

# **TEKNOFEST**

## **HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ**

### **ULAŞIMDA YAPAY ZEKA YARIŞMASI**

#### **ÖN TASARIM RAPORU**

**BAŞVURU ID: 464684**

**TAKIM ADI**

**AI FULL STACK TEAM**

# İçindekiler

1. Takım Şeması.....	3
1.1. Takım Yapısı.....	3
1.2. Takım Görev Dağılımı.....	3
2. Kullanılması Planlanan Veri Setleri, Algoritmalar ve Akış Şeması.....	3
2.1. Veri Setleri.....	3
2.1.1. Problemin çözümünde kullanılacak veri setlerinin açıklanması.....	3
2.1.1.1. Komite Tarafından Sağlanan Veri Seti.....	3
2.1.1.2. Stanford Aerial Vehicle Data Set.....	4
2.1.1.3. VisDrone Data Set.....	4
2.1.1.4. Data Augmentation.....	4
2.1.1.5. Coco Data Set.....	4
2.1.1.6. Extra Veri Setleri ve Düzenlemeler.....	5
2.2. Algoritmalar.....	5
2.2.1. Problemin çözümünde kullanılacak algoritmaların açıklanması.....	5
2.2.2 Akış Şeması.....	7
3. Özgünlük.....	8
4. Proje Takvimi.....	9
5. Referanslar.....	9

## 1.Takım Şeması

### 1.1.Takım Yapısı

AI FULL STACK TEAM Takımı bir danışman öğretmen, bir takım kaptanı olmak üzere 2 kişiden oluşmaktadır. Danışman öğretmen fizik alanında yüksek lisans mezunu olup takım koordinasyonu, yarışma ile ilgili gelişmeleri takip eder ve genel araştırmalar ile uğraşmaktadır. Takım kaptanı ise lise 2. sınıfta olup yazılım alanında 2 senedir, yapay zeka alanında ise 1 yıldır aktif çalışmaktadır. Computer vision alanında bu zamana kadar birçok algoritma ve proje geliştirmiştir.

### 1.2.Takım Görev Dağılımı

Adı Soyadı	Ünvan	Okul	Projedeki Görevi
Bülent Ünal	Danışman Öğretmen	Mezun-Yüksek Lisans	Takım kordinasyonu, mentorluk ve araştırma.
Yiğit Can Özdemir	Takım Kaptanı	Adnan Oğuz Lisesi 2. Sınıf	Nesne tespiti algoritmasını geliştirmek, veri setini derlemek ve yazılım mimarisini oluşturmak.

## 2. Kullanılması Planlanan Veri Setleri, Algoritmalar ve Akış Şeması

### 2.1.Verit Setleri

- 1) Komite Tarafından Sağlanan Veri Seti
- 2) Stanford Aerial Vehicle Data Set
- 3) VisDrone Data Set
- 4) Data Augmentation
- 5) Coco Data Set
- 6) Extra Veri Setleri ve Düzenlemeler

#### 2.1.1.Problemin çözümünde kullanılacak veri setlerinin açıklanması

##### 2.1.1.1.Komite Tarafından Sağlanan Veri Seti

Komite tarafından sağlanan videoları frame frame ayırıp veri seti olarak kullanmayı planlıyoruz. Yarışma sırasında komite tarafından tahmin edilmesi için yarışmacılara veri seti

verilecek, verilecek veri seti komitenin mail yolu ile gönderdiği dokümandaki veri seti ile benzerlik göstereceği için kullanmayı uygun gördük.

#### **2.1.1.2.Stanford Aerial Vehicle Data Set**

Stanford Aerial Vehicle Data Set [2], Stanford Üniversitesi tarafından oluşturulmuş 30,000 görüntüden oluşan bir veri setidir. Çekilen görüntülerin açıları, yükseklikleri şartnamede ki şartları sağladığı için ve geniş bir veri seti olduğu için bu veri setini harici olarak dahil etmeyi düşünüyoruz.

