

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**İNSANSIZ HAVA ARAÇLARINDAN ALINAN
GÖRÜNTÜLERDE SÜPER ÇÖZÜNÜRLÜK YÖNTEMLERİ
KULLANARAK NESNE TESPİTİNİN İYİLEŞTİRİLMESİ**

Settar YİĞİT

YÜKSEK LİSANS TEZİ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Programı

Danışman

Prof. Dr. Tülay YILDIRIM

ubat, 2024

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İNSANSIZ HAVA ARAÇLARINDAN ALINAN
GÖRÜNTÜLERDE SÜPER ÇÖZÜNÜRLÜK YÖNTEMLERİ
KULLANARAK NESNE TESPİTİNİN İYİLEŞTİRİLMESİ

Settar YİĞİT tarafından hazırlanan tez çalışması 27.02.2024 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, Elektronik Programı **YÜKSEK LİSANS** olarak kabul edilmiştir.

Prof. Dr. Tülay YILDIRIM
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Prof. Dr. Tülay YILDIRIM, Danışman
Yıldız Teknik Üniversitesi

Doç. Dr. Bülent BOLAT, Üye
Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Nurullah ÇALIK, Üye
İstanbul Medeniyet Üniversitesi

Danışmanım Prof. Dr. Tülay YILDIRIM sorumluluğunda tarafımca hazırlanan “İnsansız Hava Araçlarından Alınan Görüntülerde Süper Çözünürlük Yöntemleri Kullanılarak Nesne Tespitinin İyileştirilmesi” başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Settar YİĞİT

İmza



*Aileme
ve
arkadaşlarıma*

TEŐEKKÜR

Bu tez alıőmamda gerekleőtirdiđim tım akademik alıőmalarına yın veren ve desteklerini hibir zaman esirgemeyen danıőman hocam Prof. Dr. Tılay Yıldırım'a teőekkırı bor bilirim. Ayrıca bu sırete her tırlı manevi desteđi sađlamalarından ıtırı aileme ve arkadaőlarımaya itenlikle teőekkır ediyorum.

Settar YİĐİT

İÇİNDEKİLER

TEŞEKKÜR	iv
İÇİNDEKİLER	v
KISALTMA LİSTESİ	vii
ŞEKİL LİSTESİ	viii
ÖZET	x
ABSTRACT	xii
1 GİRİŞ	1
1.1 Problem Tanımı.....	2
1.2 Tezin Amacı.....	3
1.3 Literatür özeti.....	4
1.3.1 Nesne Tespiti	4
1.3.2 Süper Çözünürlük	7
1.4 Tezin Yapısı.....	13
2 ÇEKİŞMELİ ÜRETİCİ AĞLAR (GAN)	14
2.1 SRGAN.....	15
2.2 ESRGAN.....	16
2.3 Real-ESRGAN.....	17
3 EVRİŞİMLİ SİNİR AĞLARI	18
YOLO	19
4 MATERYAL VE YÖNTEM	21
4.1 Veri Setleri.....	21
4.1.1 VisDrone	21
4.1.2 Stanford Drone Dataset (SDD)	22
4.2 Yöntem.....	22
5 DENEYSEL ÇALIŞMALAR	25
5.1 YOLO-V7 LR_VisDrone ile Eğitim ve Test Sonuçları.....	27
5.2 YOLO-V7 SR_VisDrone ile Eğitim ve Test Sonuçları.....	32
5.3 YOLO-V8 LR_VisDrone ile Eğitim ve Test Sonuçları.....	38
5.4 YOLO-V8 SR_VisDrone ile Eğitim ve Test Sonuçları.....	40
5.5 YOLO-V8 LR_SDD ile Eğitim ve Test Sonuçları.....	42
5.6 YOLO-V8 SR_SDD ile Eğitim ve Test Sonuçları.....	44

6 SONUÇ	47
KAYNAKÇA	49
TEZDEN ÜRETİLMİŞ YAYINLAR	57



KISALTMA LİSTESİ

İHA	İnsansız Hava Aracı
UAV	Unmanned Aerial Vehicle
GAN	Generative Adversarial Network
SRGAN	Super Resolution Generative Adversarial Network
ESRGAN	Enhancement Super Resolution Generative Adversarial Network
EEGAN	Edge Enhancement Generative Adversarial Network
EDSR	Enhanced Deep Residual Network
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
YOLO	You Look Only Once
SDD	Stanford Drone Dataset
SR	Super Resolution
LR	Low Resolution
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
VGG	Visual Geometry Group
SSD	Single Shot Detector

ŞEKİL LİSTESİ

Şekil 1.1 CNN tabanlı modellerin VOC2012 üzerindeki sonuçları [18].....	5
Şekil 2.1 GAN ağının mimari yapısı [52].....	15
Şekil 2.2 SRGAN üretici ve ayırıcı ağlarının mimarisi [29]	16
Şekil 2.3 ESRGAN mimarisi [29]	16
Şekil 2.4 Real-ESRGAN mimarisi [36]	17
Şekil 3.1 Evrişimli Sinir Ağlarının genel mimarisi [52]	19
Şekil 3.2 YOLO genel mimarisi [16]	19
Şekil 4.1 Önerilen mimari	22
Şekil 5.1 LR_VisDrone ile eğitilen modelin karışıklık matrisi	27
Şekil 5.2 LR_VisDrone ile eğitilen modelin F1 skoru	28
Şekil 5.3 LR_VisDrone ile eğitilen modelin kesinlik eğrisi.....	28
Şekil 5.4 LR_VisDrone ile eğitilen modelin duyarlılık eğrisi.....	29
Şekil 5.5 LR_VisDrone ile eğitilen modelin kesinlik-duyarlılık eğrisi.....	29
Şekil 5.6 LR_VisDrone ile eğitilen modelin gerçek etiketleri	30
Şekil 5.7 LR_VisDrone ile eğitilen modelin tahmin etiketleri.....	30
Şekil 5.8 LR_VisDrone ile eğitilen modelin LR ile test karışıklık matrisi	31
Şekil 5.9 LR_VisDrone ile eğitilen modelin LR ile test mAP sonuçları.....	31
Şekil 5.10 LR_VisDrone ile eğitilen modelin SR ile test karışıklık matrisi.....	32
Şekil 5.11 LR_VisDrone ile eğitilen modelin SR ile test mAP sonuçları.....	32
Şekil 5.12 SR_VisDrone ile eğitilen modelin karışıklık matrisi	33
Şekil 5.13 SR_VisDrone ile eğitilen modelin F1 skoru	33
Şekil 5.14 SR_VisDrone ile eğitilen modelin kesinlik-duyarlılık eğrisi	34
Şekil 5.15 SR_VisDrone ile eğitilen modelin kesinlik eğrisi.....	34
Şekil 5.16 SR_VisDrone ile eğitilen modelin duyarlılık eğrisi	35
Şekil 5.17 SR_VisDrone ile eğitilen modelin tahmin etiketleri	35
Şekil 5.18 SR_VisDrone ile eğitilen modelin gerçek etiketleri.....	36
Şekil 5.19 SR_VisDrone ile eğitilen modelin LR ile test karışıklık matrisi.....	36
Şekil 5.20 SR_VisDrone ile eğitilen modelin LR ile test mAP sonuçları	37
Şekil 5.21 SR_VisDrone ile eğitilen modelin SR ile test karışıklık matrisi.....	37
Şekil 5.22 SR_VisDrone ile eğitilen modelin SR ile test mAP sonuçları	38
Şekil 5.23 LR_VisDrone ile eğitilen modelin LR ile test mAP sonuçları.....	38

Şekil 5.24	LR_ VisDrone ile eğitilen modelin LR ile test karışıklık matrisi	39
Şekil 5.25	LR_ VisDrone ile eğitilen modelin SR ile test mAP sonuçları	39
Şekil 5.26	LR_ VisDrone ile eğitilen modelin SR ile test karışıklık matrisi.....	40
Şekil 5.27	SR_ VisDrone ile eğitilen modelin LR ile test mAP sonuçları	40
Şekil 5.28	SR_ VisDrone ile eğitilen modelin LR ile test karışıklık matrisi.....	41
Şekil 5.29	SR_ VisDrone ile eğitilen modelin SR ile test mAP sonuçları	41
Şekil 5.30	SR_ VisDrone ile eğitilen modelin SR ile test karışıklık matrisi.....	42
Şekil 5.31	LR_ SDD ile eğitilen modelin LR ile test mAP sonuçları	42
Şekil 5.32	LR_ SDD ile eğitilen modelin LR ile test karışıklık matrisi.....	43
Şekil 5.33	LR_ SDD ile eğitilen modelin SR ile test mAP sonuçları	43
Şekil 5.34	LR_ SDD ile eğitilen modelin SR ile test karışıklık matrisi	44
Şekil 5.35	SR_ SDD ile eğitilen modelin LR ile test mAP sonuçları	44
Şekil 5.36	SR_ SDD ile eğitilen modelin LR ile test karışıklık matrisi	45
Şekil 5.37	SR_ SDD ile eğitilen modelin SR ile test mAP sonuçları.....	45
Şekil 5.38	SR_ SDD ile eğitilen modelin SR ile test karışıklık matrisi	46

İnsansız Hava Araçlarından Alınan Görüntülerde Süper Çözünürlük Yöntemleri Kullanılarak Nesne Tespitinin İyileştirilmesi

Settar YİĞİT

Elektronik ve Haberleşme Anabilim Dalı

Elektronik Programı

Yüksek Lisans Tezi

Danışman: Prof. Dr. Tülay YILDIRIM

İnsansız Hava Araçları (İHA) ya da yaygın olarak bilinen adıyla dronlar, son yıllarda eğlence sektöründen hizmet sektörüne kadar pek çok alanda popülerlik kazanmıştır. Bu hava araçlarının kullanım alanları her geçen gün genişlemekte ve giderek daha çeşitli hale gelmektedir. Savunma sanayisinde de son zamanlarda sıkça kullanılmakta olup hem savunma hem de taarruz durumlarında vazgeçilmez bir envanter haline gelmiştir. Bu bağlamda sahip ülkelerin stratejik avantajlarına önemli katkılarda bulunmakta ve üstünlük kurabilmesinde ciddi derecede etkili olabilmektedir. Tüm bunların bir getirisi olarak, İHA'lardan elde edilen görüntülerle gerçekleştirilen nesne tespiti, takip ve diğer özelleştirilmiş görevler gün geçtikçe daha da önem kazanmaktadır. Fakat bu hava araçlarından elde edilen görüntüler, güvenli bir uçuş mesafesinden çekilmesi gerektiğinden genellikle düşük çözünürlük ve kalitede olmaktadır. Bu durum nesne tespiti ve takibi uygulamaları için bir dezavantaj teşkil etmektedir. Bu dezavantajı aşmak

adına bir dizi süper çözünürlük tekniği geliştirilmiştir. Bu çalışmada, GAN ağlarının bir çeşidi olan Real-ESRGAN modeli kullanılarak İHA'lerden alınan görüntülerin daha yüksek çözünürlüklü hale getirilmesini sağlamak amaçlanmaktadır. Uygulanan bu yöntemin nesne tespitine olan etkisini inceleyebilmek adına YOLO modelleri kullanılmıştır. Bu tez çalışmasındaki temel amaç, nesne tespiti performansını artırarak hem tespit edilen nesnelerin sayısını hem de tespit sürecinin genel doğruluğunu iyileştirmektir. Bu çalışma, iki farklı yöntemle elde edilen sonuçların karşılaştırmalı analizini sunmaktadır. İlk aşamada, VisDrone ve SDD veri setlerindeki düşük çözünürlüğe sahip görüntüler kullanılarak, YOLO-V7 ve YOLO-V8 modelleri eğitilip test edilmiştir. Daha sonra aynı YOLO-V7 ve YOLO-V8 modelleri, VisDrone ve SDD veri setlerindeki görüntülerin Real-ESRGAN modeli kullanılarak elde edilen süper çözünürlüklü (SR) versiyonları üzerinde eğitilerek test edilir. Bu iki aşama sonucunda elde edilen veriler karşılaştırılarak Real-ESRGAN modelinin nesne tespiti üzerindeki etkisi ortaya konulmaktadır. Yapılan deneysel sonuçlara göre, VisDrone veri setinin SR görüntüleri ile eğitilen YOLO-V7 modelinde, LR görüntüleri ile eğitilen YOLO-V7 modeline göre doğruluk oranında yaklaşık %1'lik artış olduğu gözlemlenmiştir. Bu model haricindeki, VisDrone veri setinin YOLO-V8 ile eğitilmesi, SDD veri setinin hem YOLO-V7 hem de YOLO-V8 ile eğitilmesi ile elde edilen modellerde herhangi bir doğruluk artışı gözlemlenmemiştir.

Anahtar Kelimeler: İHA, Real-ESRGAN, Süper çözünürlük, YOLO, Nesne Tespiti.

Improving Object Detection Using Super Resolution Methods in Unmanned Aerial Vehicle Images

Settar YİĞİT

Department of Electronics and Communication

Master of Science

Supervisor: Prof. Dr. Tülay YILDIRIM

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have gained popularity in various fields, from the entertainment sector to the service industry in recent years. The usage areas of these aerial vehicles continue to expand, becoming increasingly diverse. They have become indispensable inventory in both defense and attack situations, frequently employed in the defense industry. In this context, they significantly contribute to the strategic advantages of nations, playing a crucial role in establishing superiority. As a result, object detection, tracking, and other customized tasks performed using images captured by UAVs are gaining increasing importance. However, images obtained from these aircraft often have low resolution and quality due to the need for a safe flight distance. This poses a disadvantage for object detection and tracking applications. To overcome this drawback, a series of super-resolution techniques have been developed. In this study, the Real-ESRGAN model, a type of GAN network, is used to enhance the resolution of images captured by UAVs. To examine the impact of this applied method on object detection, the YOLO model is utilized. The main objective of this thesis is to improve object detection

performance, enhancing both the number of detected objects and the overall accuracy of the detection process. This study presents a comparative analysis of results obtained through two different methods. Initially, YOLO-V7 and YOLO-V8 models are trained and tested using LR images from the VisDrone and SDD datasets. Subsequently, the same YOLO-V7 and YOLO-V8 models are trained and tested on super-resolution (SR) versions of images obtained using the Real-ESRGAN model on the VisDrone and SDD datasets. The data obtained from these two stages are compared to highlight the impact of the Real-ESRGAN model on object detection. According to experimental results, the YOLO-V7 model trained on VisDrone dataset's SR images showed an approximately 1% increase in accuracy compared to the YOLO-V7 model trained on LR images. No accuracy increase was observed in models obtained by training the VisDrone dataset with YOLO-V8 or both YOLO-V7 and YOLO-V8 on the SDD dataset.

Keywords: UAV, Real-ESRGAN, Super Resolution, YOLO, Object Detection.

1 GİRİŞ

İnsansız Hava Araçları (İHA) içerisinde fiziksel olarak bir insan bulundurmayan uzaktan kontrol edilebilen veya otonom sürüş yapabilen bir çeşit uçaktır. İHA'lar tarihleri boyunca çeşitli kullanım amaçlarına göre geliştirilerek ve özelleştirilerek günümüzdeki karmaşık ve çok yönlü yapısına ulaşmışlardır. İlk olarak 1916 yılında kullanılmış daha sonra 1. Dünya Savaşı sırasında da keşif amaçlı kullanılmıştır. Savaş sonrasında ise birçok ülkenin dikkatini çeken İHA'lar, askeri ihtiyaçlar ve teknolojik gelişmeler ile birlikte uzun bir gelişim sürecine girmiştir. Bu gelişim sürecinin en büyük dönüm noktalarından biri 2. Dünya Savaşı sürecinde yaşanmıştır. Bu dönemde Almanya'nın ünlü V-1 uçan bombası, otomatik bir güdüm sistemine sahip ilk operasyonel insansız hava aracı olarak kayıtlara geçmiştir. Özellikle Afganistan ve Irak gibi çatışma bölgelerinde, İHA'lar keşif, gözetleme ve hedef tespiti gibi görevlerde yaygın olarak kullanılmıştır. Predator ve Reaper gibi uzaktan kumandalı silahlı İHA'lar, savaş alanındaki rolüyle öne çıkmıştır. Riskli görevlerdeki insan hayatını riske atmaması, normal uçaklara kıyasla maddi açıdan daha uygun olması bu sistemlerin tercih edilmesinde etkili olmuştur. İlerleyen yıllarda gelişen teknolojiyle birlikte boyutları küçültülüp işlevleri arttırılarak kullanım alanları geniş bir yelpazeye yayılmasının önü açılmıştır. Bugün büyük ölçüde otonom çalışabilen, çeşitli sensörler ve kameralarla donatılan, hava sahasında etkileşime girebilen ve hatta uzaktan kumandalı silah sistemleri taşıyan İHA'lar savunma sanayisinin vazgeçilmez bir parçası haline gelmiştir.

Askeri uygulamaların yanı sıra sivil havacılık uygulamalarında da oldukça yaygın bir şekilde kullanılmaktadır. Bu uygulamaları örneklendirmek gerekirse; haritalama uygulamaları, yüksek gerilim veya doğal gaz hattı izleme ve denetleme, yol yapım projeleri izleme ve denetleme, madencilik uygulamaları, tarım uygulamaları, trafik izleme ve denetleme, deprem-sel gibi doğal afetlerde arama kurtarma faaliyetlerinde kullanılmaktadır. [1] Örneklerden de anlaşılacağı üzere hayatımıza direkt etkisi olan birçok önemli alanda kullanılmaktadır ve hala geliştirme çalışmaları da devam etmektedir. Üzerine entegre edilebilen birçok

sensörlerle (görüntü, ses, sıcaklık vs.) amaca yönelik çözümler üretilebilmektedir. Hiç şüphesiz bunlardan en yaygın olanı kamera entegrasyonu ile çeşitli görüntü işleme uygulamalarıdır. Fakat görüntüyü alıp işleme konusunda birtakım dezavantajlar bulunmaktadır. Bunlar kullanılan sektöre göre değişiklik gösterse de temel olarak İHA'lerden elde edilen düşük çözünürlüklü görüntüler, görüntü işlemeyi zorlaştıran belki de tüm sektörlerin ortak problemidir. Bu problemi çözebilmek için elbette tasarımsal bazı değişiklikler yapmak seçenek olarak görülebilir. Örneğin daha kaliteli kamera lensleri kullanmak, daha güçlü işlemcilerle görüntü sağlamak veya daha düşük irtifalardan görüntü almak gibi çözümler düşünülebilir. Fakat bu durumlarda yine amaca yönelik olarak düşünüldüğü takdirde bu ek özellikleri sağlayabilecek şekilde batarya kapasitesinin de doğru orantılı olarak artırılması beklenir. Bu durumda İHA'nın taşıma yük kapasitesinin artırılması gerekir ki bu da daha güçlü motorlar anlamına gelmektedir. Sonuç olarak görüldüğü üzere tasarımsal olarak yapılacak bir değişiklik maliyeti oldukça artırabilmektedir. Maliyet artışı haricinde diğer yönlerden de bahsi geçen çözümler uygulanamayabilir. Örneğin askeri bir alanda kullanılmak üzere bir görüntü işleme uygulaması özelinde kullanılan İHA'nın güvenlik açısından belirlenmiş belirli bir irtifanın altında uçuş yapmaması gerekebilir. Bu sebeplerden dolayı yazılımsal çözümler ortaya atılmıştır. Bu çalışmada düşük çözünürlüklü olan İHA görüntülerinin derin sinir ağlarından faydalanılarak yüksek çözünürlüklü hale getirilmesi hedeflenmiştir.

