DÖRT ROTORLU HAVA ARACI İLE GÖRME TABANLI NESNE TAKİBİ

Abdulaziz CEYLAN

Yüksek Lisans Tezi

Teknoloji Fakültesi Mekatronik Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Beşir DANDIL

TEMMUZ-2018

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DÖRT ROTORLU HAVA ARACI İLE GÖRME TABANLI NESNE TAKİBİ

YÜKSEK LİSANS TEZİ

Abdulaziz CEYLAN

(122134109)

Tezin Enstitüye Verildiği Tarih : 6 Temmuz 2018
Tezin Savunulduğu Tarih : 2 Ağustos 2018

Tez Danışmanı: Prof. Dr. Beşir DANDIL (F.Ü.)
Diğer Jüri Üyeleri: Dr. Öğr. Üyesi Cafer BAL (F.Ü.)
Dr. Öğr. Üyesi Mehmet ÜSTÜNDAĞ (B.Ü.)

TEMMUZ-2018

ÖNSÖZ

Lisansüstü eğitimimin ilk gününden itibaren bilgi ve tecrübesiyle beni her zaman yönlendiren ve destekleyen ayrıca bu çalışmanın planlanması ve gerçekleştirilmesinde ilk andan itibaren her türlü yardımlarını esirgemeyen çok değerli ve kıymetli danışman hocam Prof. Dr. Beşir DANDIL'a saygı ve şükranlarımı sunarım.

Bilgi ve desteklerini esirgemeyen çok değerli arkadaşlarım Öğr. Gör. Hüseyin KUTLU'ya, Öğr. Gör. Ersan YAZAN'a, Serhat OKUR'a, M.Ali SICAK'a, üzerimde çok emeği olan anne ve babama, bu topraklar için canını feda eden tüm şehitlerimize ve yaralanan gazilerimize, desteklerini esirgemeyen bölüm hocalarıma teşekkürü bir borç bilirim.

Yüksek Lisans eğitimim süresince bana destek veren, hep yanımda olan ve cesaretlendiren sevgili eşim Tuğba CEYLAN ve kızlarım Rümeysa ile Rüveyda'ya yürekten teşekkür ederim.

Abdulaziz CEYLAN
ELAZIĞ-2018

İÇİNDEKİLER

Sayfa No

ÖZET

Hem askeri hem de sivil sektörlerde insansız hava araçlarına olan ihtiyaç, son yıllarda önemli ölçüde artmıştır. Günümüzde insansız hava araçları arama kurtarma, hasar tespiti, keşif gibi alanların dışında istihbarat ve gözetleme gibi askeri alanlarda da çok önemli bir rol oynayarak, tehlikeli ortamlarda yapılacak görevlerde kullanılmaktadır. Çok çeşitli modelleri bulunan insansız hava araçlarından olan dört rotorlu insansız hava aracı, yüksek hareket kabiliyeti ile ön plana çıkmaktadır.

Tezde dört rotorlu insansız hava aracının görüntü tabanlı nesne takibi gerçekleştirilmiştir. Bunun için ilk önce dört rotorlu insansız hava aracının matematiksel modeli oluşturulmuştur. Nesne takip aşaması için Parrot firmasının Parrot AR.Drone 2.0 Power Edition ürünü kullanılmıştır. Nesne takip işleminde, cihazın ön kamerasından alınan görüntüler kullanılarak takip edilecek nesnenin renk, şekil ve koordinat bilgileri kullanılmıştır. Python programlama dilinde AT (Attention) komutları kullanılarak yazılan kod parçacıkları ile insansız hava aracının yapması istenilen hareketler gerçekleştirilmiştir. Görüntü işlemede önemli bir platform olan OpenCV'den faydalanılarak, hareketli nesnenin tespitinde renk özelliği ve şekil özelliği ayrı ayrı incelenerek işlem yapılmıştır. Sabit kameradan elde edilen görüntülerden hareketli nesnenin tespiti için renk ve şekil özellikleri birlikte kullanılarak Python programlama dilinde, nokta ve çekirdek tabanlı nesne takibi gerçekleştiren program oluşturulmuştur. Oluşturulan program ile dört rotorlu insansız hava aracına wifi üzerinden bağlantı kurularak, cihazın ön kamerasından elde edilen görüntüler kullanılmış ve görme tabanlı nesne takip işlemi gerçekleştirilmiştir. Autoflight programı üzerinden AR.Drone 2.0 ile bağlantı sağlanarak cihazın hareket halinde iken izlemiş olduğu rota çizgisi incelenmiştir. EZ-Builder programında AR.Drone 2.0 üzerindeki kameradan alınan görüntülerden RGB (Red-Green-Blue) renk uzayında kırmızı renkli hareketli bir nesnenin, HLS (Hue-Luminance-Saturation) renk uzayında sarı renkli hareketli bir nesnenin ve YCbCr (Luminance-Chrominance blue- Chrominance red) renk uzayında ise sarı ve yeşil renkli dikdörtgen nesnelerin sisteme tanıtılma ve takip işlemleri gerçekleştirilmiştir.

Anahtar Kelimeler: Dört Rotorlu İnsansız Hava Aracı, Görüntü İşleme, Nesne Takibi,

SUMMARY

Visual-Based Object Tracking With Four-Rotor Aerial Vehicle

The need for unmanned aerial vehicles in both the military and civil sectors has increased significantly in recent years. Today, unmanned aerial vehicles are used in missions, such as search and rescue, damage detection, and reconnaissance, as well as in military areas such as intelligence and surveillance. The four-rotor unmanned aerial vehicle of unmanned aerial vehicles with a wide variety of models comes to the forefront with its high mobility.

In the thesis, an image-based object tracking of a four-rotor unmanned aerial vehicle was carried out. First of all, a mathematical model of a four-rotor unmanned aerial vehicle was created. Parrot AR.Drone 2.0 Power Edition of Parrot is used for object tracking. In the object tracking process, the color, shape and coordinate information of the object to be tracked are used by using the images taken from the front camera of the device. In the Python programming language, the desired movements of the unmanned aerial vehicle have been achieved with code fragments written using AT (Attention) commands. By using OpenCV which is an important platform in image processing, color feature and shape feature are examined and processed separately in detecting moving object. In order to detect moving objects from images obtained from fixed cameras, color and shape features were used together to create a program that performs point and core based object tracking in the Python programming language. The generated program was connected to the four-rotor unmanned aerial vehicle via wifi, using the images obtained from the front camera of the device, and visual-based object tracking was performed. By connecting with AR.Drone 2.0 through the Autoflight program, the route line that the device was watching while moving was examined. In the EZBuilder program, images taken from the camcorder on AR.Drone 2.0 in the RGB (Red-Green-Blue)color space, a red moving object, in a HLS (Hue- Luminance-Saturation) color space, a yellow moving object and in the YCbCr (Luminance-Chrominance blue-Chrominance red) color space, yellow and green rectangular objects introduction and follow-up procedures were carried out.

Key Words: Quadrotor Unmanned Air Vehicle, Image Processing, Object Tracking

ŞEKİLLER LİSTESİ

Sayfa No

TABLOLAR LİSTESİ

Sayfa No

SEMBOLLER LİSTESİ

z	:Hava aracının düşey doğrultudaki konumu [m]
x	:Hava aracının yatay düzlemdeki konum bileşeni [m]
y	:Hava aracının yatay düzlemdeki konum bileşeni [m]
$\boldsymbol{\Omega_i}$	:Her bir rotorun açısal hızı rad/sn
$\boldsymbol{\Psi}$	:Hava aracının sapma açısı (yaw)
$\boldsymbol{\theta}$	:Hava aracının yunuslama açısı (pitch)
$\boldsymbol{\phi}$	:Hava aracının yalpalama açısı (roll)
$\dot{\Psi}$	:Hava aracının sapma hızı (yaw)
$\dot{\theta}$	:Hava aracının yunuslama hızı (pitch)
$\dot{\phi}$	:Hava aracının yalpalama hızı (roll)
m	:Hava aracının ağırlığı [kg]
l	:Hava aracının bir şaftının uzunluğu [m]
lx	:Hava aracının x eksenindeki atalet momenti [m^4]
ly	:Hava aracının y eksenindeki atalet momenti [m^4]
lz	:Hava aracının z eksenindeki atalet momenti [m^4]
j	**:**Hava aracına etkiyen jiroskopik etki
b	**:**Rotorlara etkiyen düşey doğrultudaki itki kuvveti (thrust)
d	**:**Rotorlara etkiyen dönel itki kuvveti (drag)

KISALTMALAR LİSTESİ

İHA	: İnsansız Hava Aracı
OpenCV	: Open Source Computer Vision Library
MLL	: Machine Learning Library
CV	: Computer Vision
AT	: ATtention
LPI	: İnç Başına Düşen Çizgi Sayısı (Line Per Inch)
DPI	: İnç Kareye Düşen Piksel Sayısı (Dot Per Inch)
RGB	: Kırmızı -Yeşil - Mavi (Red – Green - Blue) Renk Uzayı
CMYK	: Turkuaz - Pembe - Sarı - Siyah (Cyan - Magenta – Yellow - Key) Renk Uzayı
YUV	: Parlaklık - Mavi Tabanlı - Kırmızı Tabanlı (Luminance - Chrominancel1 - Chrominancel2) Renk Uzayı
YIQ	: Parlaklık - Turuncu, Mavi - Mor, Yeşil (Luminance - Orange, Blue - Purple, Green) Renk Uzayı
YCbCr	: Parlaklık - Mavi Renklilik - Kırmızı Renklilik (Luminance - Chrominance blue - Chrominance red) Renk Uzayı
HSV	: Renk Tonu - Doygunluk - Aydınlık (Hue - Saturation - Value) Renk Uzayı
HSB	: Renk Tonu - Doygunluk - Parlaklık (Hue - Saturation - Brightness) Renk Uzayı
HLS	: Renk Tonu - Aydınlık - Doygunluk (Hue - Luminance - Saturation) Renk Uzayı
HSI	: Renk Tonu - Doygunluk - Yoğunluk (Hue - Saturation – Intensity) Renk Uzayı

1. GİRİŞ

Uçma isteği veya yükseklerden aşağıya kuşlar gibi bakabilmek, insanlar tarafından çok eski yıllardan beri istenmiş bir eylemdir. Kuşların örnek alınmasıyla yapılan ilk nesne uçurtmadır. İlk uçurtmanın Çin, Endonezya veya Güney Pasifik Adaları'nda M.Ö. 1500 yıllarında yapıldığı düşünülmektedir. Uçurtmalardan günümüze kadar gelebilen en eskisi 1773 yılında Hollanda'da yapılmıştır [1].

Uçurtmalardan sonra insanların uçma girişimlerindeki ikinci nesneleri balonlardır. Balon fikri Henry Cavendish'in 1766 yılında hidrojenin havadan hafif olduğunu keşfetmesiyle başlamış ve Joseph Black'ın 1767'de hafif bir aracın hidrojenle doldurulduğu zaman uçabileceğini öne sürmesiyle devam etmiştir. Fakat ilk balon sıcak havayla uçurulmuştur. Fransız Joseph Michel Montgolfier (1740-1810) ve Jacques Etienne Montgolfier (1745-1799) kardeşler 5 Haziran 1783 tarihinde Annonay köyünde, ketenden yapılmış ve çapı 10,5 metre olan bir torbayı sıcak havayla doldurarak ilk uçuşu gerçekleştirmişlerdir. Bu balon 10 dakikada 1,5 millik mesafe kat ederek 450 metreye kadar yükselmiştir [2].

Fransız Fizikçi Jacques Charles (1746-1823) ısınan havanın, soğuk havaya göre daha yüksekte kaldığını ve soğudukça bu özelliğini kaybettiğini gözlemlemiştir. 27 Ağustos 1783 tarihinde, Jacques Charles balonun içine doğru yakılan ateşin havayı ısıtarak balonun yükselmesini sağlamasını ve hidrojen gazının daha hafif ve havada yüzme kabiliyetinin daha kalıcı olmasını kullanarak ilk hidrojen balonunu yaparak uçurmayı başarmıştır [2].

Leonardo Da Vinci tarafından tasarlanan "Hava Burgusu", pervane benzeri bir yapıya ve 5 metre çapa sahipti. Hava burgusunun pervanelerinin yeterli hıza ulaştığı takdirde kaldırma kuvvetini sağlayacağı fikrine dayanmaktaydı. "Hava Burgusu", havacılık uzmanlarına göre helikopterin atası olarak da kabul edilmektedir (Şekil 1.1.) [3].

Şekil 1.1. Hava burgusu [4].

Helikopterler dik olarak yükselebilen ve dik olarak inebilen döner kanatlı hava araçlarıdır. Aynı zamanda, dönen pervaneli hava araçları olarak da adlandırılan helikopterler, havada sabit tutunabilme, havada dönebilme, yana hareket edebilme, yukarı giderken arkaya ilerleyebilme gibi özelliklere sahiptirler. Bu hava araçları, çok çabuk bir şekilde yön değiştirebilirler ve hareket etmeyi tamamen durdurup havada sabit kalabilirler. Helikopterler ve diğer uçakların uçma ilkeleri temelde aynıdır. Uçakların havada kalabilmesi için gövdeye sabitlenen kanatlar kullanıldığından sürekli hareket ettirilir, ama helikopterlerde kanatlar sürekli döndüğünden helikopterin hareket etmesine gerek kalmamaktadır [5,6].

Pervanelerde iki ya da daha fazla kanat bulunur. Rotorlar pervaneleri döndürürken, pervanelerin üst yüzeyi ile alt yüzeyi arasında oluşan basınç farkı, helikopteri yukarı doğru harekete geçirmektedir. Pervanelerin dönüş hızı ile yerçekimine ters yönde vermiş olduğu kaldırma kuvveti doğru orantılıdır. Kaldırma kuvveti helikopterin ağırlığına eşit olduğunda helikopter havada asılı kalmaktadır. Kaldırma kuvveti helikopterin ağırlığından az ise helikopter dikey olarak alçalabilmektedir [5].

Leonardo Da Vinci'nin tasarımı olan Hava Burgusundan bu zamana kadar dikey iniş kalkış yapabilen hava araçları sürekli olarak gelişmekte ve farklı özelliklerde tasarlanmaktadır.

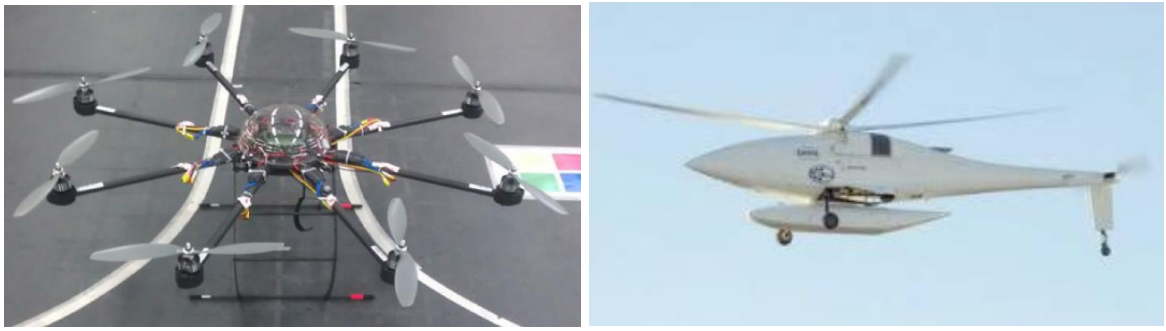
1.1. İnsansız Hava Araçları

Aerodinamik kuvvetlerin etkisiyle kendini kaldırabilen, belli bir uçuş planı üzerinden veya uzaktan kumanda edilerek hareket kabiliyetine sahip hava araçlarına İnsansız Hava Araçları (İHA) denilmektedir. İHA'lar görevlerini haberleşme, görüntüleme ve otomatik kontrol gibi alt işlevler sayesinde yerine getirmektedir.

1910'lu yıllarda ilk İHA'lar geliştirilmeye başlanmış ve az sayıda olmak üzere I. Dünya Savaşı sırasında kullanılmıştır. II. Dünya Savaşı boyunca çok sayıda İHA, saldırı görevlerinde yer almış fakat yıllarca uzaktan kumandalı uçak olmaktan ileriye gidememiştir [7,8]. İHA'ların uçaklara nazaran ucuz olması ve İHA'lara riskli görevler sırasında daha kolay işlem yaptırılabilmesinden dolayı gün geçtikçe kullanım alanı genişlemiştir. Günümüzde artık keşif ve gözetleme amacıyla kullanılan İHA'lar ayrıca silahlandırılmaktadır [3].

İHA'lar çok farklı ebat, şekil, model ve karakterde üretilmektedir. İHA'ları temel yapı olarak iki kısımda inceleyebiliriz. Bunlar; kaldırma kuvvetini kanatları vasıtasıyla (uçak) sağlayan İHA'lar (Şekil 1.2.) ve kaldırma kuvvetini pervaneleri vasıtasıyla elde eden (helikopter) İHA'lardır (Şekil 1.3.) [3,4,6].

Şekil 1.2. Kaldırma kuvvetini kanatları vasıtasıyla sağlayan İHA'lar [6,7].



Şekil 1.3. Kaldırma kuvvetini pervaneleri vasıtasıyla sağlayan İHA'lar [6,7].

1.1.1. İnsansız Hava Araçlarının Kullanım Alanları

Son yıllarda teknolojide meydana gelen gelişmelerle birlikte İHA'ların kullanımının artmaya başladığı gözlemlenmektedir. Manevra yeteneği yüksek olan İHA'lara özellikle zorlu arazi şartlarında gözetleme görevi yapabilmek için gereksinim duyulmaktadır. İnsanlar tarafından yerine getirilmesi tehlikeli ve zor olan alanlarda İHA'ların kullanımı her geçen gün artmaktadır. Özellikle hassas pilotaj gerektiren hedef ve düşman tespiti, takibi, trafik, güvenlik ve gözetim uygulamaları, sınır güvenliğini sağlama, arama kurtarma, zirai uygulamalar, doğal afet sonrası hasar tespiti, suç mahalli araştırması gibi geniş bir yelpazede İHA'ların kullanım alanı genişlemiş ve önemli ilerlemeler kaydedilmiştir [7,9].

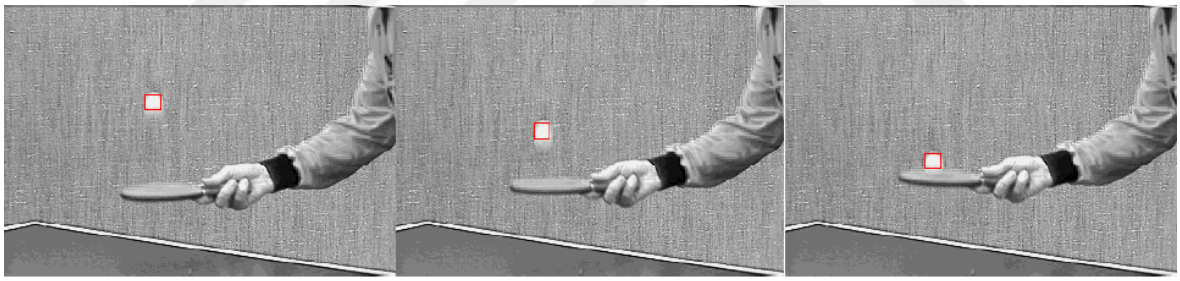
İnsansız hava araçlarının diğer kullanım alanları:

- ✓ Dijital Haritalama
- ✓ Havadan Mayın Tespiti
- ✓ Elektronik Harp
- ✓ Kentsel Harp
- ✓ Meteoroloji
- ✓ Kurtarma
- ✓ Havadan Video Çekimi
- ✓ Fotoğraf ve Video Çekimi
- ✓ Küçük paket taşıma
- ✓ Bilimsel araştırmalar vs.

1.2. Nesne Takibi Üzerine Yapılmış Çalışmalar

Bilgisayarlı görmenin konularından biri olan nesne takibi, nesnenin rengine, şekline ve hareketine göre farklılıklar göstermektedir. Literatür incelediğinde nesne takibi ile ilgili farklı çalışmalar yapıldığı gözlemlenmektedir. Trafik kontrolünde, tarım uygulamalarında, navigasyon sistemlerinde ve özellikle insansız denetim ve gözetim faaliyetlerinde, insansız hava araçları gibi bilgisayarlı görme alanının ilgilendiği birçok uygulamada nesne takibinin öne çıktığı görülmektedir.

Alper Yılmaz ve arkadaşlarının hazırlamış olduğu "Object Tracking: A Survey" isimli çalışmada, nesne takibinde ve takip aşaması öncesindeki adımlarda kullanılabilecek yöntemler, uygulamaya bağlı olarak seçilmesi gereken nesne gösterimleri, takip için kullanılacak olan özelliklerin seçimi, nesne bulma yöntemlerinden ve nesne takibi amacıyla kullanılabilecek yöntemler tanıtılmıştır [10]. Şekil 1.4.'te masa tenisi topunun takip edilmesi gösterilmiştir.



Şekil 1.4. Masa tenisi topunun takip edilmesi [10].

Qi Zang ve Reinhard Klette'in yapmış olduğu çalışmada takip edilen nesnenin hareketi Kalman filtresi kullanılarak belirlenmiştir [11]. Bu çalışmada hareketsiz kameradan alınan görüntü üzerinde nesne takibi üzerinde çalışılmıştır. Arka planın sabit olması sayesinde, hareketli nesneler arka plandan ayrılıp izlenmiştir. Hareketsiz kamera ile çalışan bu tarz uygulamalarda aktif bir nesne takibi söz konusu değildir. Bu sistemler daha ziyade trafik izleme ve düzenleme, kalabalık alanlardaki olağan dışı durumları kontrol etme gibi uygulamalara uyumlu olarak çalıştığı gösterilmiştir.

Kye Kyung Kim ve arkadaşları tarafından yapılan başka bir uygulamada, otonom robot üzerine yerleştirilmiş kameradan görüntü alınarak, nesne bulma ve takibi gerçekleştirilmeye çalışılmıştır. Bu sistemde takibi zorlaştıran; hedef nesnenin ve kameranın hareketli olmasıdır. Kamera hareketinin doğrultusu ve hızı kullanılarak takip edilecek nesnenin hareket doğrultusu ve hızı hesaplanarak problem aşılmaya çalışılmıştır [12].

Takashi Morimoto ve arkadaşları tarafından yapılan çalışmada görüntü bölütlemesi sonucu elde edilmiş parçaların özelliklerinin çıkarılması ve şablon eşleme yöntemiyle nesne takibi yapılmıştır [13]. Geliştirilen sistemin karmaşıklık düzeyinin yüksek olması, sistemin gerçek zamanlı uygulamalara adaptasyonunu güçleştirmiştir. Bunun üzerine Takashi Morimoto ve arkadaşları, sistemin modüllerini, paralel çalışabilmeleri için FPGA / ASIC yapısını kullanarak yeniden tasarlamışlardır [14].

Soon-Yong Park ve arkadaşları tarafından yapılan çalışmada ise önceden kaydedilmiş görüntüler üzerinde nesne takibi yapılmıştır. Bu çalışmada hareketli nesnelerin takip edilerek bulunması ve silinmesi amaçlanmıştır. Kullanılan kameranın hareketli olmasından dolayı arka arkaya gelen görüntüler birleştirilerek bir mozaik oluşturulmuş, böylece kamera hareketi elimine edilmiştir [15].

Isaac Cohen ve Gerard Medioni'nin yapmış olduğu çalışmada uçan bir platformdan alınan kayıtlı video verisi üzerinde nesne takibi yapılmaya çalışılmıştır. Bu çalışmada hareketli nesneleri takip etmek için iki aşamalı bir yaklaşım sunulmuştur. Öncelikle gözlemleme platformunun hareketinin neden olduğu görüntü akışı bulunmuş ve elimine edilmiş, daha sonra iki boyutlu şablon kullanılarak hareketli nesne izlenmiştir [16].

Alper Yılmaz'ın "Object Contour Tracking Using Level Sets" isimli çalışmasında, hareketli kameradan alınan video görüntülerini kullanan sistemde, doku ve renk özelliklerinin olasılık dağılım fonksiyonlarına dayanan Bayes çıkarımı kullanılarak kenar takibi algoritması ile nesne takibi yapılmaya çalışılmıştır [17].

Dorin Comaniciu'nun esnek yapılı nesnelerin takibi üzerine yapmış olduğu çalışmada, gerçek zamanlı olarak hareketli kameradan alınan görüntüler üzerinde esnek yapılı nesnelerin takibi gerçekleştirilmiştir. Dorin Comaniciu'nun yapmış olduğu çalışmada

Ortalama Değer Kayması (Mean-Shift) iterasyonları ile takip edilecek nesnenin ardışık her bir görüntüde yerinin belirlenmesi hesaplanmıştır [18].

Dorin Comaniciu'nun diğer bir çalışmasında ise esnek yapılı hedef nesnenin takibi, hedef nesneyi temsil eden bir geometrik şekil ve bu şekil içerisindeki özellik dağılımı kullanılarak gerçekleştirilmiştir. Bu çalışmada, Bhattacharyya uzaklığı ile benzerlik hesaplanmış, optimizasyon için ise Mean-Shift algoritması kullanılmıştır [19].

Chiu .C. C. ve ekibi tarafından 2010 yılında sunulan çalışmada sabit kamera ile sahne çıkartım metodu ile araç tespiti ve izleme işlemi gerçekleştirilmiştir. Çalışmada sabit mesafeden tespiti yapılan araçların yatay ve düşey eksendeki piksel sayıları kullanılarak bu araçlara ait genişlik ve yükseklik bilgileri bulunmakta ve aracın otomobil, minibüs, kamyon veya otobüs mü olduğuna ait tanıma işlemi yapılmaktadır [20].

Literatürde yapılan bazı çalışmalarda nesne takibinde karşılaşılan problemler üzerinde durulmuş ve bu problemlere çözümler üretilmeye çalışılmıştır. Bu çalışmalardan birinde, görüntü özelliklerinin değişimine ve hedef nesnenin tamamının görüntülenememesine rağmen, hedef nesne, enerji fonksiyonu kullanılarak takip edilmiştir [21].

Çek Teknik Üniversitesinde Krajn'ık liderliğindeki bir ekip, Parrot AR.Drone kullanarak robotik araştırma ve eğitim için bir platform geliştirmeye çalışmışlardır [22].

Jakob Julian ve ekibi eşzamanlı lokalizasyon ve haritalama (SLAM) yöntemini içeren bir quadcopterin otonom kamera tabanlı navigasyon yöntemini incelemiş ve uygulamıştır [23].

Burkay Birant Örten'in görsel güvenlik sistemi için nesne takibi üzerinde yapmış olduğu çalışmada, ortam şartlarının değişmesini fark edebilecek öğrenme kapasitesine sahip, ayrıca nesne maskelerinden gölgeleri de ayırabilen bir arka plan modelleme yöntemi üzerinde durulmuştur [24].

Diğer bir çalışmada ise, nesneyi bulma, nesneyi sınıflandırma ve nesneyi takip etme yeteneklerine sahip, sabit bir kameradan elde edilen renkli ve renksiz görüntüler üzerinde işlem yapabilen akıllı video gözetim sistemi üzerinde çalışılmıştır [25].

1.3. Tezin Amacı ve Yapılan Çalışma

Tezde görüntü işleme teknikleri kullanılarak dört rotorlu insansız hava aracının görme tabanlı nesne takibi yapması amaçlanmıştır. Dört rotorlu hava aracının üzerinde bulunan kameradan alınan görüntülerin wifi bağlantısı ile kontrol bilgisayarına aktarılması, OpenCV'de görüntü işleme aşamalarından geçerek, özellikleri önceden belirlenen hareketli nesnenin görüntüde gerçek zamanlı olarak değerlendirilip bulunması ve dört rotorlu insansız hava aracı tarafından takip edilmesi hedeflenmiştir.

Görüntü işlenmesinde birinci adım olarak, bilgisayar üzerinde bulunan sabit kameradan alınan renkli görüntülerden, aranan renkteki hareketli nesnenin tespiti ve nokta tabanlı takibi Python ortamında gerçekleştirilmiştir. İkinci adımda sabit kameradan gelen görüntülerden takip edilmek istenen hareketli nesnenin tespiti yapılıp etrafına geometrik şekil çizilmesi sağlanarak çekirdek tabanlı nesne takibi Python ortamında sağlanmıştır. Üçüncü adımda hem renk hem de şekil özelliği kullanılarak takip edilmek istenen nesnenin Python ortamında algılanması sağlanarak, nokta ve çekirdek tabanlı nesne takibini sağlayan programın Python'da yazılmıştır. Daha sonra yazılan bu programla dört rotorlu hava aracının üzerinde bulunan kameradan görüntü çekimi yapılarak görüntü işleme işlemini yapacak olan bilgisayara görüntü alımı yapılmaktadır. Önceden özellikleri belirlenen hedef nesne görüntü üzerinde aranmaktadır. Hedef nesne görüş açısına girene kadar dört rotorlu hava aracı hedef nesneyi beklemekte ve sabit uçuşunu gerçekleştirmektedir. Hareketli nesne dört rotorlu hava aracında bulunan kameranın görüş açısına girdiği zaman nesne bulma algoritması hedef nesnenin önceden belirlenen özelliklerini kullanarak hedef nesnenin yerini belirlemekte ve böylelikle son aşama olan nesne takibi aşaması başlamaktadır. Nesne takibi aşaması görüntüler üzerinde çalışarak hedef nesnenin bir önceki görüntüde bulunduğu bölgenin civarında hedef nesneyi aramakta ve nesnenin bulunduğu yeri bir sonraki adıma aktarmaktadır.

Benzer bir şekilde EZ-Builder programında AR.Drone 2.0 üzerindeki kameradan alınan görüntülerden RGB, HLS ve YCbCr renk uzaylarında belirlenen renklerdeki hareketli nesnelerin sisteme tanıtılması ve takip işlemleri gerçekleştirilmiştir. Ayrıca Autoflight programı üzerinden AR.Drone 2.0 ile bağlantı sağlanarak cihazın hareket halinde iken izlemiş olduğu rota çizgisi incelenmiştir.

