



Arda YAKAKAYI (19253519)
Ahmet BÜREOĞLU (19253704)
Bilgisayar Mühendisliği
24.05.2022

Görüntü İşleme Methodları ile Otonom Takip Sistemi

İÇİNDEKİLER



❖Sunum Videosu	3
❖Giriş	4
❖Projenin Amacı ve Kapsamı	5
❖Çalışmaya Giriş	8
❖İlk Adım “Görüntüyü Tanıma”	10
❖Renk Tonu ve Doygunluk	11
❖Contour	13
❖Takibi Sağlama	15
❖Sağ ve Sol Manevralar	16
❖Proje Süreci	18
❖Cihazın Kamerası Karşıya Bakıyordu	20
❖Ayna Kullanılması Cihazın Ağırlık Merkezinin Değişmesine Yol Açtı	21
❖Sonuç	25
❖Çalışmada Kullanılan Python Kodları	26

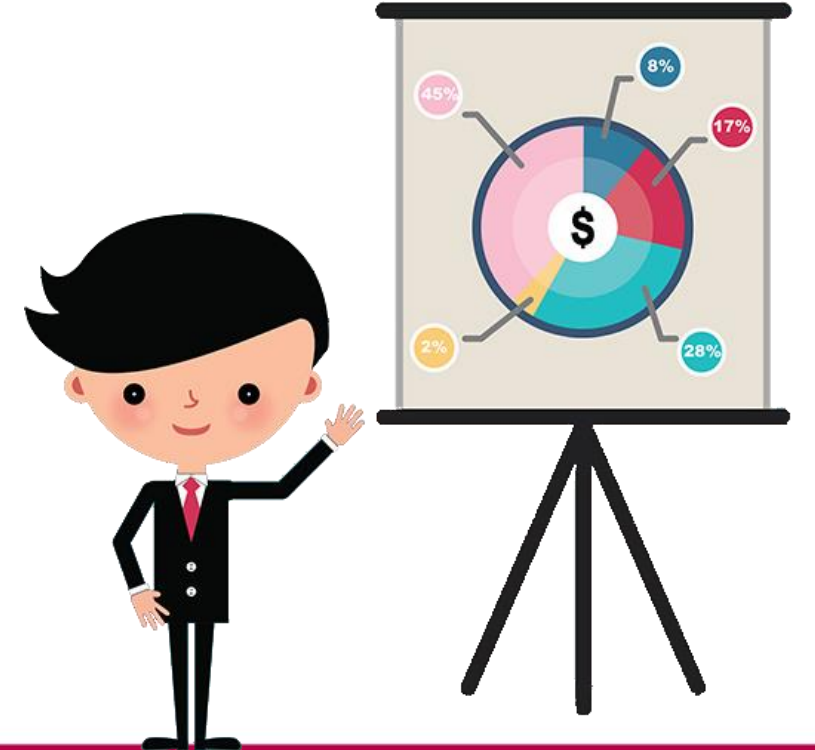
SUNUM VIDEOSU



Sunum Videosu için tıklayınız.

GİRİŞ

- ◆ Projenin amacı ve kapsamından bahsedeceğiz.
- ◆ Kullanılan framework ve kütüphanelerden bahsedeceğiz.
- ◆ Yapım aşamalarından bahsedeceğiz.
- ◆ Eksikliklerden ve zorluk çıkaran kısımlardan bahsedeceğiz.
- ◆ Nasıl geliştirilebileceğinden bahsedeceğiz.



