



# **TOBB UNIVERSITY OF ECONOMICS AND TECHNOLOGY**

## **ELE 495 – BİTİRME PROJESİ FİNAL RAPORU (2023 – 2024 YAZ DÖNEMİ)**

<b>OTONOM PARK EDEN ROBOT - Grup 6</b>	
<b>PROJE ÜYELERİ</b>	<b>NUMARA</b>
<b>BİLAL BERKEHAN KÖSE</b>	<b>201201025</b>
<b>UMUT YUMLU</b>	<b>211201050</b>
<b>CAN GEREK</b>	<b>201201024</b>
<b>ÇAĞKAN İNCE</b>	<b>221206082</b>
<b>GÖNDERİM TARİHİ</b>	<b>15.07.2024</b>
<b>MENTÖR</b>	<b>MURAT SEVER</b>

## İçindekiler

1. Özet .....	3
2. Giriş.....	3
3. Proje Aktiviteleri ve Teknik Süreç .....	3
3.1 Jetbot ve Jetson Nanonun Kurulumu .....	3
3.2 Sistem Blok Diyagramı ve Yaşam Döngüsü .....	3
3.3 Sistem Özellikleri.....	4
3.4 Yaşanan Olumsuzluklar .....	4
4. Teknik Dokümantasyon .....	5
5. Bütçe .....	6
6. Proje Planlaması (WBS, Gantt Chart) .....	7
7. Proje Gereksinimleri Karşılama Listesi.....	8
8. Referanslar .....	8
9. Ekler .....	8

## 1. Özet

Bu projede, Jetbot aracının mobil uygulama üzerinden bildirilen park alanına kırmızı çizgilere dokunmadan ve 3 dakika süre sınırına uyararak otonom bir şekilde park etmesini sağlayan bir sistem geliştirilmiştir. Bu amacı gerçekleştirmek için, Jetson Nano üzerinde çalışan görüntü işleme teknikleri ve kendi topladığımız verilerle eğitilen CNN modeli kullanılarak plaka tespiti ve otonom park özellikleri oluşturulmuştur. Sisteme park emri vermek ve güncel durumu takip etmek amacıyla mobil uygulama geliştirilmiştir.

## 2. Giriş

Projede istenen sistemleri kurmak için sayı tanıma modeli eğitmek, görüntü işleme özellikleri kullanmak, otonom sistem algoritması oluşturmak ve mobil uygulama tasarlayarak bunun haberleşme isterlerini sağlamak gibi farklı disiplinlerde çalışmalar yapılmış olup bu çalışmalar ışığında gelecekte gerçek araçlara veya evleri temizleyen otonom robotlar gibi projelere başlangıç ve yol gösterici olan çıktılar elde edilmiştir.

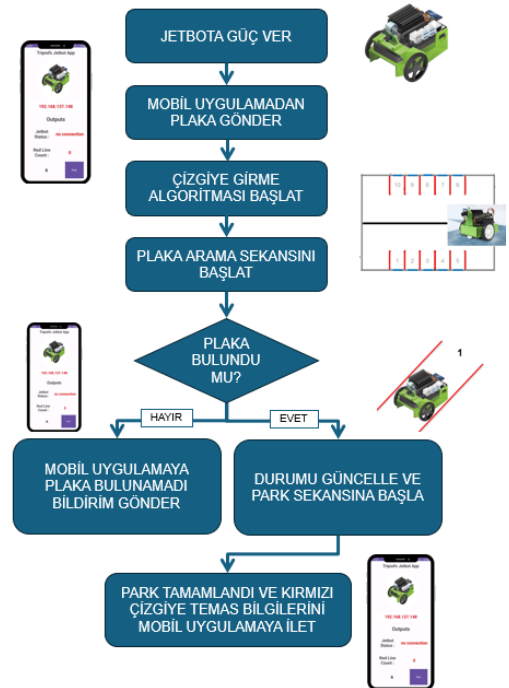
## 3. Proje Aktiviteleri ve Teknik Süreç

### 3.1 Jetbot ve Jetson Nanonun Kurulumu

Projeye başlarken öncelikle tarafımıza teslim edilen Jetson Nano kurulumu ve Jetbot montajı gerçekleştirilmiştir. Jetson Nanoda bir çalışma ortamı yaratmak için Docker imajı yüklenmiş ve bu imajın içindeki kütüphaneler kullanılmıştır. SSH protokolü kullanılmadan önce Putty uygulaması kullanılarak gerekli konfigürasyonlar yapılmış ve Wifi bağlantısı sağlanmıştır.

