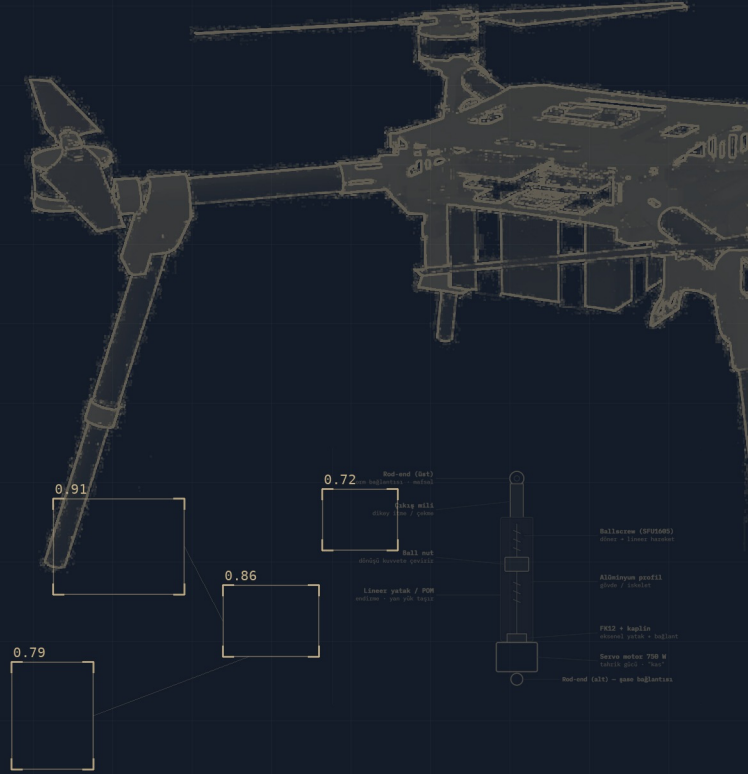


# Serdar Anıl Demirbaş

Yazılım Mühendisi · Gerçek zamanlı görüntü işleme, bilgisayarlı görü ve otonom sistemler.

2021'den bu yana savunma sanayiinde gömülü görüntü işleme hatları, hedef tespit-takip algoritmaları ve operatör arayüzleri geliştiriyorum.



loss / epoch

0,904

MAP50 · HAVA  
GÖRÜNTÜSÜ TESPİTİ

5

KİŞİLİK YAZILIM EKİBİ  
· TAKİM LİDERİ

5

YIL SAVUNMA SANAYİİ  
DENEYİMİ

9

PORTFOLYODAKİ PROJE

# Ne yapıyorum

Mekatronik mühendisliği temeli üzerine yazılım geliştiriyorum. Uzmanlık alanım **gerçek zamanlı görüntü işleme** ve bu işlemenin **gömülü donanımda çalışır hâle getirilmesi**: tespit ve takip algoritmaları, veri seti hazırlama ve model eğitimi, operatör arayüzleri, ve bunların NVIDIA Jetson gibi sınırlı kaynaklı platformlara taşınması.

## PORTFOLYO İÇERİĞİ

### 01 · OTONOM İHA GÖRÜNTÜ İŞLEME

RoboNation SUAS 2026 · Yazılım Takım Lideri  
YOLOv11 · ROS 2 · Jetson · TensorRT

### 02 · SPOR SAHASI VİDEO ANALİZİ

Freelance · Fransa merkezli girişim  
YOLO · ByteTrack · Olay tespiti · Otomatik kurgu

### 03 · ÇOK SATICILI PAZARYERİ (MUDİTA)

Freelance · Tam yığın  
Django · Next.js · PostgreSQL · Docker

### 04 · YARIŞ SİMÜLATÖRÜ PLATFORMU

Ürün sahibi · Çalışan prototip  
AC servo · Gömülü kontrol · ESP32 telemetri

### 05 · SAVUNMA SANAYİİ DENEYİMİ

Digitest · Wupsoft  
Gizlilik gereği sanitize edilmiş anlatım

### 06 · ERKEN DÖNEM PROJELER

Endüstriyel robot · Otonom drone · Elektrikli araç  
Lisans dönemi mühendislik çalışmaları

## ÇALIŞTIĞIM TEKNOLOJİLER

|         |            |       |               |              |         |            |         |                |          |
|---------|------------|-------|---------------|--------------|---------|------------|---------|----------------|----------|
| Python  | C++        | C#    | MATLAB        | OpenCV       | YOLOv11 | ByteTrack  | PyTorch | Albumentations | TensorRT |
| CUDA C  | OpenMP     | ROS 2 | NVIDIA Jetson | Raspberry Pi | ESP32   | Qt Creator | Unity   | Django         |          |
| Next.js | PostgreSQL | Redis | Docker        | TCP/UDP/UART |         |            |         |                |          |

# Otonom İHA Görüntü İşleme ve Görev Yazılımı

ROL

Yazılım Takım Lideri

KURUM

Atılım UAV

DÖNEM

2025 - Devam

EKİP

5 kişilik yazılım ekibi

YARIŞMA

RoboNation SUAS 2026



KA-FA1500 · takımın yarışma için geliştirdiği insansız hava aracı

## PROBLEM

RoboNation SUAS, insansız hava aracının 20-40 metre irtifadan otonom uçarak sahadaki hedefleri tespit etmesini, sınıflandırmasını ve göreve göre aksiyon almasını istiyor. Zorluk şurada: bu irtifada hedefler görüntüde yalnızca 80-120 piksel; üstelik tespit, sınırlı güç bütçesine sahip gömülü bir bilgisayarda gerçek zamana yakın çalışmak zorunda.