1.4. Tezin Yapısı

Dört rotorlu insansız hava aracının nesne takibinin yapılmasını amaçlayan bu tez çalışması toplam yedi bölüm olarak hazırlanmıştır. Tezin birinci bölümünde insansız hava araçları ve kullanım alanları hakkında bilgi verildikten sonra nesne takibi üzerine yapılmış çalışmalar incelenmiştir.

Tezin ikinci bölümünde nesne takibi gerçekleştirebilmek için tezde kullanılan donanım ve yazılımlardan bahsedilmektedir. Üçüncü bölümünde dört rotorlu hava aracının hareketleri hakkında bilgi verilerek dört rotorlu hava aracının matematiksel modeli çıkarılmıştır.

Dördüncü bölümde görüntünün temel bileşenleri hakkında bilgi verilerek renk uzayları açıklanmıştır. Beşinci bölümde görüntü işleme teknikleri ve nesne takip yöntemleri hakkında bilgi verilmiştir.

Altıncı bölümde Python, dili içerisinde OpenCV'nin görüntü işleme teknikleri kullanılarak, nesnenin renk özelliği seçilip nokta tabanlı nesne takip işlemi, şekil özelliği kullanılarak çekirdek tabanlı nesne takibi yapılmıştır. Hem renk hem de şekil özelliği kullanılarak nokta ve çekirdek tabanlı nesne takip işlemi gerçekleştirilmiştir. AutoFlight programında nesnenin hareketi sırasında yapmış olduğu rota incelenmiştir. EZ-Builder programında RGB, HLS ve YCbCr uzaylarında belirlenen renkteki nesnelerin sisteme tanıtılması ve takibi ile ilgili uygulamalar gerçekleştirilmiştir. Siluet tabanlı nesne takip için veri tabanına nesne ekleme işlemi sağlanarak veri tabanından seçilen nesnenin takip işlemi gerçekleştirilmiştir.

Yedinci bölümde ise elde edilen sonuçlar değerlendirilmektedir.

2. KULLANILAN DONANIM VE YAZILIMLAR

2.1. Dört Rotorlu Hava Aracı (Quadrotor)

Quadrotorlar (Unmanned Aerial Vehicle-UAV) dikey olarak kalkış ve iniş yapabilen dört rotorlu hava araçlarıdır. Birbirinden bağımsız dört rotordan oluşan, hızlı ve denetimli manevra yapabilen bir hava aracıdır. Dört rotorun hızlarını değiştirerek bütün uzaysal hareketlerini yapabilmesi, dikey iniş ve kalkış (VTOL-Vertical Take-Off and Landing) yapabilmesi, askıda kalabilmesi quadrotorun daha çok tercih edilmesini sağlamıştır. Bu avantajlarından ötürü quadrotorun birçok modeli tasarlanmış, kontrol algoritmaları yapılmış ve akademik çalışmalarda yer almıştır [7-9,26-28]. Şekil 2.1 de çeşitli quadrotor örnekleri gösterilmiştir.

Şekil 2.1. Çeşitli quadrotor örnekleri [7-9].

2.1.1. Parrot AR.Drone 2.0

Bu tezde, incelenen tekniklerin pratikte değerlendirilmesine yardımcı olmak için, dört rotorlu hava aracı olarak Parrot firmasının AR Drone adlı aracının ikinci sürümü olan Parrot AR.Drone 2.0 Power Edition ürünü kullanılmıştır. (Şekil 2.2.)

(a) **(b)**

Şekil 2.2. Parrot AR.Drone 2.0 (a) açık alan (b) kapalı alan üst gövde tasarımı [29].

Parrot AR Drone 2.0'ın tezde kullanılmasının başlıca sebepleri; üstün manevra yetenekleri, açık kaynak kodlu yazılım geliştirme ortamı (SDK), düşük maliyeti ve yedek parça desteğinin bulunmasıdır. AR Drone 2,0 " × " (çarpı) tipinde bir dört rotorlu hava aracı tasarımına sahiptir. Rotorlar karbon fiber tüpler üzerine yerleşmiştir. Yapının geneli ise sert köpükten oluşmaktadır. Gövde üzerine pili sabitleyen ve üzerini kaplayan ikinci bir köpük parça bulunmaktadır. Bu parça, kapalı alanlarda kullanılacağı zaman pervaneleri ve pervanelerin etrafına verebileceği zararları engelleyecek bir kafes görevi de görebilmektedir. Araç, 11.1 voltluk 1000 mAh'lik veya 1500 mAh'lik 10C boşalma oranı olan bir adet Li-Po pil ile beslenmektedir. Kullanılan pil, tam şarjda 12.5 volt, düşük şarj durumunda ise 9 volt çıkış vermektedir. Şarj bilgisi, gömülü sistem tarafından sürekli takip edilmektedir. Şekil 2.2.'de gösterilen, kapalı alanlar için kullanılan üst gövde ve standart pil ile birlikte aracın toplam uçuş ağrılığı 420 gramdır [29].

AR Drone, dört adet 15 Watt'lık fırçasız DC motordan güç almaktadır. Motor sürücüleri, barındırdıkları algılayıcı ile motorun dönüş ve duruş durum bilgilerine sahiptir. Acil durumlarda, motorları ve pervaneleri zorlamamak için otomatik olarak motor hızlarını azaltarak iniş yapabilmektedir. Motorların her birinin normal uçuş sırasındaki devir sayısı ortalama 28500 d/dk civarındadır. Yüksek devirli motorlar, 8 inçlik yüksek atak açılı

pervaneleri döndürmektedir. Motor ile pervaneler arasında 1/8.75 çevrim oranı olan bir dişli çifti yer almaktadır [29].

Araç mekanik özellikleri yanında sahip olduğu elektronik özelliklerle akademik çalışmalara oldukça uygundur. AR Drone 2'nin ana kartı 1 GHz ARM Cortex A8 işlemciden güç almaktadır. Sistem 1 GB'lık 200 MHz'lik DDR2 RAM belleğe sahiptir. Cihazda 800 MHz'lik paylaşımsız video destekli dijital sinyal işlemci mevcuttur. Bu özellikler sayesinde üzerinde 32 bit'lik bir işletim sistemi rahatlıkla çalıştırabilmektedir. Gömülü Linux işletim sistemlerinden biri olan BusyBox, bu anakartta rahatlıkla çalışabilmekte ve tüm sistemi gerçek zamanlı olarak kontrol edebilmektedir.

AR Drone 2.0'ın ana kartında TCP/IP iletişime olanak veren IEEE 802.11 kablosuz iletişim protokolü bulunmaktadır. Günümüzde çok sık kullanılan ve 2.4 GHz frekans bandından yayın yapan kablosuz yerel ağ (WLAN) ile aynıdır. Böylelikle Wi-Fi arayüzü bulunan tüm cihazlara anlık video aktarımı yapılabilmektedir. Bir diğer iletişim imkanı ise üzerinde barındırdığı USB 2.0 bağlantı kapısıdır. Parrot AR Drone 2.0' a istendiği takdirde GPS, Linux işletim sistemi tarafından tanınabilecek her hangi bir USB cihaz takılarak sisteme entegre edilebilmektedir.

AR Drone 2.0 üzerinde aşağıdaki sensörler bulunmaktadır.

- 3 eksen, 2000 °/sec hassasiyetli jiroskop,
- 3 eksen ±50 mg hassasiyetli ivmeölçer
- 3 eksen 6° hassasiyetli manyetometre
- ±10 Pa hassasiyetli basınç algılayıcısı
- Ultrasonik algılayıcı

AR Drone üzerinde iki tane kamera bulunmaktadır. Ön kamerada 90° açılı CMOS sensörü vardır. AR Drone görüntü verilerini otomatik olarak işleyerek kablosuz bağlantı üzerinden karşı tarafa yollar. İki kamera da 360p (640x360) ya da 720p (1280x720) görüntü çözünürlüğüne sahiptir.

2.2. OpenCV

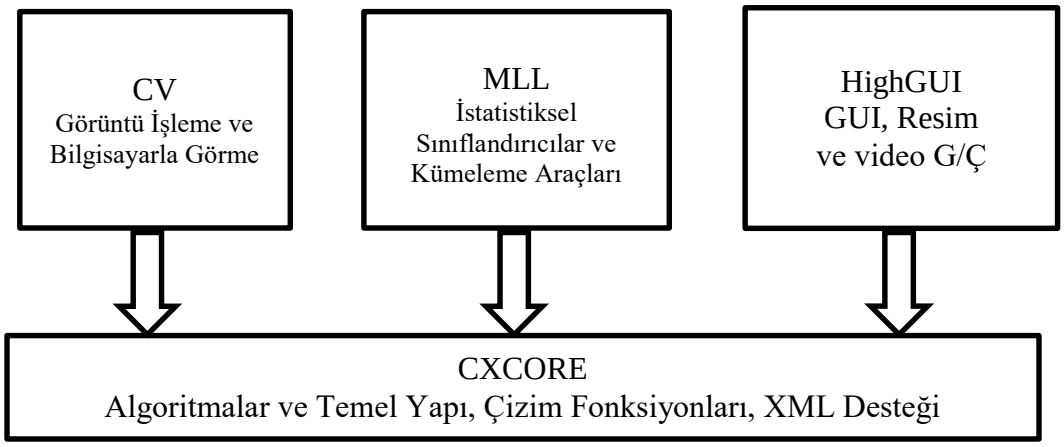
Intel firması tarafından ilk sürümü 1999 yılında çıkarılan OpenCV, bilgisayarlı görü ve gerçek zamanlı görüntü işleme uygulamalarının gerçekleştirildiği platformdur. OpenCV'nin İngilizce açılımı "Open Source Computer Vision Library" dir. "Açık Kaynak Kodlu Bilgisayarlı Görü" olarak Türkçe'de kullanılmaktadır [30].

OpenCV, Linux, Windows, PSP (PlayStation Portable), IOS, android ve birçok gömülü ve mobil sistem platformunda çalışabilen, C ve C++ dilleriyle yazılmış, görüntü işleme (image processing) uygulamaları ve gerçek zamanlı bilgisayarla görme (real time computer vision) için kullanılabilen, açık kaynak kodlu bir kütüphanedir [31]. Açık kaynak kodlu olması, ticari kullanımının ve eğitim amaçlı kullanılmasının ücretsiz olması diğer görüntü işleme araçlarından en büyük farklılığıdır [32].

OpenCV, birçok deneyimli yazılımcının çalıştığı Willow Garage şirketi tarafından desteklenmektedir. OpenCV görüntü işleme aracı birçok açık kaynak yazılımına ev sahipliği yapan SourceForge sitesinden indirilebilmektedir. İçerdiği fonksiyonların birçoğu platformdan bağımsız olarak çalışır. Mevcut olan C ara yüzüne ek olarak OpenCV 2.0 sürümünden itibaren, C++ ara yüzü de eklenmiştir.

Günümüzde, OpenCV içerisindeki görüntü işleme algoritmaları ve bilgisayarla görme kullanılarak;

- İnsan-Bilgisayar Etkileşimi
- Nesne Kimliklendirme, Nesne Bölümleme ve Nesne Tanıma
- Yüz Tanıma
- İşaret Dili Tanıma
- Hareket Yakalama, Hareket Algılama ve Hareket Takibi
- Çiftli ve Çoklu Kamera Kalibrasyonu ve Derinlik Hesaplama
- Hareketli Robot Teknolojileri uygulamaları geliştirilebilmektedir.



Şekil 2.3. OpenCV yapısı ve içeriği

OpenCV kütüphane yapısında beş temel bileşen mevcuttur. Bu bileşenlerden dördü Şekil 2.3.'te gösterilmektedir. CV bileşeni ismini Computer Vision (Bilgisayarla Görme) kelimesinin baş harflerinden almıştır. CV bileşeninin içeriğinde bilgisayarla görme algoritmaları ve görüntü işleme fonksiyonları vardır. Machine Learning Library kelimelerinin baş harflerini alan MLL bileşeni ise isminden de anlaşılacağı üzere, makina öğrenmesi için ihtiyaç duyulan istatistiksel verilere ulaşmak ve mevcut olan verileri sınıflandırmak amacıyla kullanılan araç ve fonksiyonlardan oluşur. Diğer bir bileşen olan HighGUI bileşeni ise OpenCV kütüphanesinde tanımlanmış olan pek çok nesneyi (slider, form) oluşturabilmemizi sağlayan bir grafik arabirimidir. Ayrıca bu bileşen resim ve videoları yüklemek, hafızaya kaydetmek, hafızadan veri silmek için gerekli giriş/çıkış (I/O) komutlarını da içerir. CXCore bileşeni ise, OpenCV' deki Ip1Image, cvPoint, cvMat, cvSize, cvHistogram gibi veri yapılarına sahip XML desteği ne sahip bir kütüphanedir. Son olarak CvAux bileşeni, şekil eşleştirme (shape matching), yüz tanıma (face-recognation), objenin ana hatlarını bulma (finding skeletons), vücut hareketlerini tanıma (gesture recognition), ağız hareketlerini izleme (mouth-tracking) ve kamera kalibrasyonu gibi daha pek çok deneysel algoritmaları bünyesinde barındıran bir kütüphanedir [33].

2.3. Python

Guido Van Rossum tarafından 1991 yılında geliştirilen Python, çok güçlü ve yüksek seviyeli, dinamik nesne yönelimli programlama dilidir [34]. Python platformdan bağımsız bir programlama dili olduğundan dolayı Windows, Linux, BSD, Mac OS X, Solaris, AIX, AS/400, AROS, MorphOS, BeOS, S60, iPhone, iPOD, Macintosh ve Android dahil tüm işletim sistemleri ve büyük donanım platformları üzerinde çalışabilmektedir. Ayrıca Python programlama dilinin temiz ve basit söz dizimi, onu Ruby ve Perl gibi alternatiflerin önüne geçirmiştir. Phyton programı Eric S. Raymond gibi birçok programcı ve Google tarafından tercih edilen bir dil haline gelmiştir [35]. Python dilinin söz diziliminin basit olması sayesinde hem program yazmak, hem de başkası tarafından yazılmış bir programı okumak, başka dillere kıyasla daha kolaydır [36].

Şekil 2.4. Python dilini öne çıkaran yönler

Şekil 2.4.'te Python dilini öne çıkaran yönler gösterilmiştir. "Thinking in C++" ve "Thinking in Java" kitaplarının yazarı Bruce Eckel'e göre Python dili kadar hiçbir dil üretken değildir ve diğer programlama dillerinin programcıların işini kolaylaştırmaya yönelik bir amacı yoktur [37].

2.4. Attention Komutları

Attention komutları modemleri kontrol etmek için kullanılan protokoldür. AT, ATtention kelimesinin kısaltmasıdır. AT komutları sayesinde AR.Drone ile veri alışverişi sağlanabilmektedir. Komutlar AR.Drone'ye UDP (User Datagram Protocol) veya TCP (Transmission Control Protocol) paketleri olarak gönderilir. Tablo 2.1.'de genel olarak kullanılan AT komutları ve işlevleri gösterilmiştir.

Tablo 2.1. AT komutları ve işlevleri

AT komut adı	Komutun Yaptığı İşlev
AT * REF	Kalkış, iniş, reset ve acil stop için kullanılmaktadır.
AT * PCMD	AR.Drone hareketini kontrol etmek için kullanılır.
AT * FTRIM	Yatay düzlem için referans belirler.
AT * CONFIG	AR.Drone'yi yapılandırır.
AT * ANIM	AR.Drone üzerinde bir uçuş animasyonu ayarlar.

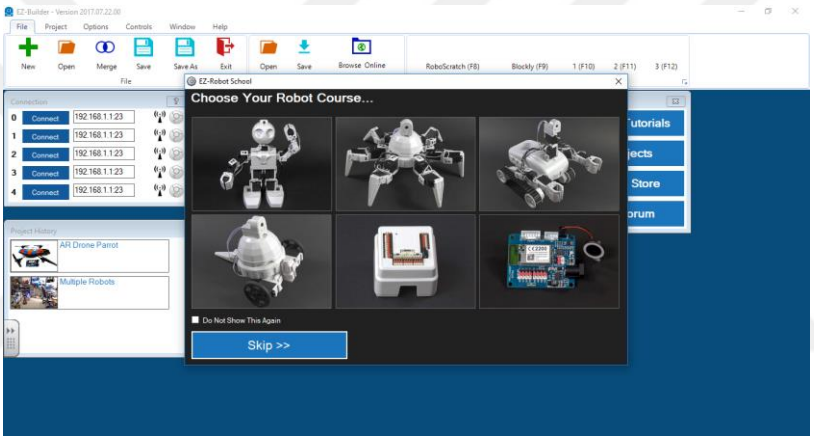
2.5. AutoFlight 0.2

İnsansız hava aracının kontrolü için hazırlanmış olan açık kaynak kodlu bir yazılımdır. Program ekranından insansız hava aracının uçuş bilgilerine ulaşılabilmekte ve uçuş kontrolü yapılabilmektedir. İnsansız hava aracının kamerasından alınan görüntüler eş zamanlı olarak program ekranından görüntülenmektedir. AutoFlight programı açık kaynak kodlu bir yazılım olduğundan Python programlama dilinin etkin bir şekilde kullanılmasına olanak sağlamaktadır. Şekil 2.5.'te AutoFlight programının ana ekran görüntüsü gösterilmiştir.

Şekil 2.5. AutoFlight programı ana ekran görüntüsü

2.6. EZ-Builder

EZ-Builder, robotları kontrol etmek için geliştirilmiş olan ücretsiz bir kontrol yazılım uygulamasıdır. Birçok donanım ve robot ile uyumlu çalışan grafiksel bir arayüze sahiptir. Farklı programlama dilleri için geliştirilmiş kütüphanelere sahiptir. Servo motor kontrolünden, görüntü işlemeye kadar birçok farklı alanda çalışmayı sağlayan bu sistem üzerinde script geliştirilmektedir. Böylece diğer programlama dillerinden bağımsız olarak da robot ve donanım kontrolü sağlanmaktadır. Şekil 2.6.'da EZ-Builder programının ekran görüntüsü gösterilmiştir.



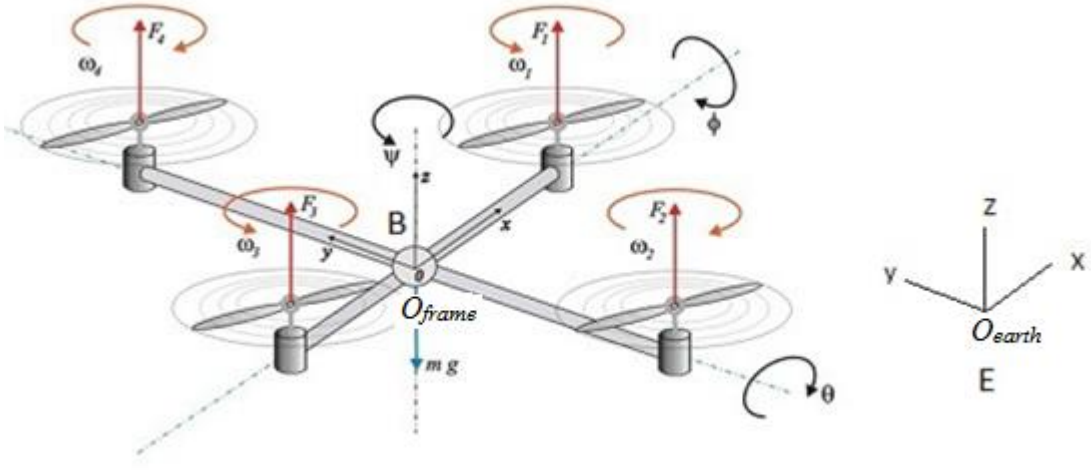
Şekil 2.6. EZ-Builder programı ekran görüntüsü

3. DÖRT ROTORLU HAVA ARACI

Quadrotorun karmaşık dinamiklerini tanımlamak için öncelikle koordinat sisteminin tanımlanması gerekmektedir. Quadrotorun yapısı Şekil 3.1.'de gösterilmiştir. Şekil üzerinde Rotorların ürettiği kaldırma kuvvetleri, iskelet üzerindeki koordinat sistemi O_{frame} ve dünya üzerindeki sabit koordinat sistemi O_{earth} gösterilmiştir.

Quadrotorun hareketlerini tanımlamak için iki eksen takımı gerekmektedir. Birinci eksen takımı quadrotorun gövdesine sabit referans ekseni (Body frame) O_{frame} eksen takımıdır. Nesne hareketli olduğu için mobil eksen takımı olarak adlandırılabilir. O_{frame} eksen takımı açısal hareketleri tanımlamak için kullanılır. İkinci eksen takımı ise dünya üzerinde sabitlenmiştir. Hareket etmediği için yeryüzü atalet referans ekseni (Earth frame) veya O_{earth} olarak isimlendirilir.

Dinamik davranış quadrotorun atalet eksenine (Earth frame) göre belirlenir. Quadrotorlar, altı serbestlik dereceli olup doğrusal eksen takımında (3 boyutta) ve açısal koordinatlarda (3 boyutta) hareket edebilmektedir. Açısal hareket yunuslama (pitch), yalpalama (roll) ve yönelme (yaw) açıları ile ifade edilir. Quadrotor pervanelerinin dönme yönü, bu dönmeden dolayı oluşan kaldırma kuvvetleri (F_i), dönme açıları ve hareket koordinatları Şekil 3.1.'de gösterilmiştir [7,9,38,39].



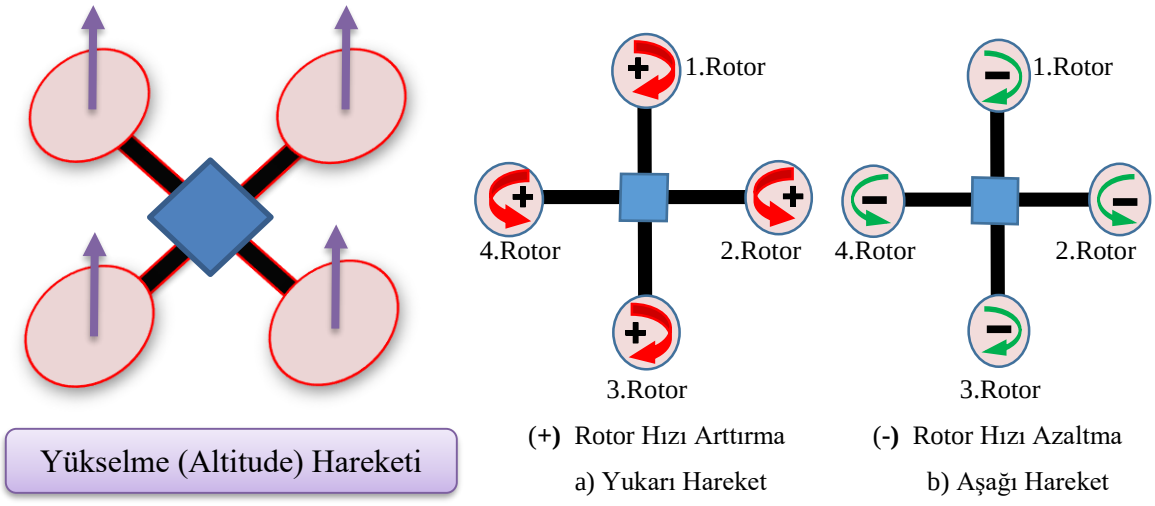
Şekil 3.1. Quadrotorun kaldırma kuvvetleri ve kullanılan koordinat sistemleri [38].

3.1. Quadrotorun Hareketleri

Quadrotor kaldırma kuvvetini, uçlarındaki pervanelerden alır. Aracın konumu ve yüksekliği bu pervanelerin dönüş yönleri ve açısal hızları denetlenerek gerçekleştirilir. Böylece araca istenen manevra yeteneği sağlanır. Kütle merkezi etrafında bir hava aracının yönünü tanımlamak için yönelme, yunuslama ve yalpalama açıları kullanılmaktadır. Pervanelerin açısal hızlarının ve yönlerinin değişimi ile bu açılar oluşur ve istenilen yönlerde hareket sağlanır [39]. Quadrotor açısal hızı daha az olan pervane yönünde ilerleme yapmaktadır. Ayrıca quadrotorun z ekseninde yapmış olduğu yükselme hareketi (altitude) pervanelerin aynı hızlarda döndürülmesi ile sağlanmaktadır.

3.1.1. Yükselme Hareketi

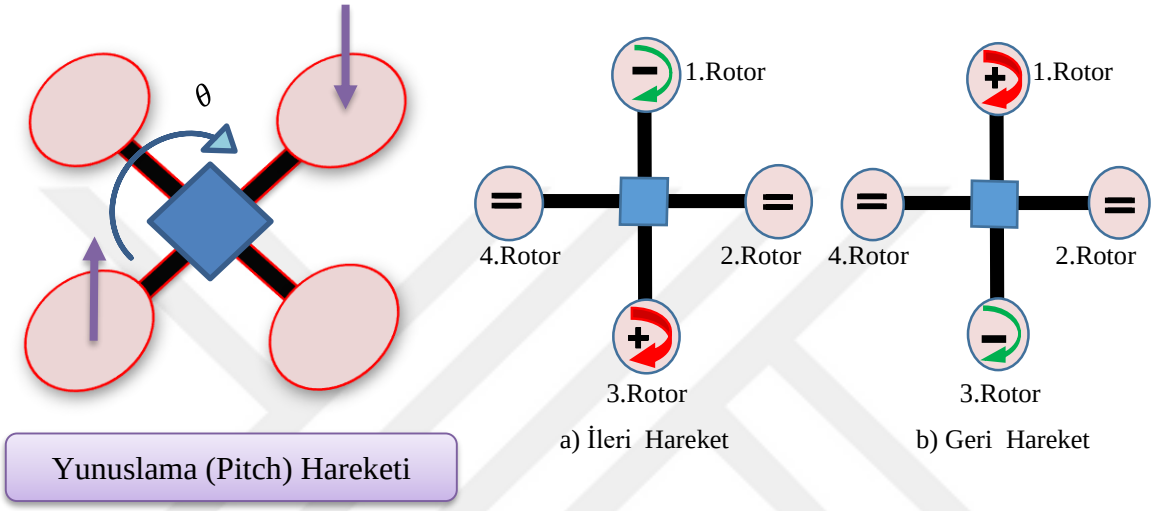
Ters yönlerde (saat yönü ve saat yönünün tersi) dönen karşılıklı iki pervane çiftinin (1-3 ve 2-4 numaralı) açısal hızlarının aynı oranda arttırılması ile z yönünde yükselme hareketi sağlanır. Bu iki pervane çiftinin açısal hızlarının aynı oranda azaltılmasıyla da iniş sağlanır. Bu iki pervane çifti belirli sabit açısal hızda dönerse sistem havada sabit olarak kalır [8,9,39]. Quadrotorun yukarı ve aşağı yöndeki hareketleri Şekil 3.2.'de gösterilmiştir.



Şekil 3.2. Quadrotorun yukarı ve aşağı yöndeki hareketleri

3.1.2. Yunuslama Hareketi

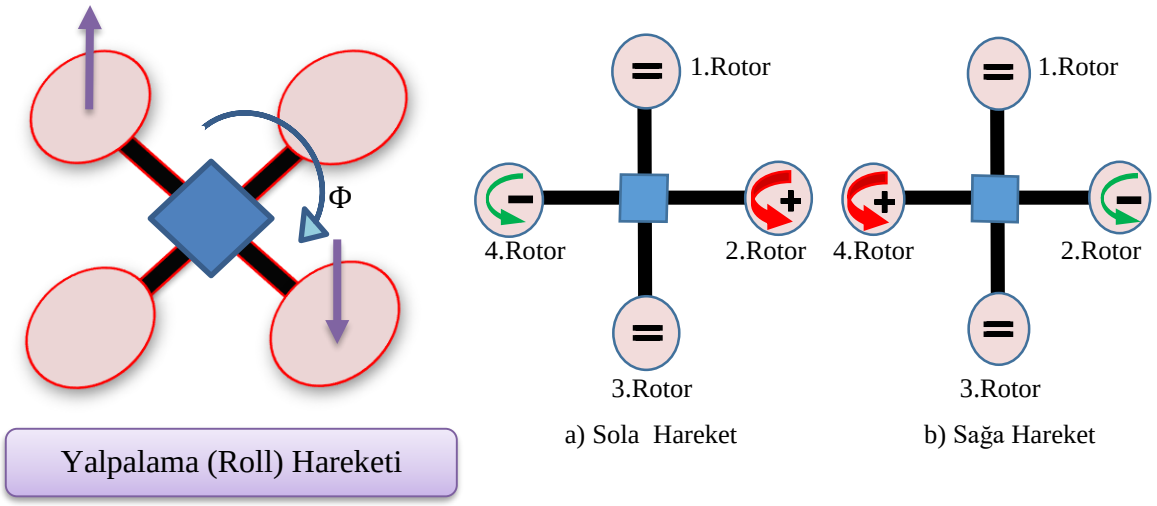
Yunuslama açısındaki (theta) değişim, aynı yönde dönen 1 ve 3 numaralı pervanelerdeki zıt hız değişimi sonucu oluşur. Böylece gövdenin y ekseni yönünde hareketi sağlanarak, aracın ön veya arkaya doğru eğilmesini sağlanır. Bu işlem boyunca z ekseni yönündeki itme kuvveti sabit tutulur [9]. Quadrotorun yunuslama hareketleri Şekil 3.3.'te gösterilmiştir.