### 3.2 Sistem Blok Diyagramı ve Yaşam Döngüsü

Şekil 1'de gösterilen blok diyagramı algoritmanın işleyişini açıklamaktadır. Jetbot'a güç verildikten sonra mobil uygulama ve Jetbot bağlanacaktır. Bu işlemin ardından mobil uygulama üzerinden park edilmesi istenen plaka bilgisi verilecektir. Plaka bilgisini alan Jetbot siyah çizgiye girecek ve çizginin diğer ucuna gidip 180 derece dönecek ve ardından plaka arama sekansına başlayacaktır. Bu sekansta bir süre ilerleyip plakaların hizasına gelince Jetbot önce 90 sonra 180 derece dönecek ve bu plakaları okuyacaktır. Uyuşma olmaması durumunda sonraki plaka hizasına gelecek ve aynı işlemi bulana kadar tekrar edecektir. Plaka bulunması durumunda arama sekansından park etme sekansına geçilecek ve park işlemi gerçekleştirilip mobil uygulamaya park edildi ve kırmızı çizgiye basma sayısı bilgileri iletilecektir. Süreç sonunda plaka platform üzerinde bulunamazsa mobil uygulamaya mesaj iletilecektir. Bu iş için belirlenen süre 3 dakika olup algoritmamız bu süre içinde görevini tamamlayabilmektedir.



Şekil 1: Sistem Blok Diyagramı

### 3.3 Sistem Özellikleri

Proje temelde mobil uygulama, plaka tanıma ve park olmak üzere üç kısımdan oluşmaktadır. Mobil uygulamamızın front-end yazılımı Android Studio IDE'sinde XML programlama dili ile yazılmıştır ve dizayn yapılırken olabildiğince sade ve işlevsel olmasına dikkat edilmiştir. Java programlama dili ile back-end yazılımını geliştirdiğimiz mobil haberleşme sistemi GET ve POST metodlarını kullanmaktadır. POST metodu ile uygulama Jetbot'a plaka bilgisi gönderecektir. GET metodu ile de park yapılıyor, plaka aranıyor gibi durumlar devamlı olarak mobil uygulamaya bildirilecektir

Park edilecek plakayı mobil uygulamadan alan Jetbot siyah çizgiye Numpy özellikleri kullanılarak sokulmaktadır. Burada siyah çizgi bulunmakta, aynı kalınlıktaki çizgi ortalananarak takip edilmekte ve hata fonksiyonu aracılığı ile sağa veya sola kayılması durumunda çizgiye geri dönülmesi sağlanmaktadır. Siyah çizginin sonuna gidip 180 derece dönüş yapan Jetbot ilk plaka hizasına gelince park yeri arama sekansını başlatmaktadır. Plaka hizasında 90 ve 180 derecelik dönüşler yapılmakta, dönüşlerin ardından da minimal sağ sol hareketler yapılarak plakanın daha iyi aranması sağlanmaktadır.

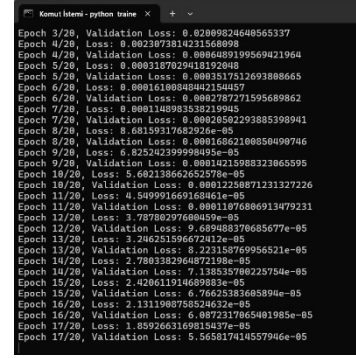
Görüntü işleme algoritması için Pytorch kütüphanesi özellikleri kullanılmış ve ardından görüntüler OpenCV kütüphanesi araçları ile binary'e dönüştürülerek daha keskin çalışan bir görüntü işleme kullanılmıştır. Burada threshold değeri sadece sayıyı görecektir şekilde seçilmiştir. CSI kamera tarafından algılanıp binary (ikilik taban) hale getirilen plakalar, Pytorch kütüphanesiyle önceden kendi örneklerimizi alarak eğittiğimiz modelle karşılaştırılarak istenen plaka tespit edilene kadar plaka araması sürdürülecektir. Aranan plaka tespit edilirse park yapma sekansı başlayacaktır. Park başarılı bir şekilde gerçekleştikten sonra mobil uygulamaya başarılı park yapıldığına dair bildirim gidecektir.