## BENİM SORUMLULUĞUM

- Hedef tespiti/sınıflandırma, otonom navigasyon ve görev planlama yazılım mimarisini kurmak
- 5 kişilik yazılım ekibini yönetmek
- Veri hazırlama hattını ve eğitim parametrelerini belirlemek
- Teknik Tasarım Raporu (TDR) ve şartnameye uygun takım web sitesi

0,904

MAP50

0,691

MAP50-95

~15.000

EĞİTİM GÖRÜNTÜSÜ

Eyl 2026

TULSA, OKLAHOMA

## 20-40 m irtifadan hedef tespiti



Eğitilen YOLOv11m modelinin doğrulama çıktıları: farklı irtifa, zemin ve ışık koşullarında güven skorlarıyla birlikte

0,904

MAP50

0,691

MAP50-95

~15.000

EĞİTİM GÖRÜNTÜSÜ

33

YAKINSAMA EPOCH'U

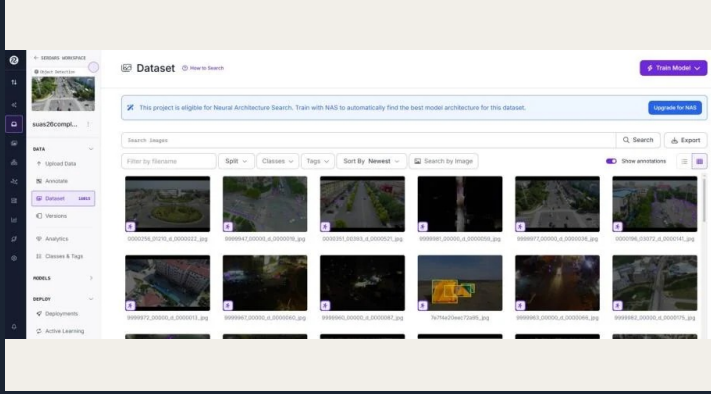
### 1 · VERİ HAZIRLAMA

Ham 4K drone görüntülerini YOLO'nun sabit girdi boyutuna indirmek için **%40 örtüşmeli 1280×1280 karolara** bölen bir hat tasarladım. Örtüşme kritik: hedef bir karo sınırına denk geldiğinde en az bir karoda tam görünmesini garantiliyor. Train/validation ayırımını **görüntü bazında** yaptım: aynı sahnenin karoları ikiye bölünürse model sızıntıyla öğrenir ve validation metriği yalan söyler.

### 2 · SINIF DENGEME

Ham veride sınıflar dengesizdi. Çevrimdışı augmentation ile **sınıf bazlı çarpan** uyguladım: az örnekli sınıfa 8x, çok örnekliye 3x. Böylece ~2.100 ham görüntüden ~15.000 görüntülük, 1:1,2 oranlı dengeli bir eğitim seti çıktı. Renk augmentasyonunu bilinçli olarak bu adımda değil eğitimde bıraktım: iki katmanda birden uygulanırsa çift augmentation olur.

# Veri setinden gömülü çıkarıma



Etiketli veri setinin yönetimi ve sürülenmesi



Eğitim döngüsü ve kayıp eğrisi

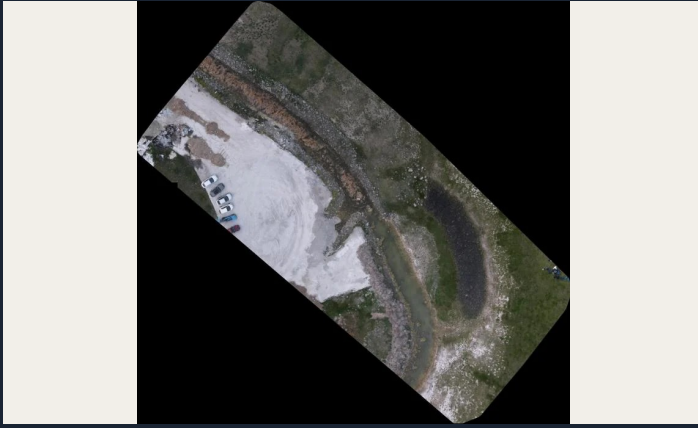
## 3 · EĞİTİM

YOLOv11m üzerinde transfer öğrenme. Model 33. epoch'ta yakınsadı, erken durdurma 48'de devreye girdi: yani **veri tavanına ulaşıldı**, daha fazla epoch değil daha çeşitli veri gerekiyor. Bu teşhisi çıkardıktan sonra 13 maddelik bir iyileştirme planı hazırladım: çözünürlük artırımı, farklı irtifa ve ışık koşullarından veri toplama, kısmi örtülme augmentasyonu.

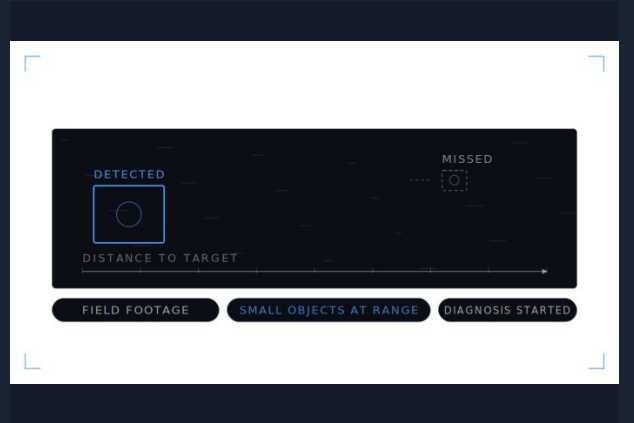
## 4 · GÖMÜLÜ DAĞITIM

Model NVIDIA Jetson üzerinde **TensorRT FP16** ile gerçek zamanlı çalışacak şekilde konumlandırıldı. Burada eğitim ve export çözünürlüğünün eşitlenmesi gerekiyor; farklı bırakılırsa model sahada beklenmedik davranıyor.