Şekil 3.3. Quadrotorun yunuslama hareketleri

3.1.3. Yalpalama Hareketi

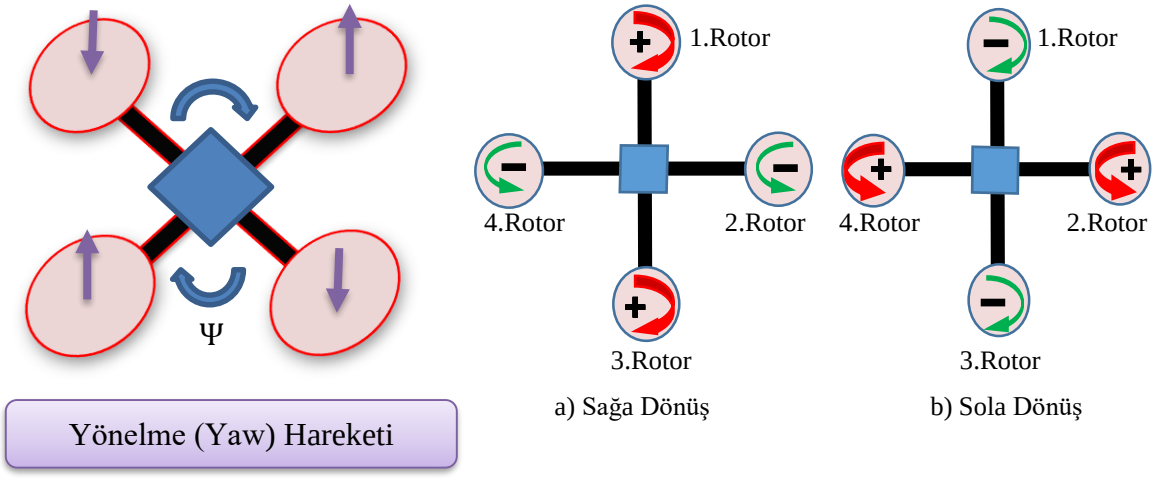
Yalpalama açısındaki (phi) değişim 2 ve 4 numaralı pervanelerindeki zıt yöndeki hız değişimiyle sağlanır. Bu değişimle gövdenin x ekseni etrafındaki hareketi oluşturulur. Yalpalama hareketinin elde edilebilmesi için yine z eksenindeki itme kuvveti sabit tutulmaktadır [9]. Quadrotorun yalpalama hareketleri Şekil 3.4.'te gösterilmiştir.



Şekil 3.4. Quadrotorun yalpalama hareketleri

3.1.4. Yönelme Hareketi

Yönelme açısı (psi) aracın dikey z-ekseni etrafında dönme hareketini ifade eder. 1-3 ve 2-4 pervanelerindeki hız değişimi sonucu oluşan aerodinamik tork dengesinin uyumsuzluğu ile oluşur. Karşılıklı pervane çiftlerinden birinin açısal hızları arttırılıp diğerinin açısal hızı azaltılırsa araç z ekseninde sapma yapar. Quadrotorun yönelme hareketleri Şekil 3.5.'te gösterilmiştir.

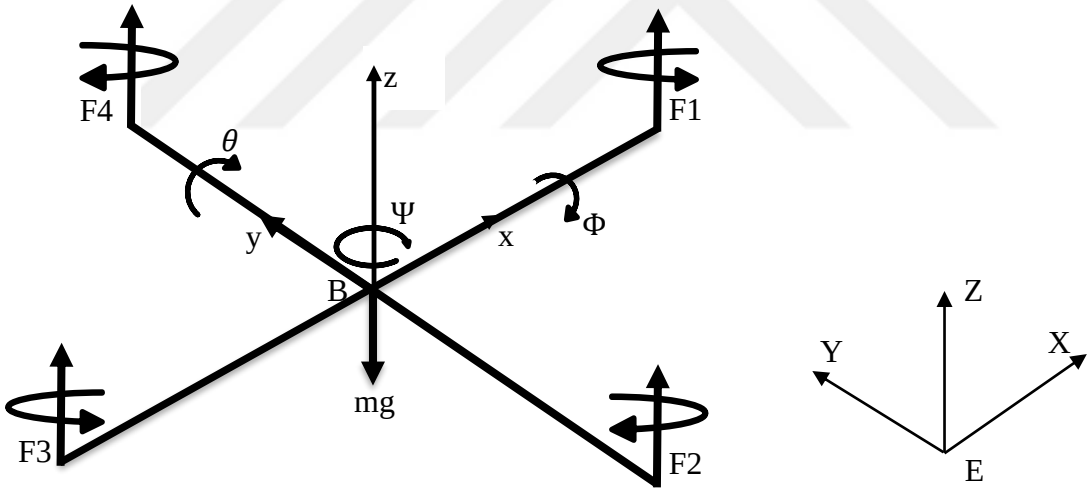


Şekil 3.5. Quadrotorun yönelme hareketleri

3.2. Quadrotorun Matematiksel Modelinin Çıkarılması

Hava araçlarının dinamik modelinin elde edilmesinde, araç üç boyutlu ortamda hareket eden katı bir cisim olarak düşünülmekte ve cismin gövdesine uygulanan tork ve kuvvetler buna göre kabul edilmektedir. Dinamik model, sistem davranışının zamanla değişimini açıklayabilmek için kullanılır. Quadrotorun dinamik modelinin elde edilmesinde bazı kabuller yapılmaktadır [40]. Bu kabuller:

- Modeldeki bütün parçalar sabit kütleli katı (rijit) cisim olduğu,
- Quadrotorun tam olarak simetrik bir yapıda olduğu,
- Hava basıncı etkisinin önemsenmediği,
- Pervanelerden elde edilen itme ve sürüklenme, rotor hızının karesiyle orantılı kabul edilmektedir.



Şekil 3.6. Quadrotorun eksenleri, Euler açıları ve etkiyen kuvvetler

Şekil 3.6.'da görüldüğü gibi yunuslama açısı θ, yalpalama açısı Φ ve sapma açısı Ψ olarak tanımlanmıştır. Quadrotorun matematiksel modelini oluşturulurken göz önüne alınan kuvvetler, eksen takımları Şekil 3.6.'da gösterilmiştir. Newton-Euler biçimlemesinden yola çıkılarak model çıkarımında iki eksen takımı göz önünde bulundurulmuştur. Bunlardan birincisi kütle yerçekimi merkezine göre alınan yer eksen takımı, ikincisi quadrotorun kütle merkezine bağlanmış gövde eksen takımıdır.

Quadrotorun modellenmesinde ilk olarak yapılması gereken, gövde eksen takımı ve yer eksen takımı arasındaki koordinat dönüşümlerinin elde edilmesidir. Gövde eksen takımı B ve yer eksen takımı E ile tanımlanırsa, ilgili eksenler { X_E,Y_E Z_E} ve { X_B,Y_B Z_B } olarak gösterilir.

Uzayda serbest hareket edebilen bir eksen takımının sabit yer eksen takımına göre koordinat dönüşümünü sağlayan dönüşüm matrisi sırasıyla z, y ve x eksenleri etrafındaki dönüşümü veren, $R(\Psi, z)$, $R(\theta, y)$ ve $R(\Phi, x)$ matrislerinin çarpılması ile elde edilir.

z ekseni etrafındaki rotasyon matrisi:

$$R(\Psi, z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Psi & -\sin\Psi \\ 0 & \sin\Psi & \cos\Psi \end{bmatrix} \quad (3.1)$$

y ekseni etrafındaki rotasyon matrisi:

$$R(\theta, y) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (3.2)$$

x ekseni etrafındaki rotasyon matrisi:

$$R(\Phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\Phi & -sin\Phi \\ 0 & sin\Phi & cos\Phi \end{bmatrix} \quad (3.3)$$

Euler açıları cinsinden B eksen takımının E eksen takımının etrafında sırasıyla Ψ,θ, Φ açılarıyla döndürüldüğü düşünülürse, buna göre B eksen takımındaki yönelmeler E eksen takımına R dönüşüm matrisiyle aktarılabilir. Buna göre R ile gösterilen tam dönüşüm (rotasyon) matrisi aşağıdaki şekilde elde edilir.

$$R = \begin{bmatrix} \cos\Psi\cos\theta & \cos\Psi\sin\theta\sin\Phi - \sin\Psi\cos\Phi & \cos\Psi\sin\theta\cos\Phi + \sin\Psi\sin\Phi \\ \sin\Psi\cos\theta & \sin\Psi\sin\theta\sin\Phi + \cos\Psi\cos\Phi & \sin\Psi\sin\theta\cos\Phi - \sin\Phi\cos\Psi \\ -\sin\theta & \cos\theta\sin\Phi & \cos\theta\cos\Phi \end{bmatrix} \quad (3.4)$$

Quadrotorun E düzlemine göre 3 boyutlu uzaydaki konumu ζ vektörü ile ifade edilir.

$$\zeta = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.5)$$

ζ vektörünün türevi alınarak quadrotorun E düzlemine göre olan hızı aşağıdaki gibi bulunur.

$$v = \dot{\zeta} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{3.6}$$

Hızın da türevi alınarak, quadrotorun E düzlemine göre olan doğrusal hareketinin ivmesi aşağıdaki gibi elde edilir.

$$\dot{v} = \ddot{\zeta} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \tag{3.7}$$

Quadrotor gövdesinin ağırlık merkezine dışarıdan uygulanan kuvvetler sonucunda ortaya çıkan dinamikler, Newton-Euler yöntemiyle aşağıdaki gibi bulunur [1,9].

$$F_{net} = \frac{d}{dt}[mv]_B + \omega' \times [mv]_B \tag{3.8}$$

$$M_{net} = \frac{d}{dt}[J\omega']_B + \omega' \times [J\omega']_B \tag{3.9}$$

$$\begin{bmatrix} mI_{3x3} & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \text{ x } mV \\ \omega \text{ x } J\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \tag{3.10}$$

Burada I birim matrisi, J atalet matrisini, V gövdenin doğrusal hızlar vektörünü, ω gövdenin açısal hız vektörünü, m toplam kütleyi, F araca etkiyen toplam kuvveti ve τ ise araca etkiyen toplam momenti ifade etmektedir.

Yerçekimi kuvveti (F_g) ve pervaneler tarafından üretilen toplam itme kuvveti (F_p) quadrotora etkiyen kuvvetlerin toplamına eşittir [39].

$$Fg = mR[0 \;\; 0 \;\; g]^T = mg[-\sin\theta \quad \cos\theta\sin\Phi \quad \cos\theta\cos\Phi]_B^T \tag{3.11}$$

$$F_{net} = Fg + Fp \tag{3.12}$$

Quadrotorun simetrik bir yapıda olması sonucu birim matris; x, y ve z eksenlerindeki atalet momentleri içeren köşegen matris olduğu kabul edilmiştir. x, y ve z eksenlerindeki atalet momentleri aşağıdaki gibi tanımlanır.

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (3.13)$$

Rotorların açısal hızları $\Omega_{1\text{-}4}$, düşey doğrultudaki itki parametresi b, sürüklenme parametresi d, aracın kütlesi m, yerçekimi ivmesi g ve kol uzunluğu l ile temsil edilmektedir. b ve d parametreleri sıcaklık, basınç, nem ve pervane şekli gibi etkenlere bağlıdır [41].

Rotorların ürettiği kuvvet:

$$F_i = b\Omega_i^2 \qquad i=1, 2, 3, 4 \quad (3.14)$$

Düşey doğrultudaki toplam itki kuvveti:

$$F_T = b\sum_{i=1}^{4} \Omega_i^2 \quad (3.15)$$

İtki kuvvetinin oluşturduğu ivme:

$$a_F = \frac{b}{m}\sum_{i=1}^{4} \Omega_i^2 \quad (3.16)$$

E düzlemindeki a_F ivmesinin etkisi Ra_F olmak üzere, İvme dengesi;

$$\dot{v} = (-g + Ra_f)\vec{Z} \quad (3.17)$$

Buradaki $\vec{Z} = [0\ 0\ 1]^T$ şeklinde bir vektör olup z ekseni için kullanılır.

Açısal hız vektörü:

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \quad (3.18)$$

Rotasyon matrisi ile gövde düzleminin açısal hızları ile arasındaki ilişki (3.19)'da verilmiştir.

$$\dot{R} = R.\widehat{\Omega} \quad (3.19)$$

(3.19)'da gösterilen $\widehat{\Omega}$, Açısal hız vektör matrisinin ters-simetrik çarpımından elde edilen matristir.

$$\widehat{\Omega} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \tag{3.20}$$

(3.21) de eksenlerdeki açısal momentumlar verilmiştir. Quadrotorun torku (τ_b); açısal momentumların zamana göre değişimi ile oluşur ve denklemleri (3.22) ile (3.23) de gösterilmiştir.

$$L_{x,y,z} = I\omega \tag{3.21}$$

$$\tau_b = \dot{L} = \omega \times I\dot{\omega} \tag{3.22}$$

$$\tau_b = \begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{\Phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \tag{3.23}$$

Rotorların ve aracın kendi etrafında dönmesinden dolayı oluşan tork (τ_g) :

$$\tau_g = \sum_{i=1}^{4} J(\omega \times \overrightarrow{e_z})\, \Omega_i\, (-1)^i \tag{3.24}$$

$$\tau_g = J \begin{pmatrix} \dot{\Phi} & 0 \\ \dot{\theta} & \times 0 \\ \dot{\psi} & 1 \end{pmatrix} (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) \tag{3.25}$$

x,y,z düzlemindeki torklar (τ_e):

$$\tau_e = \begin{bmatrix} lb({\Omega_4}^2 - {\Omega_2}^2) \\ lb({\Omega_3}^2 - {\Omega_1}^2) \\ d(-{\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2) \end{bmatrix} \tag{3.26}$$

Tork dengesi $\tau_e = \tau_g + \tau_b$ şeklinde olmaktadır. (3.27)

$$\tau_e = J \begin{pmatrix} \dot{\Phi} & 0 \\ \dot{\theta} & \times 0 \\ \dot{\psi} & 1 \end{pmatrix} (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{\Phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \tag{3.28}$$

Tork dengesi için (3.28) deki denklem elde edilmiş olur. Burada τ_e yerine (3.26) daki ifade yazılıp $\begin{bmatrix} \ddot{\Phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}$ ifadesi eşitlikte yalnız bırakılarak işlemler aşağıdaki şekilde devam ettirilir.

Quadrotorun açısal ivmeleri:

$$\ddot{\Phi} = \dot{\Psi}\dot{\theta}\left(\frac{I_y - I_z}{I_x}\right) + \frac{J}{I_x}\dot{\theta}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \frac{l}{I_x}(b(\Omega_4{}^2 - \Omega_2{}^2)) \quad (3.29)$$

$$\ddot{\theta} = \dot{\Psi}\dot{\Phi}\left(\frac{I_z - I_x}{I_y}\right) - \frac{J}{I_y}\dot{\Phi}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \frac{l}{I_y}(b(\Omega_3{}^2 - \Omega_1{}^2)) \quad (3.30)$$

$$\ddot{\Psi} = \dot{\theta}\dot{\Phi}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}(d(-\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 + \Omega_4{}^2)) \quad (3.31)$$

(3.17) deki denklemin genişletilmiş şekli (3.32) deki gibidir. (cos=c, sin=s ile gösteriliştir.)

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\0\\-g\end{bmatrix} + \begin{bmatrix}c\Psi c\theta & c\Psi s\theta s\Phi - s\Psi c\Phi & c\Psi s\theta c\Phi + s\Psi s\Phi\\ s\Psi c\theta & s\Psi s\theta s\Phi + c\Psi c\Phi & s\Psi s\theta c\Phi - s\Phi c\Psi\\ -s\theta & c\theta s\Phi & c\theta c\Phi\end{bmatrix}\begin{bmatrix}0\\0\\-g\end{bmatrix}\frac{b}{m}(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \quad (3.32)$$

Koordinat çerçevesi pozisyonlarındaki ivmeler;

$$\ddot{x} = (\sin\Psi\sin\Phi + \cos\Psi\sin\theta\cos\Phi)\frac{b}{m}(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \quad (3.33)$$

$$\ddot{y} = (-\cos\Psi\sin\Phi + \sin\Psi\sin\theta\cos\Phi)\frac{b}{m}(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \quad (3.34)$$

$$\ddot{z} = -g + (\cos\theta\cos\Phi)\frac{b}{m}(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \quad (3.35)$$

Giriş değişkenleri, aşağıdaki gibidir. (3.33), (3.34) ve (3.35) deki denklemleri basitleştirmek için aşağıdaki ifadeler kullanılır.

$$U_1 = b(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \quad (3.36)$$

$$U_2 = b(\Omega_4{}^2 - \Omega_2{}^2) \quad (3.37)$$

$$U_3 = b(\Omega_3{}^2 - \Omega_1{}^2) \quad (3.38)$$

$$U_4 = d(-\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 + \Omega_4{}^2) \quad (3.39)$$

$$\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (3.40)$$

Giriş değişkenlerinin matris formunda gösterimi (3-41) deki gibidir.

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ -b & 0 & b & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} {\Omega_1}^2) \\ {\Omega_2}^2) \\ {\Omega_3}^2) \\ {\Omega_4}^2) \end{bmatrix} \tag{3.41}$$

Rotor açısal hızlarının giriş değişkenlerine göre matris formunda gösterimi denklem (3.42)'de verilmiştir.

$$\begin{bmatrix} {\Omega_1}^2) \\ {\Omega_2}^2) \\ {\Omega_3}^2) \\ {\Omega_4}^2) \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ -b & 0 & b & 0 \\ -d & d & -d & d \end{bmatrix}^{-1} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{3.42}$$

Sonuç olarak, quadrotorun matematiksel parametreleri aşağıdaki gibi yazılır.

Quadrotorun açısal ivmeleri denklem (3.43), (3.44), ve (3.45)'te verilmiştir.

$$\ddot{\Phi} = \dot{\Psi}\dot{\theta}\left(\frac{I_y - I_z}{I_x}\right) + \frac{J}{I_x}\dot{\theta}\Omega_r + \frac{U_2}{I_x} \qquad \text{x ekseninin açısal ivmesi} \tag{3.43}$$

$$\ddot{\theta} = \dot{\Psi}\dot{\Phi}\left(\frac{I_z - I_x}{I_y}\right) - \frac{J}{I_y}\dot{\Phi}\Omega_r + \frac{U_3}{I_y} \qquad \text{y ekseninin açısal ivmesi} \tag{3.44}$$

$$\ddot{\Psi} = \dot{\theta}\dot{\Phi}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}U_4 \qquad \text{z ekseninin açısal ivmesi} \tag{3.45}$$

Quadrotorun hareket ivmeleri denklem (3.46), (3.47), ve (3.48)'te verilmiştir.

$$\ddot{x} = (\sin\Psi\sin\Phi + \cos\Psi\sin\theta\cos\Phi)\frac{U_1}{m} \qquad \text{x eksenindeki hareket ivmesi} \tag{3.46}$$

$$\ddot{y} = (-\cos\Psi\sin\Phi + \sin\Psi\sin\theta\cos\Phi)\frac{U_1}{m} \qquad \text{y eksenindeki hareket ivmesi} \tag{3.47}$$

$$\ddot{z} = -g + (\cos\theta\cos\Phi)\frac{U_1}{m} \qquad \text{z eksenindeki hareket ivmesi} \tag{3.48}$$

4. GÖRÜNTÜ VE GÖRÜNTÜYÜ OLUŞTURAN BİLEŞENLER

Işığın bir cisme çarparak yansıması ile görme olayı meydana gelirken, cismin optik bir cihaz kullanılarak elde edilen resmine görüntü adı verilmektedir. Görüntünün işlenebilmesi için çeşitli algoritmalarla beraber bazı programlama dilleri ve bu programlama dillerine ait kütüphaneleri kullanması gerekmektedir.

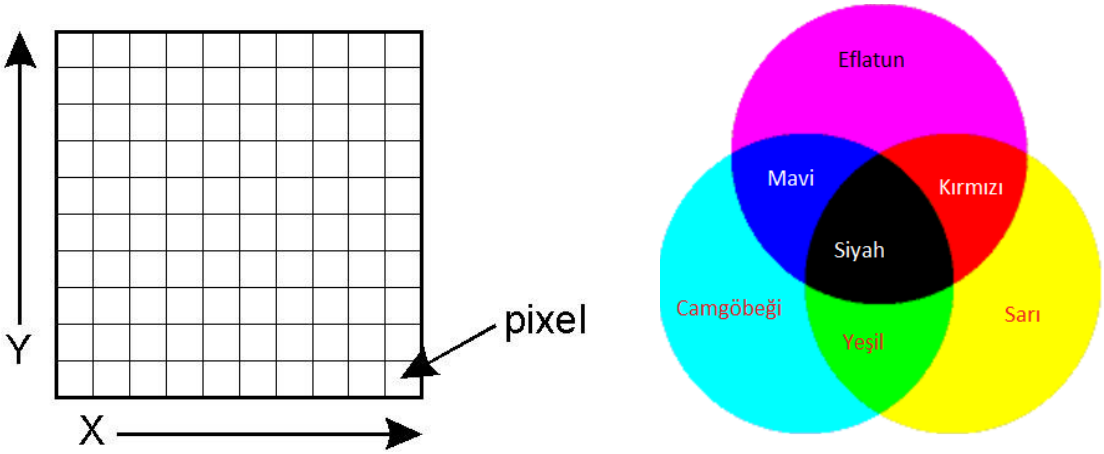
Elektronik ortamda sayısal olarak elde edilen renkli veya renksiz görüntülerin birtakım kurallar ve yöntemler kullanılarak birçok farklı amaç için elektronik ortamda filtreden geçirilme işlemine sayısal görüntü işleme denilmektedir. Ayrıca günümüzde sayısal görüntü işleme birçok farklı alanda ihtiyaç doğrultusunda kullanılmaktadır. Görüntülerin işlenmesi esnasında, gürültüden arındırma, görüntü düzeltme ve görüntü içeriğinin saptanması işlemleri yapılmaktadır. Günümüzde ise parmak izi tanıma, yüz algılama ve özellikle tıp alanında tarama işlemleri gibi gereksinim duyulan çoğu alanlarda sayısal görüntü işleme kullanılmaktadır [42].

4.1. Görüntünün Temel Bileşenleri

4.1.1. Piksel

Görüntülerin içerisinde küçük kare şeklinde noktalar bulunmaktadır. Görüntüye çok yakından bakıldığı zaman bu küçük noktalar fark edilebilmektedir. Görüntüyü oluşturan kare şeklindeki bu noktalara piksel denilir. İngilizcedeki "**Pic**ture **El**ement" kelimelerini başındaki harflerden türetilen Piksel ifadesi "Resim Parçası" anlamına gelmektedir. Görüntüyü oluşturan pikseller Şekil 4.1. (a)'da gösterilmiştir.

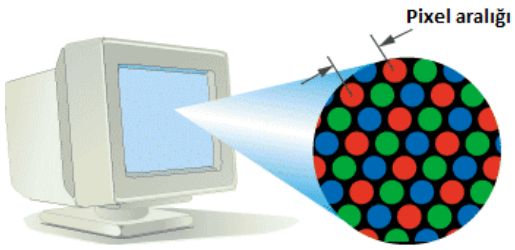
Renkli görüntünün oluşabilmesi için piksellerin içerisinde üç ana renk olan Kırmızı, Yeşil ve Mavi renkleri kullanılmaktadır. Diğer renklerin oluşumu da bu üç rengin belirli oranlarda karışımı ile gerçekleşmektedir. Görüntü teknolojilerinde kullanılan renkler Şekil 4.1. (b)'de gösterilmiştir. Dört renk kullanan piksel teknolojilerinde ise dördüncü renk olarak Turkuaz, Pembe, Sarı ve Siyah renklerini kullanmaktadır.



Şekil 4.1. a) Görüntüyü oluşturan pikseller **b)** Görüntü teknolojilerinde kullanılan renkler

4.1.2. Nokta ve Nokta Aralığı

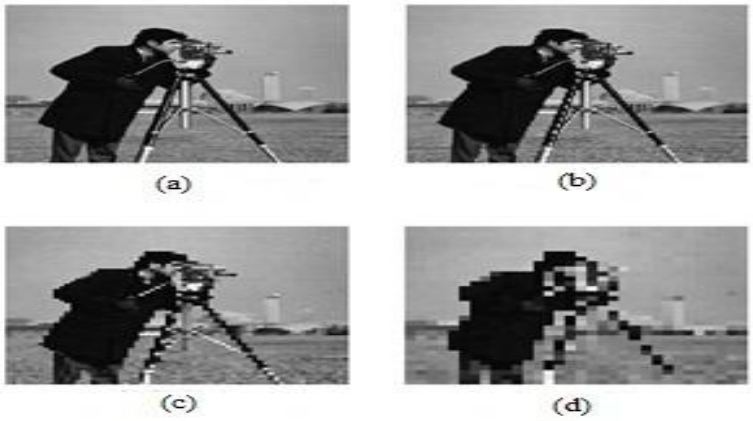
Pikseli oluşturan üç ana rengin her birine nokta (dot) denir. Bir pikseldeki noktaların birbirine olan mesafesine ise nokta aralığı (dot pitch) denilmektedir. Nokta ve nokta aralığı Şekil 4.2.'de gösterilmiştir. Görüntü kalitesi nokta aralığı mesafesine bağlıdır. Görüntü kalitesinin yüksek olabilmesi için nokta aralığının az olması gerekmektedir.



Şekil 4.2. Nokta ve nokta aralığı

4.1.3. Çözünürlük

Ekranda görüntülenebilen piksel sayısına çözünürlük denilmektedir. Çözünürlük değeri, sütunda ve satırda bulunan piksel sayılarının çarpılmasıyla bulunmaktadır. 800 x 600, 1024 x 768, 1280 x 768, 1366 x768 gibi bazı çözünürlük değerleri örnek gösterilebilir (Şekil 4.3.). Çözünürlük değeri arttıkça, görüntü kalitesi de artmaktadır.



Şekil 4.3. Piksel sayısı ile çözünürlük ilişkisi
(a)256 x 256 piksel, **(b)**128 x128 piksel, **(c)**64 x 64 piksel, **(d)**2 x32 piksel görüntü

4.1.4. Piksel Yoğunluğu

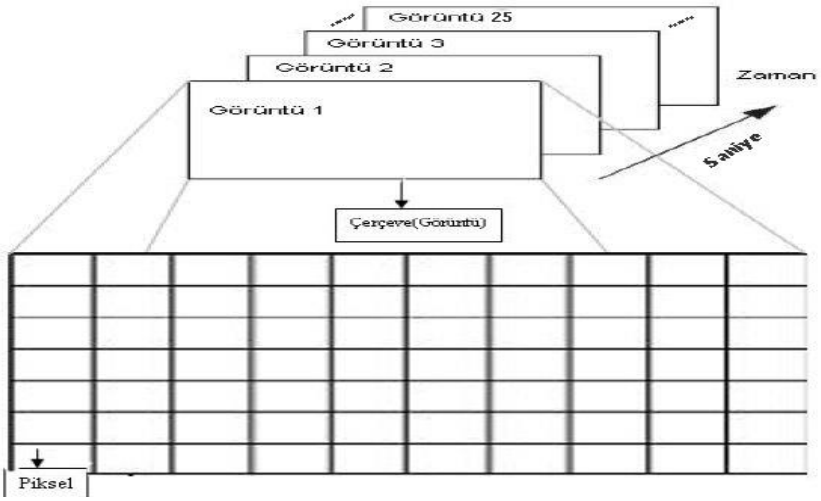
Piksel yoğunluğu (Rezolasyon), bir inç kare içerisinde mevcut olan piksel sayısını belirtmektedir. Şekil 4.4.'te farklı rezolasyon değerlerine sahip görüntüler gösterilmiştir.

49 piksel 34 piksel 24 piksel

Şekil 4.4. Farklı rezolasyon değerlerine sahip görüntüler

4.1.5. Çerçeve

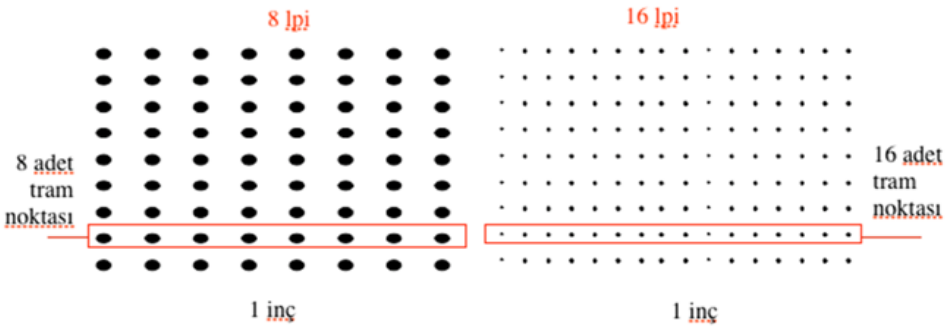
Gerçek zamanlı görüntü veya video, bir saniyede arka arkaya çekilen 25 adet sayısal olarak derlenmiş görüntünün bir araya gelmesiyle oluşmaktadır. Videoyu meydana getiren her bir görüntüye ise çerçeve veya frame adı verilir (Şekil 4.5.) [31].



Şekil 4.5. Çerçeve (frame)

4.1.6. İnç Başına Düşen Çizgi Sayısı

Renkli veya ara tonlara sahip siyah beyaz baskıda ara ton oluşturmak için kullanılan küçük noktacıklara tram denir. Baskıdaki renk yoğunluğu tramların sıklığı ile kontrol edilir. İnç başına düşen çizgi sayısı (Line Per Inch – LPI) görüntüdeki bir inç başına düşen çizgi sayısını (Şekil 4.6.) ifade etmektedir. Burada çizgi ile bahsedilmek istenen tramdır.



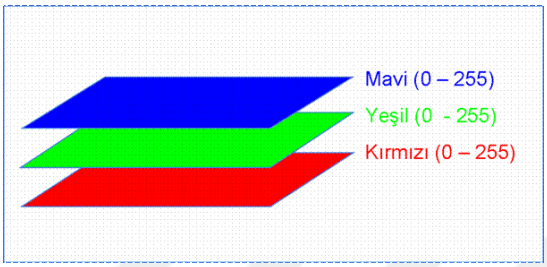
Şekil 4.6. LPI (inç başına düşen çizgi sayısı)

4.1.7. İnç Kareye Düşen Piksel Sayısı

Yazıcı veya çizici cihazlarda gerçekleştirilen baskılarda bir inç karede basılan piksel (Dot Per Inch –DPI) sayısını ifade etmektedir [42]. DPI değeri arttıkça basılan görüntünün kalitesi de artmaktadır.

