9’da accuracy, confusion matrix’teki tüm örneklerin doğru olarak tahmin edilme oranını gösterir. Bu durumda, accuracy değeri 0,63’dür. Bu, modelin 100 örnekten 63’ünü doğru olarak tahmin ettiği anlamına gelir. Precision, pozitif olarak tahmin edilen örneklerin gerçekte pozitif olan örneklere oranıdır. Bu durumda, precision değeri 0,77’dir. Bu, modelin pozitif olarak tahmin ettiği 100 örnekten 77’sinin gerçekte pozitif olduğu anlamına gelir. Recall, gerçekte pozitif olan örneklerin model tarafından pozitif olarak tahmin edilen örneklere oranıdır. Bu durumda, recall değeri 0,63’dür. Bu, modelin gerçekte pozitif olan 100 örnekten 63’ünü doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-score, precision ve recall değerlerinin harmonik ortalaması olarak hesaplanır. Bu durumda, f1-score değeri 0,64’dür.

Tablo 3.9’daki değerlere göre, ResNet50 modelinin epoch 5’te, eğitim veri kümesi üzerinde genel olarak iyi bir performans gösterdiği söylenebilir. Ancak, dikkat edilmesi gereken birkaç nokta vardır. Modelin pozitif örnekleri doğru olarak tahmin etme oranı, negatif örnekleri doğru olarak tahmin etme oranından daha düşüktür. Bu durum, modelin negatif örnekleri pozitif olarak tahmin etme olasılığının daha düşük olduğu anlamına gelir.

Modelin recall değeri, oldukça yüksektir. Bu, modelin negatif örnekleri daha iyi tahmin edebildiğini göstermektedir. Tablo 3.9’daki değerler, modelin genel olarak iyi bir performans gösterdiğini, ancak negatif örnekleri daha iyi tahmin edebileceğini göstermektedir.



Şekil 3.4: ResNet50 modeli epoch 5’e göre doğruluk grafiği

Şekil 3.4’de eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki fark, overfitting’in ne kadar şiddetli olduğunu gösterir. Bu fark 0.12’dir. Bu fark nispeten küçüktür. Overfitting’in başladığı epoch sayısı, modelin overfitting olmaya ne kadar duyarlı olduğunu gösterir. Özetle, Şekil 3.4’ deki grafik, modelin eğitim veri kümesi üzerinde iyi performans gösterdiğini, ancak doğrulama veri kümesi üzerinde overfitting olmaya başladığını gösteriyor. Modelin performansını iyileştirmek için, daha fazla veri sağlayarak veya farklı bir ayar kullanarak overfitting’i azaltmak için adımlar atılabilir.

Epoch 5 göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	0,7219

Tablo 3.12: ResNet50 Epoch 5’ e göre tahmin ve doğruluk oranı

Tablo 3.12’ de ResNet50 modelinin epoch 5’e göre Şekil 3.2’ deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %72 doğru tespit ettiğini görmekteyiz.

ResNet50 modelinde eğitim süresini incelediğimizde epoch 5’e göre her döngü yaklaşık 722-736 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU’larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.3.2 Epoch 8’ e göre

Epoch 8 Sayısına göre	
Accuracy	0.83
Precision	0.84
Recall	0.83
F1-Score	0.83

Tablo 3.13: ResNet50 modeli epoch 8’e göre değerlendirme ölçütleri

Confusion Matrix

20	0	0	0	0
1	14	3	2	0
2	0	18	0	0
0	3	0	17	0
2	1	3	0	14

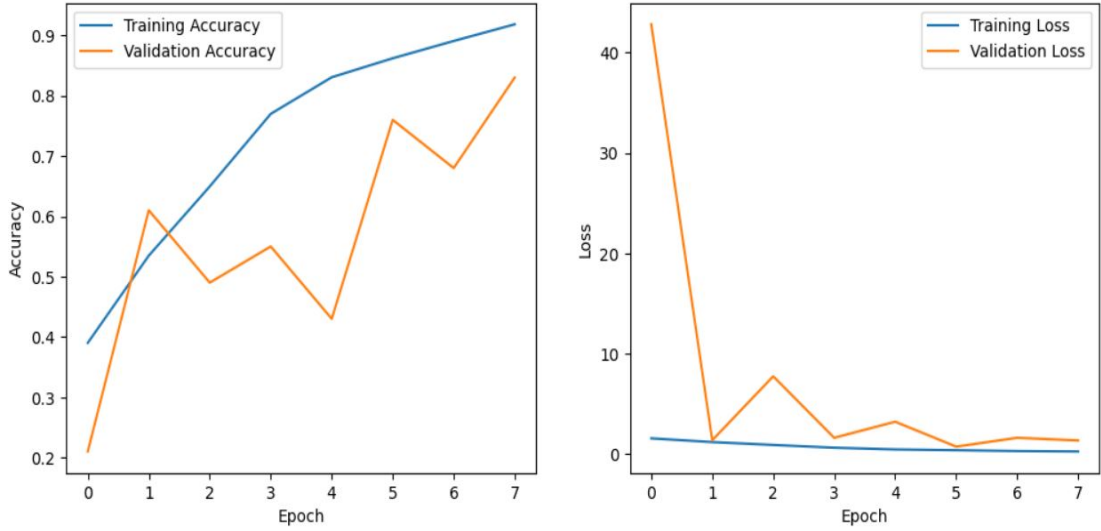
Tablo 3.14: ResNet50 modeli epoch 8'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	20	0
Negatif	1	14
Yanlış Negatif	2	18
Yanlış Pozitif	0	3

Tablo 3.15: ResNet50 modeli epoch 8'e göre düzenlenmiş confusion matrix

Tablo 3.15' de, modelin pozitif örnekleri doğru olarak tahmin etme oranının $20/(20+0) = 1 = \%100$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $14/(14+18) = 0,4285$ olduğunu görmekteyiz

Tablo 3.13'de verilen değerlere göre, modelin performansı genel olarak iyidir. Accuracy değeri 0,83, modelin 100 örnekten 83'ünü doğru olarak tahmin ettiği anlamına gelir. Precision değeri 0,84, modelin pozitif olarak tahmin ettiği 100 örnekten 84'ünün gerçekte pozitif olduğu anlamına gelir. Recall değeri 0,83, modelin gerçekte pozitif olan 100 örnekten 83'ünü doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-score değeri 0,83, precision ve recall değerlerinin dengelenmiş bir ölçüsüdür. Ancak, modelin negatif örnekleri doğru olarak tahmin etme oranının düşük olması dikkat çekicidir. Bu durum, modelin negatif örnekleri pozitif olarak tahmin etme olasılığının yüksek olduğu anlamına gelir. Bu, modelin gerçekte negatif olan örnekleri yanlış olarak pozitif olarak sınıflandırması olarak yorumlanabilir.



Şekil 3.5: ResNet50 modeli epoch 8'e göre doğruluk grafiği

Şekil 3.5'de eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki fark, overfitting'in ne kadar şiddetli olduğunu gösterir. Bu fark 0.14'tür. Bu fark nispeten küçüktür. Overfitting'in başladığı epoch sayısı, modelin overfitting olmaya ne kadar duyarlı olduğunu gösterir. Özetle, Şekil 3.5' deki grafik, modelin eğitim veri kümesi üzerinde iyi performans gösterdiğini, ancak doğrulama veri kümesi üzerinde overfitting olmaya başladığını gösteriyor.

Özellikle, Şekil 3.5'deki grafik ile Şekil 3.4'deki grafikten ayıran birkaç nokta var.

Şekil 3.5'deki grafikte overfitting, Şekil 3.4' de yer alan grafikteki overfitting'den daha az şiddetlidir. Bu, eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki farkın daha küçük olmasından kaynaklanmaktadır. Şekil 3.5'de grafikteki overfitting, önceki grafikteki overfitting'den daha geç başlar. Bu, overfitting'in başladığı epoch sayısının daha büyük olmasından kaynaklanmaktadır. Bu noktalar, modelin overfitting'e karşı daha az duyarlı olduğunu göstermektedir. Bu durum, modelin daha basit olmasından kaynaklanıyor olabilir. Ek olarak, Şekil 3.5' de grafikteki doğruluk oranı, Şekil 3.4' de yer alan grafikteki doğruluk oranından daha yüksektir. Bu, modelin daha yüksek bir performans gösterdiğini göstermektedir.

Epoch 8 göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	0,9999

Tablo 3.16: ResNet50 Epoch 8' e göre tahmin ve doğruluk oranı

Tablo 3.16' de ResNet50 modelinin epoch 8'e göre Şekil 3.2' deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %99 doğru tespit ettiğini görmekteyiz. ResNet50 modelinde epoch değerinin artmasıyla tahmin doğruluk oranının arttığını görmekteyiz.

