

TEKNOFEST

**HAVACILIK, UZAY VE TEKNOLOJİ
FESTİVALİ**

**ULAŞIMDA YAPAY ZEKA YARIŞMASI
KRİTİK TASARIM RAPORU**

**TAKIM ADI
AI FULL STACK TEAM**

BAŞVURU ID

464684



İÇİNDEKİLER

İÇİNDEKİLER.....	2
ŞEKİL LİSTESİ.....	3
TABLO LİSTESİ.....	4
KISALTMALAR.....	5
1. TAKIM ŞEMASI (5 PUAN).....	6
2. PROJE MEVCUT DURUM DEĞERLENDİRMESİ (15 PUAN).....	7
3. ALGORİTMALAR VE SİSTEM MİMARİSİ (25 PUAN).....	8
3.1. Veri Setleri (10 Puan).....	8
3.2. Algoritmalar (15 Puan).....	8
4. ÖZGÜNLÜK (25 PUAN).....	9
5. SONUÇLAR VE İNCELEME (25 PUAN).....	10
6. KAYNAKÇA (5 PUAN).....	11
RAPOR TASLAKLARI İLE İLGİLİ NOTLAR.....	12

ŞEKİL LİSTESİ

Tahmin Şekil 1

Tahmin Şekil 2

TABLO LİSTESİ

1. Kısaltma Tablosu

2. Takım Görev Dağılımı Tablosu

KISALTMALAR

SSD	Single Shot Detection
UAİ, UAP	Uçan Ambulans İniş, Uçan Araba Park
KTR	Kritik Tasarım Raporu
mAP	mean Average Precision
YOLO	You Only Look Once

Kısaltma Tablosu



1. TAKIM ŐEMASI (5 PUAN)

Ünvan	
Takım Kaptanı	<ul style="list-style-type: none">Konu hakkında bilimsel araştırma yapılması, pipeline oluşturulması, gerekli script (tool)'ların yazılması ve kullanılacak metotun belirlenmesi ve modelin seçilip eğitilmesi
Danışman Öğretmen	<ul style="list-style-type: none">Takım mentorluğu, iletişim sorumlusu ve ekip koordinasyonunun sağlanması.

Takım Görev Dağılımı Tablosu



2. PROJE MEVCUT DURUM DEĞERLENDİRMESİ (15 PUAN)

Ön Tasarım Raporundan sonra yapılan araştırmalarda YOLOV5-YOLOX5, Keras RetinaNet ve Faster R-CNN mimarileri baseline model olarak seçilmiştir. Buna ek olarak Mask R-CNN yedek mimari olarak seçilmiştir. Ön Tasarım Raporunda bahsettiğimiz SSD mimarisi ise kritik teslim raporundan sonra test edilip diğer mimariler ile sonuçları karşılaştırılacaktır. Ön Tasarım Raporunda bahsedilen veri setlerinin sayısının yeterli olmayacağını düşündüğümüz için Kaggle ve state of the art üzerinden ek veri seti kullanılmasına karar verilmiştir.

Buna ek olarak Ön Tasarım Raporunda eksikimiz olan UAİ ve UAP verilerinin nasıl oluşturulacağı konusu düşünülmüştür. Teknik şartnamede belirtilen renk ve ölçülere göre fotoğraflarda augmentation uygulanacaktır buna ek olarak komite tarafından sağlanan videodan UAİ ve UAP iniş bölgeleri etiketlenip frameleri veri setine eklenmiştir. Ön Tasarım Raporundan sonra pipeline oluşturulmaya başlanmıştır ve veri setleri için preprocess scriptleri yazılmıştır. Bu süre zarfında YOLOV5 modeli farklı checkpoint ağırlıkları ve mimarileri ile eğitilmiştir bunlara raporun ileri kısımlarında detaylı olarak bahsedeceğiz.



3. ALGORİTMALAR VE SİSTEM MİMARİSİ (25 PUAN)

3.1. Veri Setleri (10 Puan)

- **Visdrone 2019-DET**

288 video üzerinden derlenmiş 10,209 statik frameden oluşan Visdrone 2019-DET [1] veri seti teknik şartnamede belirtilen açılar ve ölçülere uygun olduğu için veri setinin eğitim kısmında kullanılması tercih edilmiştir. Veri setinin iyi tarafı çok fazla görüntüye ve araç çeşitliliğinin fazla olmasıdır. Dezavantajı ise veri setinde insan sayısı çok fazladır ve kuş bakışı çok görüntü yoktur bu yüzden ek veri setleri ile veri setinin dağılımını düzenleyeceğiz.