1.1 Problem Tanımı

İHA'lar, görüntü toplama görevlerini gerçekleştirirken bir dizi engelle karşılaşabilmektedir. Özellikle düşük irtifalarda seyir halindeyken, çevresel engellere takılma riski artmaktadır. Ayrıca İHA'ların yapısal tasarımı gereği hafif olması beklenirken, bu durum yeterli ekipmanın entegre edilmesini zorlaştırabilir. Bu ve benzeri sebeplerden dolayı, İHA'lar tarafından elde edilen görüntülerin çözünürlük seviyeleri düşebilmektedir. Örneğin, askeri bir uygulama bağlamında güvenlik nedenleriyle İHA'nın çok yüksek irtifadan görüntü alması gerekebilir. Ancak bu durum temel olarak görüntünün çözünürlüğünün düşmesine yol açabilir ve bu da görüntü üzerindeki detayların kaybolmasına neden olabilir. Sonuç olarak, görüntü işleme uygulamalarında,

özellikle nesne tespiti gibi önemli alanlarda, tespit edilen nesne sayısı ve doğruluk oranı önemli ölçüde azalabilir. Görüntü çözünürlüğündeki bu düşüş, özellikle askeri alanlarda güvenlik açısından önem arz eden uygulamalarda ciddi güvenlik zafiyetlerine yol açabilir. Nesne tespiti uygulamalarında, düşük çözünürlüklü görüntülerle çalışmak, tespit edilen nesnelerin sayısını ve tespitin doğruluk oranını olumsuz etkileyebilir. Bu durum hassas askeri operasyonlarda, güvenlik risklerini artırarak olumsuz sonuçlara neden olabilir. Bu nedenle İHA'ların görüntü çözünürlüğündeki bu potansiyel düşüşleri ele almak ve geliştirmek, özellikle savunma ve güvenlik odaklı uygulamalarda kritik bir öneme sahiptir.

1.2 Tezin Amacı

Günümüzde, İnsansız Hava Araçlarının (İHA) yaygın olarak kullanılması, özellikle askeri ve sivil alanlarda önemli bir rol oynamaktadır. İHA'lar, çeşitli uygulama alanlarında görüntü elde etme kabiliyetleri sayesinde tercih edilmektedir. Hem askeri hem de sivil sektörlerde, elde edilen bu görüntüler üzerinde nesne tespiti, takibi ve benzeri birçok uygulama gerçekleştirilmektedir. Ancak bazı durumlar ve sebeplerle elde edilen görüntülerin çözünürlük seviyelerinin düşük olması, nesne tespiti uygulamalarındaki doğruluk oranını olumsuz etkilemektedir. Bu çalışma, düşük çözünürlük sorununa odaklanarak, elde edilen görsellerin çözünürlüklerini artırarak nesne tespiti ve takibindeki doğruluk oranını geliştirmeyi amaçlamaktadır. İki önemli veri seti olan VisDrone ve Stanford Drone Dataset (SDD), İHA'lar tarafından elde edilen çeşitli resim ve videoları içermekte, içerdikleri nesnelere arasında; insanlar, araçlar, bisikletler gibi farklı öğeler bulunmaktadır. Ancak bu görüntülerin düşük çözünürlükte olması, bu veri setlerinin kullanımını zorlaştırmaktadır. Bu araştırmanın ilk aşamasında, Real-ESRGAN (Real Enhancement Super Resolution Generative Adversarial Network) süper çözünürlük yöntemi kullanılarak, düşük çözünürlüklü veri setlerinin çözünürlükleri yükseltilmeye çalışılmıştır. Ardından elde edilen yüksek çözünürlüklü (SR) veri seti, popüler nesne tespit modelleri olan YOLO ile eğitilerek, nesne tespiti doğruluk oranının nasıl etkilendiği ortaya konulmuştur. Diğer bir deyişle bu çalışma Real-ESRGAN modelinin nesne tespitine olan etkisini direkt olarak göstermektedir. Bu sayede, İHA'lar tarafından

elde edilen düşük çözünürlüklü verilerin daha etkili bir şekilde kullanılabilmesi için yeni yaklaşımlar ve çözümler önerilebilecektir.

1.3 Literatür özeti

Bu çalışmada Süper Çözünürlüğün nesne tespitine olan etkisi araştırılmıştır. Bu bölümde literatürdeki ilgili çalışmalar incelenmiştir. Bu çalışmalar 2 ana başlık altında toplanmıştır: Nesne Tespiti ve Süper Çözünürlük.

1.3.1 Nesne Tespiti

Nesne tespiti ve takibi uygulamaları literatürdeki popülerliğini uzun zamandır sürdürmektedir. Geleneksel nesne tespiti uygulamalarında, bir modelin yeni bir nesneyi tanıyabilmesi için kullanılan nesnenin görsel özellikleri genellikle manuel olarak belirlenirdi. Bu özellikler, ne kadar derin ve karmaşık olarak elde edilirse, eğitilen modelin o nesneyi tanıma yeteneğinin o kadar arttığı bilinmektedir. Bu yüzden bu özelliklerin manuel olarak veya eski yöntemlerle çıkarılması, modelin hassasiyetini artırabilir ve düşük kaliteli bir görsel ile karşılaşıldığında hızla bozulmasına neden olabilir. Yeni nesnelerin tanınması ve bu nesnelerin özelliklerinin belirlenmesi süreci, gelişmiş ve otomatik öğrenme teknikleri kullanılarak gerçekleştirilmelidir. Bu yaklaşım, modelin genellemesi ve çeşitli görsel veri koşullarında daha dirençli olması açısından daha etkili bir çözüm sunabilir. Bu nedenle, derin öğrenme ve özellikle evrişimli sinir ağı (CNN) tabanlı yöntemler, nesne tespiti alanında daha esnek ve güçlü bir çerçeve sunabilir. Geliştirilen yeni özellik çıkarma metotları ile birlikte daha derin özellik çıkarma sağlanarak nesne tespitinde doğruluğun artmasını sağlamıştır. Bu modeller ağ mimarisi, öğrenme stratejisi ve optimizasyon fonksiyonları gibi özellikleri bakımından farklı amaçlar için özelleştirilebilmektedir. Kullanım alanları olarak sınıflandırma [2], [3], insan davranışı analizi [4], yüz tanıma [5], oto-pilot [6], [7] gibi yaygın kullanım alanları örnek verilebilir. Farklı nesnelerin tanınabilmesi için özellik çıkarımı oldukça önemlidir. Geçmişte manuel olarak çıkarılan bu özellikler için birçok farklı yöntem geliştirilmiştir. Bunlardan Ölçek Değişmez Unsur Dönüşümü (SIFT) [8], Yönlü Gradyanlar Histogramı (HOG) [9] ve Haar Like literatürün önemli özellik çıkarım araçlarıdır.

Derin sinir ağlarının literatüre katılımıyla [3] [10] karmaşık girdileri daha iyi öğrenebilen yapısı sayesinde nesne tespit uygulamaları oldukça hız kazanmıştır. Geleneksel yaklaşımlardan farklı bir yaklaşıma sahip olan CNN modelinin [11] derin katmanlar oluşturabilme yeteneği ile daha kompleks yapıların daha hızlı öğrenilebilmesinin önü açılmıştır. Bu sayede katman sayısı artırılarak daha fazla parametre barındırabilen CNN ağları doğruluğa direkt etki eden yeni modelleme mimarilerinin geliştirilmesine de olanak sağlamıştır. Bu mimarilerin GPU (Graphics Processing Unit) işlemcilerinin yüksek paralel işlem yapabilme yeteneği ile kullanılmaya başlaması gelişim sürecini oldukça hızlandırmıştır. Gelişim sürecinin devamında öğrenim süresini kısaltmak ve daha kararlı hale getirebilmek adına normalizasyon tekniği olan Batch Normalizasyonu ortaya atılmış olup daha verimli modellerin üretilmesine olanak sağlanmıştır [12]. Performans iyileştirmesi birçok farklı model ile devam etmiştir. Bunlardan en önemli kilometre taşlarından birisi 2012 yılında bir nesne tanıma yarışması sırasında ortaya çıkan AlexNet'tir [3]. Ardından 2014 yılında Google tarafından üretilen GoogleNet [13] modeli tanıtılmıştır. Bunlara ek olarak VGG (Visual Geometry Group) [14], ResNet(Residual Net) [15] gibi modeller de literatüre oldukça katkı sağlamış mimarilerdir. YOLO (You Look Only Once) [16] ve SSD (Single Shot Detector) [17] CNN mimarisine sahip en önemli nesne tespit

COMPARATIVE RESULTS ON VOC 2012 TEST SET (%)

Methods	Trained on	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN(Alex) [15]	12	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1	53.3
R-CNN(VGG16) [15]	12	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3	62.4
Bayes [85]	12	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2	66.4
Fast R-CNN [65]	07++12	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2	68.4
SuffNet30 [100]	12	83.0	76.9	71.2	51.6	50.1	76.4	75.7	87.8	48.3	74.8	55.7	85.7	81.2	80.3	79.5	44.2	71.8	61.0	78.5	65.4	70.0
NOC [114]	07+12	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1	68.8
MR-CNN&S-CNN [105]	07++12	85.5	82.9	76.6	57.8	62.7	79.4	77.2	86.6	55.0	79.1	62.2	87.0	83.4	84.7	78.9	45.3	73.4	65.8	80.3	74.0	73.9
HyperNet [101]	07++12	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7	71.4
OHEM+Fast R-CNN [113]	07++12+coco	90.1	87.4	79.9	65.8	66.3	86.1	85.0	92.9	62.4	83.4	69.5	90.6	88.9	88.9	83.6	59.0	82.0	74.7	88.2	77.3	80.1
ION [95]	07+12+S	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9	57.8	82.0	64.7	88.9	86.5	84.7	82.3	51.4	78.2	69.2	85.2	73.5	76.4
Faster R-CNN [17]	07++12	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5	70.4
Faster R-CNN [17]	07++12+coco	87.4	83.6	76.8	62.9	59.6	81.9	82.0	91.3	54.9	82.6	59.0	89.0	85.5	84.7	84.1	52.2	78.9	65.5	85.4	70.2	75.9
YOLO [18]	07++12	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8	57.9
YOLO+Fast R-CNN [18]	07++12	83.4	78.5	73.5	55.8	43.4	79.1	73.1	89.4	49.4	75.5	57.0	87.5	80.9	81.0	74.7	41.8	71.5	68.5	82.1	67.2	70.7
YOLOv2 [72]	07++12+coco	88.8	87.0	77.8	64.9	51.8	85.2	79.3	93.1	64.4	81.4	70.2	91.3	88.1	87.2	81.0	57.7	78.1	71.0	88.5	76.8	78.2
SSD300 [71]	07++12+coco	91.0	86.0	78.1	65.0	55.4	84.9	84.0	93.4	62.1	83.6	67.3	91.3	88.9	88.6	85.6	54.7	83.8	77.3	88.3	76.5	79.3
SSD512 [71]	07++12+coco	91.4	88.6	82.6	71.4	63.1	87.4	88.1	93.9	66.9	86.6	66.3	92.0	91.7	90.8	88.5	60.9	87.0	75.4	90.2	80.4	82.2
R-FCN (ResNet101) [65]	07++12+coco	92.3	89.9	86.7	74.7	75.2	86.7	89.0	95.8	70.2	90.4	66.5	95.0	93.2	92.1	91.1	71.0	89.7	76.0	92.0	83.4	85.0

Şekil 1.1 CNN tabanlı modellerin VOC2012 üzerindeki sonuçları [18]

modelleridir. Şekil 1.1'de CNN tabanlı modellerin PASCAL VOC (Visual Object Class) 2012 veri seti ile eğitim ve test sonuçları gösterilmiştir. [18]

Nesne tespitinde birçok yardımcı araç kullanılmaktadır. Bunlardan birisi de son zamanlarda kullanımı oldukça genişleyen İHA'lardır. Birçok problemde veri seti eksikliğinden kaynaklı yetersizlikler görülmektedir. İHA'lar bu konuda son zamanlarda oldukça sık başvurulan bir yöntem olarak izlenmektedir. Gallo ve arkadaşları [19], tarım alanında sıkça rastlanan yabancı ot problemi için bir nesne tespit algoritması geliştirmeyi amaçlamışlardır. Fakat tespit etmek istedikleri yabancı otların, bir derin sinir ağı eğitecek kadar yeterli miktarda veri seti bulunmamaktaydı. İHA'lardan elde ettikleri 3.000'den fazla görüntü ile yaklaşık 12.000 kadar etiketleme yaparak yabancı ot veri seti oluşturarak başladılar. Ardından elde ettikleri veri setlerini YOLO-v7 modelini eğitmek için kullandılar. Bu eğitim sonucunda mAP@0,5 skorunda %61 doğruluk oranına ulaşmıştır. Başka bir çalışmada Wang ve arkadaşları [20], İHA'lardan elde edilen görüntülerin işlenebilmesi için YOLO-v8 modelini ele alarak optimizasyonlar uygulamıştır. Uygulanan bu optimizasyonlar sayesinde diğer modellere oranla %7,7'lik bir doğruluk artışı elde etmeyi başarmıştır.

Zeng ve arkadaşları [21], YOLOv7 temel alınarak optimize edilmiş bir nesne tespiti modeli olan "UAV-YOLOv7" önermişlerdir. UAV-YOLOv7 modeli, Wise-IoU (WIoU) v3 isimli bir sınırlayıcı kutu regresyon kaybı kullanır ve modelin lokalizasyon yeteneğini iyileştirmek için özenli bir gradyan tahsisi stratejisi içerir. Ayrıca, BiFormer adlı bir dikkat mekanizması, modelin önemli bilgilere daha iyi odaklanmasını sağlamak için arka plan ağını optimize etmek için tanıtılmıştır. Buna ek olarak, FFNB adlı bir özellik işleme modülü tasarlanmış ve bu modüle dayalı olarak iki yeni tespit ölçüğü önerilmiştir. Bu çok ölçekli özellik birleştirme ağı, modelin tespit performansını önemli ölçüde artırmış ve küçük nesnelerin kaçırılma oranını azaltmıştır. Deney sonuçları, UAV-YOLOv7 modelinin daha az parametreye sahip olduğunu ve YOLOv7'ye göre ortalama tespit doğruluğunda daha yüksek bir başarı elde ettiğini göstermektedir. Ayrıca, bu model, VisDrone2019 ve TinyPerson veri setlerinde YOLOv7'yi geçerek daha iyi bir performans sergilemiştir. Bu veri setleri üzerinde YOLOv7'ye kıyasla mAP@0,5 skorunda %2,89'luk bir artış gözlemlenmiştir. YOLOv7'ye kıyasla daha düşük parametre sayısı ve daha yüksek hızda gerçekleştirdiği tespit ile dikkat çeken bu model, küçük nesnelerin tespiti konusundaki etkinliği artırmaktadır.

Diğer yandan Zhao ve arkadaşları [22], deniz arama kurtarma görevlerinde kullanılan nesne tespiti algoritmalarının önemini vurgulayarak YOLOv7-sea adını verdikleri modeli tanıtmışlardır. Bunun yanı sıra kullandıkları SeaDronesee adlı veri kümesi, küçük hedeflere ve büyük deniz yüzeyi girişimine sahip olduğu için genel nesne tespitleri için zorluklar içermektedir. YOLOv7 temel alınarak, özellikle tiny-scale (küçük ölçekli) insanları veya nesnelere tespit etmek için bir tahmin başlığı eklenmiştir. Ayrıca, sahnede dikkat bölgelerini belirlemek için Simple, Parameter-Free Attention Module (SimAM) entegre edilmiştir. YOLOv7-sea'nın performansını daha da artırmak için veri artırma, Test Zamanı Artırma (TTA) ve birleştirilmiş kutu füzyonu (WBF) gibi stratejiler kullanılmıştır. Yapılan testler, YOLOv7-sea'nın mAP@0,5 skorunun %59 olduğunu göstermiştir. Bu, temel model YOLOv7'ye kıyasla yaklaşık %7 daha yüksek bir performans anlamına gelir. Yani önerilen YOLOv7-sea modeli, küçük hedefleri başarıyla tespit edebilen ve deniz arama kurtarma görevlerinde daha etkili bir şekilde kullanılabilen bir nesne tespit algoritması olarak öne çıkmaktadır.

1.3.2 Süper Çözünürlük

Günümüzde görüntü işleme uygulamaları oldukça geniş bir alanda kullanılmaktadır. Fakat alınan görüntüler her zaman işlenmeye uygun veya probleme yönelik uygulanacak çözümlerde kullanılacak kalitede bir görsel olmayabilir. Buna sebep olarak çözünürlüğü düşük, bulanık veya bozunuma uğramış olması gibi durumlar örnek verilebilir. Görselleri daha işlenebilir hale getirebilmek adına süper çözünürlük, bulanıklık giderme [23] gibi algoritmalar ortaya atılmıştır. Bu algoritmalar geliştirilerek günümüze kadar gelmiş ve hala gelişim sürecine devam etmektedir. Yaygın olarak kullanılan yöntemlerin başında GAN (Generative Adversarial Network) [24] ağları gelmektedir. GAN ağları her ne kadar sentetik veri üretiminde yaygın olarak kullanılsa da çekişmeli ağ yapısı ve yüksek doğruluklara erişebilme yeteneğinden kaynaklı birçok uygulama alanı bulunmaktadır. Bunlardan birisi de çözünürlük artırma uygulamalarıdır. Yüksek verimle çalışması problem özelinde geliştirilen birçok GAN algoritmasının literatüre kazandırılmasını sağlamıştır.

Wang ve arkadaşları [25] bulanıklık giderme algoritmalarını İHA özelinde inceleyerek DUGAN (Denoise UAV Image Generative Adversarial Network) modelini ortaya atmıştır. Bu çalışma İHA'lerden elde edilen görüntülerdeki gürültü ve bulanıklıkları GAN ağları yardımıyla gidermeyi amaçlamıştır. Diğer bir İHA görüntüleri üzerinden çözünürlük iyileştirme çalışması olan [26], DCNN (Deep Convolutional Neural Networks) ağlarını kullanarak bir GAN modeli ile süper çözünürlüklü görüntü elde etmeyi amaçlamışlardır. İHA'lar üzerinde elde edilen görüntülerde benzerlik oranının performans metriği olan LPIPS (Learned Perceptual Image Patch Similarity) değeri iki görüntü arasındaki farklılık oranını sunmaktadır yani sıfıra yaklaştıkça iki görselin benzerliği artarken 1'e yaklaşırken görsellerin benzerlikleri azalmaktadır. Guo ve arkadaşları LPIPS değerini düşürmek için yapmış oldukları çalışmada 0.33'lük bir değer elde ederek benzer modellerin önüne geçmiştir. Bunu sağlarken kayıp fonksiyonlarını güncelleyerek ve gerekli optimizasyonları sağlayarak NDSRGAN (Novel Dense Generative Adversarial Network for Real Aerial Imagery Super-Resolution Reconstruction) adını verdikleri modeli literatür kullanımına sunmuşlardır.

2017 yılında NTIRE2017 [27] süper çözünürlük yarışmasının kazananı olan ve benzer şekilde DCNN ağlarını kullanarak EDSR (Enhanced Deep Super-Resolution Network) modelini literatüre kazandıran Lim ve arkadaşları [28] süper çözünürlükte bir performans metriği olan PSNR (Peak Signal to Noise Ratio) değerinin 29.38 dB 'ye ulaşmasını sağlayarak Bicubic interpolasyon gibi diğer süper çözünürlük yöntemlerinin önüne geçmeyi başarmıştır. Bu modeli oluştururken eğitim veri seti olarak 2K çözünürlüğünde görseller içeren DIV2K veri setini kullanmışlardır. Fakat kullanılan GAN ağları genellikle yapısı nedeniyle yüksek hesaplama gücü ve sonuç olarak da eğitim süresinin uzamasına sebebiyet vermektedir. Genellikle eğitimleri sırasında yüksek sayıda parametre ve derin katmanları kullanması sayesinde doğruluk oranı yüksek sonuçlar verebilmekte. Gupta ve arkadaşları yapmış oldukları çalışmalarında GAN eğitimi için kullanılan katman sayısını azaltarak ve parametre sayısını neredeyse yarisına kadar azaltarak benzer PSNR ve LPIPS değerlerini elde etmeyi başarmışlardır. Katman sayısının ve parametre sayısının daha az olması sebebiyle eğitim süresi oldukça kısalmış ve elde edilen sonuç olarak da çok fazla bir kayba neden olmamıştır. Bu modellerine RSRGAN (Rapid-SRGAN) ismini vermişlerdir.

