



**DERİN ÖĞRENME MİMARİLERİ KULLANILARAK DRONE İLE  
ASKERİ ARAÇ VE SİLAHLARIN SINIFLANDIRILMASI**

**Doğın EROL**

**YÜKSEK LİSANS TEZİ  
SAVUNMA TEKNOLOJİLERİ ANA BİLİM DALI**

**SİVAS BİLİM VE TEKNOLOJİ ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**ARALIK 2023**

Dođan EROL tarafından hazırlanan “DERİN ÖĐRENME MİMARİLERİ KULLANILARAK DRONE İLE ASKERİ ARAÇ VE SİLAHLARIN SINIFLANDIRILMASI ” adlı tez çalışması aşıđıdaki jüri tarafından OY BİRLİĐİ ile Sivas Bilim ve Teknoloji Üniversitesi Savunma Teknolojileri Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

**Danışman:** Doç. Dr. Kemal ADEM

Savunma Teknolojileri Ana Bilim Dalı, Sivas Bilim ve Teknoloji Üniversitesi .....

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum.

**Başkan:** Dr. Öğr. Üyesi Sibel ARSLAN

Yazılım Mühendisliği Ana Bilim Dalı, Sivas Cumhuriyet Üniversitesi .....

Bu tezin, kapsam ve kalite olarak Yüksek Lisans olduğunu onaylıyorum/onaylamıyorum.

**Üye:** Doç. Dr. Kemal ADEM

Savunma Teknolojileri Ana Bilim Dalı, Sivas Bilim ve Teknoloji Üniversitesi .....

Bu tezin, kapsam ve kalite olarak Yüksek Lisans olduğunu onaylıyorum/onaylamıyorum.

**Üye:** Dr. Öğr. Üyesi Melih YILMAZ

Savunma Teknolojileri Ana Bilim Dalı, Sivas Bilim ve Teknoloji Üniversitesi .....

Bu tezin, kapsam ve kalite olarak Yüksek Lisans olduğunu onaylıyorum/onaylamıyorum.

Tez Savunma Tarihi: 27/12/2023

Jüri tarafından kabul edilen bu çalışmanın Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum

.....

Dr. Öğr.Üyesi Ali Suat YILDIZ

Lisansüstü Eğitim Enstitüsü Müdürü

## ETİK BEYAN

Sivas Bilim ve Teknoloji Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

.....  
Doğan EROL  
27/12/2023

# DERİN ÖĞRENME MİMARİLERİ KULLANILARAK DRONE İLE ASKERİ ARAÇ VE SİLAHLARIN SINIFLANDIRILMASI

(Yüksek Lisans Tezi)

Doğan EROL

SİVAS BİLİM ve TEKNOLOJİ ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Aralık 2023

## ÖZET

Bu çalışmada derin öğrenme yöntemleri kullanılarak eğitilen model ile askeri araçları, silahlı-silahsız kişileri havadan otonom uçuşlar yaparak tespit edebilen bir drone tasarlanmıştır. Nesneleri sınıflandırmak için transfer öğrenme mimarilerinden ResNet50, InceptionV3, Xception, Mobilenet, EfficientNet gibi 5 farklı transfer öğrenme mimarisi ile çalışılmış olup performans karşılaştırmaları yapılmıştır. Ayrıca nesne tespit işlemleri için YOLOv8 ve YOLO-NAS mimarilerinin performansları karşılaştırılmıştır. Mimariler için askeri ve sivil araçlar, sivil ve silahlı insan görüntülerinden oluşan 4195 adet görsel içeren 6 sınıflı yeni bir hibrit veri seti hazırlanmıştır. Modellerin aşırı öğrenmemesi için veri setindeki görüntülere veri çoğaltma yöntemlerinden rastgele yaklaştırma, yatay yönde rastgele kaydırma, dikey yönde rastgele kaydırma, yatay döndürme, belirli açıyla döndürme ve yatay kayma işlemleri uygulanmıştır. Çalışmada transfer öğrenme mimarilerindeki en başarılı sonuçlar 0,9824 doğruluk, 0,9802 F1-skoru değeri ile ResNet50 mimarisi ile elde edilmiştir. Nesne tespit işlemlerinde en yüksek skorlar YOLOv8 nesne tespit algoritması ile 0,961 mAP50, 0,9116 F1-skoru olarak elde edilmiştir. Tasarlanan İHA üzerinde yapılan gerçek zamanlı çalışma testlerinde ResNet50 mimarisi YOLOv8'e göre daha düşük performans gösterdiğinden İHA üzerine YOLOv8 ile oluşturulan model eklenmiştir. Veri setine havadan görüntüleri çekilmiş daha fazla askeri araç görüntüsü eklenerek veriler çoğaltılabilir. Veri setindeki her sınıf kendi arasında özelleştirilerek farklı sınıflara ayrılabilir.

Bilim Kodu : [Bilim Kodu]

Anahtar Kelimeler : Derin Öğrenme, Drone, Nesne Tespiti, YOLOv8, Transfer Öğrenme, YOLONAS

Sayfa Adedi : 64

Danışman : Doç.Dr. Kemal ADEM

# CLASSIFICATION OF MILITARY VEHICLES AND WEAPONS USING DEEP LEARNING ARCHITECTURES WITH DRONES

(M. Sc. Thesis)

Dođan EROL

SİVAS UNIVERSITY OF SCIENCE AND TECHNOLOGY  
INSTITUTE OF GRADUATE STUDIES

December 2023

## ABSTRACT

In this study, a drone that can detect military vehicles and armed and unarmed people by making autonomous flights from the air was designed with the model trained using deep learning methods. To classify objects, 5 different transfer learning architectures such as ResNet50, InceptionV3, Xception, Mobilenet, EfficientNet were studied and performance comparisons were made. Additionally, the performances of YOLOv8 and YOLO-NAS architectures were compared for object detection. A new 6-class hybrid dataset was prepared for architectures, containing 4195 images of military and civilian vehicles, civilian and armed people. In order to prevent the models from overfitting, data multiplication methods such as random approximation, horizontal random shift, vertical random shift, horizontal rotation, rotation at a certain angle and horizontal shift were applied to the images in the data set. In the study, the most successful results in transfer learning architectures were obtained with the ResNet50 architecture with an accuracy of 0,9824 and an F1-score value of 0,9802. The highest scores in object detection were obtained with the YOLOv8 object detection algorithm as 0,961 mAP50 and 0,9116 F1-score. Since the ResNet50 architecture showed lower performance than YOLOv8 in the real-time operating tests performed on the designed UAV, the model created with YOLOv8 was added to the UAV. The data can be increased by adding more aerial images of military vehicles to the data set. Each class in the data set can be customized and divided into different classes.

Science Code : [Science Code]

Key Words : Deep learning, Drone, Object Detection, YOLOv8, Transfer Learning, YOLONAS

Page Number : 64

Supervisor : Assoc.Prof.Dr. Kemal ADEM

## TEŞEKKÜR

Tez çalışmam sürecinde engin bilgilerinden yararlandığım, yeni alanları keşfetmemi sağlayan, farklı alanlara yönlendiren ve manevi desteğini hiçbir zaman esirgemeyen saygıdeğer danışmanım Doç. Dr. Kemal ADEM'e; yüksek lisanstaki eğitim sürecimde bilgilerime bilgi katan sayın Dr. Öğretim üyesi Halit BAKIR'a; beni her zaman motive eden, iyi hissettiren ve destekleyen eşime en içten dileklerle teşekkür ederim.

Aralık 2023

Doğan EROL



## İÇİNDEKİLER

	Sayfa
ÖZET .....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ .....	xi
RESİMLERİN LİSTESİ .....	xiii
SİMGELER VE KISALTMALAR.....	xiv
1. GİRİŞ .....	1
2. KAYNAK ARAŞTIRMASI .....	4
3. MATERYAL ve YÖNTEM.....	9
3.1. Kullanılan Veri Setinin Özellikleri .....	9
3.2. Yapay Sinir Ağı .....	10
3.3. Evrişimli Sinir Ağları ve Modelleri .....	12
3.3.1 Evrişimli sinir ağları katmanları .....	13
3.3.2 Transfer öğrenme (Transfer Learning) .....	19
3.3.3 ResNet50 (Residual network) mimarisi .....	20
3.3.4 MobileNet mimarisi.....	21
3.3.5 EfficientNet mimarisi .....	22
3.3.6 InceptionV3 mimarisi .....	24
3.3.7 Xception ağı mimarisi .....	26
3.3.8 YOLO mimarisi.....	27
3.4. Performans Metrikleri .....	31

3.5.	Modellerin Eğitilmesi .....	32
3.5.1	Öğrenme oranı (Learning rate) .....	33
3.5.2	Maksimum devir sayısı (Epok).....	33
3.5.3	Yığın boyutu (Batch Size) .....	33
3.6.	Yapılan İHA.....	34
3.6.1	İHA Gövde Tasarımı ve Üretimi .....	34
3.7.	Sistem Çalışma Prensibi.....	37
4.	BULGULAR ve TARTIŞMA.....	38
4.1.	Sınıflandırma Modelleri ile Gerçekleştirilen Analiz Sonuçları .....	38
4.1.1.	ResNet50 Modeli ile Gerçekleştirilen Analiz Sonuçları .....	39
4.1.2.	Inceptionv3 Modeli ile Gerçekleştirilen Analiz Sonuçları.....	40
4.1.3.	Xception Modeli ile Gerçekleştirilen Analiz Sonuçları .....	42
4.1.4.	MobileNet Modeli ile Gerçekleştirilen Analiz Sonuçları .....	44
4.1.5.	EfficientNet Modeli ile Gerçekleştirilen Analiz Sonuçları .....	45
4.1.6.	Modellerin Performans ve Özellik Karşılaştırması.....	47
4.2.	Nesne Tespit Algoritması YOLO ile Gerçekleştirilen Analiz Sonuçları .....	47
4.3	Gerçek Zamanlı Nesne Tespiti ve Sınıflandırma Performans Karşılaştırması .....	51
5.	SONUÇ VE ÖNERİLER .....	52
	KAYNAKLAR .....	54

## ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Tablo 2.1 Askeri alana yönelik çalışmalar.....	7
Tablo 2.2 Sivil alana yönelik çalışmalar.....	7
Tablo 2.3 Sağlık alanına yönelik çalışmalar.....	8
Tablo 3.1. Biyolojik ve yapay sinir sistemlerinin karşılaştırılması.....	11
Tablo 3.2. EfficientNet temel ağ yapısı.....	24
Tablo 3.3. Karışıklık matrisi.....	31
Tablo 4.1 Eğitim, test ve doğrulama görüntü verileri sayıları.....	38
Tablo 4.2 Derin öğrenme ve Nesne tespit mimarilerinde kullanılan parametreler.....	38
Tablo 4.3 Modellere girilen görüntü boyutları.....	39
Tablo 4.4 ResNet50 mimarisinin sınıf bazında performansı.....	39
Tablo 4.5 ResNet50 mimarisi doğruluk skoru.....	39
Tablo 4.6 InceptionV3 mimarisinin sınıf bazında performansı.....	41
Tablo 4.7 InceptionV3 mimarisi doğruluk skoru.....	41
Tablo 4.8 Xception mimarisinin sınıf bazında performansı.....	42
Tablo 4.9 Xception mimarisi doğruluk skoru.....	42
Tablo 4.10 MobileNet mimarisinin sınıf bazında performansı.....	44
Tablo 4.11 MobileNet mimarisi doğruluk skoru.....	44
Tablo 4.12 EfficientNet mimarisinin sınıf bazında performansı.....	45
Tablo 4.13 EfficientNet mimarisi doğruluk skoru.....	45
Tablo 4.14 Mimarilerin performans metrikleri.....	47
Tablo 4.15 Mimarilerin sınıf bazındaki performansları.....	47
Tablo 4.16 YOLOv8 performans skorları.....	47

<b>Çizelge</b>	<b>Sayfa</b>
Tablo 4.17 YOLO-NAS mimarisi performans skorları .....	50
Tablo 4.18 ResNet50 ile YOLOv8 performans metrikleri .....	51



## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Veri setine ait sınıflardaki görsellerin sayıları.....	9
Şekil 3.2. Veri setindeki sınıflara ait görseller.....	10
Şekil 3.3 İnsan beyin nöronu ve yapay sinir ağı yapısının karşılaştırılması.....	11
Şekil 3.4. Çok katmanlı yapay sinir ağı yapısı.....	11
Şekil 3.5. Yapay sinir hücresi yapısı.....	12
Şekil 3.6. ESA yapısı ve katmanları.....	13
Şekil 3.7. Evrişim işlemi.....	14
Şekil 3.8. Havuzlama işlemleri.....	15
Şekil 3.9. Düzleştirme işlemi.....	16
Şekil 3.10. İki katmanda tam bağlı katman örneği.....	16
Şekil 3.11. Relu aktivasyon fonksiyonu grafiği.....	17
Şekil 3.12. Sigmoid aktivasyon fonksiyonu grafiği.....	18
Şekil 3.13. Softmax aktivasyon fonksiyonu grafiği.....	18
Şekil 3.14. Aktarmalı öğrenme.....	20
Şekil 3.15. Artık yapı blokları (a)RBB-1 bloğu (b)RBB-2 bloğu.....	20
Şekil 3.16. ResNet50 mimarisi.....	21
Şekil 3.17. MobileNetV2 mimarisi.....	22
Şekil 3.18. EfficientNet mimarisi.....	23
Şekil 3.19. 3x1 mini ağ yapısı.....	24
Şekil 3.20. İki katmanlı 3x3'lük evrişim katmanı.....	25
Şekil 3.21. Izgara boyutunu azaltma prensipleri (temsili darboğaz).....	25
Şekil 3.22. Önerilen darboğaz yapısı.....	26

<b>Şekil</b>	<b>Sayfa</b>
Şekil 3.23. Xception ağ mimarisi.....	27
Şekil 3.24. Kesişim ve birleşim alanları .....	28
Şekil 3.25. YOLOv8 ağ mimarisi .....	29
Şekil 3.26 (a)YOLOv8 C2f modülü yapısı, (b) darboğaz yapısı .....	30
Şekil 3.27. YOLO-NAS Mimarisi, QSP ve QCI blokları.....	31
Şekil 3.28. X tipindeki dronun motor dönüş yönleri .....	35
Şekil 3.29 Sistem çalışma prensibi şeması .....	37
Şekil 4.1 ResNet50 epok sayısına göre doğruluk ve kayıp değeri eğrileri .....	40
Şekil 4.2 ResNet50 mimarisi karışıklık matrisi sonuçları .....	40
Şekil 4.3 InceptionV3 epok sayısına göre doğruluk ve kayıp değeri eğrileri .....	41
Şekil 4.4 InceptionV3 mimarisi karışıklık matrisi sonuçları .....	42
Şekil 4.5 Xception epok sayısına göre doğruluk ve kayıp değeri eğrileri .....	43
Şekil 4.6 Xception mimarisi karışıklık matrisi sonuçları.....	43
Şekil 4.7 MobileNet epok sayısına göre doğruluk ve kayıp değeri eğrileri.....	44
Şekil 4.8 MobileNet mimarisi karışıklık matrisi sonuçları .....	45
Şekil 4.9 EfficientNet epok sayısına göre doğruluk ve kayıp değeri eğrileri .....	46
Şekil 4.10 EfficientNet mimarisi karışıklık matrisi sonuçları .....	46
Şekil 4.11 F1 skoru güven eğrisi.....	48
Şekil 4.12 Duyarlık kesinlik eğrisi.....	48
Şekil 4.13 Epok sayısına göre elde edilen performans metrikleri .....	49
Şekil 4.14 Görüntüler üzerinde yapılan nesne tespit işlemi örnekleri .....	49
Şekil 4.15 YOLOv8 karışıklık matrisi sonuçları .....	50
Şekil 4.16. YOLO-NAS-M mimarisi mAP50 skor grafiği .....	50
Şekil 4.17. YOLO-NAS-M F1-Skor grafiği .....	51

**Şekil****Sayfa****RESİMLERİN LİSTESİ****Resim****Sayfa**

Resim 3.1. Yapılan İHA.....34

Resim 3.2. Katmanlı üretim aşaması .....35

Resim 3.3. Mission planner yer istasyonu programı.....37

Resim 4.4. İHA ile gerçek zamanlı yapılan tespitler .....51



## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

### Simgeler

**A**

### Açıklamalar

Amper

**kV**

Boş yükte dakikadaki tur sayısı

**V**

Volt

### Kısaltmalar

**AP**

### Açıklamalar

Average Precision (Ortalama Kesinlik)

**ESA**

Evrişimli Sinir Ağı

**FN**

False-Negative (Yanlış-Negatif Değer)

**FP**

False-Positive (Yanlış-Pozitif Değer)

**FPS**

Frame Per Second

**İHA**

İnsansız Hava Aracı

**mAP**

Mean Average Precision (Ortalama Hassasiyet)

**RELU**

Rectified Linear Unit (Doğrusal Birim Katmanı)

**TN**

True-Negative (Doğru-Negatif Değer)

**TP**

True-Positive (Doğru-Pozitif Değer)

**YOLO**

You Only Look Once

**IOU**

Intersection Over Union

|

## 1. GİRİŞ

Savunma sanayii, temelleri teknoloji üzerine atılan bir sektördür. Hızla gelişen teknolojiler sayesinde savunma sanayisi teknolojiye iyice odaklanır olmuştur. Savunma sanayiinde hızla değişen yeni teknolojilerin yakından takip edilmesi, rekabetçi ortam için sürekli geliştirilmesi gerekmektedir. Günümüzde otonom sistemler sağlık, sanayii, tarım gibi birçok alanda kullanılmaktadır (Yıldız, 2017; Aktaş,2022). Otonom sistemlerin en yaygın kullanıldığı alanlarından biri de askeri ve savunma teknolojileri alanıdır. Savunma sanayiinin geliştirilmesi ve güçlendirilmesinde yeni teknolojilerin önemi bir hayli büyüktür (Beşdok vd., 2009). Son zamanlarda savunma sanayisi için geliştirilen yeni teknolojiler incelendiğinde otonom sistemler, insansız araçlar ve yapay zekanın kullanımının arttığı gözlemlenmektedir. Otonom sistemlerin gereksinim duyduğu ihtiyaçlar hem insansız platformlarda hem de insanlı platformlarda yeni teknolojiler ile karşılanmaya başlanmıştır. Otonom sistemler ile verilecek önemli karar aşamalarında ve olumlu sonuçlar elde edilmek istenildiğinde etki durumlarındaki insan müdahalesi azaltılarak daha doğru sonuçlar elde edilebilmektedir. Otonom sistemler üzerindeki güvenlik kontrol sistemlerinin performansının artırılması için derin öğrenme yöntemleri kullanılmaktadır (Şeker vd., 2017).

Derin öğrenme yöntemleri ile görüntüler ve videolardan nesne tespit edilip tanımlanmasına yönelik uygulamaların sayısı gün geçtikçe artmaktadır. Nesne tespit ve tanıma algoritmaları güvenlik, savunma sanayii, savunma teknolojileri, doğal afetler, sağlık, tarım, ormancılık gibi birçok alanda kullanılarak bu alanlarda karşılaşılan çeşitli problemlere çözüm üretip ihtiyaçlar karşılanmaktadır (Altaş vd., 2023; Cömert vd., 2019; Özgüven&Adem, 2019; Umut vd., 2019). Teknolojinin gelişmesiyle birlikte ve yapay zekâ alanındaki derin öğrenme algoritmalarının gün geçtikçe daha fazla önem kazanmasından dolayı bilgisayarla görü dünyasında daha verimli tasarımların geliştirilmesi ihtiyacını ortaya çıkarmıştır. Bununla birlikte insansız hava araçlarının (İHA) savunma sistemlerinde kullanımı oldukça yaygın hale gelmiştir. İHA'ların yüzlerce nesneyi algılayıp sınıflandırması veya ayırt etmesi için ESA kullanılmakta ve nesne tespit algoritmaları kullanılmaktadır. Bilgisayarla görü yöntemleri kullanılarak farklı araştırma problemlerini çözmek için makine öğrenme algoritmaları ve özellikle derin öğrenme algoritmalarının kullanılma sayısı gün geçtikçe artmaktadır (Wang vd., 2018). Derin öğrenme yöntemi,

yapay zekanın alt dallarından biri olmakla birlikte makine öğrenmesi sınıfına aittir. Derin öğrenme mimarilerinde özellik çıkarma veya dönüştürme amacıyla çok sayıda yapay sinir ağı katmanları kullanılmaktadır. Derin öğrenme katmanlarında bir önceki katmanda yapılan işlemler bir arada yürütüldükten sonra bir sonraki katman için veriler hazır hale getirilmektedir. İşlemlerdeki katman sayılarının artırılması işlem sayısının artışıyla dolayısıyla bir karmaşaya neden olmaktadır. Bu işlem karmaşasının çözümü için veya daha hızlı bir çözüme ulaşmak amacıyla her geçen gün yeni derin öğrenme mimarileri geliştirilmektedir. Çalışmalarda en çok kullanılan derin öğrenme mimarileri ise ESA, Uzun Kısa Vadeli Hafıza Ağları (LSTM) ve Tekrarlayan Sinir Ağlarıdır (RNN) (Doğan&Türkoğlu, 2019; Özkan&Ülker, 2017; Schütt vd. 2019; Zang vd., 2018).

Bu çalışmada ESA mimarileri ve nesne tespit algoritmalarından YOLOv8 ve YOLO-NAS üzerinde durulacaktır. Derin öğrenmede sınıflandırma ve nesne tespiti gibi birçok uygulama alanı mevcuttur. Son yıllarda İHA'lar ile savunma alanında önemli işler yapılmaktadır. Savunma teknolojilerinde üstün yetenekli İHA'ların kullanılması savunma sistemleri açısından büyük önem taşımaktadır. Son yıllarda gelişmiş ülkeler savunma sistemlerine otonom uçabilen İHA'ları eklemeye başlamıştır (Karaağaç, 2014). Bu çalışmada askeri araç, sivil araç ve silahların İHA ile tespit edilip ve sınıflandırılması için derin öğrenme mimarileri ve nesne tespiti algoritmaları karşılaştırılarak en başarılı mimari belirlenerek ve otonom tespit karar sistemi olarak önerilmiştir. Yapılan çalışma ile sınır güvenliğinin sağlanması ve düşman birliklerinden istihbarat alma konuları üzerine faydalar sağlanacaktır. Çalışmada Kaggle platformundan alınan "Ukrainian War Images", "detection of military equipment", "Military vehicles", "Military/Civilian Vehicles Image Classification", "Aiming soldiers image dataset", "Guns Object Detection", "Normal vs Military Vehicles" adlı 7 farklı veri setinden uygun görüntüler seçilerek veri seti hazırlanmıştır. Bu veri setine telif hakkı içermeyen askeri ve sivil araçların görüntüleri de eklenerek 6 sınıflı yeni bir hibrit veri seti hazırlanmıştır. 7 ayrı ESA modeli askeri ve sivil araçları, sivil ya da askeri personeli sınıflandırmak için kullanılacak ve modellerden elde edilen sonuçlar "Kesinlik", "Kayıp değeri" ve "F1-Skoru" parametrelerine göre değerlendirilip en başarılı model olarak önerilmiştir. Ayrıca nesne tespit algoritmalarından You Only Lock Once (YOLOv8) ve You Only Look Once Neural Architecture Searc (YOLO-NAS) algoritmaları kullanılarak elde edilen sonuçlar mean average precision (mAP) parametresi ile değerlendirilerek en başarılı model belirlenmiştir.

Literatürde savunma sistemleri teknolojilerine yönelik, farklı yöntem ve algoritmalarla eğitilmiş modellerin gerçek zamanlı kullanıldığı çalışma sayısı oldukça azdır. Buradan yola çıkılarak otonom uçuş sayesinde gerçek zamanlı nesne tespiti ve sınıflandırma işlemlerini yapabilen insansız hava aracı yapılmıştır. Çalışmanın alana katkıları ve özellikleri şu şekilde sıralanabilir:

- Farklı veri setlerindeki görüntüler birleştirildikten sonra bu görüntülere ek olarak telif hakkı olmayan görüntüler eklenmiştir. ESA mimarileri ve nesne tespit algoritmalarının eğitilmesi için yeni, özgün ve hibrit veri seti hazırlanmıştır. Hibrit veri setine ayrıca veri çoğaltma yöntemleri de uygulanmıştır.
- Çalışmada yapılan İHA'nın gövdesinin çizimi ve tasarımı özgün bir yapıya sahiptir.
- Çalışmada ESA mimarileri ve nesne tespit algoritmaları ile eğitilen modellerin başarı oranları karşılaştırılarak en iyi model belirlenmiş ve belirlenen modelin gerçek zamanlı (real time) çalışması sağlanmıştır. Çalışmada eğitilen modellerden elde edilen başarı oranları literatürdeki benzer çalışmalardan daha yüksektir.
- Literatürde hem askeri, sivil araçları hem de silahlı, silahsız kişileri sınıflandırabilen bir çalışmaya rastlanmadığından bu tarz veri seti ilk defa bu çalışmada ele alınmıştır.
- Bu çalışma ile askeri araçları, silahlı, silahsız kişileri ve sivil araçları kamera ile gerçek zamanlı sınıflandırabilen ve otonom uçuşlar yaparak istihbarat toplayabilen özgün tasarıma sahip insansız hava aracı (İHA) yapılmıştır. Yapılan İHA savunma sanayiinde rahatlıkla kullanılabilir hale getirilmiştir.

## 2. KAYNAK ARAŞTIRMASI

Literatürde yapılan çalışmalar iki paragrafa ayrılarak verilmiştir. İlk paragrafta İHA kullanarak veri seti hazırlayan, derin öğrenme yöntemleri ile nesne tespiti ve sınıflandırma işlemlerini yapan benzer çalışmalara yer verilmiştir. İkinci paragrafta ise bu çalışmada kullanılan yöntemleri kullanan benzer çalışmalara yer verilmiştir.

