



**UYDU GÖRÜNTÜLERİNDEN UÇAK TESPİTİ**

**Hussein Mahmood Abdo MOHAMMED**

**Yüksek Lisans Tezi  
Elektrik-Elektronik Mühendisliği Anabilim Dalı  
Elektrik Bilim Dalı  
Dr. Öğr. Üyesi Emin Argun ORAL  
2019  
Her hakkı saklıdır**

**ATATÜRK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**UYDU GÖRÜNTÜLERİNDEN UÇAK TESPİTİ**

**Hussein Mahmood Abdo MOHAMMED**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI  
Elektronik Bilim Dalı**

**ERZURUM  
2019**

**Her hakkı saklıdır**



T.C.  
ATATÜRK ÜNİVERSİTESİ  
Fen Bilimleri Enstitüsü Müdürlüğü  
TEZ ONAY FORMU



UYDU GÖRÜNTÜLERİNDEN UÇAK TESPİTİ

Dr. Öğr. Üyesi Emin Argun ORAL danışmanlığında, Hussein Mahmood Abdo Mohammad tarafından hazırlanan bu çalışma, 17/07/2019 tarihinde aşağıdaki jüri tarafından Elektrik Elektronik Mühendisliği Anabilim Dalı Elektronik Bilim Dalı'nda Yüksek Lisans tezi olarak oybirliği ile kabul edilmiştir.

Başkan: Doç. Dr. İ. Yücel ÖZBEK

İmza :

Üye : Doç. Dr. Ahmet DURLU

İmza :

Üye : Dr. Öğr. Üyesi Emin Argun ORAL

İmza :

Yukarıdaki sonuç;

Enstitü Yönetim Kurulu'nun 18.07/2019 tarih ve 29.../88..... nolu kararı ile onaylanmıştır.

Prof. Dr. Mehmet KARAKAN  
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

## ÖZET

Yüksek Lisans Tezi

# UYDU GÖRÜNTÜLERİNDEN UÇAK TESPİTİ

Hussein Mahmood Abdo MOHAMMED

Atatürk Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektrik-Elektronik Mühendisliği Anabilim Dalı  
Elektronik Bilim Dalı

Danışman: Dr. Öğr. Üyesi Emin Argun ORAL

Uydu görüntüsü analizi, görüntü işleme alanındaki sivil ve askeri uygulamalar için birçok araştırma çalışmasının yapıldığı bir araştırma alanıdır. Uydu görüntüleri; bölgelerin, binaların, yolların, hava araçlarının ve diğer insan yapımı nesnelerin tanınması, tespit edilmesi ve sınıflandırılması dahil olmak üzere birçok uygulamaya sahiptir. Bunlar arasında, özellikle uçak tespiti askeri uygulamalar için stratejik olarak önemlidir ve bu tez çalışmasının temelini oluşturur. Çalışmanın ilk aşamasında, bu alandaki veri eksikliğini telafi etmek için 2705 uydu görüntüsünü içeren ATA-Plane veri kümesi Google Earth Pro kullanılarak oluşturuldu. Her görüntüde en az bir hedef bulunan, dünyanın farklı bölgelerinden, farklı boyutlarda yaklaşık 8584 örnek toplandı. Veri kümesindeki resim sayısını artırmak için, Google Earth Pro'nun geçmiş görüntüleri özelliğini kullanarak aynı konumlardaki resimler farklı zamanlarda elde edildi. İkinci aşamada, hava taşıtlarının tespiti Evrişimsel Sinir Ağına (CNN) dayanan algoritmalar kullanılarak yapıldı. Bölgesel Tam Bağlı Evrişimsel Ağ (R-FCN), Tek Çekim Çok Kutu Dedektör (SSD) ve Daha Hızlı R-CNN yöntemleri bu işlem için kullanıldı. R-FCN, SSD ve Faster R-CNN için elde edilen doğruluk oranları sırasıyla %98.01, %69.71 ve %96.56 olarak elde edildi.

**2019, 62 sayfa**

**Anahtar Kelimeler:** Uçak tespiti, uydu görüntü analizi, nesne tespiti, Faster R-CNN, R-FCN, SSD.

## **ABSTRACT**

MS Thesis

### **AIRCRAFT DETECTION FROM SATELLITE IMAGES**

Hussein Mahmood Abdo MOHAMMED

Atatürk University  
Graduate School of Natural and Applied Sciences  
Department of Electrical-Electronics  
Electronic Science

Supervisor : Asst. Prof. Dr. Emin Argun ORAL

Satellite image analysis is a research area in which many research studies are carried out for civil and military applications in the field of image processing. Satellite imagery has many applications including recognition, detection and classification of regions, buildings, roads, aircraft and other man-made objects. Among these, especially aircraft detection is strategically important for military applications and forms the basis of this study. In the first phase of the study, a new dataset of aircrafts is created from Google Earth images to compensate the shortage of data set in this area. This dataset, named ATA-Plane dataset, contains 2705 satellite images. Each dataset image contains at least one target of different sizes collected from different parts of the world with 8584 instances. In the second stage, the detection of air vehicles was carried out using algorithms based on Convolutional Neural Network (CNN). Region-based Fully Convolutional Network (R-FCN), Single Shot Multi Box Detector (SSD) and Faster R-CNN methods are used for this process. The obtained accuracy rate for R-FCN, SSD and Faster R-CNN are 98.01%, 69.71% and 96.56%, respectively.

**2019, 62 pages**

**Keywords:** Aircraft Detection, Satellite Image Analysis, Object Detection, Region-based Fully Convolutional Network (R-FCN), Single Shot Multi Box Detector (SSD) and Faster Region-based Convolutional Neural Network (Faster R-CNN).

## TEŐEKKÜR

Bu alıŐma sűresince eŐitli zamanlarda ilgi ve desteęi ile beni yűnlendiren ve kıymetli gűrűŐlerinden yararlandıęım baŐta tez danıŐmanım Sayın Dr. Őęr. Ŭyesi Emin Argun ORAL olmak űzere ve Sayın Do. Dr. İbrahim Yűcel ŐZBEK'e űzellikle teŐekkűrű bir bor bilirim.

GerekleŐtirdięim bu tez alıŐmasının uygulama aŐamalarında bana yardımlarını esirgemeyen Aslı Nur Őmeroęlu, Merve Polat ve Nida Kumbasar'a űkran borluyum.

Őęrenim hayatım boyunca gűsterdikleri sabır ve fedakarlıklardan dolayı aileme sonsuz teŐekkűr ederim.

**Hussein Mahmood Abdo MOHAMMED**

**Temmuz, 2019**

## İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ.....	vi
ŞEKİLLER DİZİNİ.....	viii
<b>1. GİRİŞ.....</b>	<b>1</b>
1.1. Makine Öğrenmesi.....	2
1.1.1. Denetimli öğrenme.....	4
1.1.2. Denetimsiz öğrenme.....	5
1.1.3. Pekiştirmeli öğrenme.....	6
1.2. Yapay Sinir Ağları.....	6
1.2.1. Aktivasyon fonksiyonları.....	7
1.2.1.a. Sigmoid fonksiyonu.....	8
1.2.1.b. Tanh fonksiyonu.....	8
1.2.1.c. ReLU (Rectified Linear Unit) fonksiyonu.....	9
1.2.2. Yapay sinir ağı mimarisi.....	10
1.3. Derin Öğrenme.....	11
1.3.1. Evrişimsel sinir ağları (ConvNet/CNN).....	12
1.4. Bilgisayarlı Görme.....	14
1.4.1. Görüntü sınıflandırma.....	15
1.4.2. Yerelleştirme.....	17
1.4.3. Nesne tespiti.....	17
1.4.4. Semantik segmentasyon (anlamsal bölütleme).....	18
1.5. Transfer Öğrenmesi (Transfer Learning).....	19
1.5.1. Önceden eğitilmiş model.....	20
1.5.2. İnce ayar yapmanın yolları.....	21
1.5.2.a. Öznitelik çıkarma.....	21
1.5.2.b. Önceden eğitilmiş modelin mimarisini kullanma.....	21

1.5.2.c. Dięer katmanları dondururken bazı katmanların eęitilmesi .....	21
1.6. Uydu Grntlerinde Nesne Tespiti.....	22
<b>2. KAYNAK ZETLERİ .....</b>	<b>24</b>
<b>3. MATERYAL ve YNTEM.....</b>	<b>33</b>
3.1. Veri Kmesi .....	34
3.1.1. Veri kmesinin oluřturulması .....	34
3.1.2. Veri kmesinin etiketlenmesi.....	36
3.2. Nesne Tespiti Algoritmaları .....	37
3.2.1. FASTER R-CNN.....	41
3.2.2. R-FCN .....	45
3.2.3. SSD.....	48
<b>4. ARAřTIRMA BULGULARI .....</b>	<b>51</b>
<b>5. SONUÇ .....</b>	<b>58</b>
KAYNAKLAR .....	59
ZGEÇMİř .....	63



## SİMGELER ve KISALTMALAR DİZİNİ

ANN	Yapay Sinir Ağları (Artificial Neural Network)
BING	Nesne Tahmini için Binarize Normlu Gradyanlar (Binarized Normed Gradients for Objectness Estimation)
CEM	Kısıtlı Enerji Azaltma (Constraint Energy Minimization)
CNN	Evrişimsel Sinir Ağları (Convolutional Neural Network)
DBN	Derin İnanç Ağı (Deep Belief Net)
DPM	Deforme Olabilen Parça Modeli (Deformable Part Model)
Fast R-CNN	Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı (Fast Region-based Convolutional Neural Network)
Faster R-CNN	Daha Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı (Faster Region-based Convolutional Neural Network)
FC	Tam Bağlantılı Katman
FCNN	Tamamen Konvolüsyonlu Sinir Ağı (Fully Convolutional Neural Network)
FN	Yanlış Negatif (False Negative).
FP	Yanlış Pozitif (False Positive).
HOG	Yönlendirilmiş Degrade Histogram (Histogram of Oriented Gradients)
IoU	Birleşimin Kesişime (Intersection over Union)
LARK	Yerel Uyarlamalı Regresyon Çekirdeği (Locally Adaptive Regression Kernel)
M-FCN	Markov rastgele alan tamamen evrişimli ağ (Markov random field-fully convolutional network)
NLP	Doğal dil işleme (Natural Language Processing).
PCA	Temel Bileşen Analizi (Principal Component Analysis)
R-CNN	Bölge Tabanlı Evrişimsel Sinir Ağı (Region-based Convolutional Neural Network)
R-CNN	Bölge Tabanlı Evrişimsel Sinir Ağı (Region-based Convolutional Neural Network)

R-FCN	Bölgeye Dayalı Tamamen Evrişimli Ağlar (Region-based Fully Convolutional Networks)
RLN	Bölge Konumlandırma Ağı (Region Locating Network)
RoI	İlgi Bölgesi (Region of interest –RoI)
RP	Bölge Öneri (Region Proposal –RP)
RPN	Bölge Öneri Ağı (Region Proposal Network- RPN)
SCA	Tek katmanlı Hücresel Otomatlar (Single-layer Cellular Automata)
SIFT	Ölçek Değişmez Özellik Dönüşümü (Scale-Invariant Feature Transform)
SSD	Tek Çekim Çok Kutu Dedektör (Single Shot MultiBox Detector)
SVM	Destek Vektör Makinesi (Support-Vector Machine)
TN	Doğru Negatif (True negative- TN )
TP	Doğru Pozitif (True positive -TP)
YOLO	Sadece Bir Kere Bakarsın (You Only Look Once)

## ŞEKİLLER DİZİNİ

Şekil 1.1. Yapay zeka, makine öğrenmesi ve derin öğrenme arasındaki ilişki.....	2
Şekil 1.2. Öğrenme türleri ve ilgili bazı problemler .....	4
Şekil 1.3. Denetimli öğrenme modeli .....	5
Şekil 1.4. Denetimsiz öğrenme modeli .....	5
Şekil 1.5. İnsan beyninde yer alan nöron .....	6
Şekil 1.6. Yapay sinir ağlarında yer alan nöron .....	6
Şekil 1.7. Yapay nöron yapısı .....	7
Şekil 1.8. Sigmoid fonksiyon .....	8
Şekil 1.9. Tanh fonksiyonu .....	9
Şekil 1.10. ReLU fonksiyonu .....	9
Şekil 1.11. Genel yapay sinir ağı mimarisi .....	10
Şekil 1.12. Makine öğrenimi ile derin öğrenme arasındaki fark.....	11
Şekil 1.13. Evrişim örneği .....	13
Şekil 1.14. Maksimum havuzlama örneği .....	13
Şekil 1.15. Evrişimli sinir ağı örneği .....	14
Şekil 1.16. Görüntü sınıflandırma.....	16
Şekil 1.17. Görüntü yerleştirme .....	17
Şekil 1.18. Nesne tespiti .....	18
Şekil 1.19. Semantik segmentasyon.....	19
Şekil 1.20. Transfer öğrenmesini .....	20
Şekil 1.21. İnce ayar yapmanın yolları .....	22
Şekil 1.22. Uydu görüntülerinde nesne tespiti .....	23
Şekil 3.1. Google Earth Pro arayüzü.....	33
Şekil 3.2. ATA-Plane veri kümesinin örnekleri .....	35
Şekil 3.3. LabelImg programının arayüzü .....	36
Şekil 3.4. Etiketlenmiş görüntülerden örnekler .....	37
Şekil 3.5. Görüntü sınıflandırma ve nesne tespiti .....	38
Şekil 3.6. Seçmeli Arama algoritmasının örneği .....	41
Şekil 3.7. R-CNN mimarisi.....	42

<b>Şekil 3.8.</b> Fast R-CNN mimarisi .....	43
<b>Şekil 3.9.</b> Faster R-CNN mimarisi .....	44
<b>Şekil 3.10.</b> Bölge öneri ağı (RPN) .....	45
<b>Şekil 3.11.</b> R-FCN Mimarisi .....	46
<b>Şekil 3.12.</b> Sınıf puanı sonuçları oluşturması .....	47
<b>Şekil 3.13.</b> R-FCN'de pozisyona duyarlı sınıflandırma ve sınırlayıcı kutu regresyon işlemi .....	48
<b>Şekil 3.14.</b> SSD Mimarisi .....	49
<b>Şekil 4.1.</b> Birleşimin Kesişime Oranı .....	51
<b>Şekil 4.2.</b> Faster R-CNN Precision-Recall eğrisi.....	53
<b>Şekil 4.3.</b> R-FCN Precision-Recall eğrisi .....	53
<b>Şekil 4.4.</b> SSD Precision-Recall eğrisi.....	54
<b>Şekil 4.5.</b> Faster R-CNN, R-FCN ve SSD için Precision- Recall eğrisi.....	54
<b>Şekil 4.6.</b> Faster R-CNN modelin test sonuçları.....	55
<b>Şekil 4.7.</b> R-FCN modelin test sonuçları .....	56
<b>Şekil 4.8.</b> SSD modelin test sonuçları.....	56

## 1. GİRİŞ

Yapay zekâ (YZ), günümüzde ilgi duyulan ve tartışılan konuların başında gelmektedir. Yapay zekâ ifadesi ilk olarak 1956 yılında John McCarthy, Marvin Minsky, Nathaniel Rochester ve Claude Shannon'ın Dartmouth'ta gerçekleştirdikleri çalıştay için verdikleri öneride kullanılmıştır. Bilgisayar bilimlerinin özel bir alanı olan yapay zekada insanların veya hayvanların sahip olduğu karar verme yetilerinin bilgisayarlara kazandırılması ile ilgilenilir. Bu sistemlerin kendi kendilerine öğrenebilme yeteneği taşımaları hedeflenir. Bilgisayar sistemlerinin belirli görevleri örnekler üzerinden öğrenmesini sağlayan yapay öğrenme bir yapay zeka yaklaşımıdır.

1959 yılında Arthur Samuel, makine öğrenmesini bilgisayarların açık bir şekilde programlanmadan öğrenebilme yeteneğini veren bir yapay zeka alt alanı olarak makine öğrenmesini tanımlamıştır. Makine öğrenimi, modern bilim ve mühendislikte en hızlı büyüyen ve en heyecan verici alanlardan biridir. Makine öğrenimi, bilgisayarlara açıkça programlanmadan öğrenme ve hareket etme yeteneği sağlayan bir bilgisayar bilimi alanıdır. Makine öğrenmesi, iyi performansla açık yöntemler tasarlamamanın ve programlamamanın zor ya da olanaksız olduğu çok çeşitli hesaplama görevlerinde uygulanır.

Derin öğrenme olarak bilinen yeni nesil yapay öğrenme yaklaşımları sayesinde yapay zeka çalışmaları tüm dünyada hız kesmeden devam etmektedir. Derin öğrenme, daha geniş bir makine öğrenme yöntemleri ailesinin bir parçasıdır. Derin öğrenme yöntemleri, özellik çıkarma ve dönüştürme için birçok doğrusal olmayan işlem birimi katmanını kullanır. Ardışık her katman, önceki katmandaki çıktıyı girdi olarak kullanır. Yüksek seviye özellikler, hiyerarşik bir gösterimi oluşturmak için düşük seviye özelliklerden elde edilir. Derin sinir ağları, derin inanç ağları ve tekrarlayan sinir ağları gibi derin öğrenme mimarileri, bilgisayarlı görü, konuşma tanıma ve doğal dil işleme gibi alanlarda yaygın olarak uygulanmaktadır.



**Şekil 1.1.** Yapay zeka, makine öğrenmesi ve derin öğrenme arasındaki ilişki

### 1.1. Makine Öğrenmesi

Makine Öğrenmesi, bilgisayarların insanlar gibi öğrenmelerini ve davranmalarını, gözlem ve gerçek dünya etkileşimleri şeklinde veri ile besleyerek öğrenmelerini zaman içinde geliştirmelerini sağlayan bilimdir.

Tom M. Mitchell'in Makine Öğrenmesi için bilgisayar bilimleri perspektifinden ünlü tanımı:

"Bir bilgisayar programı; eğer  $T$  görev sınıfları ve  $P$  performans ölçütüne göre  $E$  deneyimi için  $T$ 'deki görevleri  $P$  ile ölçülen performansı  $E$  deneyimleri ile artıyorsa öğrenir" (Mitchell 1997) şeklindedir.

$T$  görevi algoritmanın öğrenme amacıdır. Kullanıcı tarafından tanımlanır ve genellikle bir olayın nasıl işleneceğine dair bir örnek sağlayarak algoritmaya iletilir. Burada olay, tek bir veri noktasını tanımlayan bir dizi öznitelik anlamına gelir. Genellikle, olayın özniteliklerini temsil eden  $x_i$  girişleriyle birlikte bir  $\mathbf{X} \in \mathbf{R}_n$  vektörü olarak ifade edilir.

**Performans ölçütü P**, algoritmanın yeterliliğinin değerlendirildiği metriktir. Metrik kullanıcı tarafından seçilebilir ve genellikle T görevine göre uyarlanmalıdır. Sınıflandırma durumunda, P performans ölçütünde genellikle tahmini ve gerçek çıktı arasındaki uyumsuzluğu ölçen kayıp fonksiyonu uygulanır. Basit bir durumda, kayıp fonksiyonu, toplam olay sayısına oranla doğru şekilde sınıflandırılmış olayların miktarı olabilir (Nelli and Nelli 2018).

**E deneyimi**, sağlanan veri setini ve makine öğrenme algoritmasının öğrenmek için kullanabileceği ek bilgileri kapsar. Denetlenen ve denetlenmeyen algoritmaların farklılık gösterdiği yer burasıdır.

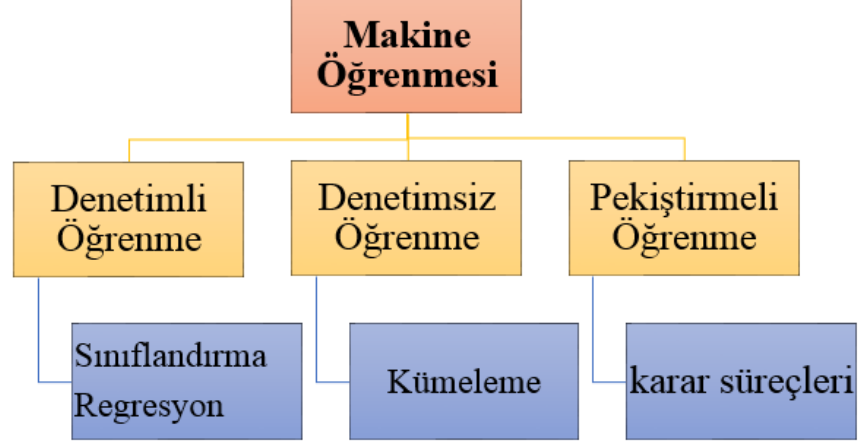
Tüm öğrenme süreci, aşağıdaki gibi bir veri seti gerektirir:

- Eğitim seti: Makine öğrenme algoritmasının parametrelerine uymak için kullanılan bilgi tabanıdır. Bu aşamada, eğitim seti en uygun ağırlıkları bulmak için kullanılır.
- Doğrulama seti: Bu, bir makine öğrenmesi modelinin parametrelerini ayarlamak için kullanılır.
- Test seti: Bu, modelin görünmeyen veriler üzerindeki performansını değerlendirmek için kullanılır.

Öğrenme teorisi, olasılık teorisinden türetilen matematiksel araçlarda kullanır. Öğrenme türleri kendi arasında 3'e ayrılır. Bunlar;

- Denetimli Öğrenme
- Denetimsiz Öğrenme
- Pekiştirmeli Öğrenme

Şekil 1.2'de öğrenme türleri gösterilmiştir.