|          |                |         |          |
|----------|----------------|---------|----------|
| YOLOv11m | Albumentations | PyTorch | TensorRT |
| Jetson   | OpenCV         | ROS 2   |          |



Uçuş sonrası ortomosaik: kareler tek harita hâlinde birleştirildi



Mesafeye bağlı tespit başarımı

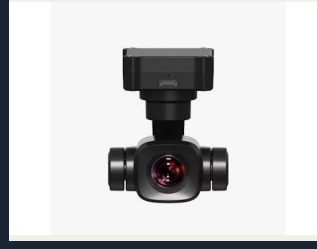
## Aviyonik ve hesaplama mimarisi



Jetson Orin NX  
Görüntü işleme bilgisayarı



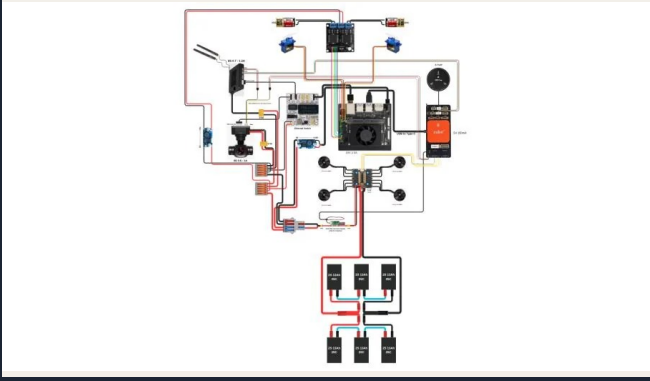
Cube Orange+  
Uçuş kontrolcüsü



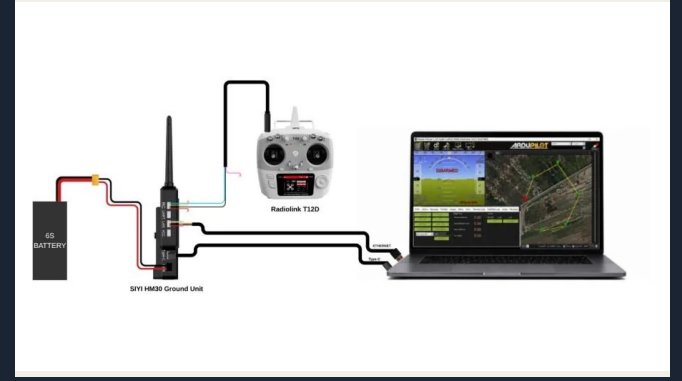
SIYI A8 Mini  
Gimballi kamera



T-Motor U7  
İtici sistemi



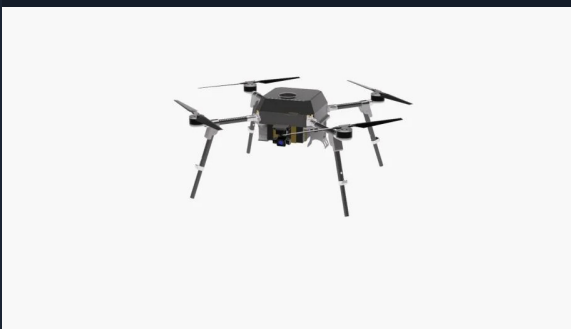
Güç dağıtımı ve kablolama mimarisi



Yer istasyonu haberleşme zinciri: HM30 datalink, kumanda, görev planlayıcı

### YAZILIM MİMARİSİ KARARI

Görüntü işleme ile uçuş kontrolünü ayrı işlemcilerle böldük: uçuş kontrolcüsü yalnızca uçuşa odaklanıyor, companion computer görüntü işlemeyi üstleniyor, ikisi arasındaki iletişim ROS 2 üzerinden yürüyor. Bu ayrım, görüntü işleme yükü arttığında uçuş güvenliğinin etkilenmemesini sağlıyor: otonom sistemlerde en kritik tasarım kararlarından biri.



KA-FA1500 · bütünleşik tasarım



Görev yükü gövdesi



Bırakma mekanizması, montajlı

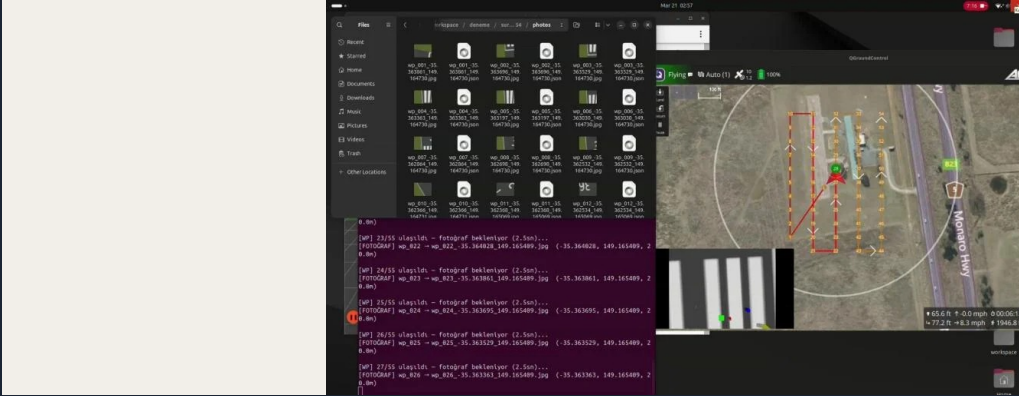
# Uçuş testleri ve proje dokümantasyonu



KA-FA1500 saha testinde



Uçuş öncesi hazırlık ve ekipman kurulumu



Otonom görev planlama ve uçuş sonrası fotoğraf konumlandırma: yer istasyonu ekranı

## EK ÇIKTILAR

Yazılım işinin yanında takımın **Teknik Tasarım Raporu**'nu ve yarışma şartnamesinin web sitesine dair gereksinimlerini karşılayan **takım sitesini** hazırladım. Site, tasarımı Framer'da kurulduktan sonra kendi koduna taşındı; 51 sayfa, 28 teknik blog yazısı ve beş teknik belge içeriyor.