Lee ve ark. çalışmalarında, orman yangınlarını tespit etmek için ESA ile insansız hava aracını birlikte kullanmışlardır. Çalışmalarında, havadan aldıkları anlık görüntüler ile orman yangınına algılayıp sınıflandıran bir sistem tasarlamışlardır. Çalışmalarında, AlexNet, GoogLeNet, VGG13 algoritmalarını kullanmışlardır. Çalışmaları sonucu en yüksek doğruluk oranını GoogLeNet ile %99 doğruluk oranıyla elde etmişlerdir (Lee vd., 2017). Dikbayır ve Bülbül derin öğrenme yöntemlerinden Faster R-CNN ve YOLO kullanarak sivil araçların tespit edilmesi için çalışma yapmışlardır. Çalışmalarında havadan çekilmiş 500 adet görüntü içeren veri seti kullanmışlardır. Yaptıkları testler sonucunda YOLO algoritmasının başarı oranının %95,61 olduğunu, Faster R-CNN algoritmasının ise %91,76 başarı oranına ulaştığını belirtmişlerdir (Dikbayır&Bülbül, 2020). Arserim ve Ayşan İHA'dan alınan görüntüdeki araçların ve yayaların tespit edilmesi için derin öğrenme algoritmalarından Darknet-53 framework üzerinde YOLOv3 algoritmasını kullanmışlardır. Çalışmalarında 877 adet resim kullanarak etiketleme işlemlerini labeling aracı ile yapmışlardır. Araç ve yayaların tespit edilmesi işleminde %79 başarı oranı elde etmişlerdir (Arserim&Ayşan, 2023). Burgaz çalışmasında insansız hava aracı ile havadan çekilmiş görüntülerden silah tespitinin yapılması için R-CNN algoritması kullanmıştır. Başarı oranını artırmak için R-CNN algoritmalarının yanında ResNet101 algoritmasını da denemiştir. Çalışmasında İHA ile havadan çekilmiş 200 adet görüntü içeren veri seti kullanmıştır. R-CNN ve ResNet101 mimarisi kullanarak veri seti üzerinde yaptığı eğitimle %99 doğruluk oranı elde etmiştir (Burgaz, 2020). Akdağ içerisinde silah nesnesi bulunan resimlerle veri seti oluşturarak çalışmasında kullanmıştır. Oluşturduğu veri setini kullanarak DetectNet ve YOLO derin sinir ağı modellerini eğitmiştir. Test görüntülerinden elde ettiği sonuçlara göre görüntülerdeki silahların tespit edilmesi işleminde DetectNet modelinin YOLO modelinden daha başarılı sonuçlar verdiğini belirtmiştir (Akdağ, 2017). Saygılı ve Özmen ESA tabanlı SSD MobileNet kütüphanesi kullanarak İHA ile gerçek zamanlı havadan nesne tespit çalışması yapmışlardır. Çalışmalarında herhangi bir veri seti

kullanmayarak SSD MobileNet algoritmasındaki daha önceden eğitilen modeli kullanmışlardır. İHA üzerindeki kameradan elde ettikleri görüntüleri NVIDIA Jetson Nano mikroişlemcili bilgisayar ile işleyerek nesnelere sınıflandırmışlardır. Çalışmalarındaki elde edilen çıktılarını mAP ortalama hassasiyet yüzdesi kullanarak değerlendirmişlerdir. İHA kamerasından aldıkları görüntü verilerinden insan tespiti için %69,45 mAP, sivil araba tespiti için ise %83,4 mAP ortalama hassasiyet elde etmişlerdir (Saygılı&Özmen, 2022). Akar drone ile havadan maske tespit edilmesi işlemi için kaggle platformundaki Real-World Masked Face Dataset ve Mask Wearing Dataset veri setlerinden elde ettiği görüntüleri kullanarak hibrit bir veri seti hazırlamıştır. Hazırladığı veri setini YOLOv4 ve ESA modelleri kullanarak eğitmiştir. Eğittiği modellerin gerçek zamanlı performanslarını karşılaştırdıktan sonra en başarılı yöntemin YOLOv4 ile elde edildiğini belirtmiştir (Akar, 2022). Bayram ve Nabiyeve ormanlık alanda kamufle edilmiş 2234 tank ve 774 adet otomobilden oluşan veri setindeki nesnelere tespit etmek için YOLO 5, 6, 7 ve 8 versiyonlarının performansını karşılaştırmıştır. Çalışmalarında en başarılı sonucu YOLOv6 ile elde etmişlerdir. F1-skoru olarak 0,95 ve map50 performansı olarak 0,983 sonuçlarını elde etmişlerdir (Bayram&Nabiyeve, 2023). Liu ve Liu çalışmalarında 4573 askeri araç görüntülerini içeren Sensing Information Analysis Center (SENSIAC) veri setini kullanmışlardır. Veri setindeki görüntülerin eğitilmesi için Faster R-CNN kullanarak yeni bir tespit algoritması önermişlerdir. Askeri araçların tespit edilmesi işleminde %98,6 doğruluk oranı elde etmişlerdir (Liu&Liu, 2017). Nandimandalam içerisinde 1400 adet çeşitli askeri araç ve sivil araç görüntüleri içeren veri setini Faster R-CNN ve SSD nesne tespit algoritmalarını kullanarak eğitmiştir. Çalışmasında veri setindeki görüntülerini analiz ederek askeri araç ve sivil araçları tespit ederek ikili sınıflandırma yapmıştır. Çalışmasında Faster R-CNN ve SSD algoritmalarından elde ettikleri sonuçları karşılaştırmış en başarılı ve en hızlı sonucu Faster R-CNN algoritması ile elde ettiğini belirtmiştir. Faster R-CNN ile %82 doğruluk oranı, SSD300 ile %63 doğruluk oranı elde etmiştir (Nandimandalam, 2020). Türkarşlan ve Hardalaç çalışmalarında Google Earth, GF-2 ve JL-1 uyduları ile topladıkları görüntü verilerinden uçak, gemi, liman ve küçük araç gibi 15 farklı sınıflar oluşturmuşlardır. Oluşturdukları sınıfları YOLOv5 ve SSD algoritmalarını kullanarak eğittikleri model ile tespit etmişlerdir. Çalışmalarındaki en yüksek mAP skorunu %86,7 ile YOLOv5 ile elde etmişlerdir (Türkarşlan&Hardalaç, 2022). Albayrak çalışmasında drone ile sivil araçların görüntülerini havadan çekerek 282 adet görüntü içeren tek sınıflı bir veri seti hazırlamıştır. Veri setindeki görüntüleri kullanarak YOLOv3 ve Mask R-CNN mimarileri ile eğitmiştir. Çalışmasında en yüksek F1 skorunu 76,41 ile YOLOv3

mimarisinden elde etmiştir (Albayrak, 2021). Liu ve arkadaşları çalışmalarında havadan çekilmiş 6471 adet görüntü içeren VisDrone2019 veri setini kullanmışlardır. Veri setinde bulunan 9 farklı sivil araç ve insan görüntülerini YOLOv8 mimarisi ile eğitmişlerdir. YOLOv8 mimarisindeki hata fonksiyonlarının mAP50 skorlarına göre performanslarını karşılaştırmıştır. Çalışmalarında en yüksek mAP50 skorunu %40,7 ile WIOUv3 kayıp fonksiyonundan elde etmişlerdir (Liu vd., 2023).

Aksoy ve Adem kaggle platformundaki “Dogs & Cats Images” ve “pizza classification data” veri setlerindeki verileri kullanarak ESA modellerinde bulunan çeşitli hiperparametreleri değiştirerek sınıflandırma başarısındaki etkilerini incelemişlerdir. Hiperparametrelerdeki eğitim tur sayısı, paket boyutu, nöron sayısı, optimizasyon algoritmaları, aktivasyon fonksiyonları ve öğrenme oranlarını değiştirerek etkilerini incelemişlerdir. Çalışma sonucunda en iyi performansı gösteren NASNetMobile ve DenseNet201 modelleri olmuştur. Çalışmalarında belirledikleri en iyi parametrelere göre NASNetMobile modelinin başarı oranını başlangıç değerine göre %6,5, DenseNet201 modelinin başarı oranını ise %11,55 oranında artırmışlardır (Aksoy&Adem, 2023). Engin ve Adem derin öğrenme yöntemlerini kullanarak maymun çiçeği hastalığının tespit edilmesi için 2 ayrı veri seti ile model oluşturmuşlardır. Çalışmalarında birinci veri seti olarak “Monkeypox Skin Image Dataset” ‘ini ikinci veri seti olarak ise “Monkeypox Orijinal Image Dataset” ve “Monkeypox Master” veri setlerini birleştirerek hibrit bir veri seti hazırlamışlardır. Xception, Inception, ResNetV2, NASNetLarge, InceptionV3 ve DenseNet201 derin öğrenme modellerini kullanarak eğittikleri modellerden birinci veri setinde %99,33 ikinci veri setinde ise %98,52 doğruluk oranı elde etmişlerdir (Engin&Adem, 2023). Tamang ve arkadaşları 2079 adet görüntü içeren Face mask detector veri setini kullanarak gerçek zamanlı maske tespit sistemi oluşturmuştur. Veri setindeki görüntüleri YOLOv5 ve YOLOv8 mimarileri ile eğiterek performanslarını karşılaştırmıştır. Çalışmalarında %96 mAP oranını YOLOv8 ile elde etmişlerdir (Tamang vd., 2023).

Literatür taraması yapıldıktan sonra belirlenen çalışmalar kullanım amacına göre askeri, sivil ve sağlık alanı şeklinde üç kısma ayrılarak Tablo 2.1, Tablo 2.2 ve Tablo 2.3’de sunulmuştur.

Tablo 2.1 Askeri alana yönelik çalışmalar

Kullanıldığı Çalışma	Kullanılan Veri Seti	Amaç	Çalışılan Mimariler	En Başarılı Mimari	Doğruluk oranı %
Burgaz, 2020	Kendi veri seti 200 adet görüntü	Silah tespiti	R-CNN, ResNet101	ResNet101	%99
Akdağ, 2017	Kaggle platformu hibrit veri seti 2605 adet görüntü	Silah tespiti	DetectNet, YOLO	DetectNet	%77,26
Bayram&Nabiyev, 2023	Kendi veri seti 3008 adet görüntü	Tank ve sivil araç tespiti	YOLOv5, v6,v7 ve v8	YOLOv6	% 98,3
Liu&Liu, 2017	Sensing Information Analysis Center veri seti 4573 adet görüntü	Askeri araç tespiti	Faster R- CNN	Faster R- CNN	%98,6
Nandimandalam, 2020	Kendi veri seti 1400 adet görüntü	Askeri ve sivil araç tespiti	Faster R- CNN, SSD	Faster R- CNN	%82
Türkarlan&Hardalaç, 2022	Dota veri seti, 1865 adet görüntü	Uçak, gemi, liman, sivil araç tespiti	YOLOv5, SSD	YOLOv5	%86,7

Tablo 2.2 Sivil alana yönelik çalışmalar

Kullanıldığı Çalışma	Kullanılan Veri Seti	Amaç	Çalışılan Mimariler	En Başarılı Mimari	Doğruluk oranı %
Lee vd., 2017	Kendi veri seti, 23053 adet görüntü	İHA ile orman yangını tespiti	Alexnet, Googlenet, VGG13	Googlenet	%99
Dikbayır&Bülbül, 2020	Kendi veri seti 500 adet görüntü	Sivil araçların tespiti	YOLO, Faster R-CNN	YOLO	%95,61
Arsenim&Ayşan, 2023	Kendi veri seti 877 adet görüntü	Araç ve yayaların tespiti	Dartnet-53, YOLOv3	YOLOv3	%79
Saygılı&Özmen, 2022	Veri seti yok, önceden eğitilmiş ağırlıklar kullanılmıştır	İnsan ve sivil araç tespiti	SSD Mobilnet	SSD Mobilnet	İnsan: %69,45 Sivil araç: %83,45
Aksoy&Adem, 2023	Dogs&Cats Images, Pizza classification veri seti	Hiperparametre etkilerinin incelenmesi	NasnetMobile, DenseNet201, ResNet101, ResNet50, Efficientnet, InceptionV3, Densenet169, Xception	DenseNet201	%78,42
Albayrak, 2021	Kendi veri seti 282 adet görüntü	Sivil araç tespiti	YOLOv3, Mask R-CNN	YOLOv3	%76,41
Liu vd., 2023	VisDrone2019 veri seti 6471 adet görüntü	İnsan ve 9 sınıflı sivil araç tespiti	YOLOv8 farklı kayıp fonksiyonları	YOLOv8 WIOUv3	%40,7

Tablo 2.3 Sağlık alanına yönelik çalışmalar

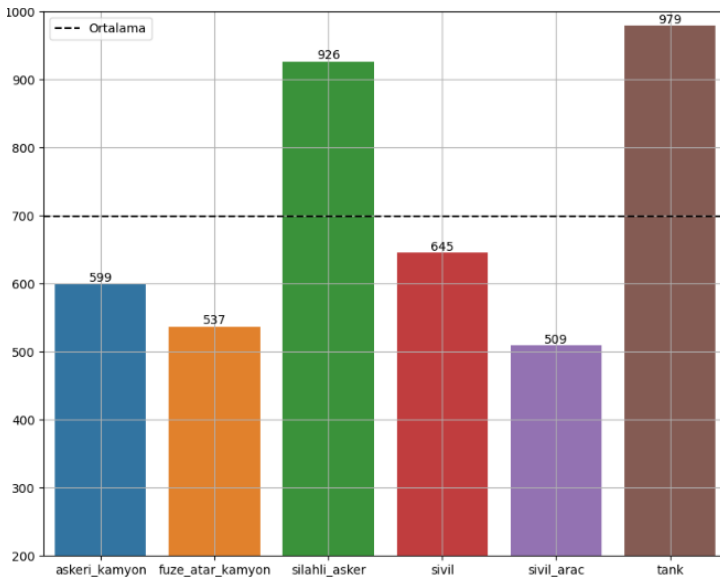
<b>Kullanıldığı Çalışma</b>	<b>Kullanılan Veri Seti</b>	<b>Amaç</b>	<b>Çalışılan Mimariler</b>	<b>En Başarılı Mimari</b>	<b>Doğruluk oranı %</b>
Akar, 2022	Real-World Masked Face ve Mask Wearing veri setleri, 1510 adet görüntü	Maske tespiti	YOLOv4 ve VGG16, VGG19, ResNet101, ResNet152, MobilenetV1	VGG16	%98
Engin&Adem, 2023	Monkeypow Skin Image Dataset, Monkeypox Master veri setleri, 2097 adet görüntü	Maymun çiçeği hastalığı tespiti	Xception, Inception, ResNetV2, Nasnet, InceptionV3, Densenet201	Densenet201	%99,33 ve %98,52
Tamang vd., 2023	Face mask detector, 2079 adet görüntü	Gerçek zamanlı maske tespiti	YOLOv8, YOLOv5	YOLOv8	%96

### 3. MATERYAL ve YÖNTEM

#### 3.1. Kullanılan Veri Setinin Özellikleri

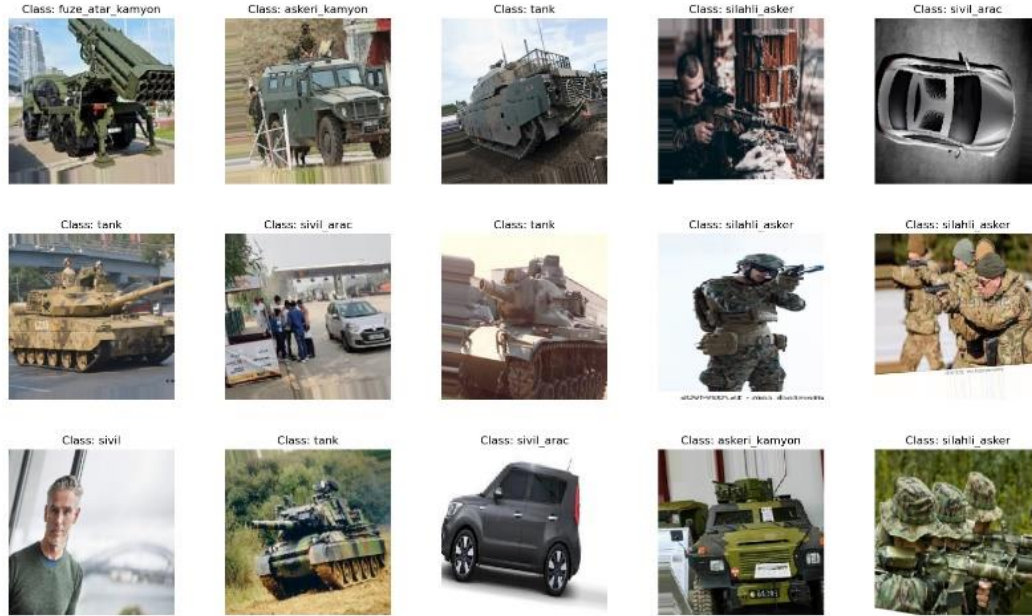
Modellerin eğitimi için gerekli olan verilerin hazırlanma aşaması en önemli aşamalardan biridir. Çözüm bulunacak problem ile ilgili doğru sonuç verebilecek veri setlerinin bulunması veya hazırlanması uzun bir süreç almaktadır. Veri setindeki sınıflara ait veri sayılarının ise homojen veya birbirine yakın olması beklenmektedir (Hu vd., 2020; Byrd & Lipton, 2019). Veri çoğaltma teknikleri derin öğrenme mimarindeki modellerinin eğitilmesi işlemlerinde ezberlemenin engellenmesi (overfitting) için kullanılmaktadır (Shorten ve Khoshgoftaar, 2019).

Çalışmada Kaggle platformundaki “Ukrainian War Images”, “detection of military equipment“, “Military vehicles”, “Military/Civilian Vehicles Image Classification”, “Aiming soldiers image dataset”, “Guns Object Detection”, “Normal vs Military Vehicles” gibi 7 farklı veri setindeki askeri araçlar, sivil insan, silahlı insan ve sivil araç görüntüleri harmanlanarak ayrıştırılmıştır. Bu görüntülere ek olarak telif hakkı olmayan askeri ve sivil araçların görüntüleri ve silahlı ve silahsız insanların görüntüleri de eklenerek 4195 adet görsel içeren 6 sınıflı yeni bir hibrit veri seti hazırlanmıştır. Oluşturulan yeni, özgün ve hibrit veri setindeki sınıflara ait görüntü sayıları Şekil 3.1’de verilmiştir. Nesne tespit mimarileri için veri setindeki görsellerin tamamındaki nesnelere etiketlenerek düzenlenmiştir. Nesne tespit mimarileri, klasik ESA ve transfer öğrenme mimarileri için veri setindeki görüntülere veri çoğaltma işlemleri yapılmıştır. Veri çoğaltma işlemi eğitilecek modellerin ezber yapmasını engellemek için yapılmıştır.



Şekil 3.1. Veri setine ait sınıflardaki görsellerin sayıları

Hazırlanan veri setine ait görsellerin örnekleri Şekil 3.2’de verilmiştir. Veri setindeki görseller rastgele bir biçimde %70’i train, %15’i validation ve %15’i de test verisi olarak ayrıştırılmıştır. Ayrıca modellerin eğitim işleminden önce görüntülere rastgele yaklaştırma (zoom\_range), yatay yönde rastgele kaydırma (width\_shift\_range), dikey yönde rastgele kaydırma (height\_shift\_range), yatay döndürme (horizontal\_flip), belirli açıyla döndürme(rotation\_range) ve yatay kayma (shear\_range) işlemleri uygulanmıştır.

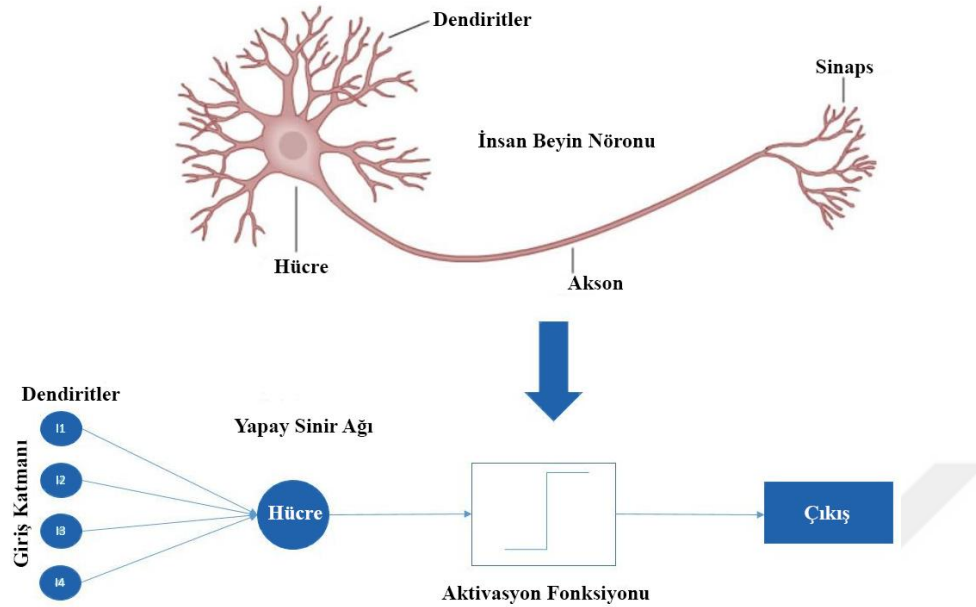


Şekil 3.2. Veri setindeki sınıflara ait görseller

### 3.2. Yapay Sinir Ağı

Yapay sinir ağı (YSA), insan beyninden esinlenerek ses, görüntü ve metin gibi verileri bilgisayarlara öğreten bir yapay zeka yöntemidir. Derin öğrenme ise katmanlı yapılardaki bağlı düğümleri veya nöronları kullanan bir tür makine öğrenimi sürecidir. Derin öğrenme, bilgisayarın yaptığı hatalardan ders çıkararak bunları geliştirmek için kullandığı bir sistemdir. Wechsler derin öğrenmeyi, birbirinin ardından gelen katman dizisinden oluşan ve her katman matematiksel işlemlerden geçirilip işlendiğinde daha verimli bilgiler elde edilebilen ve örneklerden öğrenebilen bir makine öğrenimi tekniği olarak tanımlamıştır (Wechsler, 1992). Şekil 3.3’teki insandaki biyolojik bir sinir sistemine benzer bir şekilde çalışan YSA’lar, modeldeki nöronların çeşitli yollarla birbirine bağlanmasıyla oluşmuş bir ağ sistemidir. Bu ağlar yeni bir bilgiyi öğrenme, kaydetme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarmakla görevlidirler (Öztemel, 2003). Bir nöronun görevi kendine gelen giriş verilerini topladıktan sonra iç eşik değeri aşıldığında bir çıkış verisi üretmektir. YSA’lar sistematik olarak bu şekilde çalışan bir dizi nöronun birbirlerine

bağlanarak karmaşık bir ağ oluşturmasıyla çalışan matematik sistemidir (Ersoy&Karal, 2012).



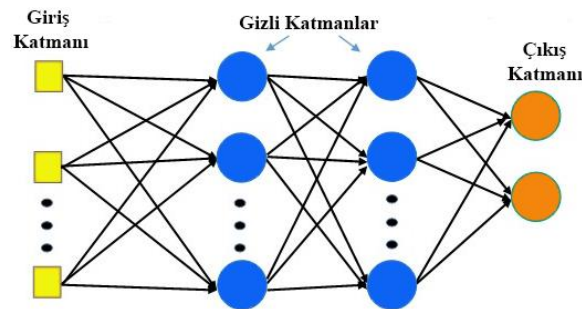
Şekil 3.3 İnsan beyin nöronu ve yapay sinir ağı yapısının karşılaştırılması

Biyolojik sinir sistemindeki elemanlar kısımlara ayrılarak her bir elemanın yapay sinir sistemindeki karşılıkları Tablo 3.1’de verilmiştir.

Tablo 3.1. Biyolojik ve yapay sinir sistemlerinin karşılaştırılması

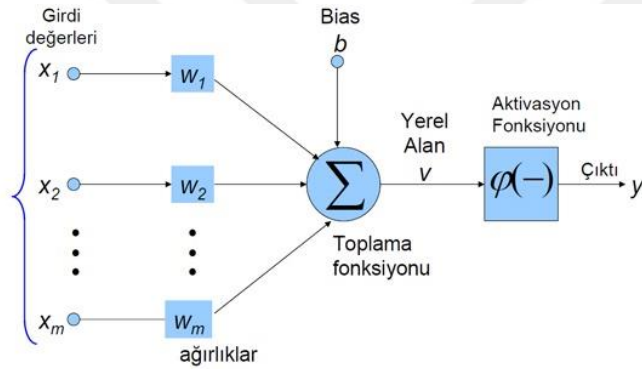
Biyolojik Sinir Sistemi	Yapay Sinir Sistemi
Nöron	İşlemci Elemanı
Dendrit	Toplama Fonksiyonu
Hücre Gövdesi	Transfer Fonksiyonu
Aksonlar	Yapay Nöron Çıkışı
Sinapslar	Ağırlıklar

Pek çok YSA’da aynı özelliklere sahip nöronlar, katmanlar şeklinde yapılandırılarak ve eşik birimlerini eş zamanlı çalıştırmaktadır. Bütün YSA’larda, Şekil 3.4’te gösterildiği gibi aldıkları verilerden çıktı üreten nöronlar vardır. Biyolojik sinir sistemlerinde olduğu gibi, birçok sinir ağındaki bilgi birçok paralel bağlantılar üzerinde dağılmıştır (Cohen, 1994).



Şekil 3.4. Çok katmanlı yapay sinir ağı yapısı

Şekil 3.5’de verilen yapay sinir hücresinin yapısı incelendiğinde giriş katmanındaki hücreye  $m$  tane veri girişinin yapıldığı gözlemlenmektedir ( $X_m$  tane veri girişi). Girilen veriler ilk nörona gelindiğinde rastgele ağırlıklarla çarpılır (rastgele çarpılan ağırlıklar eğitim sürecince güncellenmektedir) sonra veriler toplanır. Sinir ağı eğitim aşamalarında hangi ağırlığın daha önemli olduğunu belirledikten sonra o girdinin ağırlığına daha yüksek değer vererek günceller. Daha sonra tahmin eklenir ardından yerel alanda net yargı elde edilmiş olur. Net girdi belirlenen aktivasyon fonksiyonundan geçirilir ve bir veri çıktısı yani sınıf değeri elde edilmiş olur. YSA’nın giriş verilerinin çıkış verilerine ilişkilendirme başarısı, nöronların transfer fonksiyonlarının türünden, birbirine olan bağlantılarından ve bu bağlantıların ağırlıklarından etkilenmektedir (Keskenler&Keskenler, 2017; Cherkassky vd., 2012).