4.1.8. Renk Kanalları

Renk kanalları; görüntünün renkli bileşenlerini temsil eden gri tonlamalı görüntüleri ifade temektedir [43]. Renkli görüntüler, bu renk kanallarının bir araya gelmesiyle elde edilebilmektedir. Örnek olarak RGB görüntüsünün kırmızı, yeşil ve mavi renk değerleri için ayrı kanalları vardır (Şekil 4.7.) [42].

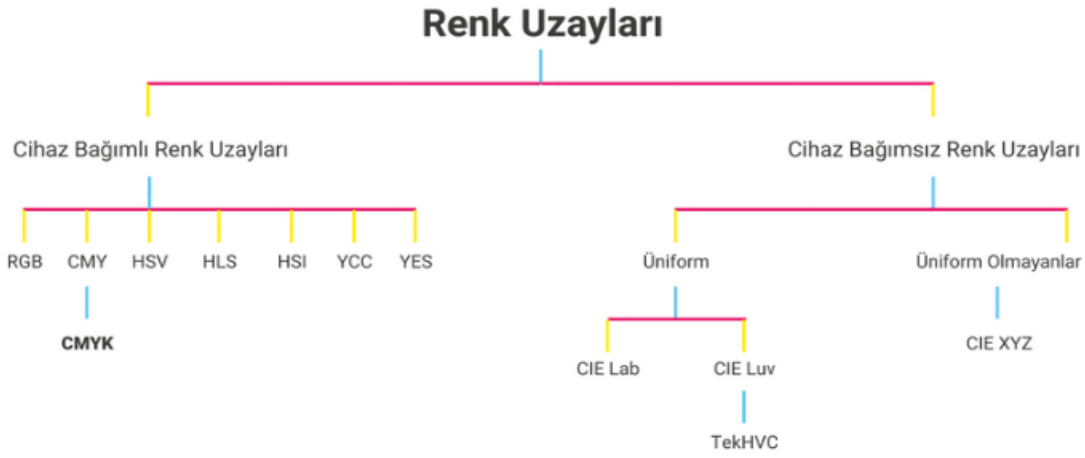


Şekil 4.7. Renk kanalları

4.1.9. Renk Modelleri

Renk modeli, dijital görüntülerde görülen renkleri tanımlar. RGB, CMYK ve HSB renk modelleri bulunmaktadır. Her renk modeli rengin tanımlanmasında farklı birer yöntemi temsil etmektedir [43].

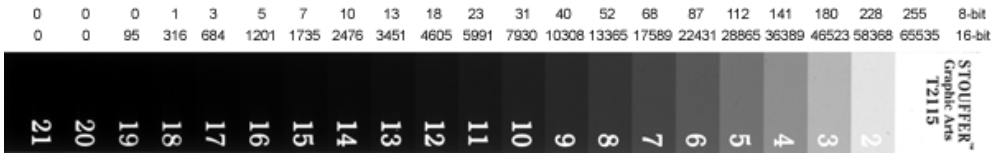
Renk çeşitliliğinin fazla olmasından dolayı renklerin gruplara ayrılmasına ihtiyaç duyulmuştur. Renklerin gruplara ayrılması gerçekleştirilerek renk uzayları (Şekil 4.8.) meydana getirilmiştir. Renk kümelerini tanımlamak için her renk uzayının kendine özgü bir yapısı vardır. Örneğin ekran, çizici gibi cihazların kendilerine uygun renk uzayı vardır ve sadece o aralıktaki renkleri üretebilirler. Her cihazın kendine has bir renk uzayı olduğundan dolayı görüntü de cihazdan cihaza farklılıklar gösterebilmektedir [44].



Şekil 4.8. Renk Uzayları

4.1.10. Gri Tonlamalı Renk Uzayı

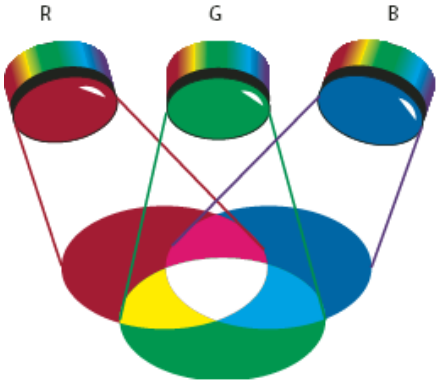
Gri rengin 256 tonunun mevcut olduğu renk uzayıdır. Her bir pikselin parlaklık değeri 0 ile 255 arasında değişmektedir. Siyah renk 0 ile Beyaz renk 255 ile ifade edilmektedir. 0 ile 255 arasındaki değerler ise grinin siyahtan beyaz renge doğru olan değişik tonlarını temsil etmektedir (Şekil 4.9.). Gri tonlamalı renk uzayı tek bir kanalı ifade etmekte ve grinin tonları ışık yoğunluğuna göre ayarlanmaktadır.



Şekil 4.9. Gri tonları (0 – 255)

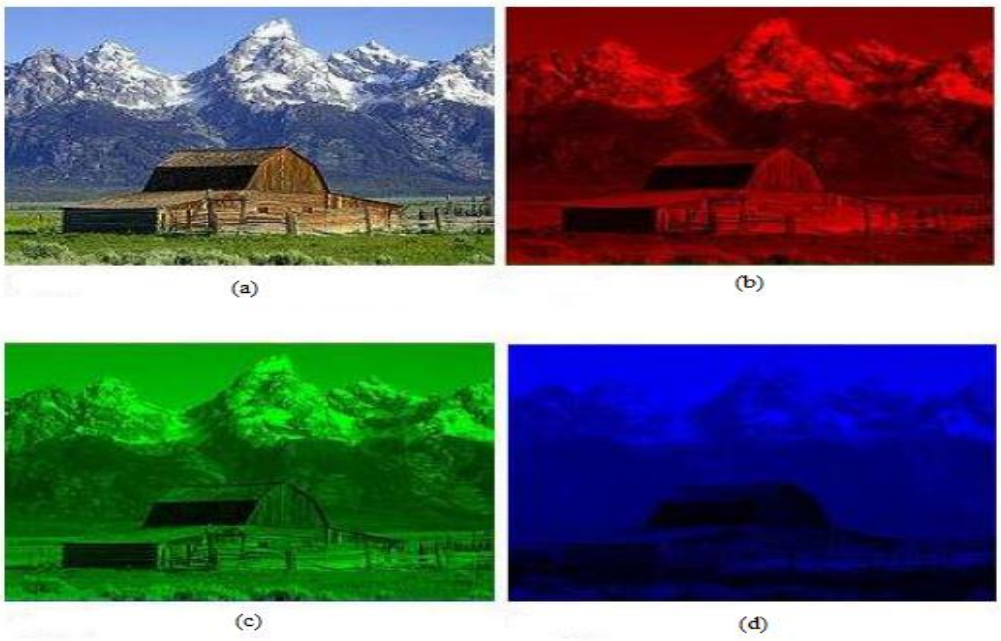
4.1.11. Kırmızı - Yeşil - Mavi Renk Uzayı

Kırmızı, Yeşil ve Mavi renkleri ifade etmek için RGB kısaltması kullanılmaktadır. Kırmızı (Red) renk için “R”, Yeşil (Green) renk için “G” ve Mavi (Blue) renk için “B” harfleri kullanılmaktadır (Şekil 4.10.) [45]. Bu renk uzayında diğer tüm renkler kırmızı, yeşil ve mavinin farklı oranlarda karışımından elde edilebilmektedir.



Şekil 4.10. RGB renk düzeni

RGB renk uzayında beyaz rengin elde edilmesi kırmızı, yeşil ve mavi renklerin %100 oranında karışımıyla sağlanmaktadır. Siyah rengin elde edilmesi ise üç rengin %0 oranında karışımıyla sağlanmaktadır. Kırmızı, yeşil ve mavi renklerin farklı oranlarda karışımında ise farklı renkler elde edilmektedir. Şekil 4.11.'de kırmızı, yeşil ve mavi renk tonlarının kullanımına ait örnekler gösterilmiştir.



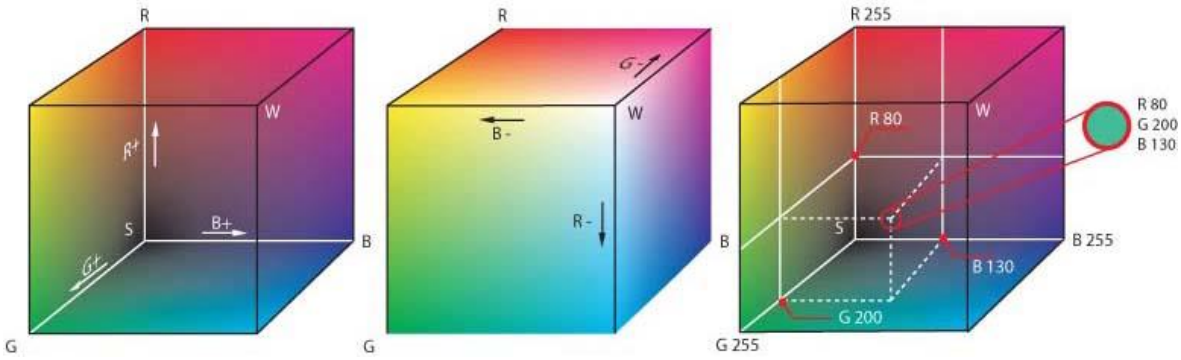
Şekil 4.11. RGB renk uzayı (a) Kırmızı, yeşil ve mavi renk tonlarının karışımı ile elde edilmiş orijinal görüntü. (b) Kırmızı renk tonları ile elde edilmiş görüntü. (c) Yeşil renk tonları ile elde edilmiş görüntü. (d) Mavi renk tonları ile elde edilmiş görüntü

Tablo 4.1.'deki çizelgede farklı renklerin elde edilebilmesi için gereken RGB karışım oranları gösterilmiştir.

Tablo 4.1. Bazı renklerin elde edilebilmesi için gerekli olan Red-Green-Blue karışım oranları.

	Aralık	Beyaz	Sarı	Turkuaz	Yeşil	Eflatun	Kırmızı	Mavi	Siyah
R	0- 255	255	255	0	0	255	255	0	0
G	0- 255	255	255	255	255	0	0	0	0
B	0- 255	255	0	255	0	255	0	255	0

RGB renk uzayı Şekil 4.12.'de gösterildiği gibi 3 boyutlu bir uzay olarak düşünülebilir. Koordinat eksenleri kırmızı, yeşil ve mavi renkten oluşmaktadır. Elde edilmek istenilen renkler kırmızı, yeşil ve mavi renklerinin koordinatları şeklinde ifade edilmektedir [46].

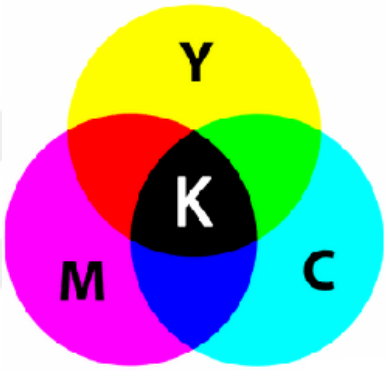


Şekil 4.12. RGB renk uzayı koordinat ekseni

RGB renk uzayında çalışmak yani burada bir görüntüyü işlemek oldukça zor ve yavaş bir şekilde gerçekleşmektedir. Bir görüntünün renk yoğunluğu değiştirilebilmek için önce görüntüden kırmızı, yeşil ve mavi renk yoğunluklarının okunması gerekmektedir. Daha sonra görüntünün renk yoğunluğu üzerinde değişiklik yapılabilmektedir. Görüntünün yoğunluk ve renk biçimlerine farklı bir renk uzayı kullanarak daha hızlı ve kolay erişilebilinir. Böylelikle bir görüntüyü işlemek daha hızlı şekilde gerçekleşecektir. Bunun için iki farklı renk sinyali ve parlaklık özelliğini kullanan renk uzayları geliştirilmiştir. Bu renk standartları HSV, YCbCr, YIQ ve YUV renk uzaylarıdır.

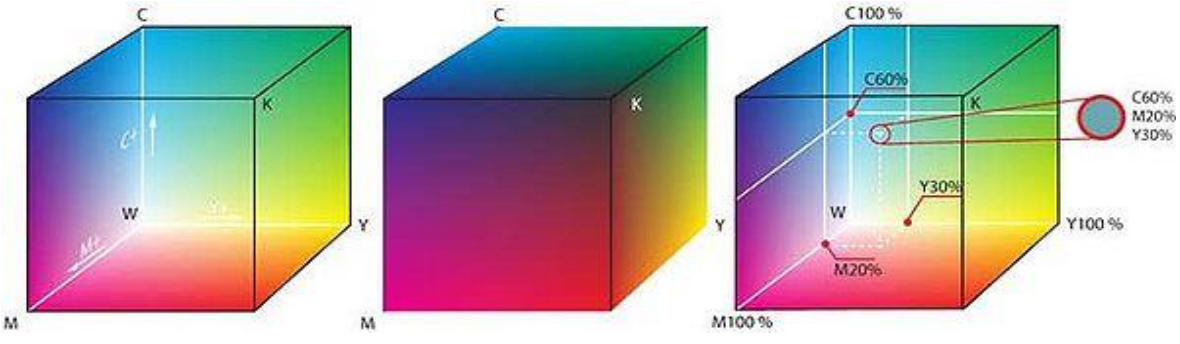
4.1.12. Turkuaz – Pembe – Sarı – Siyah Renk Uzayı

Turkuaz, Pembe, Sarı ve Siyah (Cyan, Magenta, Yellow, Key - CMYK) renklerinin oluşturmuş olduğu renk uzayıdır. CMYK'daki "K" Key anlamına gelmektedir. Anahtar renk olarak ifade edilmektedir ve siyah rengi temsil etmektedir. Tüm renkler Turkuaz, Pembe, Sarı ve Siyah renklerin karışımından elde edilir. CMYK renk uzayı renkli baskı üretiminde kullanılır [38]. CMYK yansımanın, RGB ise ışığın rengidir. CMYK'da boş zemin olarak beyaz renk kullanılırken RGB'de boş zemin olarak siyah renk kullanılmaktadır. Şekil 4.13.'te CMYK renk düzeni gösterilmiştir.



Şekil 4.13. CMYK renk düzeni

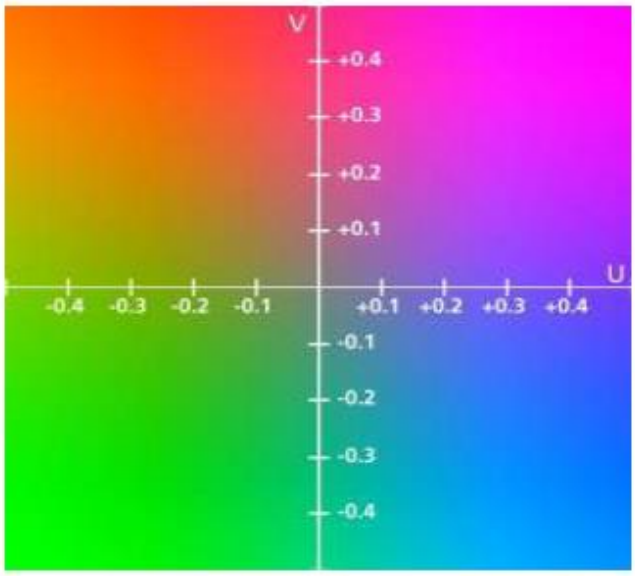
Temelde renk sayısı 3 olan CMY ye siyah renk sonradan eklenmiştir. Siyah rengin elde edilmesi teorikte üç rengin %100 oranlarının da karıştırılmasıyla elde edilmesi gerekirken, uygulamalarda bu sonuca tam olarak ulaşılmadığından ve karışım işleminin meydana getirdiği maliyetten dolayı siyah renk sonradan eklenmiştir [42]. Şekil 4.14.'te CMYK renk uzayı koordinat ekseni gösterilmiştir.



Şekil 4.14. CMYK renk uzayı koordinat ekseni

4.1.13. Parlaklık – Mavi Tabanlı – Kırmızı Tabanlı Renk Uzayı

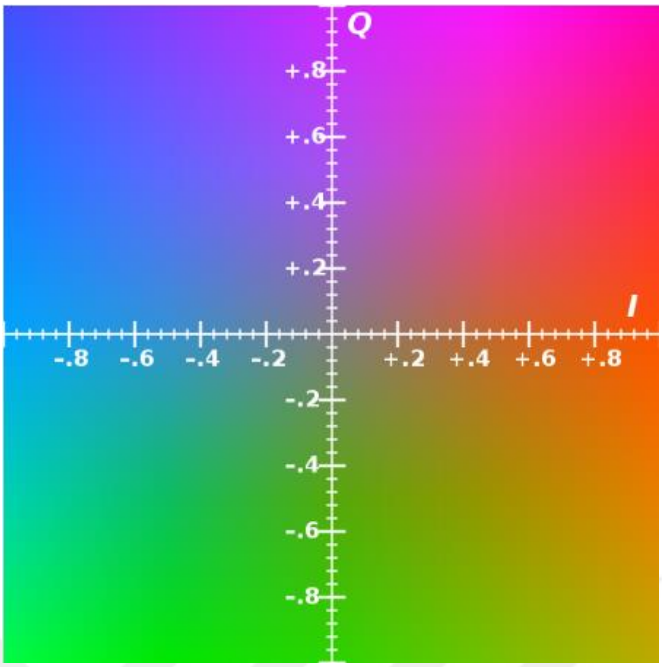
Parlaklık – Mavi Tabanlı – Kırmızı Tabanlı (Luminance - Chrominancel1 - Chrominancel2) renk uzayının kısaltması YUV'dir. YUV renk uzayında, Y siyah – beyaz yani parlaklık bilgisini temsil etmekte, U mavi tabanlı renklilik ve V ise kırmızı tabanlı renkliliği temsil etmektedir (Şekil 4.15.). Renkler bu üç kavram ile ifade edilerek oluşturulurlar. YUV renk uzayı PAL, NTSC ve SECAM kompozit renkli video standartlarında kullanılmaktadır. Bu renk uzayı ile siyah-beyaz alıcılar, renkli video sinyallerini siyah-beyaz şeklinde gösterebilmektedir [47].



Şekil 4.15. YUV renk uzayı bileşenleri.

4.1.14. Parlaklık – Turuncu, Mavi –Mor, Yeşil Renk Uzayı

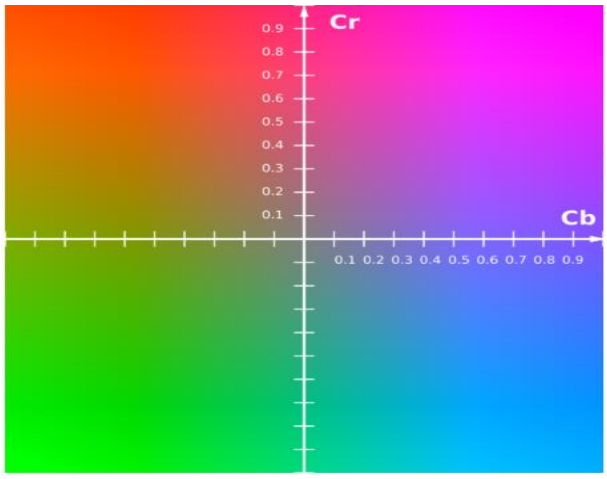
Parlaklık – Turuncu, Mavi –Mor, Yeşil (Luminance – Orange, Blue – Purple, Green) renk uzayının kısaltması YIQ'dur. YIQ, TV yayınlarında, geriye doğru uyumluluğu sağlamak ve iletim verimliliğini arttırmak için RGB' ye göre yeniden kodlanmıştır. Burada Y parlaklık değerini göstermekte I ve Q değerleri ise renk değerlerini ifade etmektedir [48]. Şekil 4.16.'da YIQ renk uzayı bileşenleri gösterilmiştir.



Şekil 4.16. YIQ renk uzayı bileşenleri.

4.1.15. Parlaklık – Mavi Renklilik – Kırmızı Renklilik Renk Uzayı

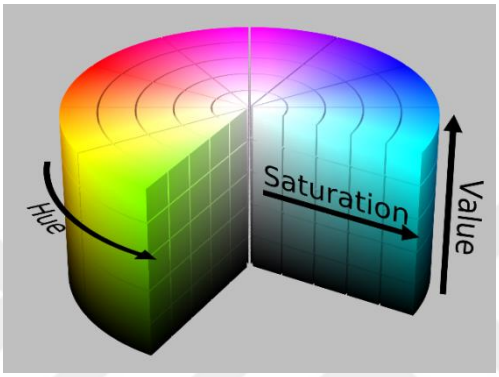
Parlaklık – Mavi Renklilik – Kırmızı Renklilik (Luminance – Chrominance blue – Chrominance red) renk uzayının kısaltması YCbCr'dir. YCbCr, parlaklık (luminance) sinyalini Y ile renk bilgilerini Cb (Chrominance blue) ve Cr (Chrominance red) ile saklayan renk uzayıdır (Şekil 4.17.). YCbCr renk uzayı, sayısal video standardını dünya çapında oluşturma işlemleri sırasında ortaya çıkmıştır. YCbCr renk uzayında Y, 8 bitlik 16 – 235 arasında tanımlanmaktadır. Cb ve Cr bileşenleri ise 16 – 240 arasında tanımlanmaktadır [49].



Şekil 4.17. YCbCr renk uzayı.

4.1.16. Renk Tonu – Doygunluk – Aydınlık Renk Uzayı

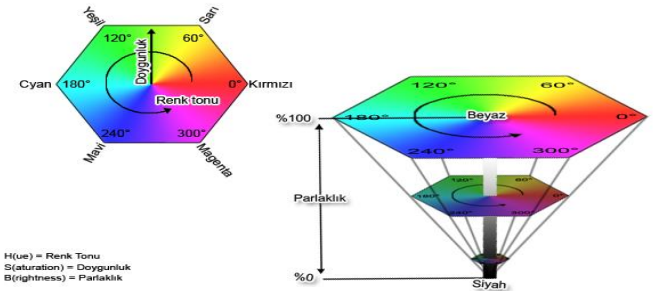
Renk tonu, Doygunluk, Aydınlık (Hue, Saturation, Value - HSV) renk uzayı, renkleri, renk tonuna, doygunluğuna ve aydınlık değerlerine göre belirten renk uzayıdır (Şekil 4.18.) [42]. Doygunluk ile rengin canlılığı belirlenirken aydınlık ile rengin siyah-beyaz arası istenen değeri seçilmektedir. HSV uzayında siyah renk elde etmek için aydınlık 0 olması, beyaz renk elde etmek için ise aydınlık değerinin 255 olması gerekmektedir.



Şekil 4.18. HSV renk uzayı

4.1.17. Renk Tonu – Doygunluk – Parlaklık Renk Uzayı

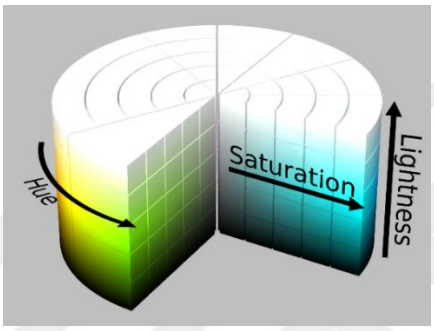
Renk Tonu, Doygunluk, Parlaklık (Hue, Saturation, Brightness - HSB) renk uzayı; renklerin insan tarafından algılanan üç temel özelliğini ifade eder. Bu üç temel özellik renklerin tonları, doygunluğu ve parlaklığıdır. HSB renk uzayında renkler rakamlarla ifade edilmektedir (Şekil 4.19.). Her özellik 0 ile 255 arasında değerler alır. Üç değerin 255 olduğu durumda beyaz, üç değerin 0 olduğu durumda ise siyah renk elde edilmektedir [50].



Şekil 4.19. HSB renk uzayı

4.1.18. Renk Tonu – Aydınlık – Doygunluk Renk Uzayı

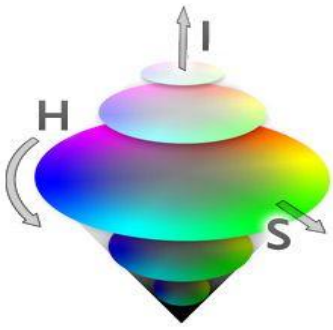
Renk Tonu, Aydınlık, Doygunluk (Hue, Lightness, Saturation - HLS) renk uzayı, HSV renk uzayı ile benzerlik göstermektedir. HLS uzayında parlaklık değeri yerine aydınlık değeri kullanılmıştır. Aydınlık değerinin %50 olması durumunda nötr renkler elde edilirken, aydınlık değerinin düşürülmesiyle renklerin daha koyu olması, aydınlık değerinin yükseltilmesiyle renklerin daha açık olması sağlanmış olur. Şekil 4.20.'de HLS renk uzayı gösterilmiştir.



Şekil 4.20. HLS renk uzayının gösterimi

4.1.19. Renk Tonu – Doygunluk – Yoğunluk Renk Uzayı

Renk Tonu, Doygunluk, Yoğunluk (Hue, Saturation, Intensity - HSI) renk uzayı, HLS renk uzayına benzemektedir. HSI renk uzayındaki yoğunluk bileşeni, HLS renk uzayındaki parlaklık bileşeninin yerine kullanılmıştır. Bu renk uzayında Hue (H) baskın renk dalga boyunu, Saturation (S) saf rengin beyaz ışık ile hangi oranda karıştığını ve Intensity (I) ışık miktarını göstermektedir (Şekil 4.21.).



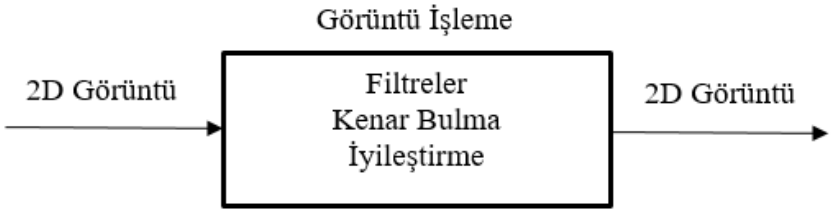
Şekil 4.21. HSI renk uzayının gösterimi

5. GÖRÜNTÜ İŞLEME VE NESNE TAKİP YÖNTEMLERİ

İnsanın sahip olduğu en önemli duyulardan birisi görme duyusudur. Gözümüz görünen nesneleri ayırt edebilmekte ve tanımamızı sağlamaktadır. Gözden ilham alınarak gelişen teknolojiyle birlikte üretilen cihazlarda görüntü işleme teknikleri kullanılarak görüntüden nesnelerin ayırt edilmesi ve şekillendirilmesi üzerine birçok işlem gerçekleştirilebilmektedir. Yapılan bu çalışmalar özellikle robotik, askeri, tıp, güvenlik, vb. alanlarda insanlığa fayda sağlamaktadır.

5.1. Görüntü İşleme

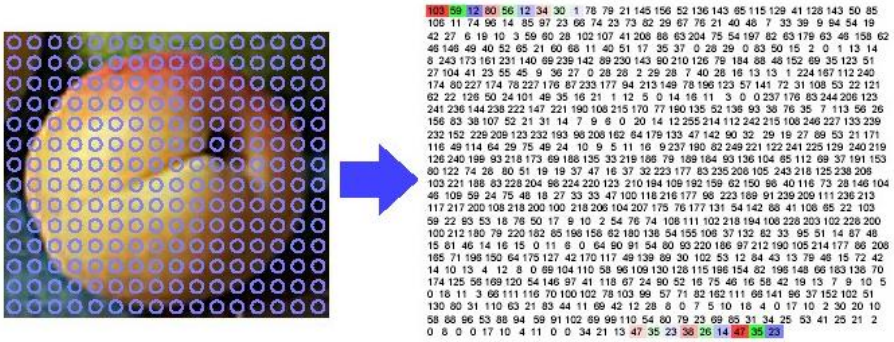
Bir görüntünün sayısal hale dönüştürülmesi ve gelişmiş bir görüntünün elde edilebilmesi ya da görüntüden yararlı bilgiler çıkarılabilmesi için yapılan işlemler (Şekil 5.1.) görüntü işleme olarak tanımlanmaktadır [51].



Şekil 5.1. Görüntü işleme

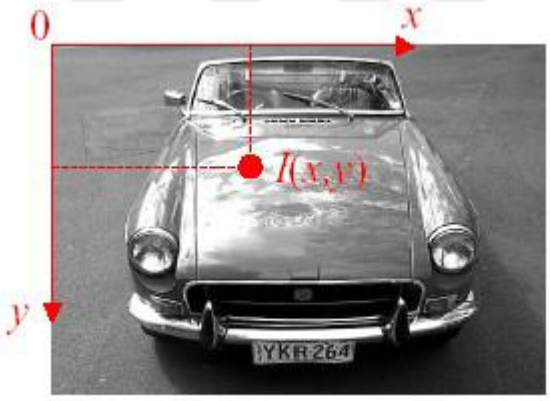
5.1.1. Sayısallaştırılmış Görüntü

Görüntü işlemenin yapılabilmesi için öncelikle görüntünün bilgisayarın anlayacağı şekle çevrilmesi gerekmektedir. Bu çevirme işlemi sonucu elde edilen görüntüye sayısallaştırılmış görüntü denilmektedir. Görüntü 256 farklı renk yoğunluğundan oluşur. 0 beyazı, 255-n diğer renkleri temsil eder. Her piksel bir sayısal değere karşılık gelir. Şekil 5.2.'de her bir pikselin almış olduğu renk kodları gösterilmektedir.