TDR

28 teknik yazı

51 sayfa

atilimuav.com



Proje tanıtım sayfası

# Spor Sahası Video Analizi

ROL

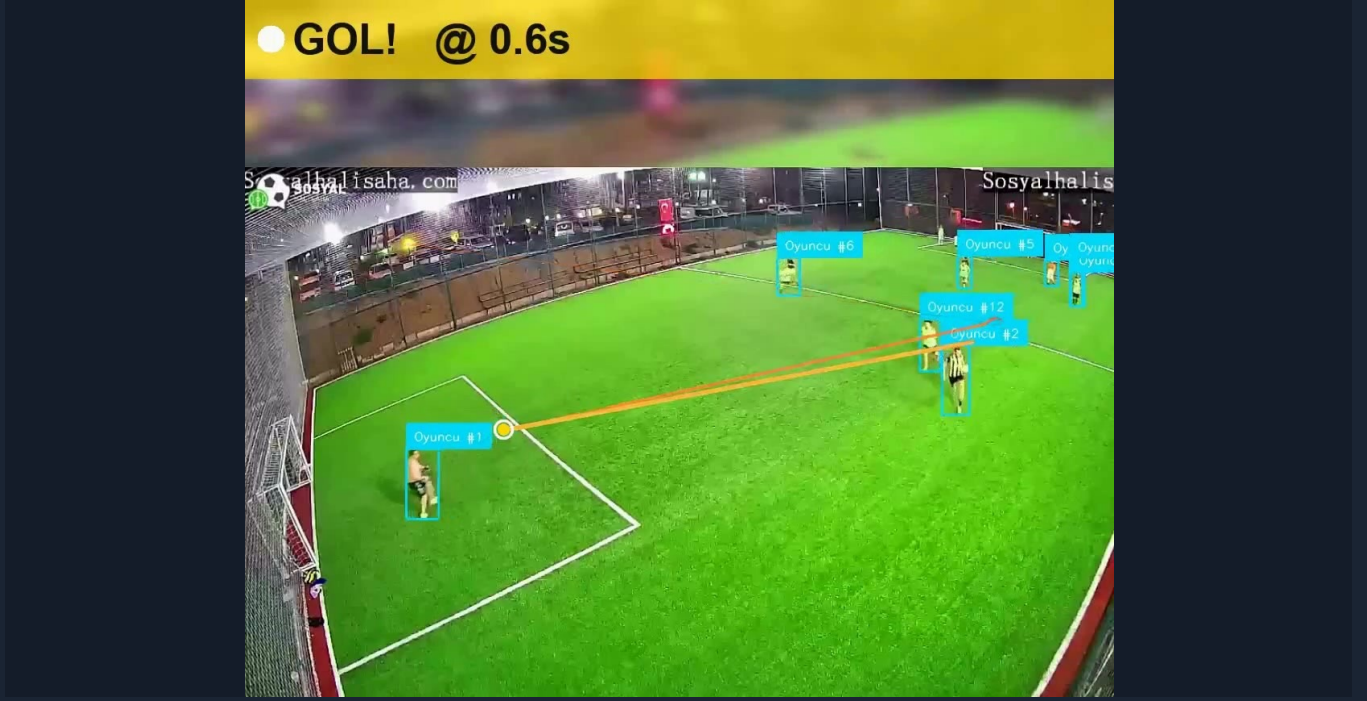
Freelance geliştirici

MÜŞTERİ

Fransa merkezli spor teknolojisi girişimi

DÖNEM

2026 – Devam



Gol anı otomatik tespit edildi ve zaman damgasıyla işaretlendi: sistem çıktısından bir kare

## PROBLEM

Halı saha işletmeleri maçları tek sabit geniş açı kamerayla kaydediyor. Oyuncuların istediği ise saatlerce ham kayıt değil, kendi golünün 15 saniyelik klibi. Bunu insan eliyle kurgulamak ölçeklenmiyor.

## ÇÖZÜM

Tek kameradan oyuncu ve top tespiti, kalıcı kimlikli takip, top yörüngesinden olay çıkarımı ve olayların etrafından otomatik klip kesimi yapan uçtan uca bir hat. Çıktı: işaretlenmiş maç videosu, olay bazlı klipler ve birleşik highlight reel.