GAN ağırları kullanılarak elde edilen ilk süper çözünürlük modeli olan SRGAN (Super Resolution GAN) 2017 yılında Leding ve arkadaşları [29] tarafından yayınlanmıştır. Daha sonra bu modelin kayıp fonksiyonlarını ve katman yapısını optimize eden Wang ve arkadaşları [30] ESRGAN (Enhancement SRGAN) modelini literatüre kazandırmışlardır. Küçük nesne tespitinde başarı oranını artırmaya yönelik diğer bir çalışmada [31], CSRGAN ismini verdikleri model ile performans iyileştirmesi yapmayı başarmışlardır. Yine benzer şekilde İHA'lerden elde edilen düşük çözünürlüklü görüntüler üzerinden yapılan nesne tespitini geliştirebilmek adına Xing ve arkadaşları [32] benzer bir çalışma yaparak SRGAN ile elde ettikleri SR görüntüleri üzerinden test etmiş olup doğruluk oranını (mAP@0,5) %64,82'den %68,38 oranına yükseltmeyi başarmışlardır. SRGAN ve ESRGAN modellerindeki yüksek frekanslı detayların kaybolma durumuna yönelik geliştirme çalışmaları devam ederken NITRE2020 yarışmasında birinci olan [33] modeli literatüre sunulmuştur. Bu model görselleri yüksek çözünürlüklü hale getirirken yüksek frekanslı detayların kaybolmasının engelleyecek şekilde mimari yapısı şekillenmiştir. Diğer yandan farklı bir çalışmada [34], asıl bilgi kaybının nesnelere kenar bilgileri üzerindeki bozunumdan kaynaklandığını öne süren bu çalışma ESRGAN (Edge-Enhanced Super-Resolution GAN) modelini literatüre kazandırmıştır. Sundukları bu çalışmalarında nesnelere kenar bilgilerini kaybetmeden çözünürlüklerini artırmak için iki farklı GAN ağı kullanmışlardır. Bunlardan ilki EEGAN (Edge Enhancement GAN) ile kenar bilgilerini yükseltir ve diğer GAN ağı olan ESRGAN görselin çözünürlüğünü artırma işlemini gerçekleştirir. Bu işlemler sonucunda kenar bilgisi kaybı azaldığından nesne tespit oranı artırılmış bulunmaktadır. Fakat iki ayrı GAN ağı kullanmak işlem yükünü arttıracığından kıyasla daha yavaş bir model olacaktır. 2021 yılında Zou ve arkadaşları [35] da benzer bir çalışma yaparak kenar tespitinin nesne tespiti için ne kadar önemli olduğunu göstermiştir. Çalışmalarında SRGAN modeli ile birlikte kenar tespiti tekniklerini kullanarak güçlendirmişlerdir. Nesne tespiti için YOLOv3 modelini kullanmışlardır. Ayrıca YOLO'nun görselleri işleme alırken 640x640 olarak sıkıştırmasından kaynaklı veri kaybının da önüne geçebilmek için her bir görseli 4'e bölerek nesne tespiti oranını arttırmayı hedeflemişlerdir. Sonuç olarak doğruluk oranının arttığını gözlemlemişlerdir.

Yakın tarihli farklı bir çalışma da ise Wang ve arkadaşları [36], tamamen sentetik verilerle eğitilen bir modelin gerçek dünya görselleri üzerindeki etkisini ortaya koyan Real-ESRGAN modelini literatüre katmışlardır. SRGAN ve ESRGAN ile kıyaslandığında daha iyi sonuçlar elde edildiği makalede yer almaktadır.

Endüstri alanında nesne tespiti ile işlemler yapmak son zamanların vazgeçilmez bir yöntemi haline gelmiştir. Özellikle çelik endüstrisinde en küçük bir bozunumun dahi önemli olduğunu göz önüne alırsak buradaki nesne tespit işlemi hayli hassas olmak zorundadır. Bu yüzden süper çözünürlük tekniklerinin kullanım alanlarından birisi de budur. Akhyar ve arkadaşları [37] bu probleme odaklanan çalışmalarını 2022 yılında literatüre katmışlardır. Yapmış oldukları bu çalışmada EnsGAN ismini verdikleri bu modelde özellik çıkarımı için yeni bir teknik ortaya atmış ve çelik görüntüleri içeren veri seti üzerinde yapılan testlerde doğruluk oranını artırmayı başarmışlardır. Farklı bir kullanım alanı olarak İHA görüntüleri ile motosiklet sürücülerinin kask takıp takmadığı kontrol edilebiliyor. [38] bu çalışmada motosiklet üzerindeki kask takma durumu tespit edilmektedir. Görüntü İHA'lardan elde edildiğinden yine ESRGAN kullanılarak çözünürlük artırılarak doğruluk oranının artırılması hedeflenmiş ve %94 doğruluk oranına erişilmiştir. Yine benzer olarak İHA'lardan elde edilen asfalt yol görüntüleri üzerinde zamanla oluşan çukurların tespiti için de ESRGAN-YOLOv5 birlikte kullanılarak tespitteki doğruluk oranını artırılması hedeflenmiştir [39]. Endüstri alanında ayrıca yazılar üzerinden de işlemler ağırlıktadır. Bu da oldukça fazla bir iş yüküne sebebiyet vermektedir. Bunun için yazıları otomatik okuyup işleyebilen algoritmalar geliştirilmektedir. Fakat bu yazılar bazen çözünürlük düşüklüğünden dolayı yanlış okunabilmektedir. Buna yönelik yapılan bir çalışmada Tsai ve arkadaşları [40] EESRGAN (Efficient & Effective SRGAN) adını verdikleri modeli literatüre kazandırmışlardır. Bu çalışmada ESRGAN modeli ile çözünürlüğü artırılan görseller Google OCR (Optical Character Recognition) algoritması kullanılarak yazıya dönüştürülmüştür. Benzer bir problem denizcilikte gemilerin plaka okuma sistemlerinde de görülmektedir [41]. Bu plakaların uzaktan ve küçük olması sebebiyle her zaman net bir şekilde okunamamaktadır. Wu ve arkadaşları da benzer bir çözümle ESRGAN modeli ile yazıların daha yüksek doğruluklarla okunmasını hedeflemişlerdir.

İletim hatlarındaki elektrik direkleri üzerinde yapılan çalışmalar oldukça fazla işgücü gerektirmektedir ve tehlike boyutu da hayli yüksektir. Örneğin iletim hatlarındaki kabloların en önemli iş yüklerinden bir tanesi de bağlandıkları direğin civatasının zamanla gevşeme veya düşme ihtimaline karşı tedbir amaçlı kontrol edilmesi durumudur. Lv ve arkadaşları [42], bu problem üzerinde yoğunlaşarak İHA'lerden elde edilen görüntülerle eksik civata tespiti yapmayı hedeflemektedirler. Ancak küçük bir nesne olduğundan çözünürlüğü düşük olacağından Real-ESRGAN modeli ile çözünürlüğü arttırılmıştır. Ardından elde edilen SR görüntüler YOLOv5 modeli ile eğitilerek nesne tespiti yapılmıştır. Tüm bu işlemlerin sonucunda mAP@0,5 doğruluk oranı %1,5 arttırılarak %73,8'e ulaşmıştır.

Jabberi ve arkadaşları [43] çalışmalarında GAN ağının birden fazla modeli farklı amaçlar için kullanılmıştır. DCGAN (Deep Convolutional GAN) modeli ile farklı yüzleri üretirken ESRGAN ile de üretilen bu yüzlerin çözünürlüklerini arttırılması hedeflenmiştir.

GAN ağları genellikle karmaşık ağlardan oluştuğundan dolayı eğitim ve test süreleri uzamaktadır. Bu yüzden gerçek zamanlı süper çözünürlük uygulamaları için hem enerji tüketimi hem de işlem süresi gibi parametreler göz önüne alınarak yapılması gerekmektedir. Vedavyas ve arkadaşları [44] bu çalışmalarında, FPGA kullanarak gerçek zamanlı bir süper çözünürlük uygulaması yapmışlardır. Yine benzer olarak gerçek zamanlı süper çözünürlük uygulamaları için Wang ve arkadaşları da [45] çalışmasını literatüre kazandırmışlardır. Zhu ve arkadaşları [46] çalışmada ise GAN ağlarının eğitim süresinin kısaltılması hedefine odaklanmıştır. Bunun için Real-ESRGAN tabanından oluşan fakat optimizasyonlar ile eğitim süresinin yaklaşık olarak %28 oranında kısaldığı PatchGAN modelini literatüre sunmuşlardır.

Tarımsal alanda da nesne tespiti ve sınıflandırması uygulamaları kullanılmaktadır. Genellikle bitkiler üzerinde sınıflandırma yapılan mimariler daha sıklıkla kullanılmaktadır. Bu sınıflandırma uygulamaları daha çok hastalık tespiti ve sınıflandırması durumları için sıklıkla kullanılmaktadır. Örneğin [47] çalışmada domates bitkisinin yaprakları üzerindeki hastalıkları tespit etme ve

sınıflandırma işlemlerinde çözünürlüğün etkisini araştırmışlardır. İHA'lerden elde ettikleri görüntüleri SRCNN (Super Resolution Convolutional Neural Network) mimarisi kullanılarak çözünürlüğünü arttırmışlardır. Yaptıkları testler sonucunda yüksek çözünürlüklü görsellerdeki sınıflandırma oranının daha yüksek seviyelerde olduğunu göstermişlerdir. Benzer bir çalışmada [48], buğday bitkisindeki hastalıklar tespit edilip sınıflandırmak amaçlanmaktadır. Burada SRGAN modeli ile düşük çözünürlüklü veri setleri yüksek çözünürlüklü hale getirerek %75 olan doğruluk oranı %83 oranına yükseldiği gözlemlenmiştir. Diğer bir çalışma [49] da zeytin ağacının verimi, büyümesi gibi parametreleri İHA'larla takip eden bir çalışma olarak literatüre girmiştir. ESRGAN modeli kullanılarak yapılan süper çözünürlük sonrasında %95'lik bir doğruluk oranına erişilmiştir. Modelleri eğitmek için gerekli olan veri setleri her zaman yeterli olmayabilir. Örneğin [50] çalışmasında pirinçler üzerindeki hastalıkların sınıflandırılması amaçlanmıştır. Fakat bu sınıflandırma için gerekli olan modeli eğitebilmek adına yeterli hastalıklı yaprak görüntüsünün bulunmaması üzerine Zhang ve arkadaşları eldeki verilerle GAN ağını eğiterek sahte hastalıklı yaprak verileri üretmişlerdir. Ardından elde edilen yeni görüntüleri ve orijinal görüntüleri Opt-Real-ESRGAN (Optimized-Real-ESRGAN) modeli ile süper çözünürlüklü hale getirerek sınıflandırma işlemini gerçekleştirmişlerdir. Bu işlemlerin sonucunda ortalama %4'lük bir doğruluk oranı artışı gözlemlenmiştir.

Orman yangınlarını tespit etmede de İHA'lar önemli rol oynamaktadır. [51] çalışmasında Real-ESRGAN modeli ile yangın özelinde geliştirilen fireYOLO modeli birlikte kullanılarak YOLOv5'e kıyasla doğruluk oranında %21 oranında bir artış sağlamayı başarmışlardır.

1.4 Tezin Yapısı

Bu tez 6 ana bölümden oluşmaktadır. İlk bölümde problem ve amaç tanımları yapılmış, literatür taraması yapılarak ilgili makaleler nesne tespiti ve süper çözünürlük başlıkları altında toplanmıştır. Bölüm-2' de süper çözünürlük uygulamalarında kullanılan mimariler ve GAN ağlarının genel yapıları anlatılmıştır. Bölüm-3'te Derin Sinir Ağlarının nesne tespiti için kullanılan model mimarisi anlatılmıştır. Bölüm-4'te önerilen yöntemin yapısından ve bu yöntemi oluştururken kullanılan materyallerden bahsedilmiştir. Bölüm-5'te ise önerilen yöntemin eğitim ve test sonuçları sunularak karşılaştırma yapılmıştır. Son bölümde ise çalışma ile ilgili sonuçlar değerlendirilmiştir.



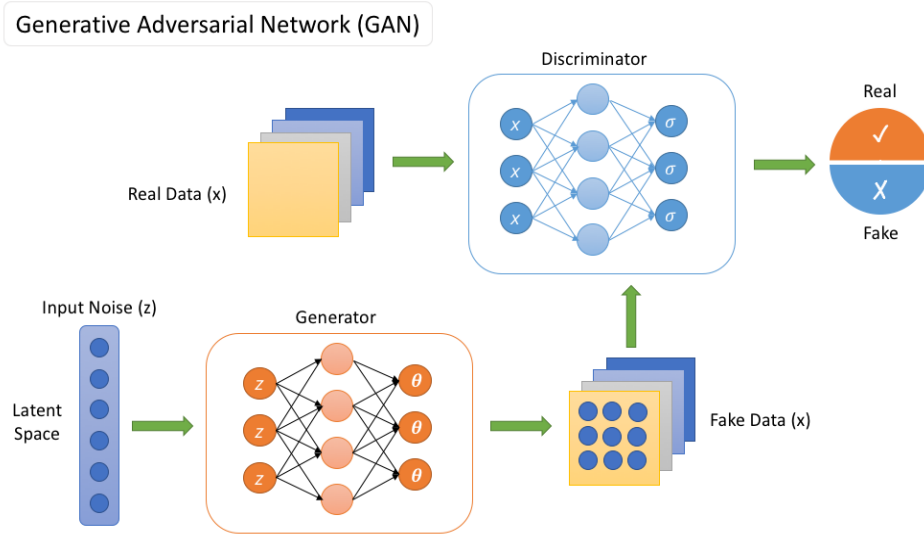
ÇEKİŞMELİ ÜRETİCİ AĞLAR (GAN)

Süper Çözünürlük uygulamaları temelde düşük çözünürlüklü görüntülerden yüksek çözünürlüklü görüntüler elde etmeyi hedefleyen uygulamalardır. Bu doğrultuda birçok yöntem kullanılmıştır. Bikübik İnterpolasyon, düşük çözünürlüklü bir görüntüyü daha yüksek çözünürlükte bir görüntüye dönüştürmek amacıyla kullanılan bir interpolasyon tekniğidir. Bu yöntem, her çıkış pikselinin değerini, çevresindeki dört pikselin bilinen değerleri üzerinden hesaplar. Adından da anlaşılacağı gibi, bu hesaplama sürecinde bir pikselin çevresindeki dört piksel arasındaki değerlerin ağırlıklı ortalaması alınır. Fakat bu yöntemde veri kaybı oldukça fazladır. Benzer olarak En Yakın Komşu yöntemi de en yakın komşu pikseli ile eşitleyerek çözünürlüğü yükseltmeyi hedefler. Fakat burada da veri kaybı yaşanabilmektedir. Klasik yaklaşımların birçoğunda benzer yöntemler uygulanmaktadır. Derin öğrenme yöntemlerinin süper çözünürlük uygulamalarında kullanılmaya başlamasıyla bu veri kaybı azaltılmış ve daha yüksek çözünürlüklü görüntüler elde etmenin önü açılmıştır. Bu çalışma kapsamında derin öğrenme tabanlı süper çözünürlük yöntemleri incelenmiştir.

2014 yılında Goodfellow tarafından literatüre kazandırılan GAN ağları [24] tanıtıldığı günden bu yana büyük ilgi ile karşılanmıştır. Mimari yapısında Üretici (G) ve Ayırıcı (D) olarak iki temel ağ bulunmaktadır. Bu modeller sürekli bir geri besleme ve gelişim halindedirler. Üretici ağı gürültü sinyallerinden yeni bir görüntü oluşturmayı hedeflerken; Ayırıcı ağı, oluşturulan bu görüntünün gerçek mi sahte mi olduğunu sınıflandırmayı hedefler. Sonuç olarak bu oyun devam eder ve Üretici tarafından üretilen veri, Ayırıcı ağı tarafından ayırt edemeyecek seviyeye geldiğinde modelin yeterince eğitildiği görüşü çıkarılabilir. Şekil 2.1'de GAN modeli gösterilmektedir.

GAN ağları çeşitli amaçlara yönelik olarak kullanılabilir. Örneğin yeterince eğitildiği takdirde istenilen türde sahte bir veri üretimi başarabilir. Farklı bir kullanım alanı olarak veri setlerinin eğitim için yetersiz olduğu durumlarda sentetik veriler üreterek veri setinin kullanılabilir hale getirilmesi

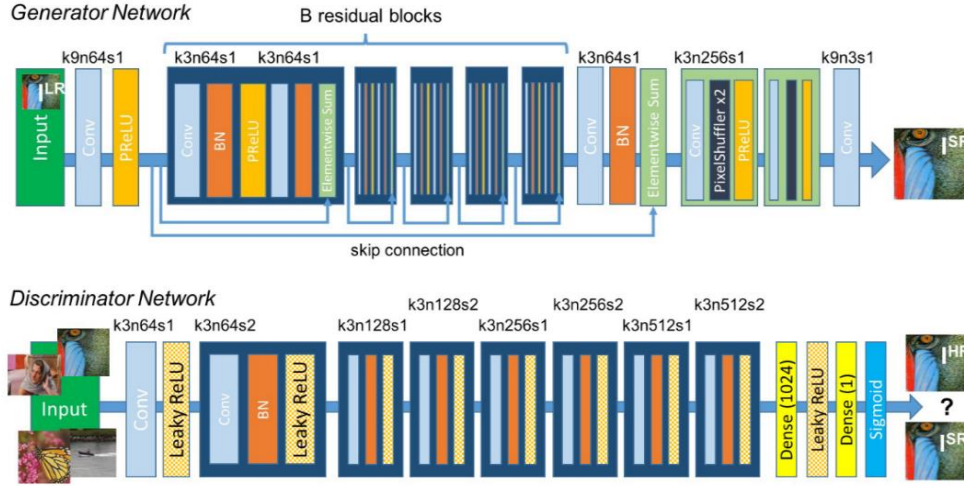
gösterilebilir. Diğer bir yaygın kullanım alanı da süper çözünürlük uygulamalarıdır. Düşük çözünürlüklü görüntülerden yüksek çözünürlüklü görüntüler elde etme noktasında oldukça başarılıdır ve literatürde de sıklıkla başvurulmaktadır. Süper çözünürlük yöntemlerinde birçok farklı GAN mimarisi oluşturulmuştur. SRGAN, ESRGAN ve Real-ESRGAN modelleri bu mimarilerin en sık kullanılanlarıdır.



Şekil 2.1 GAN ağının mimari yapısı [52]

2.1 SRGAN

SRGAN, süper çözünürlük görevleri için tasarlanmış bir derin öğrenme modelidir. Bu model, düşük çözünürlüklü görüntüleri yüksek çözünürlüklü görüntülere dönüştürmek amacıyla GAN mimarisini kullanır. Normalde Üretici ağı başlangıçtaki gürültü sinyalinden yeni veri üretme ve Ayırıcı ağı da üretilen bu verinin sahte mi gerçek mi olduğunu ayırt etmeye çalışmaktaydı. Fakat süper çözünürlük uygulamalarında kullanılan Üretici ağı düşük çözünürlüklü görselleri girdi olarak alarak yüksek çözünürlüklü çıktılar verecek şekilde işlemektedir.

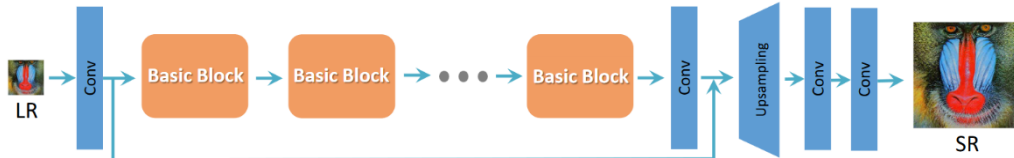


Şekil 2.2 SRGAN üretici ve ayırıcı ağlarının mimarisi [29]

Ayırıcı ağı ise üretilen yeni görüntünün düşük çözünürlüklü veya yüksek çözünürlüklü olması durumuna karar verir. Şekil 2.2’de Üretici ve Ayırıcı ağlarının yapısı gösterilmektedir.