### 3.4 Yaşanan Olumsuzluklar

Proje esnasında karşılaşılan en büyük sorun Jetson Nano'nun 2 GB RAM ihtiva eden versiyonunun kullanılması olmuştur. Yapılan proje haberleşme, görüntü işleme, park algoritması gibi pek çok farklı görevi aynı ayna yaptığından bu kısıt geliştirme aşamasında da büyük zaman kayıplarına sebep olmuştur. Sadece işletim sistemi çalıştırması esnasında %30 RAM kullanan bir cihazla geliştirme yapmak projenin başlıca dezavantajıdır. Bundan etkilenmemek amacıyla siyah şerit takip konusunda optimizasyonlar ve yeni ayarlamalar yapılması gerekmiştir. Kullandığımız Docker Container'da da kart uyum sorunu yaşanmıştır. Kullanılan görüntü işleme kütüphaneleri ile Docker Image içindeki sürümlerin uyumsuzluğu sorunlar oluşturmuştur. Kullanılması planlanan daha gelişmiş görüntü işleme, model eğitme kütüphaneleri yaşanan uyumsuzluk sorunları sebebiyle kurulamamış ve bu denemeler büyük zaman kayıplarına sebep olmuştur.

#### 4. Teknik Dokümantasyon

Modelin eğitilmesi esnasında platform üzerinde kendi topladığımız veriler kullanılmıştır. Verilerin toplanması farklı uzaklık ve açılarda her plaka için 400 civarı fotoğraf çekilmesi ile yapılmıştır. Tarafımızca çekilen fotoğraflar Pytorch kütüphanesi ve CNN modeli kullanılarak 20 epoch kadar eğitime sokulmuştur. Eğitim sonunda en verimli epoch (Loss en düşük olan epoch) model olarak kaydedilmiştir. Plaka okuma için kullanılacak model bu adımların izlenmesiyle oluşturulmuştur.