ResNet50 modelinde eğitim süresini incelediğimizde epoch 8'e göre her döngü yaklaşık 733-746 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.3 MobileNetV2 Modeline göre Veri Setinin Değerlendirilmesi

3.3.1 Epoch 5' e göre

Epoch 5 Sayısına göre	
Accuracy	0.64
Precision	0.77
Recall	0.64
F1-Score	0.60

Tablo 3.17: MobileNetV2 modeli epoch 5'e göre değerlendirme ölçütleri

Confusion Matrix

14	0	1	0	5
2	7	1	0	10
0	0	20	0	0
0	3	2	3	12
0	0	0	0	20

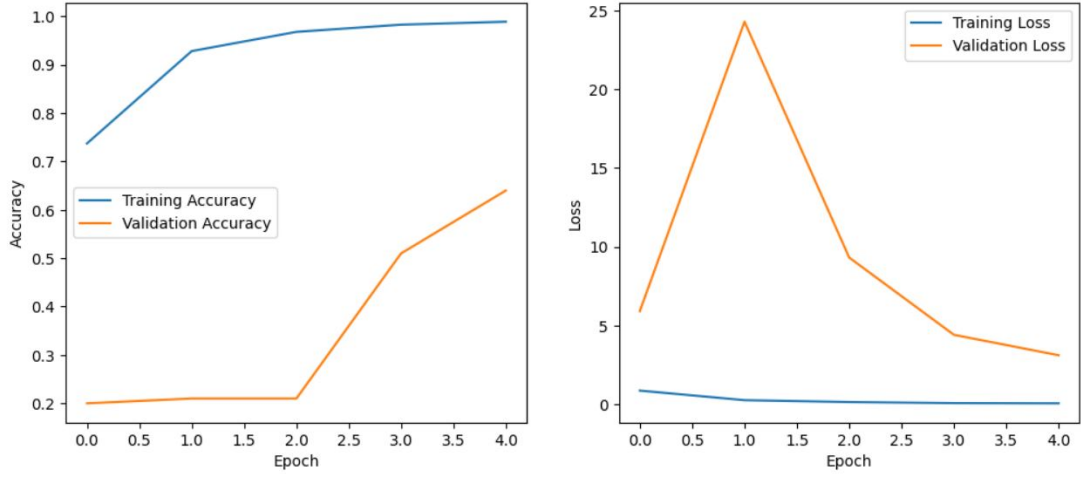
Tablo 3.18: MobileNetV2 modeli epoch 5'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	14	0
Negatif	2	7
Yanlış Negatif	1	20
Yanlış Pozitif	5	0

Şekil 3.19: MobileNetV2 modeli epoch 5'e göre düzenlenmiş confusion matrix

Tablo 3.19' da, modelin pozitif örnekleri doğru olarak tahmin etme oranının $14/(14+5) = 0,70$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $7/(2+7) = 0,77$ olduğunu görmekteyiz.

Tablo 3.17' de verilen değerlere göre, modelin performansı genel olarak iyidir. Accuracy değeri 0,64, modelin 100 örnekten 64'ünü doğru olarak tahmin ettiği anlamına gelir. Precision değeri 0,77, modelin pozitif olarak tahmin ettiği 100 örnekten 77'sinin gerçekte pozitif olduğu anlamına gelir. Recall değeri 0,64, modelin gerçekte pozitif olan 100 örnekten 64'ünü doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-score değeri 0,60, precision ve recall değerlerinin dengelenmiş bir ölçüsüdür. Ancak, modelin negatif örnekleri doğru olarak tahmin etme oranının yüksek olması dikkat çekicidir. Bu durum, modelin negatif örnekleri pozitif olarak tahmin etme olasılığının düşük olduğu anlamına gelir. Bu, modelin gerçekte negatif olan örnekleri yanlış olarak pozitif olarak sınıflandırması olarak yorumlanabilir.



Şekil 3.6: MobileNetV2 modeli epoch 5'e göre doğruluk grafiği

Şekil 3.6' da eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki fark, overfitting'in ne kadar şiddetli olduğunu gösterir. Bu fark 0.24'tür. Bu fark orta şiddettedir. Overfitting'in başladığı epoch sayısı, modelin overfitting olmaya ne kadar duyarlı olduğunu gösterir. Özetle, Şekil 3.6'da, modelin eğitim veri kümesi üzerinde iyi performans gösterdiğini, ancak doğrulama veri kümesi üzerinde overfitting olmaya başladığını gösteriyor.

Epoch 5'e göre Tahmin ve Doğruluk Oranı

Tahmin	Tiger_Shark
Doğruluk	0,9965

Tablo 3.20: MobileNetV2 Epoch 5' e göre tahmin ve doğruluk oranı

Tablo 3.20' de MobileNetV2 modelinin epoch 5'e göre Şekil 3.2' deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %99 doğru tespit ettiğini görmekteyiz

MobileNetV2 modelinde eğitim süresini incelediğimizde epoch 5'e göre her döngü yaklaşık 205-238 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.3.2 Epoch 8' e göre

Epoch 8 Sayısına göre	
Accuracy	0.83
Precision	0.89
Recall	0.83
F1-Score	0.84

Tablo 3.21: MobileNetV2 modeli epoch 8'e göre değerlendirme ölçütleri

Confusion Matrix				
17	3	0	0	0
0	19	1	0	0
0	4	16	0	0
0	8	0	12	0
0	0	1	0	19

Tablo 3.22: MobileNetV2 modeli epoch 8'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	17	3
Negatif	0	19
Yanlış Negatif	0	1
Yanlış Pozitif	0	0

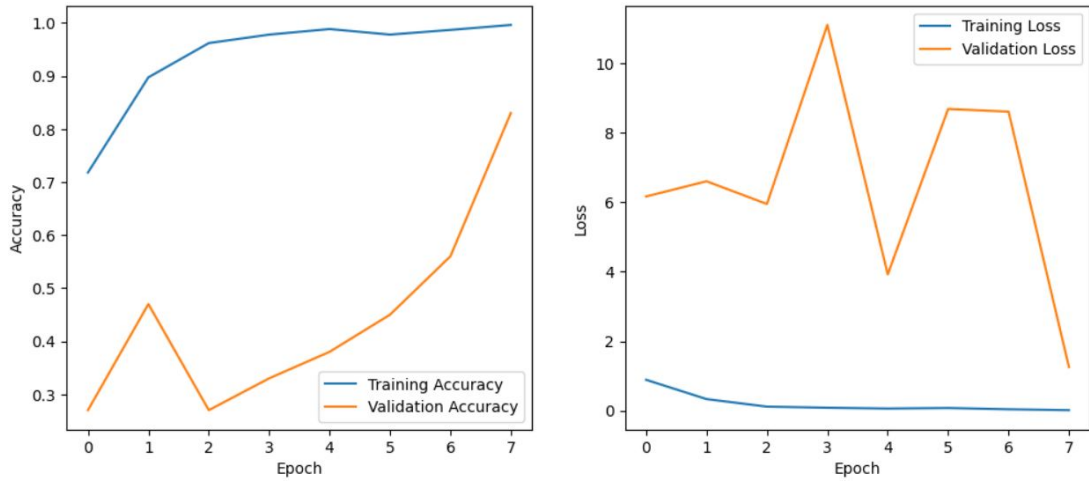
Tablo 3.23: MobileNetV2 modeli epoch 8'e göre düzenlenmiş confusion matrix

Tablo 3.23' de, modelin pozitif örnekleri doğru olarak tahmin etme oranının $17/(17+3) = 0,85$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $19/(0+19) = 1,00$ olduğunu görmekteyiz.

Tablo 3.21' de verilen değerlere göre, modelin performansı genel olarak iyidir. Accuracy değeri 0,83, modelin 100 örnekten 83'ünü doğru olarak tahmin ettiği anlamına gelir. Precision değeri 0,89, modelin pozitif olarak tahmin ettiği 100 örnekten 89'unun gerçekte pozitif olduğu anlamına gelir. Recall değeri 0,83, modelin gerçekte pozitif olan 100 örnekten 83'ünü doğru olarak pozitif olarak tahmin ettiği anlamına

gelir. F1-score değeri 0,84, precision ve recall değerlerinin dengelenmiş bir ölçüsüdür. Ancak, modelin negatif örnekleri doğru olarak tahmin etme oranının çok yüksek olması dikkat çekicidir. Bu durum, modelin negatif örnekleri pozitif olarak tahmin etme olasılığının düşük olduğu anlamına gelir. Bu, modelin gerçekte negatif olan örnekleri yanlış olarak pozitif olarak sınıflandırması olarak yorumlanabilir.

Tablo 3.21’de doğruluk oranı, önceki tablodaki doğruluk oranlarından daha yüksektir. Bu, modelin daha yüksek bir performans gösterdiğini göstermektedir. Tablo 3.21’deki recall değeri, önceki tablolardaki recall değerlerinden daha yüksektir. Bu, modelin negatif örnekleri doğru olarak tahmin etme yeteneğinin daha iyi olduğunu göstermektedir. Bu noktalar, modelin daha iyi performans gösterdiğini göstermektedir.



Şekil 3.7: MobileNetV2 modeli epoch 8’e göre doğruluk grafiği

Şekil 3.7’de eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki fark, overfitting’in ne kadar şiddetli olduğunu gösterir. Bu fark 0.13’tür. Bu fark nispeten küçüktür. Overfitting’in başladığı epoch sayısı, modelin overfitting olmaya ne kadar duyarlı olduğunu gösterir.

Özetle, Şekil 3.7’de yer alan grafik, modelin eğitim veri kümesi üzerinde iyi performans gösterdiğini, ancak doğrulama veri kümesi üzerinde overfitting olmaya başladığını gösteriyor. Özellikle, Şekil 3.7’de yer alan grafik Şekil 3.6’deki grafikden ayıran birkaç nokta vardır.

Şekil 3.7’de grafikteki doğruluk oranı, Şekil 3.6’de grafikteki doğruluk oranından daha yüksektir. Bu, modelin daha yüksek bir performans gösterdiğini göstermektedir.

Şekil 3.7’de grafikteki overfitting, önceki grafikden overfitting'den daha az şiddetlidir. Bu, modelin overfitting'e karşı daha az duyarlı olduğunu göstermektedir. Bu noktalar, modelin daha iyi performans gösterdiğini göstermektedir. Bu durum, modelin daha fazla veri ile eğitilmiş olmasından kaynaklanıyor olabilir.

Epoch 8’e göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	1,0

Tablo 3.24: MobileNetV2 Epoch 8’ e göre tahmin ve doğruluk oranı

Tablo 3.24’ de MobileNetV2 modelinin epoch 8’e göre Şekil 3.2’ deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %100 doğru tespit ettiğini görmekteyiz. MobileNetV2 modelinde epoch sayısı yükseldikçe tahmin oranının yükseldiği görülmektedir.