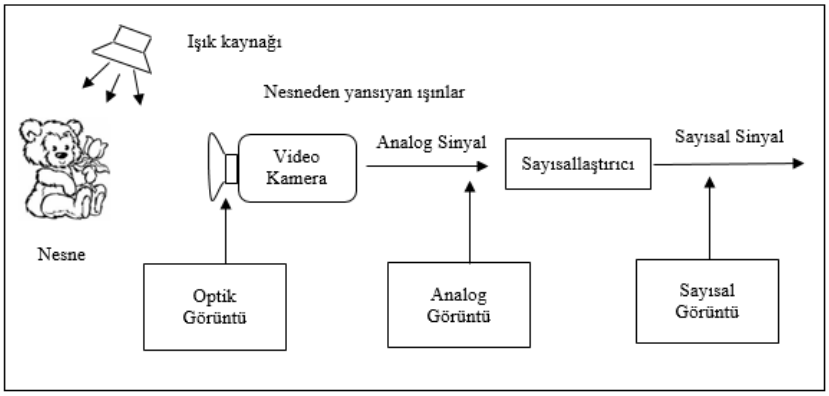
Şekil 5.2. Gerçek görüntünün sayısala dönüştürülmüş hali

Şekil 5.3.'te gösterildiği gibi görüntü, iki değişkenin bir fonksiyonu olarak tanımlanır. *I(x,y)* gibi bir fonksiyonda x ve y koordinatları gösterir. *I* ise görüntüye ait bir özellik fonksiyonudur. Görüntü bölgelerden oluşur. Görüntü işlenirken seçili bölgelere bazı operasyonlar yapılmaktadır. Örneğin görüntünün bir bölümü bulanıklaştırılırken, diğer bölümü parlaklaştırılmaktadır.



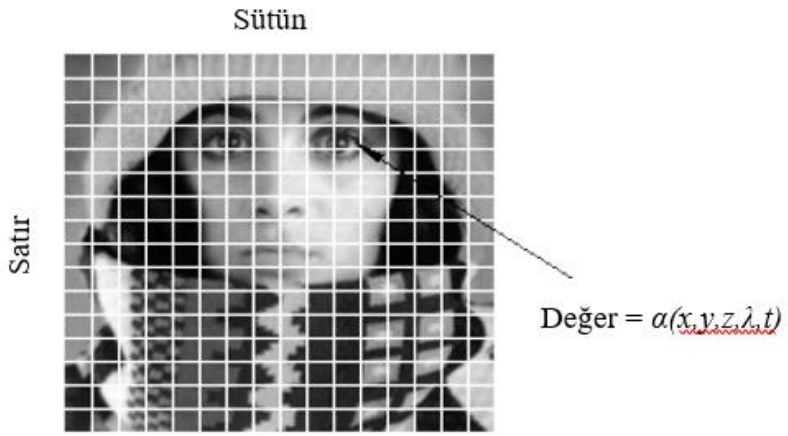
Şekil 5.3. Sayısal görüntü fonksiyonu

Sayısal görüntü 1 ve 0' lardan oluşmaktadır. Sayısal görüntünün oluşabilmesi için görüntünün örnekleme ve nicemleme safhalarından geçirilmesi gerekir [52]. Örnekleme ile görüntünün belirli zamanlarından örnekler alınmaktadır. Nicemleme de ise genlik seviyelerinin belirli değerleri alması sağlanmaktadır. Analog görüntünün sayısallaştırılması işlemi Şekil 5.4'te gösterilmiştir.



Şekil 5.4. Analog görüntünün sayısallaştırılması [52].

l(*x*, *y*) fonksiyonu örneklenerek 2 boyutlu a[m,n] matrisi elde edilir. m sütun ve n satırlarının kesiştiği her bölgeye piksel denir. Pikseldeki "z" değeri derinliği, "λ" değeri rengi ve "t" değeri zamanı gösteren bir fonksiyondur. Şekil 5.5'te sayısal bir görüntüde piksellerin aldığı değerler gösterilmektedir [51]. Her pikselin parlaklık değerini gösteren tam sayı değeri o pikselin değerini gösterir.



Şekil 5.5. Sayısallaştırılmış görüntüde koordinatlar [51].

5.2. Görüntü İşleme Aşamaları

Sayısallaştırılmış görüntü üzerinde işlem yapılabilmesi için Histogram Eşitleme, Eşikleme ve Kenar Çıkarma aşamalarının görüntü üzerine uygulanması gerçekleştirilmektedir [53].

5.2.1. Histogram Eşitleme

Görüntü Histogramı, bir görüntünün yoğunluk dağılımının grafiksel bir gösterimidir. Histogram eşitleme ise görüntüdeki renk değerlerinin belirli bir alanda kümelenmesinden dolayı meydana gelen renk dağılımı bozukluğunu giderebilmek için kullanılan bir yöntemdir. Görüntünün daha belirgin hele getirilmesi amacıyla görüntü işlemede histogramlar kullanılmaktadır [54]. Bir görüntünün kontrastını ve parlaklığını arttırabilmek amacıyla histogram eşitleme işlemi yapılmaktadır [55]. Histogram eşitleme;

$$Sk = T(rk) = \sum_{j=0}^{k} \frac{n_j}{n} (L-1) \qquad (5.1)$$

Formülü ile ifade edilir. Bu formülde:

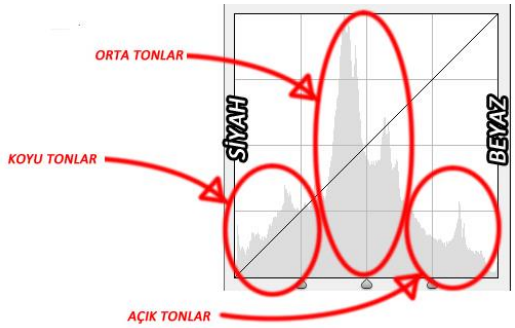
k = 0, 1, 2, …, L-1

L = Gri seviyelerin sayısı

n_j = j gri pikselin değeri

n = Toplam piksel sayısı olarak kullanılmaktadır [56].

Şekil 5.6.'da gösterilen histogram grafiğinde sağ taraf beyaz rengi sol taraf ise siyah rengi temsil etmektedir. Şekil 5.7.'de histogram eşitleme için örnek uygulama gösterilmiştir.



Şekil 5.6. Histogram grafiğinde beyaz – siyah arası tonlar

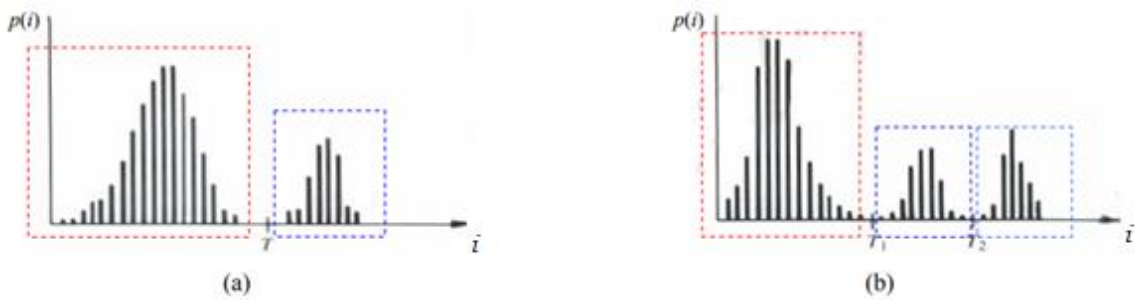
(a) (b)

Şekil 5.7. Histogram eşitleme için örnek uygulama
(a) Orijinal resim, (b) Histogram eşitlenmiş resim

5.2.2. Eşikleme

Bir görüntüyü her biri içerisinde farklı özelliklerin tutulduğu anlamlı bölgelere ayırma işlemine bölütleme denilmektedir. Bu özellikler örneğin, görüntü içerisindeki benzer parlaklıklar olabilir ve bu parlaklıklar ilgili görüntünün farklı bölgelerindeki nesneleri temsil edebilmektedir.

Görüntü arka planından, görüntü içerisindeki nesneleri ayırabilmek eşikleme işleminin amacıdır. Eşikleme işlemi için, görüntüdeki gri seviye dağılımlarını gösteren görüntü histogramından faydalanılmaktadır. Örneğin, koyu bir arka plan üzerinde açık renkli nesnelerden oluşan f(i, j) görüntüsüne ilişkin gri seviye histogramı Şekil 5.8.(a)'daki biçime sahip olacaktır. Bu histograma göre, nesnelere ve arka plana ait pikseller olmak üzere, görüntüyü iki ana grupta değerlendirmek mümkündür. Bu durumda nesneleri arka plandan ayırmak için en kolay yol, histogramdan göreceli olarak belirlenen bir T eşik değeri ile görüntüdeki piksel değerlerini karşılaştırmak olacaktır. Buna göre, görüntüdeki herhangi bir (i, j) pikseli için; $f(i, j) > T$ ise (i, j) pikseli nesneye ait bir nokta, $f(i, j) \leq T$ ise (i, j) pikseli arka plana ait bir nokta olacaktır. Diğer taraftan, görüntüye ilişkin histogram Şekil 5.8.(b)'deki gibi ikisi nesneye biri de arka plana ait olmak üzere üç gri seviye grubundan oluşabilir. Buna göre görüntüdeki herhangi bir (i, j) pikseli için; $T1 < f(i, j) \leq T2$ aralığındaki pikseller bir nesneye, $f(i, j) > T2$ aralığındaki pikseller diğer bir nesneye ve $f(i, j) \leq T1$ aralığındaki pikseller de görüntü arka planına karşı düşecektir [56].



Şekil 5.8. Tek bir eşik değeri ve birden çok eşik değeri ile bölmelenen gri seviye histogram biçimleri

Eşiklemenin temel ifadesi;

$$G(i,j) = 1 \qquad f(i,j) \geq T \text{ için}$$

$$G(i,j) = 0 \qquad f(i,j) < T \text{ için} \tag{5.2}$$

Burada:

T → Eşik değeri

G(i,j) = 1 → Objenin görüntü elemanları

G(i,j) = 0 → Objenin görüntü elemanlarıdır.

Başarılı bir sonuç için doğru bir eşik değeri seçimi çok önemlidir. Bunun için birçok eşik değeri belirleme algoritmaları kullanılmaktadır. Görüntü için tek bir eşik değeri kullanımına global eşikleme denilmektedir [57]. Bölütleme eşik değeri bölgesel görüntü karakteristiklerinin bir fonksiyonu olarak görüntü üzerinde değiştiğinde, değişken eşiker kullanılarak bu durumlara çözüm üretilir. Bu bölütleme de adaptif bölütleme olarak adlandırılır [58]. Şekil 5.9.'da eşikleme uygulaması gösterilmiştir.

(a) (b)

Şekil 5.9. Eşikleme uygulaması
(a) Orjinal resim, (b) Global eşikleme uygulanmış resim

5.2.3. Kenar Çıkarma

Kenar çıkarma işlemi, görüntüdeki renk geçişlerini keskinleştirerek objelerin elde edilmesini sağlamaktadır. Görüntü kenarlarının çıkarılması, resimlerin anlamlandırılmasında çok önemlidir ve resimlere ait özelliklerin, bilgilerin ortaya çıkmasını sağlamaktadır [59]. Kenarlar genellikle nesnelerin gölge geçişlerine ve sınırlarına uyarlanmaktadır [60].

Kenar, yönü ve büyüklüğü veren bir vektör olmakla birlikte pikselin bir bölgesinin ve yakın komşularının özelliğini taşımaktadır. Kenar hesaplamalarında çok açık renkli resimler, kullanılmaktadır. Resmin "Gradient" fonksiyonu kenarları hesaplar. Gri rengin geçişlerini anlamlı bir şekilde ölçmemizi kenar çıkarma işlemi sağlamaktadır [61].

Farklı algoritmalar veya filtreler kullanılarak görüntüdeki kenarlar bulunabilmektedir. Bu algoritma veya filtrelerle resmin üzerindeki piksel değerleri yeniden hesaplanmaktadır. Hesaplama işlemi genel olarak Denklem (5.3)'deki gibi ifade edilir:

$$\sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i,j) x\, f(x-i, y-j) \tag{5.3}$$

Denklem (5.3)'deki h fonksiyonu görüntüye uygulanmakta olan filtredir [51]. Filtre işleminde sonra yeni tek bir değer elde edilmektedir. Elde edilen bu değer resimde filtrenin uygulandığı kısımdaki merkez piksele atanmaktadır. Filtre işlemi yapılırken, resmin ilk ve son satır ve sütunlarına uygulanmayacak şekilde işlem yapılmalıdır [62].

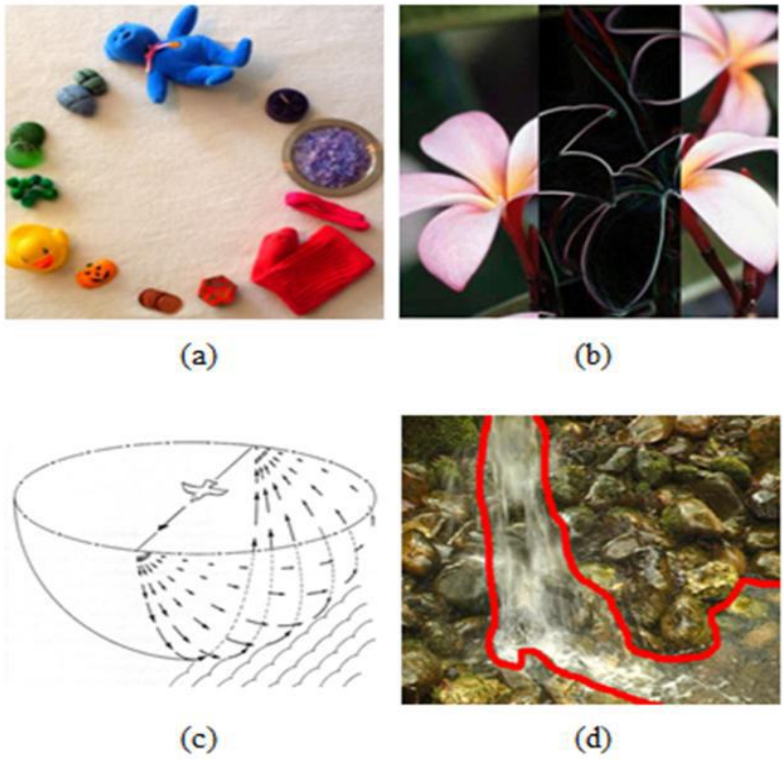
Filtrenin boyutu ve filtredeki katsayıların değeri yapılan işlemde önemlidir. Filtre boyutu genellikle 3 x 3 matris şeklinde seçilmektedir. Filtrenin boyutunun artması daha fazla komşu değerin hesaplanmasına sebep olmaktadır. Bu hesaplama işlemleri de işlem yükünü arttırmakla birlikte ince detayların kaybolmasına sebep olmaktadır. Bundan dolayı büyük boyutlu filtreler tercih edilmemektedir. Kenar çıkarmada kullanılan algoritma ya da filtre türlerinden bazıları Prewit, Sobel, Log, Roberts, Laplace ve Canny olarak bilinmektedir.

5.3. Nesne Takip Yöntemleri

Nesne takip uygulamalarında kameradan gelen veriler içerisinde takip edilecek nesnenin yakalanmasının gerçekleştirilmesi yapılması gereken ilk işlemdir. Takip edilecek nesnenin ayırt edilebilmesinde nesnenin özelliklerinin seçimi etkin rol oynamaktadır. Seçilecek görüntü içerisinde nesnenin ayırt edici özelliklerinin doğru belirlenmesi, nesne takip işleminin başarısını arttırmaktadır [63].

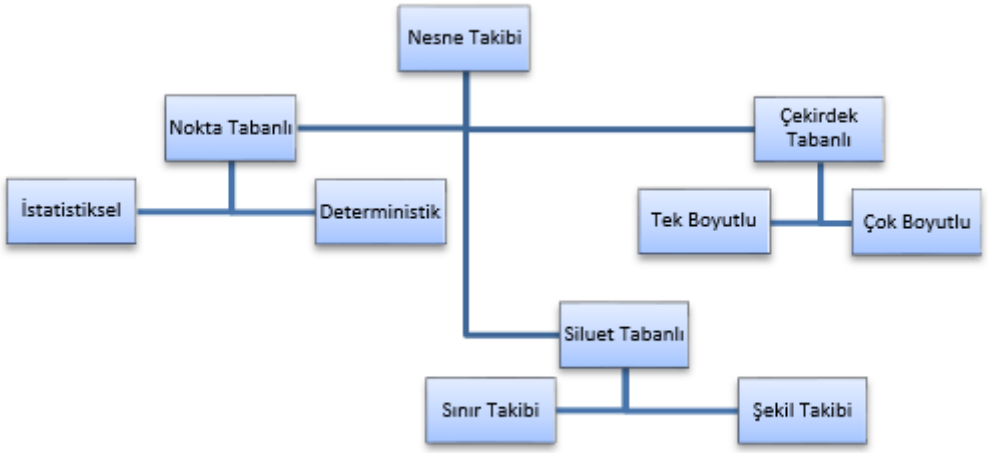
Takip edilecek nesnenin ayırt edilebilmesinde kullanılabilecek genel nesne özellikleri şunlardır. (Şekil 5.10.)

- Renk özelliği
- Kenar özelliği
- Optik akış özelliği
- Doku özelliği



Şekil 5.10. Genel nesne özellikleri **(a)** Renk özelliği **(b)** Kenar özelliği **(c)** Optik akış özelliği **(d)** Doku özelliği.

Genel özellikleri bakımından nesne takip yöntemleri, nokta tabanlı, çekirdek tabanlı ve siluet tabanlı olmak üzere üç başlık altında gruplandırılmaktadır [17]. Şekil 5.11.'de nesne takip yöntemlerinin sınıflandırılması gösterilmiştir.



Şekil 5.11. Nesne takip yöntemlerinin sınıflandırılması

5.3.1. Nokta Tabanlı Nesne Takibi

Nokta tabanlı nesne takip yöntemi nesne yakalama ve veri bağı olmak üzere iki aşamadan meydana gelmektedir [64]. Nesne yakalama aşmasında her bir nesne tek bir nokta ile ifade edilmektedir. Başlangıç pozisyon bilgisi kullanılarak nesnelerin nokta ile ifade edilebilmesi sağlanmaktadır. Bu yöntemde ilk görüntüde tespit edilen noktalar ile son görüntüde takip edilen nesneye ait noktalar arasındaki veri bağı ilişkisinin doğru bir şekilde oluşturulması sağlanmaktadır.

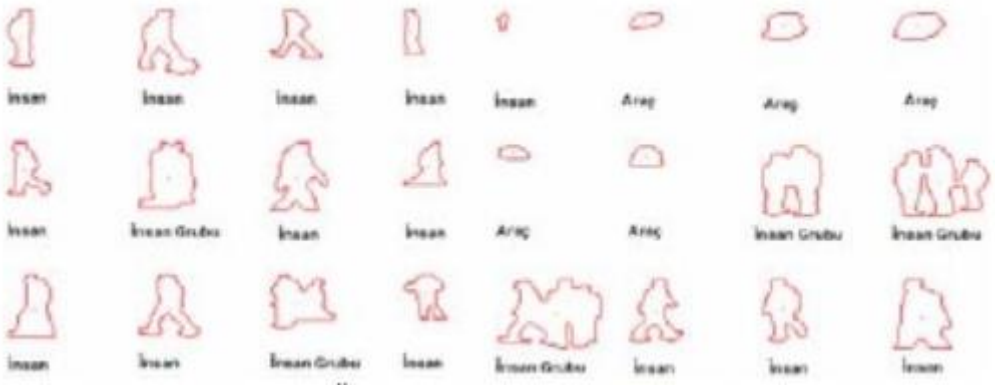
5.3.2. Çekirdek Tabanlı Nesne Takibi

Çekirdek tabanlı nesne takip yönteminde, takip edilecek nesne basit bir geometrik şekil içerisine alındıktan sonra görünüm bilgisinin olasılık yoğunluk dağılımı elde edilmektedir. Yoğunluk dağılımı ardı ardına gelen video görüntüleri boyunca incelenmektedir. Olasılık yoğunluk dağılımı, takip edilecek nesnenin her pikselinin diğer pikseller üzerindeki etkisini ifade etmektedir [65, 66].

Çekirdek tabanlı nesne izleyiciler, parametrik olmayan tahmin ediciler gurubu içerisinde yer almaktadır. Parametrik olmayan sistemlerde sabit bir fonksiyon yapısı söz konusu değildir ve bir tahmin gerçekleşeceği zaman dağılıma ait tüm veriler göz önünde bulundurulur. Ancak, parametrik sistemlerde sabit bir fonksiyon yapısı ve parametre değerleri bulunmaktadır [67].

5.3.3. Siluet Tabanlı Nesne Takibi

Daire veya dikdörtgen gibi basit geometrik şekiller ile ifade edilemeyen, karmaşık geometrik şekillere sahip nesnelerin takibi için siluet tabanlı yöntemlerin kullanılması kolaylık sağlamaktadır. Siluet tabanlı nesne takipte yapılan işlem önceden kullanılarak üretilen nesne modelini güncel resimler içerisinde bulabilmektir [68, 69]. Nesnenin renk histogramı, nesne sınırı veya kenarı nesne modeli olarak seçilebilir. Şekil takibi ve sınır takip ediciler olarak iki kısımda incelenmektedir. Şekil takibi yaklaşım, alınan görüntü içerisinde Şekil 5.12.'deki gibi örnek siluet şablon veri tabanındaki nesne siluetlerini aramaktadır. Sınır takip yaklaşım ise başlangıç aşamasında belirlenen nesne sınırları kullanılarak, bazı enerji fonksiyonlarının düşük seviyeye indirilmesine bağlı olarak nesnenin son görüntüdeki yeni yerini belirlemeye çalışır. İki yöntemde görüntüdeki geçici bir alana, nesne bölütleme işlemi uygulanmaktadır [70-72].

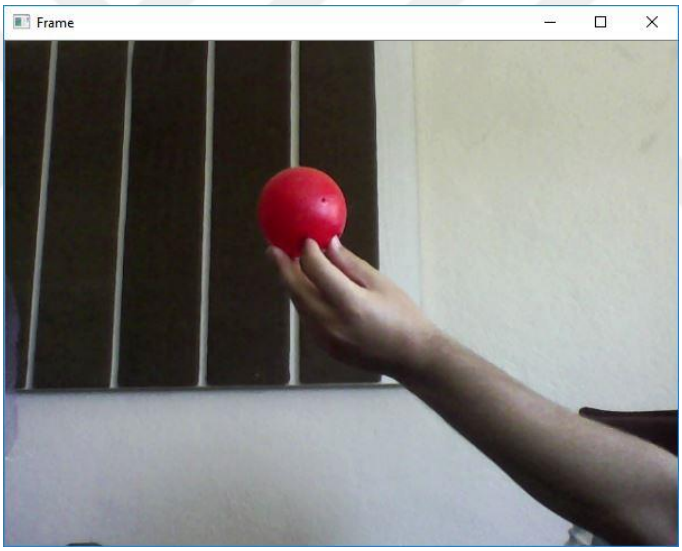


Şekil 5.12. Örnek siluet şablon veri tabanı

6. YAPILAN ÇALIŞMALAR VE UYGULAMALAR

6.1. Renk ile Nesne Takibinin Yapılması

Tez çalışmasında yapılan uygulamada ilk önce sabit kameradan alınan görüntülerden özelliği belirtilen hareketli nesnenin tespitinin yapılması sağlanmış ve bu nesnenin takip edilmesi gerçekleştirilmiştir. Şekil 6.1.'de kırmızı renkli nesnenin görüntüde algılanmadığı Şekil 6.3.'teki görüntüde ise hazırlanan yazılım sayesinde başarılı bir şekilde algılandığı gözlemlenmiştir. Bu algılama işleminde kameradan alınan görüntülerden kırmızı renkli ve hareketli nesnenin varlığı Python ve OpenCV'de renk özelliği ve nokta tabanlı nesne takip yöntemi kullanılarak tespit edilmiştir.

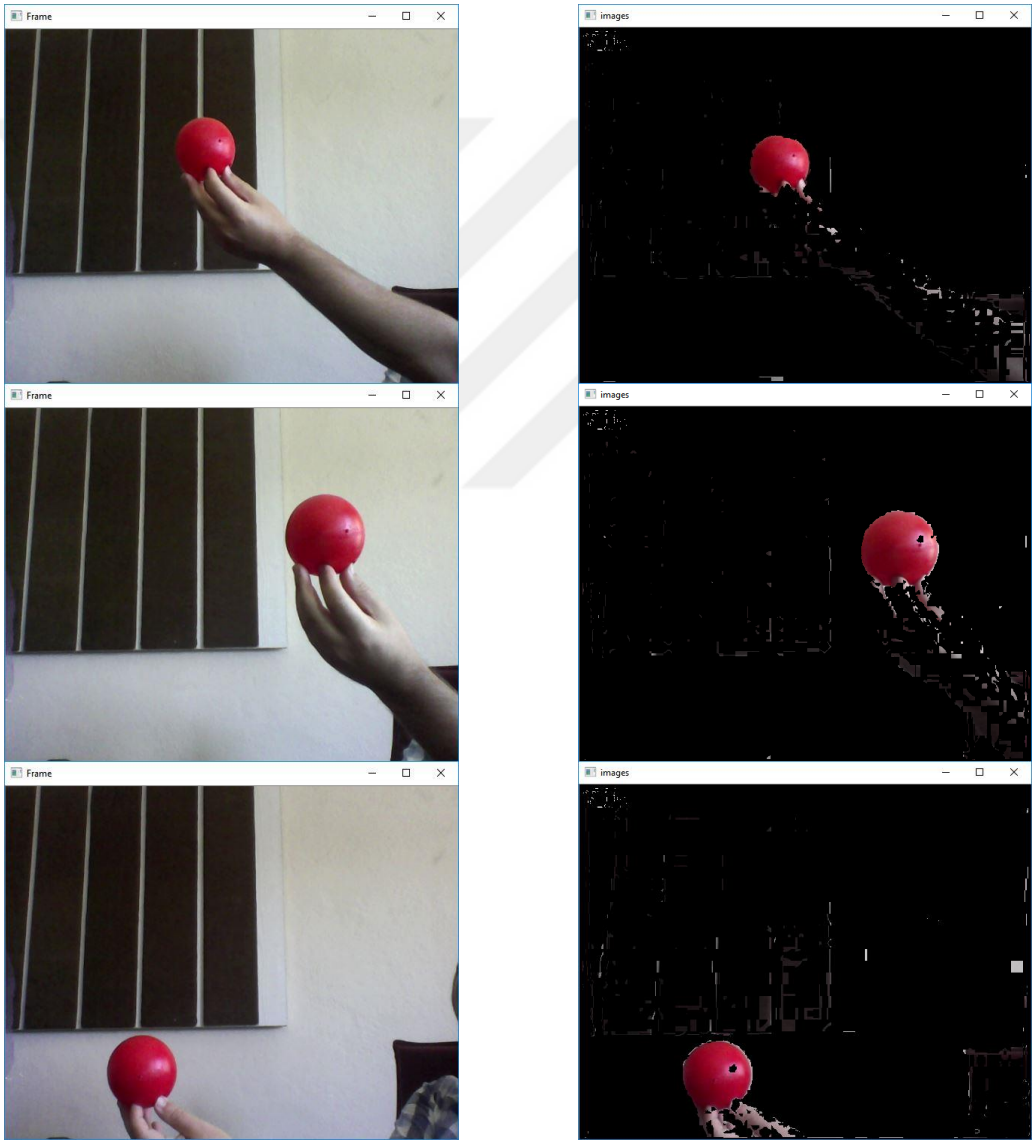


Şekil 6.1. Kırmızı renkli hareketli nesnenin tespit edilmediği durum

Hareketli nesnenin algılanabilmesi için OpenCV için gerekli olan paketlerin programa aktarılması sağlanmıştır. Bu paketlerden "deque" paketi ile veri yapısının hızlı bir şekilde eklenmesi sağlanmakta ve geçmiş verilerin bir listesi tutulmaktadır. Kameradan gelen görüntülerde hareketli nesnenin koordinatları (x,y) olarak belirtilmektedir. Bu hareketin devamlılığı süresince her (x,y) noktası bir veri oluşturmakta ve bu verilerin kaydı "deque" paketi ile yapılmaktadır. Buradaki veriler kullanılarak nesnenin hareket rotası çıkarılmakta ve nokta tabanlı nesne takip işlemi gerçekleştirilmektedir.

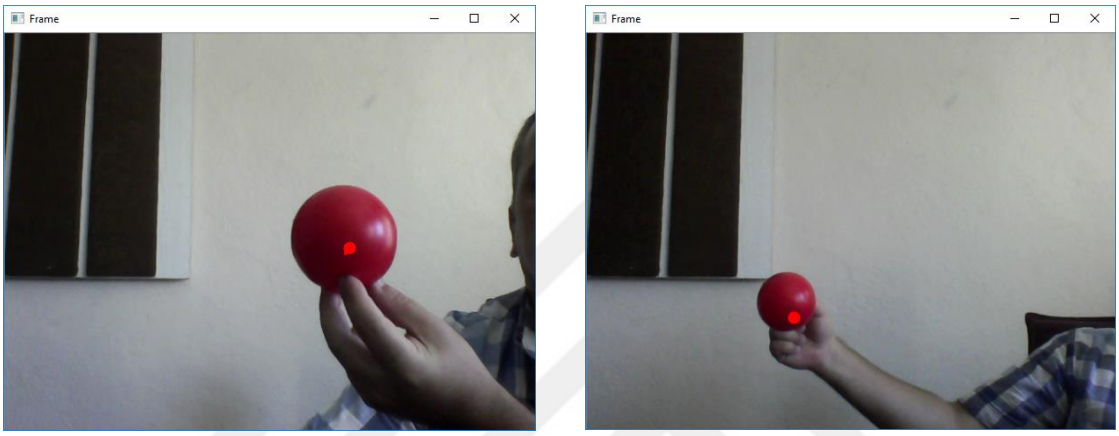
Python'da OpenCV ile döndürme, yeniden boyutlandırma, iskeletleştirme ve Matplotlib görüntülerini görüntüleme gibi temel görüntü işleme işlemlerini yapmak için kullanılan paketlerden biri olan "imutils" programa eklenmiştir. İmutils'un paketinin bilgisayara kurulumu cmd ekranına "Pip install imutils" komutu yazılarak sağlanmıştır.