PROJENİN AMACI VE KAPSAMI



PROJENİN AMACI VE KAPSAMI

- ◆ Gelişen teknoloji ile birlikte uçan araba konseptlerinin benimsenmeye başlaması.
- ◆ Şehirdışında olduğu gibi şehiriçinde de kullanımının önemli ölçüde artacak olması.
- ◆ Otonom sürüş tekniklerinin yaygınlaşması.
- ◆ Uçan otomobillerde bir takım biyolojik ve fizyolojik kısıtlar

PROJENİN AMACI VE KAPSAMI



- ◆ Yerkürenin şeklinden kaynaklı olarak yükseklik arttıkça yer değiştirme artacaktır.
- ◆ Bu durum şehir içi ulaşımında gidilmesi planlanan noktaya ulaşım konusunda sorun teşkil edecektir.
- ◆ Şehir içi kullanımda, standart otomobillerin aksine uçanların kontrolü zor olacağından kendilerine ayrılan rotalarda hareket etmeleri daha sağlıklı olacaktır.
- ◆ Bu projeye uçan otomobiller ve askeri, sivil ya da farklı kullanım alanlarında karşımıza çıkan drone'lar için düşünülen takip sisteminin gerçekleştirilmesi hedeflenmektedir.

ÇALIŞMAYA GİRİŞ



ÇALIŞMAYA GİRİŞ

- ◆ Cihazımızın istenilen veya belirtilen bir yolda ilerlemesi için kendisinde mevcut olan kamerayı kullanarak görüntü işleme yöntemleri kullanacağız.
- ◆ Python dili
- ◆ djitello, opencv, numpy başlıca kütüphaneler.