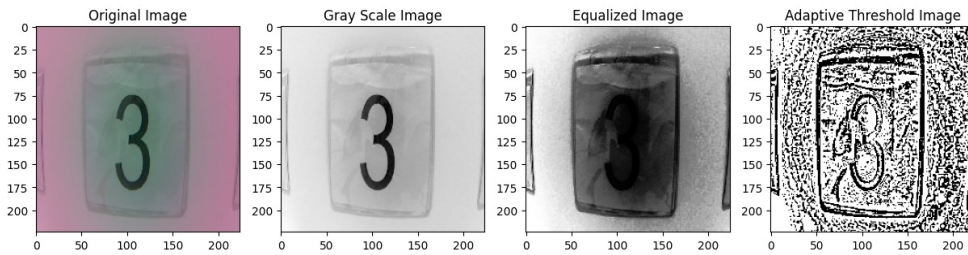


Epoch	Validation Loss	Loss
Epoch 3/20	0.82089824648565337	
Epoch 4/20	0.8023873814231568098	
Epoch 4/20	Validation Loss: 0.8086488199569421964	
Epoch 5/20	0.80319702041312048	
Epoch 5/20	Validation Loss: 0.8083517512693888665	
Epoch 6/20	0.8081610884842154457	
Epoch 6/20	Validation Loss: 0.80827827271595689862	
Epoch 7/20	0.808114898353219945	
Epoch 7/20	Validation Loss: 0.80828582293885398941	
Epoch 8/20	0.801393176829226e-05	
Epoch 8/20	Validation Loss: 0.80816862108850490746	
Epoch 9/20	0.825242399998495e-05	
Epoch 9/20	Validation Loss: 0.8084215883323865595	
Epoch 10/20	5.692138662652578e-05	
Epoch 10/20	Validation Loss: 0.80811976886913479231	
Epoch 11/20	4.509991669163843e-05	
Epoch 11/20	Validation Loss: 0.8081258871231327226	
Epoch 12/20	3.78788297688459e-05	
Epoch 12/20	Validation Loss: 9.659488370685677e-05	
Epoch 13/20	3.240751596672412e-05	
Epoch 13/20	Validation Loss: 8.223158769956521e-05	
Epoch 14/20	2.7883382964872198e-05	
Epoch 14/20	Validation Loss: 7.138535788225754e-05	
Epoch 15/20	2.428611914689883e-05	
Epoch 15/20	Validation Loss: 6.7662538488894e-05	
Epoch 16/20	2.131198783521632e-05	
Epoch 16/20	Validation Loss: 6.8872317865481985e-05	
Epoch 17/20	1.8592683169813437e-05	
Epoch 17/20	Validation Loss: 5.565817414557946e-05	

Şekil 2: Model Eğitimi Görseli

Jetbot kamerasından alınan görüntü frameleri siyah şerit takip sekansında daha kullanışlı olması için framein alt kısmının %60'lık, sağ ve sol kısmından %35'lik kısmı kırılmıştır. Plakaları okurken etraftaki diğer plakaların modele girdi olarak verilmemesi için Jetbot kamerasına 3x zoom uygulanmıştır. Böylece sadece karşıdaki plaka modele sokulmuştur.

Modelin daha iyi eğitilmesi için Jetbot kamerasından gelen frameler ortam etkilerinden arındırmak amacıyla RGB'den gray image'a, gray image'dan ise son olarak binary hale 80 threshold değeri ile getirilmiştir. Bu sayede platform üzerinde sadece rakamlar modele girdi olarak sunulmuş ve threshold değeri sayesinde plaka çerçeveleri ve yansımalar gibi istenmeyen etkenler önlenmiştir.



Şekil 3: RGB-Gray-Binary Dönüşüm Görseli

Siyah şerit takip fonksiyonu Numpy ve OpenCV kütüphaneleri vasıtasıyla PID kontrolcü kullanılarak anlık olarak kontrol edilmektedir. Jetbot kamerasından alınan framelerin Jetbot doğrultusuna olan hata payı ile Jetbot siyah çizgi güzergahında gitmektedir. Jetbot plaka tanıma için fotoğraf yakalama modunda frame açısını artırmak amacıyla küçük açılarda dönmekte ve çeşitli açılarda frame yakalamaktadır. Daha sonra işleme yaparak mobil uygulamadan gelen istenen park numarası ile kıyaslamaktadır.



Şekil 4: Siyah Çizgi Takip Görseli

**5. Bütçe**

SATIN ALINAN ÜRÜN	ADET	MALİYET
128 GB Micro-SD Kart	1	900₺
18650 Batarya	3	579₺
Wifi Adaptör	1	279₺
<b>TOPLAM HARACAMA</b>		<b>1758₺</b>

Tablo 1: Maliyet Tablosu



**18650 3.7 V 2600mAh Li-ion Şarjlı PİL - Kutup Başlı**  
Marka : ProFuse  
Ürün Kodu : 18712  
Fiyat : 150,85 TL + KDV  
KDV Dahil Fiyat : **193,2 TL**

Şekil 5: Batarya Faturası

**#408-1452489-1924360 Numaralı Siparişin Son Ayrıntıları**[Bu sayfayı yazdırın ve saklayın.](#)

Siparişin Verildiği Tarih: 7 Haziran 2024

408-1452489-1924360

Sipariş Toplamı: **200,00 TL****7 Haziran 2024 tarihinde gönderildi****Sipariş Edilen Ürünler**

1 : Tp-Link TL-WN725N, 150Mbps Kablosuz N Nano Usb Adaptör

Satıcı: Amazon Turkey Perakende Hizmetleri Limited Şirketi

Durum: Yeni

**Teslimat Adresi:**

Çağkan İNCE  
Söğütözü mah. Söğütözü cad. No:16 TOBB Üniversitesi Ana bina  
TOBB Üniversitesi Ana bina Kargo odası  
Ankara, Çankaya, Söğütözü Mh. 06510  
Türkiye