- **UAVDT**

State of the Art üzerinden kuş bakışı nesne tespiti modellerinin başarısını incelediğimizde UAVDT [2] veri setinin bir çok kuş bakışı yüksek irtifa görüntüsü içerdiğini ve bu görüntülerden bir kısmının tozlu, fırtınalı havalardan görüntüler içerdiği fark edilmiştir farklı hava koşullarına uygun olması ve Visdrone veri setinde ki kuş bakışı veri eksikliğini kapatması için UAVDT veri setinin bir kısmı kullanılmıştır. UAVDT veri seti herhangi bir labelmap içermez sadece görüntü içerir bu yüzden etiketleme kısmını manuel olarak yaptık.

- **Komite Örnek Videosu**

T3 vakfı tarafından yarışma icabında gönderilen veri seti farklı hava koşullarına sahip olduğu için ve yarışma için bir örnek teşik ettiği için validasyon ve bir kısmında eğitim için kullanılmasına karar verilmiştir. Video OpenCV kütüphanesi üzerinden yazdığımız script sayesinde saniyede 30 frameden oluşan videoyu, 4 saniyede 1 frame alacak şekilde kestik bu frameleri etiketledik ve büyük kısmını validasyon küçük kısmını eğitim olacak şekilde kullanacağız.

- **Stanford Aerial Vehicle Data Set**

Stanford Aerial Vehicle Data Set [3] , Stanford Üniversitesi tarafından oluşturulmuş 3942 görüntüden oluşan kuş bakışı bir veri setidir. Yaptığımız testlerde modelimiz insanları algılamakta bazı durumlarda zorluk çekiyor ve bazı küçük nesneleri insan sanıyor Standford veri setinin çok büyük bir kısmını insanlar oluşturuyor aynı zamanda şartnamede belirtildiği gibi bisikleti ya da motosiklet hareket halindeyken insanların etiketlenmemesi gerekiyor. Standford veri setinde bisiklet ve motosiklet görüntüleride bolca bulunmaktadır lakin veri setinin tamamı eğitim setine dahil edilmesi olumsuz sonuç getirecektir (sınıf bazında dağılımı etkiliyor) bu yüzden sadece bir kısmı eğitim setimize eklenecektir.

- **Data Augmentation**

Augmentation kısmında UAİ ve UAP iniş bölgeleri için sentetik veri üretimi metodu kullanılacaktır. Modelin zor ve uç şartlarda tahmin doğruluğunu arttırmak için AutoAugment [4] modülünü ileriki aşamalarda kullanmayı planlıyoruz.

- **Coco Veri Seti**

Coco veri setini sadece farklı modellerin tipik başarısını ölçmek için bir metrik olarak ve test için kullanacağız. Coco veri seti kuş bakışı görüntülerden oluşmadığı için eğitim kısmında kullanılması modelin başarısını düşürecektir.

3.2. Algoritmalar (15 Puan)

- **YOLOV5-YOLOX**

YOLO mimarisi gerçek zamanlı olarak nesne algılamayı tek bir uçtan-uca modelle gerçekleştirebilen, son teknoloji sonuçlar elde eden evrişimli sinir ağı mimarisidir[5]. En büyük avantajı diğer modellere nazaran çok daha hızlı olmasıdır ek olarak eğitim süresi de kısadır. Biz güncelliği ve en yüksek performansı vermesinden dolayı YOLOV5 ve YOLOX mimarilerini tercih ettik şuan ki veri setimiz ile YOLOV5 modeli eğitilmiştir donanım sorunumuzdan dolayı YOLOX modeli KTR aşamasından sonra eğitilecektir.

- **Faster R-CNN**

R-CNN mimarisi region tabanlı evrişimli sinir ağı mimari gruplarını içermektedir, Faster R-CNN ise bu mimarinin bir ürünüdür. R-CNN mimarisi resme region proposals uygulayarak bir çok öneri bölgesine böler bu sayede çok fazla nesne içeren büyük ölçekli resimlerde nesne tespiti için doğruluk artar. Faster R-CNN modelinde ilk önce resme CNN uygulanıp özellik haritası çıkarılır sonra bölge önerileri bir ağ oluşturularak alınır [6]. Faster R-CNN eski R-CNN modellere göre farklı burada ortaya çıkıyor oluşturulan öneri ağında resmi tahmin etmek yaklaşık 0.3 saniye sürüyor. YOLO modeline karşı daha yüksek doğruluk ve region tabanlı olmasının ötürü bu modeli YOLO ile beraber baseline model olarak seçtik. Dezavantajı uzun eğitim süresi ve saniye başına frame tahminin düşük kalmasıdır.