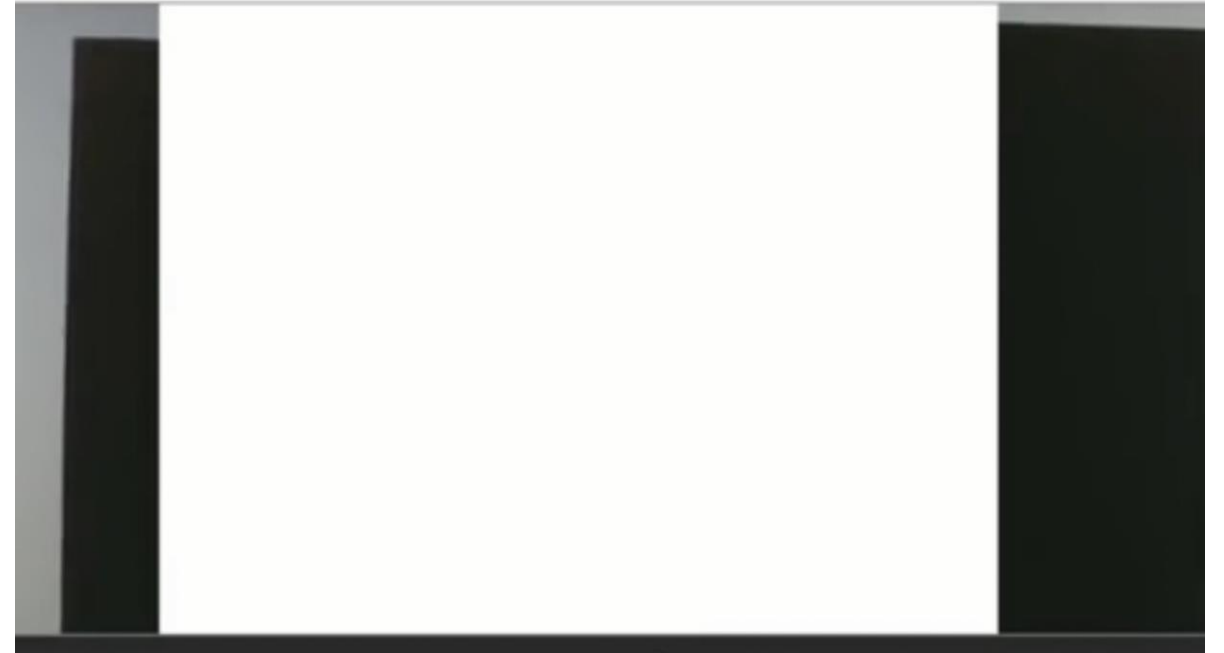
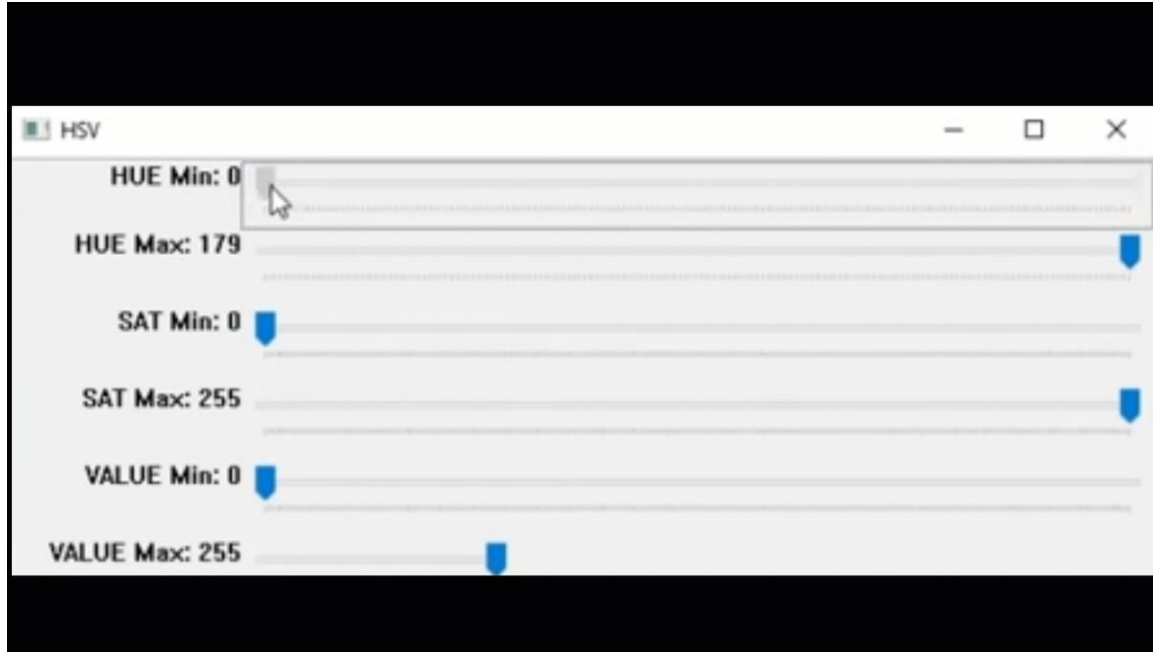
İLK ADIM: “GÖRÜNTÜYÜ TANIMA”

- ◆ djitellopy kütüphanesi ile drone ile bağlantı kurulur.
- ◆ cv2 kütüphanesine ait “streamon” methodu ile kamera kaydı True hale getirilerek aktive edilir.
- ◆ Görüntünün yatay ve dikey eksen boyutları belirlenir.

RENK TONU VE DOYGUNLUK

- ◆ Renklerle çalışmak yerine Hue ve Saturation ile çalışmak daha iyi olacaktır.
- ◆ Doygunluk rengin ışığı yeterince yansıtması ve parlaklığıyla ilgilidir.
- ◆ O rengin, aynı ışık ortamında renksiz bir griye oranıyla hesaplanır.
- ◆ Doygunluk 0'a yaklaştıkça renk siyaha yaklaşır.
- ◆ Görüntüye dönüşüm uygulanır.
- ◆ `hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)`

RENK TONU VE DOYGUNLUK



CONTOUR

- ◆ Arkaplanın da olaya dahil olması sorun çıkarmakta.
- ◆ Doğrudan bağlı olmayan benzer renk değerlerine sahip pikseller çıkarılır ve görüntüdeki alan sadeleştirilmiş olur.