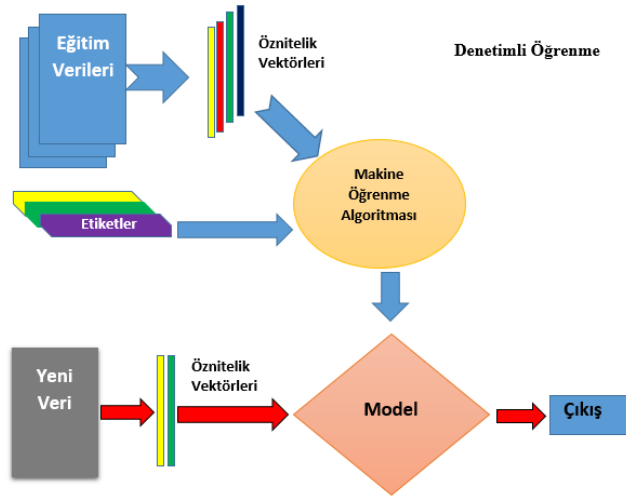
**Şekil 1.2.** Öğrenme türleri ve ilgili bazı problemler

### 1.1.1. Denetimli öğrenme

Denetimli öğrenme, modelin etiketli bir veri setinde eğitildiği öğrenme algoritmasıdır. Etiketli veri kümesi, hem giriş hem de çıkış parametrelerine sahiptir. Bu tür öğrenmede hem eğitim hem de doğrulama veri setleri etiketlenmektedir.

Örneğin, bilgisayarın kedi ve köpek resimlerinin nasıl ayırt edeceği öğrenilmek isteniyorsa, bazı kedi ve köpek resimlerinin sırasıyla “kedi” veya “köpek” ile etiketlenmesi gerekir. Etiketleme, genellikle verilerin yüksek kalitede olmasını sağlamak için insan tarafından yapılır. Resimlerin bu gerçek etiketlerine sahip olarak, görüntüleri sınıflandırmanın doğru yolu öğrenme algoritması ile bulunur. Bunların nasıl sınıflandırılacağı öğrenildikten sonra, yeni veriler üzerinde kullanılabilir ve daha önce görülmemiş görüntülerde etiketleri ("kedi" veya "köpek") tahmin edilebilir. Doğrusal Regresyon, Lojistik Regresyon, Karar Ağacı ve Destek Vektör Makineleri denetimli makine öğrenmesi algoritmalarına örnek olarak verilebilir. Şekil 1.3'te denetimli öğrenme kullanılarak oluşturulan temsili modelin işleyişi gösterilmiştir.

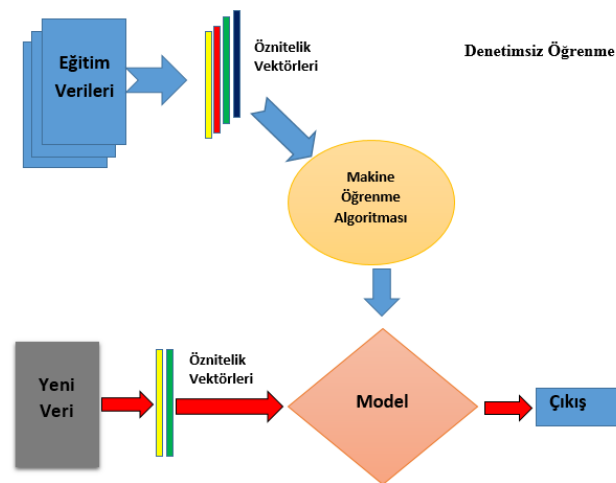




Şekil 1.3. Denetimli öğrenme modeli

### 1.1.2. Denetimsiz öğrenme

Denetimsiz Öğrenme, herhangi bir etiketli veri veya hedef veri kullanılmaz. Bu algoritma türleri, büyük miktarda veri veya bilgiyi farklı alt gruplara ayırır. Bu, nesnelere (müşteri, popülasyonlar vb.) farklı gruplara kümelemek veya bölümlenmek için yaygın olarak kullanılır. K-ortalama (K-means) , PCA vb. Denetimsiz Makine Öğrenmeye örnektir. Şekil 1.4’te denetimsiz öğrenme ile oluşturulan temsili modelin işleyişi şematize edilmiştir.



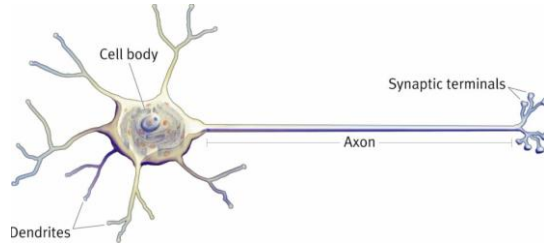
Şekil 1.4. Denetimsiz öğrenme modeli

### 1.1.3. Pekiştirmeli öğrenme

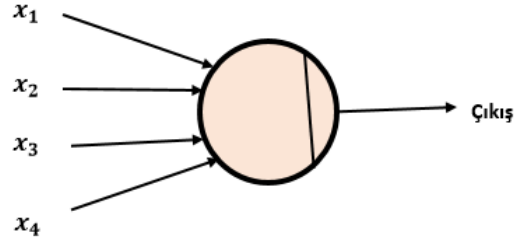
Pekiştirmeli öğrenme, bilgisayarın o an gerçekleştirmeye odaklandığı hedefinde ne yapması gerektiğini öğrendiği bir makine öğrenmesi yaklaşımıdır. Pekiştirmeli öğrenmede ajan adı verilen öğrenen makinemiz karşılaştığı durumlara bir tepki verir ve bunun karşılığında da sayısal bir ödül sinyali alır. Öğrenen makine aldığı bu ödül puanını maksimuma çıkartmak için çalışır. Bu şekilde çalışan deneme yanılma yöntemi, pekiştirmeli öğrenmenin en belirleyici özelliklerindedir.

### 1.2. Yapay Sinir Ağları

Makine öğrenmesinin ve yapay zekanın temelde amacı genel olarak insan beyninin ne düşündüğünü taklit eden akıllı sistemler oluşturmaya çalışmaktır. Yapay sinir ağları insan beynini oluşturan biyolojik sinir ağlarından ilham alan bilgisayar sistemleridir. Bu tür sistemler, genellikle göreve özel kurallarla programlanmadan, örnekleri dikkate alarak görevleri gerçekleştirmeyi “öğrenir”. İnsan beynine benzer şekilde, hepsi birbirine bağlı ve toplu halde güçlü bir sistem oluşturan çok sayıda nörondan oluşur. Şekil 1.5’de insan beyninde yer alan tek bir nöron gösterilirken, Şekil 1.6’da nöronun yapay sinir ağlarındaki temsili verilmiştir. Her ikisinin de birçok girişi vardır, ancak ağdaki diğer nöronlara bilgi dağıtmaya devam eden sadece birkaç çıkışı vardır.



**Şekil 1.5.** İnsan beyninde yer alan nöron (Anonim 2015)

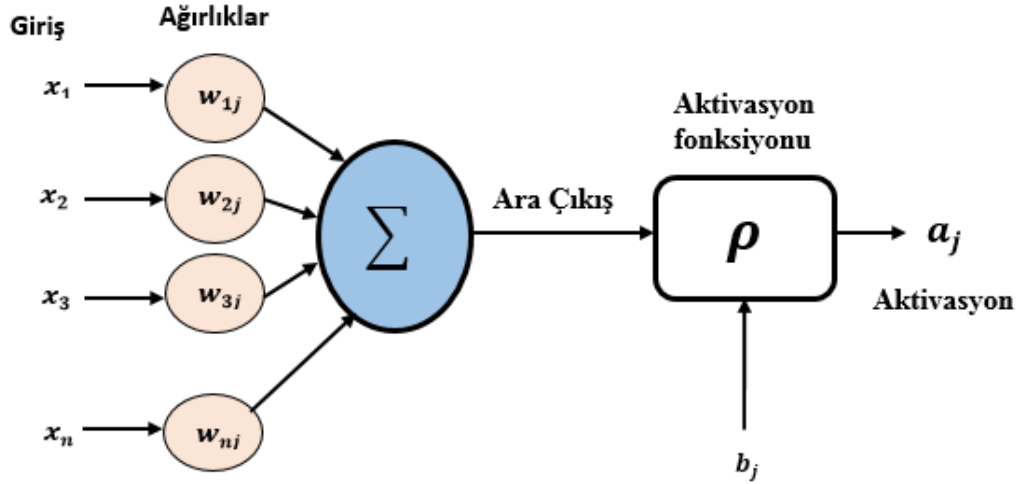


**Şekil 1.6.** Yapay sinir ağlarında yer alan nöron

Her yapay nöron, belirli bir görev sırasını tamamlar: Bunlar:

- $x_1, x_2, x_3 \dots x_n$  değerlerini girdi olarak alır, bunları karşılık gelen  $w_1, w_2, w_3 \dots w_n$  ağırlıklara göre ağırlıklandırır, toplar ve ara çıkış oluşturur. Bu şekilde transfer fonksiyondan geçirilmiş olur
- Yukarıdaki sonuç daha sonra çıkışın üretilmesi için aktivasyon işlevinden geçirilir.
- Oluşturulan çıkışı kontrol etmek için bir önyargı (bias) diye adlandırılan sabit eklenir.

Bu açıklamalar dikkate alınarak Şekil 1.7’de yapay nöronun detaylı gösterimi, denklem 1.1’de de formülü verilmiştir.



Şekil 1.7. Yapay nöron yapısı

$$a_j = \rho \left( \sum_{k=1}^n w_{kj} x_k + b_j \right) \quad (1.1)$$

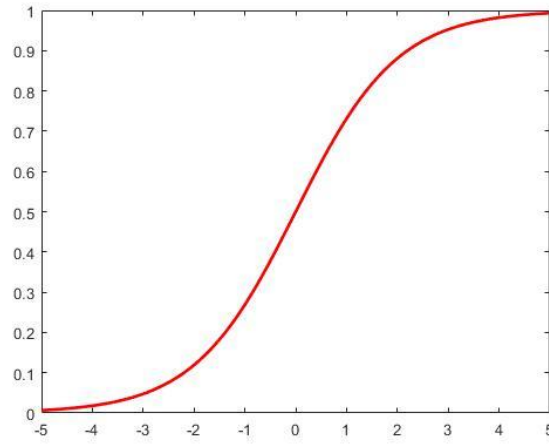
### 1.2.1. Aktivasyon fonksiyonları

Aktivasyon fonksiyonu, bir nöron çıkışında karar mekanizması olarak kullanılır. Aktivasyon fonksiyonları doğrusal veya doğrusal olmayan olabilir. Nöron, aktivasyon

fonksiyonuna dayalı doğrusal veya doğrusal olmayan karar sınırlarını öğrenir. Ayrıca, kademeli etki nedeniyle, birkaç katmandan sonra nöronların çıkışının çok büyük olmasını engelleyen nöron çıkışı üzerinde normalize edici bir etkiye sahiptir. En çok kullanılan üç aktivasyon fonksiyonu vardır.

### 1.2.1.a. Sigmoid fonksiyonu

Kendisine giriş olarak aldığı değerleri 0 ile 1 aralığına dağıtarak çıktı oluşturur. Grafiği Şekil 1.8'de, matematiksel gösterimi denklem 1.2'de verilmiştir.

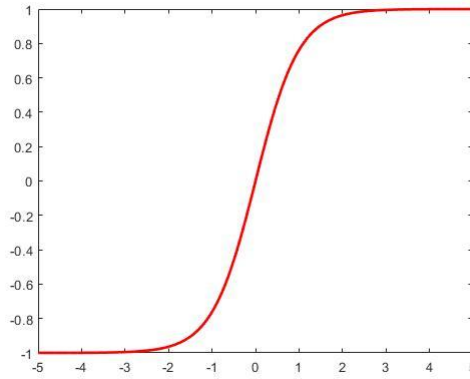


Şekil 1.8. Sigmoid fonksiyon

$$\rho(z) = \frac{1}{1+e^{-z}} \quad (1.2)$$

### 1.2.1.b. Tanh fonksiyonu

Kendisine giriş olarak aldığı değerleri -1 ile 1 aralığına dağıtarak çıktı oluşturur. Grafiği Şekil 1.9'da, matematiksel gösterimi denklem 1.3'de verilmiştir.

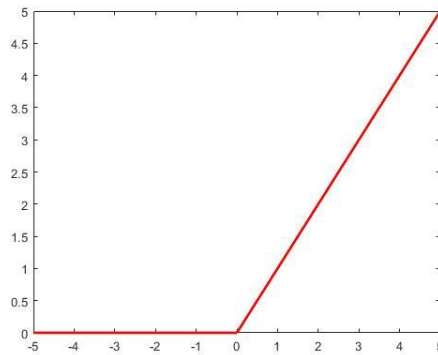


Şekil 1.9. Tanh fonksiyonu

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (1.3)$$

### 1.2.1.c. ReLU (Rectified Linear Unit) fonksiyonu

Kendisine giriş olarak aldığı değerlerden yalnızca pozitif olanları aynen çıkarıp negatif olanları 0 (sıfır)'a eşitleyerek çıkarır. Grafiği Şekil 1.10'de, matematiksel gösterimi denklem 1.4'de verilmiştir.



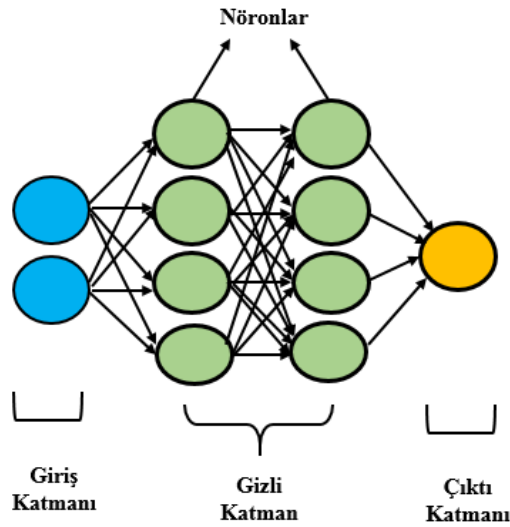
Şekil 1.10. ReLU fonksiyonu

$$a = \max(0, z) \quad (1.4)$$

### 1.2.2. Yapay sinir ağı mimarisi

YSA katmanlar halinde organize edilmiş bir dizi bağlı nörondur. Şekil 1.11’de temsili görüntüsü verilen yapay sinir ağı mimarisi aşağıdaki şekilde özetlenebilir.

- **Giriş Katmanı:** İşlenecek verilerin ağa giriş yaptığı katmandır.
- **Gizli Katman:** Yapay nöronların ağırlıklı girdiler kümesine girdiği ve bir aktivasyon fonksiyonu aracılığıyla bir çıktı ürettiği girdi katmanları ve çıktı katmanları arasındaki katmandır. Bu katmanda kullanılan en yaygın aktivasyon fonksiyonu ReLU’dur. Gizli katmanların sayısı, sinir ağının derinliği olarak adlandırılır.
- **Çıktı Katmanı:** Bu tahminleri veren katmandır. Bu katmanda kullanılacak olan aktivasyon fonksiyonu farklı problemler için farklıdır. İkili bir sınıflandırma probleminde çıkış 0 veya 1 olarak beklenmektedir. Dolayısıyla bu katmanda aktivasyon fonksiyonu olarak bir sigmoid fonksiyonu kullanılır. Çok sınıflı bir sınıflandırma problemi için, sigmoid fonksiyonun çoklu sınıflara genelleştirilmesi olarak düşünülen Softmax fonksiyonu kullanılır. Çıktının önceden tanımlanmış bir kategori olmadığı bir regresyon problemi için, sadece doğrusal bir birim kullanılabilir.

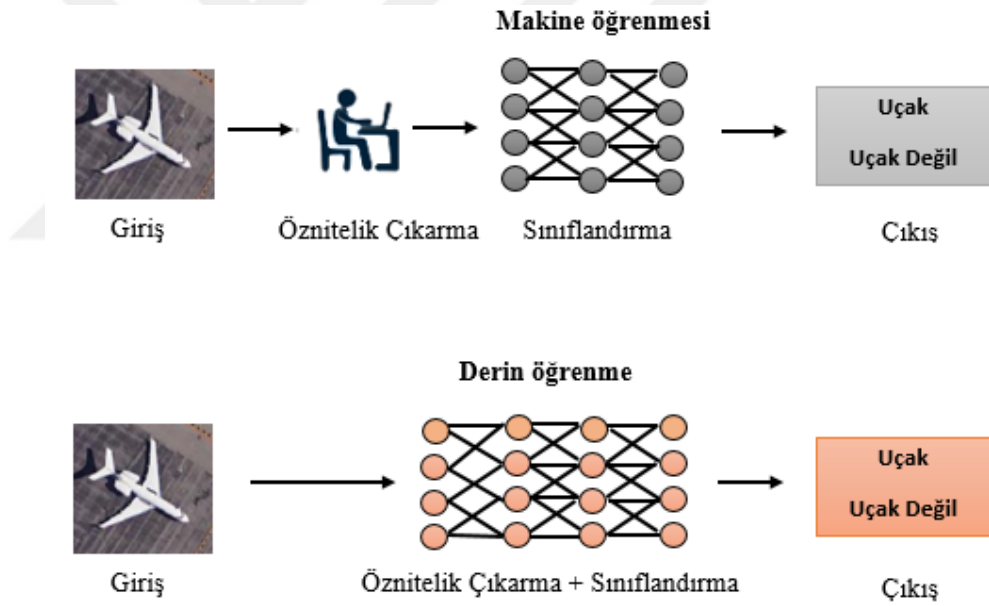


Şekil 1.11. Genel yapay sinir ağı mimarisi

### 1.3. Derin Öğrenme

Derin Öğrenme, yapay sinir ağları denilen beynin yapısından ve işlevinden ilham alan algoritmalarla ilgili bir makine öğrenmesi alanıdır. Derin öğrenme yöntemleri, özneliklerin manuel olarak çıkarıldığı klasik makine öğreniminin aksine, ilgili öznelikleri otomatik olarak ham verilerden çıkarmaktadır. Derin öğrenme yöntemleri, veriyi doğrudan kullanarak sınıflandırma, yerelleştirme ve nesne tespiti vb. görevleri gerçekleştirir.

Makine öğrenmesi ile derin öğrenme arasındaki fark, Şekil 1.12'de gösterilmektedir.



**Şekil 1.12.** Makine öğrenimi ile derin öğrenme arasındaki fark

Derin öğrenme yöntemleri, çeşitli alanlardaki en gelişmiş doğruluk sonuçlarına ulaşarak değerlerini kanıtlamışlardır. Bunlar, derin öğrenme yöntemlerinin dikkate değer bir başarı elde ettiği en belirgin alanların örnekleridir:

- Nesne algılama ve tanıma
- Konuşma tanıma ve dil çevirisi

- Kendi kendine sürüş arabaları
- Doğal dil işleme (NLP)
- Tıbbi araştırma
- Ve daha fazlası...

### 1.3.1. Evrişimsel sinir ağları (ConvNet/CNN)

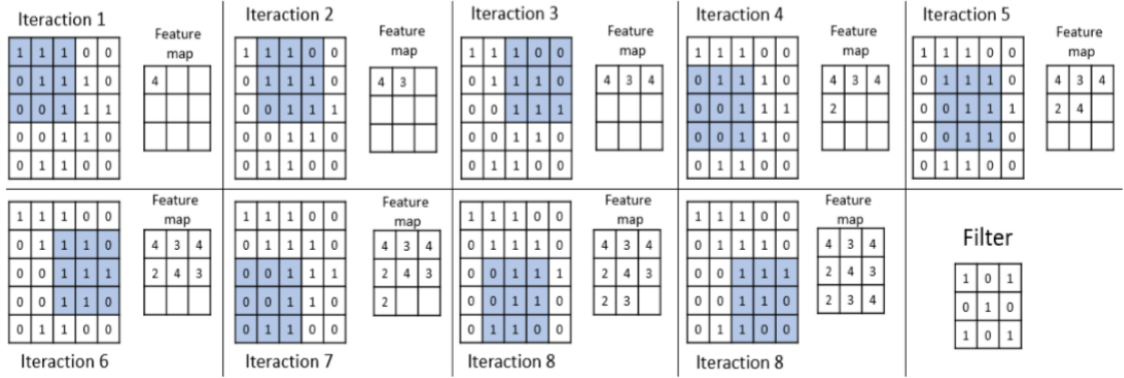
Yann LeCun (1998) tarafından önerilen CNN mimarisinin öncülerden biri olan, LeNet-5'e göre, CNN temel mimarisi, evrişimli katmana, havuzlama katmanına ve tam bağlı katmana sahip olmalıdır. Genel olarak, bu mimari, iyileştirmelere ve değişikliklere rağmen, temel kavramları sağlam tutan başka ve daha yeni CNN mimarilerinin temelidir.