Renk tabanlı nesne takibinin yapılabilmesi için sabit kameradan alınan görüntülerin HSV renk uzayına çevrilmesi Şekil 6.2'deki gibi gerçekleştirilmiştir.



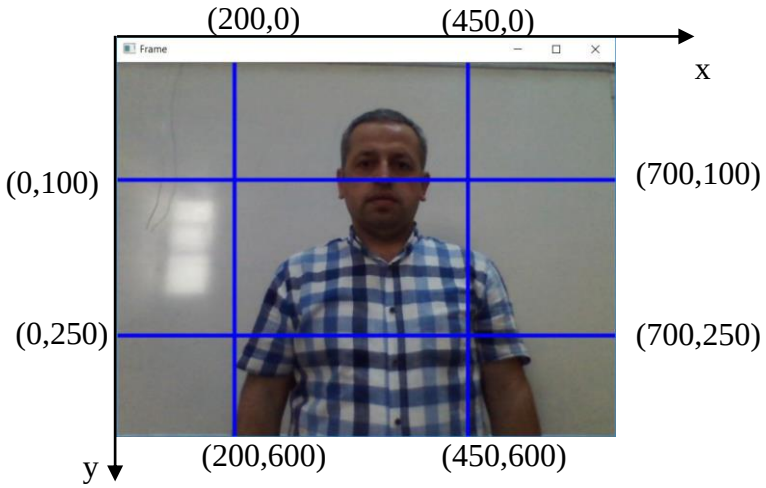
Şekil 6.2. Görüntünün RGB renk uzayından HSV renk uzayına dönüştürülmesi

Renk tabanlı nesne takibi yapmak için kullanılan renk kırmızıdır. HSV renk uzayındaki kırmızı rengin alt ve üst sınırları, imutils kütüphanesindeki aralık dedektörü (range detector) scripti kullanarak belirlenmiştir. Bu renk sınırları ile kameradan alınan görüntülerdeki kırmızı renkli nesnenin algılanması sağlanmıştır. Burada belirlenecek renk isteğe bağlı olarak HSV uzayındaki aralığa bağlı olarak değiştirilebilmektedir.

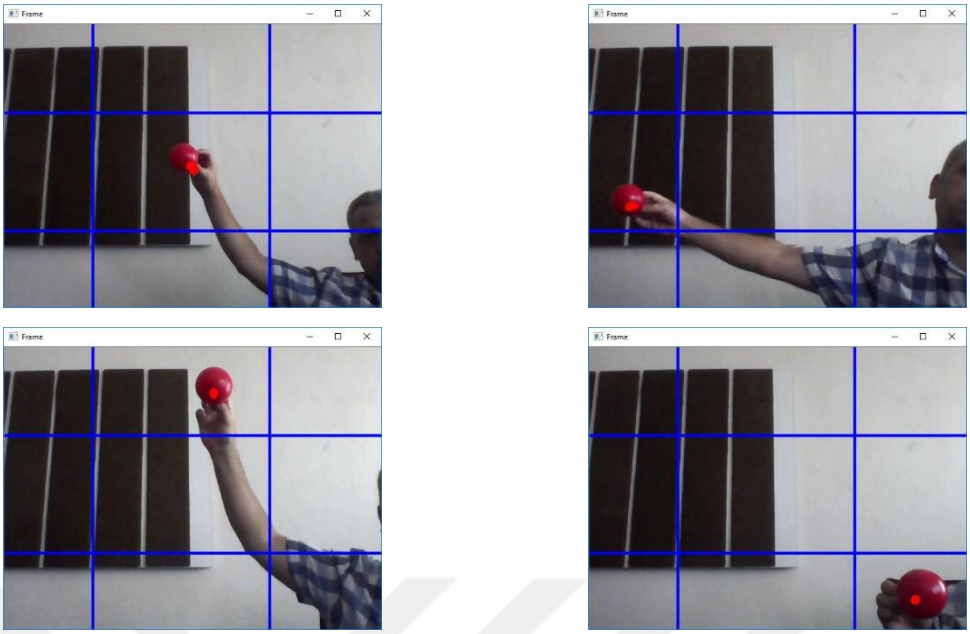


Şekil 6.3. Kırmızı renkli hareketli nesnenin tespit edildiği durum

Tespit edilen hareketli nesnenin takibinin sağlanabilmesi için görüntü ekranı Şekil 6.4.'teki gibi x ve y deki belirtilen kısımlardan çizilen çizgilerle dokuz bölüme ayrılmıştır. Elde edilen sonuçlar Şekil 6.5.'te gösterilmiştir.



Şekil 6.4. Ekran görüntüsünün dokuz parçaya ayrılması



Şekil 6.5. Kırmızı renkli hareketli nesnenin ekranın farklı bölümlerinde elde edilen takip sonuçları

6.2. Şekil ile Nesne Takibinin Yapılması

Görüntü işlemede görüntüdeki şekillerin tespit edilmesi, sıkça gerekli olan bilgilerden birisidir. Python da OpenCV görüntü işleme teknikleri kullanılarak yapılan Şekil 6.6.'daki örnek çalışma sonucu görüntüdeki şekillerin kenar çizgileri ve orta noktaları tespit edilmiştir. Bu yapılan çalışma birçok uygulamada işimizi kolaylaştırabilir ve pratik bir şekilde bizi çözüme götürebilmektedir.

Kod kısmında yapılan işlemler sırasıyla;

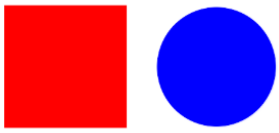
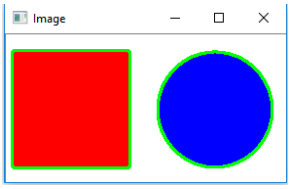
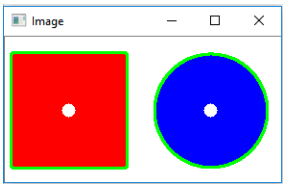
- Görüntünün yakalanması,
- Gri görüntüye çevrilmesi,
- Sınırların tespit işlemi için görüntünün bulanıklaştırılması işlemleri gerçekleştirilmektedir.

t.py | sekil.py* | tezudpland.py | tezkod.py | tezkod1.py | deneme2.py

```
import imutils
import cv2
image = cv2.imread("sekiller.png")
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (5, 5), 0)

width = image.shape[0]
height = image.shape[1]
thresh = cv2.threshold(blurred,35,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)[1]
cnts = cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)[1]
for c in cnts:
    M = cv2.moments(c)
    area = cv2.contourArea(c)
    print(area, width*height)
    if area < (width-1)*(height-1):
        cX = int(M["m10"] / M["m00"])
        cY = int(M["m01"] / M["m00"])

        cv2.drawContours(image, [c], -1, (0, 255, 0), 2)
        cv2.circle(image, (cX, cY), 7, (255, 255, 255), -1)

cv2.imshow("Image", image)
cv2.waitKey(0)
```

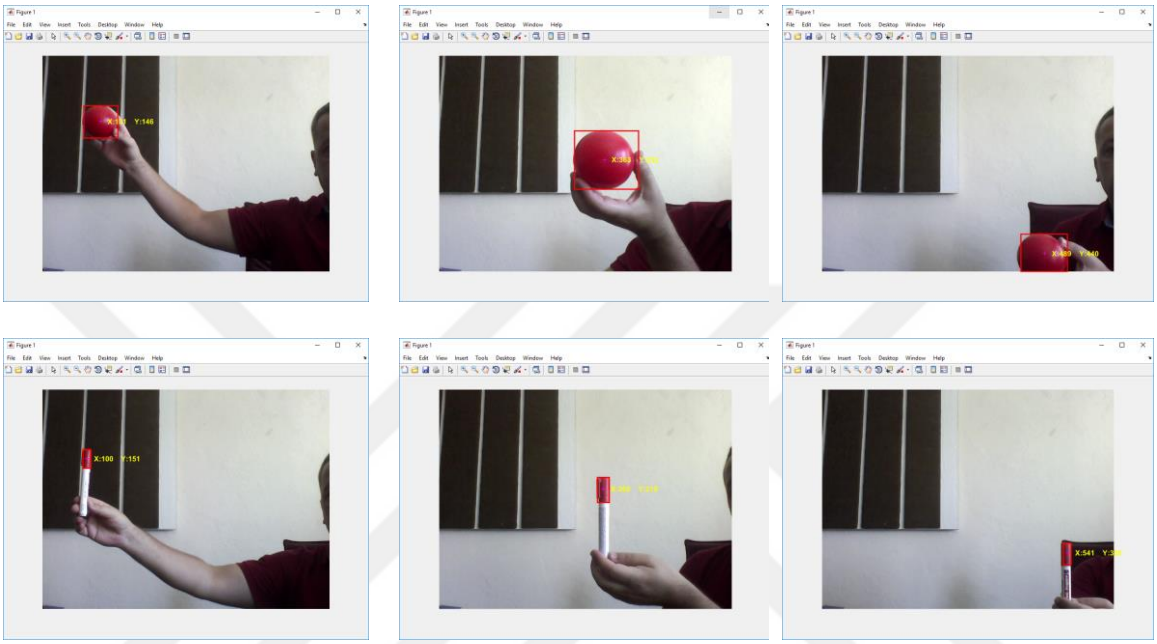
Şekil 6.6. Şekil tespitine ait Phyton kodu

Yukarıdaki programda kullanılan "cv2.moments()" komutu ile şekillere ait gereken bilgiler elde edilmektedir. Her şeklin kenar çizgileri "cv2.drawContours()" ile çizdirilmiştir. Şeklin orta noktası da benzer şekilde "cv2.circle()" komutuyla gerekli yerlerde çizilmesini sağlanmıştır. Böylelikle şekillerin orta noktası ve sınır çizgileri çizilmiş yeni görüntü elde edilmiştir. Yapılan işlemlere ait sonuçlar Şekil 6.7.'de gösterilmiştir.

		
Orjinal görüntü	Gri görüntü	Bulanık görüntü
		
Eşikleme yapılmış görüntü	Sınırların bulunması	Orta noktaların bulunması

Şekil 6.7. Şekil tespiti uygulama sonuçları

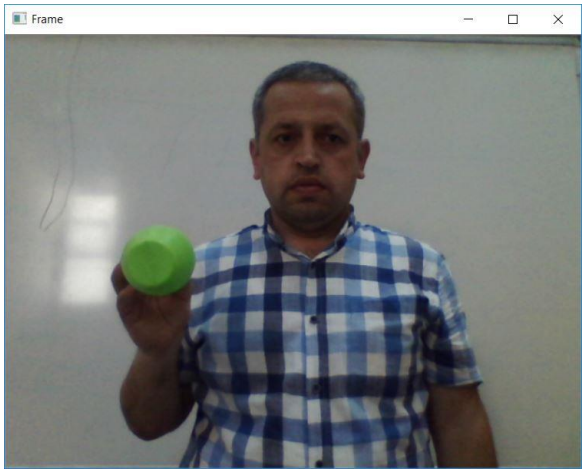
Benzer şekilde Matlab programında da hareketli nesnenin (x,y) koordinat bilgileri algılanarak etrafına geometrik bir şekil çizdirilmesi sağlanarak şekil tabanlı takip işlemi gerçekleştirilebilmektedir. Şekil tabanlı takip için Matlab'da gerçekleştirilen işlemlere ait sonuçlar Şekil 6.8.'de gösterilmiştir.



Şekil 6.8. Matlab'da yapılan şekil tespiti ve takibi uygulama sonuçları

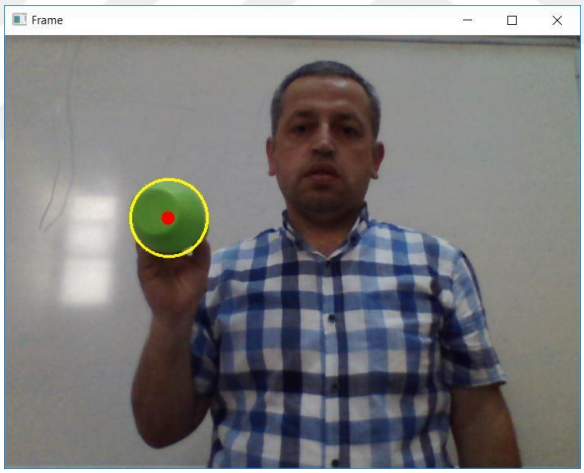
6.3. Şekil ve Renk Özelliği Kullanılarak Nesne Takibinin Yapılması

Tez çalışmasının bu bölümünde yapılan uygulamada sabit kameradan alınan görüntülerden renk özelliği belirtilen hareketli nesnenin tespiti yapılarak bir çember içerisine alınması sağlanmış ve bu nesnenin takip edilmesi gerçekleştirilmiştir. Şekil 6.9.'da yeşil renkli nesnenin görüntüde algılanmadığı Şekil 6.10.'daki görüntüde ise hazırlanan yazılım sayesinde başarılı bir şekilde algılandığı gözlemlenmiştir. Bu algılama işleminde kameradan alınan görüntülerden yeşil renkli ve hareketli nesnenin varlığı OpenCV'de renk özelliği kullanılarak nokta tabanlı nesne takip yöntemi kullanılarak tespit edilmiştir. Ayrıca algılanan nesnenin çekirdek tabanlı nesne takip yöntemine uygun olarak bir çember içerisine alınması sağlanmış ve takip işlemi gerçekleştirilmiştir.



Şekil 6.9. Yeşil renkli hareketli nesnenin tespit edilmediği durum

Algılanan renkli nesnenin merkez noktasının renk yoğunluğunun yarıçap (Radius) değeri 10 pikselden büyük olduğu durumlarda etrafına çember çizdirilmiştir. Çemberin yarıçap değeri nesnenin ileri, geri hareketini takip edebilmek amacıyla kullanılmıştır.



Şekil 6.10. Yeşil renkli hareketli nesnenin tespit edildiği durum

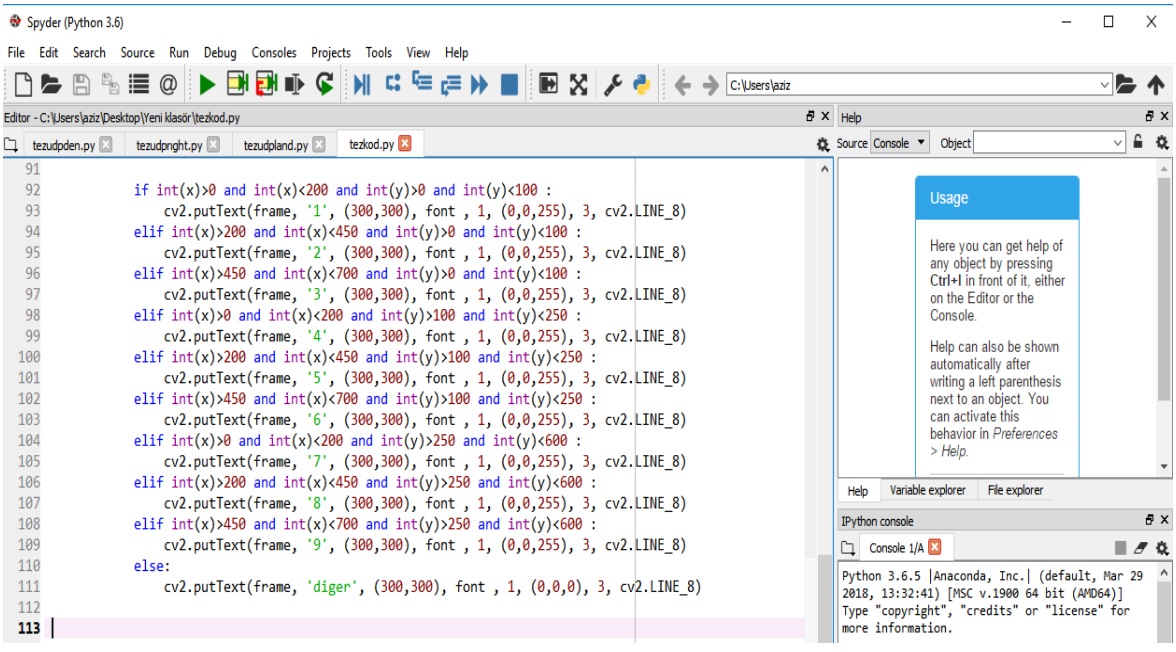
Tespit edilen yeşil renkli hareketli nesnenin takibinin sağlanabilmesi için görüntü ekranı Şekil 6.4.'teki gibi x ve y deki belirtilen kısımlardan çizilen çizgilerle dokuz bölüme ayrılmıştır.

Her bölüm numaralandırılarak hareketli nesnenin hangi bölge sınırları aralıklarında olduğuna dair şartların belirlenmesi için hazırlanan bölge aralıkları tablosu Tablo 6.1.'de gösterilmiştir.

Tablo 6.1. Ekran görüntüsünün sınırlarının belirlenmesi

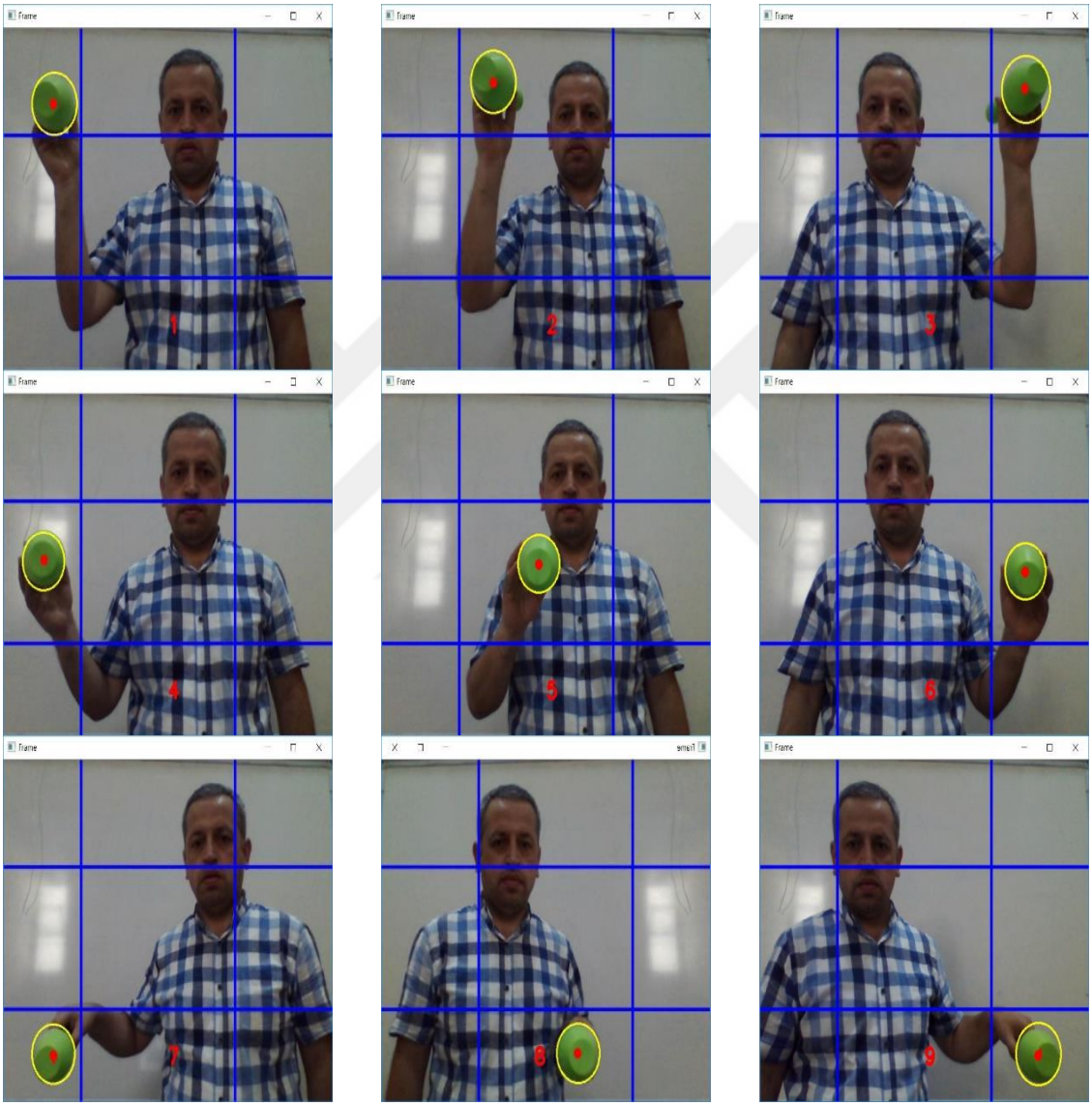
1.Bölge 0<X<200 ve 0<Y<100	**2.Bölge** 200<X<450 ve 0<Y<100	**3.Bölge** 450<X<700 ve 0<Y<100
4.Bölge 0<X<200 ve 100<Y<250	**5.Bölge** 200<X<450 ve 100<Y<250	**6.Bölge** 450<X<700 ve 100<Y<250
7.Bölge 0<X<200 ve 250<Y<600	**8.Bölge** 200<X<450 ve 250<Y<600	**9.Bölge** 450<X<700 ve 250<Y<600

Hazırlanan bölge aralıkları tablosuna bağlı olarak yazılımda şart döngülerinin bulunduğu kod kısmı Şekil 6.11.'de gösterilmiştir.



Şekil 6.11. Bölge aralıkları tablosuna göre şart döngülerinin bulunduğu kod kısmı

Hareketli yeşil renkli nesnenin hangi bölgede olduğunun tespiti x ve y düzleminde çizilen çizgiler arasında kalan alanların bulunması amacıyla oluşturulan şart döngüleri sayesinde Şekil 6.12.'deki gibi sağlanarak istenilen sonuçlar elde edilmiştir. Şekil 6.12.'deki resimlerde hareketli nesnenin hangi bölgede olduğu bilgisi, insansız hava aracının kontrolünün sağlanması gerçekleştirilmek için kullanılmıştır.



Şekil 6.12. Sabit kamera ile hareketli yeşil renkli nesne takibi uygulaması ve bölge tespiti sonuçları

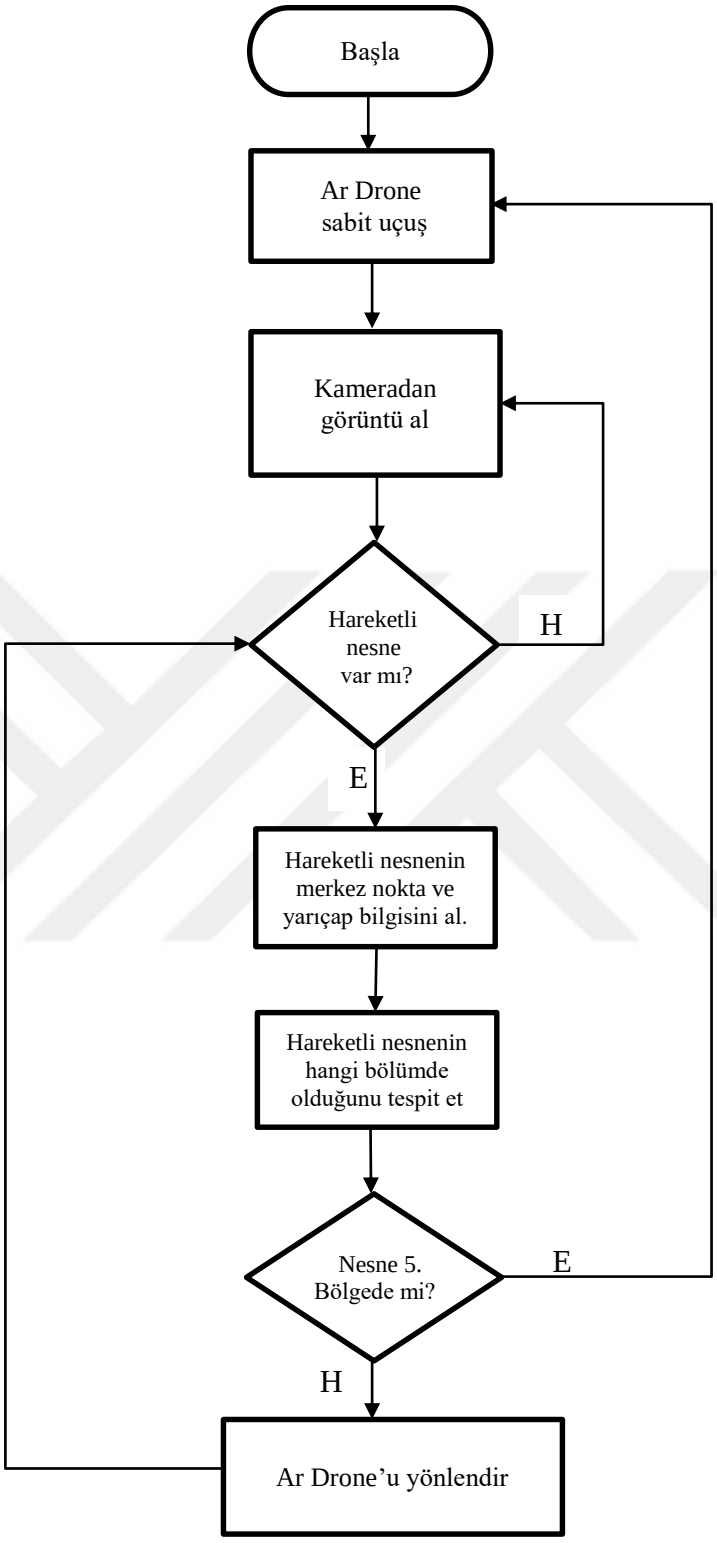
Yazılan programın çalıştırılması için hazırlanmış olan sistem düzeneği Şekil 6.13.'te gösterilmiştir.

Şekil 6.13. Hazırlanmış olan sistem düzeneği

Ön kameradan alınan bilgiler ışığında hareketli nesnenin merkezinin 5. bölgede ve takip anında kullanılan çemberin yarıçap (radius) değerinin 50 piksel olacak şekilde takip edilmesi sağlanmaktadır. Görüntünün 5. Bölgede sağlanabilmesi için diğer tüm durumlarda tespit edilen hareketli nesne için Ar.Drone sağa, sola, yukarı, aşağı, ileri ve geri olacak şekilde hareket ettirilerek takip işlemi sağlanmaktadır. Hareketli nesnenin konumuna göre Ar.Drone'nin yapması gereken hareketler Tablo 6.2.'de gösterilmiştir. Takip işlemi için hazırlanan akış diyagramı Şekil 6.14.'te gösterilmiştir.

Tablo 6.2. Hareketli Nesnenin Konumuna göre Ar.Drone'nin Yapması Gereken Hareketler

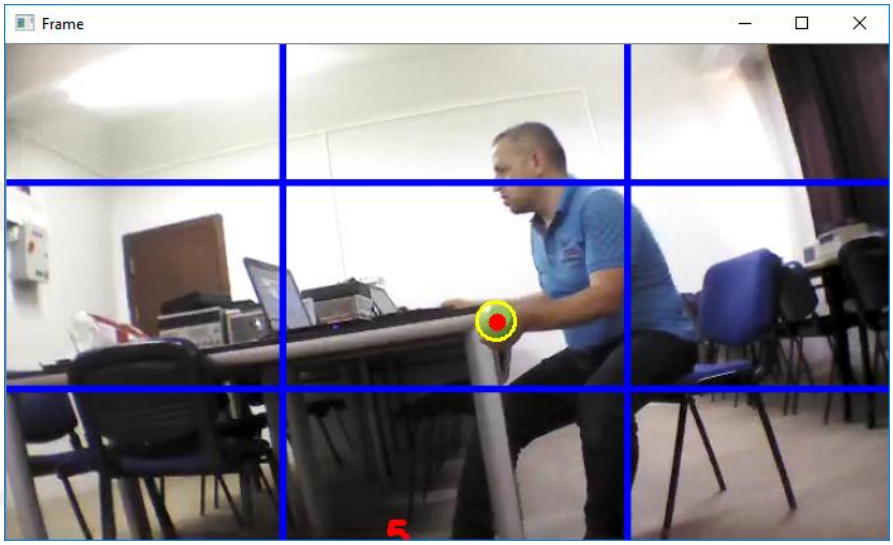
Hareketli Nesnenin Bulunduğu Bölge veya Durum	**Ar.Drone'nin Gitmesi Gereken Yönler**	**Ar.Drone'nin Yapması Gereken Hareketler**
1.Bölge	Sol + Yukarı	Roll + Altitude
2.Bölge	Yukarı	Altitude
3.Bölge	Sağ + Yukarı	Roll + Altitude
4.Bölge	Sol	Roll
6.Bölge	Sağ	Roll
7.Bölge	Sol + Aşağı	Roll + Altitude
8.Bölge	Aşağı	Altitude
9.Bölge	Sağ + Aşağı	Roll + Altitude
Radius değeri < 50	İleri	Pitch
Radius değeri > 50	Geri	Pitch



Şekil 6.14. Program akış diyagramı

6.4. Wifi Üzerinden Ar.Drone'nin Kontrol Edilmesi

Tez çalışmasının bu bölümünde AR.Drone'nin ön kamerasına bağlanılarak alınan görüntülerden özelliği belirtilen hareketli nesnenin tespitinin yapılması sağlanmış ve bu nesnenin takip edilmesi gerçekleştirilmiştir. Ar.Drone IP adresi olan "192.168.1.1" ve görüntü alınması sağlayan "5555" numaralı TCP portuna bağlanılarak görüntü alımı gerçekleştirilmiştir. Alınan örnek görüntü Şekil 6.15.'te gösterilmektedir.