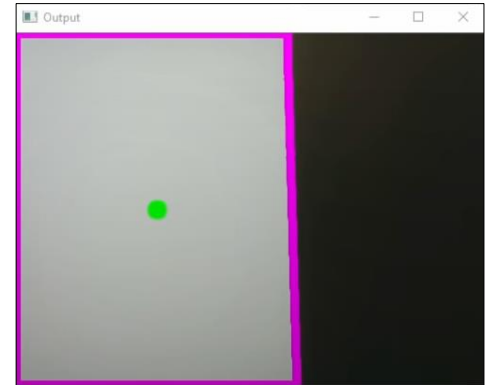
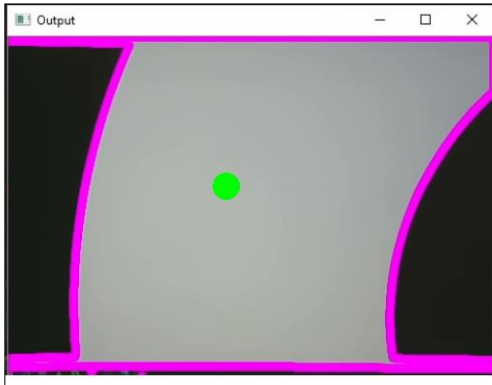
CONTOUR



Contour eklendikten sonra bahsi geçen yolun kenarında sınırlar oluşur. Bu sınırları belirlemek yine ton ve doygunluk ile orantılıdır.

TAKIBİ SAĞLAMA

- ◆ Temelinde, alınan görüntüyü taramayı ve belli bir oranın üstündeki piksel miktarının beyaz ağırlıklı olmasını baz alan bir sistem.
- ◆ Bunun için öncelikle merkezi bir hareketi sağlamayı amaçlayan bir merkez noktası oluşturulur.
- ◆ Oluşturulan nokta, öncesinde oluşturulan sınırların içerisindeyse hareket sağlanır. Sınırların içerisinde yer alabilmesi için de görüntüde belli bir oranın üstünde beyaz ağırlıklı piksel olması gerekmektedir.

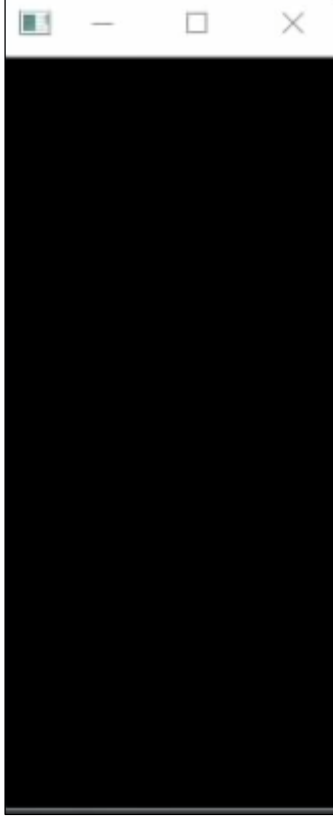


SAĞ VE SOL MANEVRALAR

- ◆ Görüntü boyutlarının üçün katı olması burada önemli.
- ◆ Görüntü çerçevesi 3 ayrı çıktı penceresinde incelenirse sağ ve sol manevraların ayarlanması daha kolay olacaktır.
- ◆ Üç dikey parçaya bölünmesi durumunda her çerçeve ya da görüntüde beyaz tonlarındaki pikseller ayrı hesaplanır ve belli bir oranın üstünde olması durumunda 1 değeri, altında olması durumunda 0 değeri döndürülür.