2.2 ESRGAN

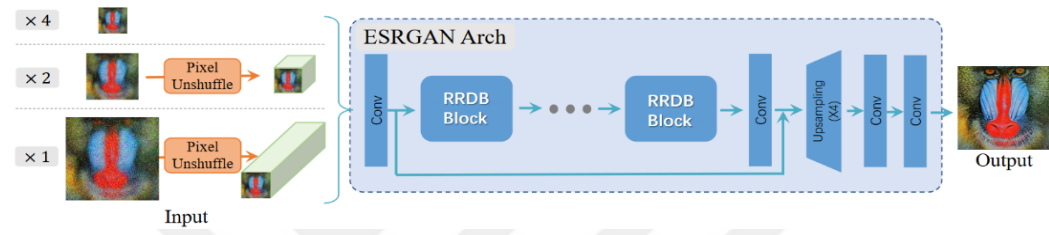
ESRGAN modeli (Şekil 2.3) SRGAN modelinin geliştirilmiş bir halidir. Mimari olarak aralarında birçok benzerlik bulunur. En büyük farklardan birisi ise SRGAN mimarisinde bulunan BN (Batch Normalization) katmanı ESRGAN modelinde kaldırılmıştır. Bunun yerine RRDB (Residual in Residual Dense Block) katmanı eklenmiştir. Bunun sebebi BN katmanında yapılan normalizasyon işleminin bilgi kaybına yol açması olarak gösterilmiştir.



Şekil 2.3 ESRGAN mimarisi [29]

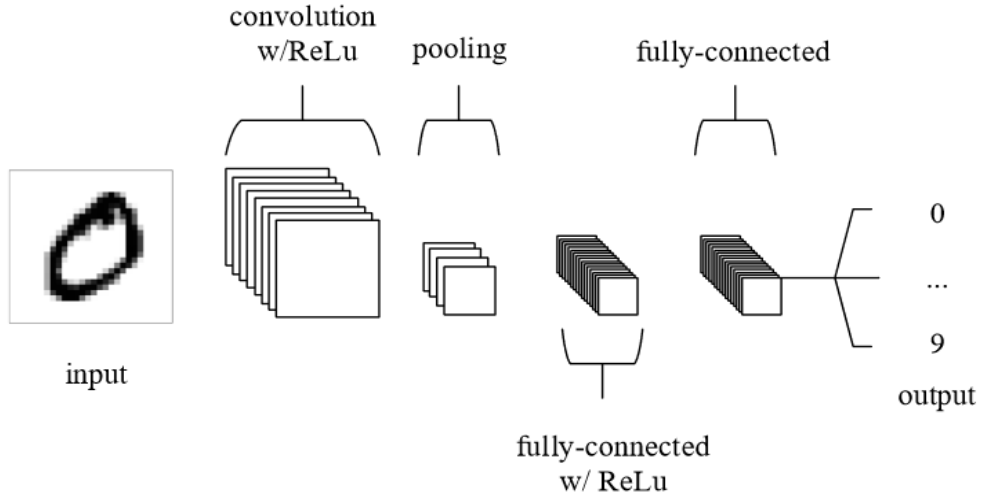
2.3 Real-ESRGAN

ESRGAN süper çözünürlük uygulamalarında oldukça başarılı sonuçlar vermiştir. Fakat gerçek görüntüler üzerindeki çeşitli yüksek seviyeli gürültü ve bozunumları düzeltmede yetersiz kalabilmektedir. Bu problem üzerine yapılan çalışmalar sonucunda Real-ESRGAN modeli sunulmuştur. Bu modelin en önemli özelliği tamamen saf sentetik verilerle eğitilmiş olmasından kaynaklı olarak gerçek dünya görüntüleri üzerindeki bozunum ve gürültüleri düzeltme konusunda daha iyi sonuçlar vermektedir. Şekil 2.4'te bu modelin mimarisi sunulmuştur.



Şekil 2.4 Real-ESRGAN mimarisi [36]

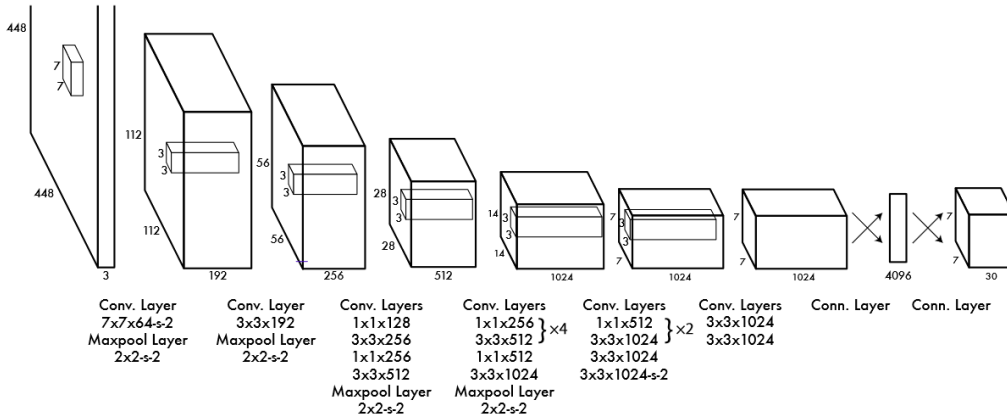
Evrişimli Sinir Ağları (CNN) Yapay Sinir Ağlarından (ANN) esinlenerek ortaya çıkan nesnelere daha iyi tanınmasını sağlayabilen, derin katmanlara sahip bir modeldir. Bu model nesnelere eşsiz özelliklerine odaklanabilecek birçok filtrenin kullanılabildiği katmanlara sahiptir. Örneğin bir filtre ile köşe bilgisini elde ederken farklı bir filtre ile şekil tespiti yapılabilmektedir. Tüm bu özellik çıkarımları sonunda bir nesnenin birçok özelliğini öğrenmekte ve tespit ederken çok daha başarılı olabilmektedir. Oldukça derin katmanlar oluşturabilmesi kompleks özelliklere sahip nesnelere daha iyi tanınmasını sağlamaktadır. Bu sayede birçok nesne tespit uygulamasında bu yapıya başvurulmaktadır. Temel olarak üç yapıya sahiptir. Bunlardan ilki nesnelere özellik çıkarımlarının yapıldığı ve çeşitli filtreler kullanılarak ağı derinlik katıldığı katman olan Evrişim katmanıdır. İkinci temel katmanı ise Havuzlama (Pooling Layer) Katmanıdır. Bu katmanda ağırlıkları minimum, maksimum veya ortalama olmak üzere üç farklı seçenek ile günceller ve son katmana giriş için uygunluğunu kontrol eder. Son olarak da elde edilen ağırlıkların hangi nesneye işaret ettiğini yani sınıflandırmasını yapan katman olan Sınıflandırma (Fully-Connected Layer) Katmanına sahiptir. Şekil 3.1’de bu katmanlar gösterilmiştir.



Şekil 3.1 Evrişimli Sinir Ağlarının genel mimarisi [52]

YOLO

YOLO (You Look Only Once), CNN mimarisi üzerine kurulmuş nesne tespit ve takip uygulamalarında kullanılan bir modeldir. YOLO modeli isminden de anlaşılacağı üzere görüntüyü tek seferde işlemektedir. Görselleri sabit bir boyuta sıkıştırarak ve ardından eşit parçalara bölerek ayrı ayrı CNN katmanına gönderip nesne tespitini gerçekleştirmektedir. Diğer bir ifadeyle tek seferde tüm resmi tarayarak nesnelere tespit etmektedir. Bu yüzden diğer nesne tespit uygulamalarıyla kıyaslandığında daha hızlı bir sonuç verdiği görülmektedir.



Şekil 3.2 YOLO genel mimarisi [16]

Şekil 3.2’de bahsedilen YOLO mimarisinin genel şeması verilmiştir. YOLO'nun evrimi boyunca her versiyon, öncekine göre daha gelişmiş performans ve doğruluk sunmaya odaklanmıştır. YOLOv1, nesne tespiti için tek bir “end-to-end” derin sinir ağı kullanarak gerçek zamanlı tespiti sağlamıştır, ancak küçük nesnelere tespit konusunda zayıf kalmıştır. YOLOv2, YOLOv1’in performansını artırmak için "anchor boxes" gibi yenilikler getirerek özellikle küçük nesnelere daha iyi tespit etmeye odaklanmıştır. YOLOv3, birden fazla ölçekli tahminler gibi tekniklerle daha fazla ölçekli nesne tespiti sağlamış ve daha büyük veri setleriyle eğitilerek doğruluğu artırmıştır. YOLOv4, YOLOv3’ün performansını daha da artırarak daha hızlı ve doğru sonuçlar elde etmek için çeşitli iyileştirmeler yapmıştır. Son olarak, YOLOv7 ve YOLOv8, daha gelişmiş performans ve doğruluk sunmak için daha büyük veri setleri ve karmaşık mimarilerle eğitilmiş ve hız ve doğruluk dengesini daha iyi optimize edilmişlerdir. Bu çalışmada YOLO mimarisinin V7 ve V8 modelleri kullanılmıştır. Kullanılan backbone yapısı ve diğer model bilgileri Deneysel Çalışmalar bölümünde detaylandırılmıştır.

Bu tez kapsamında yapılan çalışmalar düşük çözünürlüklü görüntülerde nesne tespitinin iyileştirilmeye çalışmasını temel amaç olarak almıştır. Elde edilen görüntülerin nesne tespitini zorlaştıran birçok parametresi bulunmaktadır. Örneğin sensör gürültüleri, hava şartları, uzaktan görüntü alma gibi birçok sebep görüntülerin bozunuma uğramasına sebep olabilir. Bu problem hiç şüphesiz ki en çok havadan alınan görüntülerde görülmektedir. Elde edilen bu görüntülerde çoğunlukla nesnelere küçük ve bulanık olarak görülmektedir. Bu da nesne tespitini oldukça zorlaştırmaktadır. Bu sebeple çalışma alanı olarak havadan alınan görüntülerden oluşan VisDrone ve SDD (Stanford Drone Dataset) veri setleri seçilmiştir.

4.1 Veri Setleri

4.1.1 VisDrone

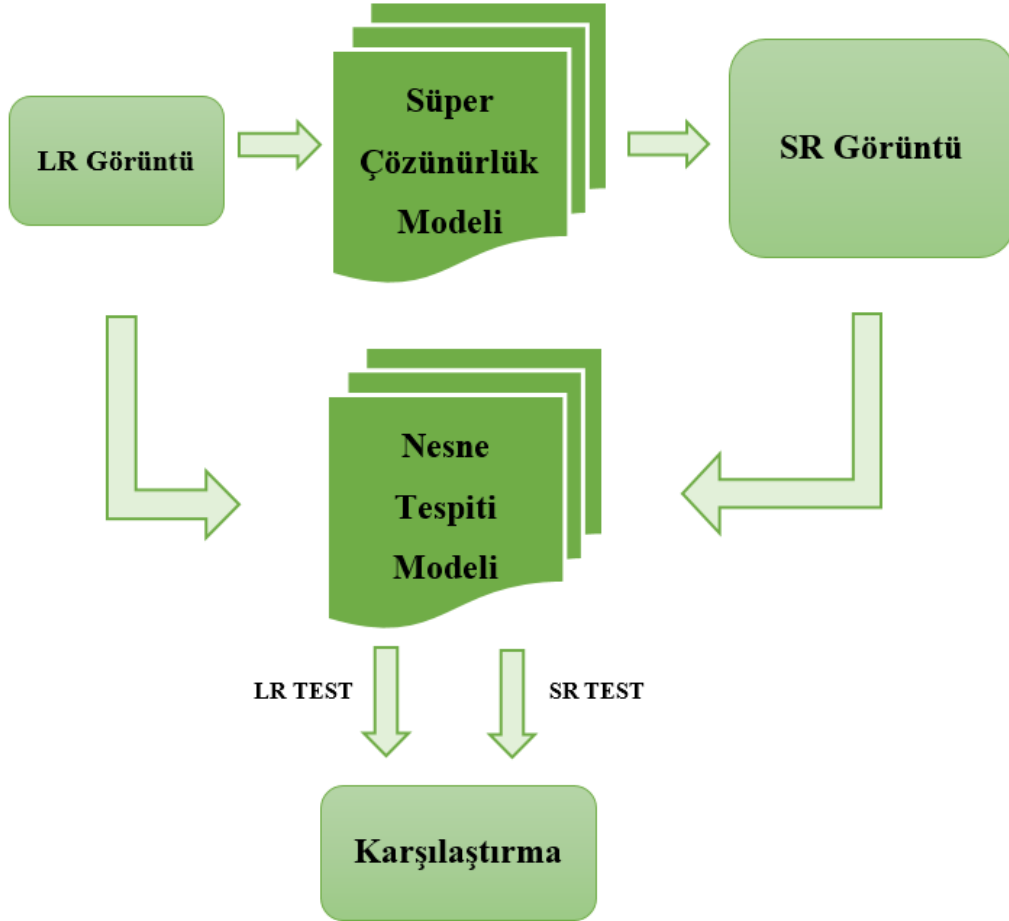
VisDrone veri seti [52] 2018 yılında Zhu ve arkadaşları tarafından yayınlanan bir veri setidir. Bu veri seti İHA'lar ile elde edilmiş görüntülerden oluşmaktadır. Bu görüntüler insan, araba, bisiklet gibi birçok farklı nesne içermektedir. Bu veri seti kendi alanındaki diğer veri setleri ile kıyaslandığında yüksek bir çözünürlüğe sahiptir. Toplamda 288 videodan oluşan ve 10,209 sabit görüntü içermektedir. Bu veriler nesne tespiti, nesne takibi, kalabalık sayısı ve daha birçok göreve yönelik ayrıştırılmıştır ve geniş bir etiketleme yapısına sahiptir. Bu çalışmada nesne tespiti görevi için özelleştirilen bölümü kullanılmıştır. Bu veri seti 6470 eğitim, 548 validasyon ve 1610 test görseli içermektedir. Bu şekilde verilerin bölümlere ayrılarak sunulması ve zengin etiketlemeye sahip olmasından dolayı VisDrone veri setini oldukça tercih edilebilir bir noktaya getirmektedir.[53]

4.1.2 Stanford Drone Dataset (SDD)

SDD veri seti havadan alınan görüntülerden oluşan bir veri setidir [54]. Yaklaşık 69 GB'lık video deposu ile kendi alanındaki en büyük veri setlerindedir. Stanford Üniversitesi tarafından nesne tespiti ve takibi yarışmalarında kullanılması için oluşturulan bu veri seti birçok farklı ortamdaki görüntü içermektedir. 5270 eğitim, 752 validasyon ve 1464 test görüntüsü içermektedir.

4.2 Yöntem

Önerilen yöntem düşük çözünürlüklü görüntülerdeki küçük nesnelerin tespiti konusunda literatürdeki yöntemlerle kıyaslanarak sunulmuştur. Yöntemin temeli LR görüntülerin GAN ağı kullanılarak SR hale getirildikten sonra, LR ve SR görüntüleri üzerinde ayrı ayrı nesne tespiti yapılarak aradaki farkı karşılaştırmaktır. Şekil 4.1'de sunulan yöntemin akış şeması verilmiştir.



Şekil 4.1 Önerilen mimari

Önerilen mimari, genellikle gözden kaçan küçük nesnelerin tespitine olanak tanımak amacıyla İnsansız Hava Araçlarından alınan görüntülerin çözünürlüğünü artırmayı hedeflemektedir. Bu amaç doğrultusunda, mimari iki ana bölümden oluşmaktadır. İlk olarak VisDrone ve SDD veri setlerindeki LR görüntü, bir GAN bloğu içinde işlenir. Bu işleme için kullanılan model DIV2K, Flickr2K ve OutdoorSceneTraining veri setleri ile eğitilmiş bir Real-ESRGAN ön-eğitilmiş (pre-trained) modeldir. Real-ESRGAN modelinin çıktısında SR görüntü elde edilir. Ardından elde edilen bu SR görüntüler ile LR görüntüler, YOLOv7 ve YOLOv8 mimarileri ile nesne tespiti için kullanılan Nesne Tespiti Modeline ayrı ayrı beslenir. Bu sürecin sonuçları karşılaştırılarak Real-ESRGAN modelinin nesne tespiti üzerindeki etkisi ortaya konulmuş olacaktır.

Real-ESRGAN modelinin performansını değerlendirebilmek için PSNR ve SSIM metrikleri kullanılmıştır. PSNR, elde edilen yeni görüntünün orijinal görüntüden sapma oranını hesaplar (4.1). Fakat sadece piksel bazında bir değerlendirme olduğundan yanılması da kolay olabilmektedir. SSIM metriği de yine benzer şekilde orijinal görüntü ile yeni görüntü arasındaki sapmayı hesaplar (4.2). Fakat bu hesaplamada piksel yerine korelasyon kaybı, parlaklık bozulması ve kontrast bozulması olmak üzere üç faktörün bir kombinasyonu olarak modeller. [0,1] aralığında değer alır. Orijinal görüntüdeki değeri 1'dir.

$$PSNR = 10.10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (4.1)$$

MAX: maksimum piksel değeri

$$MSE = \frac{1}{m.n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

[mxn]: Gürültüsüz resim matrisi

I: Orijinal Görüntü

K: Gürültü Yaklaşımı

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4.2)$$

μ_x : x'in piksel örneklemeesi

μ_y : y'nin piksel örneklemeesi

σ_x^2 : x'in varyansı

σ_y^2 : y'nin varyansı

σ_{xy} : x ve y'nin kovaryansı

c_1, c_2 : Zayıf Payda stabilizasyonu için deęişken



Derin sinir ağlarını eğitip test etmek genelde yüksek işlem gücü gerektirmektedir. Bunun sebebi derin katmanlar oluşturarak eğitimin derinleşerek daha kompleks özelliklerin öğrenilmesi temeline dayanmaktadır. Bu yüzden yüksek işlem gücü sunan işlemciler tercih edilmesi eğitim ve test sürecini kısaltacaktır. Google bu konuda kullanıcılara yüksek işlem gücü sunan sanal makineleri kiraladığı bir Colab ortamı sunmuştur. Bu çalışmada Google Colab Pro+ versiyonu kullanılarak Nvidia A100-SXM4-40Gb GPU and 12-core CPU kullanılmıştır. Bu ortamda Real-ESRGAN modeli kurulduktan sonra VisDrone veri seti SR hale getirilmek için bahsi geçen işlemci kullanılmıştır. Bu model öncelikle ESRGAN modelinin önceden eğitilmiş ağırlıklarını 'L1' kayıp fonksiyonu ile eğiterek Real-ESRGAN'ın Üretici (G) ağını oluşturmuştur. Ardından elde edilen bu üretici başlangıç olarak kullanılarak 'L1, Algısal Kayıp (Perceptual Loss) ve GAN kayıp fonksiyonları kullanılarak eğitim gerçekleştirilmiştir. Eğitim için veri seti olarak DF2K (DIV2K ve Flickr2K) + OST veri setleri kullanılmıştır. Eğitilen bu model "RealESRGAN_x4plus" isimi ile indirilerek çalışmaya dahil edilmiştir. Bu eğitilmiş modeli kullanılarak VisDrone ve SDD veri setleri süper çözünürlüklü hale getirilmiştir. Her bir model eğitimi yaklaşık olarak 36 saat sürerek bütün veri setlerinin çözünürlüğü artırılmıştır. Elde edilen SR görüntüler ve LR görüntüler karşılaştırıldığında süper çözünürlük için performans metriği olan PSNR ve SSIM değerleri sırasıyla ortalama 34 dB ve 0,94 olarak elde edilmiştir. Bu sonuçlara bakıldığında benzerlik oranı korunarak çözünürlük yükseltme işleminin başarı ile gerçekleştirildiğini söylemek mümkündür. Bu sayede LR_VisDrone, SR_VisDrone, LR_SDD ve SR_SDD şeklinde veri setlerinin ikişer versiyonları elde edilmiştir. Daha sonra bu veri setleri kullanılarak nesne tespiti için eğitim ve test adımları gerçekleştirilmiştir.