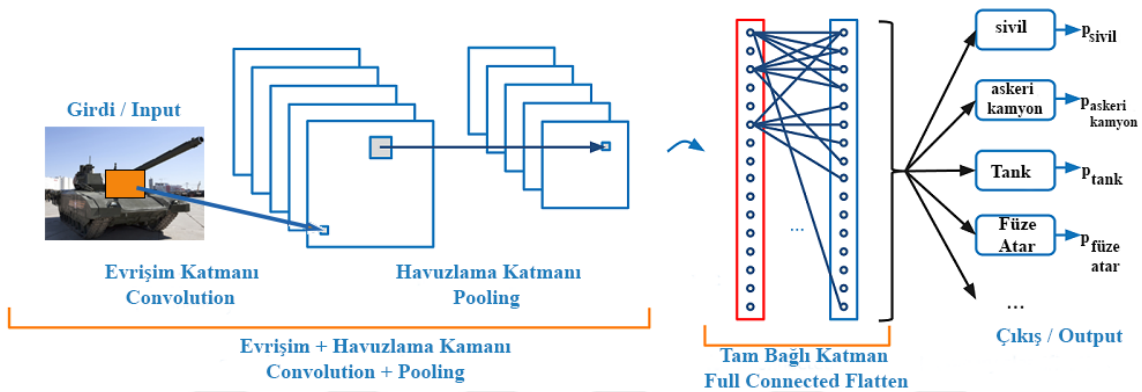
Şekil 3.5. Yapay sinir hücresi yapısı

Bir sinir ağındaki öğrenme Şekil 3.5’de gösterildiği gibi, girdi/çıkış verilerinin işlenip, bağlantı ağırlıklarının yakınsama elde edilene kadar, tekrar tekrar ayarlanması ile gerçekleşmektedir. Evrimsel Sinir Ağları (ESA), ağırlıkların belirlenmesi ile birbirlerine bağlanmış nöronlardan oluşan matematiksel sistemlerin bütünüdür. Bir ESA’daki işlem birimi, transfer fonksiyonu olarak anılan bir denklemi ifade eder. Transfer fonksiyonu, ağdaki her bir nöronun sinyalleri alır, birleştirir, dönüştürür ve en sonunda sayısal bir değer ortaya çıkartır. Genelde, ağdaki işlem birimleri biyolojik bir sistemdeki nöron hücrelerine karşılık gelmektedir ve sistematik bir ağ içinde birbirlerine bağlanarak yapay sinir ağlarını oluşturmaktadır (Cohen, 1994).

### 3.3. Evrimsel Sinir Ağları ve Modelleri

Görüntü sınıflandırma, bilgisayar görüşünde en temel ve yaygın olarak kullanılan alanlardan biridir. Son yıllarda ESA, hızlı ve doğru özellik çıkarma işlevi ve uçtan uca eğitilebilir ağ yapısı nedeniyle görüntü sınıflandırması alanında büyük başarı elde etmiştir (Lei vd., 2019). Bilgisayar görüşünde parametre paylaşımının en popüler ve kapsamlı

kullanımı evrişimli sinir ağlarındadır (Bengio vd., 2017). ESA, derin öğrenmenin alt dallarından biri olup görüntü odaklı çalışan bir yöntemdir. ESA yapısında katmanlar, filtreler ve fonksiyonlar gibi parametreler bulunmaktadır (Camgözlü&Kutlu, 2019). Geleneksel sinir ağlarında olduğu gibi ESA, giriş katmanı, gizli katman ve çıkış katmanından oluşur. ESA'da evrişim, havuzlama ve tam bağlantılı katmanların bulunmasından dolayı geleneksel sinir ağlarından farklılık göstermektedir (Krizhevsky vd., 2012). ESA, görüntü sınıflandırma çalışmalarında sıklıkla kullanılmakla birlikte özellik çıkarma ve öğrenmede yararlılığı kanıtlanmış bir modeldir. ESA'lar öncelikle girdinin görüntülerden oluşacağı temeline odaklanır (O'Shea&Nash, 2015).



Şekil 3.6. ESA yapısı ve katmanları

ESA görüntüleri girdi verisi olarak kullanan derin öğrenme algoritması olduğundan görüntülerdeki özellikleri kullanarak görüntüleri birbirinden ayırarak sınıflandırma işlemlerini yapabilmektedir. ESA'da kullanılan görüntü verilerinin bilgileri için gerekli olan ön işlem seviyesinin klasik sınıflandırma algoritmalarına göre çok daha düşük seviyede olmasının nedeni ESA'nın filtreleri öğrenme becerisine sahip olmasından kaynaklanmaktadır. Şekil 3.6'da verilen ESA'nın yapısı incelendiğinde mimarinin temelde Evrişimli Katman (Convolutional Layer), Havuzlama Katmanı (Pooling Layer) ve Tam Bağlantılı Katman'dan (Fully Connected Layer) oluştuğu gözlemlenmektedir. Evrişim ve havuzlama katmanında görüntülerden öznitelik çıkarma işlemleri, tam bağlı katmanda ise sınıflandırma işlemlerinin yapıldığı söylenebilir (O'Shea&Nash, 2015).

### 3.3.1 Evrişimli sinir ağları katmanları

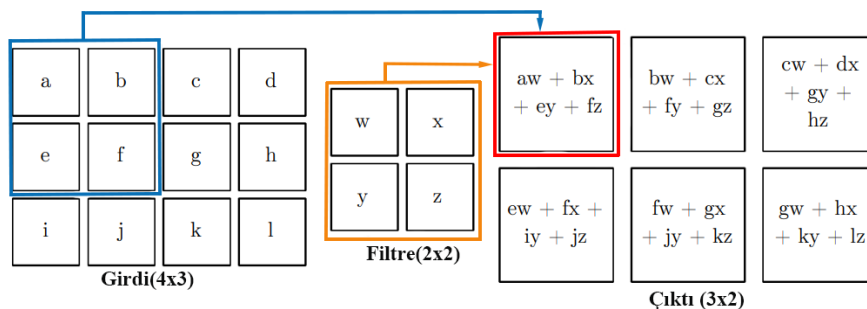
#### Evrişim katmanı (Konvolüsyon katmanı)

Evrişim katmanı, doğrusal veya doğrusal olmayan işlemlerin yani konvolüsyon işlemi ve aktivasyon fonksiyonunun bir kombinasyonundan oluşan özellik çıkarma işleminin gerçekleştirildiği ESA mimarisinin en temel bileşenidir (Yamashita vd., 2018). Öznitelik

çıkarma işlemi girdi olarak verilen bir görüntü üzerinde daha küçük boyuttaki filtre ya da katmanın, sırayla gezdirilmesiyle gerçekleştirilmektedir (Deng & Yu, 2014). YSA'da bulunan her katmandaki hücrelerin birbirleri ile bağlantılı olması ve bağlantı ağırlıklarının güncellenmesi yerine ESA'da resim üzerindeki küçük alanlarda evrişim işlemi kullanılır. ESA'nın en önemli avantajlarından birisi de evrişim katmanlarında hesaplanıp paylaşılan ağırlıkların olmasıdır. Ağırlık paylaşımı, bilgisayar belleğine olan ihtiyacı azaltmaya ve bilgisayarlı görü performansını artırmaya yöneliktir (Wu, 2017). Şekil 3.7'de verilen evrişim işlemindeki çıktıya özellik haritası (Feature Map) denmektedir. Filtre görüntü üzerinde hareket ettirilerek basit matris çarpımı ile özellikler tespit edilmektedir (Yamashita vd., 2018). Burada elde edilen sonuç, ağıdaki son katmana ulaşılmamışsa bir sonraki katmana için girdi olarak verilir. Eğer yapılan işlem son katmanda yapılmış ise elde edilen sonuç çıktı görüntüsünü temsil eder. Evrişim katmanındaki hesaplamalar Eşitlik (1)'e göre yapılır. Eşitlik (1)'deki  $r$ , katman sayısıdır;  $F_n$ , geçerli katmanın  $n$ 'inci özellik haritasına bağlanan, önceki katmanda ayarlanan özellik haritasıdır;  $x_n^r$ ,  $r$ 'inci katmanın çıktısı olan  $n$ 'inci özellik haritasıdır;  $x_m^{r-1}$ ,  $(m - 1)$ 'inci katmanın çıktısı olan  $m$ 'inci özellik haritasıdır;  $k_{mn}^r$   $(r - 1)$ 'inci katmanın  $m$ 'inci özellik haritası ile  $r$ 'inci katmanın  $n$ 'inci özellik haritası arasındaki evrişim çekirdeğidir;  $b_n^r$ ,  $r$ 'inci katmanın çıktısı olan  $n$ 'inci özellik haritasının sapmasıdır(Luo vd., 2018; Prasoon vd., 2013).

$$x_n^r = \sum_{m \in F_n} x_m^{r-1} * k_{mn}^r + b_n^r \quad (1)$$

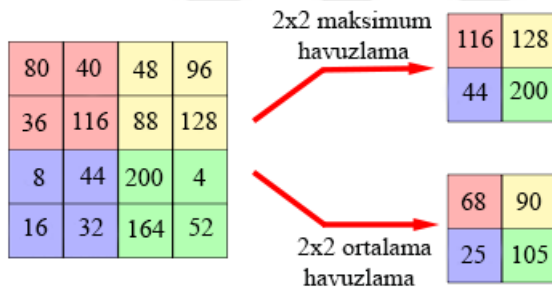
ESA'da birden fazla evrişim katmanı bulunmakla birlikte genellikle birden fazla özellik tespit edilirken birden fazla filtre kullanılmaktadır. Evrişim katmanına ilişkin bir YSA modelini eğitme sürecinde, belirli bir eğitim veri setine yönelik belirli bir görev için en iyi çalışan çekirdeklerin belirlenip tanımlanması gerekmektedir. Evrişim katmanında bulunan çekirdekler, eğitim sürecinde otomatik olarak öğrenilen tek parametrelerdir. Çekirdeklerin boyutu, çekirdek sayısı, dolgu ve adım, eğitim öncesinde ayarlanması gereken hiper parametrelerdir (Yamashita vd., 2018).



Şekil 3.7. Evrişim işlemi

### Havuzlama katmanı

Evrişim katmanının çalıştırılması sonucunda özellik haritalarının sayısında bir artış olmaktadır. Bu da özellik boyutunun genişlemesine neden olmaktadır. Özellik boyutunu ve eğitimin zorluğunu azaltmak için evrişim katmanından sonra havuzlama katmanı eklenir (Luo vd., 2018). Havuzlama katmanında ağ içindeki parametreler ve görüntünün en boy oranı azaltıldığından havuzlama katmanı işlemi sonucu hesaplama yükü hafifletilmektedir. Maksimum havuzlama (maxpooling) ve ortalama havuzlama (average pooling) olmak üzere iki adet havuzlama operatörü yaygın olarak kullanılmaktadır. Maksimum havuzlama ve ortalama havuzlama işlemlerinin nasıl yapıldığı Şekil 3.8’de verilmiştir. Maksimum havuzlamada, havuzlama operatörü bir alt bölgeyi maksimum değerine eşlerken, ortalama havuzlama bir alt bölgeyi ortalama değerine eşler (Wu, 2017). Havuzlama işlemleri zorunlu olmamakla birlikte isteğe bağlıdır. Havuzlama katmanında maksimum havuzlama daha çok tercih edilmektedir. Çünkü maksimum havuzlamada yön ve konum değişmeden yükseklik ve genişlik azaltılır. Bu sayede belirgin özellikler çıkarılarak model etkili bir şekilde eğitim sürecine devam eder. Havuzlama katmanında yapılan işlemler modelin ezberlemesini engellemektedir.

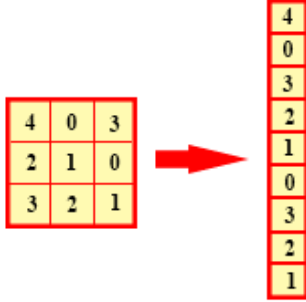


Şekil 3.8. Havuzlama işlemleri

### Tam bağlı katman

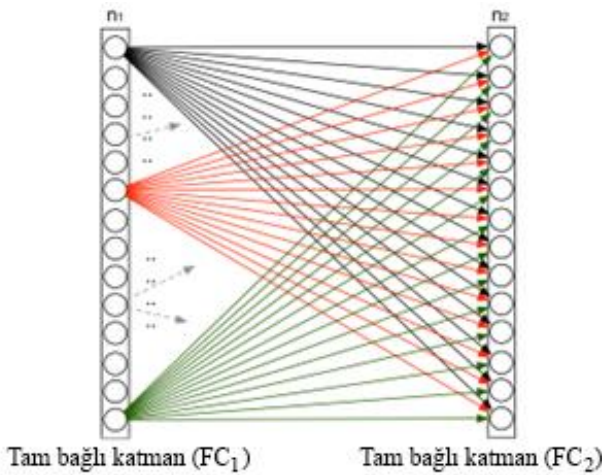
Bu katman sınıflandırmada kullanılan ESA'nın en son ve en önemli katmanıdır. Şekil 3.10'da verildiği gibi tam bağlı katman kendinden önce gelen katmanın bütün nöronlarına bağlıdır. Ağda daha derinlere doğru ilerledikçe ağ boyutu azalır ve bir dizi çıktıyla sonuçlanır. Klasik bir ESA'nın alt katmanlarına öğrenilen özelliklerin doğrusal olmayan kombinasyonlarını öğrenmek için tam bağlı katmanlar gereklidir. Bu katman, verilerini düzleştirme işleminden alır ve sinir ağı yoluyla öğrenme işlemini gerçekleştirir. ESA, giriş verilerini tek boyutlu bir diziden almaktadır. Düzleştirme (flattening) işlemi (Şekil 3.9), ağdaki evrişim ve havuzlama katmanından gelen çok boyutlu matris verilerinin tek boyutlu diziye çevrilme işlemidir. Düzleştirme işleminin amacı tam bağlı katmanın girişine veri

hazırlamaktır. Her ESA mimarisinde ağın derinliğine bağlı olmak üzere bir veya birden fazla tam bağlı katman kullanılabilir (Lin vd., 2014; Basha vd., 2020). ESA’da büyük verilerle model eğitilmek istenildiğinde aşırı öğrenme (overfitting) gibi olumsuz durumlar oluşabilir. Aşırı öğrenmeyi engellemek için Dropout katmanı kullanılarak ağdaki bazı düğümler kaldırılır.



Şekil 3.9. Düzleştirme işlemi

Klasik bir sinir ağında tam bağlı katman ağın parametrelerinin çoğunu oluşturmaktadır. Genişlik (tam bağlı katmanlardaki nöron sayısı, evrişim katmanlarındaki filtre sayısı) ve derinlik (eğitilebilir katman sayısı) sinir ağı mimarisi tasarımında iki temel unsurdur. Mimarilerden en iyi performansı elde etmek için bu iki parametre dikkatlice ayarlanmalıdır. Daha sığ olan ESA’lar tam bağlı katmanlarda daha fazla düğüme ihtiyaç duyarlar. Daha derin ESA’lar ise veri setinin türüne bağlı olmamakla birlikte tam bağlı katmanlarda daha az sayıda nörona ihtiyaç duymaktadırlar (Basha vd., 2020).

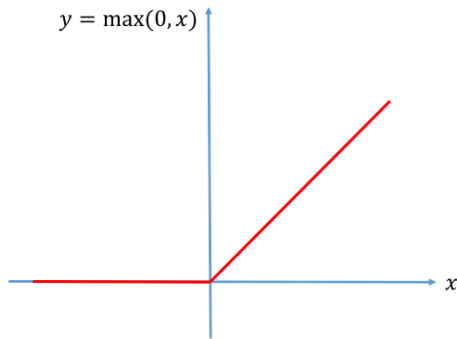


Şekil 3.10. İki katmanda tam bağlı katman örneği

### Aktivasyon fonksiyonları

Ağdaki değişkenler arasında olan sürekli ve karmaşık ilişkileri öğrenme ve tahmin etme işlemleri için aktivasyon fonksiyonları kullanılmaktadır. Her derin ağın temelinde doğrusal bir dönüşüm ve ardından  $f(\cdot)$  aktivasyon fonksiyon bulunur. Aktivasyon fonksiyonu derin

sinir ağlarının eğitiminin başarısında önemli bir role sahiptir. Aktivasyon fonksiyonu, sinir ağlarındaki katmanların en sonuna veya katmanların arasına yerleştirilen bir düğümdür. Literatürde tanh, Sigmoid, ReLU ve Softmax aktivasyon fonksiyonları sıklıkla kullanılmaktadır. Şu anda en başarılı ve yaygın olarak kullanılan aktivasyon fonksiyonu  $f(x) = \max(x, 0)$  olarak tanımlanan doğrultulmuş doğrusal birimdir (Rectified Linear Unit, RELU). RELU aktivasyon fonksiyonunun olduğu derin ağlar sigmoid ve tanh aktivasyon fonksiyonlarına göre daha kolay optimize edilir. Sadeliği ve etkinliği sayesinde RELU derin öğrenme uygulamalarında ve çalışmalarında sıklıkla varsayılan aktivasyon fonksiyonu olarak tercih edilmektedir (Ramachandran vd., 2017; Wu, 2017). RELU'nun diğer aktivasyon fonksiyonlarına göre diğer avantajı ise ağdaki tüm nöronlar aynı anda aktifleşir. RELU aktivasyon fonksiyonunun daha hızlı olmasının nedeni ağdaki tüm negatif girdiler sifira çevrilirken nöronların aktifleşmemesinden kaynaklanmaktadır. RELU aktivasyon fonksiyonu, tanh ve sigmoid aktivasyon fonksiyonlarına 6 kat oranla daha fazla yakınsamaktadır (Hanin, 2019). Şekil 3.11'de verilen grafik incelendiğinde RELU fonksiyonu tüm girdileri  $[0, +\infty]$  arasındaki çıktıya çevirmektedir (sadece pozitif değerlerde aktif olmaktadır) bu sebeple doyuma asla ulaşmamaktadır.

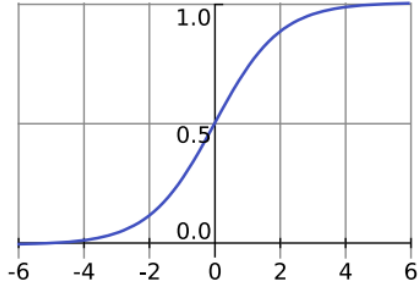


Şekil 3.11. Relu aktivasyon fonksiyonu grafiği

Sigmoid fonksiyonu tüm giriş değerleri için tanımlanabilen ve her yerde pozitif türevi olan, sınırlı, sürekli türevlenebilen bir doğrusal fonksiyondur. Ayrıca 0 ve 1 aralığında çıkış değeri üretmekte olup türevlenebilmektedir (Han&Moraga, 1995). Sigmoid fonksiyonunun matematiksel formülü Eşitlik (2)'de, grafiği ise Şekil 3.12'de verilmiştir. Grafik incelendiğinde büyük negatif piksel değerlerini 0, büyük pozitif piksel değerlerini ise 1 olarak baskıladığı gözükmemektedir. Grafikteki  $x$  değerleri ilerlediğinde uç noktalar 0'a yakınsamakta ve  $y$  ekseninde değerler gözükmemektedir bu da kaybolan gradyan problemini ortaya çıkarmaktadır. Kaybolan gradyan problemi sigmoid fonksiyonunun bir dezavantajıdır. Ayrıca sigmoid fonksiyonu sifira göre simetrik olmadığından nöronların

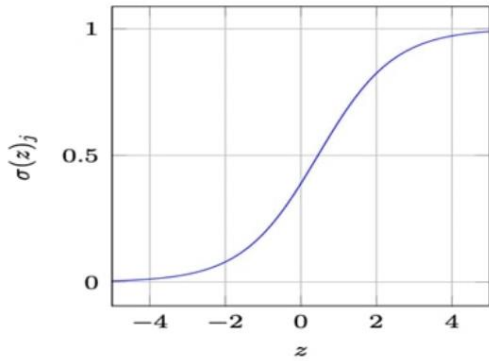
tüm çıkış değerleri aynıdır, bu sorun sigmoid fonksiyonunun ölçeklendirilmesiyle iyileştirilebilir (Sharma vd, 2017).

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (2)$$



Şekil 3.12. Sigmoid aktivasyon fonksiyonu grafiği

Softmax aktivasyon fonksiyonu birden fazla sigmoid fonksiyonunun birleşiminden oluşmaktadır. Sigmoid fonksiyonu 0 ile 1 aralığında değerler döndürdüğünden bunlar belirli bir sınıfın veri noktalarının olasılıkları olarak ele alınabilir. Sigmoid fonksiyonundan farkı ise elde edilen çıktıların 1'e kadar olan toplamlarının normalize edilmesinden kaynaklanmaktadır. Yani girdi pozitif, negatif veya 0 olduğunda softmax fonksiyonu bu girdi değerini 0 ile 1 arasına dönüştürerek olasılık ortaya çıkartır. Eğer bir girdinin değeri küçük veya negatif olduğunda küçük olasılık, büyük olduğunda ise büyük olasılık olarak atar.



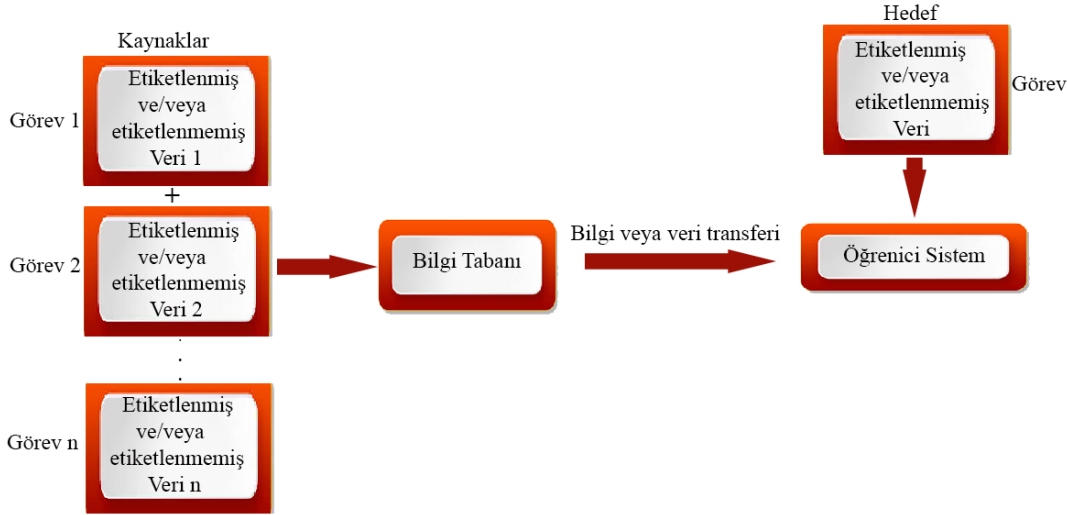
Şekil 3.13. Softmax aktivasyon fonksiyonu grafiği

Softmax aktivasyon fonksiyonu genellikle sinir ağlarının çıkış katmanında çoklu sınıflandırma işlemleri için kullanılır, kategorik bir olasılık dağılımını sağlar. Çoklu sınıflandırma problemleri için bir ağ veya model oluşturulmak istendiğinde ağdaki çıkış katmanındaki nöronla aynı sayıda sınıf sayısı olacaktır (Sharma vd, 2017). Softmax Eşitlik (3)'deki matematiksel formülle ifade edilir.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (3)$$

### 3.3.2 Transfer öğrenme (Transfer Learning)

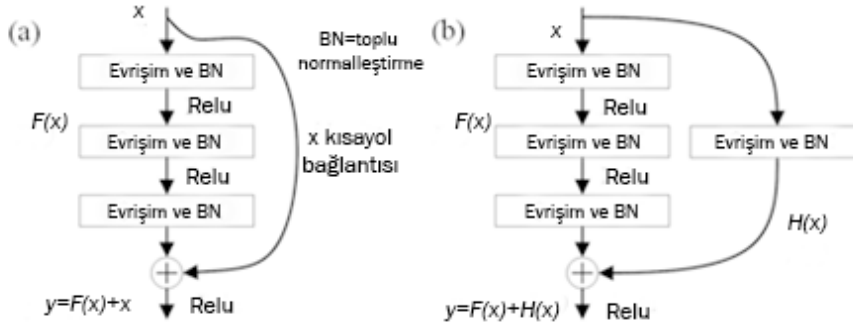
Klasik makine öğrenmesi yöntemleri, yapılacak yeni eğitim modelinde her özneliği, ağırlıkları, problemleri analiz edip her seferinde karşılaşılan problemleri sıfırdan başlayarak çözmektedir. Geçmiş deneyimlerden çıkarım yapmazlar, eğitimlerine yardımcı olacak veri kaynakları yoktur. Makine öğrenimi test verileriyle aynı dağılıma sahip bol miktarda etiketli eğitim örneğine ihtiyaç duymaktadır. Fakat günlük hayattaki çeşitli senaryolarda, yeterli eğitim verisi toplamak genellikle pahalı, zaman alıcı olmakta ve hatta gerçekçi olmamaktadır. Yarı denetimli öğrenme modelleri, toplu etiketli veri ihtiyacını azaltarak bu sorunu kısmen çözebilir. Tipik olarak, yarı denetimli bir yaklaşım yalnızca sınırlı sayıda etiketli veri gerektirir ve öğrenme doğruluğunu artırmak için büyük miktarda etiketsiz veri kullanır. Ancak çoğu durumda, etiketsiz örneklerin toplanması da zordur ve bu da genellikle ortaya çıkan geleneksel modelleri yetersiz kılmaktadır. Bilginin etki alanları arasında aktarılmasına odaklanan transfer öğrenimi, belirtilen sorunların çözümü için umut verici bir makine öğrenimi metodolojisidir. İnsan beyninin öğrenme işlemleri düşünüldüğünde insan beyninde gerçekleşen öğrenimler geçmiş deneyimlerden faydalanılarak, daha önce karşılaşılan bir nesneden yola çıkarak yeni bir nesneyi öğrenme şeklindedir. Transfer öğrenmesi, insan beynindeki geçmiş deneyimlerden yola çıkarak yeni nesnelere öğrenme yeteneğinden ilham alınmıştır. Bu teoriye göre, transferin ön koşulu, iki öğrenme faaliyeti arasında bir bağlantı olması gerektiğidir (Şekil 3.14). Pratikte, keman öğrenmiş bir kişi piyanoyu diğer nesnelere daha hızlı öğrenebilir, çünkü hem keman hem de piyano müzik aletleridir ve bazı ortak bilgileri vardır ve bu bilgiler paylaşılabilir. İnsanların bilgiyi etki alanları arasında aktarabilme yeteneklerinden esinlenen transfer öğrenme yaklaşımı, öğrenme performansını artırmak veya hedef etki alanında gereken etiketli örnek sayısını en aza indirmek için ilgili bir etki alanındaki bilgiden yararlanmayı amaçlamaktadır (Zhuang vd., 2020). Benzer şekilde transfer öğrenmede hedeflenen öğrenme transfer edilen bilgiden olumsuz etkileniyorsa buna negatif transfer denilmektedir (Jialin & Yang, 2010).