#### **2.1.1.3.VisDrone Data Set**

VisDrone2019 [3] yarışmasında AISKYEYE Takımının veri madenciliği ve makine öğrenmesi ile oluşturduğu 288 videodan derlenmiş 10,209 statik görüntüyü eğitim kümesine eklemeyi planlıyoruz. Veri setinde ki görüntüler dökümandaki örnek görüntüler ile çok benzediği için tercih edilmesi uygun görülmüştür.

#### **2.1.1.4.Data Augmentation**

GAN(Çekişmeli üreticili ağ) ile sentetik veri üretilerek etiketler arası görüntü farkı en aza indirmeye çalışılacaktır. Aynı zamanda farklı filtreler ve gürültü ekleyerek farklı hava koşullarına uygun hale getirmeyi düşünüyoruz. İlerleyen zamanlarda ihtiyaç olarak görürsek AutoAugment [23] modülü kullanılacaktır.

#### **2.1.1.5.Coco Data Set**

Eğitim kümesine COCO veri setini de eklemeyi planlıyoruz COCO veri seti şartname de belirtilen yükseklik ve açılarda pek başarılı olamasa da dışarıdan eklenecek görüntüler ile dengelenmeye çalışılacaktır. COCO veri seti ile ve YOLO ile hali hazırda dökümanda verilen videodan frame alarak tahmin yaptırarak. Algoritmalar kısmında bu kısma daha detaylı değineceğiz.

### **2.1.1.6.Extra Veri Setleri ve Düzenlemeler**

İlerleyen aşamalarda algoritmanın veri seti üzerinde ki başarısına göre daha farklı veri setleri eklenebilir. Veri setindeki veri dağılımını düzeltme amaçlı veri seti üzerinde temizlik ya da eklemeler yapılabilir.

## **2.2.Algoritmalar**

### **1) YOLO-V5**

### **2) RetinaNet**

### **3) Faster R-CNN**

### **4) Mask R-CNN**

### **5) GAN Algoritmaları**

#### **2.2.1.Problemin çözümünde kullanılacak algoritmaların açıklanması**

Algoritmalar kısmında kullanmayı düşündüğümüz birçok aday var bu adaylar üzerinde en optimistik modeli ya da modellerin kombinasyonunu gerekli testleri yaparak değerlendireceğiz. Teknik şartname de yazıldığına göre yarışma sırasında bize 10 dakikalık video verilecek ve saniyede ki frame sayısı 7.5 fps olacak. Bu demek oluyor ki gerçek zamanlı evrişimli sinir ağı modelinin çok hızlı olması bir önem arz etmiyor. Realistik olursak yarışma bizden gerçek hayata yönelik çözüm istiyor saniyede 7.5 frame ve üzeri tahmin yapabilen bir model yeterli olacaktır. Bu yüzden baseline model olarak YOLO ve SSD'yi seçsek de bu modeller daha yavaş ama daha yüksek başarı oranına sahip modeller karşısında bu yarışmada dezavantaj kazanıyor. COCO veri seti ve YOLO-V3 modeli ile yaptığımız ufak test sonucunda komite tarafından sağlanan videodan bir frame alarak tahmin ettirdik.

**TEKNOFEST**  
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ





Şekil 1

YOLO için uzakta ki nesnelerin tahmini kısmında başarı oranını arttırmak için ve küçük objelerin tespiti için SPP [11] modülünü kullanabiliriz. Keras RetinaNet, Faster R-CNN [2], Mask R-CNN algoritmaları çok yüksek başarı oranları sayesinde kullanacağımız ana modeller olacaktır. Minumum bounding box tahmini için ise Grid R-CNN [12] algoritmasını kullanmayı planlıyoruz. Augmentation kısmında GAN (generative artificial neural) Network [13] kullanmayı düşünüyoruz, kullanacağımız algoritmalar ise CycleGAN, StyleGAN, pixelRNN ve DiscoGAN'dır. İlerleyen aşamalarda algoritmalar ile yaptığımız testlerin sonuçlarını verilerle beraber detaylı bir şekilde anlatacağız.