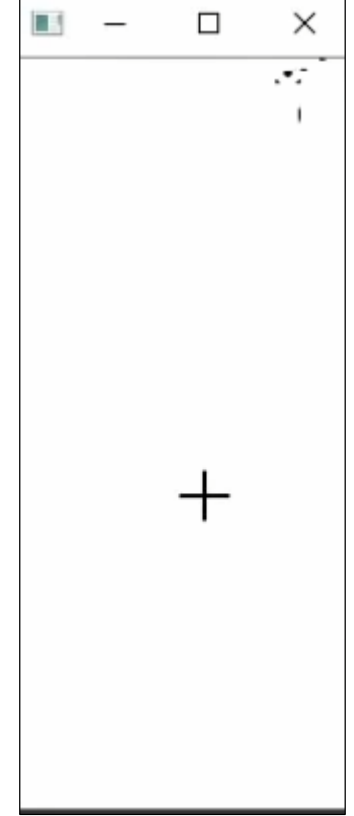
SAĞ VE SOL MANEVRALAR



0



1



1

PROJE SÜRECİ

PROJE SÜRECİ

- ◆ Proje, Görüntü işleme yöntemleri kullanıldığından teorik anlamda çok fazla zorluk yaşatmamıştır.
- ◆ Uygulama kısmına geçildiğinde bir kaç teknik sıkıntı mevcuttu.
 - Kullanılacak olan cihazın kamerası karşıya bakıyordu.
 - Çalışma ortamında beyaz tonları çok fazlaydı



CIHAZIN KAMERASI KARŞIYA BAKIYORDU

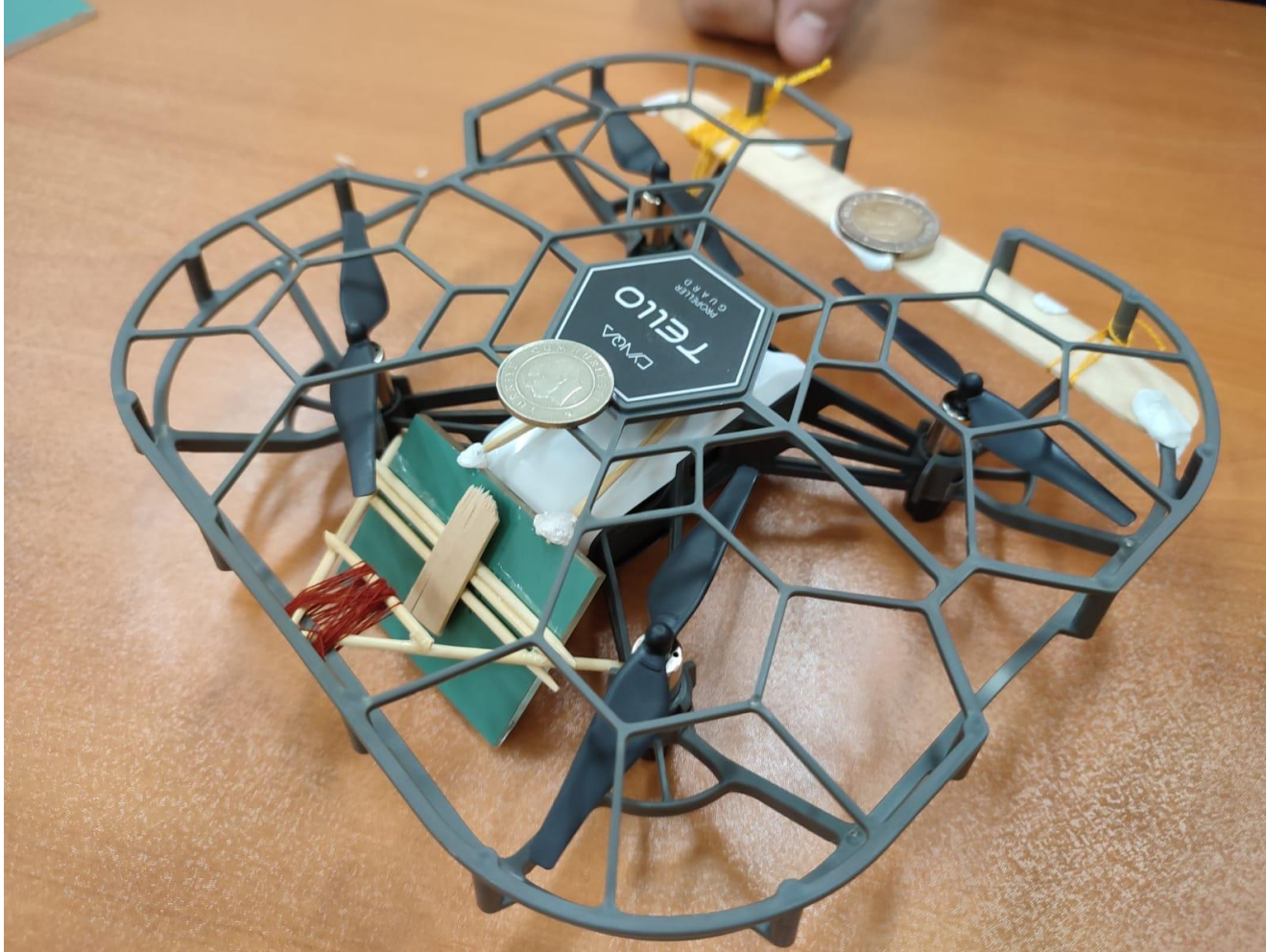
- ◆ Beyaz tonlamalı bir ortamda drone'un karşıya bakması, görüntüde çok fazla beyaz olması ve cihazın yoldan sapması anlamına geliyor.
- ◆ Kamera sabit ve sistemin içine gömülü olduğu için bir şekilde aşağıyı görmesinin sağlanması gerekiyordu.
- ◆ Bunun için öncelikle ayna kullanılmasına karar verildi. Malzemeler temin edildikten sonra ayna 45 derecelik bir açıyla kamera önüne yerleştirildi.
- ◆ Ancak bu yöntem, farklı bir soruna yol açtı.