Şekil 3.14. Aktarmalı öğrenme

### 3.3.3 ResNet50 (Residual network) mimarisi

Tipik olarak daha derin ESA'ların eğitilmesi daha zordur. Modelin doğruluğu genellikle katmanın derinliği arttıkça azalmaktadır. Daha derin sinir ağlarının eğitimi yok olan gradyan ve bozunma problemlerinden dolayı zor olmuştur. Artık öğrenme (residual network) mimarisi bu iki sorunu çözmeye ve özellik çıkartma katmanlarının arızalarını tespit etmeye dayanmaktadır. Artık yapı bloğu (Residual building block, RBB) ResNet50'deki en önemli unsur olmakla birlikte kısayol bağlantılarını kullanarak evrişim katman bloklarını atlayabilmektedir. Bu kısayollar kaybolan gradyan problemini önlemek için hatanın geri yayılımında eğitilebilir parametreleri ayarlamak için hata tespitinde kullanışlı olmaktadır. Artık yapı bloklarının gösterimi Şekil 3.15'de verilmiştir. ResNet50 mimarisi 2 adet (3x3) evrişim katmanı yerine (1x1), (3x3), (1x1) şeklinde 3 adet evrişim katmanlarını kullanır.



Şekil 3.15. Artık yapı blokları (a)RBB-1 bloğu (b)RBB-2 bloğu

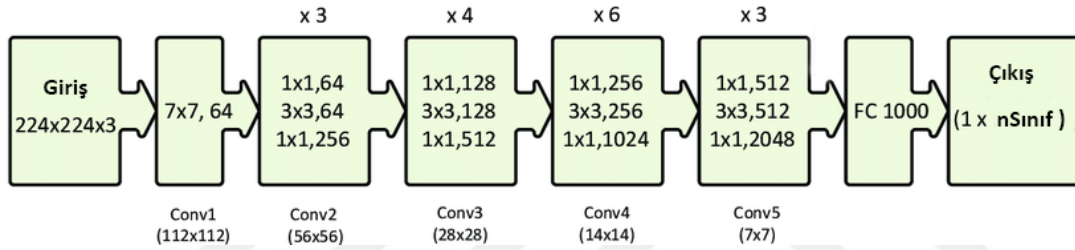
Kısayol bağlantısı kullanılmadan önce 'x' girdisi sıradaki katmanın ağırlıklarıyla çarpılır ve bu çarpıma ön yargı terimi eklenir. Ardından  $F(x)$  fonksiyonu, aktivasyon

fonksiyonundan geçerek çıktığı  $y$  olarak verir.  $H(x)$  fonksiyonu öğrenilirken, ağıdaki her katmandan geçen bilgi direkt olarak sonraki katmana aktarılır. RBB1 Eşitlik (4)'e göre RBB-2 Eşitlik (5)'e göre hesaplanır.

$$y = F(x) + x \quad (4)$$

$$y = F(x) + H(x) \quad (5)$$

ResNet50'nin optimize edilmesi daha kolay bir mimari olduğundan ResNet50 ile derinlik arttıkça yüksek doğruluk elde edilebilir. Bu model, ImageNet veri kümesi üzerinde eğitilmiş 50 katmana sahip bir ağıdır. ResNet50 modeli 2 adet  $3 \times 3$  evrişim katmanı kullanmak yerine  $1 \times 1$ ,  $3 \times 3$ ,  $1 \times 1$  şeklinde 3 adet evrişim katmanı kullanmaktadır. (Zahisham vd., 2020; Wen vd., 2020; He vd., 2016; Keskar&Socher, 2017). ResNet50'nin mimarisi Şekil 3.16'da verilmiştir.



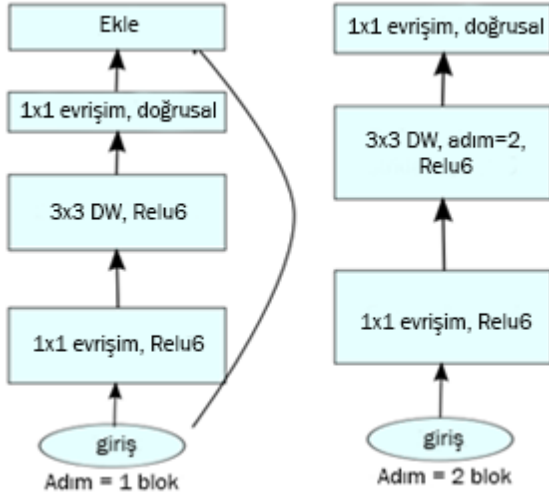
Şekil 3.16. ResNet50 mimarisi

### 3.3.4 MobileNet mimarisi

Google tarafından önerilen MobileNet V2 mimarisi derinlemesine ayrılabilir evrişim (DWC) yapısını kullanılmaktadır. DWC yönteminde derinlemesine evrişim (DC) yapısı haricinde  $1 \times 1$  boyutlarında bir filtreleme işlemi uygulanmaktadır. MobileNet V2 mimarisinde, bir önceki versiyonu olan MobileNet V1 versiyonuna göre ağırlık sayıları azaltılmıştır. MobileNetV2 ağ mimarisi 16 blok katmanına sahip, RELU6 aktivasyon fonksiyonunu kullanan bir ağıdır. Ayrıca katmanlarda ResNet mimarisinde olduğu gibi residual connection bulunmaktadır, bu bloklar Bottleneck Residual Block olarak adlandırılmaktadır. Ayrıca ResNet'de kullanılan bu bloklarda önemli iki iyileştirmeye sahiptir. Bunlardan biri doğrusal bir darboğaz yapısı önermek, diğeri ise ters bir artık yapı önermektir. MobileNetV2, özellikle boyutu geliştirmek için kullanılan "genişletme" işlemi için derinlemesine evrişim katmanından (DW) önce  $1 \times 1$  boyutundaki filtrelemeyle noktasal evrişim ekler. İkinci noktasal evrişimin (PW) ana işlevi boyutluluğu azaltmaktır. Derinlemesine ayrılabilir evrişimler birçok yapay sinir ağı mimarisinin önemli bir temel yapı taşıdır. MobileNetV2  $k=3$  ( $3 \times 3$  DW) kullanmasından dolayı işlemleri hesaplama maliyeti standart evrişimlerden yaklaşık 8 ile 9 kat arası ( $k^2$  kadar)) daha azdır Eşitlik (6).

Eşitlik (6)'daki  $h$  derinlik,  $w$  genişlik,  $d$  derinlik,  $k$  ise katmanı ifade etmektedir. Derinlemesine ayrılabilir evrişimler standart evrişim katmanların yerine geçebilecek alternatif katmandır. Yapılan deneysel çalışmalarda normal evrişimler kadar iyi çalıştığı gözlemlenmiştir.

$$h_i * w_i * d_i(k^2 + d_j) \quad (6)$$



6

Şekil 3.17. MobileNetV2 mimarisi

Şekil 3.17 incelendiğinde MobileNetV2 mimarisinin ikinci noktasal evrişimden (PW) önce gelen doğrusal olmayan aktivasyon katmanını kaldırdığı gözlenmektedir. Darboğaz (bottleneck) katmanının çıktısını doğrusal bir aktivasyon katmanı takip ettiğinden bu bir doğrusal darboğazdır. Bu işlem Relu aktivasyon fonksiyonundan kaynaklı bilgi kaybını azaltarak modelin doğruluk başarısını artıracaktır (Sandler vd., 2018; Dong vd., 2020; Zhou vd., 2020).

### 3.3.5 EfficientNet mimarisi

ESA'lar genellikle sabit bir kaynak bütçesiyle geliştirilir ve eğer daha fazla kaynak varsa daha iyi doğruluk için ölçeklendirilmektedir. EfficientNet mimarisi model ölçeklendirmeyi sistematik olarak inceleyip ve ağ derinliğini, genişliğini ve çözünürlüğünü etkili bileşik katsayısı ile dengeleyip daha iyi performans verebilmektedir. Örneğin,  $2^N$  kat daha fazla hesaplama kaynağı kullanmak istenirse, ağ derinliğini  $\alpha N$ , genişliğini  $\beta N$  ve görüntü boyutunu  $\gamma N$  kadar artırmak gerekir. Bileşik ölçeklendirmedeki bileşik katsayısı olan  $\varphi$  Eşitlik (7)'ye göre hesaplanmaktadır. Eşitlik (7)'deki  $\alpha$ ,  $\beta$ ,  $\gamma$  gridsearch ile belirlenen katsayılar,  $\varphi$  ise model ölçeklendirmesi için ne kadar kaynağın mevcut olduğunu kontrol edebilen kullanıcı tarafından girilen katsayıdır. Denklemden ifade edilen  $d$  ağ derinliğini

ölçeklendirmek için birçok ESA'da kullanılan yoldur. Ağ genişliğinin ( $w$ ) ölçeklendirilmesi genellikle küçük boyutlu modeller için kullanılır. Daha yüksek çözünürlüklü ( $r$ ) giriş görüntüleri ile ESA potansiyel olarak daha ince küçük desenleri tespit edebilmektedir.

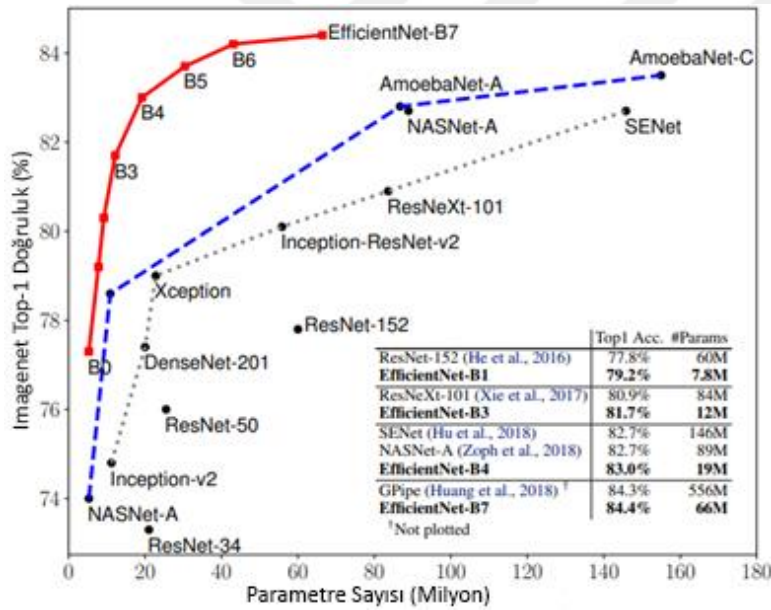
$$\text{derinlik: } d = \alpha^\varphi$$

$$\text{genişlik: } w = \beta^\varphi$$

$$\text{çözünürlük: } r = \gamma^\varphi$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \quad (7)$$

EfficientNet mimarisi [B0, B7] aralığında 8 modelden oluşmaktadır (Şekil 3.18). Genelde model sayısı arttıkça parametre sayısının artması beklenirken bu modelde artmamakta ve doğruluk artmaktadır. Klasik ESA modellerinde genelde RELU aktivasyon fonksiyonu kullanılırken bu mimaride Swish adı verilen yeni bir aktivasyon fonksiyonu kullanılmıştır.



Şekil 3.18. EfficientNet mimarisi

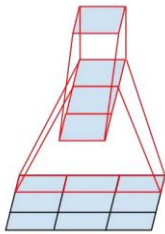
EfficientNet için ana yapı taşı, ilk olarak MobileNetV2 mimarisinde ortaya çıkarılmış olan ters çevrilmiş darboğazdır (MBConv). Genişleme katmanlarına oranla daha az sayıda kanalı birbirine bağlayan darboğazlar arasındaki kısayolları kullanmaktadır (Tan&Le, 2019; Atilla vd., 2021).

Tablo 3.2. EfficientNet temel ağ yapısı

Adım $i$	Operatör $\hat{F}_i$	Çözünürlük $\hat{H}_i \times \hat{W}_i$	Kanal $\hat{C}_i$	Katman $\hat{L}_i$
1	Conv3x3	224 x 224	32	1
2	MBCConv1, k3x3	112 x 112	16	1
3	MBCConv6, k3x3	112 x 112	24	2
4	MBCConv6, k5x5	56 x 56	40	2
5	MBCConv6, k3x3	28 x 28	80	3
6	MBCConv6, k5x5	14 x 14	112	3
7	MBCConv6, k5x5	14 x 14	192	4
8	MBCConv6, k3x3	7 x 7	320	1
9	Conv1x1 & Pooling & FC	7 x 7	1280	1

### 3.3.6 InceptionV3 mimarisi

Inception ağ modeli 42 katmana sahip bir mimaridir ve modüllerden (inception) oluşmaktadır. Ağdaki her bir modül, farklı boyuttaki evrişim ve maksimum havuzlama (max-pooling) işlemlerinden oluşmaktadır. Szegedy ve arkadaşları ağdaki her evrişim katmanından önce 1x1 evrişim katmanı kullanarak zorlu hesaplamaların olduğu karmaşık bir ağ modelinde daha az işlem yapıp daha hızlı bir tasarım yaparak işlem yükünü hafifletmeyi amaçlamışlardır. Inception mimarisinde klasik ağlara göre yaklaşık 10 kat daha az parametre hesabı yapılmaktadır. Bu 1x1 evrişim işlemi darboğaz (bottleneck) olarak tanımlanmıştır. InceptionV3 mimarisinin InceptionV2 mimarisinden farkı ise zaten var olan evrişim katmanlarının yanına yardımcı sınıflayıcı olarak, küme normalizasyonu (batch-normalized) ve tam bağlı katmanının eklenmesidir. Şekil 3.19 incelendiğinde oluşturulan mini ağlarının alt katmanlarının 3x1 evrişimli 3 çıkış birimine sahip olduğu gözlemlenmektedir.



Şekil 3.19. 3x1 mini ağ yapısı

Daha büyük filtrelere (örneğin 5x5 veya 7x7) sahip evrişimler hesaplama yükü getirmektedir. Örneğin  $m$  filtreli bir model üzerinde olan  $n$  filtreli bir 5x5 evrişimi aynı sayıda filtreye sahip 3x3 evrişimine  $25/9=2,78$  kat daha hesap yüküne sahiptir. Klasik sinir ağı modellerindeki 5x5 filtreye sahip evrişimde 3x3 lük evrişim katmanını kaydırma işlemi

yerine InceptionV3 modelinde 3x3 filtreye sahip 3x1 evrişim katmanını kaydırma işlemi yapılarak parametre sayısı azaltılmaktadır (Şekil 3.19). Bu mimarinin ilk versiyonunda  $n = \alpha \times m$  olduğu varsayıldığında, aktivasyon fonksiyonu sabiti bir  $\alpha$  değişkenine bağlı olarak değişir. 5x5 katmanının iki katman olarak değiştirilip genişletilmesiyle filtre boyutu  $\sqrt{\alpha}$  kadar değiştirilmekteydi. Fakat V3 modelinde ise  $\alpha$  değerini (genişleme yok) 1 olarak seçerek basitleştirildiğinde %28 oranında hesaplama kazancı sağlanmaktadır. Yani sonuç olarak InceptionV3 mimarisi ile 5x5 evrişim yerine, 2 katmanlı 3x3 evrişim haline gelen bir ağ yapısı elde edilmektedir (Şekil 3.20).

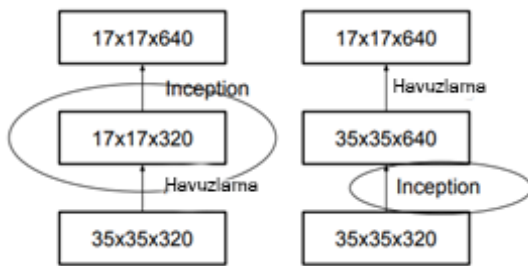


Şekil 3.20. İki katmanlı 3x3'lük evrişim katmanı

Örneğin,  $k$  filtrelili bir  $d \times d$  ızgarasından başlayarak,  $2k$  filtrelili bir  $\frac{d}{2} \times \frac{d}{2}$  d ızgarasına ulaşılacak istendiğinde, önce  $2k$  filtrelili bir evrişim hesaplanarak tek bir havuzlama adımı uygulanması gerekir. Bu daha büyük ızgaraya sahip evrişimlerde hesaplama maliyetinin  $2d^2k^2$  işlemleri kullanılarak hesaplandığında daha fazla olduğu gözlemlenmektedir. Şekil 3.19'daki evrişimli havuzlamaya geçmek için Eşitlik (8)'e göre hesaplandığında işlem maliyetini dörtte bir oranında azaltmaktadır.

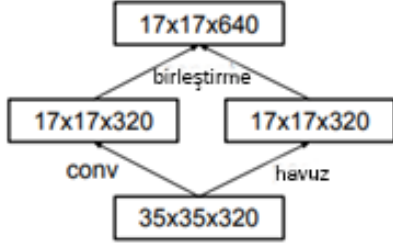
$$2 \left(\frac{d}{2}\right)^2 \times k^2 \quad (8)$$

Bununla birlikte, temsilin genel boyutluluğu  $\left(\frac{d}{2}\right)^2 \times k$ 'ye düştüğünden bu durum Şekil 3.21'deki gibi temsili bir darboğaz yaratır ve sonuç olarak daha az anlamlı ağlar yaratır.



Şekil 3.21. Izgara boyutunu azaltma prensipleri (temsili darboğaz)

Klasik ağlarda yapılan bu işlemler yerine InceptionV2 mimarisinde temsili darboğazlar ortadan kaldırılırken hesaplama maliyetini düşüren yeni bir değişken önerilmiştir (Şekil 3.22). Yeni modelde havuzlama (ortalama veya maksimum havuzlama) ve evrişim katmanı olarak iki paralel adım bloğu kullanılmıştır.

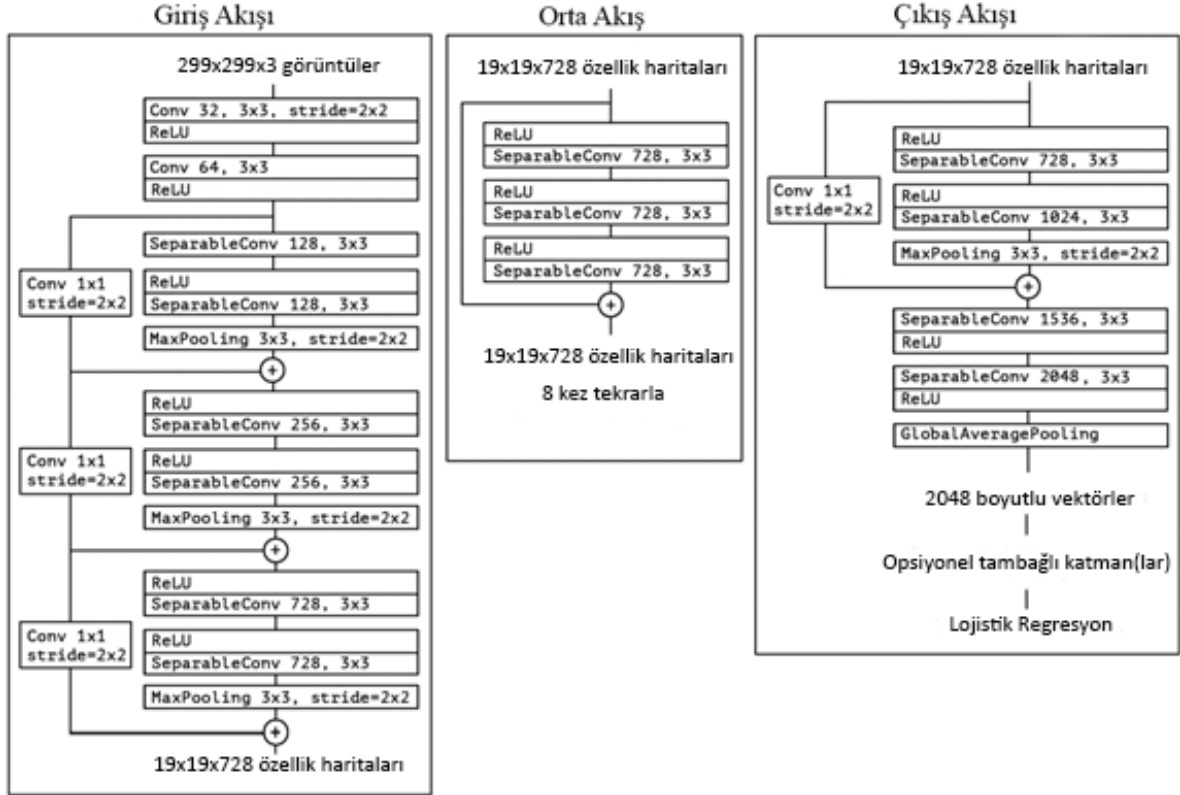


Şekil 3.22. Önerilen darboğaz yapısı

Ağdaki her bir Inception için 35x35 piksel genişliğinde 288 filtrelili 3 adet Inception modülü oluşturulmaktadır. Ardından ızgara azaltma tekniği (grid reduction) ile 17x17 piksellik 768 filtrelili gride düşürülen 5 adet Inception modeli daha sonra 8x8x1280'lik Inception modeli oluşturulur. En sonunda 2048 filtre bankası elde edilmektedir (Szegedy vd., 2016; Xia vd., 2017).

### 3.3.7 Xception ağı mimarisi

Chollet tamamen derinlemesine ayrılabilir evrişim katmanlarına dayanan evrişimli bir sinir ağı mimarisi önermiştir. Xception ağı, Inception ağ mimarisinden türetilmiş olmasından dolayı Aşırı Inception (extreme Inception) olarak adlandırılmıştır. Xception ağı tamamen bağlı tüm katmanları kaldırması ve ortalama havuzlamayı kullanması nedeniyle sistemde 5 milyon parametre bulunmaktadır. Bu model, geleneksel Inception modüllerini, noktalara özgü bir konvolüsyondan (1×1 konvolüsyon) önce gelen derinlemesine ayrılabilir konvolüsyonlarla (her giriş kanalı üzerinde ayrı ayrı gerçekleştirilen uzamsal konvolüsyon) değiştirmiştir. Inceptionda her iki işlemi de ReLU'nun doğrusal olmaması takip eder, ancak derinlemesine ayrılabilir evrişimler genellikle doğrusal olmayan durumlar olmadan uygulanır. Xception ağı ile evrişim katmanında InceptionV3'deki yeniliklere ek olarak akıllı derinlik evrişimi (depthwise convolution) ve akıllı nokta evrişimi (Pointwise convolution) olmak üzere 2 farklı yaklaşım öne çıkmıştır. Ayrıca Xception mimarisi ağın özellik çıkarma tabanını oluşturan 36 evrişim katmanına sahiptir. 36 evrişimli katman Şekil 3.23'de verildiği gibi giriş, orta ve çıkış akışı olmak üzere 3 bileşene ayrılmıştır. Veriler giriş akışından geçtikten sonra orta akışa gelir ve burada 8 kez tekrar ederek çıkış akışına geçer (Chollet, 2017; Kassani vd., 2019).



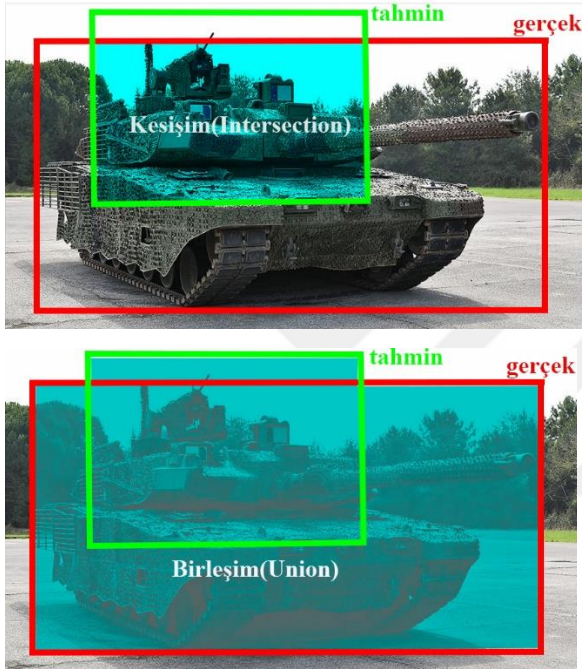
Şekil 3.23. Xception ağ mimarisi

### 3.3.8 YOLO mimarisi

#### YOLOv8 mimarisi

Nesne tespitine yönelik yeni bir yaklaşım olan You Only Look Once (YOLO), bir görüntüdeki nesnelerin bir bakışta ne olduğunu ve nerede bulduklarını tahmin etmektedir. Nesne tespitini, tek bir değerlendirmede tüm görüntülerden tek bir sinir ağı ile tahmin edebilen, uzamsal olarak ayrılmış sınırlayıcı çerçevelere ayırabilen bir algoritmaya sahiptir. Ayrıca YOLO, temel ağ olarak GoogleNet'i seçer, çünkü bu ağın hassaslığa karşı aktarım hesaplama oranları diğer ağlara göre yüksektir. YOLO, tasarımı gereği hızlıdır ve yüksek doğruluk oranını korurken gerçek zamanlı çalışabilmektedir. Oluşturulması kolay olan YOLO modeli doğrudan etiketlenmiş görüntüler üzerinde eğitilmektedir. Klasik sınıflandırma tabanlı yaklaşımlardan farklı olarak YOLO, nesne tespit performansına doğrudan karşılık gelen bir kayıp fonksiyonu üzerinde eğitilir. YOLO, çeşitli veri setlerinde diğer algılama sistemlerine göre nesne tespit işlemleri için gerçek zamanlı hız ile mükemmel doğruluk arsında dengeyi sağlayarak hızlı ve sağlam nesne algılamaya dayanan uygulamalar için ideal kılıkmaktadır (Du, 2018). YOLO mimarisinin YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6 ve YOLOv7 gibi eski versiyonları bulunmaktadır. YOLO'nun en son mimarisi olan YOLOv8, YOLOv5 modelini de üreten

Ultralytics tarafından 2023 yılında üretilmiştir (Çalışkan&Demir, 2022). YOLO’da giriş görüntüsü ilk önce  $S \times S$  ızgarasına(grid) bölünür daha sonra bölünen her bir ızgara hücresinde her biri bir güven puanına sahip  $B$  sınırlayıcı çerçeveleri tanımlanır. Buradaki güven, bir nesnenin her sınırlayıcı kutuda var olma olasılığını ifade eder ve Eşitlik (9) ve 10’a göre tanımlanır. Eşitlik (9) ve (10)’daki  $Pr$  kutunun nesneyi kapsayıp kapsamadığını, IOU gerçek kutu ile tahmin edilmiş kutu arasındaki bağlı sınıf olasılığını ifade etmektedir. Şekil 3.24’de tahmin edilen ve gerçek kutu alanlarının oluşturduğu kesişim ve birleşim alanları verilmiştir.