1

KAMERA · SABİT GENİŞ AÇI

3

ÇIKTI TÜRÜ · VIDEO, KLİP, REEL

H.264

TARAYICI UYUMLU KODLAMA

YOLO

ByteTrack

OpenCV

PyTorch

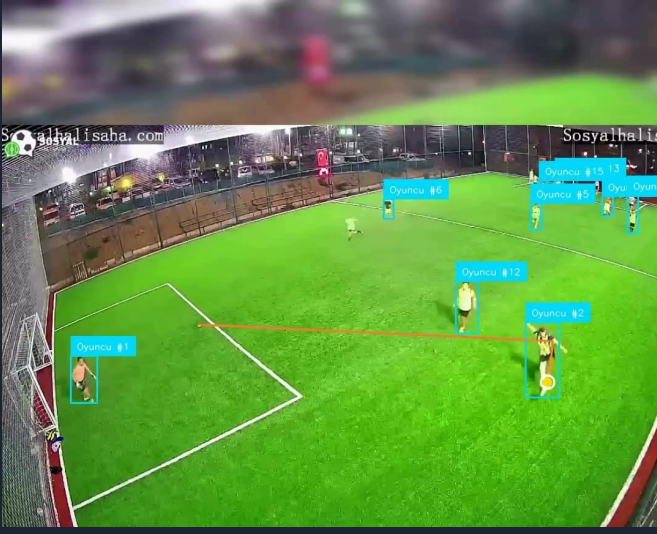
supervision

scikit-learn

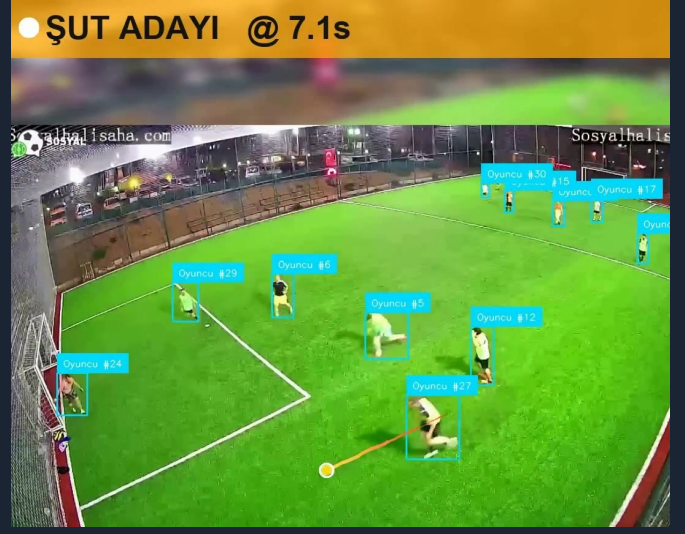
ffmpeg

Streamlit

# Tespit, takip ve olay çıkarımı



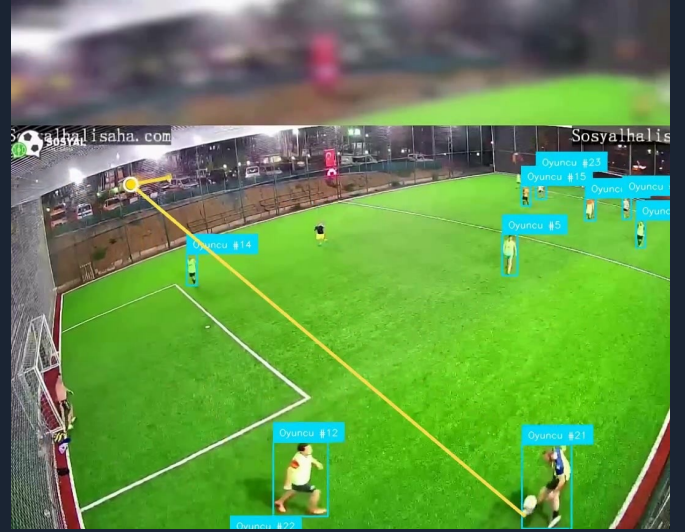
Oyuncu tespiti ve kalıcı kimlik atama (ByteTrack)



Şut adayı olayı: top yörüngesindeki hız artışından tespit



Forma renginden KMeans ile takım sınıflandırma



Top yörüngesi izleme: kare arası konum bağlantısı

## HAT

- **Tespit:** YOLO (Ultralytics) ile oyuncu ve top
- **Takip:** ByteTrack ile kalıcı oyuncu kimlikleri
- **Olay:** Top yörüngesindeki ani hız artışından şut/gol adayı
- **Kurgu:** Olay etrafından klip kesimi + birleşik reel
- **Çıktı:** ffmpeg ile tarayıcı uyumlu H.264
- **Arayüz:** Streamlit ile sunulabilir demo

## MÜHENDİSLİK NOTU

Örnek videolar elde/gimbalde çekildiği için kale bölgesi kalibrasyonu yapılamadı; bu yüzden olaylar "gol" yerine **şut adayı** sezgiseliyle işaretleniyor. Olay tespiti fonksiyonu kale bölgesi parametresi kabul edecek şekilde yazıldı: prodüksiyondaki sabit kamera kurulumunda aynı kod tam gol tespitine dönüşüyor. Sınırı baştan tanımlamak, sonradan mimariyi değiştirmekten ucuz.

YOLO

ByteTrack

OpenCV

PyTorch

supervision

scikit-learn

ffmpeg

Streamlit

# Çok Satıcı E-Ticaret Pazaryeri

| ROL                   | DÖNEM | KAPSAM                       | ÖLÇEK                      |
|-----------------------|-------|------------------------------|----------------------------|
| Freelance · Tam yığın | 2026  | Backend + Frontend + Dağıtım | 10 alan modülü · 36 commit |

Alıcı, satıcı ve platform arasında para akışı olan bir pazaryeri. Bu tür sistemlerde en pahalı hata para hatasıdır: bu yüzden mimari kararların çoğu **doğruluk ve denetlenebilirlik** etrafında verildi. Müşteri gizliliği gereği arayüz görselleri paylaşılmamaktadır; anlatım mimari düzeydedir. **Sistem, talep hâlinde yüz yüze görüşmede gösterilebilir.**

## DÖRT KRİTİK KARAR

### ÇOK SATICILI PİVOT

Alicının tek ödemesi, satıcı bazlı alt siparişlere otomatik bölünüyor. Her satıcı yalnızca kendi siparişini görüyor; kargo, iade ve hakediş bu alt sipariş üzerinden yürüyor. Tek sipariş modeliyle başlanırsa bu ayırım sonradan eklenemiyor.

### ÇİFT KAYITLI DEFTER

Satıcı hakedişi tek bir bakiye alanında tutulmuyor. Her hareket ayrı bir satır: başarılı ödeme CREDIT, iptal edilen lojistik otomatik DEBIT yazıyor. Bakiye her seferinde toplanarak hesaplanıyor, hiçbir zaman üzerine yazılmıyor: muhasebenin yüzyıllık yöntemi, denetlenebilirliği ücretsiz getiriyor.