#### Evrişim Katmanı

Evrişim katmanı, eşit boyuttaki küçük karoların kullanımıyla, uzamsal bağlantıyı piksellerden koruyan bir görüntüden görüntü içindeki öğrenilen özellikleri çıkarmaktır. Öğrenilen özellikler, girdi görüntüsündeki her eleman ile filtre matrisi arasındaki matematiksel işlemin bir sonucudur. Başka bir deyişle, filtre veya özellik algılayıcısı olarak da bilinen yapının eldeki giriş görüntüsü üzerinde kaydırılarak örtüşen elemanların çarpılması ve bu çarpımın toplanarak çakışma esnasında merkez olan elemanın yeni değeri olarak çıkış matrisine yerleştirilmesi işlemidir. Filtreler, görüntüden özellik algılayıcı bir mekanizma gibi davranır. Kullanılacak filtre sayısı değişken olup filtre boyutu ölçeklendirilebilmektedir. Filtre derinliği ise katmanın çıktısını oluşturacak öznetelik haritasının boyutunu belirler (Gibiansky, 2015).

Şekil 1.12'de 5x5 boyutunda matris üzerinde 3x3 lük filtre ile kaydırma adımı 1 olacak şekilde evrişim işlemi örneklendirilmiştir.





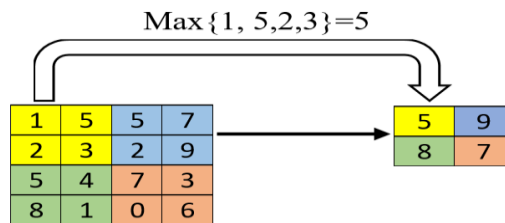
**Şekil 1.13.** Evrişim örneği (Zaccone *et al.* 2017)

CNN'lere doğrusal olmayan özellikler ekleyen, doğrusal olmayan modelleri öğrenmesini sağlamak amacıyla aktivasyon için ReLU fonksiyonu kullanılır. ReLU, nesne tanıma için CNN'lerin performansını önemli ölçüde artırmaktadır (Gomez, Cortes, & Noguer, 2015).

### Havuzlama Katmanı

Daha önce de belirtildiği gibi, evrişimli sinir ağlarının ayırt edici kavramlarından biri havuzlamadır. Havuzlama adımı, her öznetelik haritasının boyutunu azaltarak, gürültüyü hesaplama karmaşıklığını azaltmaya yarayan bir adımdır. Maksimum, toplam veya ortalama gibi birden fazla havuzlama türü vardır, ancak en yaygın ve en çok tercih edilen havuzlama maksimum havuzlamadır. Maksimum havuzlama işleminde matris içinde alt matris olabilecek bölgede komşular içinde en büyük olan elamanın seçilerek çıkış matrisine atanması işlemi yapılır.

Şekil 1.14'de 4x4lük matris üzerine 2x2lik maksimum havuzlama işlemi uygulanmıştır.

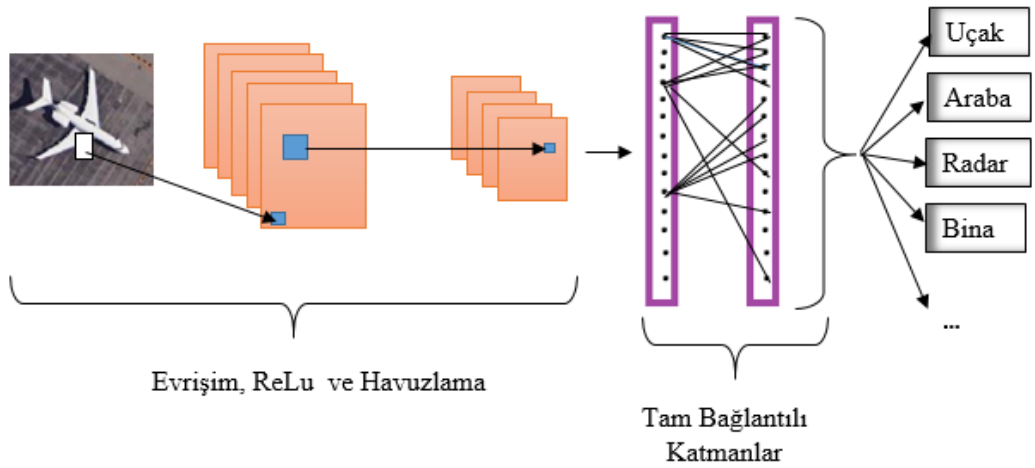


**Şekil 1.14.** Maksimum havuzlama örneği

## Tam Bağlantılı Katman

Evrişimli sinir ağlarının katmanlarından biri olan ve çıktı katmanından bir önce gelen tam bağlantılı katman o aşamaya gelen yüksek seviyeli özelliklere bakar ve ağırlıklar ile önceki katman arasındaki yeni hesaplamaları yaparak her bir kategori için olasılık hesaplama işlemlerini gerçekleştirir. Diğer bir deyişle tam bağlantılı katman, çıktı özelliklerini evrişim veya havuzlama katmanı olabilen önceki katmandan kullanmayı ve görüntüyü eğitim veri kümesine göre sınıflandırmayı amaçlar (Wan *et al.* 2014).

Şekil 1.15’de Genel olarak bir evrişimli sinir ağında tam bağlantılı katman işlevi resmedilmiştir.



Şekil 1.15. Evrişimli sinir ağı örneği

## 1.4. Bilgisayarlı Görme

Bilgisayarlı Görme, önemli bir bilgisayar bilimi alanı olarak, her zaman makine öğrenimi ile yakından ilişkili olmuştur. Son zamanlarda, bu iki alan sinir ağlarındaki gelişmeler nedeniyle birbirleriyle giderek daha fazla ilişkili hale geldi.

Bilgisayarlı Görme alanı, tıpkı insanlar gibi hatta daha da iyi görüntüleri işleyebilen sistemler oluşturma hedefi ile ilgilidir.

Bilgisayarlı görme, bilgisayarların dijital görüntüleri ve videoları nasıl gördüğünü ve anladığını çevreleyen çalışma alanıdır. Bilgisayarlı görme, görsel uyarıcıyı "görme" veya görme, ne olduğunu anlama ve karmaşık bilgileri diğer işlemlerde kullanılacak bir forma çıkarma dahil üzere biyolojik görme sistemleri tarafından gerçekleştirilen tüm görevleri kapsar. Bu disiplinlerarası alan, sensörler, bilgisayarlar ve makine öğrenmesi algoritmaları kullanarak insan görüş sistemlerinin bu öğelerini simüle eder ve otomatikleştirir. Bilgisayarlı görme'nin uygulamaları, endüstriyel makine görme sistemleri gibi görevlerden yapay zeka ve bilgisayarları araştırmaya ve etraflarındaki dünyayı anlayabilen robotlara kadar uzanmaktadır.

Bilgisayarlı görme görevleri dört farklı alana ayrılabilir.

1. Görüntü sınıflandırma
2. Yerelleştirme
3. Nesne Tespiti
4. Semantik Segmentasyon (Anlamsal Bölütleme)

#### **1.4.1. Görüntü sınıflandırma**

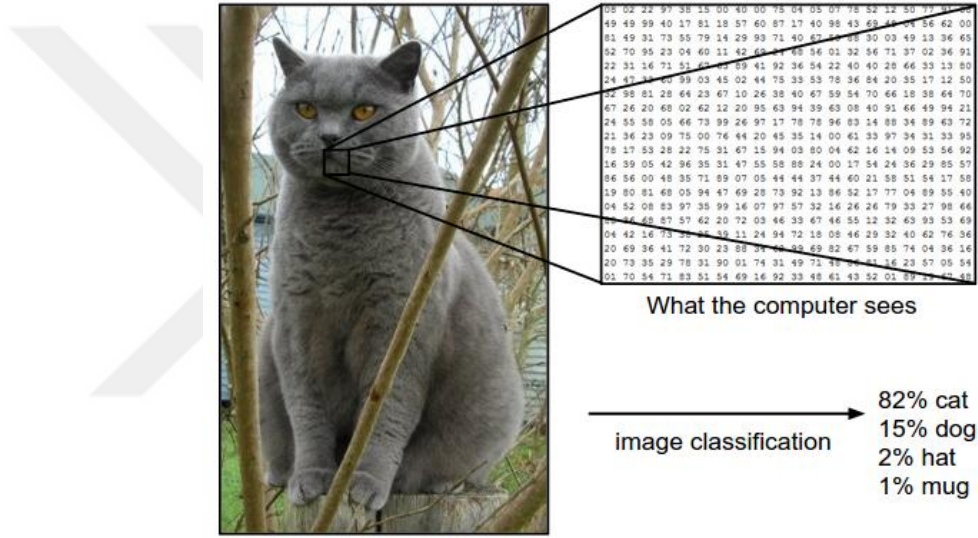
Sınıflandırma, özelliklerine göre gruplarda ve kategorilerde sistematik bir düzenlemedir.

Görüntü sınıflandırma bilgisayarları verilerle eğiterek insanların algılama yeteneklerine yaklaştırmayı hedefleyen olgulardan biridir.

Görüntü sınıflandırması, görüntünün içeriğine bağlı olarak görüntüyü belirtilen kategoriye ayırarak gerçekleştirilir. Görüntü sınıflandırmanın görevi, sabit bir kategori

setinden bir giriş görüntüsüne bir etiket atama işlemidir. Şekil 1.16'da, yaygın olarak bilinen bir görüntü sınıflandırma örneğini göstermektedir.

Görüntü sınıflandırma için kullanılan geleneksel yöntemler, makine öğrenmesi olarak adlandırılan yapay zeka alanının bir parçasıdır. Görüntü sınıflandırma amacıyla kullanılan çeşitli sınıflandırıcılar; Bayes sınıflandırıcısı, Yapay Sinir Ağları, Karar Ağacı sınıflandırıcısı, Genetik Algoritmalarıdır.

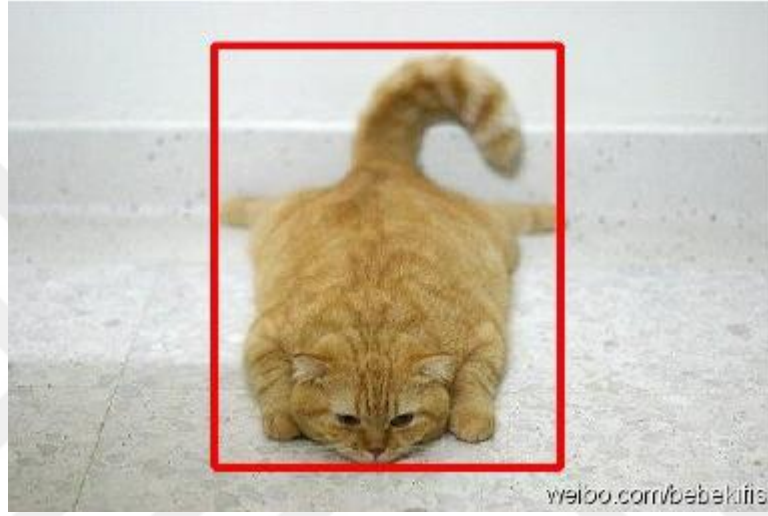


Şekil 1.16. Görüntü sınıflandırma (Karpathy 2019)

Görsel bir kavramı tanıma görevi bir insanın gerçekleştirmesi için göreceli olarak basit olduğundan, bakış açısı farklılığı, ölçek değişimi, farklı ışık koşulları, arka plan karışıklığı, deformasyon gibi daha birçok faktör bilgisayarlı görmede karşılaşılan problemlerdir. Etkili ve doğru model tasarımı bu zorlukların üstesinden gelebilecek potansiyelde olmalıdır.

### 1.4.2. Yerelleştirme

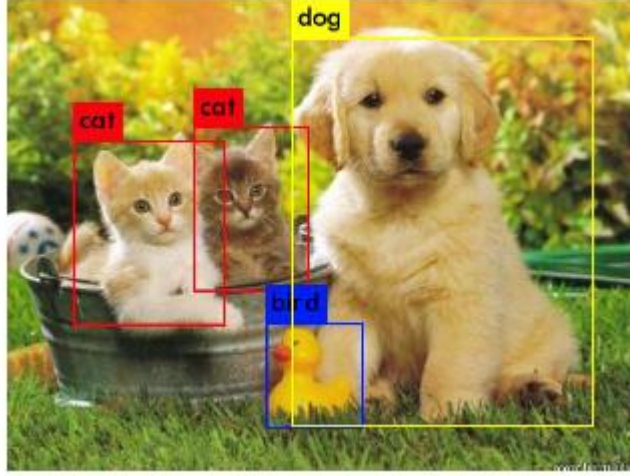
Yerelleştirme, belirli bir görüntüdeki tek bir nesneyi bulma işlemidir. Nesnenin nerede olduğu tespit edildikten sonra etrafına sınırlayıcı bir kutu çizilir. Şekil 1.17’de, bir görüntüdeki bir kedinin lokalizasyonunun basit bir örneğini göstermektedir.



Şekil 1.17. Görüntü yerelleştirme (Yinan Y 2019)

### 1.4.3. Nesne tespiti

Sınıflandırma ve yerelleştirme işlemlerinin eş zamanlı birleştirilmesinin sonucunda oluşan nesne tespiti, nesnenin çevresine sınırlayıcı kutunun çıktısını alarak nesnenin ait olduğu sınıfın örneğini belirleme ve nesnenin konumunu tahmin etme işlemidir. Tek bir sınıf örneğinin görüntüden tespit edilmesi, tek sınıf nesne tespiti olarak adlandırılırken, görüntüdeki tüm nesnelerin sınıflarının tespit edilmesi çok sınıflı nesne tespiti olarak bilinir. Şekil 1.18’de, üç tür nesnenin (kedi, köpek, kuş) tespit edilmesine ilişkin bir örneği göstermektedir.



**Şekil 1.18.** Nesne tespiti (Khan 2017)

Geçtiğimiz birkaç yıl boyunca, nesne tespiti alanında yapılan araştırmalar kayda değer bir ilerleme göstermiştir. Bu başarıda büyük ölçüde, manuel öznitelikler tasarlamaya veya karmaşık öncelikleri ortaya koymaya gerek kalmadan, örneklerden baştan sona öğrenmeye izin veren derin öğrenme tekniklerinin hızlı bir şekilde geliştirilmesinin katkısı vardır. Nesne tespitini yaparken değişken aydınlatma koşulları, pozlar, ölçek vb. gibi farklı zorlukların ele alınması gerekir. Bu uygulamadaki en büyük zorluk nesnelerin büyüklüklerinin, pozisyonlarının ve arka planının değişmesidir. Buna bağlı olarak, nesne tespiti literatüründe işlem yükü açısından basit veya karmaşık yapıları olan R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD gibi birçok yöntem bulunmaktadır.

#### **1.4.4. Semantik segmentasyon (anlamsal bölütleme)**

Bilgisayarlı görmenin daha zor bir görevi ise yalnızca görüntüdeki nesnelerin çevresine sınırlayıcı bir kutu koymak değil, görüntüdeki her pikseli sınıflandırmaktır.

Anlamsal bölütleme, bir görüntüdeki her bir pikseli ait olduğu kategori ile etiketlemekle ilgilidir. Piksel seviyesindeki etiketlerin birincil öneme sahip olduğu robotik, haritalama veya tıbbi görüntü analizi gibi geniş bir alanda çeşitli uygulamalar vardır. Son yıllarda,

derin sinir ağıları etkileyici sonuçlar göstermiş ve çeşitli tespit görevleri için son teknoloji haline gelmiştir.

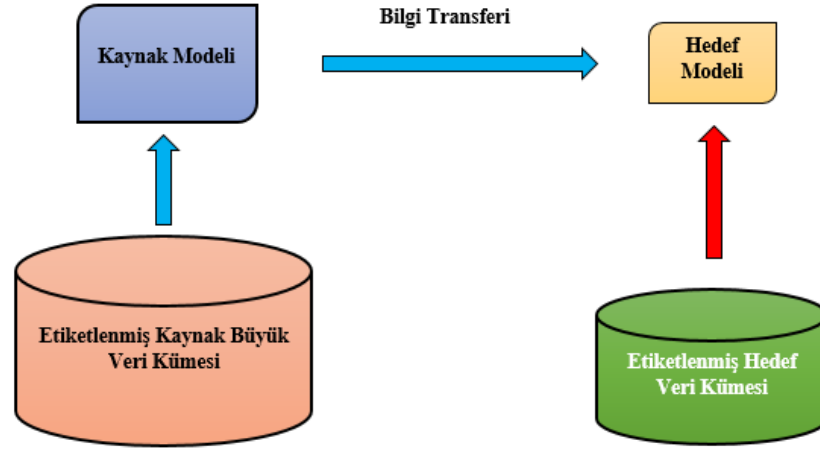


**Şekil 1.19.** Semantik segmentasyon

### 1.5. Transfer Öğrenmesi (Transfer Learning)

Bir CNN'nin uygulanması, derin ağların gerektirdiği geniş veri setine dayanarak uzun bir eğitim süresi gerektirir. Bu olumsuz durumun önüne geçmek için bir yaklaşım transfer öğreniminin uygulanmasıdır. Transfer Öğrenme daha önce geliştirilen bir modelin, büyük bir veri setinin kullanımına gerek olmadan yeni bir görevin başlangıç noktası olarak yeniden kullanılmasıdır. Bu nedenle, transfer öğrenmesi, yenisinin az miktarda veriye sahip olması durumunda, başka bir görevle ilgili önceden edinilmiş olan bilgilerden yararlanılmasını sağlar (Khan 2017). Uygulamada, çok az sayıda insan tüm evrişim ağını rastgele başlatarak sıfırdan eğitir çünkü yeterli büyüklükte bir veri kümesine sahip olmak her zaman kolay olmayabilir.

Örneğin 1000 kategoride 1,2 milyon görüntü içeren ImageNet gibi çok büyük bir veri setinde bir ConvNet'i önceden tanımlamak ve ardından ConvNet'i ilgili amaç için bir başlatma veya sabit bir öznitelik çıkarıcı olarak kullanmak yaygındır. Şekil 1.20 transfer öğrenmeyi göstermektedir.



**Şekil 1.20.** Transfer öğrenmesini

### 1.5.1. Önceden eğitilmiş model

Önceden eğitilmiş bir model, örneğin sınıflandırma ya da nesne tespiti gibi görevi yerine getirmek için geniş veri kümesi üzerinde eğitilmiş bir modeldir. Eğitim süreci, ağırlıkları rastgele başlatmak suretiyle başlar.

Eğitim başlar başlamaz, görevi daha az hatayla yerine getirmek için ağırlıklar değiştirilir yani optimizasyon işlemi gerçekleştirilir. Eğitim iyi sonuçlandığında, ağırlıkları bir yere kaydedilir.

Bir ağ, farklı bir veri setinde (örneğin görüntüler de, ancak daha önce kullanılanlarla aynı değil) yeni bir görev (örneğin, nesne algılama) gerçekleştirmek için eğitiliyorsa, rastgele başlatılmış ağırlıklarla eğitime başlamak yerine, önceki denemeden kaydedilen ağırlıkları yeni deneme için başlangıç ağırlık değerleri olarak kullanılması mümkündür. Ağırlıkları bu şekilde başlatmak, önceden eğitilmiş bir ağ kullanmak olarak adlandırılır. İlk ağ, önceden eğitilmiş ağıdır, ikincisi ise ince ayar yapılan ağıdır.



## **1.5.2. İnce ayar yapmanın yolları**

### **1.5.2.a. Öznitelik çıkarma**

Öznitelik çıkarma: Önceden eğitilmiş bir model öznitelik çıkarma mekanizması olarak kullanılabilir. Çıktı katmanını kaldırabilmek (1000 sınıfın her birinde olma olasılığını veren) ve ardından tüm ağı sabit bir öznitelik çıkarıcı olarak kullanabilmektir.

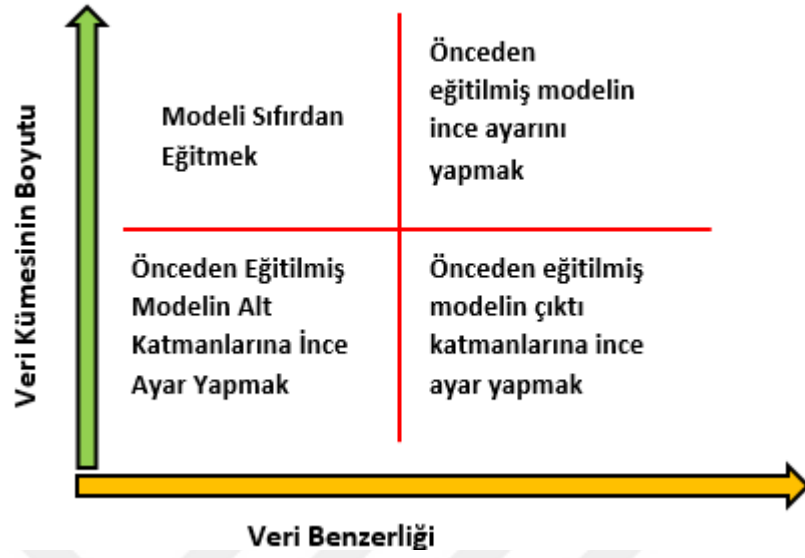
### **1.5.2.b. Önceden eğitilmiş modelin mimarisini kullanma**

Önceden Eğitilmiş Modelin mimarisini kullanma: Tüm ağırlıklar rastgele başlatılır ve yeni veri setine göre modelin eğitilmesi için model mimarisi kullanılır.

### **1.5.2.c. Diğer katmanları dondururken bazı katmanların eğitilmesi**

Diğer Katmanları Dondururken Bazı Katmanların eğitilmesi: Önceden eğitilmiş bir modeli kullanmanın başka bir yolu da kısmen eğitmektir. Modelin ilk katmanlarının ağırlıklarına dokunulmadan, sadece daha yüksek katmanlar yeniden eğitilir. Kaç katmana müdahale edilmeyeceği ve kaç katmanın eğitileceği test edilerek belirlenebilir.