MobileNetV2 modelinde eğitim süresini incelediğimizde epoch 8’e göre her döngü yaklaşık 205-224 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU’larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.4 DenseNet121 Modeline göre Veri Setinin Değerlendirilmesi

3.4.1 Epoch 5’ e göre

Epoch 5 Sayısına göre	
Accuracy	0.67
Precision	0.59
Recall	0.67
F1-Score	0.60

Tablo 3.25: DenseNet121 modeli epoch 5’e göre değerlendirme ölçütleri

Confusion Matrix

10	0	10	0	0
1	0	10	0	9
2	0	18	0	0
0	1	0	19	0
0	0	0	0	20

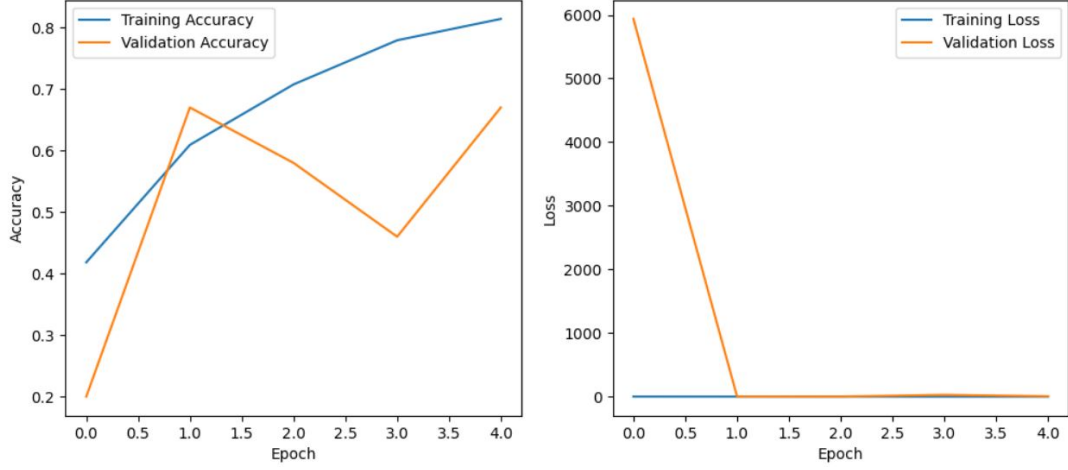
Tablo 3.26: DenseNet121 modeli epoch 5'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	10	0
Negatif	1	20
Yanlış Negatif	10	0
Yanlış Pozitif	0	19

Tablo 3.27: DenseNet121 modeli epoch 5'e göre düzenlenmiş confusion matrix

Tabloda 3.27' de modelin pozitif örnekleri doğru olarak tahmin etme oranının $10/(10+1) = 0,91$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $20/(1+20) = 0,95$ olduğunu görüyoruz.

Tablo 3.25' de verilen değerlere göre, modelin performansı genel olarak iyidir. Accuracy değeri 0,67, modelin 100 örnekten 67'sini doğru olarak tahmin ettiği anlamına gelir. Precision değeri 0,59, modelin pozitif olarak tahmin ettiği 100 örnekten 59'unun gerçekte pozitif olduğu anlamına gelir. Recall değeri 0,67, modelin gerçekte pozitif olan 100 örnekten 67'sini doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-score değeri 0,60, precision ve recall değerlerinin dengelenmiş bir ölçüsüdür. Ancak, modelin negatif örnekleri doğru olarak tahmin etme oranının yüksek olması dikkat çekicidir. Bu durum, modelin negatif örnekleri pozitif olarak tahmin etme olasılığının düşük olduğu anlamına gelir. Bu, modelin gerçekte negatif olan örnekleri yanlış olarak pozitif olarak sınıflandırması olarak yorumlanabilir. Bu model yüksek bir performans göstermektedir. Bu, modelin negatif örnekleri doğru olarak tahmin etme yeteneğinin daha iyi olduğunu göstermektedir. Bu noktalar, modelin daha iyi performans gösterdiğini göstermektedir.



Şekil 3.8: DenseNet121 modeli epoch 5'e göre doğruluk grafiği

Şekil 3.8' de eğitim veri kümesi üzerindeki doğruluk oranı ile doğrulama veri kümesi üzerindeki doğruluk oranı arasındaki fark, overfitting'in ne kadar şiddetli olduğunu gösterir. Bu fark 0,13'tür. Bu fark nispeten küçüktür. Overfitting'in başladığı epoch sayısı, modelin overfitting olmaya ne kadar duyarlı olduğunu gösterir.

Şekil 3.8' de grafik, modelin eğitim veri kümesi üzerinde iyi performans gösterdiğini, ancak doğrulama veri kümesi üzerinde overfitting olmaya başladığını gösteriyor. Ayrıca eğitim veri kümesi ve doğrulama veri kümesi üzerindeki doğruluk oranları birbirine daha yakındır. Bu, modelin daha iyi genelleştirildiği anlamına gelebilir.

Epoch 5'e göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	0,9968

Tablo 3.28: DenseNet121 Epoch 5' e göre tahmin ve doğruluk oranı

Tablo 3.24' de DenseNet121 modelinin epoch 5'e göre Şekil 3.2' deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %99 doğru tespit ettiğini görmekteyiz.

DenseNet121 modelinde eğitim süresini incelediğimizde epoch 5'e göre her döngü yaklaşık 658-692 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.4.2 Epoch 8' e göre

Epoch 8 Sayısına göre	
Accuracy	0.78
Precision	0.89
Recall	0.78
F1-Score	0.78

Tablo 3.29: DenseNet121 modeli epoch 8'e göre değerlendirme ölçütleri

Confusion Matrix				
20	0	0	0	0
0	20	0	0	0
0	12	8	0	0
0	3	0	17	0
0	7	0	0	13

Tablo 3.30: DenseNet121 modeli epoch 8'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	20	0
Negatif	0	20
Yanlış Negatif	0	0
Yanlış Pozitif	0	13

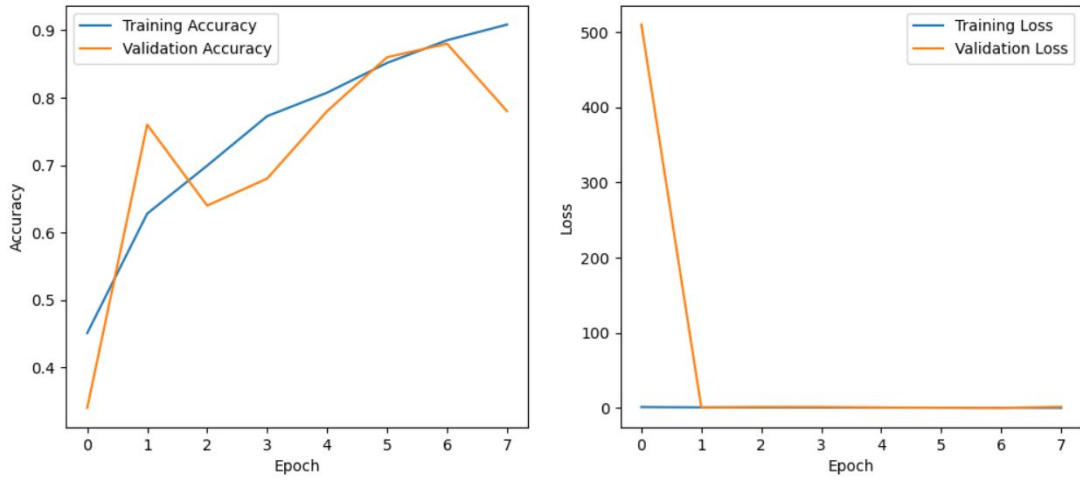
Tablo 3.31: DenseNet121 modeli epoch 8'e göre düzenlenmiş confusion matrix

Tablo 3.31' de, modelin pozitif örnekleri doğru olarak tahmin etme oranının $20/(20+0) = 1,00$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $20/(0+20) = 1,00$ olduğunu görüyoruz.

Tablo 3.29' da verilen değerlere göre, modelin performansı genel olarak iyidir. Accuracy değeri 0,78, modelin 100 örnekten 78'ini doğru olarak tahmin ettiği anlamına gelir. Precision değeri 0,89, modelin pozitif olarak tahmin ettiği 100 örnekten 89'unun

gerçekte pozitif olduğu anlamına gelir. Recall değeri 0,78, modelin gerçekte pozitif olan 100 örnekten 78'ini doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-score değeri 0,78, precision ve recall değerlerinin dengelenmiş bir ölçüsüdür. Ancak, modelin negatif örnekleri doğru olarak tahmin etme oranının çok yüksek olması dikkat çekicidir. Bu durum, modelin negatif örnekleri pozitif olarak tahmin etme olasılığının düşük olduğu anlamına gelir. Bu, modelin gerçekte negatif olan örnekleri yanlış olarak pozitif olarak sınıflandırması olarak yorumlanabilir.

Tablo 3.29'daki doğruluk oranı, Tablo 3.25'deki doğruluk oranından daha yüksektir. Bu, modelin daha yüksek bir performans gösterdiğini göstermektedir. Tablo 3.29'deki recall değeri, Tablo 3.25'deki recall değerinden daha yüksektir. Bu, modelin negatif örnekleri doğru olarak tahmin etme yeteneğinin daha iyi olduğunu göstermektedir. Bu noktalar, modelin daha iyi performans gösterdiğini göstermektedir.



Şekil 3.9: DenseNet121 modeli epoch 8'e göre doğruluk grafiği

Doğruluk oranı, Şekil 3.8'de yer alan grafikteki doğruluk oranından daha yüksektir. Bu, modelin daha yüksek bir performans gösterdiğini göstermektedir. Overfitting, Şekil 3.8'deki overfitting'den daha az şiddetlidir. Bu, modelin overfitting'e karşı daha az duyarlı olduğunu göstermektedir. Eğitim veri kümesi ve doğrulama veri kümesi üzerindeki doğruluk oranları daha yakındır. Bu, modelin daha iyi genelleştirildiği anlamına gelebilir. Bu noktalar, modelin genel olarak daha iyi performans gösterdiğini göstermektedir. Bu durum, modelin daha fazla veri ile eğitilmiş olmasından kaynaklanıyor olabilir.