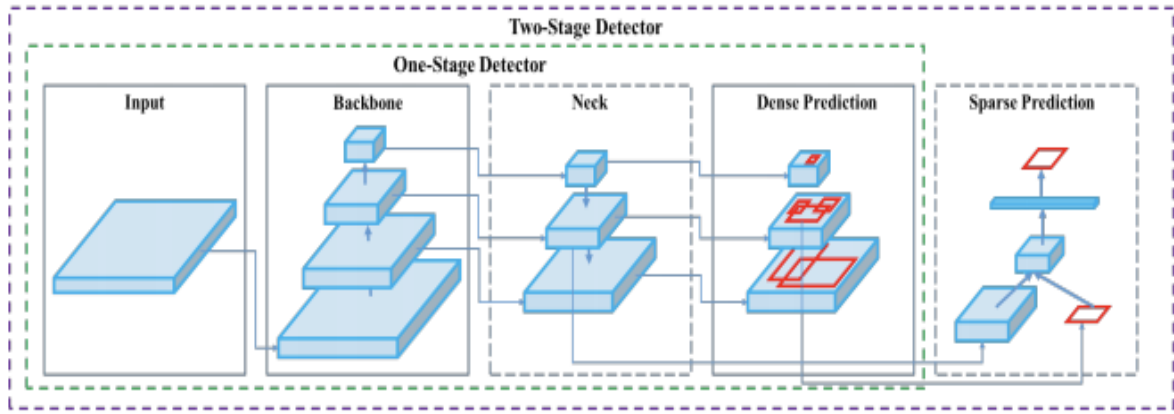
**YOLO Algoritması;** YOLO (You Look Only Once) [4] temelde R-CNN algoritmasıdır. YOLO-V4 modelini ele alırsak 4 pipeline(payplayn)'dan oluşmaktadır bunlar; Evrişim-Örnekleme bloğu, normalizasyon bloğu (dense connection block), SPP ve nesne tespiti bloğudur. YOLO yüksek başarısı ve saniye başı frame tahmini konusunda gayet hızlı olmasından dolayı çok popüler olmuştur. Projenin ileriki aşamalarında güncelliğinden ve daha iyi olmasından dolayı YOLO-V5 ile devam etmeyi planlıyoruz.

**Keras RetinaNet;** RetinaNet [14] tek aşamalı bir dedektöre sahip olmasına rağmen 2 dedektöre sahip Faster R-CNN modelleri gibi yüksek doğruluk oranları vardır bunun sebebi beraberlerinde getirdiği odak kaybıdır (focal loss). Odak kaybı kullanarak, “kolay” negatif örnekler tarafından daha düşük kayıp sağlanır, böylece model kayıp “zor” örneklere odaklanır. YOLO ile SSD modellerinin bir nevi birleşimi gibi olan bu modeli de değerlendirmeye alacağız.

Ekleme olarak üstte bahsettiğimiz ama açıklamasını yapmadığımız modeller var, bu modelleri kullanacağımız kesin olmadığından bahsetmedik. KTR aşamasına kadar gerekli testler yapılacaktır ve KTR de kullanılacak algoritmaların açıklamaları yapılacaktır.

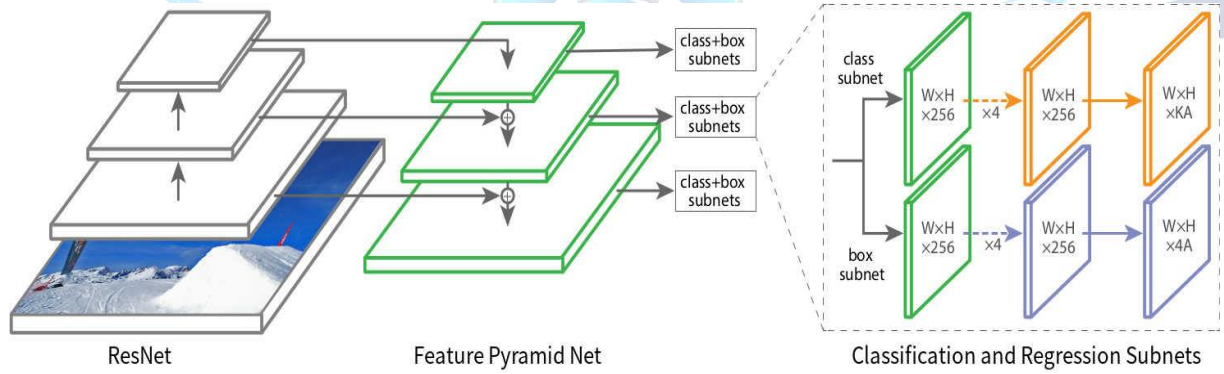
### 2.2.2. Akış Şeması

YOLO:



Şekil 2

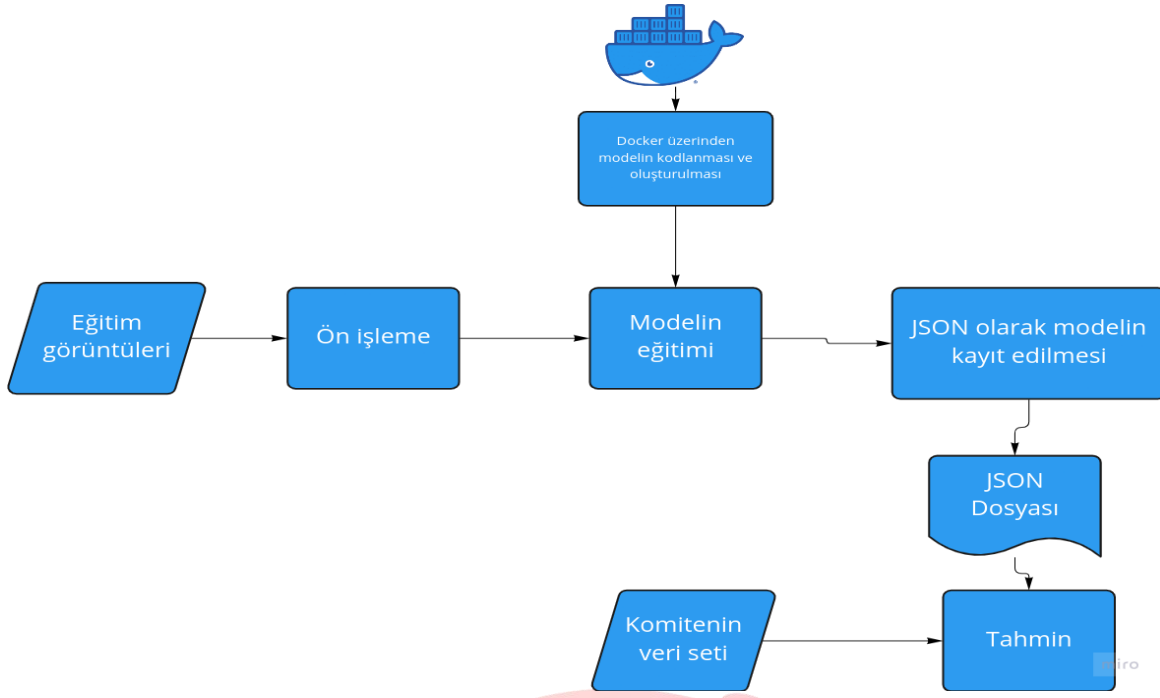
Retina Net:



Şekil 3

Yazılım Mimarisi:

Geliştireceğimiz yazılım mimarisinin genel hatları Şekil 4’ de gösterilmiştir.



Şekil 4

### 3.Özgünlük

YOLO modeli çok küçük nesnelerde muadillerine nazaran çok başarılı değil bunun için ise YOLO için geliştirilen SPP modelini kullanmayı düşündük. Bu model küçük nesneler ve uzakta ki nesneleri tespit etme konusunda modelin başarımını artırıyor.

Data augmentation kısmında GAN ile veri setini büyüteceğiz ve modelimizi farklı koşullarda yüksek başarımla elde etmek için GAN kullanacağız. Veri setimizde ki görüntülere gürültü, filtre ekleyeceğiz. GAN sayesinde fotoğrafları farklı açılardan derleyerek başarı oranını arttırmayı planlıyoruz.