Herhangi bir durumda önceden eğitilmiş modeli kullanmaya nasıl devam edileceğine karar vermeye yardımcı olacak diyagram aşağıda gösterilmektedir.



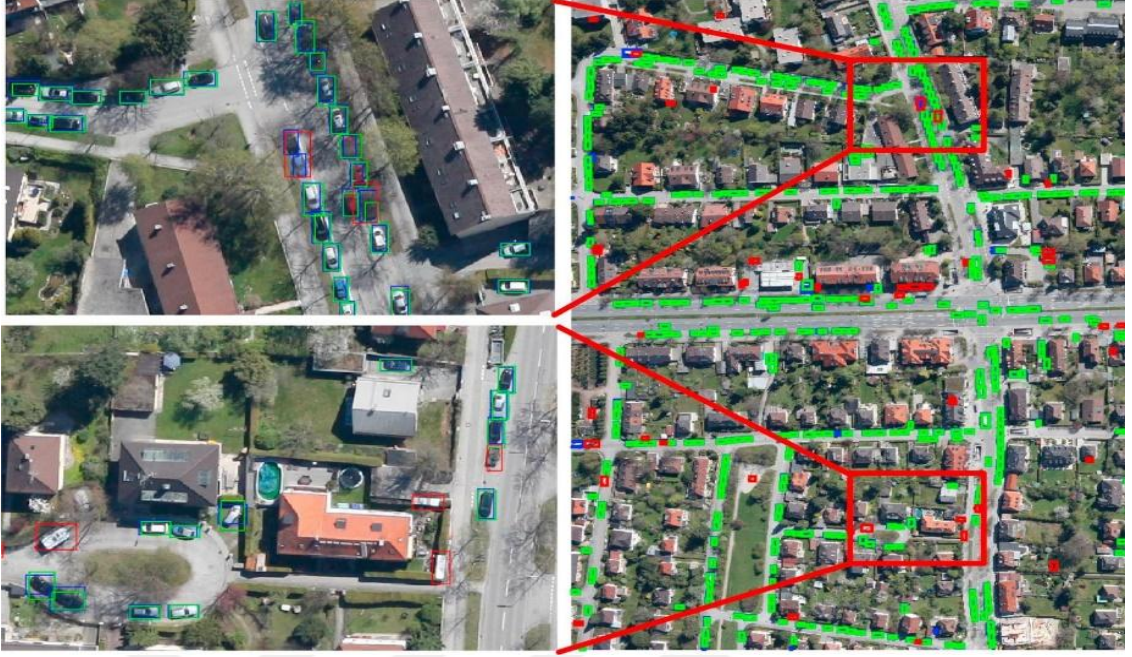
**Şekil 1.21.** İnce ayar yapmanın yolları

## 1.6. Uydu Görüntülerinde Nesne Tespiti

Uydu görüntüleri meteorolog tarafından kullanılan en güçlü ve önemli araçlardan biridir. Onlar esasen gökyüzündeki gözlerdir.

Uydu görüntülerinde nesne tespiti, verilen bir uydu görüntüsünün ilgilenilen sınıfa ait bir veya daha fazla nesne içerip içermediğini tespit etmek ve görüntüdeki her tahmin edilen nesnenin konumunu bulmak içindir.

Binalar, yol bölütlemeleri ve kentsel alan sınırları gibi uydu görüntülerinden nesnelere tespiti, özellikle devlet kurumları, kurtarma ekipleri, askeri ve diğer sivil kurumlar için çok önemli rol oynamaktadır. Uzman bir insan için, bu değerli bilgiyi elle çıkarmak sıkıcıdır ve hatalara açıktır. Bu nedenle, uzaktan algılanan görüntülerden nesnelere otomatik olarak algılamak için gelişmiş nesne tespit algoritmaları geliştirilmelidir. Uydu görüntülerinde nesne tespit görevi, evrimsel sinir ağlarında (CNN) elde edilen ilerlemeler nedeniyle son yıllarda kayda değer bir şekilde iyileşmiştir.



Şekil 1.22. Uydu görüntülerinde nesne tespiti (Deng *et al.* 2018)

## 2. KAYNAK ÖZETLERİ

Uydu görüntülerinden nesnelere algılanması ve tanınması, alandaki önemli araştırma alanlarından biridir. Uydu görüntü analizi, su ve iklim gözlemi, arazi örtüsü sınıflandırması ve enerji keşfi gibi birçok uygulamada faydalı olabilir. Özellikle, sabit uçakların tespiti askeri uygulamalarda stratejik olarak önemli olabilir. Bu tür bilgiler dinamik olarak değişen askeri hareketlerde daha sağlam ve başarılı karar mekanizmaları sağlar. Görüntü işleme algoritmaları kullanarak uçakların uydu görüntülerinden tespit edilmesi, uçakların geometrik yapılarının öğrenilmesine bağlıdır. Bu yapıyı daha belirgin hale getirmek, bu uçakların uydu görüntülerinden tespitinde önemli rol oynamaktadır.

Bu tez çalışmasında farklı nesne tespit etme yöntemlerine ait özellikle bu tez çalışmasının bir parçası olan uçak tespiti çalışmalarına ait sonuçlar aşağıda kısaca özetlenmiştir.

Li *et al.* (2018) çalışmalarında evrimsel sinir ağlarını (CNN'ler), görüntü analistlerinin algılama tarzını simule eden kaba ve ince bir şekilde kullanan bir uçak algılama yöntemi önermişlerdir. Önerilen metodta ilk önce birden fazla uçak içeren kaba aday bölgeler önerilmekte, ardından bu aday bölgelerdeki her bir uçak hassas şekilde tespit edilmektedir. Aynı yapıya sahip iki CNN'den oluşan kesin ve verimli bir algılama çerçevesi tasarlayarak bir CNN'i, aday bölgeleri kaba bir şekilde belirlemek için kullanırken, diğerini uçakları hassas şekilde tespit etmek için kullanmışlardır. CNN'leri eğitmek için Google Earth'ten  $2048 \times 1536$  piksel boyutunda 200 eğitim görüntüsü kullanmışlardır. Daha az aday bölge ve ayırt edici derin özellikler ile Google Earth resimlerindeki denemeler neticesinde %98.8 başarıya ulaşmışlardır.

Chen *et al.* (2018) uzamsal çözünürlüğe sahip yüksek çözünürlüklü uzaktan algılamalı görüntülerde uçağı etkin bir şekilde algılamak için alt-örnekleme yapılan görüntüdeki hedef alanı bulma ve ardından hedefin orijinal yüksek çözünürlükteki görüntüde hedefi bulma işlemi için hiyerarşik bir algılama çerçevesi önermişlerdir. Önerdikleri sistemde, Google Earth, Quickbird ve Worldview2 veri setinden elde edilen yaklaşık 60 uzaktan

algılama görüntüsü kullanmışlardır. Tüm görüntüler  $10000 \times 10000$  piksel ile  $19000 \times 23000$  piksel arasındadır ve uzamsal çözünürlük 0,5 m'dir. Ölçek değişmez özellik dönüşümü (Scale-Invariant Feature Transform, SIFT) ve destek vektör makinesi (Support-vector machine SVM) kullanarak sınıflandırma işlemi gerçekleştirmişlerdir %96.0 duyarlılık sonucu elde etmişlerdir.

Chen *et al.* (2018) çalışmalarında, Daha Hızlı RCNN (Faster R-CNN) çerçevesini iyileştirmek için Region Locating Network (RLN) adlı bir bölge önerisi stratejisi geliştirmişlerdir. Önerilen RLN, pistin ve park önlüğünün parçaları gibi, uçakların genellikle bulunduğu alanları tespit etmektedir. Kullanılan veri kümesinin tamamı 12 farklı havaalanının 265 görüntüsünü ve toplamda 6.000'den fazla açıklamalı uçağı içeriyor. Her resim, Google Earth'ten indirilen, havaalanının tamamı veya büyük bir kısmının anlık görüntüsüdür. Orijinal uzaktan algılama görüntülerinin boyutları  $2.048 \times 2.048 - 40.000 \times 30.000$  arasındadır. Eğitim verilerinin miktarını artırmak için, eğitim setindeki görüntüleri yatay ve dikey olarak ters çevirmişlerdir. Küçük görüntü bölgesindeki uçakları tespit etmek için önerdikleri Faster R-CNN'i kullanarak %53.64 doğruluk oranı elde etmişlerdir.

Yang *et al.* (2017) Markov rasgele alan tam katmanlı evrişimli ağ (M-FCN) adı verilen etkili uzaktan algılama görüntüleme bir uçak algılama çerçevesi önermişlerdir. M-FCN, FCN tabanlı bir kaba aday çıkarma aşaması, birçok Markov rastgele alanı tabanlı bölge teklif (RP) üretim aşaması ve bir son sınıflandırma aşamasından oluşmaktadır. Önerdikleri yöntemde üç farklı veri seti kullanmışlardır. İlk veri seti, yaklaşık 900 piksel x 800 piksel boyutlarında 80 havaalanı görüntüsü içeren NWPU VHR-10 veri kümesidir 757 uçak içeren uçak sınıfı bölümü alınmıştır. İkinci veri seti Google Earth'ten elde edilen üç havaalanı görüntüsünü (Berlin, Tokyo ve Sydney) içerir. Üçüncü veri kümesi, yaklaşık 4000 uçak ve arka plan görüntüsü içeren ve 50 piksel x 50 piksel boyutlarında ve ayrıca Google Earth'ten elde edilen kendi veri setleridir. Deneyler neticesinde, M-FCN'nin en yüksek %97.5 doğruluğa ulaştığını gözlemlemişlerdir.

Luo and Shi (2016) uzaktan algılamalı görüntülerde uçakların verimli bir şekilde nasıl bulunacağı problemini ele almışlardır. Çalışmalarında uçak yer tespiti için basit ama etkili bir teknoloji olan nesne teklifini (Object Proposal) kullanmışlardır. Bu yöntemin temel amacı, büyük olasılıkla, ilgilenilen nesnelere içeren nispeten küçük sınırlayıcı kutular seti üretmektir. Kullandıkları yöntemde ilk önce uçakları içeren küçük bir dizi sınırlama kutusu nesne teklif algoritması tarafından üretilir. Daha sonra, uçakları tespit etmek için HOG özellikleri bir SVM sınıflandırıcısı ile eğitilir. Son olarak, eğitilmiş nesne detektörü, algılama görevini tamamlamak için kapsamlı arama yerine sınırlayıcı kutulara uygulanır. Oryantasyondaki ve boyuttaki değişiklikler için, eğitim numunelerini 32 oryantasyona ve aynı boyuta (örn. 40×40) ayırmışlardır. Her yönelimde bir SVM öğrenilir. Daha sonra, tüm 32 sınıflandırıcıda olumsuz cevap varsa, ilk aşamadan oluşturulan teklifler kaldırılmıştır. Test verileri google earth'ten toplanmış 20 havaalanı görüntüsüdür. Görüntü boyutu 1000 × 1000 ile 2000 × 2000 arasında değişmektedir. Deneyler sonucunda %80 tespit doğruluğuna ulaşmışlardır.

Yang *et al.* (2018) yüksek çözünürlüklü optik pankromatik görüntülerde uçak algılama problemini araştırmışlardır. İlk olarak LARK (locally adaptive regression kernel)'lar hedef görüntüden çıkarılır ve hiperspektral formda üç boyutlu bir veri küpü oluşturur. Hesaplanan özellik vektörleri arasındaki benzerliği ölçmek için, özellik vektörleri hiperspektral bir perspektifte ele alınır ve hiperspektral tespit algoritması CEM (constraint energy minimization) kullanılır. Konvansiyonel benzerlik ölçümlerinin aksine, CEM, ilgilenilen nesnelere (yani, uçaklar) korunurken, arka plan çıktısının sıkıştırıldığı, yansıtılan bir yön bulmaya çalışır. Bu algoritma, arka plan istatistikleri de dahil olmak üzere hedef görüntünün genel bilgilerini analiz eder. Veri kümeleri, Google-Earth yazılımından kesilen 50 pankromatik uzaktan algılama görüntüsü içerir. Görüntülerin boyutları yaklaşık 5000 × 5000 ve çözünürlükler 0,5m ile 1m arasında değişiyor. 10 görüntüde özel şekilli uçaklar yer alırken, diğer 40 görüntüde genel uçaklar yer alan havaalanı sahneleridir. Uçaklar farklı yönlerde ve farklı şekillerdedir ve boyutları 30×30 ile 100×100 arasındadır.

Liu *et al.* (2017) uçakları uzaktan algılayan görüntülerden otomatik olarak bölütlere ayırmak için Single Shot MultiBox Detector (SSD) ve Tek katmanlı Hücreli Otomatlar (Single-layer Cellular Automata ,SCA)'yı birleştiren yeni bir yöntem önermişlerdir. İlk olarak SSD modeli ile her uçağın bir sınırlayıcı kutu ile ifade edilen yerlerde algılama sonucu elde edilir. Uzaktan algılama görüntüsünde küçük nesnelerin bulunabilmesi için orijinal SSD'deki son iki geleneksel katman kaldırılmıştır. Daha sonra sonuç piksel tabanlı bölütlemeye elde etmek için tek katmanlı SCA aracılığıyla işlenir. Veri kümesi,  $1280 \times 659$  boyutunda 1000 görüntüye sahiptir. Orijinal SSD için, görüntüler  $300 \times 300$  veya  $500 \times 500$  olarak yeniden şekillendirilmiştir. Veri setini 900 imgeli eğitim setine ve kalanlar test setine ayrılmıştır. Yaptıkları deneyler, algoritmanın veri setinin pikselini etiketlemeden iyi sonuçlar elde edebileceğini göstermektedir.

Kharchenko and Chyrka (2018) insansız hava araçlarından veya uydulardan alınan hava görüntülerinde uçakların tespitinde en gelişmiş nesne tespit yöntemlerinin performansını değerlendirmişlerdir. Performans için “Sadece Bir Kere Bakın” (YOLO) yaklaşımına dayanan iki popüler tek aşamalı sinir ağı YOLO v.3 ve Tiny YOLO v.3 kullanmışlardır. Eğitim veri setinde toplam 203 uçak içeren 50 görüntüden oluşan yeni bir veri seti kullanmışlardır. Her iki eğitilmiş sinir ağı da  $224 \times 224$  ile  $608 \times 608$  arasında farklı boyutlarda giriş görüntüsü ile test etmişlerdir ve en yüksek %90.73 doğruluk elde etmişlerdir.

Li *et al.* (2016) yüksek çözünürlüklü uzaktan algılama görüntülerinden gelen uçakları tespit etmek için Destek Vektör Makinesi (SVM) kullanan otomatik bir sistem sunmuşlardır. Veri tabanını, pek çok ünlü uluslararası şehirdeki 20 havaalanından google-earth'den farklı dönemlerde toplanan 40 görüntü oluşturmaktadır. Eğitim seti olarak 15 resim ve 8124 örnek test seti olarak 25 görüntü 150303 örnek kullanmışlardır. Veri tabanında bulunan uçaklar çok bulanık ve arka planları karmaşıktır. Uçak adaylarını hızlı bir şekilde bulmak için istatistiksel bölge birleştirme ve yol bölütlemesine kullanmışlardır. Kesin algılama işlemi, SVM sınıflandırıcısını arka plan bölütlemesi ve yol bölütlemesi özelliklerini eğiterek gerçekleştirmişlerdir ve %92.4 doğruluk elde etmişlerdir.

(Qiu, Wen, & Fan, 2017) uzaktan algılama görüntülerinde kısmen görünür uçak tespiti için katmanlı bir nesne tespiti önermişlerdir. Model, uzaktan algılamalı görüntülerde bulunan uçakların çoğunlukla kısmen görülebilir olması ve yaygın olarak kullanılan (Deformable Part Model, DPM) modellerinin performansının genellikle büyük ölçüde düşmesi sorununu hedeflemektedir. Google Earth'ten topladıkları uçak görüntülerini veri setini arttırmak için farklı yönlere elle döndürmüşlerdir. Örnekler birkaç ünlü Kuzey Amerika uluslararası havaalanından alınmıştır. Uçak içermeyen örnekler iki gruptan oluşuyor, birinde hava limanlarında hiçbir uçak bulunmayan alanlarından rastgele seçiliyor, diğerinde ise (LULC) veri setinden uçak sınıfı çıkarılıyor. Görüntüler, çoğunlukla okyanusa yakın alanda yer alan ve bulutlarla tıkanma olasılığı daha yüksek olan uluslararası havaalanlarından olan ortalama 0,5 m'lik bir uzamsal çözünürlüğe sahip. Önerdikleri yöntemin doğruluğu karşılaştırdıkları diğer yöntemlere en yüksek değer olup %92.57'dir .

J. Yang *et al.* (2018) Derin Artık Ağ (ResNet) ve Süper Vektör (SV) kodlamasına dayanan uzaktan algılama görüntülerinde uçak tespiti için yeni bir yöntem geliştirmişlerdir. İlk olarak, daha az katman içeren bir ResNet çeşidi öznetelik haritasının çözünürlüğünü artırmak için tasarlanmıştır ve çok düzeyli evrişimsel öznetelikler, bölge önerisi için birleştirilmiştir. Bu arada, nesne sınıflandırmasını tamamlamak için evrişimsel özneteliklere yardımcı olan ilgili her bölgeden SV kodlamasıyla yönlendirilmiş gradyan (HOG) histogramını çıkarmışlardır. Önerdikleri metodu kendi uzaktan algılama veri setleri üzerinde kapsamlı bir şekilde değerlendirmişlerdir. Deneysel sonuçlar, ResNet varyantına dayalı modellerinin %90.3'lük bir kesinlik ve %86.8'lik bir duyarlılık sahip olduğunu göstermiştir .

Hui Wu *et al.* (2016) uydu görüntülerindeki uçakları hızlıca tespit etmek için BING ve CNN'e dayanan bir yapı önerdi. CNN, ham görüntülerden öznetelikleri öğrenebilir ve küçük döndürme ve kaydırmaya karşı değişmez. BING tekniğine dayalı aday seçme yöntemi, tahmin için kayan pencerelerden daha az sayıda aday üretir ve bu sayede dedektörün hesaplama süresi azalır. Önerilen yöntemi, Google Earth'ten alınan uçak



tespit için kendi veri kümeleri üzerinde değerlendirdiler. Deneyle, yöntemlerinin uçak tespitinde iyi bir performans sağladığını gösterdi.

Hui Wu *et al.* (2015) uçak tespit görevindeki performanslarını geleneksel yöntemlerle (el yapımı öznitelikler sığ bir sınıflayıcı ile) karşılaştırarak iki tür derin öğrenme yöntemi, derin evrimsel sinir ağı (DNN) ve derin inanç ağı (DBN) uyguladı. Bu yöntemler daha iyi bir performans elde etmek için çok sayıda eğitim örneğinden sağlam öznitelikleri öğrenir. Deneyle, her iki derin öğrenme yönteminin de geleneksel yöntemlerin (HOG, LBP + SVM) yanlış tespit oranının en az %40'ını azalttığını ve DNN'nin DBN'den biraz daha iyi performans gösterdiğini göstermiştir.

Yu *et al* (2018) Tam Katmanlı Evrimsel Sinir Ağına (FCNN) dayanarak uçağın tespiti için bir yöntem önerdi. FCNN, çoklu görev kaybını en aza indirerek uçağın yerini hızlı ve doğrudan alabilir. Veri artırma ve transfer öğrenmesi sayesinde, FCN'nin sınıflandırma doğruluğu daha da geliştirilmiştir. Küçük hedefleri tanımak için, test görüntülerinde görüntü piramit yapısını oluşturarak çözünürlük bilgilerini uçaktaki önceden edinilen bilgilerle birleştirirler. Deneysel sonuçlar, daha az parametrelili yöntemin daha yüksek doğruluğa sahip olduğunu ve modelin eğitilmesinin basit olduğunu göstermiştir.

Chen *et al.* (2017) uzaktan algılama görüntülerinde uçak algılama için derin öğrenmeye dayalı nesne algılama teknolojisini kullanmışlardır. Uzaktan algılama görüntülerinin bazı özelliklerini kullanmaya ek olarak, bazı yeni veri artırma teknikleri önermişlerdir. Giriş için büyük boyutlu uzaktan algılama görüntüleri ile uğraşırken, görüntünün doğrudan yeniden boyutlandırılmasından kaynaklanan küçük nesne kaybı sorununu etkin bir şekilde çözen MultiBlock ve MapBlock katmanlarını önermişlerdir. Transfer öğrenme (VGG-16) ile uçtan uca eğitilebilir uçak algılamasını uygulamak için tek bir derin evrimsel sinir ağı ve sınırlı eğitim örnekleri kullanmışlardır. Google Earth'ten topladıkları havaalanlarının uzaktan algılama görüntüleri 581 imgeden oluşmaktadır. Önerdikleri yöntem, farklı katmanlardan özellik haritalarını hızlı algılama hızı ve yüksek doğrulukla birleştiren tek bir evrimsel sinir ağı olan SSD'ye dayanmaktadır. SSD yöntemi ile %86.28

bulunan tespit doğru doğruluğu önerdikleri yöntemle %96.23 bulunarak önerilen uçak algılama algoritmasının iyi algılama performansına sahip olduğunu göstermiştir.