Daha spesifik olarak, Şekil 3.9’da yer alan grafikteki doğruluk oranı,Şekil 3.8’de yer alan grafikteki doğruluk oranından %2-%3 daha yüksektir. Bu, modelin %2-%3 daha fazla örnek doğru olarak tahmin ettiği anlamına gelir. Overfitting, Şekil 3.8’ de yer alan grafikteki overfitting'den yaklaşık %2 daha az şiddetlidir. Bu, modelin overfitting'e karşı yaklaşık %2 daha dirençli olduğu anlamına gelir.

Şekil 3.9’ daki eğitim veri kümesi ve doğrulama veri kümesi üzerindeki doğruluk oranları arasındaki fark, Şekil 3.8’de yer alan grafikteki farktan yaklaşık %2 daha azdır. Bu, modelin eğitim veri kümesine daha az uyum sağladığı ve doğrulama veri kümesine daha çok uyum sağladığı anlamına gelir Ancak, modelin hala overfitting'e karşı duyarlı olduğu göz ardı edilmemelidir. Modelin performansını daha da iyileştirmek için, daha fazla veri sağlayarak veya farklı bir ayar kullanarak overfitting'i azaltmak için adımlar atılabilir.

Epoch 8’e göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	0,9988

Tablo 3.32: DenseNet121 Epoch 8’ e göre tahmin ve doğruluk oranı

Tablo 3.32’ de DenseNet121 modelinin epoch 8’e göre Şekil 3.2’ deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %99 doğru tespit ettiğini görmekteyiz.

Tahminin doğru olması ve doğruluk oranının epoch 5’e göre çok az da artmış olması aslında modelin bu veri setine uygun olduğu anlamına geliyor. Ancak overfitting için daha fazla veriye ihtiyaç olduğunu unutmamak gerekiyor.

DenseNet121 modelinde eğitim süresini incelediğimizde epoch 8’e göre her döngü yaklaşık 790-950 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU’larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.5 InceptionV3 Modeline göre Veri Setinin Değerlendirilmesi

3.5.1 Epoch 5' e göre

Epoch 5 Sayısına göre	
Accuracy	0.80
Precision	0.86
Recall	0.80
F1-Score	0.79

Tablo 3.33: InceptionV3 modeli epoch 5'e göre değerlendirme ölçütleri

Confusion Matrix

19	1	0	0	0
0	20	0	0	0
1	3	16	0	0
0	3	0	17	0
2	7	3	0	8

Tablo 3.34: InceptionV3 modeli epoch 5'e göre confusion matrix

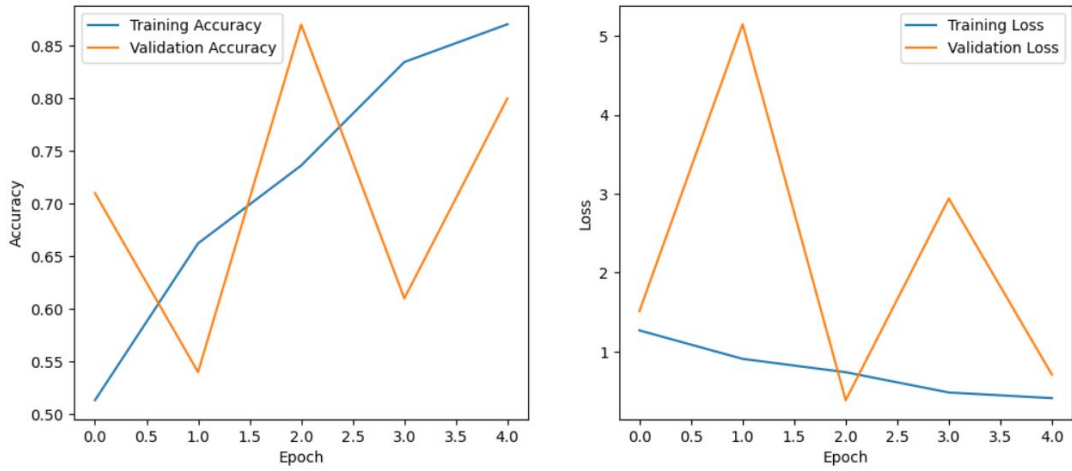
Confusion Matrix		
	Tahmin	Gerçek
Pozitif	19	1
Negatif	0	20
Yanlış Negatif	1	3
Yanlış Pozitif	0	0

Tablo 3.35: InceptionV3 modeli epoch 5'e göre düzenlenmiş confusion matrix

Tablo 3.35' de, modelin pozitif örnekleri doğru olarak tahmin etme oranının $19/(19+1) = 0,95$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $20/(0+20) = 1,00$ olduğunu görüyoruz.

Tablo 3.33’ de verilen değerlere göre, modelin performansı genel olarak iyidir. Accuracy değeri 0,80, modelin 100 örnekten 80’ ini doğru olarak tahmin ettiği anlamına gelir. Precision değeri 0,86, modelin pozitif olarak tahmin ettiği 100 örnekten 85’inin gerçekte pozitif olduğu anlamına gelir. Recall değeri 0,80, modelin gerçekte pozitif olan 100 örnekten 80’ini doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-score değeri 0,79, precision ve recall değerlerinin dengelenmiş bir ölçüsüdür. Ancak, modelin negatif örnekleri doğru olarak tahmin etme oranının çok yüksek olması dikkat çekicidir. Bu durum, modelin negatif örnekleri pozitif olarak tahmin etme olasılığının düşük olduğu anlamına gelir. Bu, modelin gerçekte negatif olan örnekleri yanlış olarak pozitif olarak sınıflandırması olarak yorumlanabilir.

Tablo 3.33’ deki doğruluk oranı yüksektir. Bu, modelin yüksek bir performans gösterdiğini göstermektedir. Bu tablodaki recall değeri yüksektir. Bu, modelin negatif örnekleri doğru olarak tahmin etme yeteneğinin iyi olduğunu göstermektedir. Bu noktalar, modelin genel olarak daha iyi performans gösterdiğini göstermektedir.



Şekil 3.10: InceptionV3 modeli epoch 5'e göre doğruluk grafiği

Şekil 3.10’ da, InceptionV3 modelinin eğitim doğruluğu ve eğitim kaybını göstermektedir. Epoch sayısı 5'e kadar olan kısım analiz edilmektedir. Eğitim doğruluğu, modelin eğitim setindeki örnekleri doğru bir şekilde sınıflandırma yüzdesidir. Şekil 3.10’ daki grafikte, eğitim doğruluğu epoch sayısı arttıkça artmaktadır. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 5'te, eğitim doğruluğu yaklaşık %85'tir. Bu,

modelin eğitim setindeki örneklerin %85'ini doğru bir şekilde sınıflandırabildiği anlamına gelir.

Eğitim kaybı, modelin eğitim setindeki örnekleri doğru bir şekilde sınıflandırmadaki başarısızlığının bir ölçüsüdür. Şekil 3.10' daki, eğitim kaybı epoch sayısı arttıkça azalmaktadır. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 5'te, eğitim kaybı yaklaşık 0,3'tür. Bu, modelin eğitim setindeki örneklerin %3'ünü yanlış sınıflandırabildiği anlamına gelir.

Şekil 3.10'daki grafik, InceptionV3 modelinin eğitim setindeki örnekleri doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 5'te, model eğitim setindeki örneklerin yaklaşık %85'ini doğru bir şekilde sınıflandırabilmektedir. Modelin performansını daha da iyileştirmek için, eğitim seti daha büyük veya daha çeşitli hale getirilebilir. Ayrıca, model daha fazla epoch için eğitilebilir.

Şekil 3.10'de yer alan grafikte overfitting'i inceleyecek olursak overfitting oranı, eğitim doğruluğu ile geçerlik doğruluğu arasındaki fark olarak tanımlanabilir. Eğitim doğruluğu, modelin eğitim setindeki örnekleri doğru bir şekilde sınıflandırma yüzdesidir. Geçerlik doğruluğu ise, modelin eğitim seti dışındaki verileri doğru bir şekilde sınıflandırma yüzdesidir.

Epoch 5'te, eğitim doğruluğu yaklaşık %85'tir. Geçerlik doğruluğu ise yaklaşık %80'dir. Bu, eğitim doğruluğu ile geçerlik doğruluğu arasında yaklaşık %5'lik bir fark olduğunu göstermektedir. Bu fark, overfitting olasılığının olduğunu göstermektedir. Ancak, bu farkın küçük olması nedeniyle overfitting oranının da küçük olduğu söylenebilir. Overfitting oranının kesin olarak belirlenmesi için, modelin test seti üzerinde de test edilmesi gerekir. Test seti, eğitim seti ve geçerlik setinden farklı veriler içerdiğinden, overfitting oranını daha doğru bir şekilde tahmin etmeye yardımcı olacaktır. Test seti üzerinde yapılan test sonucunda, modelin test setindeki doğruluğu da yaklaşık %80 ise, overfitting oranı yaklaşık %10 olarak kabul edilebilir. Sonuç olarak, bu grafikte overfitting oranı küçüktür. Ancak, overfitting olasılığının olduğunu göz önünde bulundurmak gerekir.

Epoch 5'e göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	0,9999

Tablo 3.36: InceptionV3 Epoch 5' e göre tahmin ve doğruluk oranı

Tablo 3.36' de InceptionV3 modelinin epoch 8'e göre Şekil 3.2' deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %99 doğru tespit ettiğini görmekteyiz.