Şekil 3.24. Kesişim ve birleşim alanları

$$IOU_{\text{tahmin}}^{\text{gerçek}} = \frac{\text{alan}(\text{çerçeve}(\text{gerçek}) \cap \text{çerçeve}(\text{tahmin}))}{\text{alan}(\text{çerçeve}(\text{gerçek}) \cup \text{çerçeve}(\text{tahmin}))} \quad (9)$$

$$\text{Güven skoru}(\text{nesne}) = \text{Pr}(\text{nesne}) * IOU_{\text{tahmin}}^{\text{gerçek}}, \text{Pr}(\text{nesne}) \in (0,1) \quad (10)$$

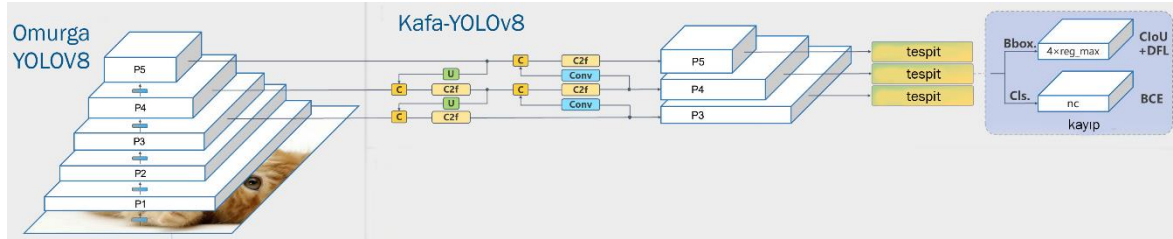
Sınırlayıcı kutular oluşturulurken eş zamanlı olarak her bir ızgara hücresi  $C$  koşullu sınıf olasılığını da tahmin eder. Her bir ızgara hücresi için sınıfa özgü olasılık eşitlik (11)’e göre hesaplanır.

$$\begin{aligned} C &= \text{Pr}(\text{Sınıf}_i | \text{Nesne}) * \text{Pr}(\text{Nesne}) * IOU_{\text{tahmin}}^{\text{gerçek}} \\ &= \text{Pr}(\text{Sınıf}_i) * IOU_{\text{tahmin}}^{\text{gerçek}} \end{aligned} \quad (11)$$

Kayıp, bir modelin performansını ölçmek için kullanılırken, nesne tespitinde modeller tarafından yapılan tahminlerin doğruluğu, gösterilen ortalama hassasiyet denklemi eşitlik (12) ile hesaplanır.

$$avgPrecision = \sum_{k=1}^n P(k) \Delta r(k) \quad (12)$$

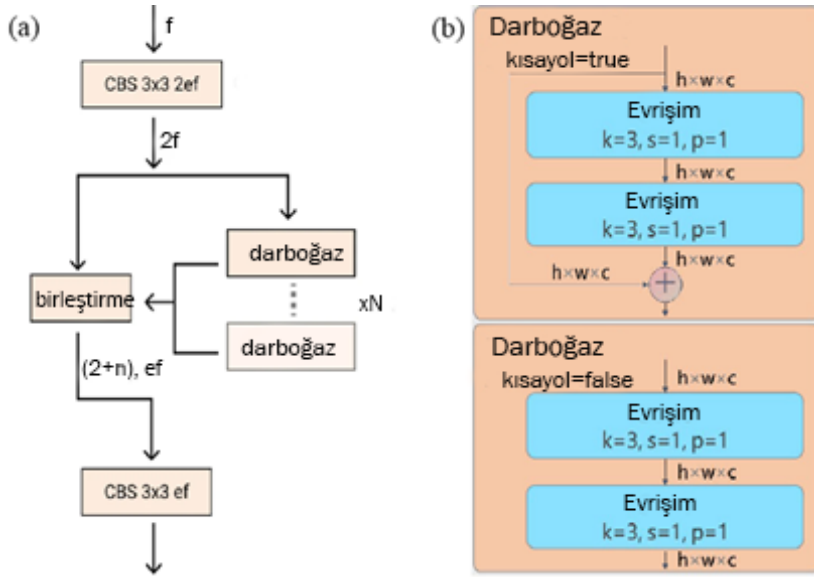
Buradaki  $P(k)$ ,  $k$  eşikindeki kesinliği ifade ederken,  $\Delta r(k)$  hatırlamadaki değişikliği ifade eder. YOLO mimarisinin nesne algılama başarısı genellikle ortalama hassasiyet (mAP) ile ölçülmektedir. YOLO'nun sinir ağı mimarisi şekil 3.25'de verildiği gibi 24 evrişimli katman ve 2 tam bağlantılı katman içerir (Huang vd., 2018; Liu vd., 2018).



Şekil 3.25. YOLOv8 ağı mimarisi

YOLO mimarisinde ağı eğitmeden önce veri setindeki nesnelerin tahmin edilmesi için önceden belirlenen yükseklik ve genişliğe sahip kutulara Anchor Box denmektedir. YOLOv8 mimarisinde ise Anchor Box'lar yoktur. Bundan dolayı bir nesnenin bilinen bir Anchor Box'a olan uzaklığı tahmin edilmeyip nesnenin merkezi tahmin edilir. Anchor Box olmadan yapılan tespit kutu tahminlerinin sayısını azaltır. Bundan dolayı potansiyel tespitleri sıralayan zorlu bir işlem gerektiren maksimum olmayanı kaldırma algoritmasını (Non-Maximum Suppression-NMS) hızlandırır. NMS algoritması gereksiz bağlantı kutularını (bounding box) kaldırmaktadır. YOLOv8 mimarisi ile gelen yenilikler ise şu şekilde sıralanabilir:

- C3 modülü yerine Şekil 3.26'da gösterilen C2f modülü kullanılmaktadır.
- Backbone kısmında olan başlangıçtaki  $6 \times 6$  evrişim katmanı,  $3 \times 3$  evrişim katmanı olarak değiştirilmiştir.
- YOLOv5 mimarisindeki 10 ve 14 nolu evrişim katmanı kaldırılmıştır.
- Şekil 3.26'da verilen darboğaz kısmındaki ilk  $1 \times 1$  Evrişim katmanını,  $3 \times 3$  Evrişim katmanı olarak değiştirilmiştir (Talaat&ZainEldin, 2023; Hussain, 2023, Sirisha vd., 2023).



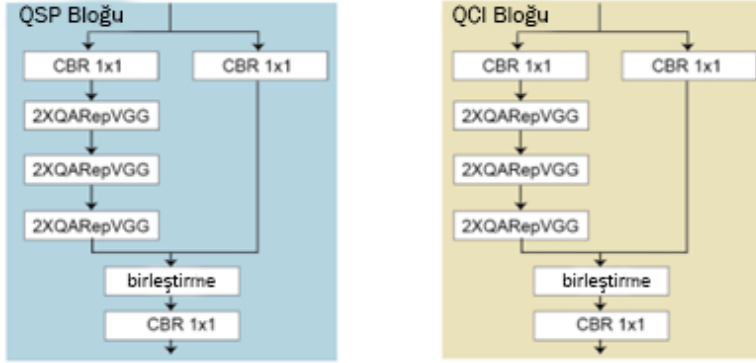
Şekil 3.26 (a)YOLOv8 C2f modülü yapısı, (b) darboğaz yapısı

### YOLO-NAS Mimarisi

Network Architecture Search (NAS, Ağ Mimarisi Arama), otomatik makine öğrenimi (AutoNAC) içerisinde yer alan bir alt alandır. NAS, araştırmacılar tarafından manuel olarak tasarlanan, yapılandırılan ve inşa edilen benzerlerinden daha iyi performans göstermek amacıyla derin sinir ağı mimarilerini otomatik olarak tasarlayan ve yapılandıran makine öğrenimi modellerinin geliştirilmesi sürecidir. NAS teknikleri aracılığıyla üretilen yeni mimari tasarım ve geliştirme çabaları, doğal dil işleme, nesne algılama ve görüntü sınıflandırma gibi görevlerde insan mühendisliği mimarilerini geride bırakmıştır. NAS teknikleri, mimarileri akıllı telefonlar veya gerçek zamanlı çalışılacak senaryolara uyarlamak için çok önemli olan hesaplama kaynakları, verimlilik, doğruluk, güç tüketimi ve bellek kullanımı gibi faktörleri optimize etmeye odaklandıkları için günümüzde daha da önemli hale gelmiştir.

YOLO-NAS mimarisi Deci'nin araştırma ve mühendislik ekibi tarafından oluşturulmuştur. Mimarinin eğitim süreci çok aşamalı ve yüksek donanım gerektirmesine rağmen, ortaya çıkan modeller düşük gecikmeyle çalışmakta, yüksek doğruluk skoruyla nesne algılama sonuçları sağlamakta ve daha karmaşık nesne algılama görevleri için ince ayar kolaylığı sağlamaktadır. YOLO-NAS mimarisi ile eğitilen modeller uç cihazlar (mobil telefonlar) gibi düşük bilgi işlem ortamlarında rahatlıkla kullanılmaktadır. YOLO-NAS'ın oluşturulmasında etkili olan AutoNAC sistemi çok yönlüdür ve her türlü göreve, verilerin özelliklerine, çıkarım yapma ortamına ve performans hedeflerinin belirlenmesine uyum sağlayabilmektedir. Kullanıcılara, kendi özel kullanımları için hassasiyet ve çıkarım

hızının mükemmel karışımını sunan en uygun parametreleri belirlemelerine yardımcı olmaktadır. Ayrıca, eğitim sonrası niceleme ve uyumluluk için NAS süreci sırasında RepVGG blokları model mimarisine dahil edilmiştir. YOLO-NAS mimarisinde Şekil 3.27’de verilen QSP ve QCI bloklarının derinlik ve konumları değiştirilerek YOLO-NAS-S (küçük), YOLO-NAS-M (orta) ve YOLO-NAS-L (büyük) adında üç mimari oluşturulmuştur (Terven vd., 2023; Choutri vd., 2023).



Şekil 3.27. YOLO-NAS Mimarisi, QSP ve QCI blokları

### 3.4. Performans Metrikleri

Eğitilmiş bir modelin başarısını, performansını değerlendirmek için çeşitli performans ölçütleri kullanılmaktadır. Bu ölçütler arasında karışıklık matrisi, doğruluk (accuracy), kesinlik (precision), duyarlık (recall), ortalama kesinlik değerlerinin ortalaması ve F1-skor vardır. Sınıflandırma problemlerinde, performans ölçümlerinin birincil kaynağı olarak bir karışıklık matrisi kullanılır. Karışıklık matrisine sınıflandırma matrisi veya bir olasılık tablosu da denmektedir.

Tablo 3.3. Karışıklık matrisi

		GERÇEK	
		Pozitif	Negatif
TAHMİN	Pozitif	Doğru Pozitif TP	Yanlış Pozitif FP
	Negatif	Yanlış Negatif FN	Doğru Negatif TN

Doğru Pozitif (TP): Modelinin pozitif olarak doğru tahmin ettiği sonuç sayısıdır.

Doğru Negatif (TN): Gerçekte negatif olan verileri eğitilen modelin negatif olarak tahmin ettiği doğru sonuç sayısıdır.

Yanlış Pozitif (FP): Gerçekte negatif olan verileri eğitilen modelin yanlış yaparak tahminini pozitif olarak işaretlediği sonuç sayısıdır.

Yanlış Negatif (FN): Gerçekte pozitif olan verileri modelin yanlış yaparak tahminini negatif olarak işaretlediği ettiği sonuç sayısıdır

Karışıklık matrisinden hesaplanabilen, en sık kullanılan metriklerin denklemleri eşitlik 13, 14, 15, 16'da verilmiştir.

$$\text{Doğruluk} = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (13)$$

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (14)$$

$$\text{Duyarlık} = \frac{TP}{TP+FN} \quad (15)$$

$$F1 - \text{Skor} = 2 \times \frac{\text{duyarlık}}{\text{kesinlik} + \text{duyarlık}} \quad (16)$$

F1 skoru, kesinlik (precision) ve duyarlılık (recall) değerlerinin harmonik ortalaması alınarak hesaplanmaktadır. Veri setlerindeki sınıflar eşit dağılıma sahip değilse veri setlerinde doğru modeli seçmek için F1-skora bakılmalıdır (Olson&Delen, 2008; Godbole&Sarawagi, 2004).

Nesne tespit algoritmalarının performans değerlendirmeleri de belirli metriklerle yapılmaktadır. Özellikle hedef görüntüdeki nesnelerin tespit edilesi için doğruluk (accuracy) hesaplaması tek başına yeterli gelmemektedir. Bu nedenle nesne tespit algoritmalarının performanslarını değerlendirmek için F1 skor ve genel ortalama hassasiyet (mAP) gibi metrikler kullanılır.

Hesaplanan kesinlik ( $p$ ) ve duyarlılık ( $r$ ) değerleri kullanılarak ortalama hassasiyet (AP) eşitlik (17)'e göre hesaplanır.

$$\text{Ortalama Hassasiyet}(AP) = \int_{r=0}^1 p(r) dr \quad (17)$$

Veri setinde bulunan tüm sınıflar için ortalama hassasiyet hesaplanıp ortalamaları Eşitlik 17'e göre alınırsa genel ortalama hassasiyet (mAP) elde edilir ve modelin başarısı bu değerlerin durumuna göre değerlendirilir (Yohanandan vd., 2018; Flach ve Kull, 2015).

### 3.5. Modellerin Eğitilmesi

ESA ile yapılan sınıflandırma işlemlerinde modelin başarısının artırılması için kullanılan birçok özellik bulunmaktadır. Derin öğrenme yöntemlerinde daha az hesaplama maliyetli ve daha az donanıma ihtiyaç duyan ideal bir model yaratılmak istendiğinde veya model başarısının artırılmasına yönelik çeşitli parametreler kullanılmaktadır. Bunlara hiper-parametre denilmektedir. ESA ile yapılmış modellerin eğitilmesinde aktivasyon

fonksiyonu, eğitim öncesi girişteki görüntülerin boyutu (piksel), öğrenme katsayısı (learning rate), mini-batch size, filtre boyutu, seyreltme (dropout), katman sayısı, epok gibi hiper-parametreler kullanılmaktadır. Hiper-parametrelerin ayarlanması derin sinir ağları için etkili bir makine öğrenimi modeli oluşturmanın önemli bir bileşeni olarak kabul edilir. (ZahediNasab& Mohseni, 2020; Yang&Shami, 2020; Kuş, 2019; Yu&Zhu, 2020).

### 3.5.1 Öğrenme oranı (Learning rate)

Öğrenme oranı model ağırlıklarının, geriye yayılım (backpropagation) sırasında değiştirilerek güncellenmesi işlemine denmektedir. Öğrenme oranı parametresi yapılan modelde büyük bir değer olarak girildiğinde mimari hızlı bir şekilde özellikleri öğrenir fakat mimari büyük adımlarla öğrenmeye çalışarak en uygun ve doğru sonucu bulamayabilir. Bunun nedeni kullanılan mimari yüksek doğruluk üretebileceği ağırlık değerlerini ıskalamasından kaynaklanmaktadır. Öğrenme parametresi olarak çok küçük bir değer girilmesi halinde ise ağırlıklar küçük değerlerle güncellenir ve optimum ve doğru sonuç bulunmayabilir. Bu durum eğitim süresinin çok uzun sürmesine yüksek donanım gereken bilgisayar ihtiyacına yol açar. Bu nedenlerden dolayı yapılacak modelde en uygun öğrenme oranı parametresinin bulunması gerekir. Öğrenme oranı parametreleri genelde 0,1'den başlanarak 1 değerleri azaltılarak belirlenmektedir (Masters&Luschi, 2018; Feng& Li, 2018; Eldem, 2023).

### 3.5.2 Maksimum devir sayısı (Epok)

Ağın eğitiminde kullanılacak tüm veri setinin tüm ağ boyunca çalışacağı tekrar sayısına maksimum devir sayısı denmektedir. Eğitim süresince modelin doğru değerleri bulması için ideal epok sayısının girilmesi gereklidir. Eğer düşük bir epok değeri girilirse model eğitimi tamamlayamayabilir. Yüksek bir epok değeri girilirse model aşırı öğrenme eğilimine girebilir, ezberleyebilir dolayısı ile genelleme yeteneği düşer. ESA'ları kullanılarak eğitilen modellerde genelde epok sayısı olarak 10 ile 150 arası değerler girilmektedir (Shorten& Khoshgoftaar, 2019; Reznikov vd., 2020).

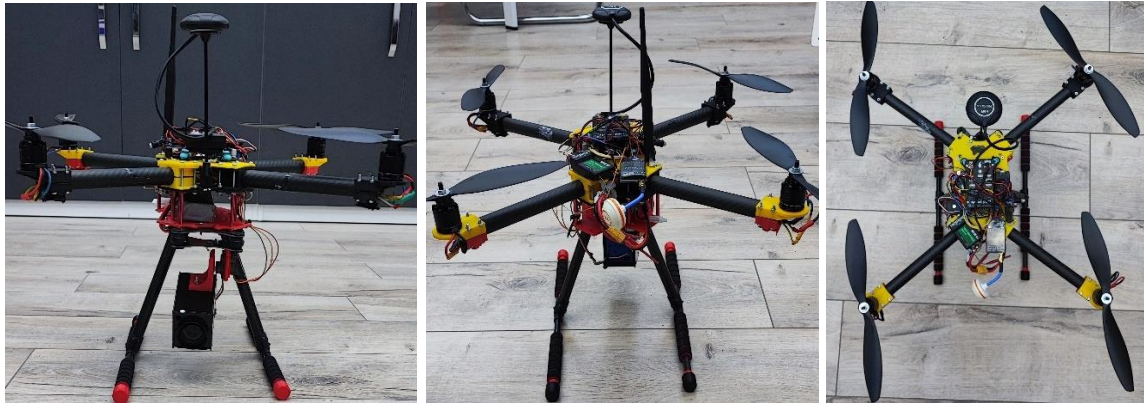
### 3.5.3 Yığın boyutu (Batch Size)

ESA mimarisinde eğitim için verilen alt örneklerin sayısına yığın boyutu denmektedir. Belirlenen örnekler, girilen her bir epok kadar ESA'nın sonuna kadar gider ve ağın hata oranı hesaplanır ve ağın ağırlıkları güncellenmiş olur yığın boyutu sayısı genelde 2'nin katları şekilde girilirken eğitimin yapıldığı donanım gözetilerek belirlenmektedir. Eğer çok

büyük yığın boyutu girilirse bilgisayar donanımı bunu karşılayamaz ve eğitim yapılamayabilir (Masters ve Luschi, 2018).

### 3.6. Yapılan İHA

Literatürde İHA'lar üzerinde insan pilot barındırmayan, belirli amaca yönelik çalışan hava araçları olarak tanımlanmıştır (Clarke, 2014; Restas, 2015). İHA'lar kanatlarının türene göre sabit kanat ve döner kanat olmak üzere ikiye ayrılmaktadır (Korchenko&Illyash, 2013). QuadCopter olarak da bilinen dronelar, dört rotorlu bir hava aracıdır. Dronelar fırçasız elektrikli motorlarla döndürülen rotorların açısız hızları ayarlanarak kontrol edilmektedir. Dronelar basit ve küçük yapısından dolayı İHA sınıfına ait tasarlanmış hava araçlarıdır. Dronelar askeri alan, tarım, gözetleme, arama ve kurtarma, inşaat denetimleri ve diğer birçok alanda kullanılmaktadır (Luuukkonen, 2011). İHA'lar havanın bulutlu olmasından etkilenmediğinden İHA ile çekilen fotoğraflar ile arazinin tamamı belirli amaca göre görüntülenebilir. İHA'ların diğer hava araçlarına göre yüksek manevra kabiliyetinin olmasından dolayı yapılacak işlerde daha az zaman ve maliyet harcanmaktadır (Urbahs&Jonaite, 2013). Çalışmada nesnelere havadan alınacak görüntülerle sınıflandırmak veya tespit etmek için 4 rotorlu döner kanatlı bir İHA yapılmıştır (Resim 1).

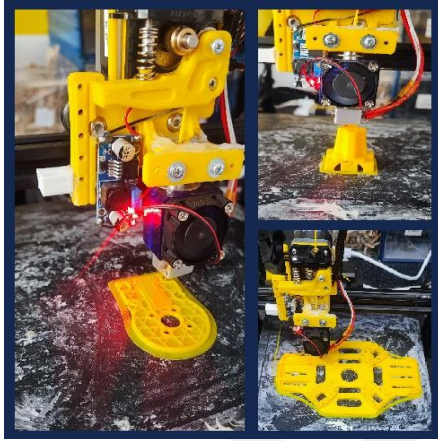


Resim 3.1. Yapılan İHA

#### 3.6.1 İHA Gövde Tasarımı ve Üretimi

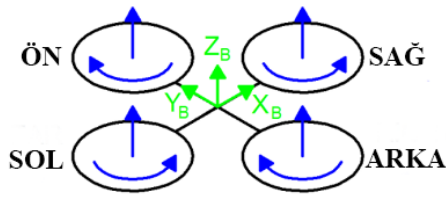
SketchUp, 3D modelleme görüntüsünü hızlı bir şekilde derlemek, değiştirmek ve oluşturmak için kullanılan dinamik bir yazılım programıdır. SketchUp yazılımı Google ve Trimble şirketi tarafından geliştirilip pazarlanmıştır. Bu yazılım genellikle mimari, tasarım ve mühendislik alanlarında kullanılmaktadır (De Yong vd., 2020). Dronun gövdesi, motor tutucuları, iniş takımı gibi parçalar SketchUp programıyla çizilmiştir. Kanatlar ve iniş takımı ayakları karbonfiber borular kullanılarak yapılmıştır. Katmanlı üretim tekniği,

bilgisayarla üç boyutlu çizilmiş tasarımlardan fiziksel modeller elde etmeyi sağlayan teknolojilere denmektedir. Hızlı prototipleme cihazlarından olan 3 boyutlu yazıcılarla bilgisayarda üç boyutlu çizimi yapılmış modellerin saatler içerisinde üretilmesine olanak sağlanmıştır (Aktimur&Gökpınar, 2015). Dronun tasarlanan parçalarının baskıları 3d yazıcılar ile alınarak gövdesi oluşturulmuştur.



Resim 3.2. Katmanlı üretim aşaması

4 kanatlı X tipindeki dronelar(quadcopter) ön sol ve arka sağ kanatlardaki motorların saat yönünde (CW), ön sağ ve arka sol kanatlardaki motorların ise saat yönünün tersine (CCW) dönmesiyle oluşturulmaktadır (Şekil 3.28). Araç hareketinin kontrolü, bir veya daha fazla rotor diskinin dönüş hızı değiştirilerek, dolayısıyla tork yükü ve itme/kaldırma özellikleri değiştirilerek sağlanmaktadır (Ostojic vd. 2015).