- **Keras RetinaNet**

YOLO modelinin hızı iyi olmasından ama yeterli gelmeyen mAP puanından ve Faster R-CNN'in yavaş lakin yüksek mAP puanından dolayı Keras RetinaNet kullanılması düşünülmektedir. Keras RetinaNet temelde YOLO ve SSD mimarilerinden esinlenerek ortaya çıkmıştır. Faster R-CNN'e karşı daha hızlıdır ve YOLO'dan daha yüksek başarıma sahiptir bunun sebebi Focal-Loss ve FPN (Feature Pyramid Network) yapısını beraberinde getirmesidir[7]. KTR raporundan sonra Keras RetinaNet ile testler yapıp deneysel testler diğer modeller ile karşılaştırılacaktır yüksek potansiyel barındırdığı için baseline model olabilecek kapasitedir.

4. ÖZGÜNLÜK (25 PUAN)

Preprocessing kısmında veri setlerinin ayarlamak için 2 farklı script yazılmıştır ve bu scriptler otomasyon sağlaması için pipeline'a eklenmiştir. Önceki sayfalarda bahsettiğimiz OpenCV kütüphanesi tarafından yazılan script sayesinde veri setinin hazırlanma süreci kısaltılmıştır ve istenilen parametreler ile video framelere bölünebilir olmuştur. Bir diğer script ise veri setlerinde class sorunu yaşadığımız için ve zamandan kazanmak için yazılmıştır. Veri setimiz sabit olmadığı için custom veri seti kullanıyoruz bu yüzden farklı veri setlerinden gelen labelmapler ile uyumsuzluk söz konusu oluyor bütün işleri elle yapmak çok uzun süreceğinden dolayı labeları indirgemeyi başardık. Script labelmap belgelerini içeren dizini tarıyor ve txt dosyalarının içinde ki class kolonunda ki labeları istenilen (0,1,2,3) classa çeviriyor.

Yağmurlu, karlı, tozlu tozlu hava şartlarında kuş bakışı görüntüler veri setine eklenmiştir bu sayede model zor şartlar altında yüksek başarı gösterecektir yaptığımız komite tarafından gönderilen video üzerinde testlerde karlı hava modelin başarısını düşürmemiştir ve üstü karlı arabalar tespit edilebilmiştir. Ek olarak VisDrone veri seti gece görüntülerini de içeriyor. Yaptığımız testlerde nesneleri fazla etikete ayırmak (araba, karavan, kamyon, otobüs vs.gibi) karar matrisini kötü etkiledi ve başarı oranı düşük kaldı bu yüzden bütün araçları taşıt olarak sınıflandırdık sonucunda başarı oranımızın arttığını gördük. Tahmin edilmesi gereken 4 sınıf olduğundan dolayı öz nitelik çıkartmak 4 sınıf içinde daha kolay olacaktır. Data Augmentation metodları ile de veri setinin ileriki aşamalarda çeşitlendirilmesi düşünülmektedir ek olarak kuş bakışı görüntüler ile veri seti zenginleştirilecektir.