Bir sonraki adım olarak veri setlerindeki etiketlemelerin eğitime uygun hale getirilmesi hedeflenmiştir. Burada öncelikle çözünürlük artışı sonrasında etiketleme verilerinde herhangi bir değişiklik olup olmadığının testi yapılmıştır. Yapılan test sonrasında LR ve SR görüntülerindeki nesnelerin konum verileri birebir uyduğu görülmüştür. Dolayısıyla aynı etiket verisi iki veri seti için de

geçerli olmaktadır. Bu sonuçlar elde edildikten sonra eğitim için gerekli ön işlemlere geçilmiştir. Nesne tespiti çalışmaları için YOLO ortamı seçildiğinden bu etiket verilerinin YOLO formatına dönüştürülmesi gerekmektedir. YOLO'nun etiket formatı şu şekildedir;

<ClassID, Center_X, Center_Y, Width, Height>

ClassID: Sınıf Bilgisi

Center_X: Nesnenin Merkez koordinatının X verisi

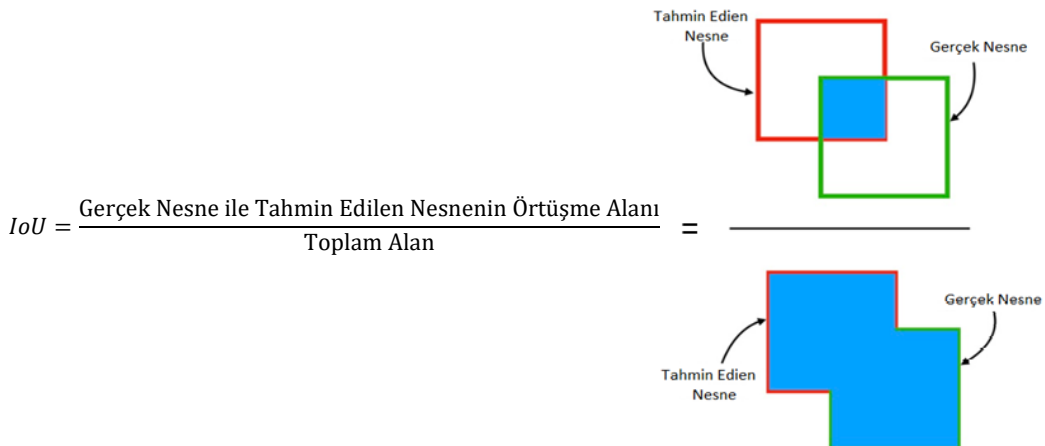
Center_Y: Nesnenin Merkez koordinatının Y verisi

Width: Nesnenin Genişlik Verisi

Height: Nesnenin Yükseklik Verisi

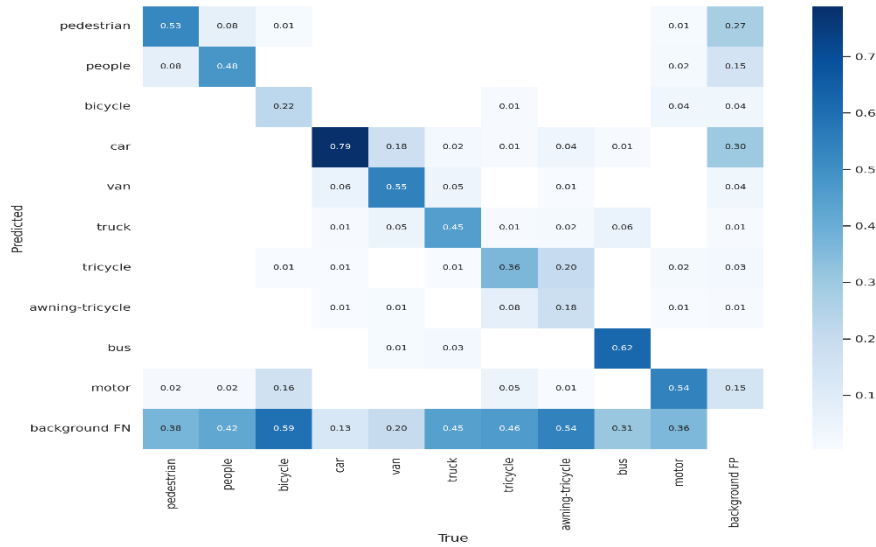
Dolayısıyla VisDrone veri setindeki etiketlerin bu formata dönüştürülmesi gerekmektedir. Python üzerinden yapılan gerekli işlemler sonucunda bu format elde edilerek veri seti eğitime hazır hale getirilmiştir. Elde edilen yeni formattaki etiketlerin doğruluğu veri seti üzerine çizdirilerek doğruluğu test edilmiştir. Bu testler sonucunda da elde edilen veri setinin eğitim için hazır hale getirildiği doğrulanarak eğitim adımına geçilmiştir.

Sonuçlar elde edilirken nesne tespitinde çoğunlukla kullanılan mAP@0,5 (Mean Average Precision) performans metriği kullanılmıştır. mAP performans metriği, tahmin edilen nesne alanı ile gerçek nesne alanının örtüşme oranını kullanarak modelin performansını belirler. Bu örtüşme belirleme işlemine IoU (Intersection Over Union) denir. Eşik (threshold) değeri ise bu örtüşmenin oranının sınırını belirler. Örneğin mAp@0,5 metriğinde eşik 0,5 olarak seçilmiş olup bu oranın üzerindeki örtüşmeler doğru olarak kabul edilir.

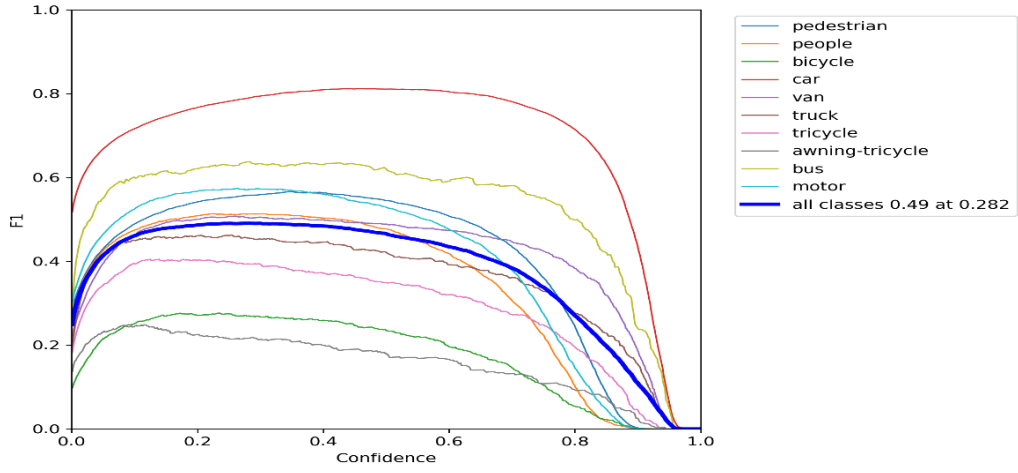


5.1 YOLO-V7 LR_VisDrone ile Eğitim ve Test Sonuçları

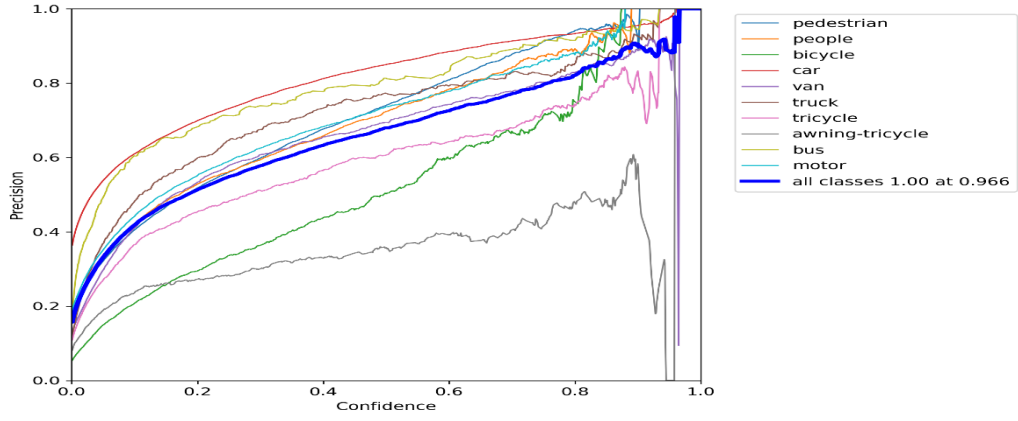
Eğitim ortamı olarak yine Colab üzerindeki Nvidia A100-SXM4-40Gb GPU ve 12-core CPU ortamı kullanılmıştır. Nesne tespitinin test edilmesi için YOLOv7 modeli tercih edilmiştir. Kullanılan YOLOv7 modeli 415 katman ve 37245102 parametreden oluşmaktadır. Eğitim için önceden eğitilmiş bir modelin ağırlıkları başlangıç ağırlığı olarak alınmıştır. Bu model COCO veri setiyle eğitilen YOLOv7 modelidir. Bu ağırlıklar başlangıç ağırlığı olarak alınarak LR_Visdrone veri seti ile eğitim işlemi gerçekleştirilmiştir. Bu eğitim için kullanılan bazı hiperparametreler; Learning Rate: 0.01 (Adam Optimizasyonu) , momentum: 0.937 (Önceki YOLO eğitim verilerine göre optimum momentum değeri), weight_decay: 0.0005, scale: 0,5, anchor_t: 4.0 (anchor-multiple threshold), batch_size:32 (GPU belleğine göre seçilir) şeklindedir. Toplamda 300 adım eğitilen modelin eğitim sonuçları Şekil 5.1-5.7’de gösterilmiştir.



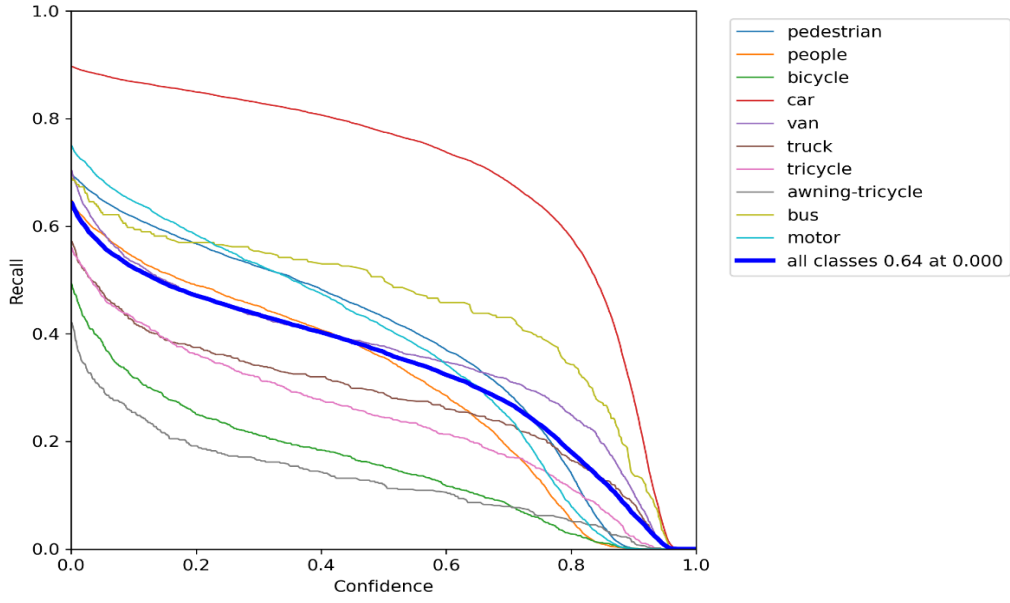
Şekil 5.1 LR_VisDrone ile eğitilen modelin karışıklık matrisi



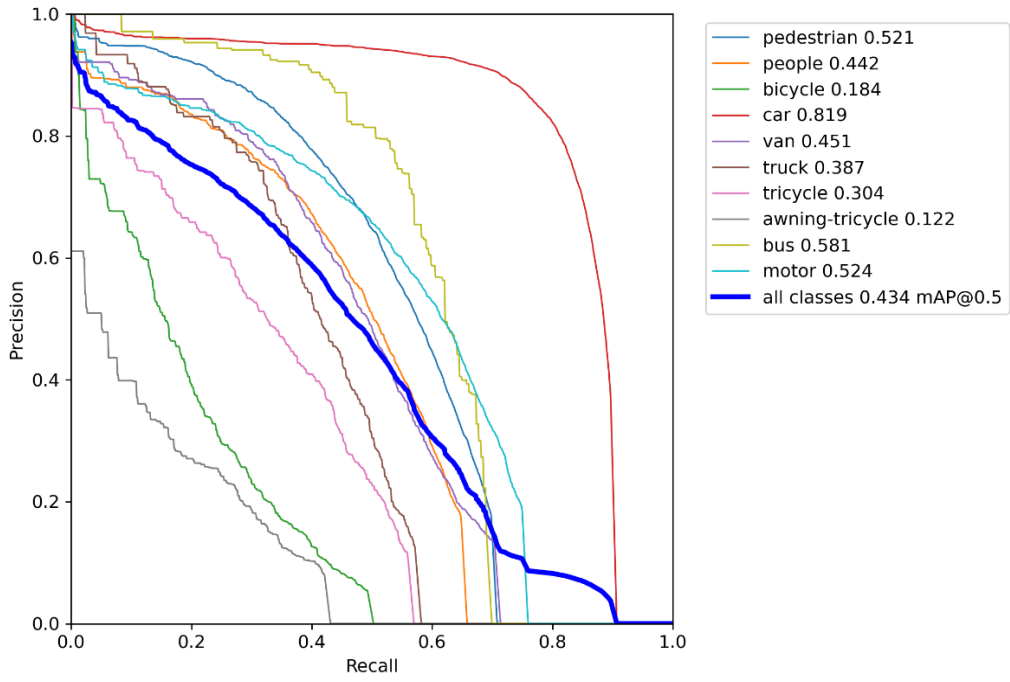
Şekil 5.2 LR_VisDrone ile eğitilen modelin F1 skoru



Şekil 5.3 LR_VisDrone ile eğitilen modelin kesinlik eğrisi



Şekil 5.4 LR_VisDrone ile eğitilen modelin duyarlılık eğrisi



Şekil 5.5 LR_VisDrone ile eğitilen modelin kesinlik-duyarlılık eğrisi

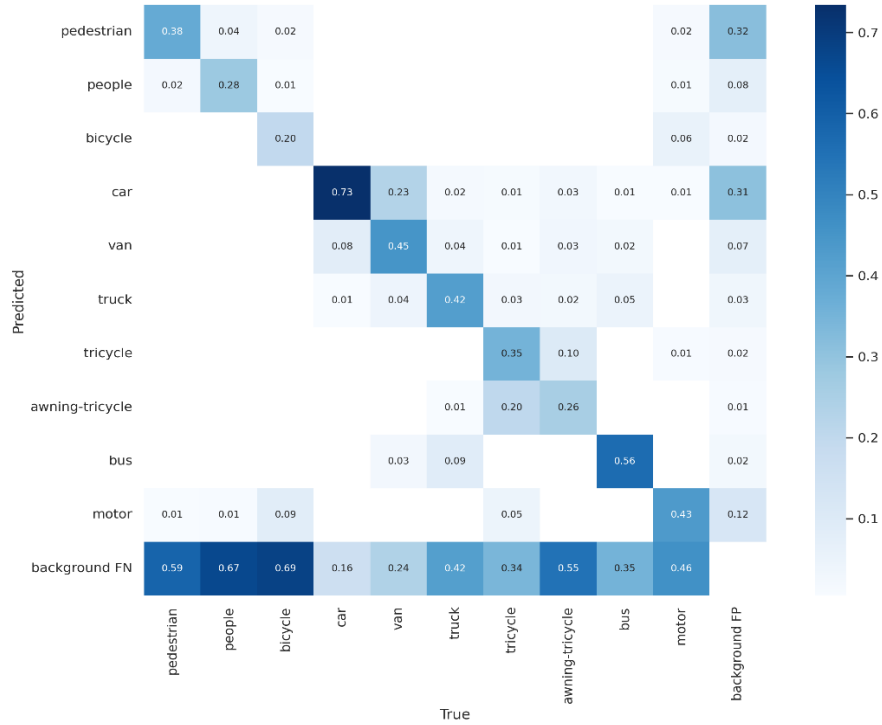


Şekil 5.6 LR_VisDrone ile eğitilen modelin gerçek etiketleri



Şekil 5.7 LR_VisDrone ile eğitilen modelin tahmin etiketleri

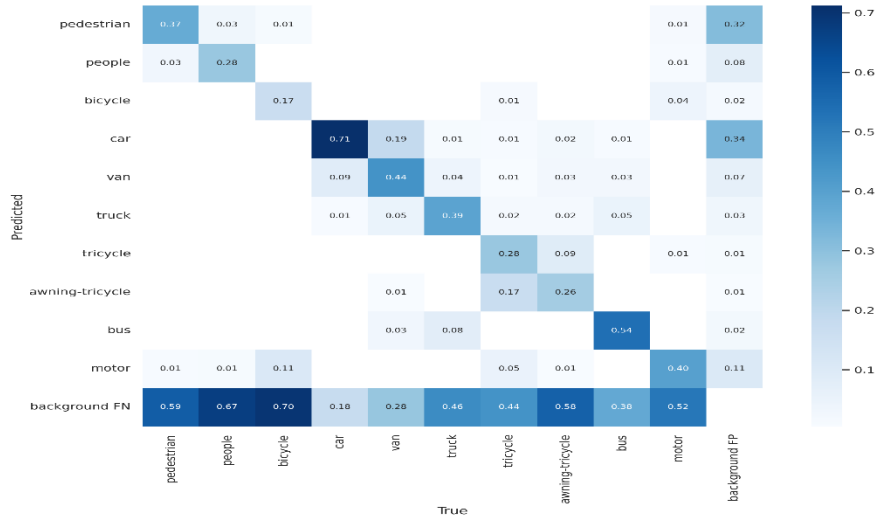
Eđitilen bu model ile iki farklı test gerekleřtirilmiřtir. Bunlardan ilkinde LR veri setinin test grntleri ile model test edilmiřtir. Daha sonra yine aynı model SR veri setinin test grntleri ile test edilmiřtir. Test sonuları řekil 5.8-5.11’de verilmiřtir.