Yu *et al.* (2015) uçakları yüksek çözünürlüklü uydu görüntülerinden saptamak için dönme ve ölçeğe göre değişmeyen bir yöntem önermişlerdir. Yerel görüntüler için yüksek dereceli özellik sunumlarını geliştirmek için, ortak eğitilmiş iki katlı Gaussian-Bernoulli DBM modelinden türetilmiş çok katmanlı bir özellik oluşturma modeli oluşturmuşlardır. Uçak merkezlerini etkin bir şekilde tahmin etmek ve rastgele yönelimleri ve boyutları olan uçakları ele almak için, gömülü yama oryantasyonlu bir Hough ormanı modeli ve uçak merkezlerinin olası yerleri hakkında olasılıklı oylar almak için ölçek faktörleri oluşturmuşlardır. Bu çalışmalarında, Google Earth'ten kendi görüntü veri kümelerini toplamışlardır. Eğitim aşamasında, farklı oryantasyonlara sahip sekiz farklı havalimanından 350 uçak örneği ayrıca doğal sahnelerin 350 negatif örneğini de toplamışlardır. Tespit aşamasında, on iki farklı havalimanından önerilen uçak tespit yöntemini test etmek ve değerlendirmek için 100 adet havaalanı sahnesi görüntüsü toplamışlardır. Her görüntü, farklı oryantasyonlarda ve boyutlarda birden fazla uçak örneği içerir. Eğitim ve test aşamalarında kullanılan resimler, 0,27 m çözünürlüğe sahiptir. %97.2 tespit doğruluğu elde etmişlerdir.

Yokoya and Iwasaki (2015) yüksek çözünürlüklü optik uzaktan algılama görüntülerinde bir nesne sınıfının veya belirli bir nesnenin örneklerini tespit etmek için yeni bir yöntem sunmuşlardır. Sınıf nesnelere veya belirli bir nesnenin parçaları, öğrenilen hedef ve arka planlar kullanarak her (patch) yamanın seyrek temsili ile tespit edilmiştir. Bir parça algılandığında, nesnenin merkezi Hough dönüşüm çerçevesi içinde aktive edilmiş, böylece parçaların bir arada oluşu nesne tespiti için kullanılmıştır. Önerilen yöntem, araba, tekne ve uçak tespitinde denenmiş uçak tespiti için çalışma görüntüleri, New Chitose Havaalanı, Hokkaido, Japonya ve Los Angeles Havaalanı, CA, ABD üzerinden GeoEye-1 tarafından pankromatik kanalda 0,5 m GSD ile alınmış 55 uçaktan oluşmaktadır. %91.8 uçak tespiti doğruluğu elde etmişlerdir.

X. Li and Wang (2017) evrişimli sinir ađlarını (CNN) iyi bir Őekilde kullanan bir nesne algılama yöntemi önermişlerdir. İlk olarak nesnelere içerebilecek kaba aday bölgeleri önermişlerdir. Sonraki adımda, iyi aday bölgeleri kaba aday bölgelerden kırpılmış ve nesnelere veya arka planlar olarak sınıflandırmışlardır. Daha az sayıda aday bölge önerilebilecek ve daha fazla ayırt edici özellik çıkarabilen kısa ve verimli bir framework (çerçeve) tasarlamışlardır. Çerçeve, iki sekiz katmanlı CNN'den oluşmaktadır. Kıyı gemilerini tespit etmek için CNN'leri kullanmak için, her biri yalnızca bir gemi içermesi gereken görüntü örnekleri gerekir. Bununla birlikte, geleneksel görüntü kırma yöntemi bu tür örnekleri üretemez. Bu sorunu çözmek için, dikdörtgen numuneler yerine yamuk üretebilen, CNN tarafından kıyıya doğru gemi tespitini mümkün kılan, oryantsiyonsuz bir görüntü kırma yöntemi sunmuşlardır. Google Earth'ten 650 eğitim ve  $2048 \times 1536$  piksel boyutunda 130 test görüntüsü toplamışlar ve %96,9 duyarlılık sonucu elde etmişlerdir.

Polat ve Yıldız (2012) Google Earth'ten elde edilen uydu görüntülerinde sabit uçakları tespit eden öğrenmeye dayalı bir sistem önerdi. Bir uçağın geometrik yapısını vurgulayan özellikler 2D Gabor filtresi kullanılarak belirlendi. Destek Vektör Makineleri (SVM) sınıflandırma metodu kullanılarak tespit gerçekleştirildi. Bir SVM eğitim algoritması, her biri iki kategorideki uçaklar veya uçaklara ait olmayanlar olarak işaretlenmiş bir dizi eğitim örneğinden (Google Earth programını kullanarak havaalanlarından toplanan uçak görüntüleri) matematiksel bir model oluşturuldu. Sistemin tespit performansı, Avrupa ve Amerika Birleşik Devletleri'ndeki tanınmış havaalanlarından alınan görüntüler kullanılarak gösterildi. 58 resim içeren veri setinde %91 oranında tespit elde edildi.

Xu et al. (2018) örneklemeden sonra aynı ölçekte farklı katmanlardan özellikleri birleştiren bir uçak tespit yöntemi önerdi. Evrişimli katmanlar, farklı büyüklükteki girdi görüntüleri için uygun olan ađ parametrelerini azaltmak için Tam Bağlı Katmanları değiştirmiştir. Önerilen yöntemlerinin Fast R-CNN ve Faster R-CNN gibi farklı karşılaştırmalı yöntemlerle karşılaştırılmasını sağladılar ve önerilen yöntem 125 görüntü içeren veri setinde %95.5 tespit oranı elde etti.

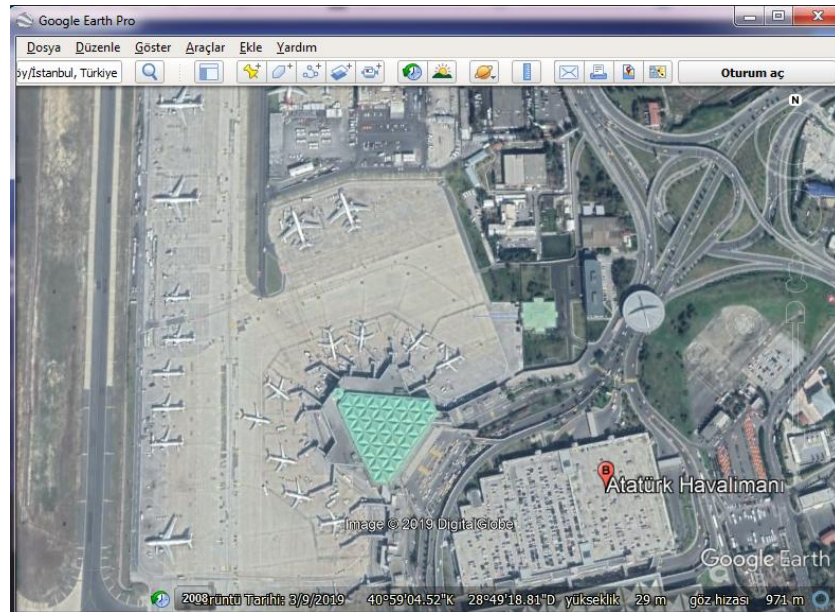
Mohammadi (2018) öneri oluřturma, derin sınıflandırma ve maksimum olmayan baskılama olmak üzere üç ana modülden oluřan uçak tespit algoritması sundu. NWPU VHR-10 veri setindeki deneyleri, yöntemlerinin çok yüksek çözünürlüklü görüntülerde otomatik uçak tespiti için umut verici bir yaklaşım olduğunu gösterdi, %97.3 Ap elde etti. Dahası, önerilen algoritma ekstra bir başarı olarak kutu düzeyinde ek açıklamalar kullanarak uçakların yönünü tahmin edebilir.



### 3. MATERYAL ve YÖNTEM

Google popüler bir arama motoru olmasının yanında kullanıcıya çeşitli alanlarda hizmet verebilecek ürünlere sahiptir. Bunlardan biri de coğrafi bilgi hizmeti ile Google Earth uygulamasıdır. Google Earth (Google 2001), öncelikle uydu görüntülerine dayalı olarak dünya'nın üç boyutlu temsilini sağlayan bir bilgisayar programıdır. Kullanıcılar adresleri ve koordinatları girerek ya da bir fare kullanarak dünyayı keşfedebilirler. Google Earth Geçmişe ait keşiflerin dış mekân görüntülerine, NASA uydu görüntülerine ve gemi tabanları dahil okyanus tabanının 3 boyutlu görüntülerinin erişimini de içeren temel özelliklere sahiptir. Google Earth Pro masaüstü uygulaması olup daha fazla ayrıntıyla hizmet vermektedir. Her iki uygulama da ücretsizdir.

Google Earth, bazıları tarafından gizliliğe ve ulusal güvenliğe yönelik bir tehdit olarak görülmüştür, bu da programın birçok ülkede yasaklanmasına yol açmıştır. Bazı ülkeler Google'ın uydu görüntülerinde, genellikle askeri tesisleri içeren alanlarda belirli alanların gizlenmesini talep etmiştir. Şekil 3.1'de de Google Earth Pro'nun arayüzü gösterilmektedir.



Şekil 3.1. Google Earth Pro arayüzü

### 3.1. Veri Kümesi

Son zamanlarda yaygınlaşan derin öğrenme arařtırmalarında kullanılan veri kümelerinin büyüklükleri önemli rol oynamaktadır. Özellikle ImageNet (ImageNet 2016) ve MSCOCO (Lin *et al.* 2014) gibi büyük veri kümeleri nesne algılama ve sınıflandırma problemlerinin başarısında oldukça etkilidir. Ancak uydu görüntülerinde nesnelere algılamak için gereken veri kümeleri barındırdıkları görüntü sayısının az olmasından dolayı yetersiz olup, bu alandaki çalışmalarda en büyük engellerden birisi olarak karşımıza çıkmaktadır.

Son zamanlarda arařtırmacılar çalışmalarını uydu görüntülerin üzerine yoğunlařtırmış ve bu amaçla çeşitli veri setleri oluşturulmuştur. Bunlardan biri hava görüntülerinde nesne tespiti için büyük ölçekli bir veri kümesi olan DOTA'dır (Xia *et al.* 2018) . Bu veri seti uçak, helikopter, liman, gemi, yüzme havuzu, depolama tankı vb. 15 sınıftan oluşmaktadır. DOTA veri setine benzer olarak AID (Aerial Image) (Xia *et al.* 2017) , veri kümesi de bulunmaktadır. Bu veri tabanlarına paralel olarak, derin öğrenme algoritmalarının ihtiyaç duyduğu büyük veri miktarlarına yaklaşmak için çalışmada uydu görüntülerinden elde edilen, çoğunluğu sabit kanatlı sivil ve askeri hava taşıtlarından oluşan ve ATA-Plane olarak isimlendirilen yeni bir veri kümesi hazırlanmıştır. Aşağıdaki iki bölümde bu veri kümesi kısaca tanıtılmaktadır.

#### 3.1.1. Veri kümesinin oluşturulması

Google Earth Pro kullanılarak dünyanın farklı, birçok askeri hava üslerinden ve sivil havalimanlarından olmak üzere çeşitli türlerde ve çok sayıda, sabit askeri ve sivil uçaklar toplanarak veri kümesi ATA-Plane oluşturuldu. Google Earth Pro'da görüntü toplanırken dikkat edilmesi gereken en önemli nokta aynı konumlardan farklı yüksekliklerde birkaç görüntü alınmasıdır. Böylelikle modelin daha iyi eğitilmesi sağlanmıştır. Ayrıca veri kümesinin sayısına katkı sağlamak, farklı zamanlar içerisinde konumlardaki uçak varlığının olup olmaması ve değişen hava koşulları göz önünde tutularak görüntüler elde edilmeye çalışılmıştır. Veri kümesi, 256x256 boyutlarında 2401 adet görüntü eğitim için,

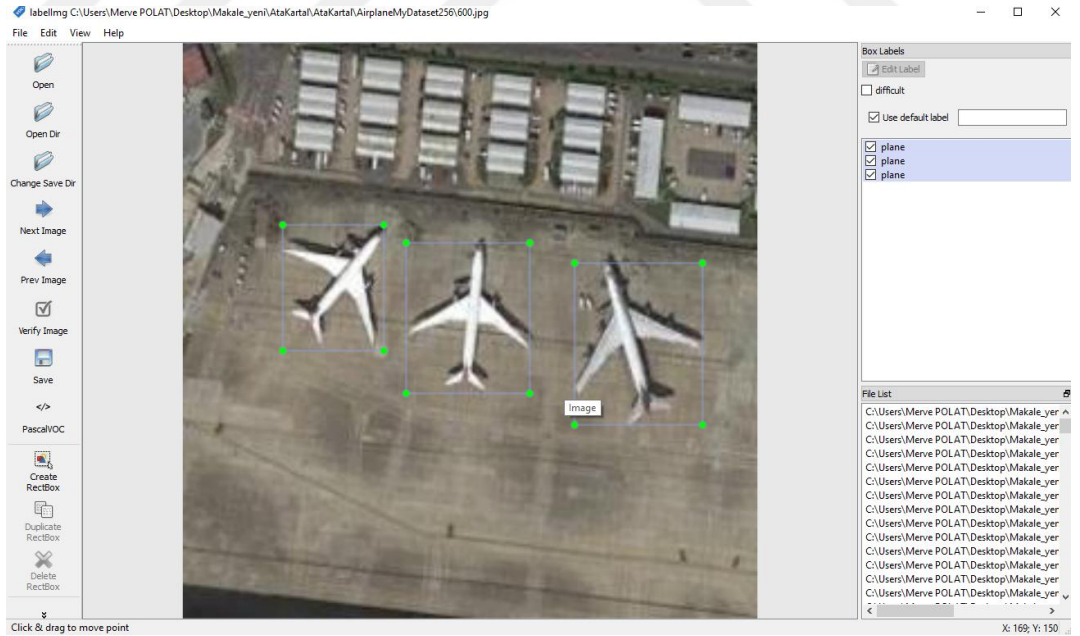
304 adet görüntü ise test için ayrılmış olan toplamda 2705 adet görüntüden oluşmaktadır. Veri kümesindeki görüntülerin çoğu birden fazla hedef nesne içermekte olup, görüntülerdeki hedef nesne sayısı yaklaşık 8584 adet olarak bulunmuştur. Şekil 3.2'de ATA-Plane veri kümesine ait bazı örnekler sunulmuştur.



**Şekil 3.2.** ATA-Plane veri kümesinin örnekleri

### 3.1.2. Veri kümesinin etiketlenmesi

Veri kümesinin her bir görüntüsü içerisinde hedef nesnelere çevreleyen bir sınırlayıcı kutu (bounding box) elle yerleştirilerek etiketleme işlemi gerçekleştirilmektedir. Bu işlem için, LabelImg (Windows\_1.8.0) (Anonim 2017) programı kullanılmaktadır. Sınırlayıcı kutular, LabelImg programında kullanılan yatay ve dikey bileşenlerin sırasıyla minimum ve maksimum değerlerinden oluşan  $[x_{min} \ y_{min} \ x_{max} \ y_{max}]$  formatı kullanılarak tanımlanmaktadır. Şekil 3.3'te LabelImg programının arayüzü gösterilmiştir. Program veri kümesinin bulunduğu klasör ile etiketlendikten sonra kaydedileceği klasör seçilerek etiketleme işlemi başlamaktadır. Programda PascalVOC ve YOLO olmak üzere iki farklı format bulunmaktadır. PascalVOC seçili durumda iken .xml uzantılı sınırlayıcı kutu bilgileri elde edilirken, YOLO seçilmesi halinde .txt uzantılı konum bilgisi ve sınıf bilgisini içeren ayrı text dosyaları elde edilmektedir. Kullanılan program ve algoritmaya uygun olarak PascalVOC seçilerek .xml uzantılı etiketleme bilgileri elde edilmektedir.



Şekil 3.3. LabelImg programının arayüzü

Şekil 3.4'te ATA-Plane veri kümesinin bazı etiketli örnekleri gösterilmektedir.





**Şekil 3.4.** Etiketlenmiş görüntülerden örnekler

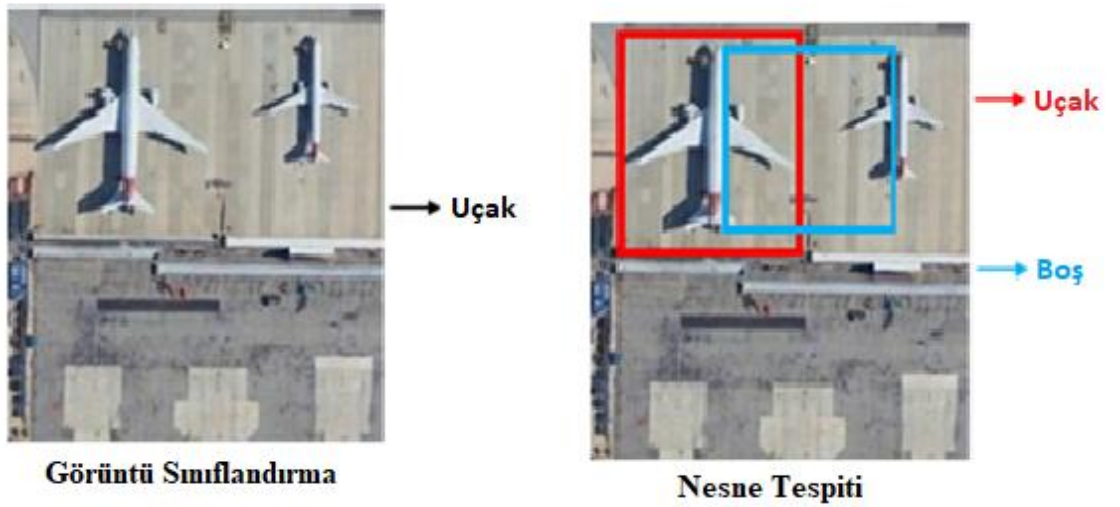
### 3.2. Nesne Tespiti Algoritmaları

Sınıflandırma, bir uyarıcıyı sonlu sınıflar veya etiketler grubunda kategorize etme işlemidir. Bu işlem normal olarak bir sahnedeki baskın içeriğin tanınmasını kapsar. Baskın içerik, ölçekleme, konum veya döndürme gibi içeriklerin dönüşümüne bakılmaksızın en güçlü güven puanını alır. Örneğin bir sınıflandırma probleminde, bir köpeğin veya başka bir şeyin görüntüsü verildiğinde, hangi baskın içeriğin olduğu

bilinmek istenmektedir. Bu nedenle, bir sınıflandırma sistemi, buradaki baskın içerik olduğu sürece, köpeğin görüntüdeki yeri ne olursa olsun, bu görüntüyü her zaman “köpek” olarak etiketlemektedir. Köpek artık baskın içerik değilse, sistem görüntünün etiketini bir sonraki baskın içeriğe değiştirmelidir. Sınıflandırmada amaç görüntü içeriğinin değişmez bir şekilde tanınmasıdır.

Nesne tespiti problemi ise nesnelerin nerede ve kaçar tane oldukları ile ilgilidir. Nesnelerin konumunu belirleme işlemine yerleştirme denir. Nesne tespiti, hem sınıflandırmayı hem de yerleştirmeyi içeren bir süreçtir. Bir nesne algılama sistemi, sahnedeki bilinen tüm içeriği sınıflandırmak ve bulmak için görevlendirilmiştir. Örneğin, bir yaya tespit sistemi yayaları tanımalı ve tanınan yayaların her birinin konumunu belirtmelidir.

Sınıflandırma görevinde belirli bir hedef, görüntüde sürekli hareket ettikçe, ağ hala bir nesneyi karşılık gelen kategoriye göre doğru bir şekilde sınıflandırabilir. Ancak, algılama görevinde hedef hareket ettikçe, ağ doğru şekilde algılayamayabilir. Bu, hedefin yerin hareketine duyarlı olduğu anlamına gelir.



Şekil 3.5. Görüntü sınıflandırma ve nesne tespiti

Şekil 3.5'te görüntünün sol tarafında gösterildiği gibi, uçak görüntüde hareket etse de, sınıflandırma ağı hala onu bir uçak olarak doğru şekilde sınıflandırır. Ancak nesne tespitinde, uçak görüntü içinde hareket ettiğinde boş olarak algılanır. Bu durum sınıflandırma ağının konum duyarsızlığına ve tespit ağının konum hassasiyetine karşılık gelmektedir.

Derin öğrenme yöntemlerinin bilgisayar görme problemlerine uygulanmasından önce, otomatik nesne tespiti sistemleri esas olarak, yönlendirilmiş gradyanların histogramları (HOG) veya daha sonra SVM tarafından sınıflandırılan ölçek değişmez özellik dönüşümü (scale-invariant feature transform, SIFT) gibi özellik tanımlayıcılarını çıkarmaya dayanıyordu. HOG özellikleri, bir görüntüyü birkaç örtüşen bloğa bölerek, her piksel için yoğunluk gradyanlarını hesaplayarak ve gradyan yönelimini birkaç bölmeyle ölçerek hesaplanır. Her blok için histogramlar daha sonra birleştirilir ve sınıflandırıcıya beslenir (Dalal and Triggs 2005). David G. Lowe tarafından geliştirilen SIFT, nesne tespiti yönteminde, özellikler etiketli görüntülerden çıkarılır ve bir veritabanında saklanır. Yeni bir resim işlenecek olduğunda, yeni resim özellikleri ile veri tabanındaki diğer resimler arasında bir eşleşme işlemi Öklid mesafesi kullanılarak gerçekleştirilir. Bunu takiben, belirli bir nesneden kümeleri tanımlamak için bir Hough dönüşümü gerçekleştirilir ve belirli bir özellik vektörünün görüntüdeki bir nesneyi temsil etme olasılığı hesaplanır. Doğrulama, affine dönüşümünden elde edilen parametrelere uygulanan en küçük kareler çözeltisi olarak gerçekleştirilir (Lowe 1999).