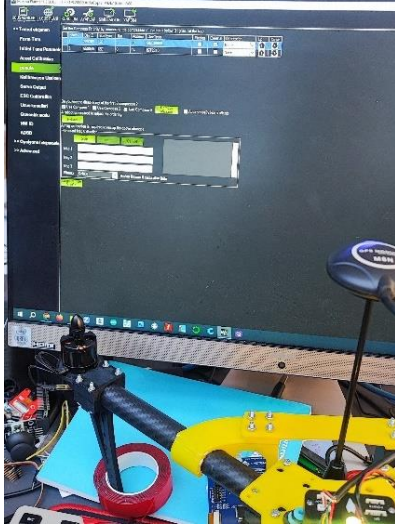


Şekil 3.28. X tipindeki dronun motor dönüş yönleri

Elektronik hız kontrol ünitesi (ESC), kendisine uygulanan voltajı değiştirerek motor hızını ve yönünü değiştirebilir. Ayrıca motorlar için kontrollü başlatma, motor dönüşünü tersine çevirme veya ters akım koruması sağlayabilir. İHA üzerine eklenecek ESC'ler belirlenirken kullanılacak motorun anlık çekeceği akımlar(A) dikkate alınmalıdır. Fırçasız doğru akım (DC) motorları, quadcopterlerde en çok kullanılan bir motor türüdür. Fırçasız motorlar, doğru akım ile çalıştırılan sabit mıknatıslı motordur. Fırçasız motorlarda kullanılan kV değeri, motorda yük yokken bir volt uygulandığında motorun dakikada döneceği devir sayısına denmektedir. Uçuş kontrol kartı dronun kontrol merkezi olmakla

birlikte üzerindeki sensörler sayesinde dronun yönünü algılayan, kullanıcının komutlarını ve kontrollerini motorlara aktaran, temel bir kontrol devre kartıdır. Akıllı telefonlar, dizüstü bilgisayarlar veya tabletlerde en çok kullanılan pil türü, Lityum Polimerdir (LiPo). LiPo pillerin diğer pil türlerine göre yüksek kapasiteye sahip olması ve hafif olması gibi bazı önemli avantajları vardır (Mogensen, 2016; Lian vd., 2021; Praveen&Pillai, 2016; Mitra&Land). İHA'nın havada konumunun sabitlemesi veya belirlenen bir konuma yönlendirilmesi için GPS (Global Position System) modülleri kullanılmaktadır. İHA'ların uzaktan otonom yönlendirilmesi, İHA üzerindeki sensörlerden alınan bilgilerin görülmesi, İHA'nın konum bilgisi veya pilin voltaj değişimlerinin izlenmesi ve İHA'nın parametrelerinin değiştirilmesi için yer istasyonu programları kullanılmaktadır. Telemetri modülü yer istasyonu programıyla drone arasındaki haberleşmeyi sağlayıp veri aktarımından sorumludur. Yer istasyonu ile çizilecek rotanın drona aktarılması ve dronun da çizilen rotada otonom uçuşlar yapması için telemetri modülü kullanılmaktadır. Yer istasyon programlarından biri olan Mission Planner ile yeni yapılacak dronun ayarları yapılabilir (Ramirez& Cahacho, 2018; Cunta, 2015; Sharma&Rajesh, 2018; Ismail, 2020). FPV (First Person View) alıcı (Receiver) ve verici (Transmitter) modülleri kamera üzerinden alınan görüntüleri uzak mesafedeki alıcısına ileterek başkasının gözünden görme hissi yaratmaktadır. Alıcı ve verici modüller radyo frekansıyla çalışmakta olup kanal seçimine olanak sağlayarak sinyallerin karışmasını önlemektedir. İHA'lar üzerine eklenen verici modüller, çok uzak mesafelerdeki alıcı modüle görüntü aktarımı yapabilmektedir (Kawasaki, 2010).

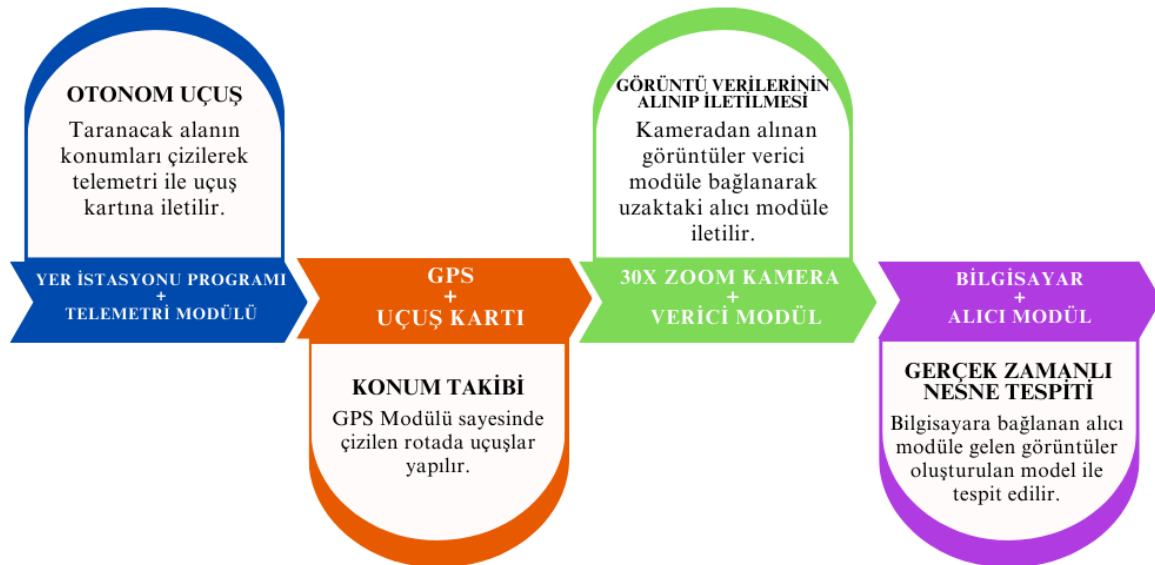
Dronun gövdesi oluşturulduktan sonra uçuş kartı, escler, motorlar, harici gps ve pusula modülü, telemetri modülü, görüntü verici modüller gövde üzerine eklenerek elektrik bağlantıları yapılmıştır. Dronun havalanması için gerekli olan 4 adet 700kV yüksek torklu fırçasız motor ve 4 adet anlık 30 Amper akım sağlayabilen Esc kullanılmıştır. Motorların ve Esclerin bağlantıları Şekil 3.27'de verilen dönüş yönlerine uygun olarak yapılmıştır. Yer istasyonu programı olan Mission Planner ile uçuş kartının ve esclerin kalibrasyon ayarları yapılarak dron uçuşa hazır hale getirilmiştir (Resim 3.3).



Resim 3.3. Mission planner yer istasyonu programı

### 3.7. Sistem Çalışma Prensibi

Otonom uçuş için rota yer istasyonu programı ile çizilerek verici telemetri modülü aracılığı ile uçuş kartına iletilecektir. İHA üzerindeki alıcı telemetri modüle gelen rota bilgilerini alıp otonom uçuşlara başlayacaktır. Otonom uçuş başladığında 30X yakınlaştırma özelliğine sahip kamera çalışarak radyo frekansıyla çalışan görüntü verici modüle aldığı anlık görüntüleri verecektir. Görüntü verici modül yer istasyonu programının olduğu uzaktaki bilgisayara bağlanan alıcı modüle görüntüleri anlık olarak ileticektir. Yer istasyonu bilgisayarına gelen anlık görüntüler ile eğitilen model görüntüleri analiz edip nesne tespiti ve sınıflandırma işlemlerini yapacaktır. Gelen görüntüler üzerinde eğitilen nesnelere tespit edilirse bildirim verecektir. Yapılan sistemin çalışma prensibi Şekil 3.29’da verilmiştir.



Şekil 3.29 Sistem çalışma prensibi şeması

## 4. BULGULAR ve TARTIŞMA

Bu çalışmada sınıflandırma modellerinin eğitilmesi işlemleri için 20 çekirdekli 2.4 ghz I7 13700H işlemcili, 32 GB ramli, 6GB ekran kartlı bir bilgisayar kullanılmıştır. Nesne tespit algoritmaları için ise ekran kartı destekli Google Colab kullanılmıştır. Çalışmada modellerin eğitilmesi için gereken kodlar Jupyter programı kullanılarak Python dilinde yazılmıştır. Derin öğrenme modellerinin eğitimi için kullanılan hibrit ve yeni bir veri seti Kaggle platformundaki 6 adet veri setindeki görüntüler ve telif hakkı olmayan sitelerden alınan görüntülerden alınarak oluşturulmuştur. Veri setindeki görüntüler Python dilinde yazılan kod ile test, train ve validation klasörüne rasgele seçilerek yerleştirilmiştir. Veri setindeki görüntülerin %70'i train, %15'i test, %15'i de validation verisi olarak ayrıştırılmıştır. Görüntü verilerinin klasörlere rastgele ayrıştırılması için Python dilinde kod yazılmıştır. Klasörlerdeki sınıflara ait görüntü sayıları Tablo 4.1'de verilmiştir.

Tablo 4.1 Eğitim, test ve doğrulama görüntü verileri sayıları

Klasör Adı	Sınıf Adı						Toplam Görüntü Sayısı
	Askeri Kamyon	Füze Atar	Silahlı Asker	Sivil	Sivil Asker	Tank	
Train	419	375	648	451	356	685	2934
Test	89	80	138	96	76	146	625
Validation	91	82	140	98	77	148	636

Aşırı öğrenmenin engellenmesi için veri çoğaltma tekniklerinden faydalanılmıştır. Nesne tespit algoritmaları için gerekli olan etiketleme işlemleri “makesense.ai” sitesinde yapılmıştır. Derin öğrenme modellerinin doğruluk oranının artırılması için epok, batchsize ve learning rate değişkenleri belirlenmiştir. Model eğitim sürecinin gereksiz yere uzun sürmemesi için kayıp değerini kontrol eden erken durdurma kodu yazılmıştır. Değişkenlerin değerleri Tablo 4.2’de verilmiştir.

Tablo 4.2 Derin öğrenme ve Nesne tespit mimarilerinde kullanılan parametreler

Parametreler	Değerler
Epok	50
Öğrenme Oranı	0.0001
Yığın boyutu	64

### 4.1. Sınıflandırma Modelleri ile Gerçekleştirilen Analiz Sonuçları

Çalışmada ResNet50, InceptionV3, Xception, Mobilenet, EfficientNet gibi 5 farklı transfer öğrenme mimarisi ile çalışılmış olup performans karşılaştırmaları yapılmıştır. Objektif bir değerlendirme yapabilmek için her bir modele görüntü boyutları (genişlik x yükseklik) hariç aynı parametreler girilmiştir. Modellerin performansları kayıp değeri, doğruluk oranı

ve f1-skoru baz alınarak değerlendirilmiştir. Modellere girilen görüntü boyutları Tablo 4.3'de verilmiştir. Modellere girilen görüntü boyutları belirlenirken mimarilere girilmesi gereken görüntü boyutları ölçüt alınmıştır.

Tablo 4.3 Modellere girilen görüntü boyutları

Model Adı	Girilen görüntü boyutu (Genişlik x Yükseklik)
ResNet50	224 x 224
InceptionV3	299 x 299
Xception	299 x 299
MobileNetV2	224 x 224
EfficientNet	224 x 224
YOLOv8	640 x 640

#### 4.1.1. ResNet50 Modeli ile Gerçekleştirilen Analiz Sonuçları

ResNet50 mimarisi ile elde edilen kesinlik, duyarlılık ve F1-skor değerleri Tablo 4.4'de verilmiştir. ResNet50 mimarisinin F1 skora göre performansı değerlendirildiğinde tanklar en yüksek skor ile sınıflandırılırken askeri kamyonlar en düşük skorla sınıflandırılmıştır.

Tablo 4.4 ResNet50 mimarisinin sınıf bazında performansı

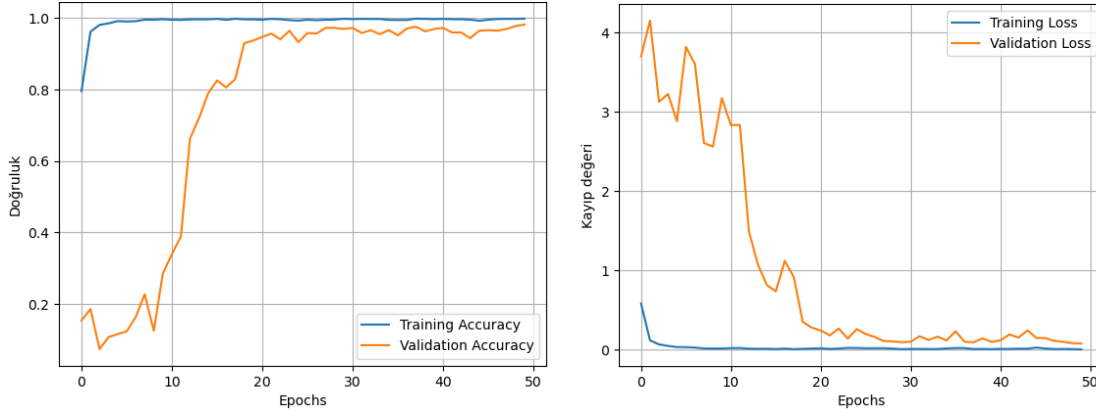
Sınıf Adı	Kesinlik	Duyarlılık	F1-Skor
askeri kamyon	0,9462	0,9888	0,9670
füze atar	0,9872	0,9625	0,9747
silahlı asker	0,9786	0,9928	0,9856
sivil	0,9892	0,9583	0,9735
sivil araç	0,9867	0,9837	0,9801
tank	0,9999	0,9999	0,9999

ResNet50 mimarisi ile tüm sınıfların sınıflandırılmasında elde edilen doğruluk oranları Tablo 4.5'de verilmiştir. Tablo 4.5 incelendiğinde tüm sınıfların sınıflandırılması işleminde 0.9824'lük doğruluk oranı elde edilmiştir.

Tablo 4.5 ResNet50 mimarisi doğruluk skoru

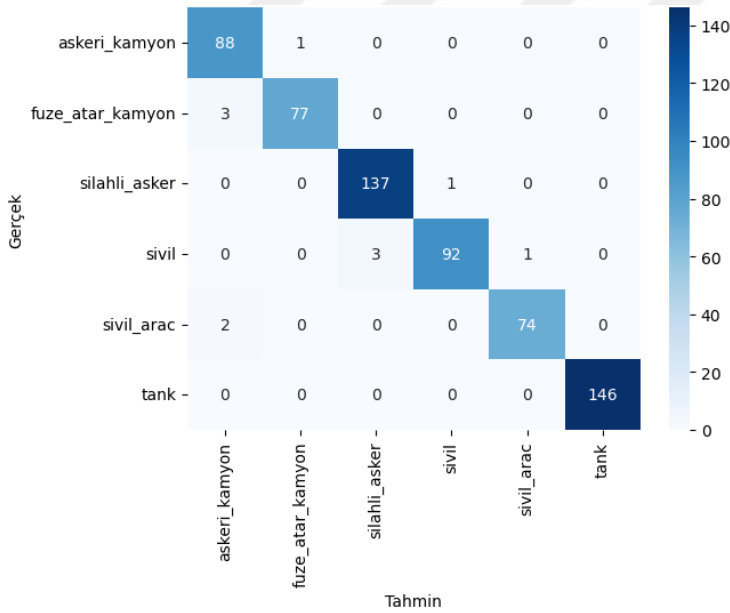
Veri Türü	Accuracy/Doğruluk
Train	0,9997
Test	0,9824

ResNet50 mimarisi ile elde edilen doğruluk ve kayıp değerleri Şekil 4.1'de verilmiştir. Şekil 4.1 incelendiğinde en iyi doğruluk oranının 0,9824 ile, en düşük kayıp değeri olarak ise 0,0789 ile 48. epokta elde edildiği gözlemlenmiştir.



Şekil 4.1 ResNet50 epok sayısına göre doğruluk ve kayıp değeri eğrileri

ResNet50 mimarisinden elde edilen karışıklık matrisi Şekil 4.2'de verilmiştir. Matris incelendiğinde askeri kamyonlar sınıflandırılırken 88 tanesi doğru 1 tanesi yanlış, füze atar kamyonlar sınıflandırılırken 77 tanesi doğru 3 tanesi yanlış, silahlı asker sınıflandırılırken 137 tanesi doğru 1 tanesi yanlış, sivil araçlar sınıflandırılırken 74 tanesi doğru 2 tanesi yanlış olarak sınıflandırılmıştır. Tanklar sınıflandırılırken ise hiçbir hata yapılmayıp görüntülerin tamamı doğru sınıflandırılmıştır.



Şekil 4.2 ResNet50 mimarisi karışıklık matrisi sonuçları

#### 4.1.2. Inceptionv3 Modeli ile Gerçekleştirilen Analiz Sonuçları

InceptionV3 mimarisi ile elde edilen kesinlik, duyarlık ve F1-skor değerleri Tablo 4.6'da verilmiştir. InceptionV3 mimarisinin F1 skora göre performansı değerlendirildiğinde tanklar en yüksek skor ile sınıflandırılırken füze atar kamyonlar en düşük skorla sınıflandırılmıştır.

Tablo 4.6 InceptionV3 mimarisinin sınıf bazında performansı

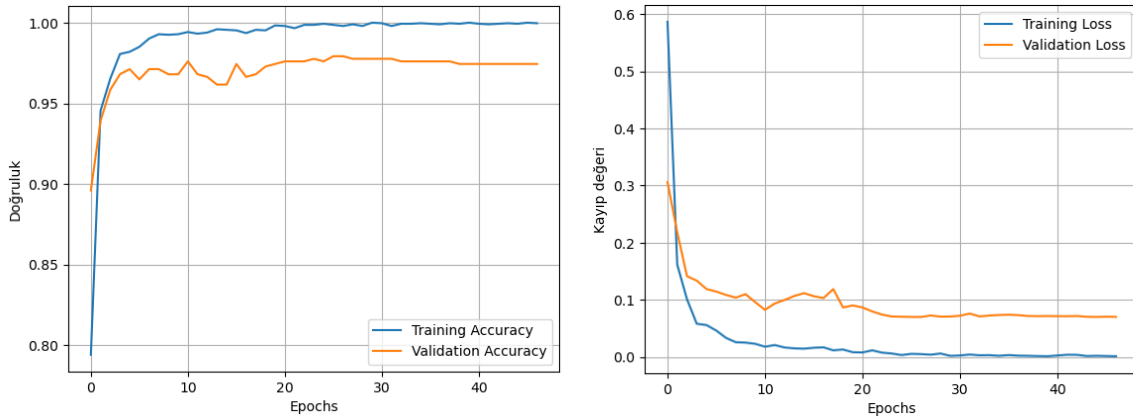
Sınıf Adı	Kesinlik	Duyarlılık	F1-Skor
askeri kamyon	0,9655	0,9438	0,9545
füze atar	0,9625	0,9625	0,9625
silahlı asker	0,9856	0,9928	0,9892
sivil	0,9792	0,9792	0,9792
sivil araç	0,9737	0,9737	0,9737
tank	0,9932	0,9999	0,9966

InceptionV3 mimarisi ile tüm sınıfların sınıflandırılmasında elde edilen doğruluk oranları Tablo 4.7’de verilmiştir. Tablo 4.7 incelendiğinde tüm sınıfların sınıflandırılması işleminde 0.9792’lik doğruluk oranı elde edilmiştir.

Tablo 4.7 InceptionV3 mimarisi doğruluk skoru

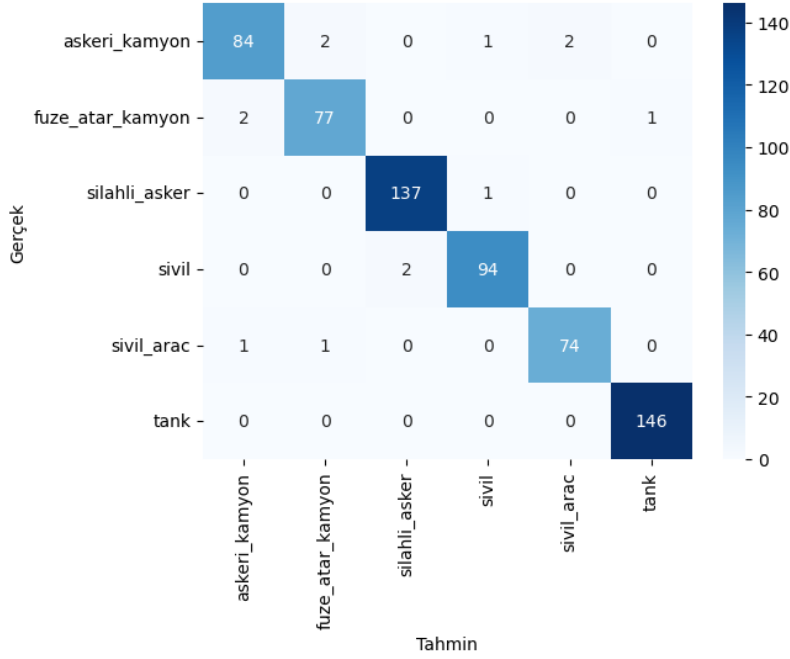
Veri Türü	Accuracy/Doğruluk
Train	0,9997
Test	0,9792

InceptionV3 mimarisi ile elde edilen doğruluk ve kayıp değerleri Şekil 4.3’de verilmiştir. Şekil 4.3 incelendiğinde en iyi doğruluk oranının 0,9792 ile, en düşük kayıp değeri olarak ise 0,0701 ile 43. epokta elde edildiği gözlemlenmiştir.



Şekil 4.3 InceptionV3 epok sayısına göre doğruluk ve kayıp değeri eğrileri

InceptionV3 mimarisinden elde edilen karışıklık matrisi Şekil 4.4’de verilmiştir. Matris incelendiğinde askeri kamyonlar sınıflandırılırken 84 tanesi doğru 5 tanesi yanlış, füze atar kamyonlar sınıflandırılırken 77 tanesi doğru 3 tanesi yanlış, silahlı askerler sınıflandırılırken 137 tanesi doğru 1 tanesi yanlış, siviller sınıflandırılırken 94 tanesi doğru 2 tanesi yanlış, sivil araçlar sınıflandırılırken 74 tanesi doğru 2 tanesi yanlış olarak sınıflandırılmıştır. Tanklar sınıflandırılırken ise hiçbir hata yapılmayıp görüntülerin tamamı doğru sınıflandırılmıştır.



Şekil 4.4 InceptionV3 mimarisi karışıklık matrisi sonuçları

#### 4.1.3. Xception Modeli ile Gerçekleştirilen Analiz Sonuçları

Xception mimarisi ile elde edilen kesinlik, duyarlılık ve F1-skor değerleri Tablo 4.8’de verilmiştir. Xception mimarisinin F1 skora göre performansı değerlendirildiğinde sivil araçlar en yüksek skor ile sınıflandırılırken askeri kamyonlar en düşük skorla sınıflandırılmıştır.

Tablo 4.8 Xception mimarisinin sınıf bazında performansı

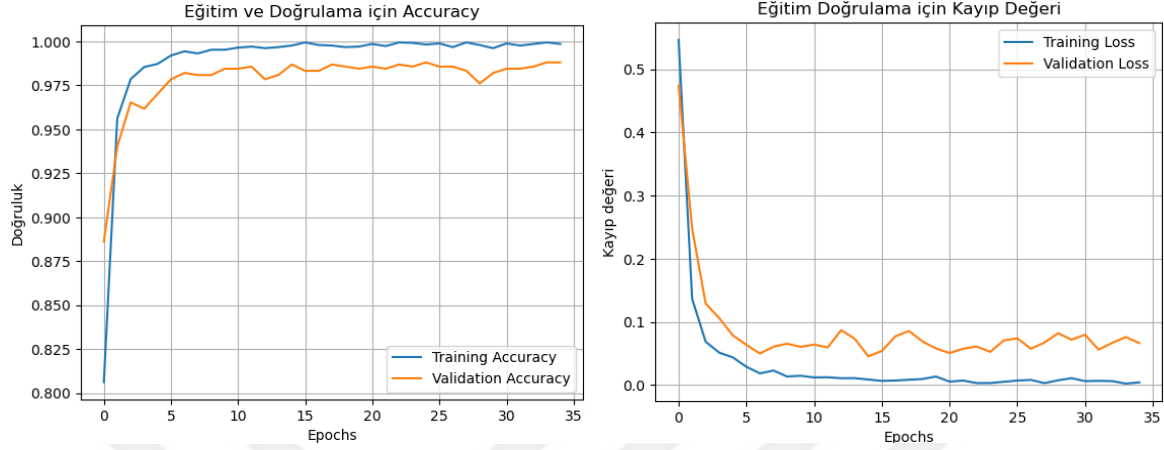
Sınıf Adı	Kesinlik	Duyarlılık	F1-Skor
askeri kamyon	0,9659	0,9551	0,9605
füze atar	0,9625	0,9625	0,9625
silahlı asker	0,9786	0,9897	0,9856
sivil	0,9688	0,9692	0,9695
sivil araç	0,9999	0,9999	0,9997
tank	0,9924	0,9932	0,9932

Xception mimarisi ile tüm sınıfların sınıflandırılmasında elde edilen doğruluk oranları Tablo 4.9’da verilmiştir. Tablo 4.9 incelendiğinde tüm sınıfların sınıflandırılması işleminde 0,9808 doğruluk oranı elde edilmiştir.

Tablo 4.9 Xception mimarisi doğruluk skoru

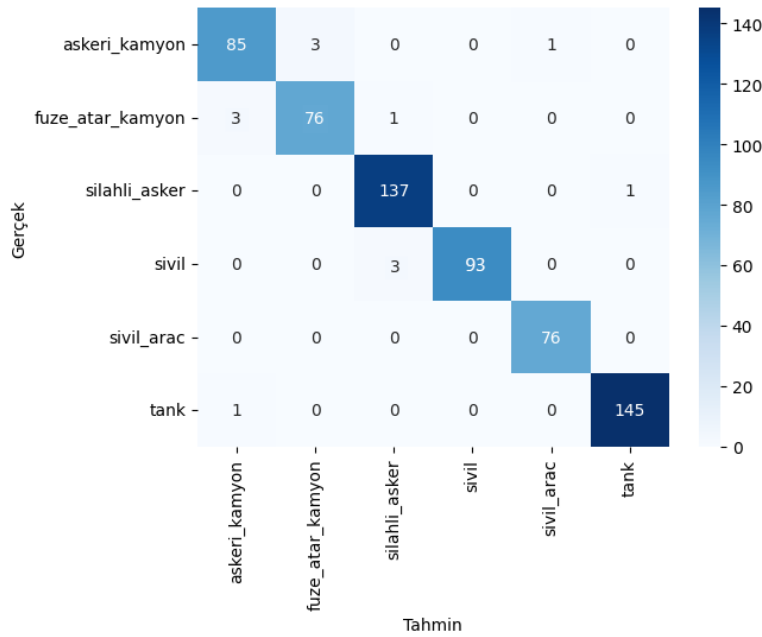
Veri Türü	Accuracy/Doğruluk
Train	0,9982
Test	0,9808

Xception mimarisi ile elde edilen doğruluk ve kayıp değerleri Şekil 4.5’de verilmiştir. Şekil 4.5 incelendiğinde en iyi doğruluk oranının 0,9808 ile, en düşük kayıp değeri olarak ise 0,0906 ile 34. epokta elde edildiği gözlemlenmiştir.