### ÖDEME SAĞLAYICI SOYUTLAMASI

Ödeme mantığı Adapter Pattern ile view katmanından tamamen ayrıldı. Sağlayıcı değiştiğinde iş mantığına dokunulmuyor. Senkron doğrulamalar çift harcama kilidiyle, asenkron webhook'lar idempotency kilidiyle korunuyor: tarayıcı kapansa bile ödeme durumu tutarlı kalıyor.

### TEK ŞEMALI API

Özel bir DRF renderer ve exception handler ile **her** yanıt: 200 de olsa 500 de olsa: aynı JSON şemasında dönüyor: `success / data / error`. Frontend'in yanıt şeklini tahmin etmesi gereken hiçbir durum kalmıyor.

## ALTYAPI

Medya yüklemeleri sunucu diskini hiç görmeden boto3 ile Cloudflare R2'ye (S3 uyumlu) akıtılıyor; böylece uygulama durumsuz kalıyor ve yatay ölçeklenebiliyor. Dağıtım Docker Compose ile altı servis üzerinden yürüyor, sürekli entegrasyon GitHub Actions'ta.

## YIĞIN

|                 |                |            |       |        |
|-----------------|----------------|------------|-------|--------|
| Django 5        | DRF            | PostgreSQL | Redis | Celery |
| Django Channels | Next.js        | TypeScript |       |        |
| Tailwind        | NextAuth       | Docker     | Nginx |        |
| Cloudflare R2   | GitHub Actions |            |       |        |

10

ALAN MODÜLÜ

6

DOCKER SERVİSİ

2

AYRI REPO · API + ARAYÜZ

1

TEK API YANIT ŞEMASI

# Yarış Simülatörü Hareket Platformu

ROL

Ürün sahibi ve geliştirici

DURUM

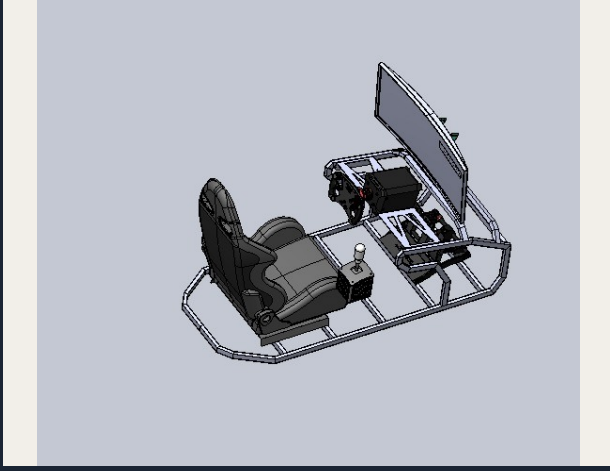
Çalışan prototip

MİMARİ

3DOF / 4DOF

TAHRİK

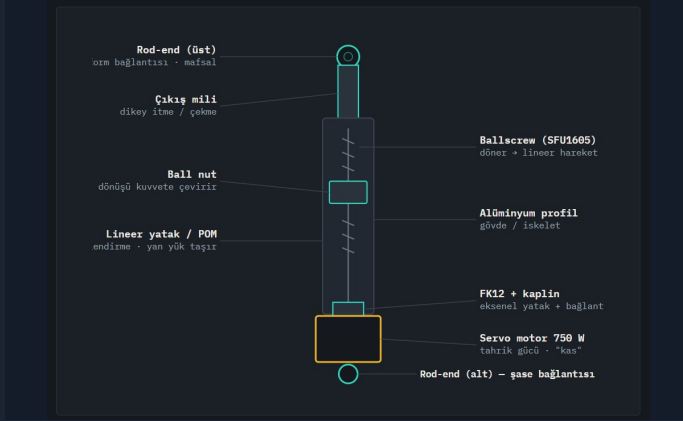
750 W AC servo · 50:1



Platform CAD tasarımı: koltuk, direksiyon, ekran ve şasi yerleşimi



Kaynaklı çelik şasi: imalat aşaması



Aktüatör kesiti: dönme hareketinin lineer itmeye çevrilmesi

## AKTÜATÖR TASARIMI

Her aktüatör, servo motorun dönme hareketini SFU1605 ballscrew ile lineer itme ve çekmeye çeviriyor. Yan yükleri lineer yatak ve POM burçlar taşıyor; motor FK12 eksenel yatak ve kaplin üzerinden bağlanıyor. Üst ve alt rod-end mafsalları platform ile şasi arasındaki açı değişimini serbest bırakıyor: aksi hâlde ballscrew eğilme yüküne maruz kalıp ömrünü hızla tüketirdi.

SFU1605

750 W AC servo

50:1 planet

AASD-15A

Thanos AMC

## MÜHENDİSLİK KARARI

3DOF ve 4DOF mimarilerini karşılaştırıp ağırlık merkezi farkındalıklı tork hesaplarını çıkardım; sürücü ağırlığı ve koltuk konumu tork ihtiyacını doğrudan belirlediği için bu hesap konfigürasyon seçiminin tamamını yönlendirdi.

## GEÇİKME PROBLEMİ

Hareket platformlarında algılanan kalite doğrudan **gecikmeye** bağlı. PC üzerinden geçen telemetri zincirini kaldırmak için ESP32 + UDP tabanlı bağımsız telemetri mimarisi geliştirdim; komut yolundan bir katman çıkararak gecikmeyi düşürdüm.

# Savunma Sanayii Deneyimi

## GİZLİLİK NOTU

Bu bölümdeki çalışmalar gizlilik yükümlülüğüne tabidir. Müşteri ve ürün adları, performans parametreleri, ekran görüntüleri ve test verileri bilinçli olarak yer almamaktadır; anlatım yalnızca çözülen mühendislik problemi ve kullanılan teknoloji düzeyindedir.