Krizhevsky *et al.* (2012) ILSVRC yarışmasını kazanması ile, sınıflandırma için CNN kullanımı bu alanda egemen olmuştur. Nesne tespiti için en ilkel CNN yaklaşımı, kayan pencere modeli olmuştur. Bu yöntemde gerçek görüntü boyutundan çok daha küçük bir pencere görüntü üzerinde sağa sola yukarı aşağı dolaştırılır, her bir pencere CNN'e aktarılır ve CNN'nin tahmin yapması sağlanır. Farklı büyüklükte pencere boyutu için tüm adımlar tekrarlanarak yeni elde edilmiş görüntüler CNN'e aktarılır ve yeni tahminler bulunur. Sonunda, nesnenin sınıfı ve sınırlayıcı kutusuyla birlikte bir nesneye sahip olacak bir dizi bölge elde edilir. Bu yöntemin dezavantajlarından biri birden fazla görüntüyü CNN'den geçirdiğinden, hesaplama maliyetinin fazla olmasıdır. Diğer bir

dezavantajı ise yanlış sınırlayıcı kutulardır. Görüntünün her yerinde kare şeklinde pencereler kaydırıldığından nesnenin kare dışında bir şekle sahip olması kayan pencere ile nesnenin gerçek boyutunun tam olarak bir uyumsuzluğuna sebebiyet verir. Her ne kadar bu algoritma bir görüntüdeki birden fazla nesneyi bulma ve yerleştirme yeteneğine sahip olsa da, sınırlama kutusunun doğruluğu kötüdür.

CNN tabanlı nesne tespiti modelleri, bölge tabanlı modeller ve regresyon tabanlı modeller olmak üzere başlıca iki yaklaşımdan oluşur. Bölgesel temelli yaklaşımlarda, ilk bölge önerileri Seçmeli Arama (Selective Search) yöntemi kullanılarak oluşturulur (Van *et al.* 2011). Bölgesel tabanlı modellerde sınıflandırma ve sınırlayıcı kutu tahmin işlemi, CNN kullanarak bu bölge önerilerine dayanarak gerçekleştirilir. Regresyon tabanlı modellerde ise, nesne kategorisi ve konum bilgisi doğrudan CNN kullanılarak tahmin edilir.

Nesneyi tüm piksel konumlarında ve tüm ölçeklerde arayan kayan pencere yaklaşımından farklı olarak, bölgesel tabanlı algoritmalar pikselleri daha az sayıda segment halinde gruplamaya çalışır. Dolayısıyla, üretilen son öneri sayısı kayan pencere yaklaşımından çok daha azdır. Oluşturulan bu bölge önerileri farklı ölçek ve en boy oranlarındadır.

Bölgesel tabanlı nesne tespiti algoritmalarında, nesne içerebilecek bölgeler bölütleme (segmentasyon) kullanılarak tanımlanır. Seçmeli Arama, nesne tespitinde kullanılan bir bölge öneri algoritmasıdır. Renk, doku, boyut ve şekil uyumluluğuna bağlı olarak benzer bölgelerin hiyerarşik gruplandırılmasına dayanır. Şekil 0.6'da Seçmeli Arama algoritmasının örneği gösterilmiştir.



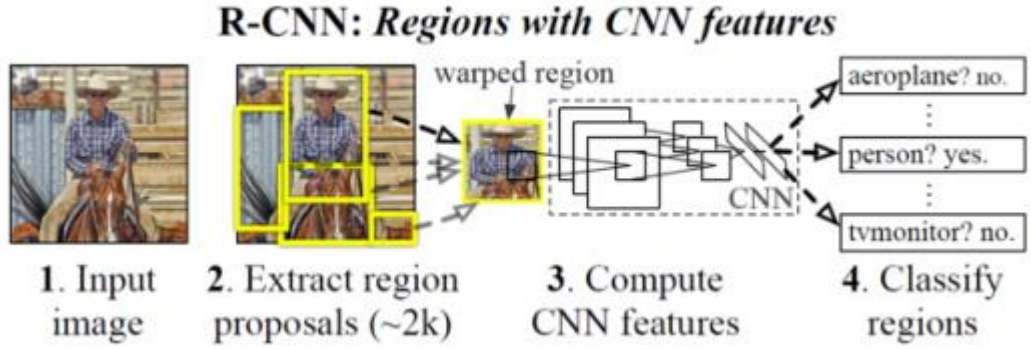
**Şekil 3.6.** Seçmeli Arama algoritmasının örneği (Van *et al.* 2011)

Şekil 3.6’da üstteki görüntü, düz renkler kullanılarak temsil edilen parçalı bölgeleri içermektedir.

Bu çalışmada uydu görüntülerinden elde edilmiş uçak görüntülerinin tespitinde bölgesel tabanlı algoritmalarından Faster R-CNN ve R-FCN, regresyon tabanlı algoritmalarından ise SSD yöntemi kullanılmıştır. Bu yöntemler ile ilgili ayrıntılı bilgi aşağıda verilmiştir.

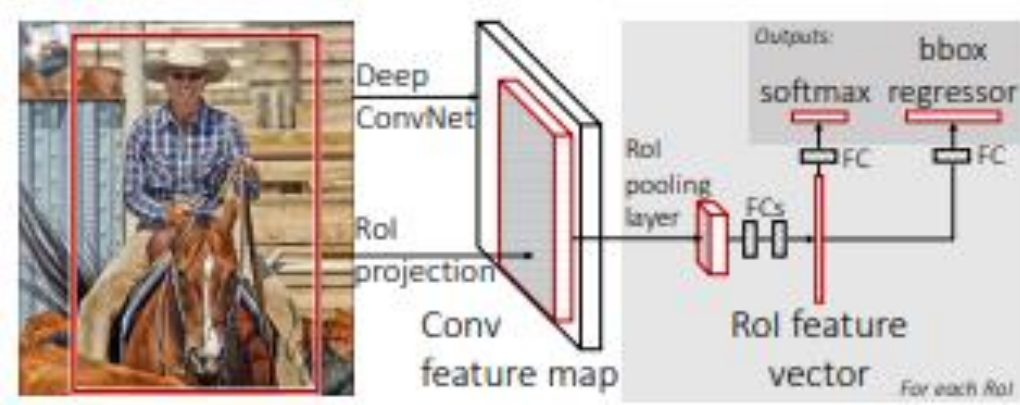
### 3.2.1. FASTER R-CNN

Bölgesel öneriye dayanan ilk nesne tespit modeli, Ross Girshick tarafından önerilen R-CNN’dir . İlk aşamada, Seçici Arama yöntemi ile yaklaşık 2000 bölge önerisi ve daha sonra ilgili bölgelerdeki CNN modeli ile özellikler çıkartılır. Her bir görüntü için İlgi Bölgesi (RoI) elde edilerek tüm bu bölgeler, CNN giriş boyutuna uyacak şekilde yeniden şekillendirilir. Bu yöntemde önceden eğitilmiş bir evrimsel sinir ağı ile ağın son katmanı, algılanması gereken sınıf sayısına göre yeniden eğitilir. Bölgeler elde edildikten sonra, her sınıf için nesnelere ve arka planı sınıflandırmak için SVM eğitilir. Son olarak, görüntüdeki her tanımlanmış nesne için sınırlayıcı kutuları oluşturmak üzere doğrusal bir regresyon modelini eğitilir.



**Şekil 3.7.** R-CNN mimarisi (Girshick *et al.* 2014)

Şekil 3.7’de gösterilen R-CNN yönteminin en büyük dezavantajı seçici aramaya dayalı olarak her resim için 2.000 bölgenin çıkarılması ve her resim bölgesi için CNN kullanarak özellikler çıkarılmasıdır. Özellik çıkarma için CNN, nesnelere tanımlamak için doğrusal SVM sınıflandırıcısı, sınırlayıcı kutuları belirlemek için regresyon modeli kullanılan bu yöntemde tüm bu işlemler RCNN’yi oldukça yavaşlatmaktadır. Her yeni görüntü için tahminler yapmak yaklaşık 40-50 saniye sürmektedir. Bu durum, modeli büyük bir veri kümesiyle karşılaştığında temelde hantal ve pratik olarak imkansız kılmaktadır. R-CNN’yi hızlandırmak için, Ross Girshick Fast R-CNN’yi önerdi (Girshick, 2015). Ross Girshick, CNN’i görüntü başına sadece bir kez çalıştırma ve bu hesaplamayı 2.000 bölgede paylaşmayı önermiştir. Fast RCNN’de giriş görüntüsü CNN’e verilir ve evrimsel özellik haritaları oluşturulur. Bu haritalar kullanılarak, öneri bölgeleri çıkarılır. Daha sonra, önerilen tüm bölgeleri sabit bir boyuta yeniden şekillendirmek için bir RoI havuzlama katmanı kullanılır. Ardından, her bölge tamamen bağlı bir ağa aktarılır. Çıkış sınıflarına tamamen bağlı olan ağın üzerinde bir softmax katmanı uygulanır. Softmax katmanı ile birlikte, öngörülen sınıflar için sınırlayıcı kutu koordinatlarını belirlemek için paralel olarak bir doğrusal regresyon katmanı da kullanılır. Bu nedenle, üç farklı model kullanmak yerine (RCNN’de olduğu gibi), Hızlı RCNN, bölgeleri farklı sınıflara ayıran ve aynı anda tanımlanmış sınıflar için sınır kutularını aynı anda ifade eden tek bir model kullanır.



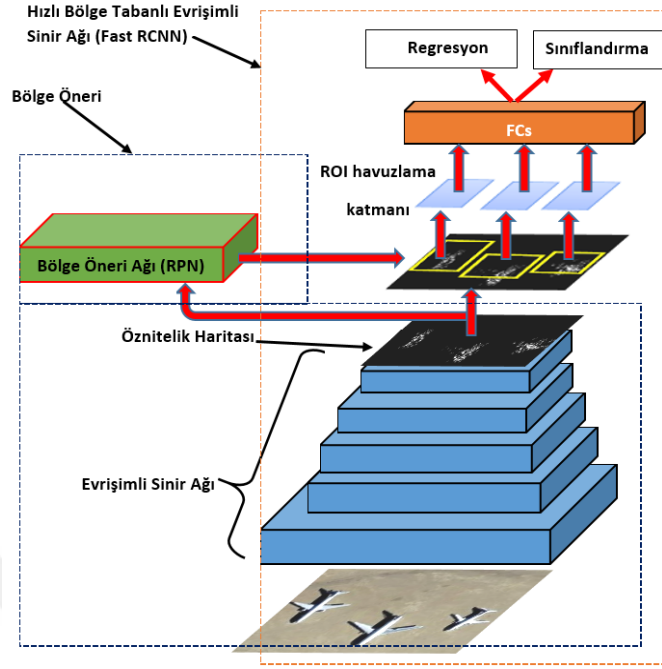
**Şekil 3.8.** Fast R-CNN mimarisi (Girshick 2015)

Ancak Şekil 3.8’de gösterilen Fast R-CNN’de de hala bir takım problemler mevcuttur. Fast R-CNN, yavaş ve zaman alan bir işlem olan İlgi Bölgelerini bulmak için bir bölge öneri yöntemi olarak seçici aramayı kullanmaktadır. RCNN’ye kıyasla nesnelere algılamakta çok daha iyi olan bu yöntemde görüntü başına tahminler yaklaşık 2 saniye sürmektedir. Ancak gerçek hayattaki büyük veri kümelerini göz önüne alındığında, bir Fast R-CNN bile artık çok hızlı bir performans sergileyememektedir.

Faster RCNN (Girshick 2015), Fast RCNN’nin değiştirilmiş versiyonudur. Aralarındaki en büyük fark, Fast RCNN’nin İlgi Bölgeleri oluşturmak için seçici bir arama yapması, Faster RCNN ise “Bölge Öneri Ağı”, yani RPN kullanmasıdır. RPN, görüntü özellik haritalarını bir girdi olarak alır ve her biri çıktı olarak bir nesne puanına sahip olan bir dizi nesne teklifi oluşturulur.

Bir görüntü girişi olarak alınıp, ConvNet’e iletilerek görüntünün özellik haritası elde edilir.

Bu özellik haritalarına bölge öneri ağı uygulanır. Bu ağ nesne önerilerini, nesnellik puanlarıyla birlikte döndürür. Tüm öneri bölgelerini aynı boyuta getirmek için bu bölgelere bir RoI havuz katmanı uygulanır. Son olarak, öneri bölgeleri, nesnelere için sınırlayıcı kutuları sınıflandırmak ve çıkarmak için, bir softmax katmanına ve bir doğrusal regresyon katmanına sahip olan tamamen bağlı bir katmana geçirilir.

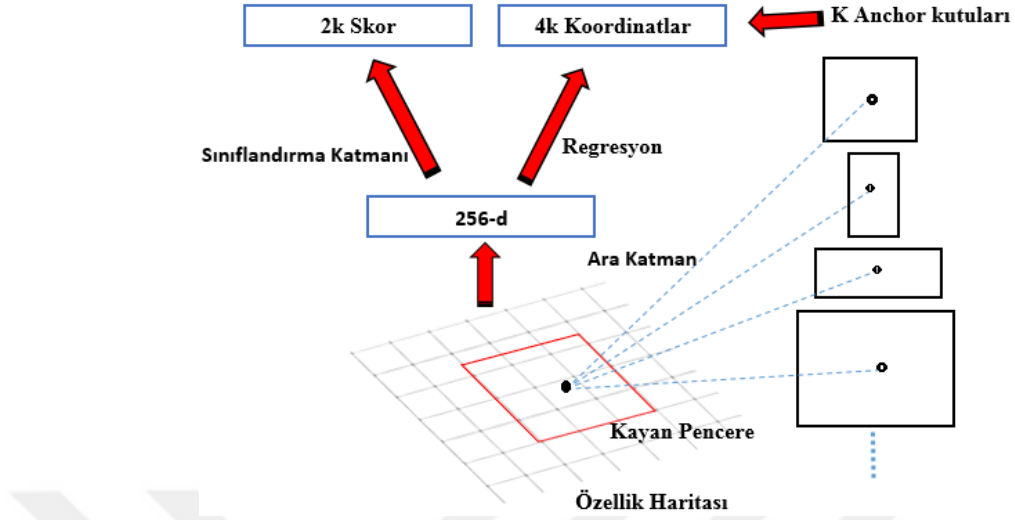


**Şekil 3.9.** Faster R-CNN mimarisi

Şekil 3.9’da gösterilen Faster RCNN, özellik haritalarını CNN’den alır ve bunları Bölge Öneri Ağı’na iletir. CNN ağında, bir sınır kutusunu ve nesnellik puanını tahmin etmek için 2 ayrı tamamen bağlı katmana beslenen 256 değer üretilir. RPN, bu özellik haritaları üzerinde kayan bir pencere kullanır ve her pencerede, farklı şekil ve boyutlarda  $k$  ankraj kutusu oluşturur. Bu nedenle RPN konum başına  $4 \times k$  koordinat ve  $2 \times k$  puan çıkarır.

Ankraj kutuları, görüntü boyunca yerleştirilmiş ve farklı şekil ve boyutlarda sabit boyutta sınır kutularıdır. Her ankraj için, RPN iki şeyi öngörür: İlki, ankrajın bir nesne olma olasılığıdır (nesnenin hangi sınıfa ait olduğu ile ilgilenmez). İkincisi, ankrajları nesneye daha iyi uyacak şekilde ayarlamak için sınırlayıcı kutu regresörüdür.



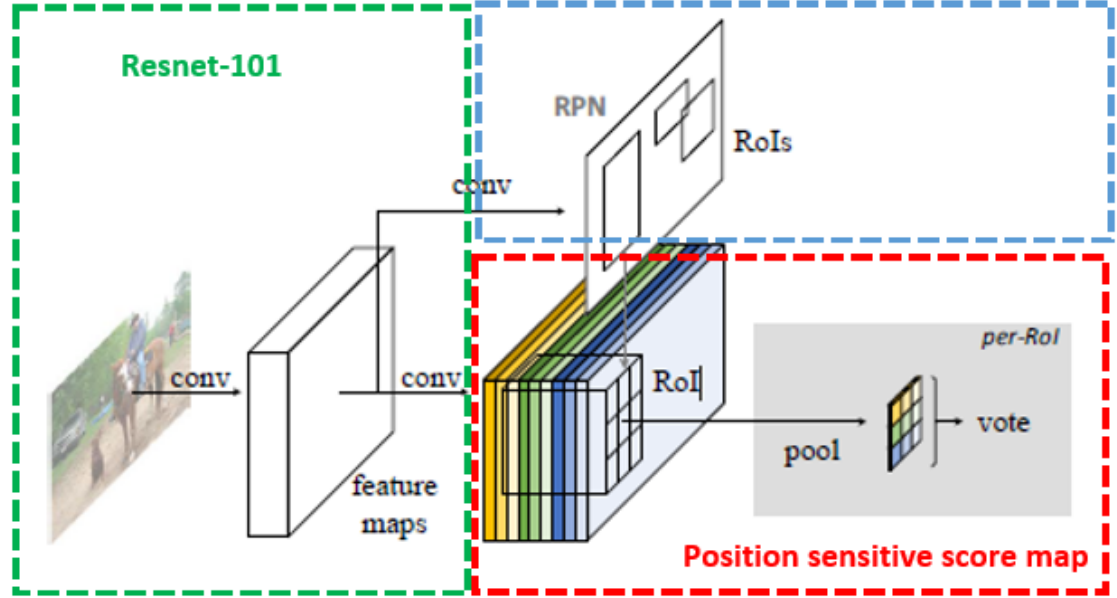


**Şekil 3.10.** Bölge öneri ağı (RPN)

Şekil 3.10'da gösterilen farklı şekil ve boyutlarda sınırlayıcı kutular, RoI havuz katmanına aktarılır. RPN adımından sonra, kendilerine atanmış sınıf bulunmayan bölge önerileri olabilir. Her bölge önerisi işleme alınabilir ve her öneri bir nesne içerecek şekilde kırılabilir. RoI havuzlama katmanının yaptığı bu işlem ile her ankraj için sabit boyutlu özellik haritaları çıkartılır. Daha sonra bu özellik haritaları softmax ve doğrusal bir regresyon katmanına sahip olan tamamen bağlı bir katmana geçirilerek nesne sınıflandırılır ve tanımlanan nesnelere için sınırlayıcı kutuları tahmin edilir.

### 3.2.2. R-FCN

Faster R-CNN'de, RPN tarafından oluşturulan bölge önerileri, RoI havuzlama katmanı tarafından işlenir ve anlaşılabilir olan sabit boyutlu bir özellik vektörü haline getirilir. Her RoI'nin FC katmanına ayrı ayrı bağlanması gerekmektedir. Bu yüzden ağ katmanını paylaşabilmek için, Dai ve arkadaşları tarafından R-FCN önerildi (Dai *et al.* 2016). R-FCN'de, orijinal FC katmanının hesaplanması evrişimli katmanda (ortak) hesaplanma ile değiştirilir, böylece nihai RoI doğrudan elde edilir.



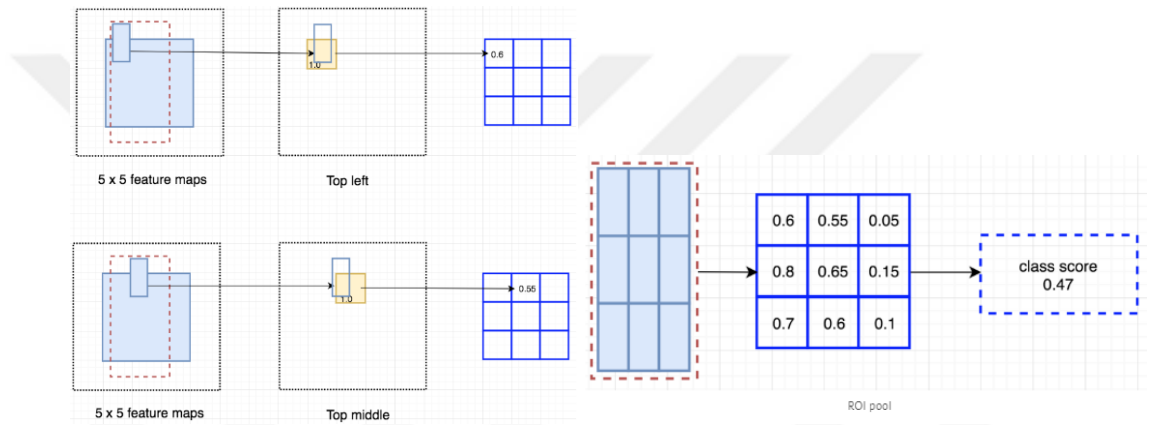
**Şekil 3.11.** R-FCN Mimarisi (Dai *et al.* 2016)

Şekil 3.11’de bir R-FCN yapısı gösterilmiştir. Yeşil kutudaki içerik ResNet-101 modelidir. Bu aşamada giriş görüntüsünden öznetelik haritası çıkartılır. Bu öznetelik haritaları RPN ve Pozisyon Duyarlı ağ tarafından kullanılmaktadır. Mavi kutu RPN’dir ve R-FCN’de, önerilen alanın ve sınırlama kutusunun çıktısından sorumludur bu yapı Faster R-CNN’deki ile aynıdır. Kırmızı kutudaki pozisyona duyarlı evrişim ve havuzlanma işlemi, Faster R-CNN’deki RoI havuzu + tam bağlantılı Katmanlar + çoklu görev kaybı işlevlerine karşılık gelmektedir. R-FCN’de, bu kısım pozisyona(konum) duyarlı evrişim ile değiştirilir ve RPN öneri kutusu, konuma duyarlı havuzlama için skor haritasının üzerine yerleştirilir.