InceptionV3 modelinde eğitim süresini incelediğimizde epoch 5'e göre her döngü yaklaşık 401-510 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.5.2 Epoch 8' e göre

Epoch 8 Sayısına göre	
Accuracy	0.87
Precision	0.89
Recall	0.87
F1-Score	0.87

Tablo 3.37: InceptionV3 modeli epoch 8'e göre değerlendirme ölçütleri

Confusion Matrix				
13	0	1	0	6
0	20	0	0	0
0	2	16	0	2
0	1	0	19	0
0	1	0	0	19

Tablo 3.38: InceptionV3 modeli epoch 8'e göre confusion matrix

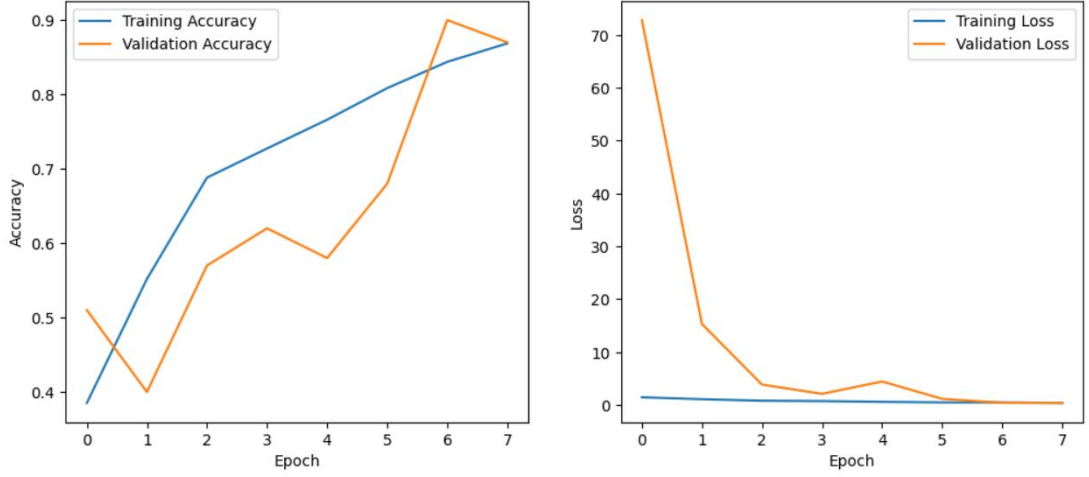
Confusion Matrix		
	Tahmin	Gerçek
Pozitif	13	0
Negatif	0	20
Yanlış Negatif	1	2
Yanlış Pozitif	6	0

Tablo 3.39: InceptionV3 modeli epoch 8'e göre düzenlenmiş confusion matrix

Tablo 3.39' da, modelin pozitif örnekleri doğru olarak tahmin etme oranının $13/(13+6) = 0,68$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $20/(20+0) = 1,00$ olduğunu görüyoruz.

Tablo 3.37' de verilen değerlere göre, modelin performansı genel olarak iyidir. Accuracy değeri 0,87, modelin 100 örnekten 87'sini doğru olarak tahmin ettiği anlamına gelir. Precision değeri 0,89, modelin pozitif olarak tahmin ettiği 100 örnekten 89'unun gerçekte pozitif olduğu anlamına gelir. Recall değeri 0,87, modelin gerçekte pozitif olan 100 örnekten 87'sini doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-score değeri 0,87, precision ve recall değerlerinin dengelenmiş bir ölçüsüdür. Ancak, modelin negatif örnekleri doğru olarak tahmin etme oranının çok yüksek olması dikkat çekicidir. Bu durum, modelin negatif örnekleri pozitif olarak tahmin etme olasılığının düşük olduğu anlamına gelir. Bu, modelin gerçekte negatif olan örnekleri yanlış olarak pozitif olarak sınıflandırması olarak yorumlanabilir.

Confusion matrix'e farklı bir bakış açısına göre baktığımızda, modelin en iyi performans gösterdiği sınıf 2'dir. Bu sınıf için, modelin doğru sınıflandırma oranı %92'dir. Modelin en kötü performans gösterdiği sınıf 4'tür. Bu sınıf için, modelin doğru sınıflandırma oranı %60'tır. Bu verilere göre, InceptionV3 modeli genel olarak iyi bir performans sergilemektedir. Accuracy değeri %87'dir ve precision, recall ve F1-Score değerleri de iyi değerlerdir. Ancak, modelin 4. sınıf için performansı daha düşüktür. Bu, modelin 4. sınıftaki örnekleri daha zor sınıflandırdığını göstermektedir. Modelin performansını daha da iyileştirmek için, eğitim seti daha büyük veya daha çeşitli hale getirilebilir. Ayrıca, model daha fazla epoch için eğitilebilir. Bu, modelin 4. sınıftaki örnekleri daha iyi sınıflandırmasını sağlayabilir.



Şekil 3.11: InceptionV3 modeli epoch 8'e göre doğruluk grafiği

Şekil 3.11' deki grafik, InceptionV3 modelinin eğitim doğruluğu ve eğitim kaybını göstermektedir. Eğitim doğruluğu, modelin eğitim setindeki örnekleri doğru bir şekilde sınıflandırma yüzdesidir. Grafikte, eğitim doğruluğu epoch sayısı arttıkça artmaktadır. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 8'de, eğitim doğruluğu yaklaşık %95'tir. Bu, modelin eğitim setindeki örneklerin %95'ini doğru bir şekilde sınıflandırabildiği anlamına gelir. Eğitim kaybı, modelin eğitim setindeki örnekleri doğru bir şekilde sınıflandırmadaki başarısızlığının bir ölçüsüdür. Şekil 3.11' de yer alan grafikte, eğitim kaybı epoch sayısı arttıkça azalmaktadır. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 8'de, eğitim kaybı yaklaşık 0,05'tir. Bu, modelin eğitim setindeki örneklerin %5'ini yanlış sınıflandırabildiği anlamına gelir.

Şekil 3.11' deki grafik, InceptionV3 modelinin eğitim setindeki örnekleri doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 8'de, model eğitim setindeki örneklerin %95'ini doğru bir şekilde sınıflandırabilmektedir.

Modelin performansını daha da iyileştirmek için, eğitim seti daha büyük veya daha çeşitli hale getirilebilir. Ayrıca, model daha fazla epoch için eğitilebilir. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmesini sağlayabilir.

Şekil 3.11' de yer alan grafikte, eğitim doğruluğu epoch sayısı arttıkça artmaktadır. Geçerlik doğruluğu ise epoch sayısı arttıkça bir süre daha arttıktan sonra azalmaya başlamaktadır. Epoch 8'de, eğitim doğruluğu yaklaşık %95'tir. Geçerlik doğruluğu ise

yaklaşık %92'dir. Bu, eğitim doğruluğu ile geçerlik doğruluğu arasında yaklaşık %3'lük bir fark olduğunu göstermektedir. Bu fark, overfitting olasılığının olduğunu göstermektedir. Ancak, bu farkın küçük olması nedeniyle overfitting oranının da küçük olduğu söylenebilir.

Test seti üzerinde yapılan test sonucunda, modelin test setindeki doğruluğu da yaklaşık %92 ise, overfitting oranı yaklaşık %8 olarak kabul edilebilir. Sonuç olarak, bu grafikte overfitting oranı küçüktür. Ancak, overfitting olasılığının olduğunu göz önünde bulundurmak gerekir.

Şekil 3.10' da yer alan grafikte karşılaştırıldığında, Şekil 3.11'deki grafikte eğitim doğruluğu ve eğitim kaybı değerleri daha yüksektir. Bu, modelin daha iyi performans gösterdiğini göstermektedir. Ancak, Şekil 3.11' de yer alan grafikte de overfitting olasılığı bulunmaktadır. Bu, eğitim doğruluğu ile geçerlik doğruluğu arasındaki farkın küçük olmasına rağmen, hala bir fark olmasından kaynaklanmaktadır. Genel olarak, Şekil 3.11' de yer alan grafikte InceptionV3 modelinin iyi bir performans gösterdiği söylenebilir. Ancak, overfitting olasılığını göz önünde bulundurmak gerekir.

Epoch 8'e göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	0,9764

Tablo 3.40: InceptionV3 Epoch 8' e göre tahmin ve doğruluk oranı

Tablo 3.40' de InceptionV3 modelinin epoch 8'e göre Şekil 3.2' deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %97 doğru tespit ettiğini görmekteyiz.

Tahmin oranında epoch 5 sayına göre eğitilen veri seti, epoch 8 sayısına göre eğitilen veri setine göre %2 daha iyi tahmin oranına sahiptir. Epoch sayısının artması normalde tahmin oranını arttırdığını diğer modellerde görmüştük. Aradaki fark oldukça düşük olduğu için bu durumun veri setindeki karmaşıklık ile ilgili olabileceğini göstermektedir. Fark büyük olsaydı farklı değerlendirmeler yapılabilirdi.