Şekil 4.5 Xception epok sayısına göre doğruluk ve kayıp değeri eğrileri

Xception mimarisinden elde edilen karışıklık matrisi Şekil 4.6’da verilmiştir. Matris incelendiğinde askeri kamyonlar sınıflandırılırken 85 tanesi doğru 4 tanesi yanlış, füze atar kamyonlar sınıflandırılırken 76 tanesi doğru 4 tanesi yanlış, silahlı askerler sınıflandırılırken 137 tanesi doğru 1 tanesi yanlış, siviller sınıflandırılırken 93 tanesi doğru 3 tanesi yanlış, tanklar sınıflandırılırken 145 tanesi doğru 1 tanesi yanlış olarak sınıflandırılmıştır. Sivil araçlar sınıflandırılırken ise görüntülerin tamamı doğru olarak sınıflandırılmıştır.



Şekil 4.6 Xception mimarisi karışıklık matrisi sonuçları

#### 4.1.4. MobileNet Modeli ile Gerçekleştirilen Analiz Sonuçları

MobileNet mimarisi ile elde edilen kesinlik, duyarlılık ve F1-skor değerleri Tablo 4.10'da verilmiştir. MobileNet mimarisinin F1 skora göre performansı değerlendirildiğinde tanklar en yüksek skor ile sınıflandırılırken askeri kamyonlar en düşük skorla sınıflandırılmıştır.

Tablo 4.10 MobileNet mimarisinin sınıf bazında performansı

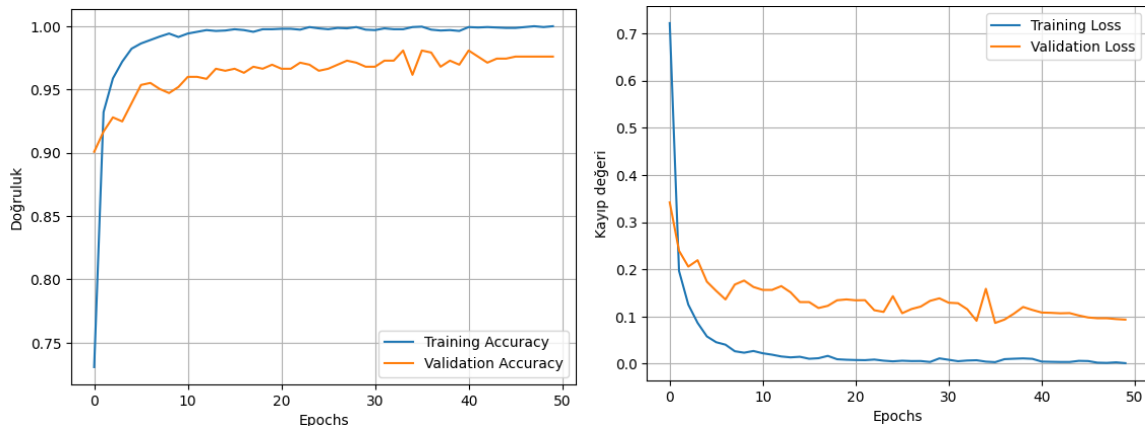
Sınıf Adı	Kesinlik	Duyarlılık	F1-Skor
askeri kamyon	0,9355	0,9775	0,9560
füze atar	0,9868	0,9375	0,9615
silahlı asker	0,9856	0,9928	0,9892
sivil	0,9896	0,9896	0,9896
sivil araç	0,9867	0,9737	0,9801
tank	0,9932	0,9932	0,9932

MobileNet mimarisi ile tüm sınıfların sınıflandırılmasında elde edilen doğruluk oranları Tablo 4.11'de verilmiştir. Tablo 4.11 incelendiğinde tüm sınıfların sınıflandırılması işleminde 0.9808'lik doğruluk oranı elde edilmiştir.

Tablo 4.11 MobileNet mimarisi doğruluk skoru

Veri Türü	Accuracy/Doğruluk
Train	0,9997
Test	0,9808

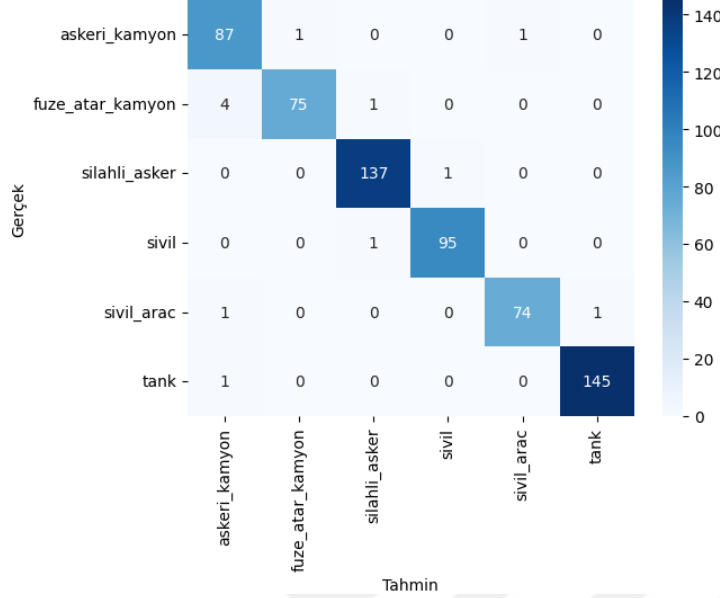
MobileNet mimarisi ile elde edilen doğruluk ve kayıp değerleri Şekil 4.8'de verilmiştir. Şekil 4.7 incelendiğinde en iyi doğruluk oranının 0,9808 ile, en düşük kayıp değeri olarak ise 0,0865 ile 36. epokta elde edildiği gözlemlenmiştir.



Şekil 4.7 MobileNet epok sayısına göre doğruluk ve kayıp değeri eğrileri

MobileNet mimarisinden elde edilen karışıklık matrisi Şekil 4.8'de verilmiştir. Matris incelendiğinde askeri kamyonlar sınıflandırılırken 87 tanesi doğru 2 tanesi yanlış, füze atar kamyonlar sınıflandırılırken 75 tanesi doğru 5 tanesi yanlış, silahlı askerler

sınıflandırılırken 137 tanesi doğru 1 tanesi yanlış, siviller sınıflandırılırken 95 tanesi doğru 1 tanesi yanlış, sivil araçlar sınıflandırılırken 74 tanesi doğru 2 tanesi yanlış, tanklar sınıflandırılırken 145 tanesi doğru 1 tanesi yanlış olarak sınıflandırılmıştır.



Şekil 4.8 MobileNet mimarisi karışıklık matrisi sonuçları

#### 4.1.5. EfficientNet Modeli ile Gerçekleştirilen Analiz Sonuçları

EfficientNet mimarisi ile elde edilen kesinlik, duyarlılık ve F1-skor değerleri Tablo 4.12’de verilmiştir. EfficientNet mimarisinin F1 skora göre performansı değerlendirildiğinde sivil araçlar en yüksek skor ile sınıflandırılırken füze atar askeri kamyonlar en düşük skorla sınıflandırılmıştır.

Tablo 4.12 EfficientNet mimarisinin sınıf bazında performansı

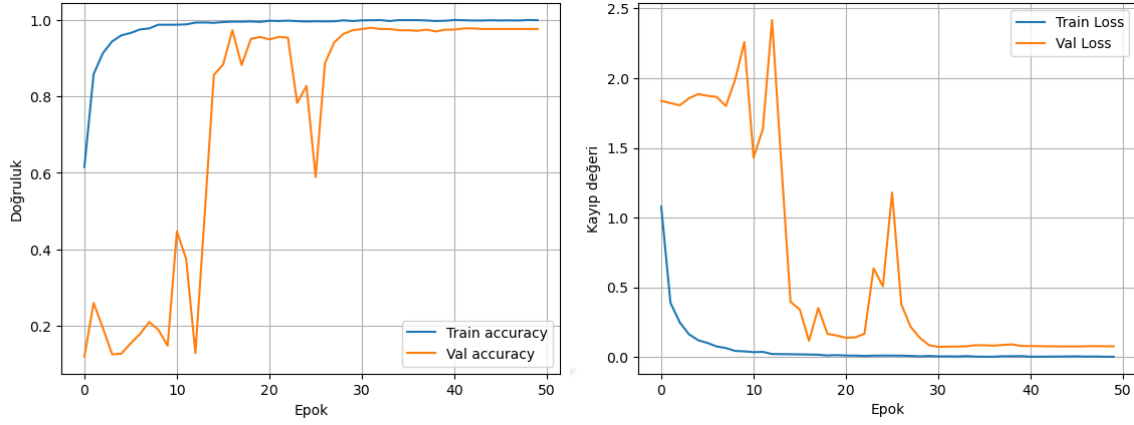
Sınıf Adı	Kesinlik	Duyarlılık	F1-Skor
askeri kamyon	0,9335	0,9775	0,9560
füze atar	0,9487	0,9250	0,9367
silahlı asker	0,9928	0,9928	0,9928
sivil	0,9896	0,9896	0,9896
sivil araç	0,9999	0,9868	0,9934
tank	0,9931	0,9863	0,9897

EfficientNet mimarisi ile tüm sınıfların sınıflandırılmasında elde edilen doğruluk oranları Tablo 4.13’de verilmiştir. Tablo 4.13 incelendiğinde tüm sınıfların sınıflandırılması işleminde 0.9792’lik doğruluk oranı elde edilmiştir.

Tablo 4.13 EfficientNet mimarisi doğruluk skoru

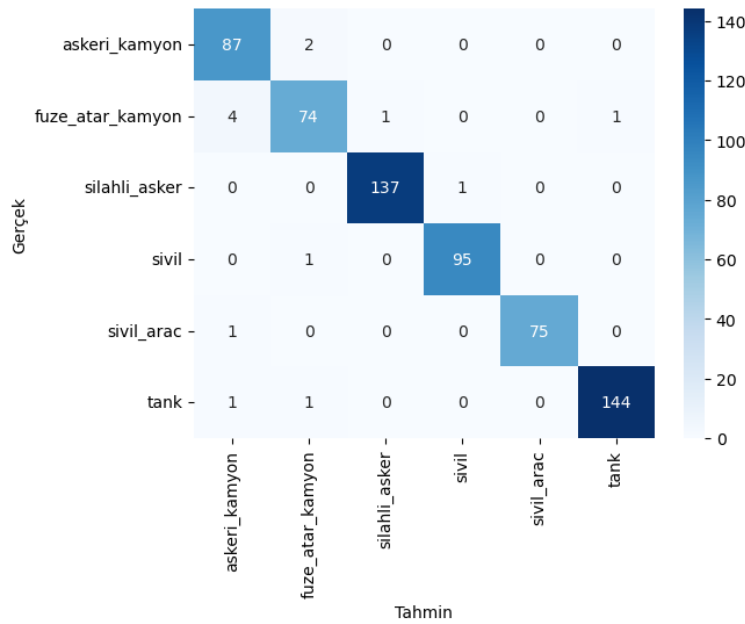
Veri Türü	Accuracy/Doğruluk
Train	0,9997
Test	0,9792

EfficientNet mimarisi ile elde edilen doğruluk ve kayıp değerleri Şekil 4.9’da verilmiştir. Şekil 4.9 incelendiğinde en iyi doğruluk oranının 0,9792 ile, en düşük kayıp değeri olarak ise 0,0762 ile 43. epokta elde edildiği gözlemlenmiştir.



Şekil 4.9 EfficientNet epok sayısına göre doğruluk ve kayıp değeri eğrileri

EfficientNet mimarisinden elde edilen karışıklık matrisi Şekil 4.10’da verilmiştir. Matris incelendiğinde askeri kamyonlar sınıflandırılırken 87 tanesi doğru 2 tanesi yanlış, füze atar kamyonlar sınıflandırılırken 74 tanesi doğru 6 tanesi yanlış, silahlı askerler sınıflandırılırken 137 tanesi doğru 1 tanesi yanlış, siviller sınıflandırılırken 95 tanesi doğru 1 tanesi yanlış, sivil araçlar sınıflandırılırken 75 tanesi doğru 1 tanesi yanlış, tanklar sınıflandırılırken 144 tanesi doğru 2 tanesi yanlış olarak sınıflandırılmıştır.



Şekil 4.10 EfficientNet mimarisi karışıklık matrisi sonuçları

#### 4.1.6. Modellerin Performans ve Özellik Karşılaştırması

Çalışmada 5 farklı transfer öğrenme mimarisi kullanılarak elde edilen sonuçlar Tablo 4.14'de verilmiştir. Tablo incelendiğinde en yüksek doğruluk 0,9824 ve en yüksek F1 skoru 0,9802 değerleri ResNet50 mimarisi ile elde edilmiştir. Mimarilerle yapılan eğitim sürelerine bakıldığında ise MobileNetV2 mimarisinin en hızlı sonuçlar ürettiği gözlemlenmiştir.

Tablo 4.14 Mimarilerin performans metrikleri

Model Adı	Kesinlik	Duyarlılık	F1-Skor	Doğruluk	Eğitim Süresi
<b>ResNet50</b>	<b>0,9813</b>	<b>0,9793</b>	<b>0,9802</b>	<b>0,9824</b>	14 s 12 dk 36 sn
InceptionV3	0,9766	0,9753	0,9759	0,9792	12 s 11 dk 24 sn
Xception	0,9812	0,9760	0,9785	0,9792	13 s 49 dk 11 sn
MobileNetV2	0,9796	0,9774	0,9783	0,9808	<b>10 s 52 dk 34 sn</b>
EfficientNetB3	0,9766	0,9763	0,9764	0,9792	13 s 19 dk 48 sn

Eğitilen modellerin sınıflara ait performansları Tablo 4.15'da verilmiştir. Tablodaki doğru sayısı D ile yanlış sayısı ise Y ile gösterilmiştir.

Tablo 4.15 Mimarilerin sınıf bazındaki performansları

Model adı	Veri Setindeki Sınıflar											
	Askeri Kamyon		Füze atar Kamyon		Silahlı Asker		Sivil		Sivil Araç		Tank	
	D	Y	D	Y	D	Y	D	Y	D	Y	D	Y
ResNet50	88	1	77	3	137	1	92	4	74	2	146	-
InceptionV3	84	5	77	3	137	1	94	2	74	2	146	-
Xception	85	4	76	4	137	1	93	3	76	-	145	1
MobileNetV2	87	2	75	5	137	1	95	1	74	2	145	1
EfficientNet	87	2	74	6	137	1	95	1	75	1	144	2

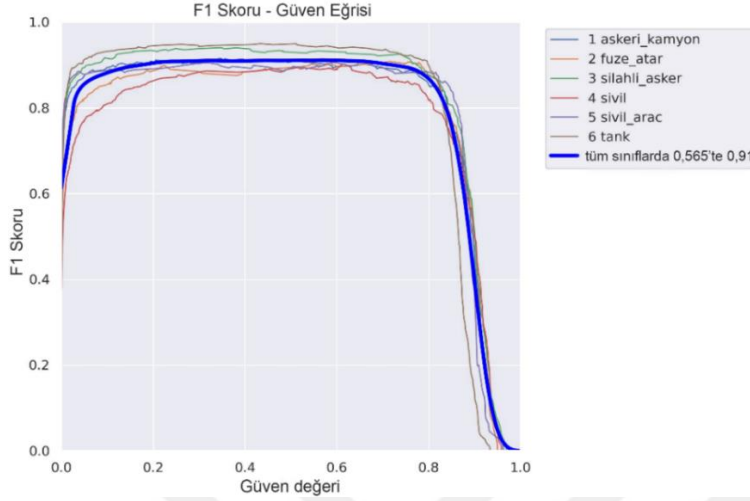
#### 4.2. Nesne Tespit Algoritması YOLO ile Gerçekleştirilen Analiz Sonuçları

Çalışmanın bu aşamasında ise YOLO'nun en son versiyonlarından YOLOv8 ve YOLO-NAS mimarileri kullanılmış, mimarilerin performansı mAP50 ve F1 skoru baz alınarak değerlendirilmiştir. YOLOv8 mimarisi ile eğitilen modelden elde edilen skorlar Tablo 4.16'da verilmiştir. Veri setindeki sınıflara ait nesnelerin tespit edilmesi işleminde F1 skoru olarak 0,9116, mAP50 skoru olarak ise 0,961 elde edilmiştir. Çalışmada elde edilen yüksek F1 skoru ve mAp50 skorları göstermektedir ki eğitilen model çalışmadaki nesnelere yüksek bir oranla tespit edip sınıflandırabilmektedir.

Tablo 4.16 YOLOv8 performans skorları

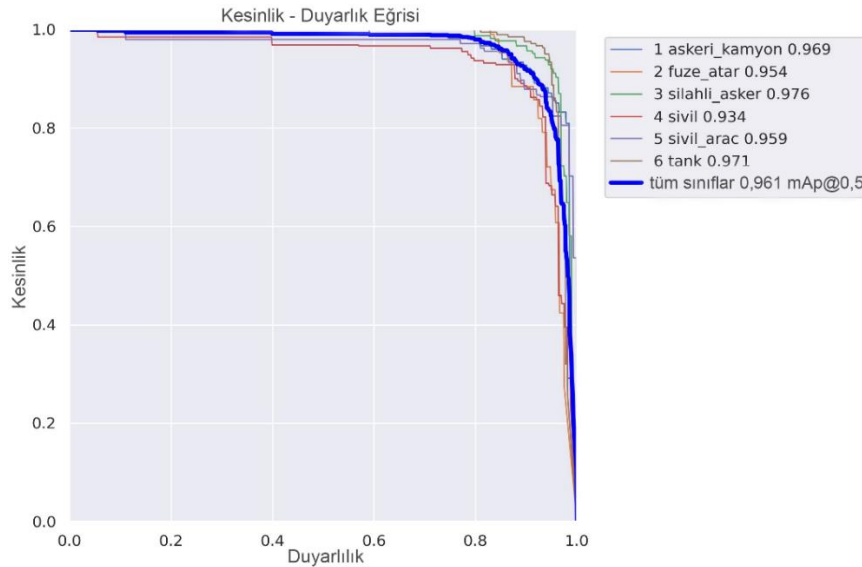
Model Adı	Kesinlik	Duyarlılık	F1-Skor	mAP50	Eğitim Süresi
YOLOv8	0,961	0,964	0,9116	0,961	6 s 04 dk 16 sn

Eđitim sonucu yapılan deęerlendirme sonucunda elde edilen F1 skoru gven eęrisi Őekil 4.11’de verilmiřtir. Tm sınıflara ait F1 skoru olarak 0,91 deęeri elde edilirken F1 skorunun en iyi olduęu gven deęerinin ise 0,565 deęerinde olduęu grlmektedir.



Őekil 4.11 F1 skoru gven eęrisi

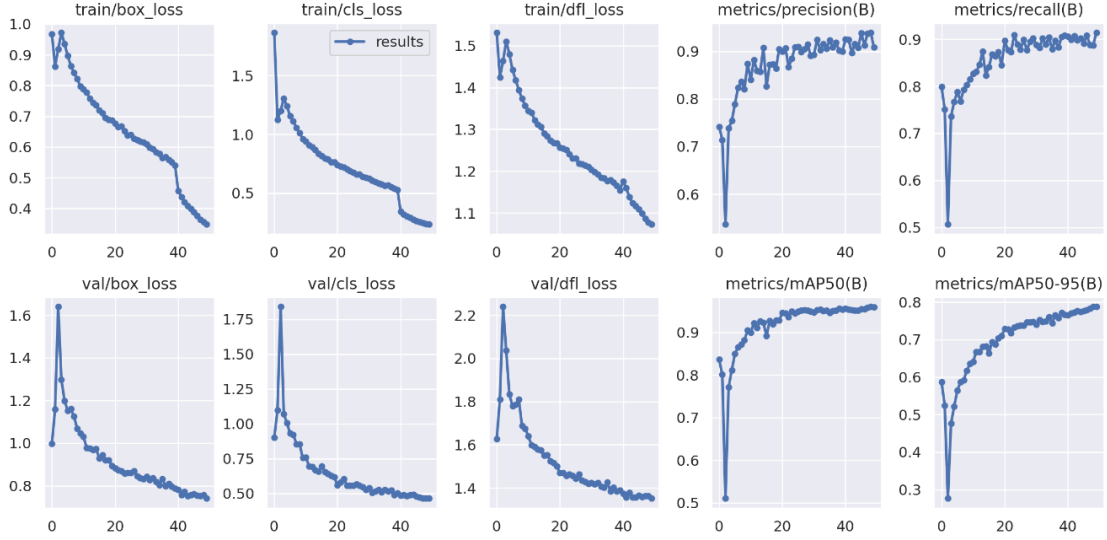
Őekil 4.12’de verilen kesinlik duyarlılık eęrisi incelendięinde en yksek kesinlik deęerinin silahlı asker sınıfında, en dřk kesinlik deęerinin ise sivil sınıfında elde edildięi gzlemlenmektedir. Eđitilen model btn sınıfları sınıflandırırken 0,961 kesinlik deęeri elde etmiřtir. Model askeri kamyonları 0,969, fze atar kamyonları 0,954, silahlı askerleri 0,976, sivilleri 0,934, sivil araları 0,959, tankları ise 0,971 kesinlik deęeri ile tespit etmiřtir.



Őekil 4.12 Duyarlılık kesinlik eęrisi

Modelin eđitim sreci ncesi girilen epok sayısına gre elde edilen performans metriklerini gsteren grafikler Őekil 4.13’de verilmiřtir. En iyi performanslar 48. epok sayısında elde

edilmiştir. YOLOv8 ile en iyi map50 değeri olarak 0,9605, en iyi kesinlik değeri olarak 0,94, en iyi duyarlılık değeri olarak 0,9142, en iyi mAP50-95 değeri olarak 0,787 elde edilmiştir.



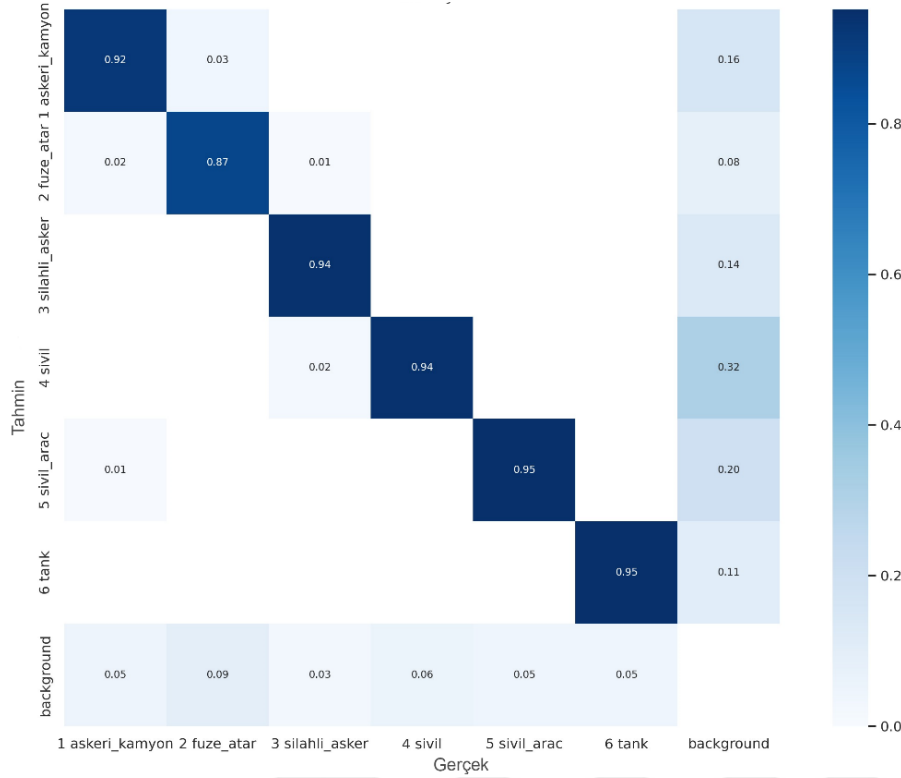
Şekil 4.13 Epok sayısına göre elde edilen performans metrikleri

Eğitilen model ile görüntüler üzerinde yapılan nesne tespit işleminin testlerinden elde edilen örnek görüntüler Şekil 4.14’de verilmiştir.



Şekil 4.14 Görüntüler üzerinde yapılan nesne tespit işlemi örnekleri

Eğitim sonrası elde edilen Şekil 4.15’deki karışıklık matrisi incelendiğinde görüntü verileri üzerindeki nesnelerin tespit edilmesinde en iyi performans tank sınıfında elde edilmiştir. Model tankları tespit ederken herhangi bir hata yapmamış ve %95 doğruluk oranı elde etmiştir. Sivil araçların tespit edilirken %95 doğruluk oranı alınmış sivil araçların %1’lik kısmı askeri kamyon olarak yanlış sınıflandırılmıştır. En düşük performans ise %87 oranla füze atar kamyon sınıfına ait görüntülerden elde edilmiştir. Model füze atar kamyonların %2’sini askeri kamyon, %1’ini ise silahlı asker olarak sınıflandırmıştır. Yapılan yanlışların veri setindeki füze atar kamyon görüntülerinde diğer sınıflara ait görüntülerin olmasından kaynaklandığı düşünülmektedir.



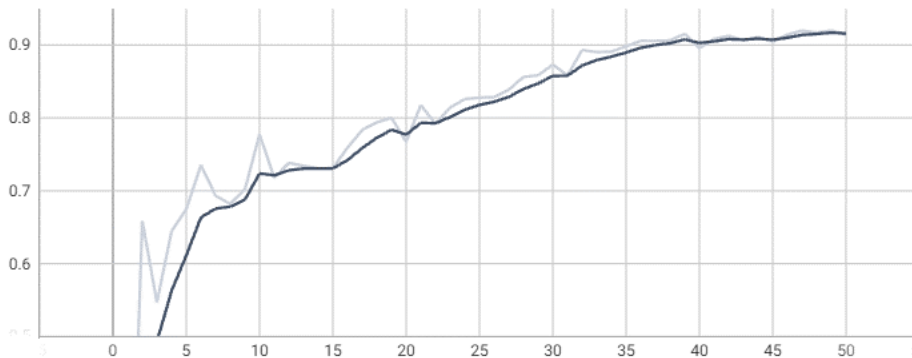
Şekil 4.15 YOLOv8 karışıklık matrisi sonuçları

YOLO-NAS-M mimarisi ile eğitilen modelden elde edilen skorlar Tablo 4.17’de verilmiştir. Veri setindeki sınıflara ait nesnelerin tespit edilmesi işleminde F1 skoru olarak 0,95, mAP50 skoru olarak ise 0,92 elde edilmiştir.