**Teknik detaylar, gizlilik yükümlülüklerinin izin verdiği ölçüde yüz yüze görüşmede aktarılabilir ve gösterilebilir.**  
Yazılı veya dijital paylaşım için ilgili kurumlardan izin alınması gerekir.

## GERÇEK ZAMANLI HEDEF TESPİT VE TAKİP SİSTEMİ

Digitest Savunma ve Havacılık · 2021–2024 · 6–10 kişilik yazılım ekibi

Savunma sanayii müşterisi için gerçek zamanlı görüntü işleme tabanlı hedef tespit ve takip yazılımı geliştirdim. Tespit tarafı C++, takip tarafı Python ile yazıldı. Gerçek zamanlı görüntü üzerine veriye dayalı sembol çizdirme (overlay) ve operatör arayüzünü Qt Creator/C++ ile geliştirdim.

Gerçek zamanlı işleme gereksinimini karşılamak için C++ multithreading, CUDA C ve OpenMP ile paralel işleme hatları kurdum. Tüm hat NVIDIA Jetson gömülü platformlar üzerinde çalışacak şekilde entegre edildi; veri alışverişi TCP, UDP ve UART protokolleri üzerinden yürütüldü.

C++

Python

CUDA C

OpenMP

Qt Creator

NVIDIA Jetson

TCP/UDP/UART

Multithreading

## KARMA GERÇEKLİK SİMÜLASYON VE GÖRSELLEŞTİRME

Wupsoft Savunma & Havacılık · 2024 – Devam

Unity C# ve Oculus ile savunma sanayiine yönelik VR/AR karma gerçeklik uygulamaları geliştiriyorum. Savunma sanayii fuarında sergilenmek üzere bir karma gerçeklik tanıtım uygulaması geliştirdim.

TÜBİTAK destekli bir dış balistik simülasyon projesinde MATLAB ile simülasyon modellemesi ve XR ortamında büyük veri görselleştirmesi geliştiriyorum. Ayrıca 6–20 geliştiriciye XR/VR geliştirme eğitimi verdim.

Unity

C#

MATLAB

Oculus

VR/AR

Simülasyon modelleme

Veri görselleştirme

## YÜKSEK LİSANS TEZİ

**Sanal gerçeklik ile biyosibernetik döngü kullanarak kullanıcı stres ölçümü ve adaptif uygulama:** Atılım Üniversitesi, Yazılım Mühendisliği (Tezli), tez dönemi. Kullanıcıdan alınan biyosinyallere göre sanal gerçeklik ortamının kendini uyarladığı kapalı döngü bir sistem. Simülasyon ve eğitim sistemlerinde doğrudan karşılığı olan bir konu.