**Teslimat Tercihi**

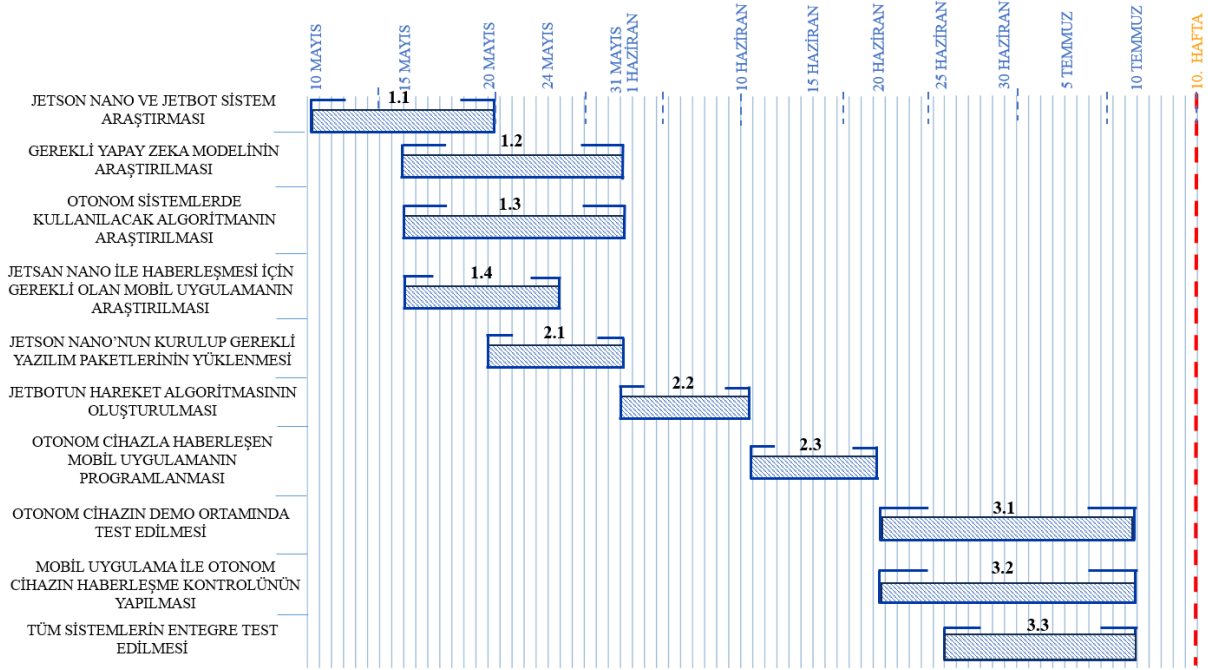
Ekspres Teslimat

**Fiyat**

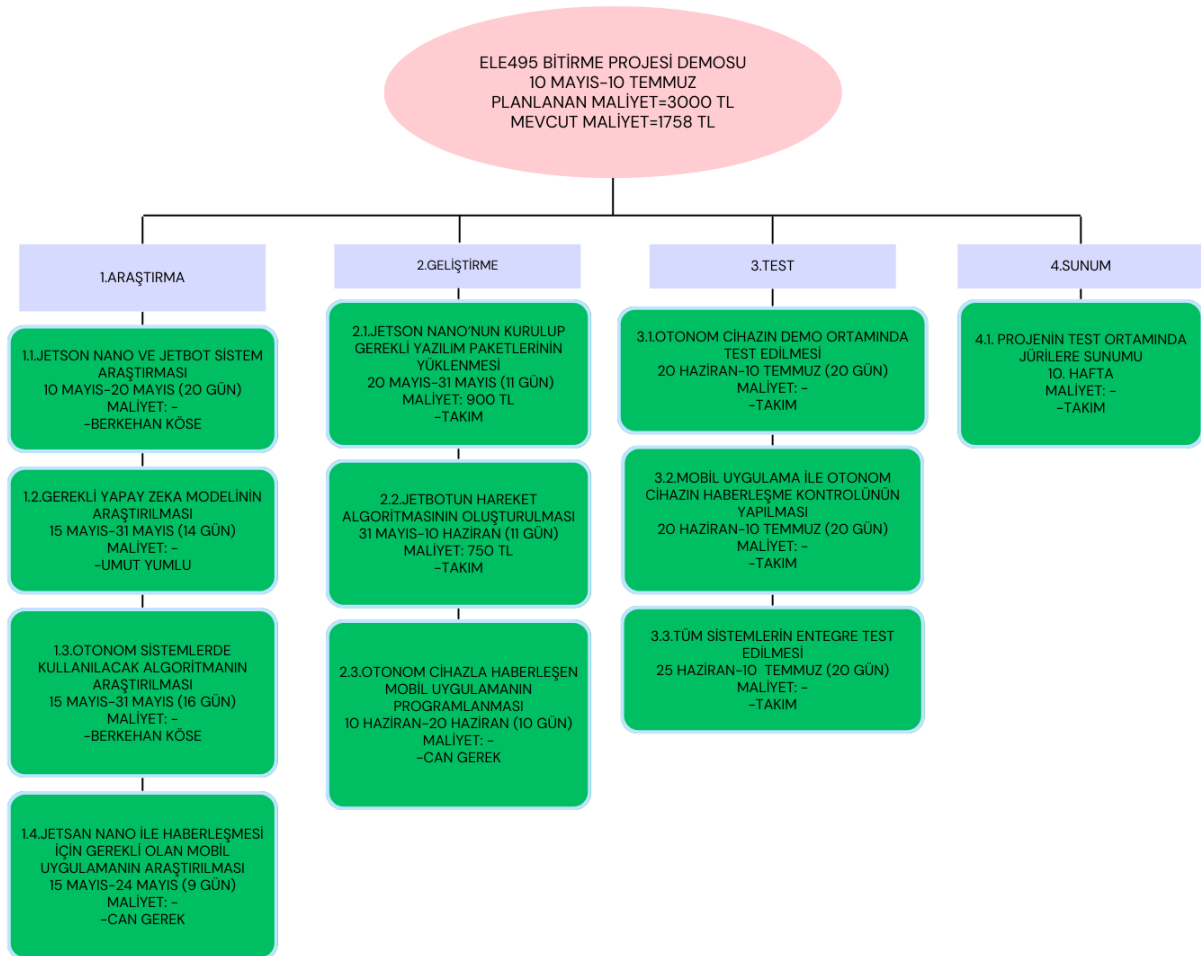
279,00 TL

Şekil 6: Wifi Adapter Faturası

## 6. Proje Planlaması (WBS, Gantt Chart)



Şekil 7: Proje WBS Şeması



Şekil 8: Proje Gantt Şeması

## 7. Proje Gereksinimleri Karşılama Listesi

Proje İsterleri	İster Durumu
Aracın kırmızı çizgilere değmeden tahsis edilen park yerine park etmesi	Kısmen Başarıldı
Plaka numarasının demo sırasında değiştirilmesi sonrasında kırmızı çizgilere değmeden 3 dakika içerisinde park etmesi	Başarıldı
Park Etme İşlemi Bitti” bilgisinin akıllı telefona iletilmesi	Başarıldı
Platformda plaka olmaması durumunda durumun akıllı telefona iletilmesi.	Başarıldı
Kırmızı çizgi ihlalinin akıllı telefona iletilmesi	Başarıldı
Github hesabı açılarak kodların paylaşımı	Başarıldı
Maliyet kısıtını yerine getirme	Başarıldı

## 8. Referanslar

- [1] <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-2gb-devkit>
- [2] <https://developer.nvidia.com/embedded/learn/jetson-nano-2gb-devkit-user-guide>
- [3] <https://jetbot.org/master/index.html>
- [4] [https://www.waveshare.com/wiki/JetBot\\_2GB\\_AI\\_Kit](https://www.waveshare.com/wiki/JetBot_2GB_AI_Kit)

## 9. Ekler

- [1] Proje kodları verilen Github linkinde paylaşılmıştır.

<https://github.com/ELE495-2324Summer/capstoneproject-tripod-grup-6/tree/main>