řekil 5.8 LR_VisDrone ile eđitilen modelin LR ile test karıřıklık matrisi

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 51/51 [00:44:00:00,
all	1620	75102	0.52	0.404	0.373	0.208
pedestrian	1620	21006	0.55	0.366	0.358	0.141
people	1620	6376	0.537	0.277	0.261	0.0887
bicycle	1620	1302	0.329	0.165	0.14	0.0561
car	1620	28074	0.732	0.786	0.772	0.484
van	1620	5771	0.504	0.401	0.385	0.255
truck	1620	2659	0.535	0.416	0.393	0.251
tricycle	1620	530	0.33	0.377	0.249	0.127
awning-tricycle	1620	599	0.427	0.252	0.197	0.11
bus	1620	2940	0.76	0.549	0.599	0.419
motor	1620	5845	0.5	0.453	0.381	0.152

řekil 5.9 LR_VisDrone ile eđitilen modelin LR ile test mAP sonuları



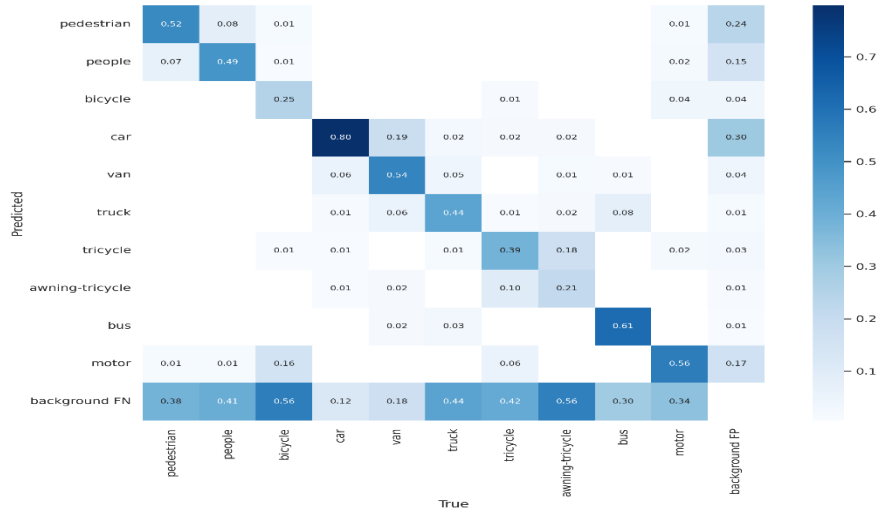
Şekil 5.10 LR_VisDrone ile eğitilen modelin SR ile test karışıklık matrisi

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100%
all	1403	63933	0.508	0.372	0.349	0.194
pedestrian	1403	16905	0.513	0.355	0.335	0.131
people	1403	5151	0.499	0.263	0.239	0.0786
bicycle	1403	1062	0.304	0.154	0.123	0.0473
car	1403	24553	0.707	0.771	0.746	0.469
van	1403	5056	0.516	0.359	0.358	0.238
truck	1403	2314	0.536	0.375	0.365	0.234
tricycle	1403	465	0.317	0.295	0.199	0.0981
awning-tricycle	1403	518	0.433	0.228	0.202	0.118
bus	1403	2557	0.731	0.522	0.558	0.386
motor	1403	5352	0.519	0.399	0.364	0.142

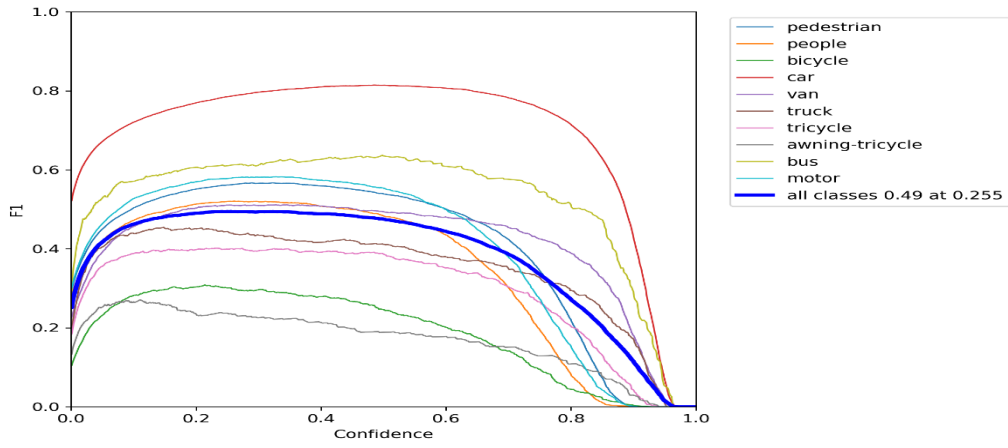
Şekil 5.11 LR_VisDrone ile eğitilen modelin SR ile test mAP sonuçları

5.2 YOLO-V7 SR_VisDrone ile Eğitim ve Test Sonuçları

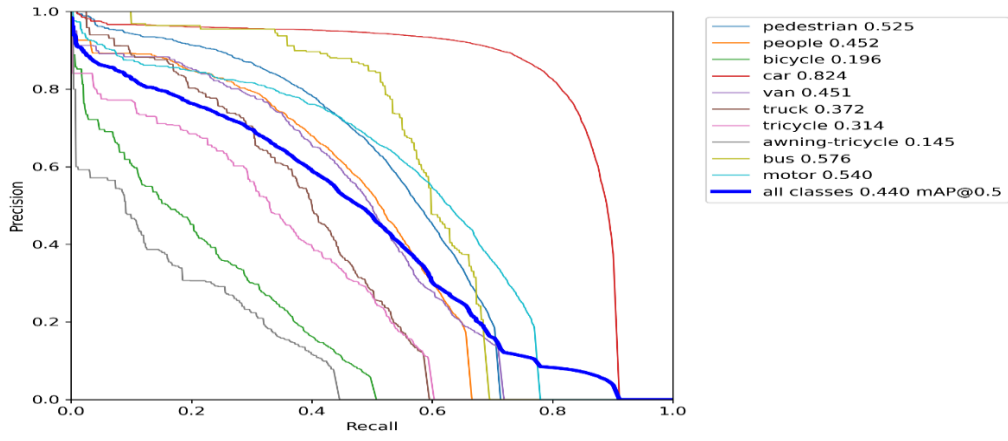
Bir sonraki adımda yeni bir model eğitilecektir. Bu model yine önceden COCO veri setiyle eğitilmiş YOLOv7'nin ağırlıklarını başlangıç olarak alacaktır. Ardından Real-ESRGAN ile elde edilen SR_VisDrone veri seti ile eğitilmiştir. Bir önceki model ile aynı hiperparametreler, aynı iskelet (415 katman ve 37245102 parametre sayısı) ve aynı ortam kullanılarak eğitilmiştir. Yaklaşık 4 kat daha uzun süren bu eğitim 300 adım devam ettirilmiştir. Elde edilen eğitim sonuçları Şekil 5.12- 5.18'da gösterilmiştir.



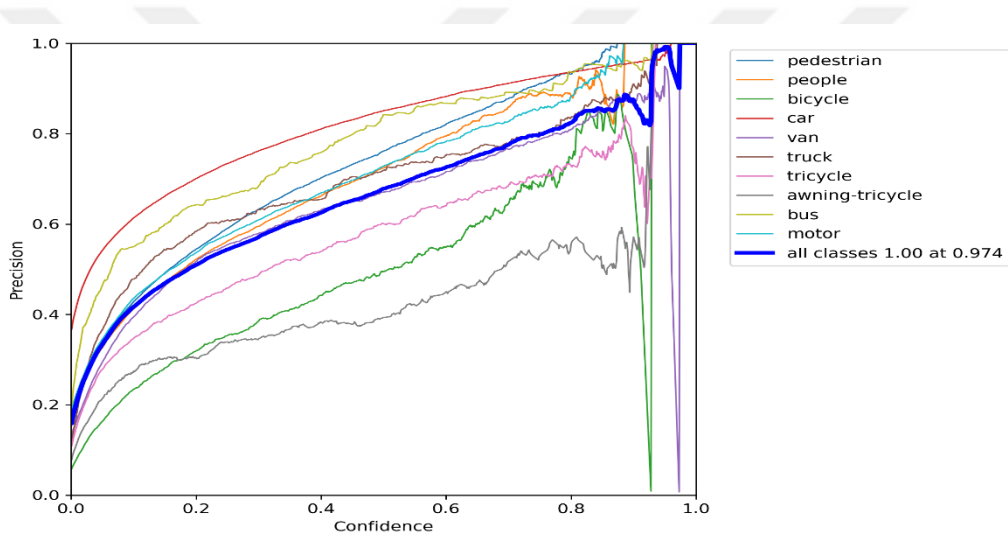
Şekil 5.12 SR_VisDrone ile eğitilen modelin karışıklık matrisi



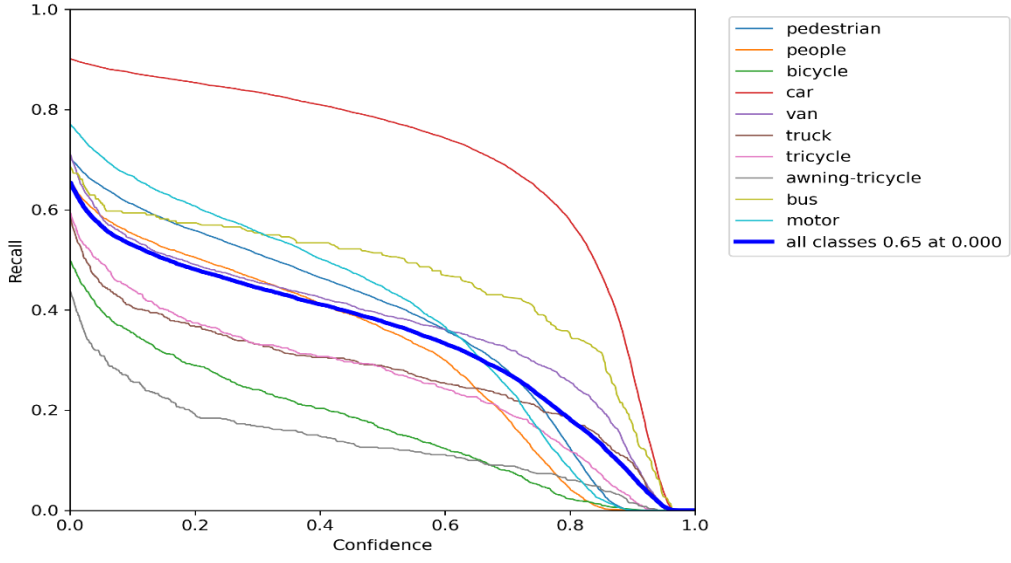
Şekil 5.13 SR_VisDrone ile eğitilen modelin F1 skoru



Şekil 5.14 SR_VisDrone ile eğitilen modelin kesinlik-duyarlılık eğrisi



Şekil 5.15 SR_VisDrone ile eğitilen modelin kesinlik eğrisi



Şekil 5.16 SR_VisDrone ile eğitilen modelin duyarlılık eğrisi

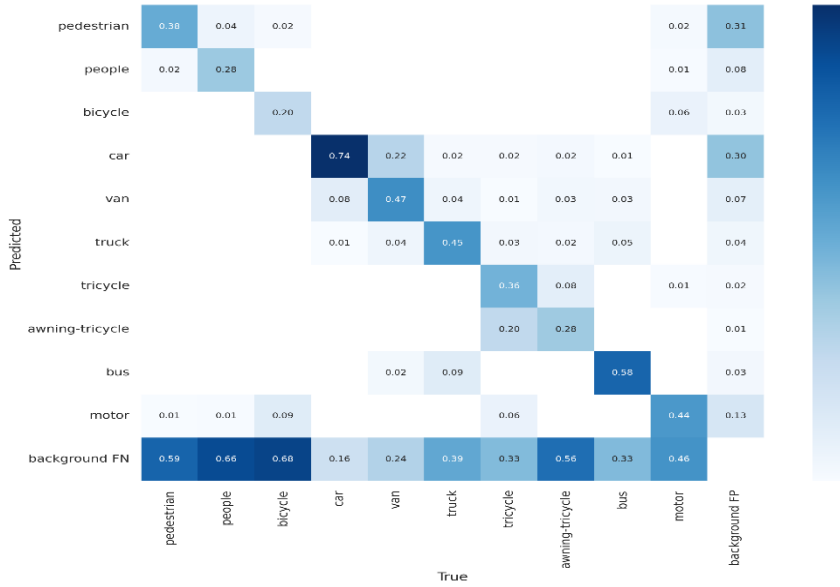


Şekil 5.17 SR_VisDrone ile eğitilen modelin tahmin etiketleri



Şekil 5.18 SR_VisDrone ile eğitilen modelin gerçek etiketleri

Bir önceki modelde olduğu gibi bu modelde de iki farklı şekilde test işlemleri yapılmıştır. Bunlardan ilkinde SR görüntülerle eğitilen yeni model LR görüntülerin test seti ile test edilmiştir. Sonuçlar Şekil 5.19- 5.20’de gösterilmiştir.

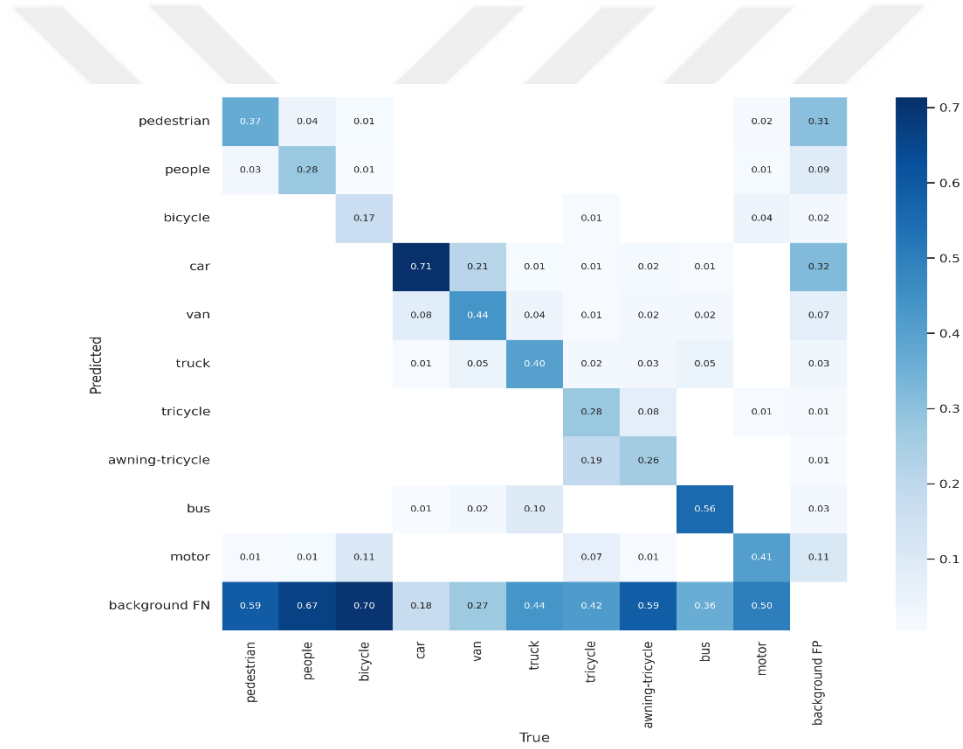


Şekil 5.19 SR_VisDrone ile eğitilen modelin LR ile test karışıklık matrisi

Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 100%	51/51	[00:44<00:00
all	1620	75102	0.528	0.41	0.382	0.214		
pedestrian	1620	21006	0.562	0.359	0.357	0.142		
people	1620	6376	0.54	0.276	0.258	0.0869		
bicycle	1620	1302	0.333	0.177	0.15	0.0623		
car	1620	28074	0.741	0.788	0.776	0.487		
van	1620	5771	0.522	0.399	0.394	0.26		
truck	1620	2659	0.551	0.445	0.42	0.268		
tricycle	1620	530	0.336	0.394	0.256	0.135		
awning-tricycle	1620	599	0.434	0.236	0.21	0.118		
bus	1620	2940	0.751	0.57	0.615	0.427		
motor	1620	5845	0.506	0.453	0.387	0.153		

Şekil 5.20 SR_VisDrone ile eğitilen modelin LR ile test mAP sonuçları

Daha sonra yine aynı model SR veri setinin test görüntüleri ile test edilmiştir. Sonuçlar Şekil 5.21- 5.22'de gösterilmiştir.



Şekil 5.21 SR_VisDrone ile eğitilen modelin SR ile test karışıklık matrisi

Class	Images	Labels	P	R	mAP@.5	mAP@.5: .95: 100% 44/44 [02:13<00:00,
all	1403	63933	0.479	0.393	0.355	0.197
pedestrian	1403	16905	0.481	0.367	0.34	0.133
people	1403	5151	0.441	0.283	0.241	0.0784
bicycle	1403	1062	0.271	0.174	0.124	0.05
car	1403	24553	0.694	0.778	0.752	0.473
van	1403	5056	0.489	0.387	0.367	0.244
truck	1403	2314	0.511	0.404	0.38	0.243
tricycle	1403	465	0.271	0.323	0.191	0.0919
awning-tricycle	1403	518	0.432	0.232	0.207	0.121
bus	1403	2557	0.719	0.545	0.579	0.396
motor	1403	5352	0.485	0.436	0.371	0.143

Şekil 5.22 SR_VisDrone ile eğitilen modelin SR ile test mAP sonuçları

5.3 YOLO-V8 LR_VisDrone ile Eğitim ve Test Sonuçları

Bu adımda yine Colab ortamında 268 katmana ve 68.200.608 parametreye sahip YOLOv8 modelinin “v8x” backbone yapısı kullanılarak LR_VisDrone ile eğitilmiştir. Bu modelin LR görüntüler ile elde edilen test sonuçları Şekil 5.23-5.24’de gösterilmiştir.

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100% 102/102 [01:51<00:00, 1.10s/it]
all	1620	75102	0.507	0.389	0.38	0.223
pedestrian	1620	21006	0.561	0.302	0.326	0.138
people	1620	6376	0.529	0.127	0.174	0.0639
bicycle	1620	1302	0.294	0.176	0.133	0.0567
car	1620	28074	0.724	0.759	0.767	0.491
van	1620	5771	0.481	0.466	0.444	0.296
truck	1620	2659	0.528	0.496	0.47	0.31
tricycle	1620	530	0.276	0.319	0.226	0.126
awning-tricycle	1620	599	0.464	0.267	0.255	0.146
bus	1620	2940	0.732	0.586	0.639	0.457
motor	1620	5845	0.479	0.393	0.362	0.149

Speed: 0.5ms preprocess, 31.6ms inference, 0.0ms loss, 3.6ms postprocess per image
Results saved to runs/detect/val4
Learn more at <https://docs.ultralytics.com/modes/val>

Şekil 5.23 LR_VisDrone ile eğitilen modelin LR ile test mAP sonuçları



Şekil 5.24 LR_ VisDrone ile eğitilen modelin LR ile test karışıklık matrisi

Aynı modelin SR görüntüler ile elde edilen test sonuçları Şekil 5.25- 5.26'da gösterilmiştir.

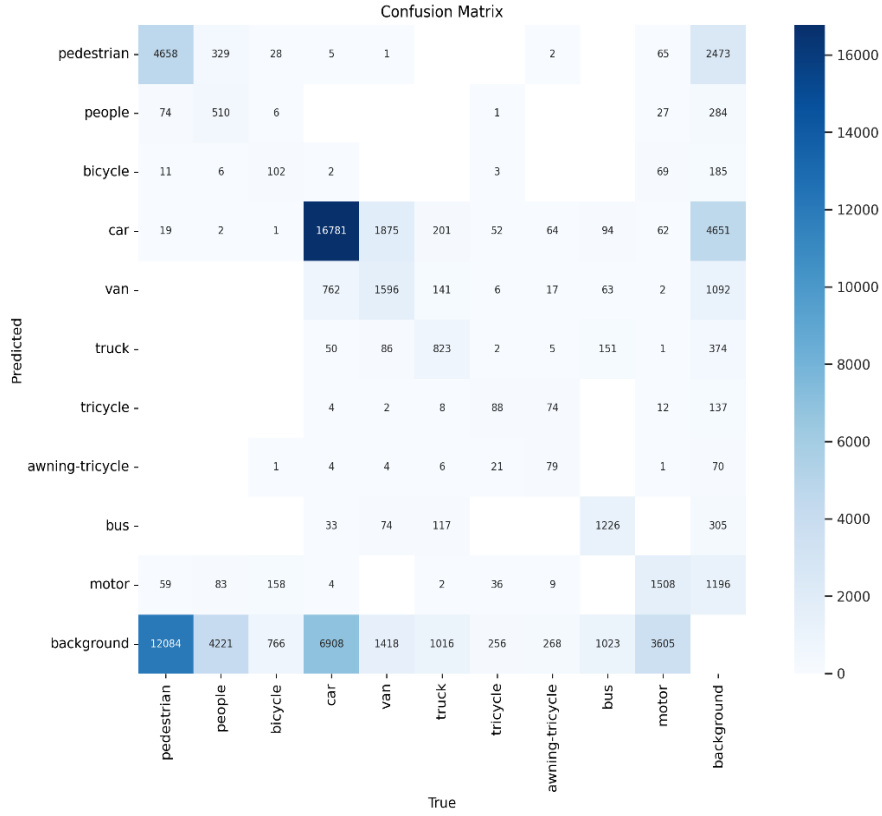
```

Class      Images  Instances  Box(P)    R      mAP50  mAP50-95): 100% 88/88 [06:35<00:00, 4.49s/it]
all        1403    63933     0.464     0.34   0.327   0.188
pedestrian 1403    16905     0.49      0.283  0.287   0.116
people     1403    5151      0.494     0.128  0.159   0.0564
bicycle    1403    1062      0.211     0.136  0.0883  0.0341
car        1403    24553     0.649     0.709  0.708   0.444
van        1403    5056      0.453     0.414  0.39    0.253
truck      1403    2314      0.518     0.411  0.401   0.263
tricycle   1403    465       0.256     0.241  0.153   0.0844
awning-tricycle 1403    518       0.464     0.231  0.236   0.134
bus        1403    2557      0.681     0.517  0.557   0.385
motor      1403    5352      0.419     0.329  0.285   0.112

Speed: 0.6ms preprocess, 34.3ms inference, 0.0ms loss, 6.0ms postprocess per image
Results saved to runs/detect/val
Learn more at https://docs.ultralytics.com/modes/val

```

Şekil 5.25 LR_ VisDrone ile eğitilen modelin SR ile test mAP sonuçları



Şekil 5.26 LR_VisDrone ile eğitilen modelin SR ile test karışıklık matrisi

5.4 YOLO-V8 SR_VisDrone ile Eğitim ve Test Sonuçları

Bu adımda Colab ortamında 268 katmana ve 68.200.608 parametreye sahip YOLOv8 modelinin “v8x” backbone kullanılarak SR_VisDrone ile eğitilmiştir. Bu modelin LR görüntüler ile elde edilen test sonuçları Şekil 5.27- 5.28’da verilmiştir.