### **Konum Duyarlı Havuzlama İşlemi**

RoI havuzlamanın bir çeşidi olan konum-duyarlı havuzlama, skor haritasındaki bir havuzlama işlemidir. Puan haritalarını ve RoI’leri oy dizisine eşleme işlemine, pozisyona duyarlı RoI havuzu adı verilir. Örneğin tespit edilecek C sınıf var ise bunlar  $C + 1$  sınıflarına genişletilir, böylece arka plan için yeni bir sınıf eklenmiş olur (nesne olmayan). RPN çıktısı olarak elde edilen bölge önerileri  $3 \times 3$  bölgelere ayrılarak her bölgenin nesne

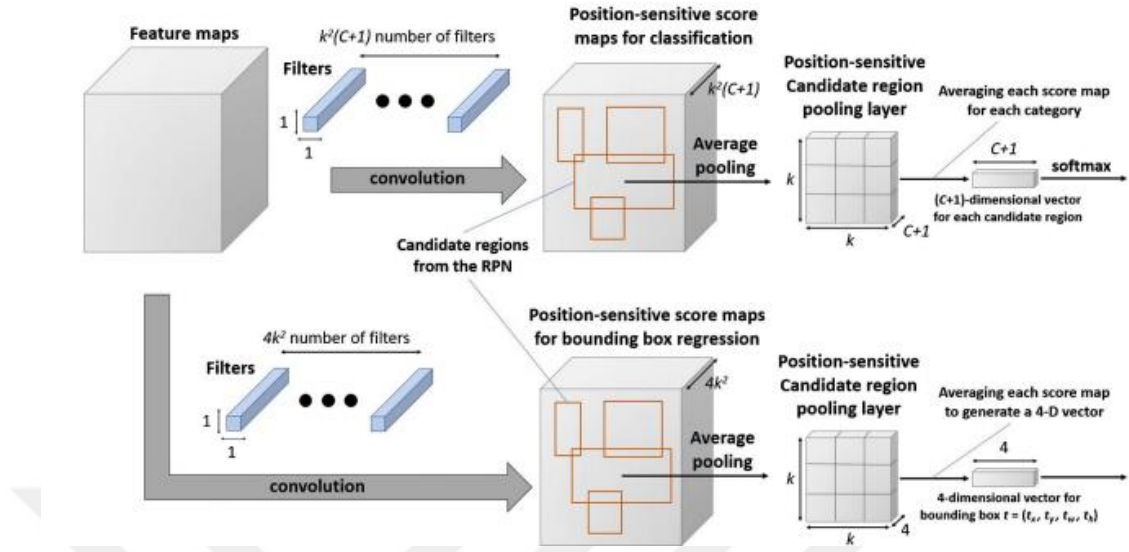
içerme olasılığı sorgulanmaktadır. Her sınıfın kendine ait  $3 \times 3$  puan haritaları ve dolayısıyla toplam  $(C + 1) \times 3 \times 3$  puan haritaları olmaktadır. Kendi skor haritaları setini kullanarak, her sınıf için bir sınıf puanı tahmin edilir. Şekil 3.11’de koyu sarı renkli kanalların sayısı  $C + 1$  değerine sahiptir. Sonra, konum duyarlı havuzlama işlemi gerçekleştirildiğinde, koyu sarı renkli kanal sol üst köşedeki değer olduğu sürece  $k \times k$  ızgarasının sol köşe konumu gösterir. Bu durum soldan sağa ve yukarıdan aşağıya devam etmektedir.



**Şekil 3.12.** Sınıf puanı sonuçları oluşturması (Hui 2018)

Sınıf puanı sonuçları Şekil 3.12’de sağdaki  $3 \times 3$  oy dizisine kaydedilir. Sonra her sınıfın olasılığını hesaplamak için bu puanlara softmax işlemi uygulanır.

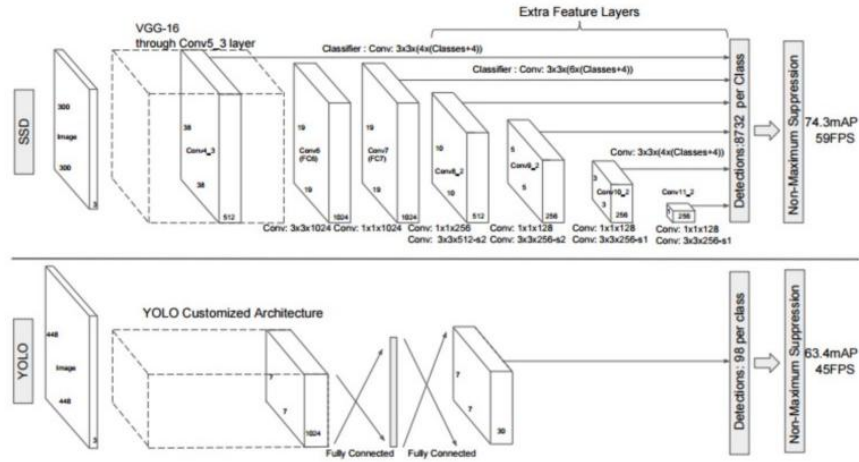
Sınıflandırmada  $k \times k \times (C + 1)$  puan haritalarını oluşturmak için evrişim filtreleri kullanılmıştı. Sınırlayıcı kutu regresyonunu gerçekleştirmek için bu mekanizma neredeyse aynıdır. Aynı özellik haritalarından bir  $4 \times k \times k$  haritası oluşturmak için başka bir evrişim filtresi kullanılır. Bir sınır kutusu içeren her elemanla bir  $k \times k$  dizisi hesaplamak için pozisyon tabanlı RoI havuzu uygulanır. Elde edilen son tahmin, bu öğelerin ortalamasıdır.



**Şekil 3.13.** R-FCN'de pozisyona duyarlı sınıflandırma ve sınırlayıcı kutu regresyon işlemi (Kim *et al.* 2017)

### 3.2.3. SSD

SSD (Single Shot MultiBox Detector) (W. Liu *et al.* 2016), özellikle gerçek zamanlı problemlerde nesne tespiti için hazırlanmış ve bu problemlerde çokça kullanılan önemli bir algoritmadır. Bu algoritmadaki anahtar fikir, tek ağ kullanılmasıdır ve bu yöntemde bölge önerilerine gerek yoktur, bunun yerine farklı sınırlayıcı kutular kullanılır ve sınırlayıcı kutu tahmininin bir parçası olarak ayarlanır. Aşamalı olarak daha küçük sınırlama kutusu için, tahminlerden sorumlu olan ağın son birkaç katmanının her biri tarafından farklı sınırlama kutusu tahminleri gerçekleştirilir ve son tahmin, tüm bu tahminlerin birleştirilmesinden oluşur. SSD, sınırlayıcı kutuların çıktı alanını, öznitelik haritası konumuna göre farklı en boy oranları ve ölçekler üzerinden bir dizi varsayılan kutuya ayırır. Tahmin zamanında, ağ, her varsayılan kutudaki her nesne kategorisinin varlığı için puanlar(skorlar) üretir ve nesne şeklini daha iyi eşleştirmek için kutuya ayarlamalar yapar.



Şekil 3.14. SSD Mimarisi (W. Liu *et al.* 2016)

SSD, eğitim sırasında her nesne için yalnızca bir giriş görüntüsüne ve kesin referans (ground truth) gerçek kutularına ihtiyaç duyar. Evrimsel bir biçimde, farklı ölçek oranlarına sahip birkaç öznetelik haritasındaki her bir konumdaki farklı boyut oranlarına sahip varsayılan kutuların küçük bir seti değerlendirilir. Her varsayılan kutu için, hem şekil ofsetlerini hem de tüm nesne kategorileri için güvenilirlikleri tahmin edilir. Eğitim sırasında ilk önce bu varsayılan kutuları kesin referans (ground truth) kutularıyla eşleştirilir. SSD yaklaşımı, sabit boyutta bir sınırlama kutuları koleksiyonu ve bu kutulardaki nesne sınıfı örneklerinin varlığı için puanlar (skorlar) üreten ileri beslemeli bir evrimsel ağa ve ardından son tespitleri üretmek için maksimum olmayan bir bastırma (non-maximum suppression) adımına dayanmaktadır (en üst üste binen sınırlayıcı kutular en yüksek puana sahip olanları saklar). İlk ağ katmanları, temel ağ olarak adlandırdığımız yüksek kaliteli görüntü sınıflandırma için kullanılan (herhangi bir sınıflandırma katmanından önce kesilmiş) standart bir mimariye dayanmaktadır. Daha sonra aşağıdaki temel özneteliklerle tespitler üretmek için ağa yardımcı yapı eklenir. Tespit için çok ölçekli öznetelik haritaları: Kısaltılmış ana ağın sonuna evrimsel öznetelik katmanları eklenir. Bu katmanlar boyut olarak aşamalı olarak azalır ve çoklu skalalarda tespit tahminlerine izin verir. Tespitlerin tahmin edilmesi için evrimsel model, her bir öznetelik katmanı için farklıdır.

Faster RCNN ile karşılaştırmak gerekirse, Faster RCNN bounding box oluşturarak bir bölge öneri ağı (regional proposal network-RPN) yardımıyla bölge önerileri tanımlar ve bu bölge önerilerini nesnelere sınıflandırmak için kullanır. Bu iki işlem basamağı için iki çekim gerekirken, SSD’de görüntüdeki birden fazla nesneyi tespit etmek için bölgesel önerileri ve özellik çıkarımını birleştiren tek bir derin sinir ağı kullanıldığından tek çekim (one single shot) gerekir ve bundan dolayı RPN tabanlı yaklaşımlara göre daha hızlı bir algoritma olduğu bilinmektedir.

SSD dedektörü, farklı ölçeklerdeki nesnelere daha hassas bir doğruluk sağlayan çoklu katmanlar kullanmaktadır.

Herhangi bir derin ağ tabanlı model (Resnet, Inception gibi) ile kullanılabilen SSD mimarisinin temeli VGG16’ya (Simonyan and Zisserman 2014) dayanır, ve ardından daha büyük nesnelere işlenmesinde yardımcı olacak bazı katmanlar eklenir.

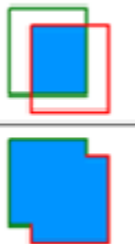
#### 4. ARAŞTIRMA BULGULARI

Eđitimli bir modelin deęerlendirilmesi, bir test setinin uygulanmasından sonraki sonuçlara dayanmaktadır. ‘‘Uçak’’ ve ‘‘Uçak deęil’’ řeklindeki iki sınıflı problemimizde, ATA-Plane veri kümesi, eđitim için 2401 ve geri kalan 304 test için kullanılan toplamda 2705 görüntü içermektedir. Tüm modellerin nesne algılama performansını deęerlendirmek için Precision-Recall eęrileri çıkarılmıştır. Ayrıca Jaccard indeksi gerçek ve tahmin edilen bölgeler arasındaki benzerlięi ölçmek için de kullanılır. Jaccard indeksi, bu gerçek ve tahmin kutular arasındaki Birleşimin Kesişime Oranı (Intersection over Union, IoU) temsil eder.

##### Intersection over Union (IoU)

Hedef nesnelerin tespitinin doęruluęunu ölçmek için kesin referans ile model tahmini arasındaki benzerlięi ölçmede Jaccard indeksi kullanılır. Bu amaçla, kesin referans ve model tahmini kutularını tanımlayan iki dikdörtgenin kesiştikleri alanın (intersection) bu iki dikdörtgenin bileşim (union) alanına bölümü büyüklüęü hesaplanır.

Şekil 4.1’te, IoU’yu kesin referans sınırlama kutusu (yeşil) ve tespit edilen bir sınırlama kutusu (kırmızı) arasında gösterir.

$$\text{IoU} = \frac{\text{kesişim alanı}}{\text{birleşim alanı}} = \frac{\text{Yeşil kutu} \cap \text{Kırmızı kutu}}{\text{Yeşil kutu} \cup \text{Kırmızı kutu}}$$


Şekil 4.1. Birleşimin Kesişime Oranı

Gerçek			
Pozitif	Negatif		
<i>TP</i>	<i>FP</i>	Pozitif	Tahmin
<i>FN</i>	<i>TN</i>	Negatif	

Metrikler tarafından kullanılan bazı temel kavramlar:

TP (True positive—Doğru Pozitif): Uçağa uçak demek.

FP (False positive—Yanlış Pozitif): Uçak olmayana uçak demek.

TN (True negative—Doğru Negatif): Uçak olmayana uçak değil demek.

FN (False negative—Yanlış Negatif): Uçak olana uçak değil demek.

### Precision-Recall

Precision-Recall (Kesinlik-Duyarlılık), eğrisi özellikle sınıflar çok dengesiz olduğunda tahmin başarısının faydalı bir ölçüsüdür. Precision, sonuç alaka düzeyinin bir ölçüsüdür, Recall ise gerçekten ilgili sonuçların ne kadarının doğru döndürüldüğünün bir ölçüsüdür.

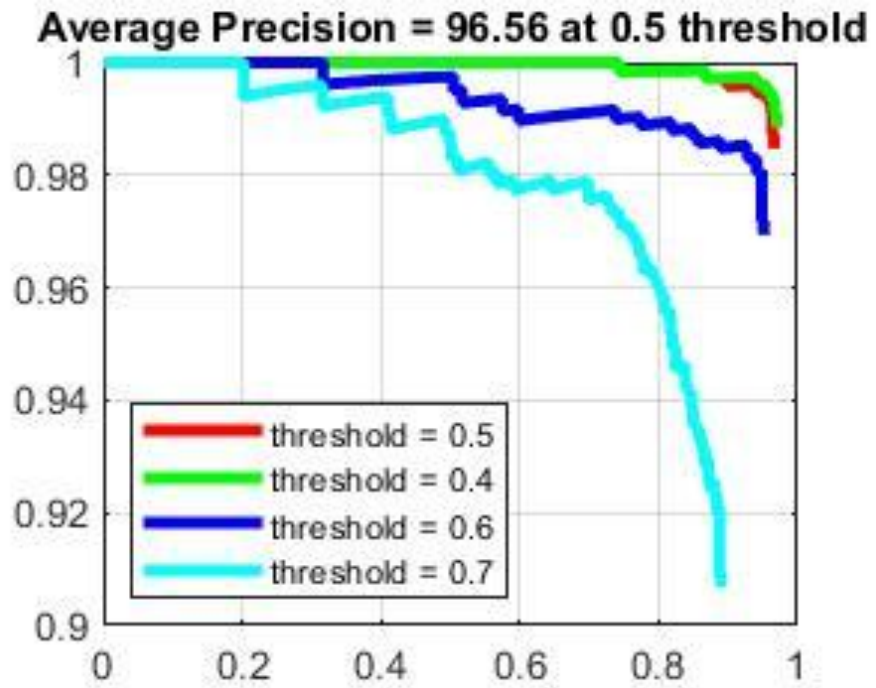
$$Precision = \frac{TP}{TP+FP} \quad 4.1$$

$$Recall = \frac{TP}{TP+FN} \quad 4.2$$

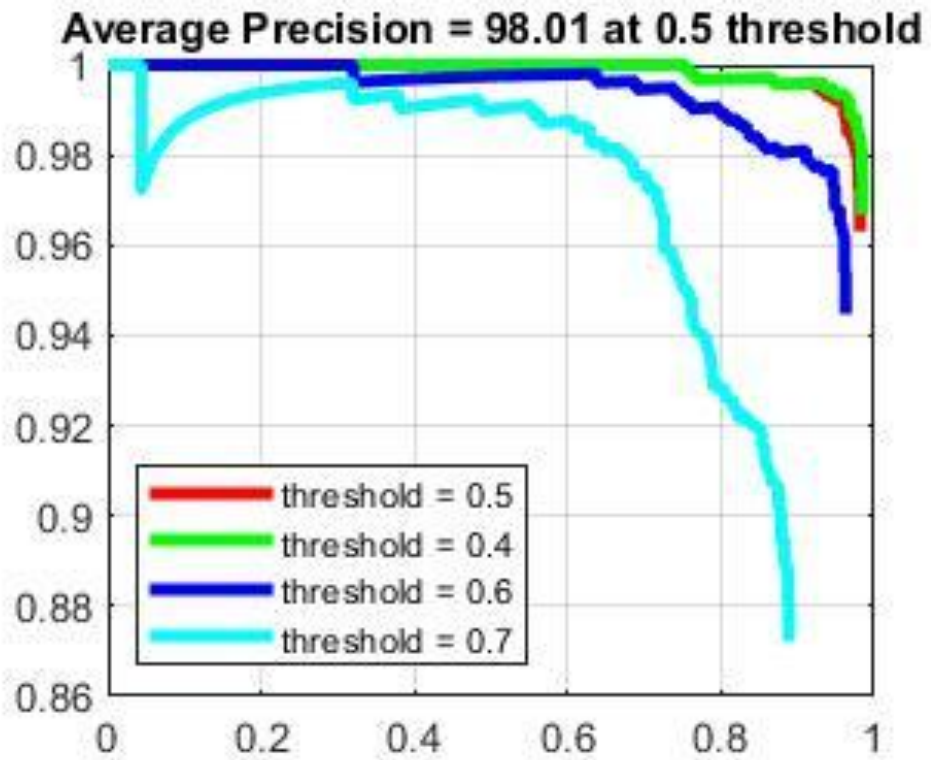
Recall bizim problemimizde uçakları doğru tespit etme oranına, Precision ise uçak dediklerimizin gerçekten kaç uçağıdır sorularımıza cevaptır.

Bu çalışmada elde edilen Precision-Recall eğrileri Faster R-CNN nesne tespit modeli için Şekil 4.2'de, R-FCN nesne tespit modeli için Şekil 4.3'de, SSD nesne tespit modeli için Şekil 4.4'te gösterilmiştir Şekil 4.5'de ise 0.5 eşik değerinde Faster R-CNN, R-FCN ve SSD için Precision- Recall eğrisi yer almaktadır.