Stanford Aerial Vehicle veri setinin daha önceki yıllarda ki projelerde kullanılmadığını gördük bu veri seti hem çok büyük hem de istenilen koşullar için gayet uygun. Bu veri seti dahil olmak üzere bir çok veri setinde tren, tramvay, monoray ve föniküler fotoğrafları diğer etikete sahip fotoğraflara nazaran azınlıktadır. Bu şekildeki veri dağılımı tren, tramvay, monoray ve föniküler objelerini tahmin ederken ki başarımın oranını düşürür o yüzden ekstra olarak veri madenciliği ile veri setinin dağılımını olabildiğince eşitlemeye çalışacağız.



#### 4.Proje Takvimi

14.04.2022-15.06.2022	Algoritma ve veri setleri için araştırma, ilk testlerin yapılması. Temel kodların yazılması.
15.06.2022	KTR'nin teslimi
15.06.2022-04.07.2022	Çevrimiçi Yarışma Simülasyonu için hazırlık. Kullanılacak modellerin kesinleştirilmesi.
04.07.2022	Çevrimiçi Yarışma Simülasyonu
04.07.2022-10.08.2022	Yazılım mimarisinin bitirilmesi, pipeline'ın bitirilmesi. Fine tuning yapılması. Son hazırlıklar

#### 5.Referanslar (Kaynakça) ve Rapor Düzeni

1. [https://drive.google.com/file/d/1E6ZEeZdf6pbqQYmNe62VrKz4PWAi\\_s5O/view](https://drive.google.com/file/d/1E6ZEeZdf6pbqQYmNe62VrKz4PWAi_s5O/view)
2. [https://cvgl.stanford.edu/projects/uav\\_data/](https://cvgl.stanford.edu/projects/uav_data/)
3. <https://github.com/VisDrone/VisDrone-Dataset>
4. <https://jonathan-hui.medium.com/yolov4-c9901eaa8e61>
5. <https://arxiv.org/pdf/2004.10934v1.pdf>
6. <https://odsc.medium.com/overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0>
7. <https://machinelearningmastery.com/object-recognition-with-deep-learning/>
8. <https://towardsdatascience.com/yolo-object-detection-with-opencv-and-python-21e50ac599e9>
9. [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)
10. <https://medium.com/teknopar-akademi/yolo-ile-opencv-ve-python-kullanarak-obje-tan%C4%B1ma-6303f0376c25>
11. [https://www.researchgate.net/figure/Architecture-of-SPP-module-used-in-YOLOv3-SPP3-We-integrate-a-SPP-module-in-YOLOv3\\_fig1\\_334694812](https://www.researchgate.net/figure/Architecture-of-SPP-module-used-in-YOLOv3-SPP3-We-integrate-a-SPP-module-in-YOLOv3_fig1_334694812)
12. [https://openaccess.thecvf.com/content\\_CVPR\\_2019/papers/Lu\\_Grid\\_R-CNN\\_CVPR\\_2019\\_paper.pdf](https://openaccess.thecvf.com/content_CVPR_2019/papers/Lu_Grid_R-CNN_CVPR_2019_paper.pdf)

13. <https://neptune.ai/blog/6-gan-architectures>
14. <https://towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fba6afabe4>
15. <https://medium.com/@mrvturan96/mask-r-cnn-ile-glioma-segmentasyonu-5da71ebaeb18>
16. <https://blog.zenggyu.com/en/post/2018-12-05/retinanet-explained-and-demystified/>
17. <https://medium.com/yavuzkomecoglu/teknofest19-yapayzeka-yarismasi-nesne-tespiti-retinanet-2e9ab25e7589>
18. <https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>
19. <https://www.analyticsvidhya.com/blog/2019/07/computer-vision-implementing-mask-r-cnn-image-segmentation/>
20. <https://arxiv.org/pdf/1506.01497.pdf>
21. <https://www.analyticsvidhya.com/blog/2018/11/implementation-faster-r-cnn-python-object-detection/>
22. <https://arxiv.org/pdf/1703.06870.pdf>
23. <https://arxiv.org/pdf/1805.09501.pdf>