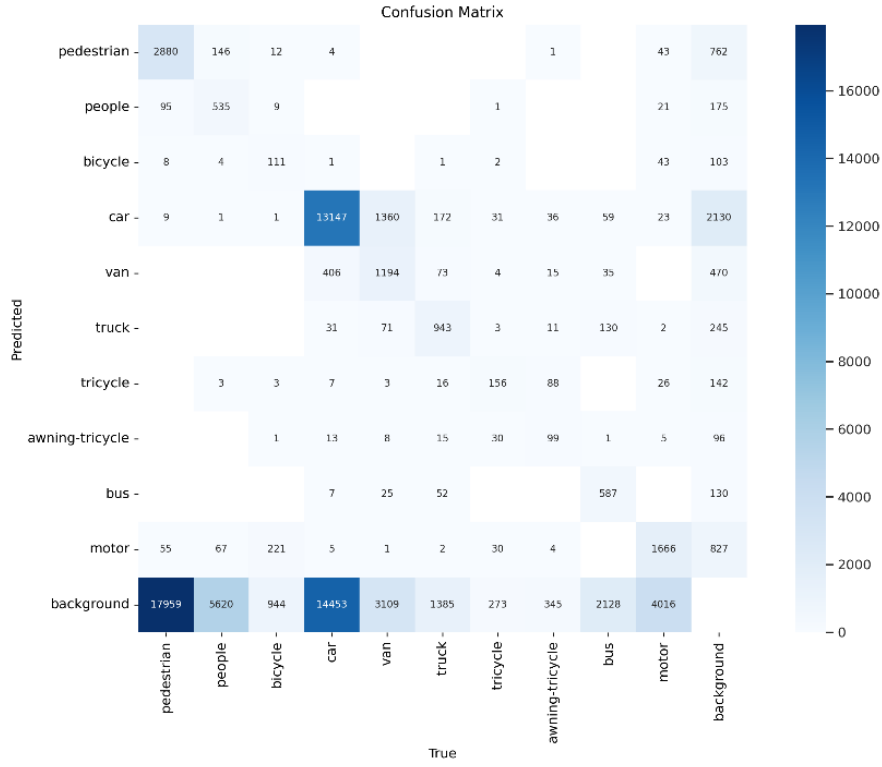
```

Class      Images  Instances  Box(P)  R      mAP50  mAP50-95): 100% 102/102 [02:01<00:00, 1.19s/it]
all        1620    75102     0.485   0.287  0.284   0.163
pedestrian 1620    21006     0.572   0.163  0.21    0.0881
people     1620    6376     0.49    0.119  0.15    0.055
bicycle    1620    1302     0.291   0.149  0.123   0.0505
car        1620    28074    0.682   0.501  0.549   0.349
van        1620    5771     0.51    0.274  0.304   0.207
truck      1620    2659     0.561   0.402  0.403   0.268
tricycle   1620    530      0.274   0.399  0.25    0.133
awning-tricycle 1620    599      0.327   0.258  0.197   0.113
bus        1620    2940     0.703   0.244  0.337   0.235
motor      1620    5845     0.438   0.363  0.318   0.127

Speed: 0.3ms preprocess, 31.3ms inference, 0.0ms loss, 4.6ms postprocess per image
Results saved to runs/detect/val
Learn more at https://docs.ultralytics.com/modes/val

```

Şekil 5.27 SR_VisDrone ile eğitilen modelin LR ile test mAP sonuçları



Şekil 5.28 SR_VisDrone ile eğitilen modelin LR ile test karışıklık matrisi

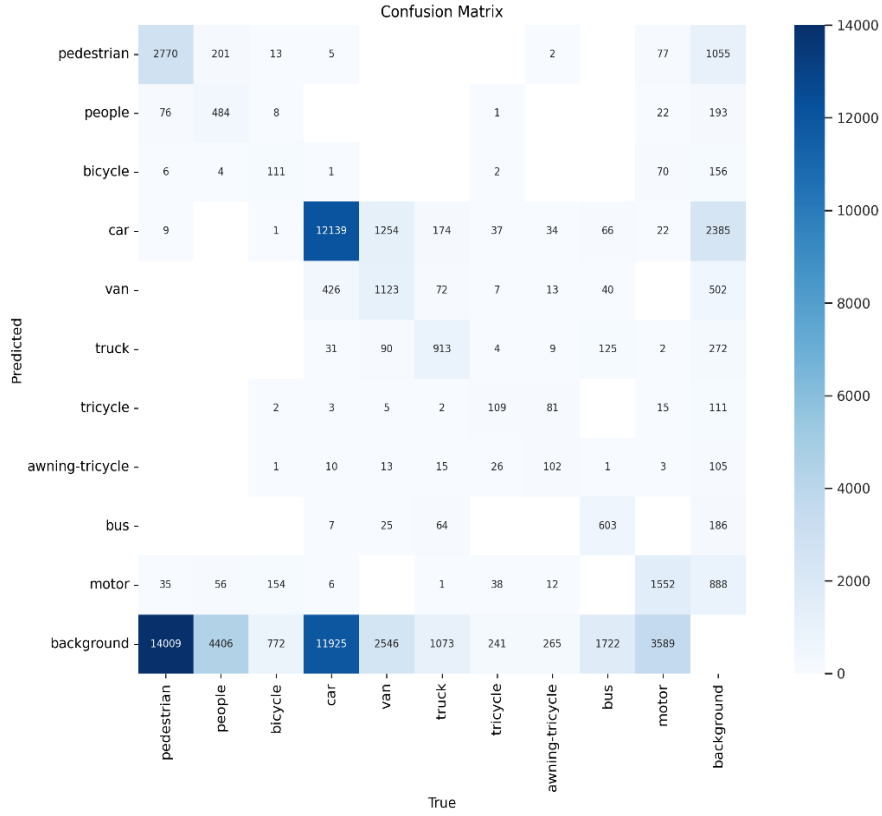
Aynı modelin SR görüntüler ile elde edilen test sonuçları Şekil 5.29-5.30'da verilmiştir.

```

Class      Images  Instances  Box(P)   R        mAP50  mAP50-95): 100% 88/88 [06:33<00:00, 4.47s/it]
all        1403    63933     0.479    0.297    0.295    0.171
pedestrian 1403    16905     0.522    0.184    0.215    0.0898
people     1403    5151      0.514    0.124    0.16     0.059
bicycle    1403    1062      0.234    0.165    0.117    0.0492
car        1403    24553     0.676    0.521    0.558    0.357
van        1403    5056      0.495    0.286    0.304    0.209
truck      1403    2314      0.566    0.443    0.443    0.297
tricycle   1403    465       0.3      0.318    0.22     0.119
awning-tricycle 1403    518       0.363    0.294    0.24     0.142
bus        1403    2557      0.678    0.278    0.362    0.249
motor      1403    5352      0.446    0.356    0.326    0.134
Speed: 0.6ms preprocess, 34.3ms inference, 0.0ms loss, 6.4ms postprocess per image
Results saved to runs/detect/val
Learn more at https://docs.ultralytics.com/modes/val

```

Şekil 5.29 SR_VisDrone ile eğitilen modelin SR ile test mAP sonuçları



Şekil 5.30 SR_VisDrone ile eğitilen modelin SR ile test karışıklık matrisi

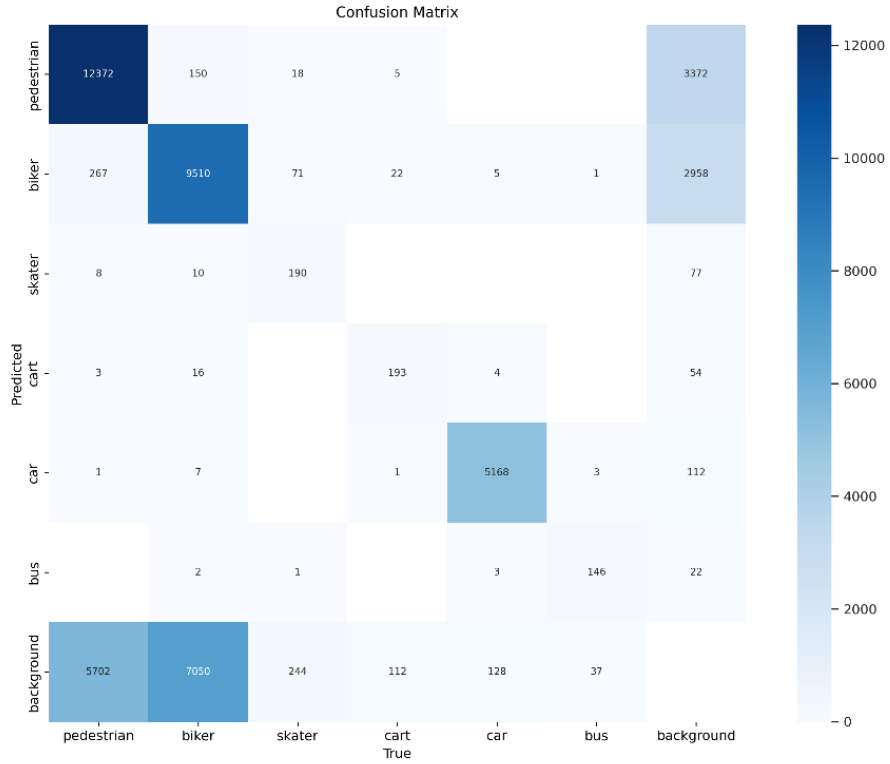
5.5 YOLO-V8 LR_SDD ile Eğitim ve Test Sonuçları

Bu adımda yine Colab ortamında 268 katmana ve 68.200.608 parametreye sahip YOLOv8 modelinin “v8x” backbone kullanılarak LR_SDD ile eğitim ve test verileri elde edilmiştir. LR görüntülerle eğitilen modelden elde edilen test sonuçlar Şekil 5.31- 5.32’de gösterilmiştir.

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	92/92 [01:33<00:00, 1.02s/it]
all	1463	41450	0.833	0.644	0.736	0.491	
pedestrian	1463	18353	0.802	0.636	0.729	0.381	
biker	1463	16745	0.781	0.524	0.649	0.305	
skater	1463	524	0.762	0.384	0.521	0.252	
cart	1463	333	0.767	0.565	0.653	0.37	
car	1463	5308	0.981	0.972	0.993	0.904	
bus	1463	187	0.902	0.783	0.872	0.731	

Speed: 0.4ms preprocess, 38.4ms inference, 0.0ms loss, 3.6ms postprocess per image
 Results saved to runs/detect/val3
 Learn more at <https://docs.ultralytics.com/modes/val>

Şekil 5.31 LR_SDD ile eğitilen modelin LR ile test mAP sonuçları



Şekil 5.32 LR_ SDD ile eğitilen modelin LR ile test karışıklık matrisi

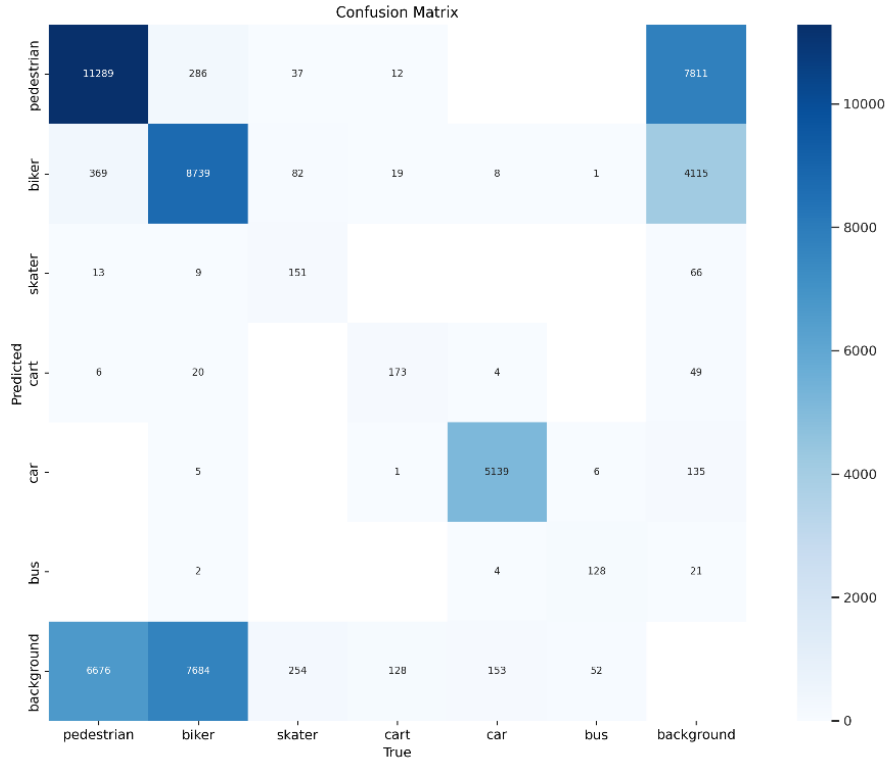
Aynı modelin SR görüntüler ile elde edilen test sonuçları Şekil 5.33- 5.34'te gösterilmiştir.

```

Class      Images  Instances  Box(P)  R    mAP50  mAP50-95): 100% 92/92 [13:45<00:00, 8.97s/it]
all        1463    41450     0.757   0.596 0.655   0.39
pedestrian 1463    18353     0.576   0.581 0.583   0.238
biker      1463    16745     0.663   0.488 0.562   0.222
skater     1463    524       0.717   0.311 0.403   0.162
cart       1463    333       0.719   0.508 0.581   0.278
car        1463    5308     0.975   0.967 0.99    0.818
bus        1463    187       0.894   0.721 0.809   0.62
Speed: 0.3ms preprocess, 40.6ms inference, 0.0ms loss, 3.8ms postprocess per image
Results saved to runs/detect/val
Learn more at https://docs.ultralytics.com/modes/val

```

Şekil 5.33 LR_ SDD ile eğitilen modelin SR ile test mAP sonuçları



Şekil 5.34 LR_SDD ile eğitilen modelin SR ile test karışıklık matrisi

5.6 YOLO-V8 SR_SDD ile Eğitim ve Test Sonuçları

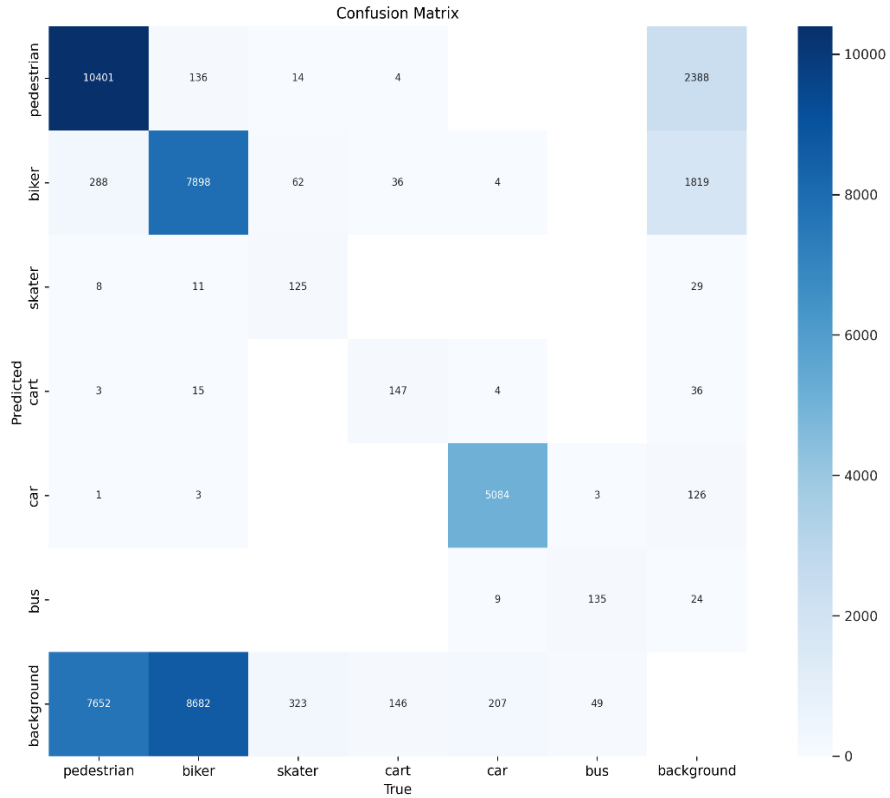
Bu adımda yine Colab ortamında 268 katmana ve 68.200.608 parametreye sahip YOLOv8 modelinin “v8x” backbone kullanılarak SR_SDD ile eğitim ve test verileri elde edilmiştir. SR görüntülerle eğitilen modelden elde edilen Test sonuçları Şekil 5.35- 5.36’da gösterilmiştir.

```

Class      Images  Instances  Box(P  R      mAP50  mAP50-95): 100% 92/92 [01:35<00:00, 1.04s/it]
  all      1463    41450     0.782  0.579  0.657  0.412
pedestrian 1463    18353     0.757  0.565  0.653  0.31
biker      1463    16745     0.725  0.47   0.58   0.243
skater     1463    524       0.743  0.261  0.367  0.168
cart       1463    333       0.69   0.476  0.538  0.286
car        1463    5308     0.971  0.959  0.986  0.839
bus        1463    187       0.806  0.743  0.818  0.629
Speed: 0.2ms preprocess, 13.1ms inference, 0.0ms loss, 3.7ms postprocess per image
Results saved to runs/detect/val
Learn more at https://docs.ultralytics.com/modes/val

```

Şekil 5.35 SR_SDD ile eğitilen modelin LR ile test mAP sonuçları



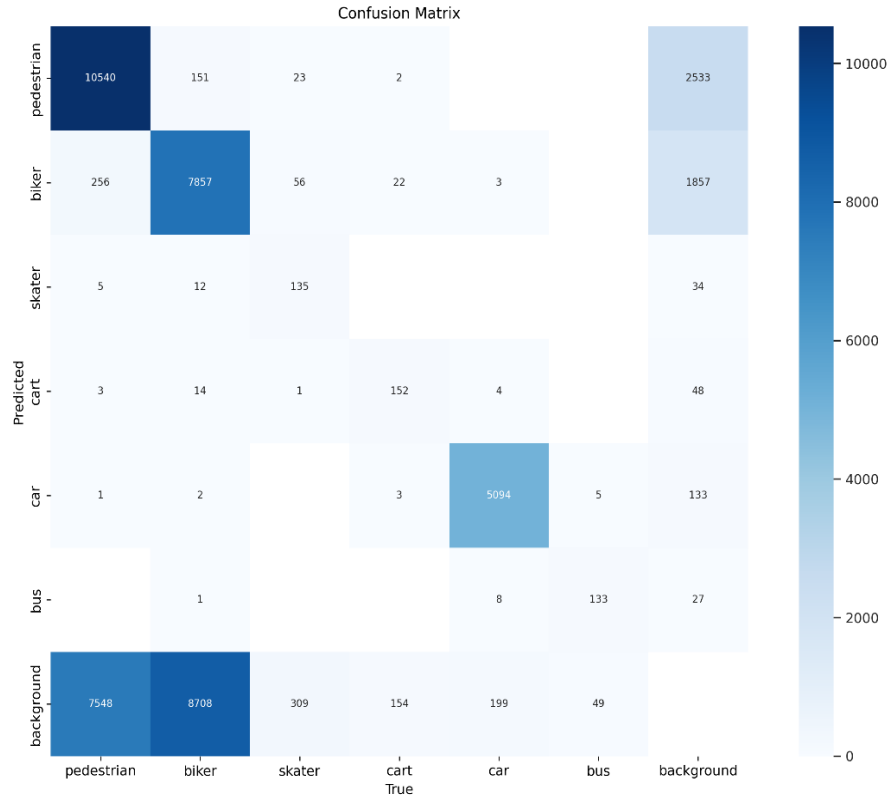
Şekil 5.36 SR_ SDD ile eğitilen modelin LR ile test karışıklık matrisi

Aynı modelin SR görüntüler ile elde edilen test sonuçları Şekil 5.37-5.38'de gösterilmiştir.