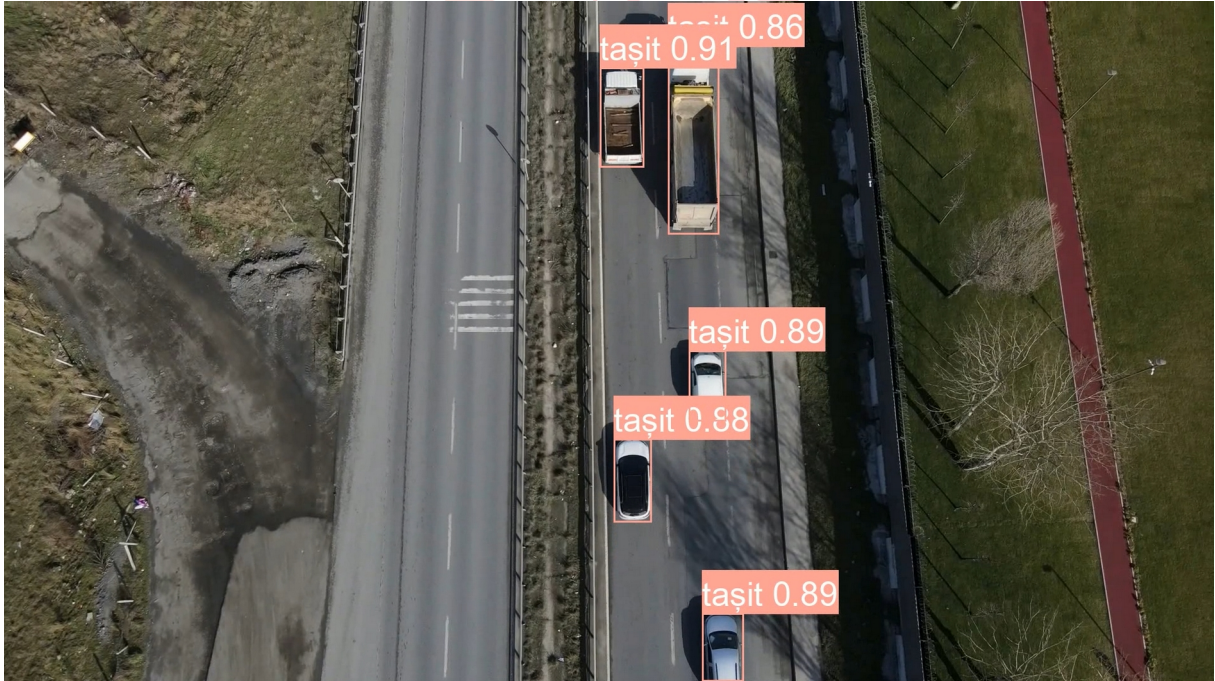
YOLOV5 ile büyük ölçekli nesne algılamak için sliced inference ve SAHI paketi [8] kullanılarak fazla nesne içeren büyük resimlerde bölgesel tahmin yaptırarak başarı oranını artırılması düşünülmektedir.



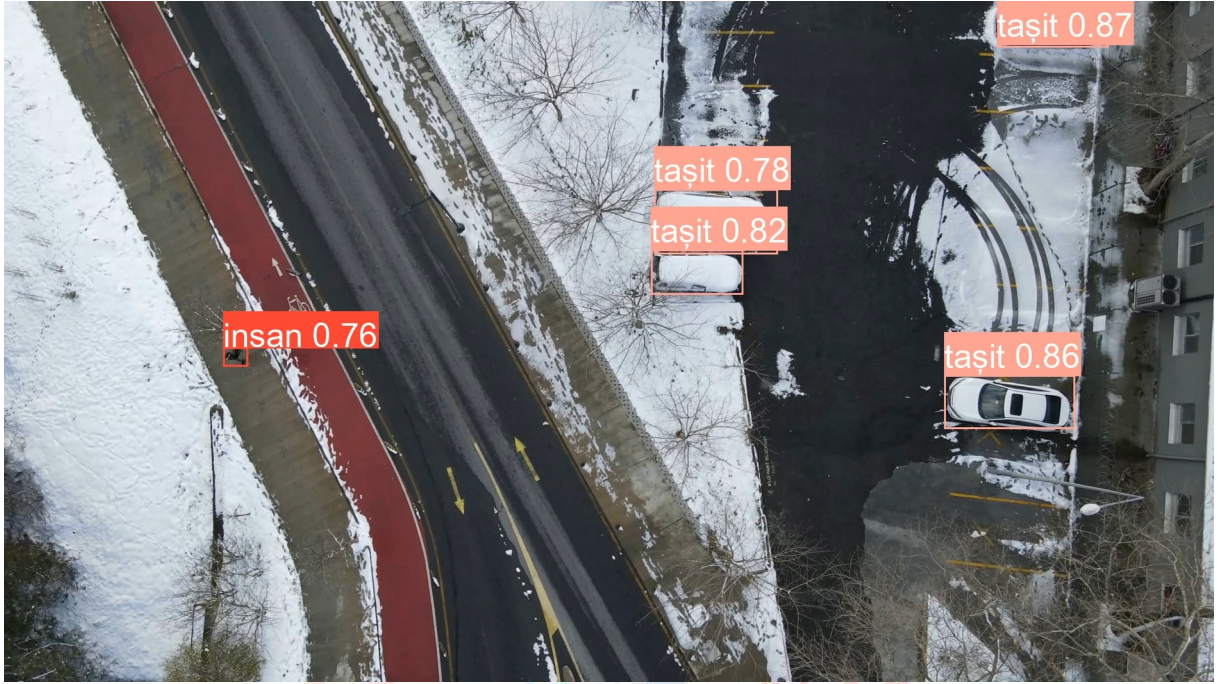
5. SONUÇLAR VE İNCELEME (25 PUAN)