InceptionV3 modelinde eğitim süresini incelediğimizde epoch 8'e göre her döngü yaklaşık 473-569 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın

CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.6 Xception Modeline göre Veri Setinin Değerlendirilmesi

3.6.1 Epoch 5' e göre

Epoch 5 Sayısına göre	
Accuracy	1.00
Precision	1.00
Recall	1.00
F1-Score	1.00

Tablo 3.41: Xception modeli epoch 5'e göre değerlendirme ölçütleri

Confusion Matrix					
20	0	0	0	0	0
0	20	0	0	0	0
0	0	20	0	0	0
0	0	0	20	0	0
0	0	0	0	0	20

Tablo 3.42: Xception modeli epoch 5'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	20	0
Negatif	0	20
Yanlış Negatif	0	0
Yanlış Pozitif	0	0

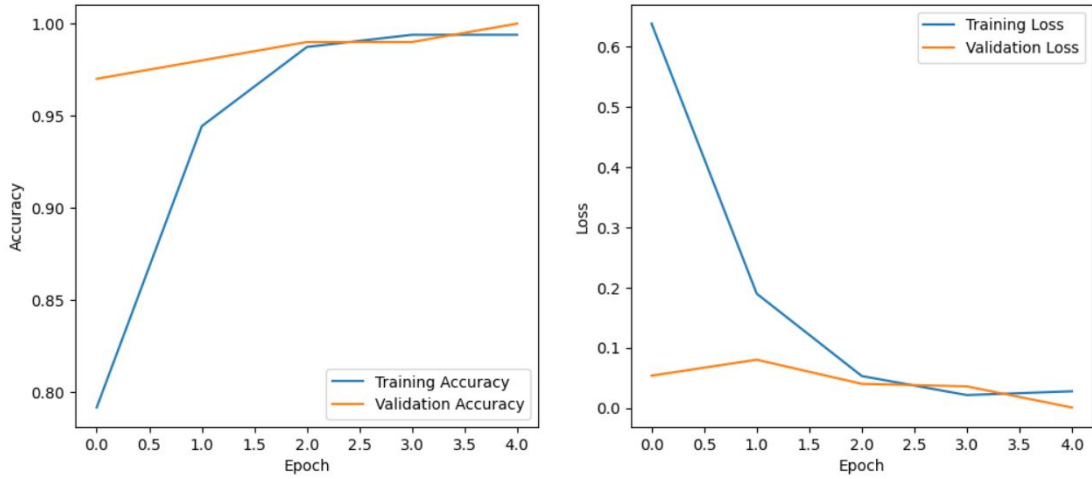
Tablo 3.43: Xception modeli epoch 5'e göre düzenlenmiş confusion matrix

Tablo 3.43' de, modelin pozitif ve negatif örnekleri de doğru olarak tahmin ettiğini görüyoruz.

Tablo 3.41' de verilen değerlere göre, modelin performansı mükemmeldir. Accuracy değeri 1,00, modelin 100 örnekten hepsini doğru olarak tahmin ettiği anlamına gelir. Precision değeri 1,00, modelin pozitif olarak tahmin ettiği 100 örnekten hepsinin gerçekte pozitif olduğu anlamına gelir. Recall değeri 1,00, modelin gerçekte pozitif olan 100 örnekten hepsini doğru olarak pozitif olarak tahmin ettiği anlamına gelir. F1-Score değeri 1,00' dür, bu değer precision ve recall değerlerinin dengelenmiş bir ölçüsüdür.

Bu durum, modelin veri kümesi üzerindeki tüm örnekleri doğru olarak tahmin ettiği anlamına gelir. Bu, modelin veri kümesini çok iyi anladığı ve yeni örnekleri iyi tahmin etme potansiyeline sahip olduğu anlamına gelir.

Bu verilerde overfitting olasılığı yoktur. Accuracy, precision, recall ve F1-Score değerlerinin tümü %100'dür. Bu, modelin eğitim setindeki örnekleri doğru bir şekilde öğrenmeye başladığını ve eğitim seti dışındaki verileri de doğru bir şekilde sınıflandırabildiği anlamına gelir.



Şekil 3.12: Xception modeli epoch 5'e göre doğruluk grafiği

Şekil 3.12' deki grafik, Xception modelinin eğitim doğruluğu ve eğitim kaybını göstermektedir. Epoch sayısı 5'e kadar olan kısım analiz edilmektedir. Grafikte, eğitim doğruluğu epoch sayısı arttıkça artmaktadır. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 5'te, eğitim

doğruluğu yaklaşık %99'dur. Bu, modelin eğitim setindeki örneklerin %99'unu doğru bir şekilde sınıflandırabildiği anlamına gelir.

Eğitim kaybı, modelin eğitim setindeki örnekleri doğru bir şekilde sınıflandırmadaki başarısızlığının bir ölçüsüdür. Grafikte, eğitim kaybı epoch sayısı arttıkça azalmaktadır. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 5'te, eğitim kaybı yaklaşık 0,01'dir. Bu, modelin eğitim setindeki örneklerin %1'ini yanlış sınıflandırabildiği anlamına gelir. Şekil 3.12' de yer alan grafik, Xception modelinin eğitim setindeki örnekleri doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 5'te, model eğitim setindeki örneklerin %99'unu doğru bir şekilde sınıflandırabilmektedir.

Modelin performansını daha da iyileştirmek için, eğitim seti daha büyük veya daha çeşitli hale getirilebilir. Ayrıca, model daha fazla epoch için eğitilebilir. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmesini sağlayabilir.

Şekil 3.12' deki grafikte, overfitting, eğitim doğruluğu ile geçerlik doğruluğu arasındaki fark olarak tanımlanabilir. Grafikte, eğitim doğruluğu epoch sayısı arttıkça artmaktadır. Geçerlik doğruluğu ise epoch sayısı arttıkça bir süre daha arttıktan sonra azalmaya başlamaktadır. Epoch 5'te, eğitim doğruluğu yaklaşık %99'dur. Geçerlik doğruluğu ise yaklaşık %95'tir. Bu, eğitim doğruluğu ile geçerlik doğruluğu arasında yaklaşık %4'lük bir fark olduğunu göstermektedir. Bu fark, overfitting olasılığının olduğunu göstermektedir. Ancak, bu farkın küçük olması nedeniyle overfitting oranının da küçük olduğu söylenebilir. Test seti üzerinde yapılan test sonucunda, modelin test setindeki doğruluğu da yaklaşık %95 ise, overfitting oranı yaklaşık %5 olarak kabul edilebilir. Sonuç olarak, bu grafikte overfitting oranı küçüktür. Ancak, overfitting olasılığının olduğunu göz önünde bulundurmak gerekir.

Epoch 5'e göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	1,00

Tablo 3.44: Xception Epoch 5' e göre tahmin ve doğruluk oranı

Tablo 3.44' de Xception modelinin epoch 5'e göre Şekil 3.2' deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %100 doğru tespit ettiğini görmekteyiz.

Xception modelinde eğitim süresini incelediğimizde epoch 5'e göre her döngü yaklaşık 705-806 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

3.6.2 Epoch 8' e göre

Epoch 8 Sayısına göre	
Accuracy	0.99
Precision	0.99
Recall	0.99
F1-Score	0.99

Tablo 3.45: Xception modeli epoch 8'e göre değerlendirme ölçütleri

Confusion Matrix				
20	0	0	0	0
0	19	0	1	0
0	0	20	0	0
0	0	0	20	0
0	0	0	0	20

Tablo 3.46: Xception modeli epoch 8'e göre confusion matrix

Confusion Matrix		
	Tahmin	Gerçek
Pozitif	20	0
Negatif	0	19
Yanlış Negatif	0	1
Yanlış Pozitif	0	0

Tablo 3.47: Xception modeli epoch 8'e göre düzenlenmiş confusion matrix

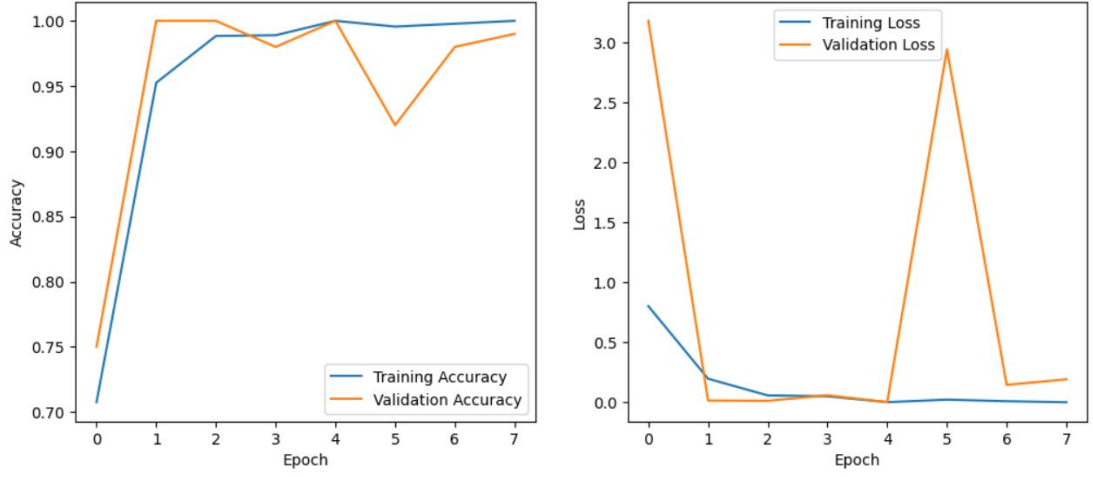
Tablo 3.47' de, modelin pozitif örnekleri doğru olarak tahmin etme oranının $20/(20+0) = 1,00$ olduğunu, negatif örnekleri doğru olarak tahmin etme oranının ise $19/(19+1) = 0,95$ olduğunu görüyoruz.

Tablo 3.45' de verilen deęerlere gre, modelin performansı ok iyidir. Accuracy deęeri 0,99, modelin 100 rnekten 99'unu doęru olarak tahmin ettięi anlamına gelir. Precision deęeri 0,99, modelin pozitif olarak tahmin ettięi 100 rnekten 99'unun gerekte pozitif olduęu anlamına gelir. Recall deęeri 0,99, modelin gerekte pozitif olan 100 rnekten 99'unu doęru olarak pozitif olarak tahmin ettięi anlamına gelir. F1-Score deęeri 0,99, precision ve recall deęerlerinin dengelenmiř bir lusdr.