AYNA KULLANILMASI CİHAZIN AĞIRLIK MERKEZİNİN DEĞİŞMESİNE YOL AÇTI.

- ◆ Cihaza takılan ayna için her ne kadar küçük bir parça düşünülse ve montaj işlemi için hafif ahşap malzemeler kullanılsa da bu durum cihazın ağırlık merkezinin değişmesinin önüne geçemedi.
- ◆ Cihaz, test uçuşları sırasında öne doğru savrulmaya başladı.
- ◆ Bunun üzerine, ağırlığını dengelemesi için farklı yöntemler denendi. İlk olarak, ağırlık, aynanın tam karşısına, drone'un da arkasına verilmeye çalışıldı.

AYNA KULLANILMASI Cihazın Ağırlık Merkezinin Değişmesine Yol Açtı.



AYNA KULLANILMASI CİHAZIN AĞIRLIK MERKEZİNİN DEĞİŞMESİNE YOL AÇTI.

- ◆ Ancak bu durum, beklenilenden farklı bir sonuç verdi.
- ◆ Hem cihazın kaldıramayacağı bir yük verilmişti. Hem de kalibrasyonu hassas bir cihaz için en ufak bir ağırlık farkı kalibrasyonu etkileyebiliyordu. Sonuç olarak cihazda sağlıklı bir uçuş gözlenemedi.



BİR BAŞKA UÇUŞ DAHA

- ◆ Tekrar ve tekrar yapılan denemelerin ardından sonuç hep aynıydı. Ufak ağırlık değişimleri ile tekrardan test uçuşları yapıldı.



SONUÇ

- ◆ Ağırlık dengelemeleri ile birlikte aynanın kaldırılması sonucunda istenilen sonuca ulaşıldı.



ÇALIŞMADA KULLANILAN PYTHON KODLARI



*LineFollower.py için
tıklayınız.*



*ColorPicker.py için
tıklayınız.*

KATILDIĞINIZ İÇİN TEŞEKKÜRLER