Şekil 6.15. AR.Drone'nin ön kamerasından alınan görüntü

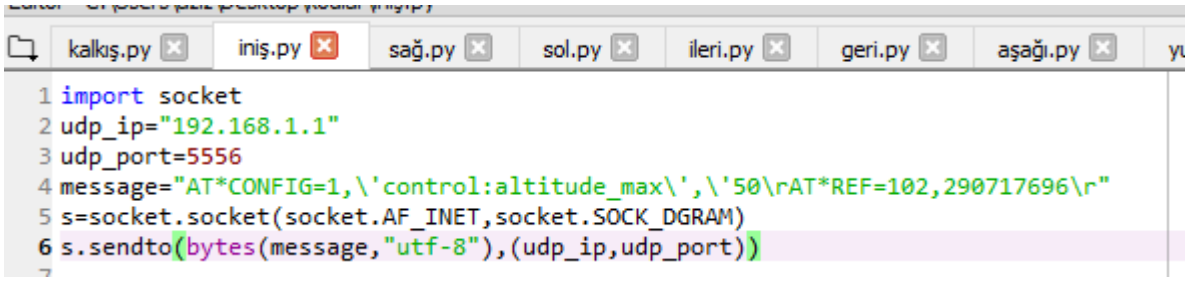
Wifi üzerinden AR.Drone kamerasına bağlanıp görüntü alındıktan sonra hareketli nesnenin takibinin sağlanabilmesi için Tablo 6.2.'deki hareketleri sağlayabilmek amacıyla Python dilinde her hareket için ayrı kodlar yazılmıştır. Yazılan kodlar AR.Drone'nin veri almasını sağlayan "5556" numaralı UDP portu üzerinden gönderilerek çalıştırılmıştır. Şekil 6.16.'da AR.Drone'nin kalkış hareketini sağlayan kod parçacığı gösterilmektedir.

kalkış.py | iniş.py | sağ.py | sol.py | ileri.py | geri.py | aşağı.py

```
import socket
udp_ip="192.168.1.1"
udp_port=5556
message="AT*CONFIG=1,\'control:altitude_max\',\'150\rAT*REF=101,290718208\r"
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.sendto(bytes(message,"utf-8"),(udp_ip,udp_port))
```

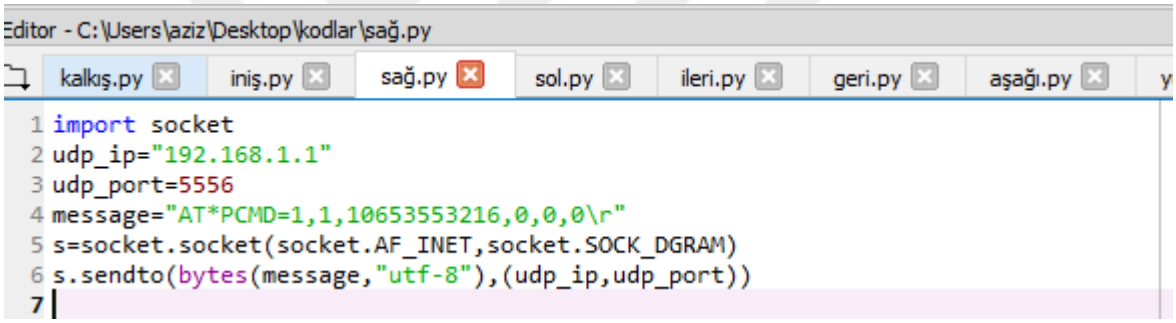
Şekil 6.16. AR.Drone'nin kalkış hareketini sağlayan Python kodu

Şekil 6.17.'de AR.Drone'nin iniş hareketini sağlayan kod parçacığı gösterilmektedir.



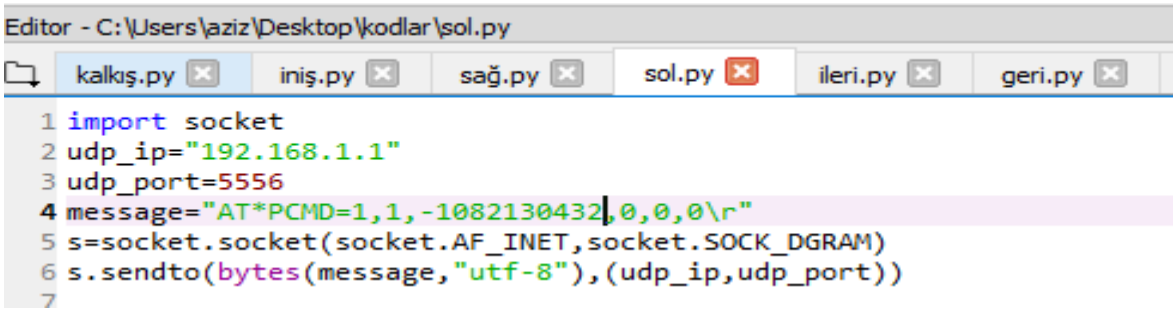
Şekil 6.17. AR.Drone'nin iniş hareketini sağlayan Python kodu

Şekil 6.18.'de AR.Drone'nin sağa doğru yalpalama hareketini sağlayan kod parçacığı gösterilmektedir.



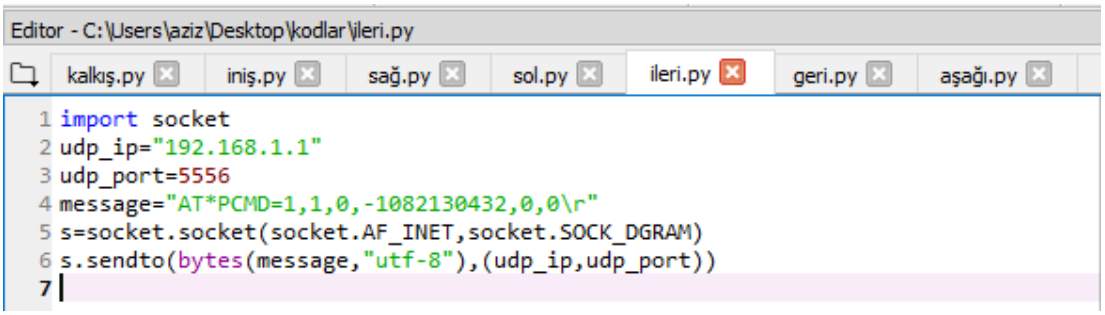
Şekil 6.18. AR.Drone'nin sağa doğru yalpalama hareketini sağlayan python kodu

Şekil 6.19.'da AR.Drone'nin sola doğru yalpalama hareketini sağlayan kod parçacığı gösterilmektedir.



Şekil 6.19. AR.Drone'nin sola doğru yalpalama hareketini sağlayan python kodu

Şekil 6.20.'de AR.Drone'nin ileri doğru yunuslama hareketini sağlayan kod parçacığı gösterilmektedir.

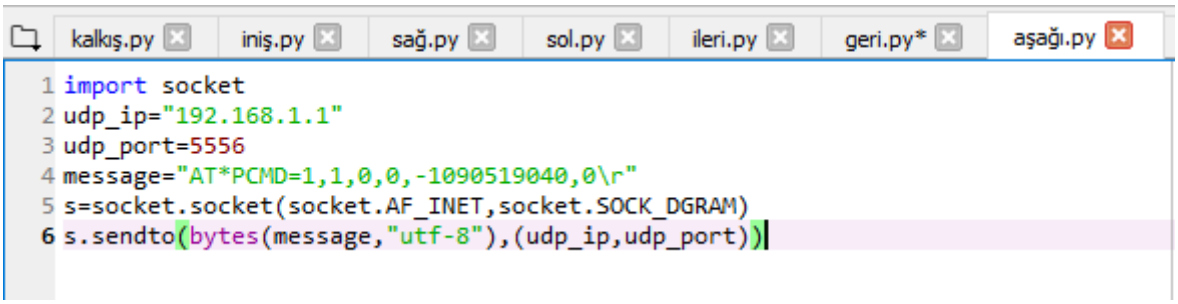


Şekil 6.20. AR.Drone'nin ileri doğru yunuslama hareketini sağlayan python kodu

Şekil 6.21.'de AR.Drone'nin geriye doğru yunuslama hareketini sağlayan kod parçacığı gösterilmektedir.

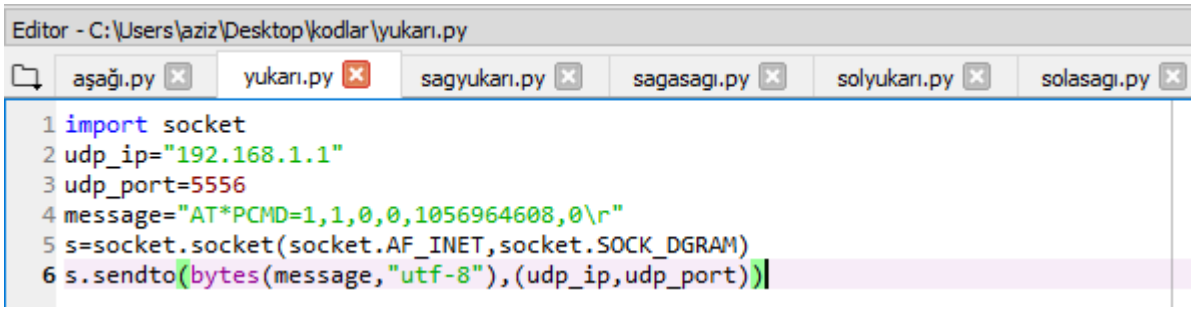
Şekil 6.21. AR.Drone'nin geriye doğru yunuslama hareketini sağlayan python kodu

Şekil 6.22.'de AR.Drone'nin aşağı doğru altitude hareketini sağlayan kod parçacığı gösterilmektedir.



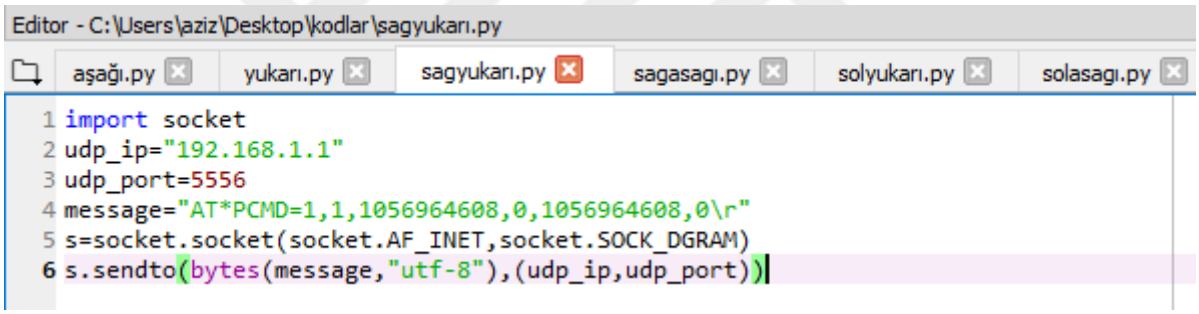
Şekil 6.22. AR.Drone'nin aşağı doğru altitude hareketini sağlayan python kodu

Şekil 6.23.'te AR.Drone'nin yukarıya doğru altitude hareketini sağlayan kod parçacığı gösterilmektedir.



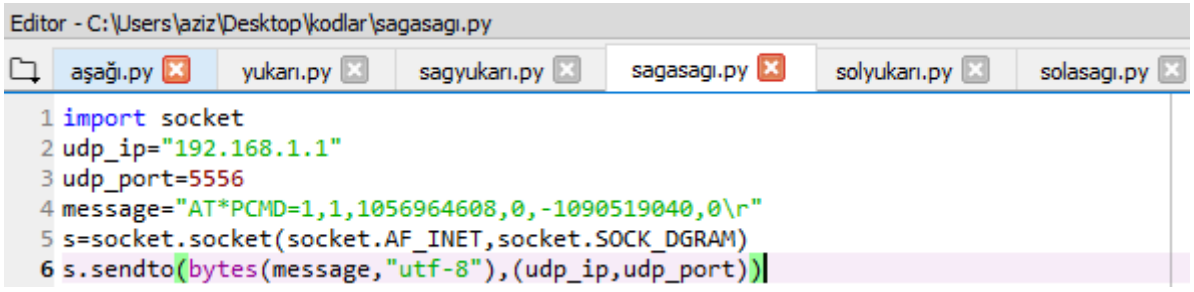
Şekil 6.23. AR.Drone'nin yukarı doğru altitude hareketini sağlayan python kodu

Şekil 6.24.'te AR.Drone'nin sağa ve yukarıya doğru hareketini sağlayan kod parçacığı gösterilmektedir.



Şekil 6.24. AR.Drone'nin sağa ve yukarıya doğru hareketini sağlayan python kodu

Şekil 6.25.'te AR.Drone'nin sağa ve aşağıya doğru hareketini sağlayan kod parçacığı gösterilmektedir.



Şekil 6.25. AR.Drone'nin sağa ve aşağıya doğru hareketini sağlayan python kodu

Şekil 6.26.'da AR.Drone'nin sola ve yukarıya doğru hareketini sağlayan kod parçacığı gösterilmektedir.

Editor - C:\Users\aziz\Desktop\kodlar\solyukarı.py

aşağı.py | yukarı.py | sagyukarı.py | sagasagı.py | solyukarı.py | solasagı.py

```
1 import socket
2 udp_ip="192.168.1.1"
3 udp_port=5556
4 message="AT*PCMD=1,1,-1090519040,0,1056964608,0\r"
5 s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
6 s.sendto(bytes(message,"utf-8"),(udp_ip,udp_port))
```

Şekil 6.26. AR.Drone'nin sola ve yukarıya doğru hareketini sağlayan python kodu

Şekil 6.27'de AR.Drone'nin sola ve aşağıya doğru hareketini sağlayan kod parçacığı gösterilmektedir.

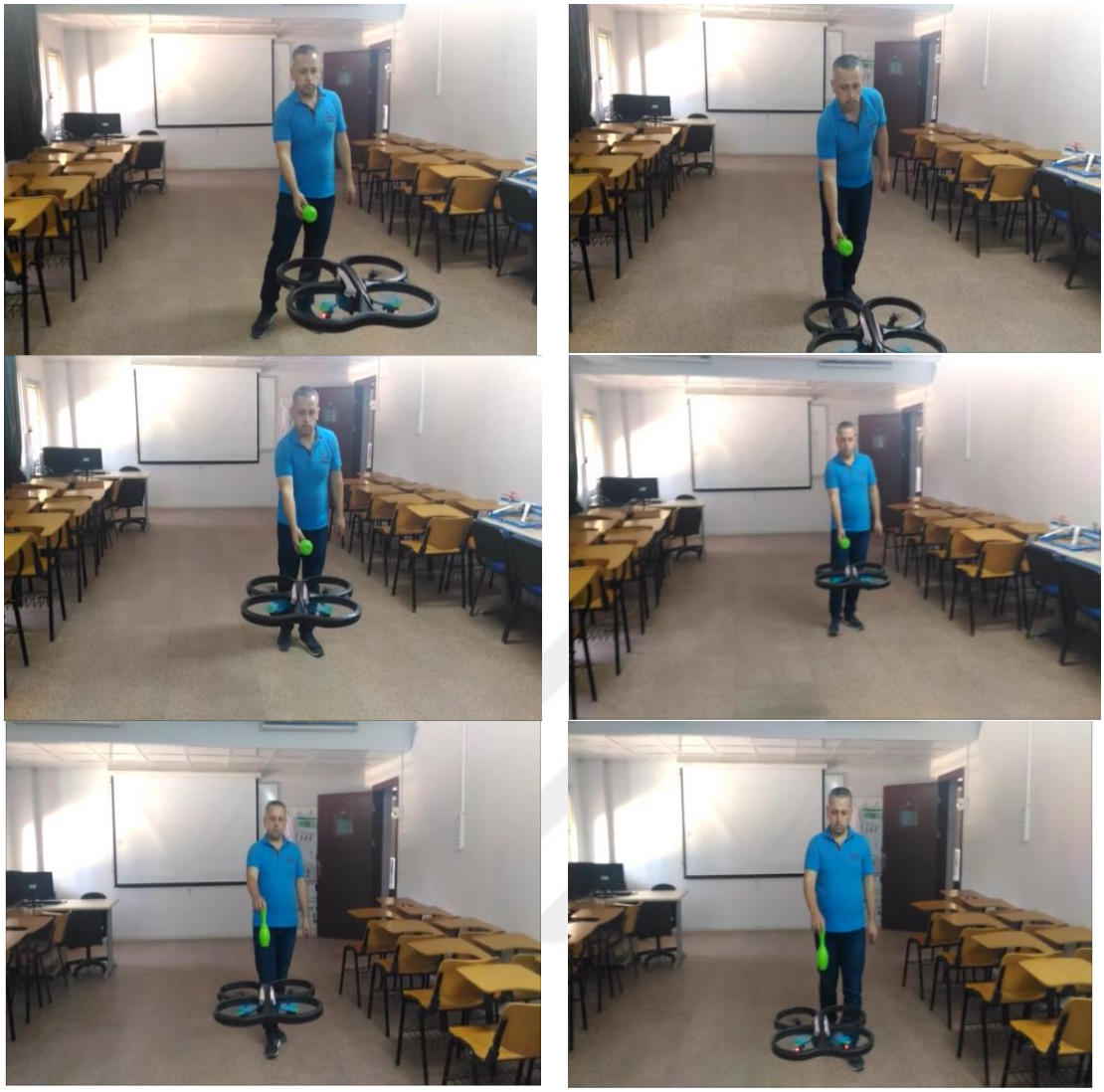
Editor - C:\Users\aziz\Desktop\kodlar\solasagı.py

aşağı.py | yukarı.py | sagyukarı.py | sagasagı.py | solyukarı.py | solasagı.py

```
1 import socket
2 udp_ip="192.168.1.1"
3 udp_port=5556
4 message="AT*PCMD=1,1,-1090519040,0,-1090519040,0\r"
5 s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
6 s.sendto(bytes(message,"utf-8"),(udp_ip,udp_port))
7
```

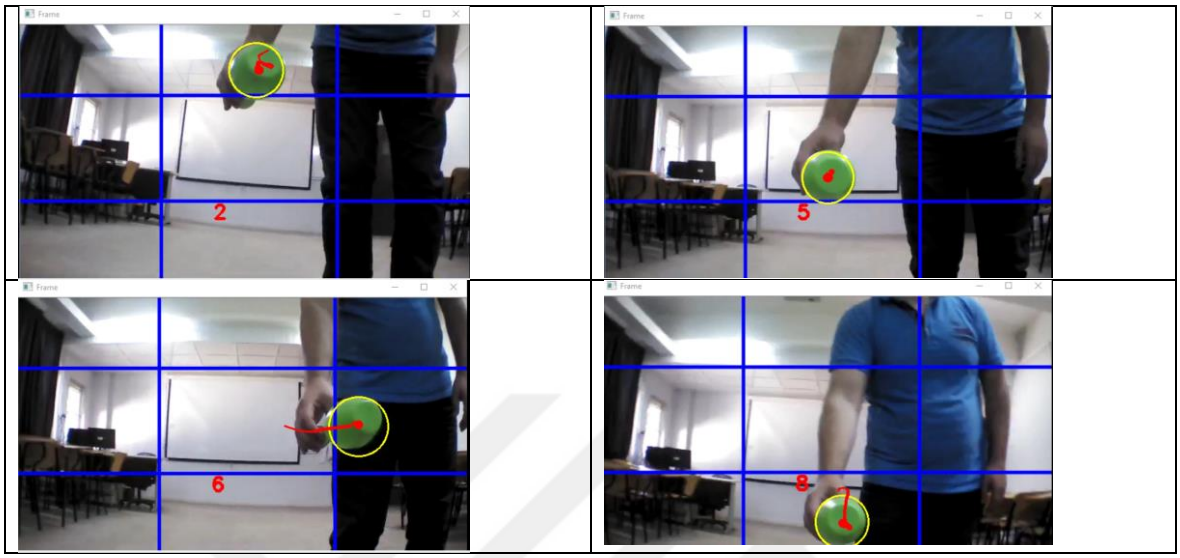
Şekil 6.27. AR.Drone'nin sola ve aşağıya doğru hareketini sağlayan python kodu

Python dilinde yazılan bu kod parçacıkları programın akış şemasına uygun olacak şekilde birleştirilerek renk bileşenine göre nokta tabanlı nesne takip işlemi gerçekleştirilmiştir. Yapılan nesne takibi görüntüleri Şekil 6.28.'de gösterilmiştir.



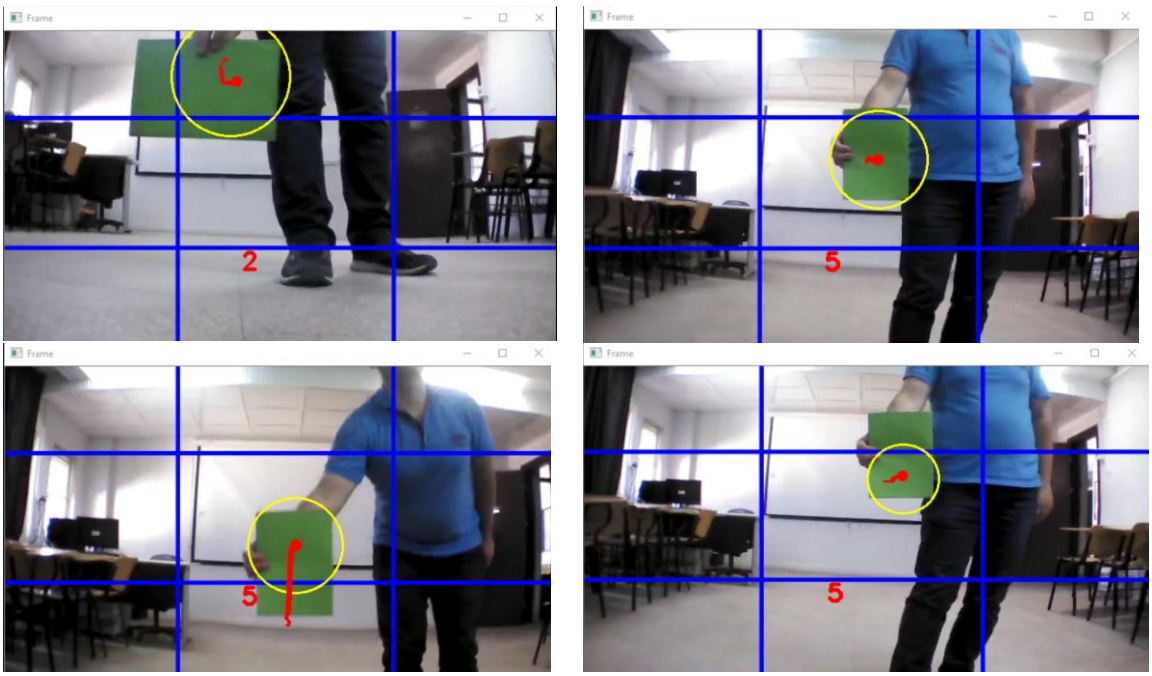
Şekil 6.28. Yeşil renkli nesnenin takibinin dış ortamdan görüntülenmesi

Yazılımın çalıştığı anda kullanılan bilgisayara ait ekran görüntüleri Şekil 6.29.'da gösterilmiştir.



Şekil 6.29. Yeşil renkli nesnenin takibinin program ekranından görüntülenmesi

Yeşil rengin farklı bir tonunda ve farklı bir boyutta nesne takibi yapıldığı bilgisayara ait ekran görüntüleri Şekil 6.30.'da gösterilmiştir.



Şekil 6.30. Farklı bir nesnenin takibinin program ekranından görüntülenmesi

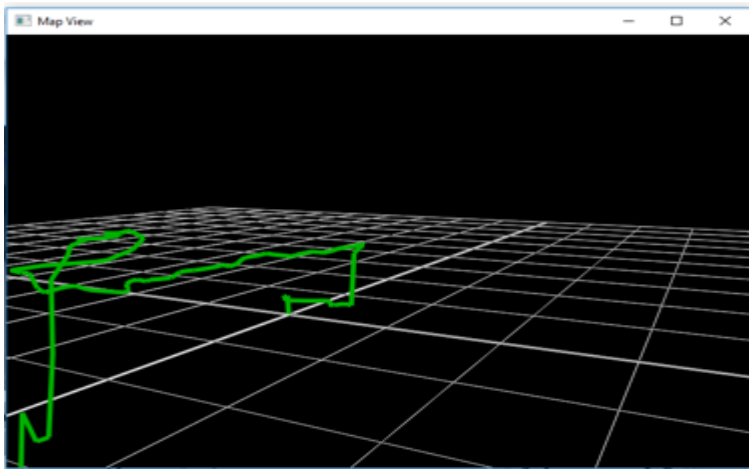
6.5. AutoFlight Programında Hareketin Rotasının Bulunması

Python programlama dili kullanılarak geliştirilen yazılımın kullanılması sırasında Ar.Drone'nin yapmış olduğu hareketlerin rotasının alınması için AutoFlight programı kullanılmıştır. AutoFlight programının AutoScript kısmından da Ar.Drone kontrol komutları kullanılarak Ar.Drone'nin hareketleri kontrol edilebilmektedir. Tablo 6.3.'de Ar.Drone kontrol komutlarından birkaç tanesi örnek olarak verilmiştir.

Tablo 6.3. Ar.Drone'nin AutoFlight programındaki kontrol komutları

Komut	Yaptığı Hareket
control.takeOff()	Kalkış hareketi
control.hover()	Sabit konumda kalma
control.right(speed)	Sağa hareket
control.left(speed)	Sola hareket
control.up(speed)	Yukarı hareket
control.down(speed)	Aşağı hareket
control.forward(speed)	İleri hareket
control.backward(speed)	Geriye hareket

AutoFlight programının Ar.Drone ile bağlantısı sağlanarak cihazın hareket halinde iken izlemiş olduğu rota çizgisi Şekil 6.31.'de gösterilmiştir.



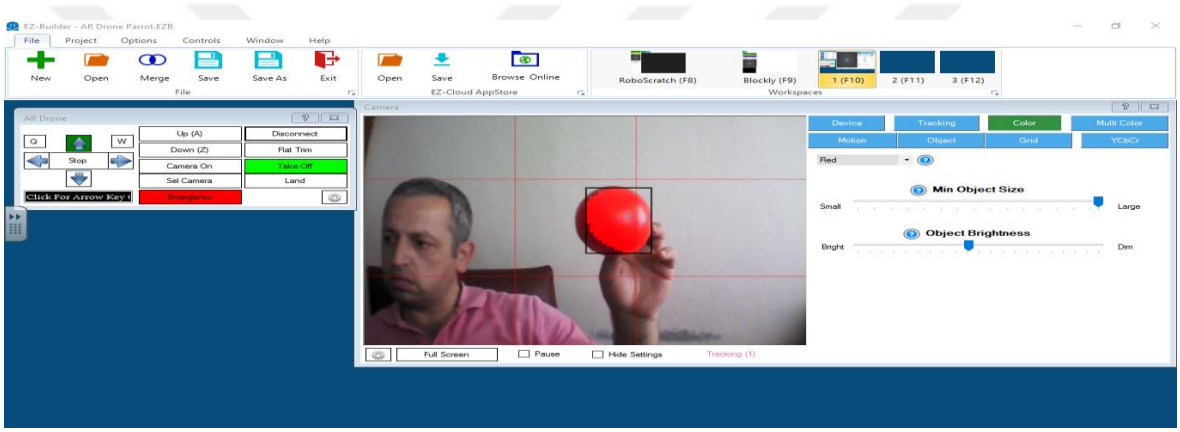
Şekil 6.31. Ar.Drone'nin izlemiş olduğu rota çizgisi

6.6. EZ-Builder Ortamında Yapılan Çalışmalar

EZ-Builder, programında da farklı renk ve nesnelerinin takibinin sağlanması ile ilgili olarak deneme amaçlı çalışmalar yapılmıştır. EZ-Builder programı içerisinde de kod blokları kullanılarak istenilen yazılımlar oluşturulabilmektedir.

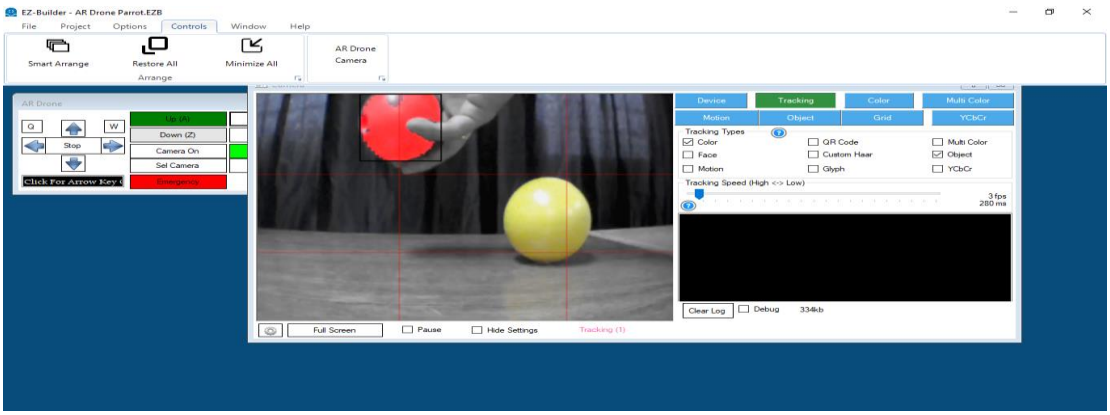
6.6.1. Red – Green – Blue Renk Uzayında Tanıtılan Nesnenin Takibi

RGB renk uzayında kırmızı renkli bir topun sisteme tanıtılma işlemi Şekil 6.32.'deki gibi yapılarak takip işlemi gerçekleştirilmiştir.