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	1463	41450	0.767	0.591	0.661	0.409
pedestrian	1463	18353	0.734	0.577	0.655	0.303
biker	1463	16745	0.716	0.48	0.585	0.244
skater	1463	524	0.734	0.282	0.391	0.176
cart	1463	333	0.662	0.508	0.553	0.285
car	1463	5308	0.967	0.962	0.986	0.835
bus	1463	187	0.791	0.738	0.793	0.611

Speed: 0.4ms preprocess, 14.6ms inference, 0.0ms loss, 3.8ms postprocess per image
 Results saved to runs/detect/val
 Learn more at <https://docs.ultralytics.com/modes/val>

Şekil 5.37 SR_ SDD ile eğitilen modelin SR ile test mAP sonuçları



Şekil 5.38 SR_ SDD ile eğitilen modelin SR ile test karışıklık matrisi

Bu tez çalışmasında insansız hava araçlarından elde edilen görüntülerde nesne tespitinin iyileştirilmesi üzerine çalışmalar yapılmıştır. Bu görüntülerdeki nesnelere genellikle az piksel kapladığı için küçük nesne tespiti problemine yol açmaktadır. Bu çalışma kapsamında görüntülerin çözünürlüklerini artırarak nesne tespitinin iyileştirilmesi hedeflenmiştir. Çözünürlük artırma metodu olarak GAN ağlarından faydalanılmıştır. Bu ağlardan tümüyle sentetik verilerle eğitilen Real-ESRGAN modeli tercih edilmiştir. VisDrone ve SDD veri setleri üzerinde uygulanan bu model ile birlikte süper çözünürlüklü veri setleri elde edilmiştir. Ardından bu veri setleri sırasıyla YOLOv7 ve YOLOv8 nesne tespiti modelleri ile eğitip test edilmiştir. Bu testler sonucunda Real-ESRGAN modelinin nesne tespiti üzerindeki etkisi ortaya konulmuştur.

Deneysel Çalışmalar bölümünde modelin performans çıktıları verilmiştir. Bu veriler 6 farklı modele ait sonuçları göstermektedir. Bu modeller sırasıyla; YOLOv7-LR_VisDrone, YOLOv7-SR_VisDrone, YOLOv8-LR_VisDrone, YOLOv8-SR_VisDrone, YOLOv8-LR_SDD ve YOLOv8-SR_SDD'dir. Tüm bu modeller LR görüntüler ve SR görüntülerle ayrı ayrı test edilmiştir. YOLOv7-LR_VisDrone modeli LR görüntüler ile test edildiğinde mAP@0,5 doğruluk değeri **%37,3**, SR görüntüler ile test edildiğinde ise **%34,9** doğruluk oranı elde edildiği tespit edilmiştir. YOLOv7-SR_VisDrone modeli ise LR görüntülerle test edildiğinde **%38,2**, SR görüntüler ile test edildiğinde **%35,5** doğruluk oranları elde edilmiştir. Bu sonuçlara bakıldığında VisDrone veri seti için modeli SR ile eğitip test etmek yaklaşık %1 oranında doğruluk değerini artırdığı görülmektedir. Aynı işlemler LR_SDD ve SR_SDD veri setleri ile eğitilmiş modeller için de uygulanmıştır. Fakat SDD veri seti için herhangi bir doğruluk artışı gözlemlenmemiştir. YOLOv8 modelinde de paylaşılan sonuçlardan anlaşılacağı üzere doğruluk artışı gözlemlenmemiştir.

Çalışma kapsamında SR görüntülerin büyük çoğunluğunun LR görüntülere kıyasla daha düşük bir doğruluk oranıyla sonuç vermesi beklenmedik bir durum

olarak not edilmiştir. Bu durumun başlıca sebebi olarak YOLO algoritmasının model girişinde görseller üzerinde bir sıkıştırma işlemi yapması olarak öngörülmüştür. Bu sıkıştırma işlemi eğitilen ve test edilen görsellerin tümünün 640 x 640 boyutlarına sıkıştırılmasıyla gerçekleştirilir. Dolayısıyla Real-ESRGAN ile SR olarak elde edilen görüntü YOLO'nun bu sıkıştırma işlemi sebebiyle özellik kaybına uğradığı ve bu kayıp sebebiyle de LR ile elde edilen nesne tespitinden daha kötü bir performans gösterdiği düşünülmektedir. Bu problemin önüne geçebilmek için SAHI adı verilen algoritma kullanılabilir. Bu algoritma geniş ölçekli görsellerin çözünürlüklerini değiştirmeden, bütün olarak almak yerine küçük pencereler halinde parça parça modele girerek nesne tespiti yapılabilmesini sağlayan bir algoritmadır. SAHI algoritması haricinde de görseli manuel olarak bölümlere ayırarak ayrı ayrı nesne tespiti yapmak bu problemin çözümü olarak önerilebilmektedir.

Diğer bir sebep de elde edilen SR görüntülerinin çözünürlük oranının fazla olması sebebiyle nesnelerin özelliklerinin bozulması olabilir. Bu çalışmada Real-ESRGAN modelinin x4-scale algoritması kullanılmıştır. Bu problemin önüne geçebilmek adına nesnelerin daha az bozunuma uğrayacağı tahmin edilen Real-ESRGAN x2-scale algoritmasının kullanılması önerilebilir.

Farklı bir bakış açısı olarak SR görüntülerde genel olarak nesne tespit performansının düşüş eğiliminde olması kullanılan veri setlerine bağlı olabilir. Yani bu veri setleri süper çözünürlük problemlerinde kullanmak için uygun olmayabilir. Literatürde sunulan birçok çalışmada ESRGAN-YOLO entegrasyonu sonucunda kullanılan veri setlerinde nesne tespitinde doğruluk oranının artış eğiliminde olduğu literatür özeti kısmında gösterilmiştir. Dolayısıyla yer alan çalışmaların bu görüşü destekler nitelikte olduğu söylenebilir. Sonuç olarak kullanılan VisDrone ve SDD veri setlerinin bu çalışma için uygun olmadıkları çıkarımı yapılabilir.

-
- [1] “İnsansız Hava Araçlarının Kullanım Alanları ve Sensör Tipleri.pdf.”
- [2] Y. Jia *et al.*, “Caffe: Convolutional Architecture for Fast Feature Embedding.” arXiv, Jun. 20, 2014. Accessed: Jan. 24, 2024. [Online]. Available: <http://arxiv.org/abs/1408.5093>
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields.” arXiv, Apr. 13, 2017. Accessed: Jan. 24, 2024. [Online]. Available: <http://arxiv.org/abs/1611.08050>
- [5] Z. Yang and R. Nevatia, “A Multi-Scale Cascade Fully Convolutional Network Face Detector.” arXiv, Sep. 12, 2016. Accessed: Jan. 24, 2024. [Online]. Available: <http://arxiv.org/abs/1609.03536>
- [6] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, “DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile: IEEE, Dec. 2015, pp. 2722–2730. doi: 10.1109/ICCV.2015.312.
- [7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-View 3D Object Detection Network for Autonomous Driving.” arXiv, Jun. 21, 2017. Accessed: Jan. 24, 2024. [Online]. Available: <http://arxiv.org/abs/1611.07759>
- [8] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
- [9] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, San Diego, CA, USA: IEEE, 2005, pp. 886–893. doi: 10.1109/CVPR.2005.177.

- [10] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.
- [11] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [12] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” arXiv, Mar. 02, 2015. Accessed: Jan. 24, 2024. [Online]. Available: <http://arxiv.org/abs/1502.03167>
- [13] C. Szegedy *et al.*, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.
- [14] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv, Apr. 10, 2015. Accessed: Jan. 24, 2024. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
- [16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
- [17] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” vol. 9905, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.
- [18] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object Detection With Deep Learning: A Review,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
- [19] I. Gallo, A. U. Rehman, R. H. Dehkordi, N. Landro, R. La Grassa, and M. Boschetti, “Deep Object Detection of Crop Weeds: Performance of YOLOv7 on a Real Case Dataset from UAV Images,” *Remote Sens.*, vol. 15, no. 2, p. 539, Jan. 2023, doi: 10.3390/rs15020539.

- [20] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios," *Sensors*, vol. 23, no. 16, p. 7190, Aug. 2023, doi: 10.3390/s23167190.
- [21] Y. Zeng, T. Zhang, W. He, and Z. Zhang, "YOLOv7-UAV: An Unmanned Aerial Vehicle Image Object Detection Algorithm Based on Improved YOLOv7," *Electronics*, vol. 12, no. 14, p. 3141, Jul. 2023, doi: 10.3390/electronics12143141.
- [22] H. Zhao, H. Zhang, and Y. Zhao, "YOLOv7-sea: Object Detection of Maritime UAV Images based on Improved YOLOv7," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA: IEEE, Jan. 2023, pp. 233–238. doi: 10.1109/WACVW58289.2023.00029.
- [23] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1646–1658, Dec. 1997, doi: 10.1109/83.650118.
- [24] I. J. Goodfellow *et al.*, "Generative Adversarial Networks." arXiv, Jun. 10, 2014. Accessed: Oct. 23, 2023. [Online]. Available: <http://arxiv.org/abs/1406.2661>
- [25] R. Wang, X. Xiao, B. Guo, Q. Qin, and R. Chen, "An Effective Image Denoising Method for UAV Images via Improved Generative Adversarial Networks," *Sensors*, vol. 18, no. 7, p. 1985, Jun. 2018, doi: 10.3390/s18071985.
- [26] M. Pashaei, M. J. Starek, H. Kamangir, and J. Berryhill, "Deep Learning-Based Single Image Super-Resolution: An Investigation for Dense Scene Reconstruction with UAS Photogrammetry," *Remote Sens.*, vol. 12, no. 11, p. 1757, May 2020, doi: 10.3390/rs12111757.
- [27] E. Agustsson and R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA: IEEE, Jul. 2017, pp. 1122–1131. doi: 10.1109/CVPRW.2017.150.

- [28] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced Deep Residual Networks for Single Image Super-Resolution.” arXiv, Jul. 10, 2017. Accessed: Oct. 23, 2023. [Online]. Available: <http://arxiv.org/abs/1707.02921>
- [29] C. Ledig *et al.*, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.” arXiv, May 25, 2017. Accessed: Oct. 23, 2023. [Online]. Available: <http://arxiv.org/abs/1609.04802>
- [30] X. Wang *et al.*, “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks.” arXiv, Sep. 17, 2018. Accessed: Oct. 24, 2023. [Online]. Available: <http://arxiv.org/abs/1809.00219>
- [31] Y. Chen, J. Li, Y. Niu, and J. He, “Small Object Detection Networks Based on Classification-Oriented Super-Resolution GAN for UAV Aerial Imagery,” in *2019 Chinese Control And Decision Conference (CCDC)*, Nanchang, China: IEEE, Jun. 2019, pp. 4610–4615. doi: 10.1109/CCDC.2019.8832735.
- [32] C. Xing, X. Liang, and Z. Bao, “A Small Object Detection Solution by Using Super-Resolution Recovery,” in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, Dalian, China: IEEE, Oct. 2019, pp. 313–316. doi: 10.1109/ICCSNT47585.2019.8962422.
- [33] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li, and F. Huang, “Real-World Super-Resolution via Kernel Estimation and Noise Injection,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 1914–1923. doi: 10.1109/CVPRW50498.2020.00241.
- [34] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, “Small-Object Detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network,” *Remote Sens.*, vol. 12, no. 9, p. 1432, May 2020, doi: 10.3390/rs12091432.
- [35] H. Zou, Y. Gao, X. Guo, and M. Zheng, “Small Object Detection Based on Super-Resolution Enhanced Detection Network,” in *2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)*, Kunming, China: IEEE, Sep. 2021, pp. 99–106. doi: 10.1109/CISAI54367.2021.00027.

- [36] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data." arXiv, Aug. 17, 2021. Accessed: Oct. 23, 2023. [Online]. Available: <http://arxiv.org/abs/2107.10833>
- [37] F. Akhyar, E. N. Furqon, and C.-Y. Lin, "Enhancing Precision with an Ensemble Generative Adversarial Network for Steel Surface Defect Detectors (EnsGAN-SDD)," *Sensors*, vol. 22, no. 11, p. 4257, Jun. 2022, doi: 10.3390/s22114257.
- [38] S. Chen, J. Lan, H. Liu, C. Chen, and X. Wang, "Helmet Wearing Detection of Motorcycle Drivers Using Deep Learning Network with Residual Transformer-Spatial Attention," *Drones*, vol. 6, no. 12, p. 415, Dec. 2022, doi: 10.3390/drones6120415.
- [39] H. Salaudeen and E. Çelebi, "Pothole Detection Using Image Enhancement GAN and Object Detection Network," *Electronics*, vol. 11, no. 12, p. 1882, Jun. 2022, doi: 10.3390/electronics11121882.
- [40] A.-C. Tsai, C.-H. Tsou, and J.-F. Wang, "EESRGAN: Efficient & Effective Super-Resolution Generative Adversarial Network," *IETE Tech. Rev.*, pp. 1–12, Jul. 2023, doi: 10.1080/02564602.2023.2236071.
- [41] H. Wu, J. Chen, T. Wang, X. Lai, and J. Cao, "Ship License Plate Super-Resolution in the Wild," *IEEE Signal Process. Lett.*, vol. 30, pp. 394–398, 2023, doi: 10.1109/LSP.2023.3262418.
- [42] H. Lv, A. Zhao, Y. Wang, H. Li, Y. Li, and G. Zou, "Research on Missing Bolt Sub-Detection Based on Real-ESRGAN Super-Resolution Reconstruction," in *2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, Nanjing, China: IEEE, Jun. 2023, pp. 520–523. doi: 10.1109/AINIT59027.2023.10212817.
- [43] M. Jabberi, A. Wali, and A. M. Alimi, "Generative Data Augmentation applied to Face Recognition," in *2023 International Conference on Information Networking (ICOIN)*, Bangkok, Thailand: IEEE, Jan. 2023, pp. 242–247. doi: 10.1109/ICOIN56518.2023.10049052.
- [44] Y. Vedavyas, S. S. Harsha, M. S. Subhash, and S. Vasavi, "Quality Enhancement for Drone Based Video using FPGA," in *2022 International*

Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India: IEEE, Mar. 2022, pp. 29–34. doi: 10.1109/ICEARS53579.2022.9751731.

[45] Q. Wang, H. Zhou, G. Li, and J. Guo, “Single Image Super-Resolution Method Based on an Improved Adversarial Generation Network,” *Appl. Sci.*, vol. 12, no. 12, p. 6067, Jun. 2022, doi: 10.3390/app12126067.

[46] Z. Zhu, Y. Lei, Y. Qin, C. Zhu, and Y. Zhu, “IRE: Improved Image Super-Resolution Based on Real-ESRGAN,” *IEEE Access*, vol. 11, pp. 45334–45348, 2023, doi: 10.1109/ACCESS.2023.3256086.

[47] K. Yamamoto, T. Togami, and N. Yamaguchi, “Super-Resolution of Plant Disease Images for the Acceleration of Image-based Phenotyping and Vigor Diagnosis in Agriculture,” *Sensors*, vol. 17, no. 11, p. 2557, Nov. 2017, doi: 10.3390/s17112557.

[48] M. H. Maqsood, R. Mumtaz, I. U. Haq, U. Shafi, S. M. H. Zaidi, and M. Hafeez, “Super Resolution Generative Adversarial Network (SRGANs) for Wheat Stripe Rust Classification,” *Sensors*, vol. 21, no. 23, p. 7903, Nov. 2021, doi: 10.3390/s21237903.

[49] Z. Ye *et al.*, “Extraction of Olive Crown Based on UAV Visible Images and the U2-Net Deep Learning Model,” *Remote Sens.*, vol. 14, no. 6, p. 1523, Mar. 2022, doi: 10.3390/rs14061523.

[50] Z. Zhang, Q. Gao, L. Liu, and Y. He, “A High-Quality Rice Leaf Disease Image Data Augmentation Method Based on a Dual GAN,” *IEEE Access*, vol. 11, pp. 21176–21191, 2023, doi: 10.1109/ACCESS.2023.3251098.

[51] H. Zheng, S. Dembélé, Y. Wu, Y. Liu, H. Chen, and Q. Zhang, “A lightweight algorithm capable of accurately identifying forest fires from UAV remote sensing imagery,” *Front. For. Glob. Change*, vol. 6, p. 1134942, Feb. 2023, doi: 10.3389/ffgc.2023.1134942.

[52] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, “Vision Meets Drones: A Challenge.” arXiv, Apr. 22, 2018. Accessed: Nov. 13, 2023. [Online]. Available: <http://arxiv.org/abs/1804.07437>

[53] Zhu, Pengfei and Wen, Longyin and Du, Dawei and Bian, Xiao and Fan, Heng and Hu, Qinghua and Ling, Haibin, “Detection and tracking meet drones

challenge,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, pp. 7380–7399, 2021.

[54] “A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes in European Conference on Computer Vision (ECCV), 2016.”

[55] S. YiĜiT, “Improving object detection of UAV images with Real-ESRGAN,” *Recent Adv. Sci. Eng.*, pp. 33–39, 2023, doi: 10.14744/rase.2023.0004.

[56] H. Zheng, S. Dembélé, Y. Wu, Y. Liu, H. Chen, and Q. Zhang, “A lightweight algorithm capable of accurately identifying forest fires from UAV remote sensing imagery,” *Front. For. Glob. Change*, vol. 6, p. 1134942, Feb. 2023, doi: 10.3389/ffgc.2023.1134942.

[57] Z. Saeed, M. H. Yousaf, R. Ahmed, S. A. Velastin, and S. Viriri, “On-Board Small-Scale Object Detection for Unmanned Aerial Vehicles (UAVs),” *Drones*, vol. 7, no. 5, p. 310, May 2023, doi: 10.3390/drones7050310.

[58] E. Schönfeld, B. Schiele, and A. Khoreva, “A U-Net Based Discriminator for Generative Adversarial Networks.” arXiv, Mar. 19, 2021. Accessed: Oct. 24, 2023. [Online]. Available: <http://arxiv.org/abs/2002.12655>

[59] D. Reis, J. Kupec, J. Hong, and A. Daoudi, “Real-Time Flying Object Detection with YOLOv8.” arXiv, May 17, 2023. Accessed: Oct. 27, 2023. [Online]. Available: <http://arxiv.org/abs/2305.09972>

[60] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.” arXiv, Jul. 06, 2022. Accessed: Oct. 27, 2023. [Online]. Available: <http://arxiv.org/abs/2207.02696>

[61] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection.” arXiv, May 09, 2016. Accessed: Nov. 02, 2023. [Online]. Available: <http://arxiv.org/abs/1506.02640>

[62] R. Timofte *et al.*, “NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results,” in *2017 IEEE Conference on Computer Vision*

and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA: IEEE, Jul. 2017, pp. 1110–1121. doi: 10.1109/CVPRW.2017.149.

[63] X. Wang, K. Yu, C. Dong, and C. C. Loy, “Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform.” arXiv, Apr. 09, 2018. Accessed: Nov. 03, 2023. [Online]. Available: <http://arxiv.org/abs/1804.02815>

Veri Setlerinin Aktif Linkleri:

1. DIV2K: http://data.vision.ee.ethz.ch/cvl/DIV2K/DIV2K_train_HR.zip
2. Flickr2K: <https://cv.snu.ac.kr/research/EDSR/Flickr2K.tar>
3. OST: https://openmmlab.oss-cn-hangzhou.aliyuncs.com/datasets/OST_dataset.zip

TEZDEN ÜRETİLMİŞ YAYINLAR

Makaleler

- [1] S. Yiğit, “Improving object detection of UAV images with Real-ESRGAN,” *Recent Adv. Sci. Eng.*, pp. 33–39, 2023, doi: 10.14744/rase.2023.0004.