Ön Tasarım Raporunda bahsedilen YOLOV5 mimarisi custom veri setimiz (Custom veri seti Visdrone veri seti, komite tarafından gönderilen video frameleri ve UAVDT veri setini içerir) ile eğitilmiştir. Eğitim sırasında Google Colab üzerinde Nvidia Tesla 4 GPU'su kullanılmıştır ama görülmüştür ki yüksek ağırlıklı checkpointler ile büyük veri setleri için Colab yetersiz kalmıştır bu yüzden KTR'den sonra donanım yükseltilmesi ile büyük modeller üzerinde veri seti test edilecektir. Colab üzerinden medium size ağırlıklar ile veri seti 2 kez eğitilmiştir. İlk testimiz 50 epoch ve 32 batch size olmak üzere 10 class olmak üzere Visdrone veri seti eğitilmiştir. Yukarıda da bahsedildiği gibi classların fazla olması karar matrisini kötü etkilemiş ve mAP_0.5 metriği üzerinden 0.32 puan almıştır.

Bir sonraki testimiz ise Visdrone veri setinine ek olarak komite tarafından gönderilen videodan frameler eklenmiş ve UAVDT veri setinin küçük bir kısmı eklenmiştir. Yazdığımız script sayesinde labellar 4'e indirgenmiş ve 100 epoch 32 batch olmak üzere tekrar eğitilmiştir sonuçlar ise şaşırtıcı şekilde artmıştır mAP metriği üzerinden 0.677 değeri elde etmiştir. Recall metriği üzerinden 0.7419 puan elde etmiştir. Komite videosu üzerinden tahmin ettirdiğimiz kareler aşağıda yer almaktadır.



Tahmin Şekil 1



Tahmin Şekil 2

Komite videosu test videosu olarak kullanılmıştır ve birkaç durum dışında model yüksek başarı göstermiştir bu durumlar:

Model trafik dubası gibi bazı küçük nesneleri insan olarak tespit etmiştir. Yürüyen insanların önüne ağaç geldiğinde insan anlık tespit edilememesi sorunu yaşanmıştır. Karlı havada bazı evler taşıt olarak algılanmıştır. Yuvarlak mavi cisimler UAP iniş alanı olarak algılandığı durumlar tespit edilmiştir.

Bu sorunların çözümü olarak KTR aşamasından sonra veri setine kuş bakışı insan görüntülerinin eklenmesi, karlı hava görüntülerinin daha fazla eklenmesi ve augmentation yöntemi ile UAP ve UAI alanları için görüntü oluşturulması ön görülmüştür. Ek olarak donanım yetersizliğinden dolayı görüntüler 640x640 ölçekte eğitilmiştir küçük nesnelerin yanlış tespit edilmesinin bu yüzden olduğu düşünülmektedir. KTR aşamasından sonra 1280x1280 veya 800x800 ölçekte eğitim planlanmaktadır. Keras RetinaNet, Faster R-CNN ve SSD mimarileri aynı veri seti üzerinden eğitilip mAP metriği üzerinden değerlendirilecektir. Ve YOLOX modeli de eğitilip değerlendirilecektir.

6. KAYNAKÇA (5 PUAN)

- [1] <https://github.com/VisDrone/VisDrone-Dataset>
- [2] <https://paperswithcode.com/dataset/uavdt>
- [3] https://cvgl.stanford.edu/projects/uav_data/
- [4] <https://paperswithcode.com/method/autoaugment>
- [5] <https://medium.com/teknopar-akademi/yolo-ile-opencv-ve-python-kullanarak-obje-tan%C4%B1ma-6303f0376c25>
- [6] <https://teknofesor.com/cnn-rcnn-fast-rcnn-faster-rcnn-nedir/>
- [7] <https://medium.com/yavuzkomecoglu/teknofest19-yapayzeka-yarismasi-nesne-tespiti-retinanet-2e9ab25e7589>
- [8] <https://medium.com/machine-learning-t%C3%BCrkiye/yolov5-ba%C5%9Flang%C4%B1%C3%A7-kiti-e666252a1cb4>
- [9] <https://github.com/ultralytics/yolov5>
- [10] <https://github.com/obss/sahi>