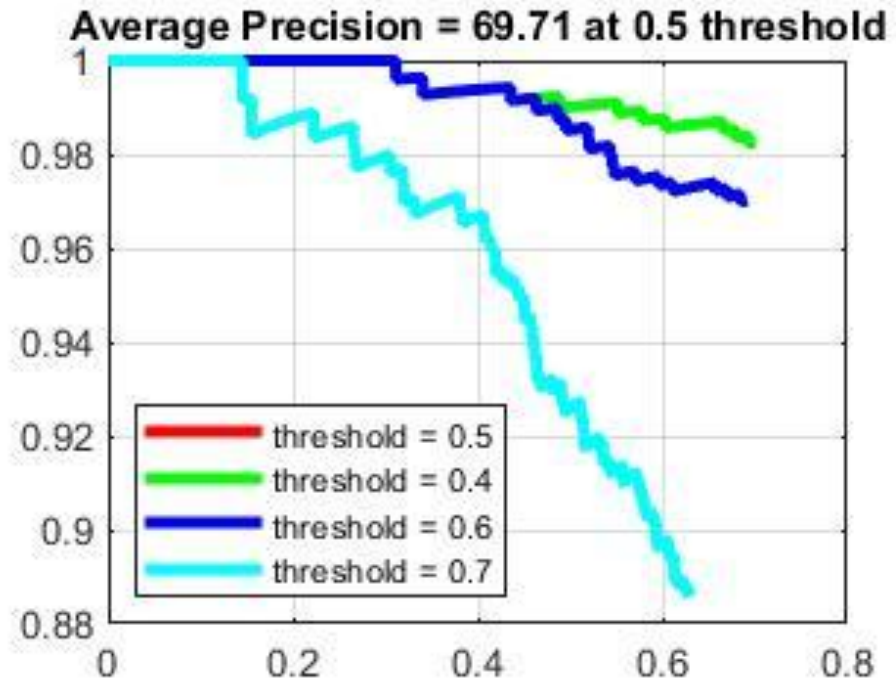




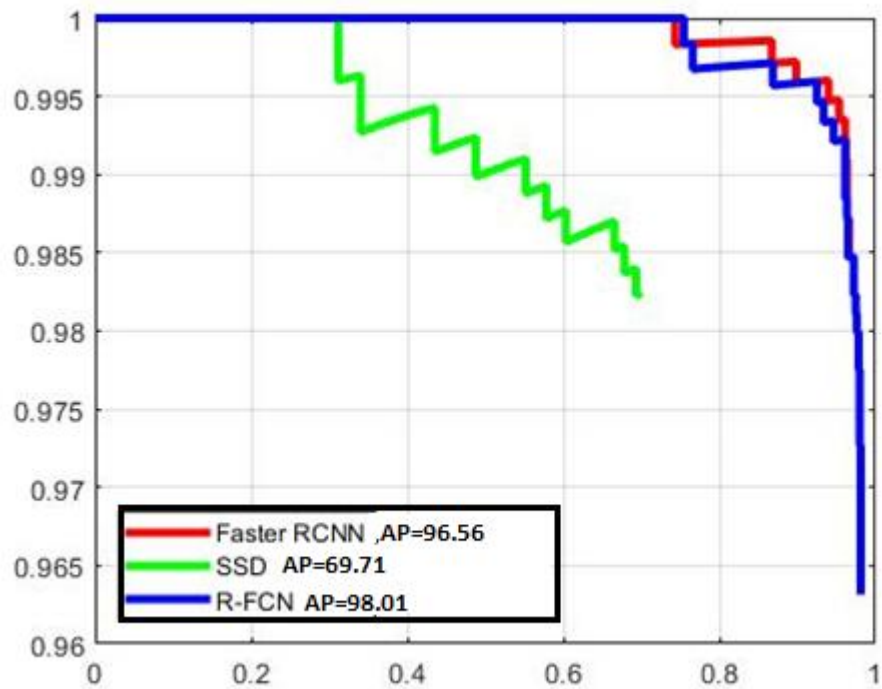
Şekil 4.2. Faster R-CNN Precision-Recall eğrisi



Şekil 4.3. R-FCN Precision-Recall eğrisi

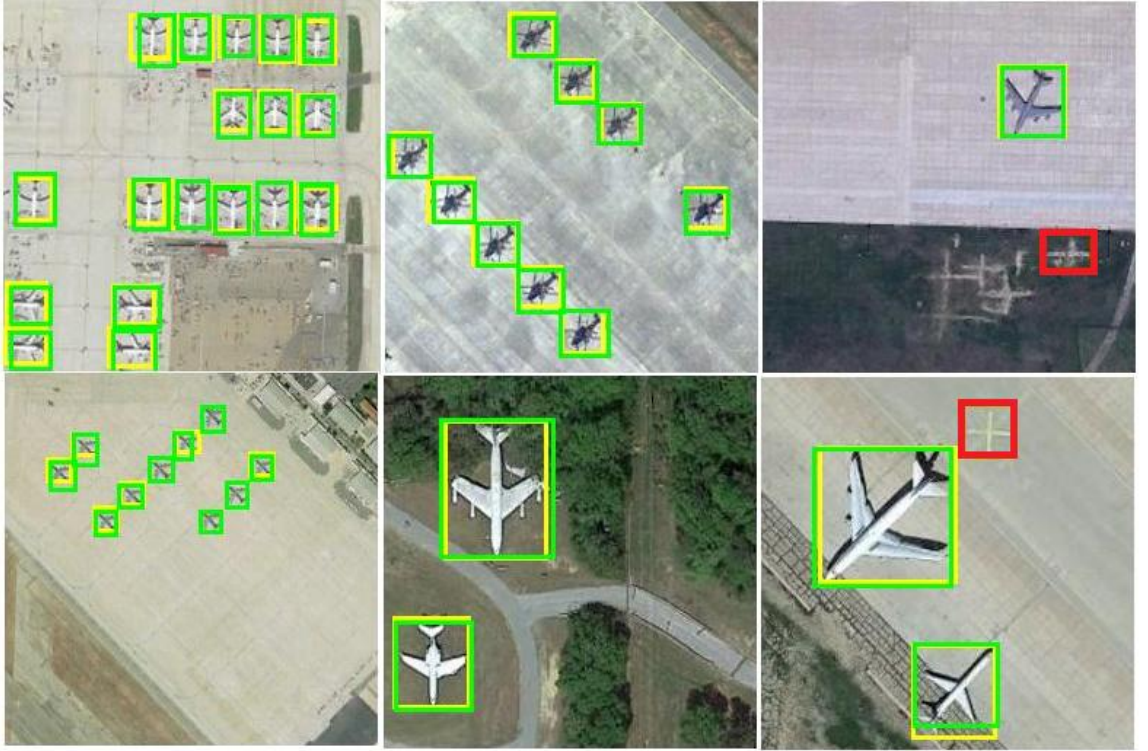


Şekil 4.4. SSD Precision-Recall eğrisi

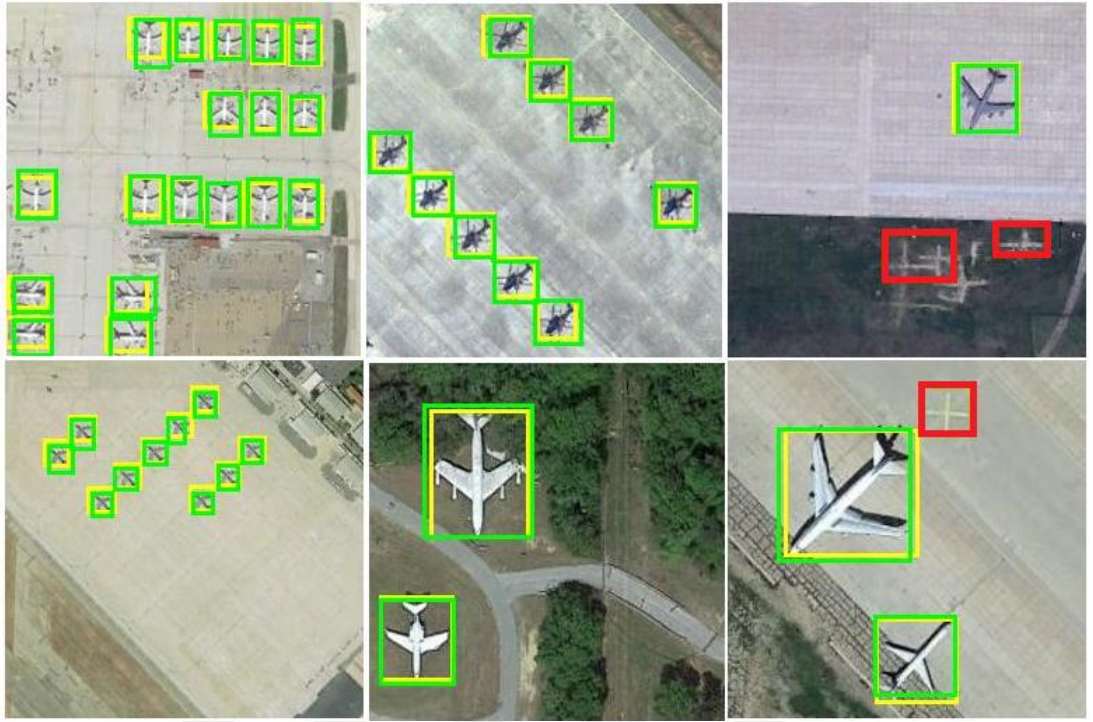


Şekil 4.5. Faster R-CNN, R-FCN ve SSD için Precision- Recall eğrisi

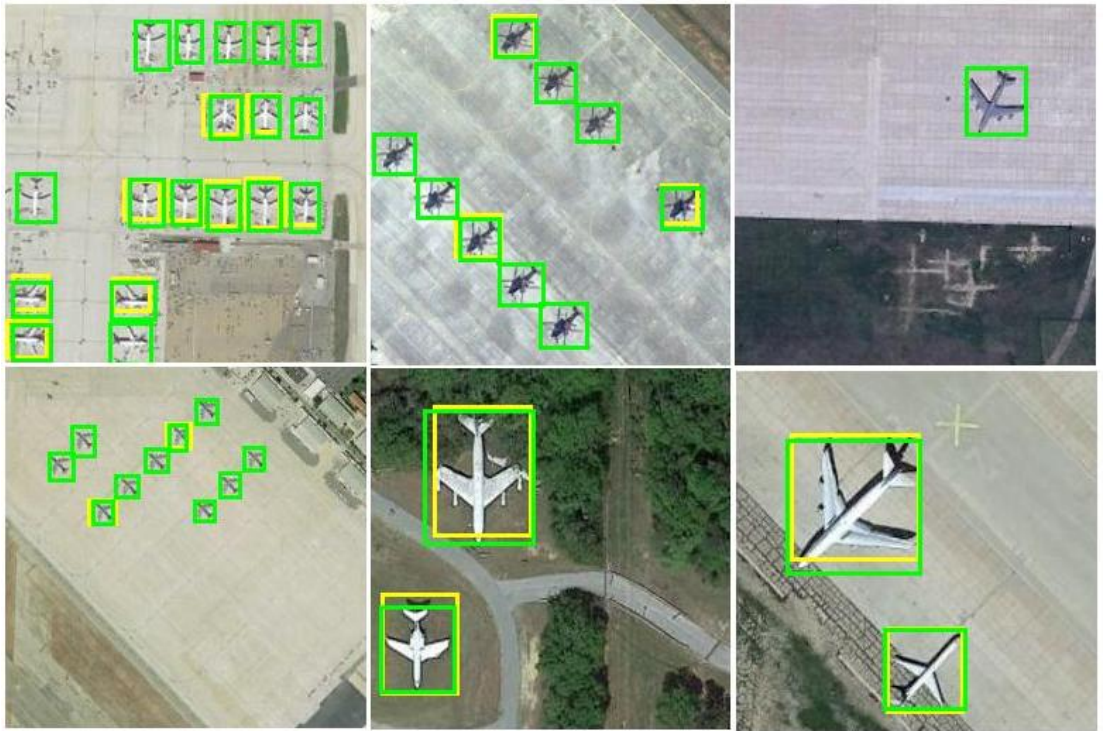
Şekil 4.6’te test verilerinden alınan bazı görüntüler üzerinde Faster R-CNN modelin test sonuçları, Şekil 4.7’te R-FCN modelin test sonuçları ve Şekil 4.8’te SSD modelin test sonuçları verilmiştir. Görsellerdeki yeşil kutular kesin referansı, sarı kutular doğru tespitleri ve kırmızı kutular da yanlış tespitleri ifade etmektedir.



Şekil 4.6. Faster R-CNN modelin test sonuçları



Şekil 4.7. R-FCN modelin test sonuçları



Şekil 4.8. SSD modelin test sonuçları

Arařtırma bulgularından elde edilen sonulara gre test verimiz zerinde en yksek performans %98.01 oranında R-FCN modeli ile elde edilmiřtir. Kullanılan nesne tespit yntemlerinde farklı eřik deęerleri ile denemeler yapılıp ortak eřik deęerinde kıyaslanmıřtır.



## 5. SONUÇ

Bu tezde, Google Earth Pro'dan farklı zamanlarda ve yerlerde alınan uçak uydu görüntüleri ile yeni bir veri seti oluşturuldu. 2705 uydu görüntüsünü içeren ATA-Plane veri kümesi, 2401 görüntü eğitim için, geri kalan 304 görüntü test için olmak üzere ayrılmıştır. Her görüntüde en az bir hedef bulunan bu görüntülerde, dünyanın farklı bölgelerinden, farklı boyutlarda yaklaşık 8584 örnek toplandı. Çok popüler bir gerçek zamanlı nesne tespit algoritması olan SSD, Faster R-CNN ve R-FCN veri setimizde eğitildi. Test verisiyle değerlendirme yapılmış ve modelin nesne tespit performansı ölçülmüş ve başarılı sonuç elde edilmiştir. R-FCN, SSD ve Faster R-CNN için elde edilen doğruluk oranları sırasıyla %98.01, %69.71 ve %96.56 olarak elde edilir.

## KAYNAKLAR

- Chen, H., Zhao, J., Gao, T., & Chen, W., 2018. Fast Airplane Detection with Hierarchical Structure in Large Scene Remote Sensing Images at High Spatial Resolution. *2018 {IEEE} International Geoscience and Remote Sensing Symposium, {IGARSS} 2018, Valencia, Spain, July 22-27, 2018, 4846–4849.* <https://doi.org/10.1109/IGARSS.2018.8517609>
- Chen, X., Xiang, S., Liu, C.-L., & Pan, C.-H., 2015. Aircraft Detection by Deep Convolutional Neural Networks. *IPSN Transactions on Computer Vision and Applications.* <https://doi.org/10.2197/ipsjtcva.7.10>
- Dai, J., Li, Y., He, K., & Sun, J., 2016. *R-FCN: Object Detection via Region-based Fully Convolutional Networks.* Retrieved from <http://arxiv.org/abs/1605.06409>
- Dalal, N., & Triggs, B., 2005. Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005.* <https://doi.org/10.1109/CVPR.2005.177>
- Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., & Zou, H., 2018. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing.* <https://doi.org/10.1016/j.isprsjprs.2018.04.003>
- Gibiansky, A., 2015. *lecture 09 Convolutional Neural Network: Architectures, Convolution/Polling Layers.*
- Girshick, R., 2015. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision.* <https://doi.org/10.1109/ICCV.2015.169>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* <https://doi.org/10.1109/CVPR.2014.81>
- GitHub. 2017. LabelImg. Retrieved from <https://github.com/tzutalin/labelImg>
- Gomez, V., Cortes, A., & Noguer, F., 2015. Object Detection for Autonomous Driving Using Deep Learning. *Meeting of the Universitat Politècnica de Catalunya, Spain, (December).*
- Google 2001. Google Earth. Retrieved from <https://www.google.com/earth/>.
- Han, Z., Zhang, H., Zhang, J., & Hu, X., 2018. Fast aircraft detection based on region locating network in large-scale remote sensing images. *Proceedings - International Conference on Image Processing, ICIP.* <https://doi.org/10.1109/ICIP.2017.8296691>
- Hui, J. (2018). Understanding Region-based Fully Convolutional Networks (R-FCN) for object detection. Retrieved from [https://medium.com/@jonathan\\_hui/understanding-region-based-fully-convolutional-networks-r-fcn-for-object-detection-828316f07c99](https://medium.com/@jonathan_hui/understanding-region-based-fully-convolutional-networks-r-fcn-for-object-detection-828316f07c99)
- ImageNet 2016. ImageNet. Retrieved from <http://www.image-net.org/>
- Karpathy, A. (2019). CS231n: Convolutional Neural Networks for Visual Recognition. Retrieved from <http://cs231n.github.io/classification/>
- Khan, P., 2017. How can i detect and localize object using tensorflow and convolutional neural network. Retrieved from

- <https://stackoverflow.com/questions/45035831/how-can-i-detect-and-localize-object-using-tensorflow-and-convolutional-neural-n>.
- Kharchenko, V., & Chyrka, I., 2018. Detection of Airplanes on the Ground Using YOLO Neural Network. *International Conference on Mathematical Methods in Electromagnetic Theory, MMET*. <https://doi.org/10.1109/MMET.2018.8460392>
- Kim, H., Kim, H., Hong, Y. W., & Byun, H., 2017. Detecting Construction Equipment Using a Region-Based Fully Convolutional Network and Transfer Learning. *Journal of Computing in Civil Engineering*. [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000731](https://doi.org/10.1061/(asce)cp.1943-5487.0000731)
- Krizhevsky, A., Sutskever, I., & Hinton, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Retrieved from <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- Li, B., Cui, X., & Bai, J., 2016. A cascade structure of aircraft detection in high resolution remote sensing images. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016-Novem, 653–656. <https://doi.org/10.1109/IGARSS.2016.7729164>
- Li, X., & Wang, S., 2017. Object Detection Using Convolutional Neural Networks in a Coarse-to-Fine Manner. *IEEE Geoscience and Remote Sensing Letters*, 14(11), 2037–2041. <https://doi.org/10.1109/LGRS.2017.2749478>
- Li, X., Wang, S., Jiang, B., & Chan, X., 2018. Airplane detection using convolutional neural networks in a coarse-to-fine manner. *Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2017*, 2018-Janua, 235–239. <https://doi.org/10.1109/ITNEC.2017.8284943>
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... Zitnick, C. L., 2014. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5), 740–755. [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
- Liu, M., Gao, Z., Li, S., Zhou, Z., & Wang, B., 2017. Pixel-based airplanes segmentation in remote sensing image. *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, 4811–4816. <https://doi.org/10.1109/CCDC.2017.7979346>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C., 2016. SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- Lowe, D. G., 1999. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*. <https://doi.org/10.1109/ICCV.1999.790410>
- Luo, Q., & Shi, Z., 2016. Airplane detection in remote sensing images based on Object Proposal. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016-Novem, 1388–1391. <https://doi.org/10.1109/IGARSS.2016.7729355>
- Mitchell, T. M., 1997. *McGrawHill, Tom.Mitchell.-Machine.Learning.pdf*.
- Mohammadi, M. R., 2018. *Deep Multiple Instance Learning for Airplane Detection in*



- High Resolution Imagery*. Retrieved from <http://arxiv.org/abs/1808.06178>
- Nelli, F., & Nelli, F., 2018. Deep Learning with TensorFlow. In *Python Data Analytics*. [https://doi.org/10.1007/978-1-4842-3913-1\\_9](https://doi.org/10.1007/978-1-4842-3913-1_9)
- Pan, B., Wang, L., Yu, X., & Shi, Z., 2018. Single-Sample Aeroplane Detection in High-Resolution Optimal Remote Sensing Imagery. *2018 {IEEE} International Geoscience and Remote Sensing Symposium, {IGARSS} 2018, Valencia, Spain, July 22-27, 2018*, 2495–2498. <https://doi.org/10.1109/IGARSS.2018.8518820>
- Polat, E., & Yildiz, C., 2012. Stationary aircraft detection from satellite images. *Istanbul University - Journal of Electrical and Electronics Engineering*, 12(2), 1523–1528.
- Qiu, S., Wen, G., & Fan, Y., 2017. Using layered object representation to detect partially visible airplanes in remote sensing images. *PIC 2016 - Proceedings of the 2016 IEEE International Conference on Progress in Informatics and Computing*. <https://doi.org/10.1109/PIC.2016.7949486>
- Ren, S., He, K., Girshick, R., & Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Service, C. U. B., 2015. Neural Networks and Machine Learning. Retrieved from <http://blogs.cornell.edu/info2040/2015/09/08/neural-networks-and-machine-learning/>
- Simonyan, K., & Zisserman, A., 2014. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 1–14. Retrieved from <http://arxiv.org/abs/1409.1556>
- Van De Sande, K. E. A., Uijlings, J. R. R., Gevers, T., & Smeulders, A. W. M., 2011. Segmentation as selective search for object recognition. *Proceedings of the IEEE International Conference on Computer Vision*. <https://doi.org/10.1109/ICCV.2011.6126456>
- Wan, J., Hoi, S. C. H., Wu, P., Zhu, J., Wan, J., Wang, D., ... Zhang, Y., 2014. Deep learning for content-based image retrieval: A comprehensive study. *Proceedings of the ACM International Conference on Multimedia*.
- Wu, H., Zhang, H., Zhang, J., & Xu, F., 2016. Typical Target Detection in Satellite Images Based on Convolutional Neural Networks. *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, 2956–2961. <https://doi.org/10.1109/SMC.2015.514>
- Xia, G. S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., ... Zhang, L., 2018. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2018.00418>
- Xia, G. S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., ... Lu, X., 2017. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*. <https://doi.org/10.1109/TGRS.2017.2685945>
- Xu, Y., Zhu, M., Xin, P., Li, S., Qi, M., & Ma, S., 2018. Rapid airplane detection in remote sensing images based on multilayer feature fusion in fully convolutional neural networks. *Sensors (Switzerland)*. <https://doi.org/10.3390/s18072335>
- Yang, J., Zhu, Y., Jiang, B., Gao, L., Xiao, L., & Zheng, Z., 2018. Aircraft detection in remote sensing images based on a deep residual network and Super-Vector coding. *Remote Sensing Letters*.

- <https://doi.org/10.1080/2150704X.2017.1415474>
- Yang, Y., Zhuang, Y., Bi, F., Shi, H., & Xie, Y., 2017. M-FCN: Effective fully convolutional network-based airplane detection framework. *IEEE Geoscience and Remote Sensing Letters*. <https://doi.org/10.1109/LGRS.2017.2708722>
- Yokoya, N., & Iwasaki, A. (2015). Object Detection Based on Sparse Representation and Hough Voting for Optical Remote Sensing Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. <https://doi.org/10.1109/JSTARS.2015.2404578>
- Yu, D., Zhao, C., Xu, J., & Lin, Y., 2018. A method of aircraft detection using fully convolutional network. *2017 International Conference on Computer Systems, Electronics and Control, ICCSEC 2017*, 914–918. <https://doi.org/10.1109/ICCSEC.2017.8446851>
- Yu, Y., 2019. Object Localization via Deep Neural Network. Retrieved from <http://www.cbsr.ia.ac.cn/users/ynyu/detection.html>
- Yu, Yongtao, Guan, H., & Ji, Z., 2015. Rotation-Invariant Object Detection in High-Resolution Satellite Imagery Using Superpixel-Based Deep Hough Forests. *IEEE Geoscience and Remote Sensing Letters*. <https://doi.org/10.1109/LGRS.2015.2432135>
- Zaccone, Giancarlo, Md Rezaul Karim, and A. M., 2017. *Deep Learning with TensorFlow*.

## ÖZGEÇMİŞ

01.01.1989 Yemen'in Taiz şehrinde doğdu. İlk, orta ve lise öğrenimini Al-Falah Motali okulunda tamamladı. 2010 yılında Taiz Üniversitesi Mühendislik Fakültesi Mekatronik ve Robotik Mühendisliği Bölümü'nü kazandı ve 2014 yılında mezun oldu. 2016 yılında Türkiye Cumhuriyetinin düzenlediği burs programına (YTB) başvurarak Atatürk Üniversitesi, Fen Bilimleri Enstitüsü Elektrik ve Elektronik Mühendisliği Anabilim Dalı, Elektronik alanında Yüksek Lisans yapmaya hak kazandı.