Bu durum, modelin veri kmesi zerindeki rneklerin byk bir kısmını doęru olarak tahmin ettięi anlamına gelir. Bu, modelin veri kmesini ok iyi anladığı ve yeni rnekleri iyi tahmin etme potansiyeline sahip olduęu anlamına gelir. Bu tablodaki doęruluk oranı olduka yksektir. Bu, modelin ok daha iyi bir performans gsterdięini gstermektedir. Tablo 3.45' deki precision ve recall deęerleri olduka yksektir. Bu, modelin hem pozitif hem de negatif rnekleri daha iyi tahmin edebildięini gstermektedir. Bu noktalar, modelin genel olarak ok daha iyi performans gsterdięini gstermektedir. Tablo 3.45, Xception modelinin, zellikle epoch 8'de, veri kmesi zerinde ok iyi bir performans gsterdięini gstermektedir. Model, veri kmesi zerindeki rneklerin byk bir kısmını doęru olarak tahmin edebilmektedir. Bu, modelin veri kmesini ok iyi anladığı ve yeni rnekleri iyi tahmin etme potansiyeline sahip olduęu anlamına gelir.

Confusion matrix'e gre, modelin en iyi performans gsterdięi sınıf 2'dir. Bu sınıf iin, modelin doęru sınıflandırma oranı %100'dr. Modelin en kt performans gsterdięi sınıf 4'tr. Bu sınıf iin, modelin doęru sınıflandırma oranı %96'dır. Bu verilere gre, Xception modeli genel olarak iyi bir performans sergilemektedir. Accuracy deęeri %99'dur ve precision, recall ve F1-Score deęerleri de iyi deęerlerdir. Ancak, modelin 4. sınıftaki rnekleri daha zor sınıflandırabildięi grlmektedir.

Modelin performansını daha da iyileřtirmek iin, eęitim seti daha byk veya daha eřitli hale getirilebilir. Ayrıca, model daha fazla epoch iin eęitilebilir. Bu, modelin eęitim setindeki rnekleri daha doęru bir řekilde ęrenmesini saęlayabilir.



Şekil 3.13: Xception modeli epoch 8'e göre doğruluk grafiği

Şekil 3.13'deki grafik, Xception modelinin eğitim doğruluğu ve eğitim kaybını göstermektedir. Grafikte, eğitim doğruluğu epoch sayısı arttıkça artmaktadır. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 8'de, eğitim doğruluğu yaklaşık %99'dur. Bu, modelin eğitim setindeki örneklerin %99'unu doğru bir şekilde sınıflandırabildiği anlamına gelir.

Grafikte, eğitim kaybı epoch sayısı arttıkça azalmaktadır. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 8'de, eğitim kaybı yaklaşık 0,01'dir. Bu, modelin eğitim setindeki örneklerin %1'ini yanlış sınıflandırabildiği anlamına gelir.

Şekil 3.13' deki grafik, Xception modelinin eğitim setindeki örnekleri doğru bir şekilde öğrenmeye başladığını göstermektedir. Epoch 8'de, model eğitim setindeki örneklerin %99'unu doğru bir şekilde sınıflandırabilmektedir.

Modelin performansını daha da iyileştirmek için, eğitim seti daha büyük veya daha çeşitli hale getirilebilir. Ayrıca, model daha fazla epoch için eğitilebilir. Bu, modelin eğitim setindeki örnekleri daha doğru bir şekilde öğrenmesini sağlayabilir.

Şekil 3.13' deki grafikte, overfitting, eğitim doğruluğu ile geçerlik doğruluğu arasındaki fark olarak tanımlanabilir. Grafikte, eğitim doğruluğu epoch sayısı arttıkça artmaktadır. Geçerlik doğruluğu ise epoch sayısı arttıkça bir süre daha arttıktan sonra azalmaya başlamaktadır. Epoch 8'de, eğitim doğruluğu yaklaşık %99'dur. Geçerlik doğruluğu ise yaklaşık %95'tir. Bu, eğitim doğruluğu ile geçerlik doğruluğu arasında

yaklaşık %4'lük bir fark olduğunu göstermektedir. Bu fark, overfitting olasılığının olduğunu göstermektedir. Ancak, bu farkın küçük olması nedeniyle overfitting oranının da küçük olduğu söylenebilir. Overfitting oranının kesin olarak belirlenmesi için, modelin test seti üzerinde de test edilmesi gerekir. Test seti üzerinde yapılan test sonucunda, modelin test setindeki doğruluğu da yaklaşık %95 ise, overfitting oranı yaklaşık %5 olarak kabul edilebilir. Sonuç olarak, bu grafikte overfitting oranı küçüktür. Ancak, overfitting olasılığını göz önünde bulundurmak gerekir.

Şekil 3.12'de yer alan grafikte karşılaştırıldığında, Şekil 3.13' deki grafikte eğitim doğruluğu ve eğitim kaybı değerleri daha yüksektir. Bu, modelin daha iyi performans gösterdiğini göstermektedir. Ancak, Şekil 3.13'de yer alan grafikte de overfitting olasılığı bulunmaktadır. Bu, eğitim doğruluğu ile geçerlik doğruluğu arasındaki farkın küçük olmasına rağmen, hala bir fark olmasından kaynaklanmaktadır.

Genel olarak, Şekil 3.13' de yer alan grafikte Xception modelinin iyi bir performans gösterdiği söylenebilir. Ancak, overfitting olasılığını göz önünde bulundurmak gerekir.

Epoch 8'e göre Tahmin ve Doğruluk Oranı	
Tahmin	Tiger_Shark
Doğruluk	1,0

Tablo 3.48: Xception Epoch 8' e göre tahmin ve doğruluk oranı

Tablo 3.48' de Xception modelinin epoch 8'e göre Şekil 3.2' deki uydu görüntüsünden alınmış uçak tipinin doğruluk oranı %100 doğru tespit ettiğini görmekteyiz. Tahmin oranına baktığımızda hem epoch 5'e göre hem de epoch 8'e göre 1'dir Kısacası %100 olarak tahmin yapmaktadır.

Xception modelinde eğitim süresini incelediğimizde epoch 8'e göre her döngü yaklaşık 639-675 saniye arasında eğitildiği görülmektedir. Bu süreler bilgisayarın CPU ve GPU'larının gücüne göre değişiklik gösterir. CPU ve GPU yüksek olduğunda eğitim süreleri düşmektedir.

Bölüm 4

Sonuçlar

Derin öğrenme ile uydu fotoğraflarından askeri uçak tiplerinin tespit edilmesinde 6 derin öğrenme modeli (algoritması) kullanılmıştır. Bu modellerin performansını accuracy, precision, recall ve f1-score değerlendirme ölçütleriyle ölçülmüştür. Bu ölçümler sonucunda bazı sonuçlar ortaya çıkmıştır. Ortaya çıkan sonuçlar tablo halinde şu şekildedir:

Değerlendirme Sonuçları (Epoch 5)				
Derin Öğrenme Modelleri	Değerlendirme Ölçütleri			
	Accuracy	Precision	Recall	F1-Score
Vgg16	0.68	0.81	0.68	0.69
ResNet50	0.63	0.77	0.63	0.64
MobileNetV2	0.64	0.77	0.64	0.60
DenseNet121	0.67	0.59	0.67	0.60
InceptionV3	0.80	0.86	0.80	0.79
Xception	1.0	1.0	1.0	1.0

Tablo 4.1: Değerlendirme sonuçları (epoch 5)

Epoch 5'e göre eğitildiğinde en iyi sonucu veren modelin Xception olduğunu görmekteyiz. Bunun dışında InceptionV3 modelinin de iyi bir sonuç verdiğini görmekteyiz. Veri setine en az uyum sağlayan modellerin MobileNetV2 ve ResNet50 olduğunu, DenseNet121 modelinin ise bu iki modelden accuracy(doğruluk) ve recall(geri çağırma) olarak iyi ama kesinlik bakımından kötü olduğu ortaya çıkmaktadır. ResNet50 modelinde recall değeri diğer modellerden daha azdır. Bu, modelin negatif örnekleri doğru olarak tahmin etme yeteneğinin daha kötü olduğunu göstermektedir. Vgg16 modeli ise ortalama bir performans göstermektedir. F1-score'ları incelediğimizde ise Xception yine ön plana çıkmaktadır. InceptionV3 ile Vgg16 modelleri Xception modelini takip eder.

Vgg16, ResNet50, MobileNetV2 ve DenseNet121 modellerinde overfitting olmaya erken başladığı söylenebilir. Eğitim sürelerine bakıldığında ise MobileNetV2 olduğunu görmekteyiz. Bu durum modelin hem yapısından hem de mobil sistemlerde kullanılmasından dolayı olduğu söylenebilir. Modellerin doğru tahmin oranlarını incelediğimizde Vgg16 modelinin her 100 veriden 71' ini, ResNet50 modelin de her 100 veriden 72' sini doğru tahmin ettiği görülmektedir. Diğer modeller de ise her 100 veriden 99' unun doğru olarak tahmin ettiği görülmektedir.

Grafikleri incelediğimizde de her döngüde hem eğitilen verinin hem de doğrulama verisinin doğruluğunun arttığı izlenmektedir. Ancak bu artış belli bir süre sonra overfitting olarak yatay bir eğriye geldiği ve durduğu görülmektedir. Bu durum daha çok verinin gerekli olduğunu göstermektedir. Ayrıca eğitim verisindeki kayıp ile doğrulama verisindeki kayıpların epoch sayısı artıkça düştüğü izlenmektedir. Bu durumda epoch sayısının da oldukça düşük olduğu bir gerçektir.