Şekil 6.32. Kırmızı renkli nesnenin sisteme tanıtılması

Şekil 6.33.'te nesnenin yukarı hareketine göre Ar.Drone nun yukarı hareket (Up) edeceği bilgisi görünmektedir.



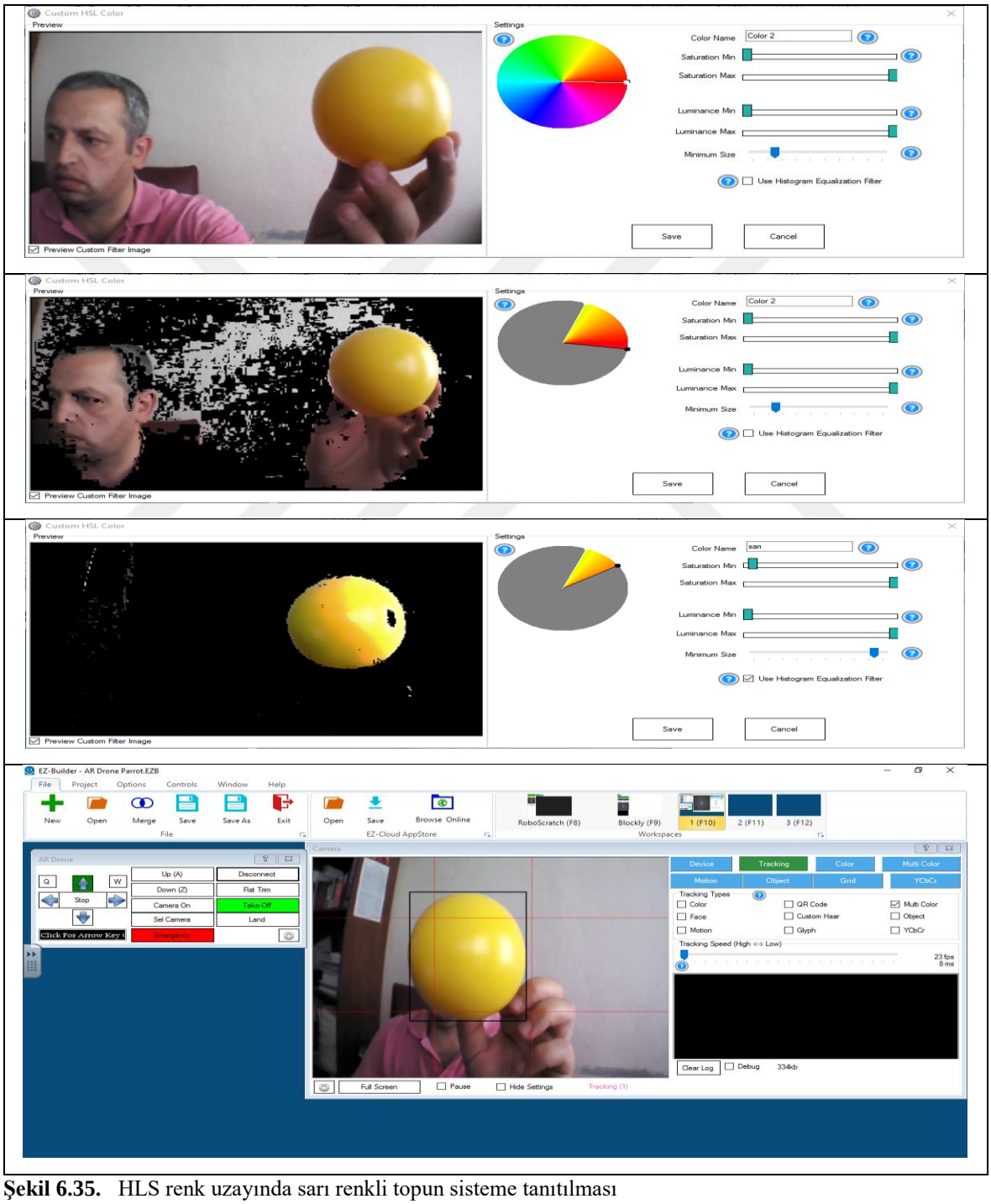
Şekil 6.33. Kırmızı renkli nesnenin hareketinin takibi

Yapılan tanıtım ve ayarlamalardan sonra Ar.Drone'nin EZ-Builder ortamında kırmızı renkli nesnenin takibine ait kamera görüntüleri Şekil 6.34.'te gösterilmiştir.

Şekil 6.34. Ar.Drone'nin EZ-Builder ortamında kırmızı renkli nesneyi takibine ait kamera görüntüleri

6.6.2. Hue – Luminance – Saturation Renk Uzayında Tanıtılan Nesnenin Takibi

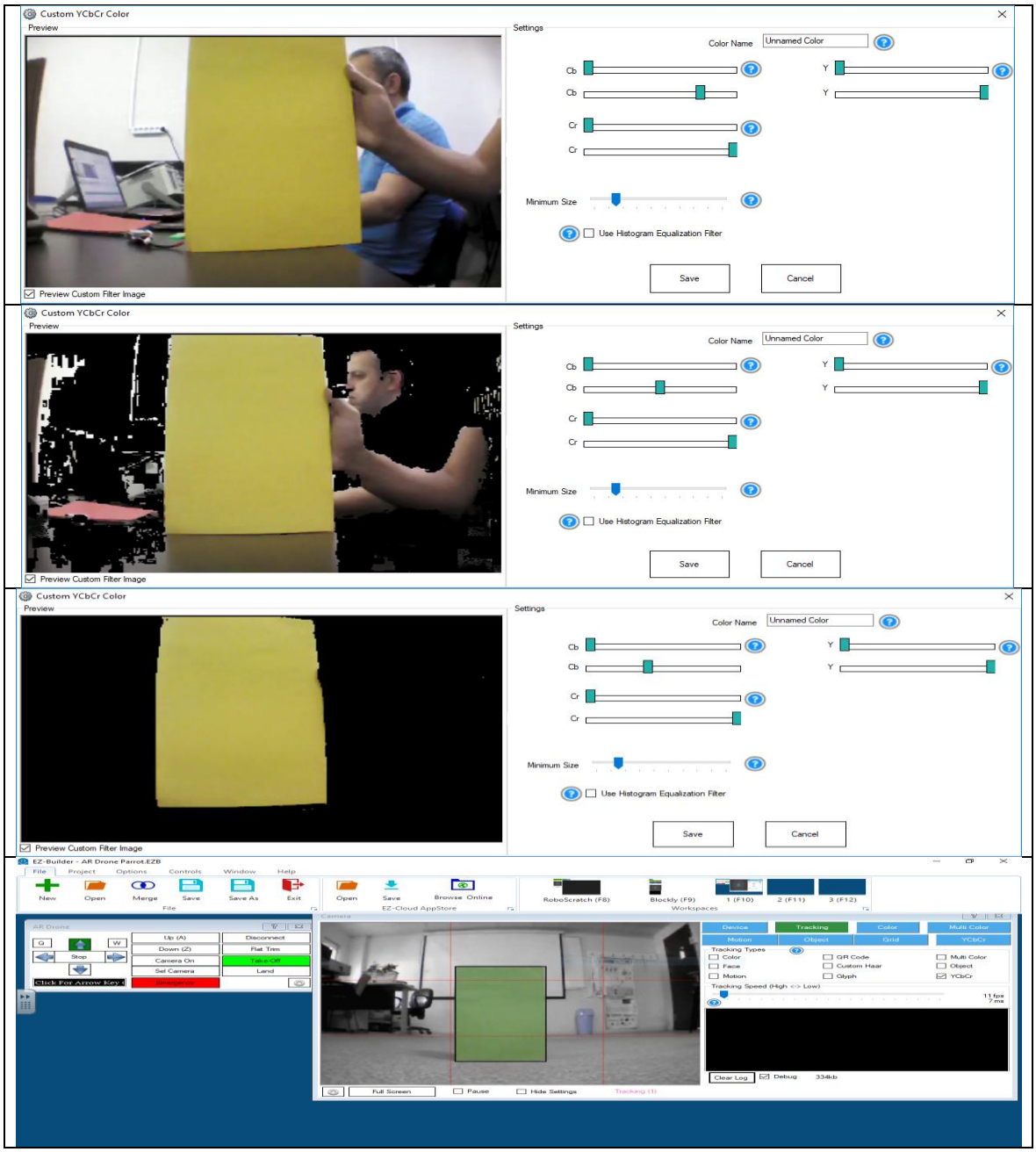
HLS renk uzayında sarı renkli bir topun sisteme tanıtılması işlemi için gerekli ayarlamalar EZ-Builder programında ayarlanarak Ar.Drone'nin istediğimiz renkteki nesneyi takip etmesi sağlanmıştır. Yapılan çalışmaya ait görüntüler Şekil 6.35.'te gösterilmiştir.



Şekil 6.35. HLS renk uzayında sarı renkli topun sisteme tanıtılması

6.6.3. Luminance - Chrominance blue - Chrominance red Renk Uzayında Tanıtılan Nesnenin Takibi

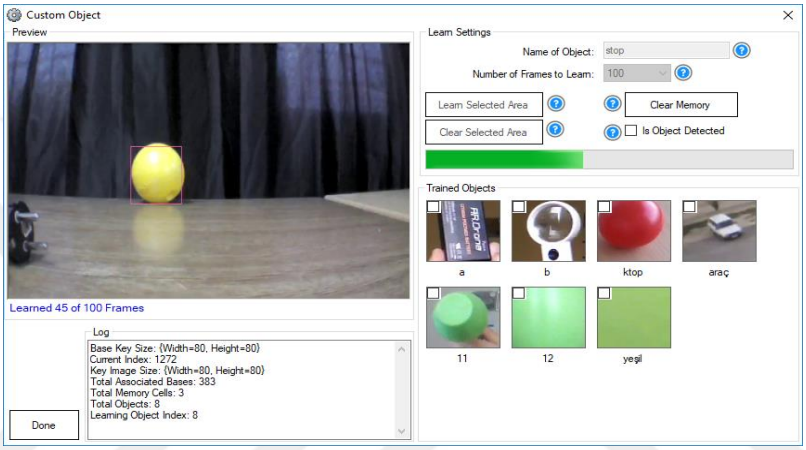
YCbCr renk uzayında sarı ve yeşil renkli dikdörtgen nesnelerin sisteme tanıtılması sağlanarak Ar.Drone'nin takip edilmesi istenilen renkteki nesneyi takip etmesi sağlanmıştır. Yapılan çalışmaya ait görüntüler Şekil 6.36.'da gösterilmiştir.



Şekil 6.36. YCbCr renk uzayında sarı ve yeşil renkli dikdörtgen nesnelerin sisteme tanıtılması

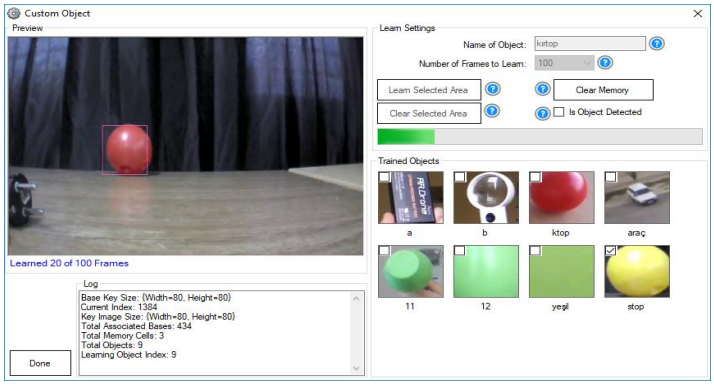
6.6.4. Siluet Tabanlı Nesne Takibi

Siluet tabanlı nesne takibinin yapılabilmesi için ilk önce veri tabanının oluşturulması ve ardından takip edilecek objenin sistemin veri tabanına kaydedilmesi gerekmektedir. Şekil 6.37.'de ekranda belirtilen sınırlar içerisine alınan sarı renkli objenin sistemin veri tabanına yüklenmesi işlemi gerçekleşmektedir.



Şekil 6.37. Objenin sistemin veri tabanına kaydedilmesi

Şekil 6.38.'de sarı renkli objenin sistemin veri tabanına yüklenmiş olduğu görülmektedir. Kırmızı renkli objenin de sistemin veri tabanına yüklenmesi işlemi gerçekleşmektedir. Yapılan bu işlemlerden sonra takip edilecek nesne seçilerek siluet tabanlı takip işlemi gerçekleştirilmektedir.



Şekil 6.38. Veri tabanındaki objeler

7. SONUÇ VE DEĞERLENDİRME

Tezde dört rotorlu insansız hava aracının görme tabanlı nesne takibi yapması gerçekleştirilmiştir. Nesne takip aşaması için Parrot firmasının Parrot AR.Drone 2.0 Power Edition ürünü kullanılmıştır. Nesne takip işleminde, cihazın ön kamerasından alınan görüntüler kullanılarak takip edilecek nesnenin renk, şekil ve koordinat bilgileri kullanılmıştır. Python programlama dilinde AT komutları kullanılarak yazılan kod parçacıkları ile insansız hava aracının yapması istenilen hareketler gerçekleştirilmiştir. Sabit kameradan alınan görüntülerden hareketli nesnenin tespiti yapılarak nokta ve çekirdek tabanlı nesne takibi gerçekleştiren program yazılmıştır. Programda AR.Drone kamerasına wifi üzerinden bağlanılarak alınan görüntüler üzerinde OpenCV'nin görüntü işleme teknikleri kullanılmış ve görüntü tabanlı nesne takip işlemi gerçekleştirilmiştir. AutoFlight programında nesnenin hareketi sırasında yapmış olduğu rota incelenmiştir. EZ-Builder programında RGB, HLS ve YCbCr uzaylarında belirlenen renkteki nesnelerin sisteme tanıtılması ve takibi ile ilgili uygulamalar gerçekleştirilmiştir. Siluet tabanlı nesne takip için veri tabanına nesne ekleme işlemi sağlanarak veri tabanından seçilen nesnenin takip işlemi gerçekleştirilmiştir.

Bu tez kapsamında geliştirilen program ile OpenCV'nin farklı görüntü işleme teknikleri kullanılarak takip edilmek istenen nesneler arttırılabilir. Kullanılan cihazlar, yazılımların geliştirilmesiyle farklı işlevleri yerine getirebilir. Bu tür cihazların kullanılmasının arttırılması ile özellikle algoritma ve programlama mantığının yeni nesillere öğretilmesi sağlanabilir. İnsansız hava araçlarının özellikle arama kurtarma faaliyetlerinde kullanılırken gece görüşlü termal kamera sistemlerinin de görüntü işleme platformlarında kullanılması sağlanabilir.

KAYNAKLAR

[1] **Hasköy, E.,** 2010. Dört Rotorlu Hava Araçlarının Bağ-Grafik Yöntemi İle Modellenmesi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

[2] **Kansu Y, Şenöz S. ve Öztuna Y.,** 1971**.** Havacılık Tarihinde Türkler, C.I, Ankara, Hava Kuvvetleri Basımevi, , s.46

[3] **Önkol, M.,** 2010, Dönerkanat Tipinde Bir İnsansız Hava Aracının Tasarımı, Modellenmesi ve Kontrolü, Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara

[4] **Castillo P., Lozano R., Dzul A.,** 2005. Modeling and Control of Mini-Flying Machines, Springer

[5] **Kurtoğlu, S.,** 2009. Dört Pervaneli Uçuş Aracı Deney Düzeneği Donanım ve Kontrol Algoritmalarının Tasarımı, Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü, Gebze

[6] **Oflaz, T.,** 2013, Dört Rotorlu Hava Aracının İrtifa Denetimi İçin Doğrusal Olmayan Denetleyici Tasarımı ve Uygulaması, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

[7] **İlhan, İ.,** 2013, Dört Rotorlu İnsansız Hava Aracı İçin Görüntü İşleme Tabanlı Akıllı Kontrol Algoritmalarının Geliştirilmesi, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.

[8] **Erginer B.,** 2007, Quadrotor Vtol Aracının Modellenmesi ve Kontrolü, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, İstanbul.

[9] **Akyüz, S.,** 2013, Dört Rotorlu Hava İnsansız Hava Aracı (Quadrotor)'nın Pd ve Bulanık Kontrolcü Tasarımı ve Benzetim Uygulaması, Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.

[10] **Alper Yılmaz, O. Javed, M. Shah,** 2006, "Object tracking: a survey", ACM Computing Surveys 38(4): 13-es.

[11] **Qi Zang, Reinhard Klette,** 2003, "Object Classification and Tracking in Video Surveillance", CAIP, 198-205.

[12] **Kye Kyung Kim, Soo Hyun Cho, Hae Jin Kim, Jae Yeon Lee,** 2005, "Detecting and Tracking Moving Object Using an Active Camera", Digital Object Identifier 0.1109/ICACT,817-820.

[13] **Takashi Morimoto, Osamu Kiriyama, Youmei Harada, Hidekazu Adachi, Tetsushi Koide,Hans Jürgen Mattausch,** 2005, "Object Tracking in Video Pictures based on Image Segmentation and Pattern Matching", IEEE, 3215-3218.

[14] **Tetsushi Koide, Hans Jürgen Mattausch, Takashi Morimoto, Hidekazu Adachi, Kosuke Yamaoka,** 2005, "Real-Time Multi-Object Tracking Based on Highly Parallel Image Segmentation and Pattern Matching".

[15] **Soon-Yong Park, Chang-Joon Park ve Inho Lee,** 2005, "Moving Object Removal and Background Completion in a Video Sequence", Image and Vision Computing New Zealand.

[16] **Isaac Cohen ve Gerard Medioni,** 1998, "Detecting and Tracking Moving Objects inVideo from an Airborne Observer", Image Understanding Workshop, 217-222.

[17] **Alper Yılmaz, X. Li, M. Shah,** 2004, "Object Contour Tracking Using Level Sets", Asian Conference on Computer Vision, ACCV 2004, Jaju Islands, Korea

[18] **Dorin Comaniciu, V. Ramesh ve P. Meer,** 2000, "Real-Time Tracking of Non-Rigid Objects using Mean Shift", Computer Vision and Pattern Recognition, 142-149.

[19] **Dorin Comaniciu, V. Ramesh ve P. Meer,** 2003, "Kernel-Based Object Tracking", IEEE Trans. Pattern Analysis and Machine Intelligence, May 2003, vol. 25, no. 5, pp. 564-575.

[20] **Chiu, C.C., Ku, M.Y., Wang, C.Y.,** 2010, Automatic Traffic Surveillance System for Vision-Based Vehicle Recognition and Tracking, Journal Of Information Science And Engineering 26, vol. 611-629

[21] **Alper Yılmaz, X. Li ve M. Shah,** 2004, "Contour Based Object Tracking With Occlusion Handling in Video Acquired Using Mobile Cameras," IEEE Trans. Pattern Anal. Mach. Intell. 26, 1531-1536.

[22] **Krajnık Tomas, Vonasek Vojtech, Fiser Daniel, Faigl Jan.** 2011, AR-Drone as a Platform for Robotic Research and Education. Heidelberg: Springer, 2011. ISSN 1865-0929

[23] **Engel, Jakob Julian.** 2011, Autonomous Camera-Based Navigation of a Quadcopter. Munich: Technical University Munich 2011. Master Thesis. Technical University Munich, Faculty of Informatics, Computer Vision Group.

[24] **Burkay Birant Örten,** 2005, "Güvenlik Amaçlı Video Sistemlerinde Hareketli Nesnelerin Tanınması ve Olay Analizi", Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

[25] **Yigithan Dedeoğlu,** 2004, "Akıllı Video Gözetimi İçin Hareketli Nesne Bulma, Takip Etme Ve Sınıflandırma", Yüksek Lisans Tezi, İhsan Doğramacı Bilkent Üniversitesi, Mühendislik ve Fen Bilimleri Enstitüsü, Ankara.

[26] **İlhan İ. and Karaköse M.,** 2013, Type-2 Fuzzy Based Quadrotor Control Approach, Asian Control Conference 2013 (ASCC2013), İstanbul, Türkiye.

[27] **Saif A., Dhaifullah M., Al-Malki M. ve El Shafie M.,** 2012, Modified Backstepping Control of Quadrotor, 9th International Multi-Conference on Systems, Signals and Devices, Dhahran, Saudi Arabia, pp. 1-6.

.

[28] **Lee G., Jeong D. Y. and Khoi N. D.,** 2011, Attitude Control System Design For A Quadrotor Flying Robot, 8th International Conference on Taesam Kang Ubiquitous Robots and Ambient Intelligence (URAI), Seoul, South Korea, pp. 74-78.

[29] https://www.parrot.com/us/drones/parrot-ardrone-20-power-edition#technicals 20 Kasım 2017

[30] **Boyraz Ö.F.,** "Mobil Damar Görüntüleme Cihazı Tasarımı", Yüksek Lisans Tezi, T.C. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Sakarya, 2015

[31] **Kutlu H.,** "İnsan Bilgisayar Etkileşimli Görüntü İşleme Uygulamaları", Yüksek Lisans Tezi, T.C. Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı Bilgisayar Sistemleri Eğitimi, Elazığ 2013.

[32] **Erişti, E.,** 2010, "Görüntü İşlemede Yeni Bir Soluk OpenCV", XII. Akademik Bilişim Konferansı Bildirileri, Muğla Üniversitesi, 223–229

[33] **Bradski, G. and Kaehler, A.,** 2008, "Learning OpenCV: Computer Vision with the OpenCV Library", O'Reilly Media, Amerika Birlesik Devletleri, 16-17

[34] **Harwani, B. M.,** 2011, Introduction to Python Programming and Developing GUI Applications with PyQT, Cengage Learning

[35] **Swaroop, C. H.,** 2003 A Byte of Python

[36] **Özgül, F.,** 2013, Her Yönüyle Python, Kodlab, İstanbul

[37] **Saralıoğlu, E. , Yıldırım, D. , Güngör, O.,** 2014, Uzaktan algılama araştırmacılarına yönelik python ara yüzü ve arcgıs yazılımı eklentisi, 5. Uzaktan Algılama-Cbs Sempozyumu (UZAL-CBS 2014

[38] **Yiğit, Z.,** 2014, Modeling And Control Of Quadrotor Unmanned Aerial Vtol Vehicle Yüksek Lisans Tezi, İstanbul İstanbul Technical University, Department of Control and Automation Engineering, İstanbul.

[39] **Domingues, J. M. B.** 2009, Quadrotor Prototype, Portugal, 101p.

[40] **Bouabdallah, S.,** 2007, Design And Control Of Quadrotors With Application To Autonomous Flying, Swiss, 129p.

[41] **Bayrakçeken, M. K.** 2013 Dikine İniş Kalkış Yapabilen Dört Rotorlu Hava Aracının (Quadrotor) Uçuş Kontrolü, Doktora Tezi, Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir.

[42] **Okur S.,** "Görüntü İşleme Yöntemleri Kullanılarak Gözdeki Damarların Tespit Edilmesi" Yüksek Lisans Tezi, T.C. Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı Bilgisayar Sistemleri Eğitimi, Elazığ 2015

[43] **Güleryüz H.İ.,** "Gri Seviye Görüntülerde Kriptografik Uygulamalar" Yüksek Lisans Tezi, T.C. Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı, Elazığ 2014

[44] **Güvenoğlu E.,** "Görüntü Şifreleme Algoritmaları ve Performans Analizleri" Yüksek Lisans Tezi T.C. Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, Edirne 2006

[45] **Yrd. Doç. Dr. Harun Hilmi P**., 2012, "Grafik Tasarım Sürecinde Kullanılan Aygıtların Renk Modelleri" İdil Dergisi, Cilt 1, Sayı 3, 116-127

[46] **Yılmaz İ.,** 2002 "Renk Sistemleri, Renk Uzayları ve Dönüşümler", Selçuk Üniversitesi Jeodezi ve Fotogrametri Mühendisliği Öğretiminde 30. Yıl Sempozyumu, Konya

[47] **Deniz T.,** 2007, "Sıkıştırılmış Video Akımının Düzensiz Haritalar ve Başlangıç Kodlarına Dayalı Şifrelenmesi", Trakya Üniversitesi Fen Bilimleri Enstitüsü, Edirne

[48] **Buğday A.** 2010 "Gerçek Zamanlı Videolarda Ön Plan Arka Plan Ayrımı", Yüksek Lisans Tezi, Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Mühendisliği Anabilim Dalı, Ankara

[49] **Jack K.,** 1995 "Video Demstified", LLH Technology Publishing

[50] **Polat, H. H."** Grafik Tasarım Sürecinde Kullanılan Aygıtların Renk Modelleri", İdil Dergisi, 2012

[51] **K. Sinan Y., Cenk., T. Emre K.,** "Görüntü İşleme", Ege Üniversitesi Bilgisayar Mühendisliği Bölümü

[52] **Doç. Dr. Sarp E., Arş. Gör. Oğuzhan URHAN,** "İmage İşleme Ders Notları", Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü

[53] **Türkoğlu İ.,** "T.C. Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı Örüntü Tanıma Ders Notları", Elazığ.,2010

[54] **Mehmet K.,** "Görüntü İşleme Teknolojileri ve Uygulamaları", Ege Üniversitesi, Uşak 2012.

[55] **Çelikdemir M., "**Yüksek Sıcaklığa Maruz Kalmış Betonlarda Meydana Gelen Çatlakların Görüntü İşleme Tekniği İle Tespit Edilmesi" Yüksek Lisans Tezi T.C. Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elazığ 2015

[56] **Atilla K.,** "Histogram Trasformation In Image Processing And Its Applications", Szeged Üniversitesi.

[57] **Mustafa A., Saime A.,** "Beyin Emar Görüntülerinin Wtershed Algoritması Yardımıyla Bölütlenmesi", Erciyes Üniversitesi Elektrik-Elktronik Mühendisliği Bölümü, Kayseri.

[58] **Wilson, J. A.** Noble Adaptive Segmentation of MRI Data D. L.

[59] **Enes Ç.,** "Görüntü İşlemeye Dayalı Avuç İçi İzinin Yapay Sinir Ağı İle Tanınması", Elektronik-Bilgisayar Eğitimi Anabilim Dalı Bilgisayar-Kontrol Eğitimi Programı Yüksek Lisans Tezi, Maramara Üniversitesi, İstanbul 2011.

[60] **Chang C., Huang C.,** "Edge Detection Based On Class Ratio", Shenkeng, Taipei, Taiwan 2002.

[61] **Gonzales R. C., Woods R. E.,** "Digital Image Processing", 2nd Edition, ABD, ISBN: 0-13-094650-8, (2001) 1-750.

[62] **Ertürk, S., 2003,** "Digital Image Processing", Kocaeli Üniversitesi Yayını, Parça Numarası: 323604A-01

[63] **Talu M.,** " İnsan Hareketlerinin Takibinde Karşılaşılan Problemlerin Çözümüne Yeni Yaklaşımlar", Doktora Tezi, T.C. Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği, Elazığ 2010

[64] **Veenman, C., Reinders, M., and Backer, E.,** 2001. ―Resolving motion correspondence for densely moving points‖, IEEE Trans. Patt. Analy. Mach. Intell. 23(1):54–72.

[65] **Ushakov, N.G.,** 2001, "Density of a probability distribution", in Hazewinkel, Michiel, Encyclopaedia of Mathematics, Kluwer Academic Publishers, ISBN 978-1556080104.

[66] **Evans, M., Hastings, N., Peacock. B.,** 2000. "Probability Density Function and Probability Function." §2.4 in Statistical Distributions, 3rd ed. New York: Wiley

[67] **Chunhua S., Brooks M.J., Hengel A.,** 2007, ―Fast Global Kernel Density Mode Seeking: Applications to Localization and Tracking‖, IEEE Transactions on Image Processing, 16(5): 1457 – 1469.

[68] **Cootes, T. F., Edwards, G. J., and Taylor, C. J.,** 1998. ―Active Appearance Model‖, Proc European Conference on Computer Vision, Vol. 2, pp. 484-498, Springer.

[69] **Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J.,** 1995. ―Active Shape Models-Their training and application‖, Computer vision and image understanding, Vol. 61. No:1, pp.38-59.

[70] **Lee, S.-W., Kang, J., Shin, J., Paik, J.,** 2007, Hierarchical active shape model with motion prediction for real-time tracking of non-rigid objects‖, IET Comput. Vis., 2007, 1, (1), pp. 17–24.

[71] **Bertalmio, M., Sapiro, G., and Randall, G.,** 2000. ―Morphing active contours‖. IEEE Trans. Patt. Analy. Mach.Intell. 22, 7, 733–737.

[72] **Cordea, M. D., Petriu, E. M., Petriu, D. C.,** 2008, ―Three-Dimensional Head Tracking and Facial Expression Recovery Using an Anthropometric Muscle-Based Active Appearance Model‖, IEEE Transactions on Instrumentation and Measurement, 57(8): 1578 – 1588.

ÖZGEÇMİŞ

27.08.1979 Malatya doğumluyum. İlkokulu Malatya'da ortaokulu, Adıyaman'da okudum. Liseyi Gaziantep M. R. Uzel Endüstri Meslek Lisesi Telekomünikasyon bölümünde, lisans eğitimimi 1998-2002 yılları arasında Elazığ Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Elektronik Öğretmenliği bölümünde tamamladım. 2004-2014 yılları arasında Milli Eğitim Bakanlığında Bilişim Teknolojileri öğretmeni olarak görev yaptım.

18.02.2014 tarihinden itibaren Adıyaman Üniversitesi Besni Meslek Yüksekokulu Elektronik ve Otomasyon bölümünde öğretim görevlisi olarak çalışmaya başladım. Halen Adıyaman Üniversitesi Besni Meslek Yüksekokulu Elektronik ve Otomasyon bölümünde öğretim görevlisi olarak çalışmakta ve Adıyaman Üniversitesi Besni Meslek Yüksekokulu Müdür Yardımcılığı görevini yürütmekteyim.