# Görüntü İşleme ile XOX Oynayabilen Endüstriyel Robot

ROL

Yazılım birimi

KURUM

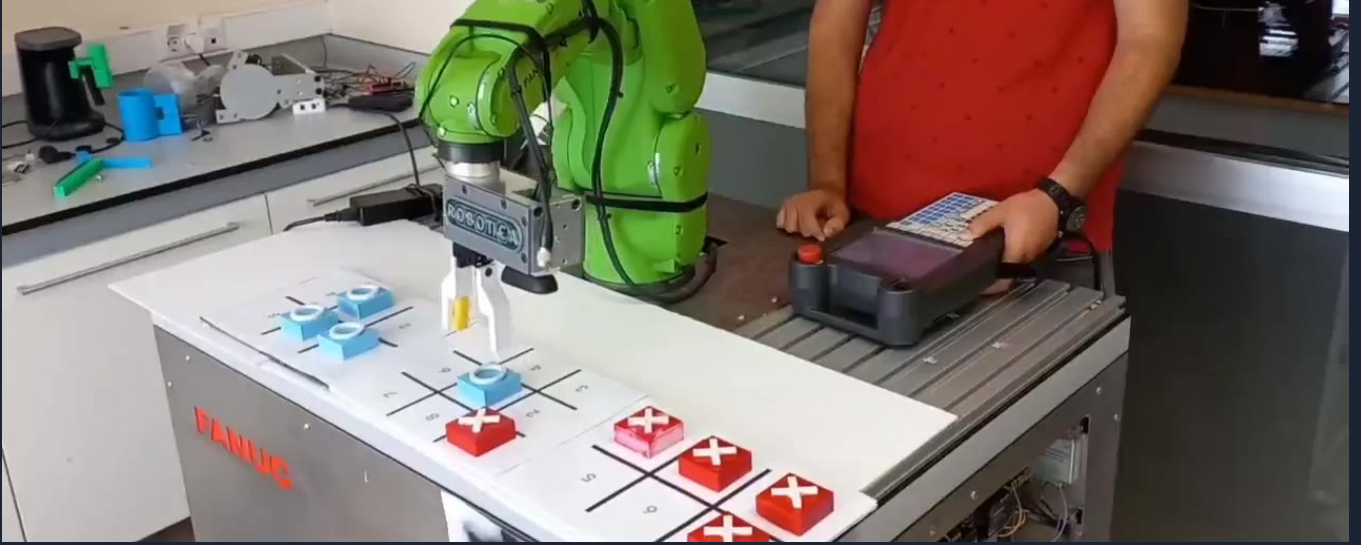
Lisans bitirme projesi

DÖNEM

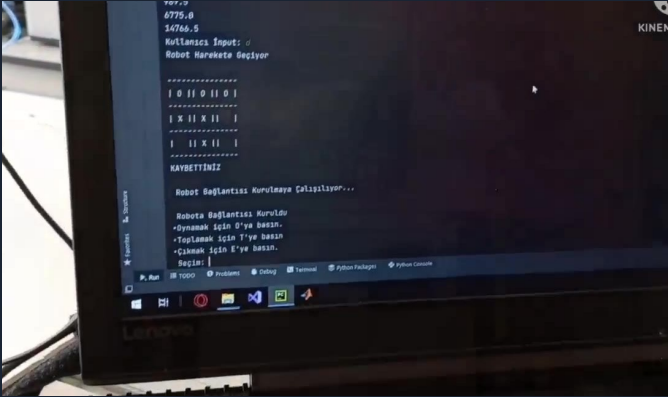
2021-2022

EKİP

4 kişi



Robot, görüntü işlemeyle okuduğu tahta durumuna göre hamlesini yapıyor



Görüntü işleme çıktısı: kameradan okunan tahta matrisi ve robot hamle kararı

## YAPTIĞIM İŞ

- Kamera görüntüsünden **tahta durumunu matrise çeviren** görüntü işleme hattı: Python
- **Minimax ağacı** tabanlı oyun çözme algoritmasının tasarımı ve kodlanması
- Operatör arayüzü: önce Kodular Creator ile prototip, sonra Qt Creator ile yeniden yazım
- Robottan gelen ham verinin kullanıcının anlayacağı dile çevrilmesi

Python

Görüntü işleme

Minimax

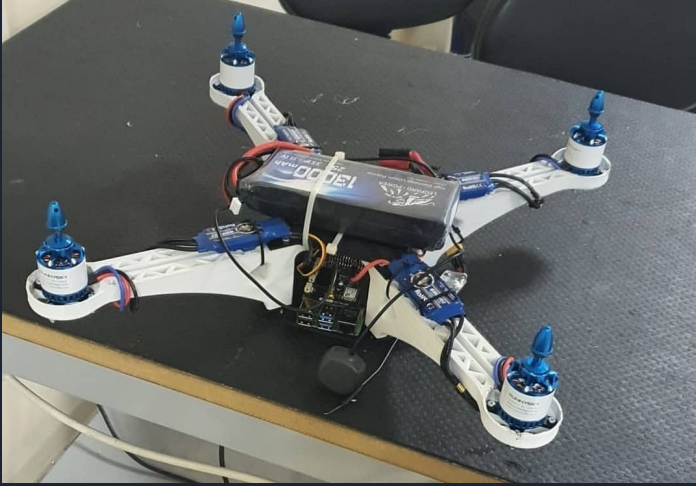
Qt Creator

Android

Endüstriyel robot

Bugün YOLO ile yaptığım işin ilk hâli bu projeydi: kameradan gelen görüntüyü makinenin karar verebileceği bir veri yapısına çevirmek. Derin öğrenme yerine klasik görüntü işleme kullanıldı, ama problem aynıydı: **görüntüyü aksiyona çevirmek**.

## Otonom Drone ve Elektrikli Araç



Aviyonik yerleşimi: Raspberry Pi + Navio 2, GPS anteni, ESC'ler ve 13.000 mAh LiPo

### OTONOM DÖNER KANAT DRONE

TÜBİTAK Uluslararası Drone Yarışması · Atılım Üniversitesi Robot Topluluğu · 2020-2021

Aviyonik sistemin tasarımını ve montajını yaptım; **Navio 2** uçuş kontrol kartını Raspberry Pi üzerine HAT olarak entegre ettim, telemetri ve GPS haberleşmesini kurup otonom uçuş yeteneğini sağladım.

Motor seçimi, itki ve uçuş süresi hesaplarını yaparak ESC ve batarya seçimini yaptım; görev mekanizmasını tasarladım ve görev için Python ile görüntü işleme geliştirdim.

Navio 2

Raspberry Pi

Python

Telemetri

GPS

İtki hesabı

### ELEKTRİKLİ ARAÇ TELEMETRİ VE AKS/ECU

TÜBİTAK Efficiency Challenge · Kocaeli 2021 · İlk 10

Araç Kontrol Sistemi (AKS/ECU) ve telemetri yazılımını Python ile geliştirdim. Batarya yönetim sisteminden gelen veriyi **LoRa** ile kablosuz aktarıp Raspberry Pi üzerinden **Nextion** dokunmatik ekrana işledim.

Yazılımın yanında aracın elektrik tesisatında da görev aldım: motor sürücü, elektronik bileşenler ve yerleşik şarj birimi. Bu proje, yazılımın altındaki donanımı okuma alışkanlığımın başladığı yer.

Python

LoRa

Raspberry Pi

Nextion

AKS / ECU

Güç elektroniği



Efficiency Challenge 2021, Kocaeli: takımın yarış aracı

## Teşekkürler

### UZMANLIK ÖZETİ

Gerçek zamanlı görüntü işleme ve bilgisayarlı görü sistemleri; bu sistemlerin gömülü platformlarda çalışır hâle getirilmesi; otonom hava aracı yazılımı; veri seti mühendisliği ve model eğitimi. Mekatronik temeli sayesinde yazılımın altındaki donanımı da okuyabiliyorum: sensör seçiminden güç bütçesine, tork hesabından haberleşme protokolüne kadar.

### ŞU ANDA

Wupsoft Savunma & Havacılık'ta kıdemli yazılım mühendisi olarak XR/VR ve simülasyon projelerinde çalışıyor, Atılım UAV'de yazılım takım liderliği yürütüyor, Atılım Üniversitesi'nde yazılım mühendisliği yüksek lisans tezimi yazıyorum.



### İLETİŞİM

[serdar.anil33@gmail.com](mailto:serdar.anil33@gmail.com)

[linkedin.com/in/sanildemirbas](https://www.linkedin.com/in/sanildemirbas)

Ankara, Türkiye

### BU DOKÜMANDA

9 proje · 6 bölüm

Savunma çalışmaları sanitize edilmiştir

Sürüm: Ağustos 2026

Gizlilik kapsamındaki çalışmalar, yükümlülüklerin izin verdiği ölçüde yüz yüze görüşmede gösterilebilir.