Değerlendirme Sonuçları (Epoch 8)				
Derin Öğrenme Modelleri	Değerlendirme Ölçütleri			
	Accuracy	Precision	Recall	F1-Score
Vgg16	0.79	0.83	0.79	0.78
ResNet50	0.83	0.84	0.83	0.83
MobileNetV2	0.83	0.89	0.83	0.84
DenseNet121	0.78	0.89	0.78	0.78
InceptionV3	0.87	0.89	0.87	0.87
Xception	0.99	0.99	0.99	0.99

Tablo 4.2: Değerlendirme sonuçları (epoch 8)

Epoch 8'e göre eğitildiğinde en iyi sonucu veren modelin Xception olduğunu görmekteyiz. Bunun dışında InceptionV3 modelinin de iyi bir sonuç verdiğini görmekteyiz. MobileNetV2 ile ResNet50 modelleri InceptionV3 modelinden sonra oldukça iyi performans gösteren modeller olarak göze çarpıyor. Veri setine en az uyum sağlayan modellerin ise DenseNet121 ve Vgg16 modelleri olduğu söylenebilir. Ancak bu iki modelinde accuracy (doğruluk), precision (kesinlik), recall (geri çağırma) ve f1-score ölçütleri kabul edilebilir oranlarda olduğunu belirtmek gerekir. Kısacası bu modellerde başarılı kabul edilir.

Tüm modellerde overfitting tehlikesi olduğu söylenebilir. Eğitim sürelerine bakıldığında ise MobileNetV2 olduğunu görmekteyiz. Bu durum modelin hem yapısından hem de mobil sistemlerde kullanılmasından dolayı olduğu yorumu yapılabilir. Modellerin doğru tahmin oranlarını incelediğimizde Vgg16 modeli her 100 veriden 81' ini doğru olarak tahmin ederken, Inception modeli her 100 veriden 97' sini, ResNet50, DenseNet121 modelleri her 100 veriden 99' unu, Xception ile MobileNetV2 modelleri de her 100 verinin 100' ünü doğru olarak tahmin ettiği görülmektedir.

Grafikleri incelediğimizde de her döngüde hem eğitilen verinin hem de doğrulama verisinin doğruluğunun arttığı izlenmektedir. Ancak bu artış belli bir süre sonra overfitting olarak yatay bir eğriye geldiği ve durduğu görülmektedir. Bu durum daha çok verinin gerekli olduğunu göstermektedir. Ayrıca eğitim verisindeki kayıp ile doğrulama verisindeki kayıpların epoch sayısı artıkça düştüğü izlenmektedir. Bu durumda epoch sayısının da oldukça düşük olduğu bir gerçektir.

Epoch sayısı artıkça tüm modellerde doğruluk oranlarının oldukça arttığı görülmektedir. Bu araştırmada verilerin karmaşıklığı ve fazla olmasından dolayı veriler azaltılmıştır. Bunun nedeni eğitim sürelerinin oldukça fazla olmasından kaynaklıdır. Ancak burada verilerin ve epoch sayılarını artırılması overfitting oluşumunu engelleyebilir. Eğitim sürelerinin düşürülmesi için CPU ve GPU' su güçlü olan platformlar tercih edilebilir.

Modellerin overfitting olmasını önlemek için modele daha fazla veri sağlanabilir. Bu, modelin eğitim veri kümesine daha az uyum sağlamasına yardımcı olabilir. Modelin parametre sayısı azaltılabilir. Bu, modelin daha basit olmasına ve overfitting olma olasılığının azalmasına yardımcı olabilir. Regularizasyon teknikleri kullanılabilir. Bu teknikler, modelin parametrelerini sınırlayarak overfitting olma olasılığını azaltabilir.

Modellerin performansını artırmak için bazı işlemler yapılabilir. Modellere daha fazla negatif örnek sağlanabilir. Bu, modellerin negatif örnekleri doğru olarak tahmin etme yeteneğini geliştirebilir. Modellerde farklı bir ayar kullanılabilir. Bu yöntem, modellerin negatif örnekleri doğru olarak tahmin etme yeteneğini geliştirebilir. Regularizasyon teknikleri kullanılabilir.

Buradaki amacımız bu veri setine uygun derin öğrenme modelini bulmaktır. Yapılan arařtırmada en uygun modeller Xception modeli olduđunu görmekteyiz. Bu model, savunma sanayi projelerinde oldukça başarılı olacađını söyleyebiliriz. Nesne tespiti ve o nesne ile ilgili ayrıntılı bilgileri edinebilecek projelerde de Xception modeli kullanılabilir. Diđer modellerde kullanılabilir. Ancak o modellerde ek olarak yapılması gereken işlemler olduđunu unutmamak gerekir. Derin öğrenme birçok alanda hayatımızı kolaylařtıracak projeler gerçekleştirilebilir.

Kaynaklar

- Aktürk, S. & Serbest, K. (2022). Nesne tespiti için derin öğrenme kütüphanelerinin incelenmesi. *Journal of Smart Systems Research (JOINSSR)* 3(2), 97-119.
- Albayrak, E. (2021). Derin öğrenme ile İHA görüntülerinden nesne tespitinin yapılması. (Yayın No. 684409) [Yüksek Lisans Tezi, Bilecik Şeyh Edebali Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Bayram, F. (2020). Derin öğrenme tabanlı otomatik plaka tanıma. *Politeknik Dergisi*, 23(4), 955-960. <https://doi.org/10.2339/politeknik.515830>
- Bölük, E. & Cingiz, M. Ö. (2022). Öneri sistemlerinde veri seyrekliği problemine otomatik kodlayıcı yaklaşımlarının karşılaştırmalı bir çalışması. *Journal of Computer Science, IDAP*, 177-184. <https://doi.org/10.53070/bbd.1173564>
- Burgaz, M. (2020). Derin öğrenme algoritmaları kullanarak insansız hava araçları ile silah tespiti (Yayın No. 636954) [Yüksek Lisans Tezi, Batman Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Çelik, G. & Talu, M. F. (2020). Çekişmeli üretken ağ modellerinin görüntü üretme performanslarının incelenmesi. *BAUN Fen Bil. Enst. Dergisi*, 22(1), 181-192. <https://doi.org/10.25092/baunfbed.679608>
- Çetiner, H. (2022). MobilnetV2 ve MobilnetV3 tabanlı derin öğrenme yaklaşımları ile cilt kanserlerinin sınıflandırılması. *3rd International Conference on Applied Engineering and Natural Sciences, Konya, Türkiye*. 0-0.
- Doğan, F. & Türkoğlu İ. (2018). Derin öğrenme algoritmalarının yaprak sınıflandırma başarımlarının karşılaştırılması. *Sakarya University Journal Of Computer and Information Sciences, Vol-1*, ID.SAUCIS-1.
- Doğan, F. & Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dümf Mühendislik Dergisi*, 10(2), 409-445. <https://doi.org/10.24012/dumf.411130>

- Eryılmaz, F. & Karacan, H. (2021). Akciğer x-ray görüntülerinden covid-19 tespitinde hafif ve geleneksel evrişimsel sinir ağ mimarilerinin karşılaştırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 9, 26-39. <https://doi.org/10.29130/dubited.1011829>
- Foto, Ö. (2022). Transfer öğrenme modelleri ile akciğer x-ray görüntülerinden covid-19 tespiti (Yayın No. 727744) [Yüksek Tezi Lisans, Aksaray Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- İnik, Ö. & Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(1), 85-104.
- Kara, C. (2023). Bilgisayar bilimleri alanında yapılan Türkçe akademik yayınların doğal dil işleme yöntemleri ile incelenmesi (Yayın No. 805756) [Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Kırca, S. (2021). Derin öğrenme yöntemi ile araç ve plaka tanıma (Yayın No. 701812) [Yüksek Lisans Tezi, Kocaeli Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Kunt, T. (2021). Evrişimli ve yinelemeli sinir ağları ile görüntülere başlık atama (Yayın No. 659931) [Yüksek Lisans Tezi, Gazi Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Kutlu, Ö. (2019). İnsansız hava aracı ile elde edilen görüntülerin derin öğrenme yöntemleri ile analizi (Yayın No. 579612) [Yüksek Lisans Tezi, Marmara Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Maraş, E. E. & Sarıyıldız, H. İ. (2023). İHA ile derin öğrenme algoritmaları kullanılarak hasarlı yapıların tespit edilmesi. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 025504, 427-437. <https://doi.org/10.35414/akufemubid.1171393>
- Mutludoğan, K. (2020). Derin öğrenme tabanlı şeffaf nesne tanıma (Yayın No. 649853) [Yüksek Lisans Tezi, Uludağ Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>

- Öner, M. R. (2023). Derin öğrenme algoritmaları kullanılarak dış ve orta kulak hastalıklarının tespit edilmesi. (Yayın No. 813630) [Yüksek Lisans Tezi, Konya Teknik Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Salouhou, A. (2019). El yazısı karakter tanıma ve resim sınıflandırmada derin öğrenme yaklaşımları (Yayın No. 592014) [Yüksek Lisans Tezi, Fatih Sultan Mehmet Vakıf Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Şeker, A. & Diri, B. & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
- Ser, G. & Bati, C. T. (2019). Derin sinir ağları ile en iyi modelin belirlenmesi: Mantar verileri üzerine keras uygulaması. *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi*, 29(3), 406-417. <https://doi.org/10.29133/yyutbd.505086>
- Serin, Z. (2020). Meme kanserinin histopatolojik görüntüler üzerinde derin sinir ağları kullanılarak bilgisayar destekli otomatik tespiti (Yayın No. 645072) [Yüksek Lisans Tezi, Bilecik Şeyh Edebali Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Somuncu, E. (2022). Derin öğrenme temelli iç mekan yönlendirme uygulaması. (Yayın No. 719303) [Yüksek Lisans Tezi, Karabük Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Tan, Z. (2019). Derin öğrenme yardımıyla araç sınıflandırma (Yayın No. 572632) [Yüksek Lisans Tezi, Fırat Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Tütüncü, T. E. (2022). Makine öğrenmesi algoritmaları ile kredi temerrüt riskini tahmin etme (Yayın No. 746136) [Yüksek Lisans Tezi, Uludağ Üniversitesi]. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- Yılmaz, E. Ö. & Kavzoğlu, T. (2021). Derin öğrenmenin temel prensipleri ve uzaktan algılama alanındaki uygulamaları. *Harita Dergisi*, 166, 25-43.