Tablo 4.17 YOLO-NAS mimarisi performans skorları

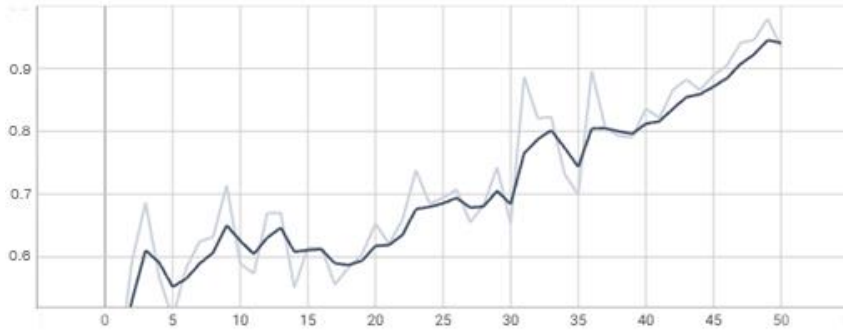
Model Adı	Kesinlik	Duyarlılık	F1-Skor	mAP50	Eğitim Süresi
YOLO-NAS-M	0,97	0,96	0,95	0,92	5 s 52 dk 46 sn

Modelin eğitim süreci öncesi girilen epok sayısına göre elde edilen mAp50 skorunun değişimi grafiği Şekil 4.16’da verilmiştir. En iyi performanslar 49. epok sayısında elde edilmiştir.



Şekil 4.16. YOLO-NAS-M mimarisi mAP50 skor grafiği

Modelin eğitim süreci öncesi girilen epok sayısına göre elde edilen F1 skorunun değişimi grafiği Şekil 4.17’de verilmiştir. En iyi performans 49. epok sayısında 0,95 olarak elde edilmiştir.



Şekil 4.17. YOLO-NAS-M F1-Skor grafiği

YOLOv8 ve YOLO-NAS mimarilerinden elde edilen mAP50 parametreleri (Tablo 4.15 ve Tablo 4.16) karşılaştırıldığından en yüksek skor 0,961 ile YOLOv8 mimarisinden elde edilmiştir. İHA ile nesne tespit işlemleri için kullanılacak model, YOLOv8 mimarisi kullanılarak eğitilen modelden seçilmiştir.

### 4.3 Gerçek Zamanlı Nesne Tespiti ve Sınıflandırma Performans Karşılaştırması

Çalışmada kullanılan transfer öğrenme mimarilerinden en başarılı olan ResNet50 mimarisi seçilerek YOLOv8 nesne tespit mimari ile gerçek zamanlı çalışma performansları karşılaştırılmıştır. ResNet50 ve YOLOv8 mimarilerinin performans skorları Tablo 4.18’de verilmiştir. Eğitilen modellerin en iyi ağırlıkları ve parametrelerini içeren dosyaları ile gerçek zamanlı nesne tespiti ve sınıflandırma işlemleri yapılmıştır. Yapılan testlerde YOLOv8 modeli daha az donanım gerektirmesinden dolayı İHA ile yapılacak nesne tespit işlemlerinde kullanılmıştır (Resim 4.4).

Tablo 4.18 ResNet50 ile YOLOv8 performans metrikleri

Model Adı	Kesinlik	Duyarlılık	F1-Skor	Doğruluk	Eğitim Süresi
ResNet50	0,9813	0,9793	0,9802	0,9824	14 s 12 dk 36 sn
YOLOv8	0,961	0,964	0,9116	0,961	6 s 04 dk 16 sn



Resim 4.4. İHA ile gerçek zamanlı yapılan tespitler

## 5. SONUÇ VE ÖNERİLER

Yeni gelişen teknolojiler sayesinde savunma teknolojilerine olan ihtiyaç her geçen gün artmaktadır. Ülkeler savunma sistemlerini güçlendirmek için yeni teknolojiler kullanmakta ve yeni teknolojiler üzerinde araştırma yapmaktadırlar. Savunma teknolojilerinde akıllı ve otonom sistemlerin kullanılma sayısı gün geçtikçe artmaktadır. İstihbarat sağlama görevlerinde İHA'lar sıklıkla kullanılmaktadır. Bu çalışmada derin öğrenme yöntemleri kullanılarak eğitilen model ile askeri araçları, silahlı-silahsız kişileri havadan tespit edebilen, otonom uçuşlar yapabilen bir İHA tasarlanmıştır. Tasarlanan akıllı, otonom İHA ile istihbarat sağlama görevlerinde alana katkı sağlanacağı düşünülmektedir. Çalışmada nesnelere sınıflandırmak için transfer öğrenme mimarilerinden ResNet50, InceptionV3, Xception, Mobilenet, EfficientNet gibi 5 farklı transfer öğrenme mimarisi ile çalışılmış olup performans karşılaştırmaları yapılmıştır. Ayrıca nesne tespit mimarilerinden YOLO8 ve YOLO-NAS mimarileri kullanılarak veri setindeki görüntülerden nesne tespit işlemi yapılmıştır. Mimariler için oluşturulan yeni ve hibrit veri setine Kaggle platformundaki "Ukrainian War Images", "detection of military equipment", "Military vehicles", "Military/Civilian Vehicles Image Classification", "Aiming soldiers image dataset", "Guns Object Detection", "Normal vs Military Vehicles" gibi 7 farklı veri setindeki askeri araçlar, sivil insan, silahlı insan ve sivil araç görüntülerinden uygun olanlar eklenmiştir. Ayrıca bu görüntülere ek olarak telif hakkı olmayan askeri ve sivil araçların görüntüleri ve silahlı ve silahsız insanların görüntüleri de eklenerek 4195 adet görsel içeren 6 sınıflı yeni bir hibrit veri seti hazırlanmıştır. Modellerin aşırı öğrenme yapmaması için veri setindeki görüntülere rastgele yaklaştırma, yatay yönde rastgele kaydırma, dikey yönde rastgele kaydırma, yatay döndürme, belirli açıyla döndürme ve yatay kayma işlemleri uygulanmıştır. Sınıflandırma ve nesne tespit mimarileri için objektif bir değerlendirme yapabilmek için her bir modele görüntü boyutları (genişlik x yükseklik) hariç aynı parametreler girilmiştir. Transfer öğrenme mimarilerinin performansları kayıp değeri, doğruluk oranı ve F1-skoru baz alınarak değerlendirilmiştir. Nesne tespit mimarisinin performansı mAp50, F1-skoru baz alınarak değerlendirilmiştir. Çalışmada transfer öğrenme mimarilerindeki en başarılı sonuçlar 0,9824 doğruluk, 0,9802 F1-skoru değeri ile ResNet50 mimarisi ile elde edilmiştir. YOLOv8 nesne tespit algoritması ile 0,961 mAP50, 0,9116 F1-skoru elde edilmiştir. Çalışmada kullanılan transfer öğrenme mimarilerinden en başarılı olan ResNet50 mimarisi ile YOLOv8 mimarisinin ağırlıkları ve seçilen en iyi parametrelerinden oluşturulan model dosyaları İHA üzerinde gerçek zamanlı olarak denenmiştir. Havadan alınan görüntülerdeki nesnelere küçük boyutta olmasından dolayı

sınıflandırma tabanlı eğitilen modeller nesnelere gerçek zamanlı sınıflandırırken hata yapmıştır. Nesne tespit algoritmaları ile eğitilen modeller ise gerçek zamanlı test çalışmalarında nesnelere başarılı bir şekilde tespit edip sınıflandırabilmiştir. Bundan dolayı İHA üzerine YOLO-NAS'a göre daha iyi sonuç veren YOLOv8 ile eğitilen model eklenmiştir.

Veri setine havadan görüntüleri çekilmiş daha fazla askeri araç görüntüsü eklenerek veri çoğaltılabilir. Çalışma askeri gizliliğinden dolayı çalışmada kullanılan askeri araç görüntüleri ile sınırlıdır. Bu çalışmada kullanılan veri setindeki her bir askeri araç sınıfı marka modeline göre ayrıştırılmamıştır. Veri setindeki her bir sınıf kendi arasında özelleştirilerek farklı sınıflara ayrılabilir. Örneğin tankları modellerine göre sınıflandırabilen bir çalışma yapılabilir. Eğitim öncesi girilen parametreler hiperparametre optimizasyonu kullanılarak belirlenebilir. İHA'nın olumsuz hava şartlarından etkilenmemesi için gövdesi enjeksiyon kalıp makinesi ile yapılmış koruyucu plastik malzeme ile kapatılabilir.

## KAYNAKLAR

- Akar, M. (2022). *COVID-19'a karşı insansız hava aracı ile derin öğrenme tabanlı maske tespiti* (Master's thesis, Karamanoğlu Mehmetbey Üniversitesi).
- Akdağ, A. (2017). *Derin öğrenme algoritmaları kullanılarak gerçek zamanlı silah tanıma uygulaması* (Doctoral dissertation, Necmettin Erbakan University (Turkey)).
- Aksoy, İ., & Adem, K. (2023). *Görüntü sınıflandırması için evrimsel sinir ağlarında parametrelerin sınıflandırma performansına etkisi* (yüksek lisans tezi). YÖK tez merkezinden edinilmiştir (826574).
- Aktaş, M. (2022). Yapay zekâ ve mesleklerin geleceği artificial intelligence and the future of professions. *Proceeding book*, 50.
- Aktimur, B., & Gökpınar, E. (2015). Katmanlı üretimin havacılıktaki uygulamaları. *Gazi University Journal of Science Part C: Design and Technology*, 3(2), 463-469.
- Albayrak, E. (2021). *Derin öğrenme ile İHA görüntülerinden nesne tespitinin yapılması* (Master's thesis, Bilecik Şeyh Edebali Üniversitesi, Fen Bilimleri Enstitüsü).
- Altaş, Z., Özgüven, M. M., & Adem, K. (2023). Bazı Bağ Hastalıklarının Faster R-CNN Modeli ile Otomatik Tespit Edilmesi ve Sınıflandırılması. *Turkish Journal of Agriculture-Food Science and Technology*, 11(1), 97-103.
- Arserim, M. A., & Ayşan, U. (2023). İnsansız hava aracından çekilen videolar kullanılarak derin öğrenme yaklaşımı ile nesne tespiti. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 14(1), 9-15.
- Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*, 61, 101182.
- Basha, S. S., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378, 112-119.
- Bayram, A. F., & Nabiyeve, V. (2023). Derin öğrenme tabanlı saklanan kamufle tankların tespiti: son teknoloji YOLO ağlarının karşılaştırmalı analizi. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, 13(4), 1082-1093.
- Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). Cambridge, MA, USA: MIT press.
- Beşdok, E., Özçelik, A. E., Jeodezi, E. Ü. M. F., & Kayseri, F. M. B. (2009). Kalman Filtreleme Yöntemiyle Otonom Hava Araç Navigasyonunda GPS/INS Entegrasyonu. *TMMOB Harita ve Kadastro Mühendisleri Odası, 12. Türkiye Harita Bilimsel ve Teknik Kurultayı, 11-15 Mayıs 2009, Ankara*.
- Burgaz, M. (2020). *Derin öğrenme algoritmaları kullanarak insansız hava araçları ile silah tespiti* (Master's thesis, Batman Üniversitesi Fen Bilimleri Enstitüsü).
- Byrd, J., & Lipton, Z. (2019). What is the effect of importance weighting in deep learning?. In *International conference on machine learning* (pp. 872-881). PMLR.
- Camgözlü, Y. & Kutlu, Y. (2019). Analysis of Pooling Effect on CNN using Leaf Database. *Natural and Engineering Sciences*, 4(3), 118 – 123.

- Cherkassky, V., Friedman, J. H., & Wechsler, H. (Eds.). (2012). *From statistics to neural networks: theory and pattern recognition applications* (Vol. 136). Springer Science & Business Media.
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).
- Choutri, K., Lagha, M., Meshoul, S., Batouche, M., Bouzidi, F., & Charef, W. (2023). Fire Detection and Geo-Localization Using UAV's Aerial Images and Yolo-Based Models. *Applied Sciences*, 13(20), 11548.
- Clarke, R. (2014). Understanding the drone epidemic. *Computer Law & Security Review*, 30(3), 230-246.
- Cohen, I. L. (1994). An artificial neural network analogue of learning in autism. *Biological psychiatry*, 36(1), 5-20.
- Cömert, O., Hekim, M., & Kemal, A. (2019). Faster r-cnn kullanarak elmalarda çürük tespiti. *International Journal of Engineering Research and Development*, 11(1), 335-341.
- Cunta, A. (2015). *Novel quadcopter flight controller and telemetry remote* (Doctoral dissertation, Murdoch University).
- Çalışkan, D., & Demir, Ö. (2022). Derin Öğrenme Yöntemleri ile Şüpheli Davranış Tespiti. *International Periodical Of Recent Technologies In Applied Engineering*, 3(1), 28-43.
- Deng, L. & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends® in signal processing*, 7(3-4), 197-387.
- De Yong, S., Kusumarini, Y., & Tedjokoemo, P. E. D. (2020). Interior design students' perception for AutoCAD, SketchUp and Rhinoceros software usability. In *IOP*
- Dikbayır, H. S., & Bülbül, H. İ. (2020). Derin Öğrenme Yöntemleri Kullanarak Gerçek Zamanlı Araç Tespiti. *TÜBAV Bilim Dergisi*, 13(3), 1-14.
- Doğan, F., & Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.
- Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020). MobileNetV2 model for image classification. In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)* (pp. 476-480). IEEE.
- Du, J. (2018). Understanding of object detection based on CNN family and YOLO. In *Journal of Physics: Conference Series* (Vol. 1004, p. 012029). IOP Publishing.
- Eldem, H. (2023). *Derin öğrenme yaklaşımlarıyla tıbbi yara görüntülerinin segmentasyonu ve sınıflandırılması*. (Doktora tezi), YÖK tez merkezinden edinilmiştir (803647).
- Engin, M. T., & Adem, K. (2023). *Derin öğrenme yöntemleri kullanılarak deri lezyon görüntülerinden maymun çiçeği hastalığının tespiti* (yüksek lisans tezi). YÖK tez merkezinden edinilmiştir (826557).
- Ersoy, E., & Karal, Ö. (2012). Yapay sinir ağları ve insan beyni. *İnsan ve Toplum Bilimleri Araştırmaları Dergisi*, 1(2), 188-205.

- Feng, Y., & Li, Y. (2018). An overview of deep learning optimization methods and learning rate attenuation methods. *Hans J. Data Mining*, 8(4), 186-200.
- Flach, P., & Kull, M. (2015). Precision-recall-gain curves: PR analysis done right. *Advances in Neural Information Processing Systems*, 28(1), 838-846.
- Godbole, S., & Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 22-30). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Han, J., & Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks* (pp. 195-201). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Hanin, B. (2019). Universal function approximation by deep neural nets with bounded width and relu activations. *Mathematics*, 7(10), 992.
- He KM, Zhang XY, Ren SQ, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778.
- Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., ... & Leskovec, J. (2020). Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33, 22118-22133.
- Huang, R., Pedoeem, J., & Chen, C. (2018). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In *2018 IEEE international conference on big data (big data)* (pp. 2503-2510). IEEE.
- Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, 11(7), 677.
- Ismail, I. (2020). Komparasi akurasi global position system (gps) receiver u-blox neo-6m dan u-blox neo-m8n pada navigasi quadcopter. *Elektron: Jurnal Ilmiah*, 12-15.
- Jialin Pan, S. and Yang, Q. (2010). "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359.
- Karaağaç, C. (2014). Geleceğin Harekât Ortamında İHA Sistemleri: Askeri Uygulamalar & Teknoloji Gereksinimleri. III. Ulusal Havacılıkta İleri Teknolojiler Konferansı, İstanbul.
- Kassani, S. H., Kassani, P. H., Khazaeinezhad, R., Wesolowski, M. J., Schneider, K. A., & Deters, R. (2019). Diabetic retinopathy classification using a modified xception architecture. *IEEE international symposium on signal processing and information technology (ISSPIT)* (pp. 1-6). IEEE.
- Kawasaki, H., Iizuka H., Okamoto S., Ando H., and Maeda T. (2010). Collaboration and skill transmission by first-person perspective view sharing system. In *19th International Symposium in Robot and Human Interactive Communication*, 125–131.
- Keskar, N. S., & Socher, R. (2017). Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*.
- Keskenler, M. F., & Keskenler, E. F. (2017). Geçmişten günümüze yapay sinir ağları ve tarihçesi. *Takvim-i Vekayi*, 5(2), 8-18.

- Korchenko, A. G., & Illyash, O. S. (2013). The generalized classification of unmanned air vehicles. In *2013 IEEE 2nd International Conference Actual Problems of Unmanned Air Vehicles Developments Proceedings (APUAVD)* (pp. 28-34). IEEE.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Kuş, Z. (2019). “Mikrokanonikal Optimizasyon Algoritması ile Konvolüsyonel Sinir Ağlarında Hiper Parametrelerin Optimize Edilmesi.” Fatih Sultan Mehmet Vakıf Üniversitesi, Lisansüstü Eğitim Enstitüsü, 2019.
- Lee, W., Kim, S., Lee, Y.T., Lee, H.W. and Choi, M. (2017). “Deep neural networks for wild fire detection with unmanned aerial vehicle,” In *2017 IEEE International Conference on Consumer Electronics*, 2017, pp. 252-253.
- Lei, X., Pan, H., & Huang, X. (2019). A dilated CNN model for image classification. *IEEE Access*, 7, 124087-124095.
- Lian, S., Meng, W., Lin, Z., Shao, K., Zheng, J., Li, H., & Lu, R. (2021). Adaptive attitude control of a quadrotor using fast nonsingular terminal sliding mode. *IEEE Transactions on Industrial Electronics*, 69(2), 1597-1607.
- Lin, M., Chen, Q., Yan, S. (2014). Network in network. In: *Proceedings of ICLR* (2014).
- Liu, S., & Liu, Z. (2017). Multi-channel CNN-based object detection for enhanced situation awareness. *arXiv preprint arXiv:1712.00075*.
- Liu, C., Tao, Y., Liang, J., Li, K., & Chen, Y. (2018). Object detection based on YOLO network. In *2018 IEEE 4th information technology and mechatronics engineering conference (ITOEC)* (pp. 799-803). IEEE.
- Luukkonen, T. (2011). Modelling and control of quadcopter. *Independent research project in applied mathematics, Espoo*, 22(22).
- Luo, Y., Li, Y., Lu, Y., Lin, S., & Liu, X. (2018). The prediction of hypertension based on convolution neural network. In *2018 IEEE 4th International Conference on Computer and Communications (ICCC)* (pp. 2122-2127). IEEE.
- Masters, D., & Luschi, C. (2018). Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*.
- Mitra, S., & Land, B. (2013). Autonomous quadcopter docking system. *Project report, Cornell University, USA*.
- Mogensen, K. N. (2016). Motor-control considerations for electronic speed control in drones. *Analog Applications Journal [online]. Texas Instruments*.
- Nandimandalam, V. D. (2020). *Military and Non-Military Vehicle Detection by Faster R-CNN and SSD300 Models using Transfer Learning* (Doctoral dissertation, Dublin, National College of Ireland).
- Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*. Springer Science & Business Media.
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

- Ostojčić, G., Stankovski, S., Tejić, B., Đukić, N., & Tegeltija, S. (2015). Design, control and application of quadcopter. *International Journal of Industrial Engineering and Management*, 6(1), 43.
- Özguven, M. M., & Adem, K. (2019). Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms. *Physica A: statistical mechanics and its applications*, 535, 122537.
- Özkan, İ., & Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.
- Öztemel, E. (2003). Yapay Sinir Ağları, sf 29, Papatya Yayıncılık, İstanbul.
- Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., & Nielsen, M. (2013). Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In *International conference on medical image computing and computer-assisted intervention* (pp. 246-253). Berlin, Heidelberg.
- Praveen, V., & Pillai, S. (2016). Modeling and simulation of quadcopter using PID controller. *International Journal of Control Theory and Applications*, 9(15), 7151-7158.
- Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.
- Ramirez-Atencia, C., & Camacho, D. (2018). Extending QGroundControl for automated mission planning of UAVs. *Sensors*, 18(7), 2339.
- Reznikov, N., Buss, D. J., Provencher, B., McKee, M. D., & Piché, N. (2020). Deep learning for 3D imaging and image analysis in biomineralization research. *Journal of Structural Biology*, 212(1), 107598.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).
- Saygılı, Z., & Özmen, G. (2022). İnsansız Hava Araçlarında Gömülü Sistem Üzerinden Derin Öğrenme ile Nesne Tespiti. *Avrupa Bilim ve Teknoloji Dergisi*, (34), 292-298.
- Schütt, K. T., Sauceda, H. E., Kindermans, P. J., Tkatchenko, A., & Müller, K. R. (2018). SchNet—a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24).
- Sharma, V. A., & Rajesh, M. (2018). Building a quadcopter: An approach for an Autonomous Quadcopter. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1252-1258). IEEE.
- Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12), 310-316.
- Shorten, C. ve Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning, *Journal of big data*, 6 (1), 1-48.
- Sirisha, U., Praveen, S. P., Srinivasu, P. N., Barsocchi, P., & Bhoi, A. K. (2023). Statistical analysis of design aspects of various YOLO-based deep learning models for object detection. *International Journal of Computational Intelligence Systems*, 16(1), 126.

- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).
- Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
- Talaat, F. M., & ZainEldin, H. (2023). An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Computing and Applications*, 35(28), 20939-20954.
- Tamang, S., Sen, B., Pradhan, A., Sharma, K., & Singh, V. K. (2023). Enhancing covid-19 safety: Exploring yolov8 object detection for accurate face mask classification. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2), 892-897.
- Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR.
- Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716.
- Türkarslan, K., & Hardalac, F. (2022). Derin Öğrenme Yöntemleri Kullanılarak Havadan Elde Edilen Görüntüler Üzerinde Nesne Tespiti. *El-Cezeri*, 9(4), 1398-1410.
- Urbahs, A., & Jonaite, I. (2013). Features of the use of unmanned aerial vehicles for agriculture applications. *Aviation*, 17(4), 170-175.
- Umut, K., Yılmaz, A., & Dikmen, Y. (2019). Sağlık alanında kullanılan derin öğrenme yöntemleri. *Avrupa Bilim ve Teknoloji Dergisi*, (16), 792-808.
- Wang, X., Cheng, P., Liu, X., Uzochukwu, B. (2018). Fast and accurate, convolutional neural network based approach for object detection from UAV, 44th annual conference of the IEEE industrial electronics society.
- Wechsler H., (1992). Neural Networks for Perception Theory of the Backpropagation Neural Network Based on “nonindent” by Robert Hecht-Nielsen. Academic Press, 65-92.
- Wen, L., Li, X., & Gao, L. (2020). A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Computing and Applications*, 32, 6111-6124.
- Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23), 495.
- Xia, X., Xu, C., & Nan, B. (2017). Inception-v3 for flower classification. In *2017 2nd international conference on image, vision and computing (ICIVC)* (pp. 783-787). IEEE.
- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9, 611-629.
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295-316.
- Yıldız, T. (2017). Yaklaşan dördüncü endüstri devrimi ve Türkiye’deki mevcut durum. *İzmir İleri Teknoloji Enstitüsü*, 1-18.

- Yohanandan, S., Song, A., Dyer, A. G., & Tao, D. (2018). Saliency preservation in low-resolution grayscale images. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 235-251).
- Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*.
- ZahediNasab, R. & Mohseni, H. (2020). "Neuroevolutionary based convolutional neural network with adaptive activation functions," *Neurocomputing*, vol. 381, pp. 306–313.
- Zahisham, Z., Lee, C. P., & Lim, K. M. (2020). Food recognition with ResNet-50. In *2020 IEEE 2nd international conference on artificial intelligence in engineering and technology (IICAJET)* (pp. 1-5). IEEE.
- Zhang, M., Cui, Z., Neumann, M., & Chen, Y. (2018). An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).
- Zhou, Y., Chen, S., Wang, Y., & Huan, W. (2020). Review of research on lightweight convolutional neural networks. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)* (pp. 1713-1720). IEEE.
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1), 43-76.

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : E\*\*\*\*, D\*\*\*\*  
Uyruğu : T.C.  
Doğum tarihi ve yeri :  
Medeni hali :  
Telefon : 0  
e-mail :



### Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	SBTÜ/ Savunma Teknolojileri	Devam ediyor
Lisans	Balıkesir Üniversitesi / BÖTE	2008
Lise	Sivas Atatürk Lisesi	2001

### İş Deneyimi

Yıl	Yer	Görev
2016-Halen	Sivas Bilim ve Sanat Merkezi	Bilişim Teknolojileri Öğretmeni
2011-2015	Sinop Ayancık Yenikonak YBO	Bilişim Teknolojileri Öğretmeni
2009-2011	Van Özalp Atatürk Ortaokulu	Bilişim Teknolojileri Öğretmeni

### Yabancı Dil

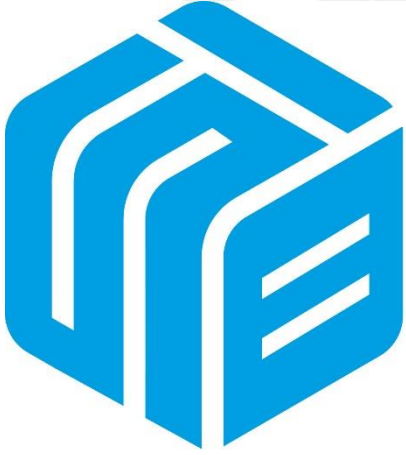
İngilizce

### Yayınlar

- Erol, D., & Bakır, H. (2023). Hiperparametreleri Ayarlanmış Makine Öğrenmesi Yöntemleri Kullanılarak Ağdaki Saldırıların Tespiti. In *International Conference on Pioneer and Innovative Studies* (Vol. 1, pp. 274-286).
- Erol, D., & Adem, K. (2023). Classification of Military Vehicles and Weapons using Deep Learning Architectures. *AS-Proceedings*, 1(7), 10-15.

### Hobiler

Elektro Gitar, Bateria ve Bas Gitar Çalma, Balık tutma, İHA pilotluğu



**SİVAS  
BİLİM VE TEKNOLOJİ  
ÜNİVERSİTESİ**

*KÖKLERDEN GÖKLERE...*