T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ İLETİŞİM YOLUYLA BİR ANDROİD CİHAZA BAĞLI NESNE TANIMA SİSTEMİ KULLANAN YARI ÖZERK HAREKETLİ ROBOTUN TASARIMI VE GERÇEKLEŞTİRİLMESİ

Majeed ABDULMAJEED

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliği Anabilim Dalı

Mayıs-2019
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Majeed ABDULMAJEED tarafından hazırlanan "Kablosuz İletişim Yoluyla Bir Android Cihaza Bağlı Nesne Tanıma Sistemi Kullanan Yarı Özerk Hareketli Robotun Tasarımı ve Gerçekleştirilmesi" adlı tez çalışması 24/05/2019 tarihinde aşağıdaki jüri tarafından oy birliği / ~~oy çokluğu~~ ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Prof.Dr. Ercan YALDIZ	.....
Danışman Doç.Dr. Levent SEYFİ	.....
Üye Dr.Öğr.Üyesi Sabri ALTUNKAYA	.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. ....
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Majeed ABDULMAJEED

Tarih: 20.05.2019

ÖZET

YÜKSEK LİSANS TEZİ

KABLOSUZ İLETİŞİM YOLUYLA BİR ANDROİD CİHAZA BAĞLI NESNE TANIMA SİSTEMİ KULLANAN YARI ÖZERK HAREKETLİ ROBOTUN TASARIMI VE GERÇEKLEŞTİRİLMESİ

Majeed ABDULMAJEED

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Levent SEYFİ

2019, 39 Sayfa

Jüri
Doç.Dr. Levent SEYFİ
Prof.Dr. Ercan YALDIZ
Dr.Öğr.Üyesi Sabri ALTUNKAYA

Nesne algılama ve tanıma, robotik görüş alanındaki en zorlu araştırma konularından biri olarak kabul edilir. Çevre ve ışık koşullarının değişmesi ile ilgili birçok problemi çözmek için birçok algoritma önerilmiştir. Ancak konu aydınlatma koşulları, dönme açısı, kısmi nesneler vb. gibi tanıma sürecine dahil olan birçok zorluk nedeniyle hala araştırmya açık bir alandır. Bu çalışmada, ORB (Yönlendirilmiş FAST ve Döndürülmüş BRIEF) özelliklerini bulmaya dayalı bir tanıma yöntemi üzerinde çalışılmış, bu özellikler görüntüde belirli köşe noktalarıdır ve bu noktalar, nesnelerin dönüş açısı veya nesne büyüklüğünün değişmesi ile değişmez. Ayrıca bu özellikler aydınlatma değişikliğinden etkilenmez. ORB'nin önemli avantajları bulunmaktadır. ORB, BRIEF (İkili Sağlam Bağımsız Temel Özellikler) ve FAST (hızlı tanımlayıcılar) temelli metodolojileri içerir. İlave olarak, SIFT tanımlayıcı ve SURF tanımlayıcısındaki lisans kısıtlamalarından etkilenmez. Üçüncü olarak, rotasyon ve ölçek değişmez. Son olarak da GPU kullanımına gerek kalmadan düşük güçte elektronik kartlar üzerinde gerçek zamanlı olarak kullanıma imkan tanımaktadır. Bu tezde, özellik çıkarma yöntemi olarak ORB ile bir nesne tanıma sistemi gerçekleştirilmiştir. Nesneleri algılama ve tanıma görevlerini gerçekleştiren bir robot sistemi oluşturulmuştur. Gerçek zamanlı olarak kameradan elde edilen belirli bir video karesinden interset özelliklerini bulmak için ORB kullanılmakta. Ardından Brute-Force eşleştiricisini kullanarak tanıma gerçekleştirilmiştir. Görüntü eşleştirme aşaması için Brute-Force algoritmasında 0.65'lik bir uzaklık oranı kullanılmıştır.

Anahtar Kelimeler: BRIEF, Eşleştirme özellikleri, FAST, Nesne tanıma, ORB, Özellik çıkarımı, RC robot, SIFT, SURF.

ABSTRACT

MS THESIS

DESIGNING AND IMPLEMENTATION OF SEMI AUTONOMOUS MOVING ROBOT WITH OBJECT RECOGNITION SYSTEM CONNECTED TO AN ANDROID DEVICE THROUGH WIRELESS COMMUNICATION

Majeed ABDULMAJEED

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCEOF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN ELECTRICAL-ELECTRONICS ENGINEERING

Advisor: Assoc.Prof.Dr. Levent SEYFİ

2019, 39 Pages

Jury
Assoc.Prof.Dr. Levent SEYFİ
Prof.Dr. Ercan YALDIZ
Dr.Öğr.Üyesi Sabri ALTUNKAYA

The object detection and recognition are considered one of the most challenging research topics in the field of robotic vision. There are many algorithms proposed to solve many problems related to changing of environment and lighting conditions. Many algorithms have been proposed in this topic, but the topic is still to research because of many difficulties included in the recognition process like lighting conditions, the angle of rotation, partial objects, and so on. In this study, a recognition method based on finding the ORB (Oriented FAST and Rotated BRIEF) properties will be studied, these properties are certain corner points in the image and these points do not change with the rotation angle of the objects or the change in the object size. In addition, these features are not affected by lighting changes. ORB has important advantages. ORB includes BRIEF (Binary Robust Independent Elementary Features) and FAST (Features from Accelerated Segment Test) descriptors. In addition, the SIFT identifier and the SURF identifier are not affected by license restrictions. Third, rotation and scale do not change. Finally, it allows real-time use of low-power electronic cards without the need for GPU usage. In this thesis, an object recognition system was realized with ORB as a feature extraction method. A robot system has been created to perform objects of detection and recognition. We used ORB to find interset features in a given video frame obtained from the camera at real-time and then using Brute-force matcher to match a new video frame features with the object image that we want the robot to track it and recognize it. The matched point must have a distance ratio of 0.65.

Keywords: BRIEF, Brute-force, FAST, Feature extraction, Features matching, Object recognition, ORB, RC robot, SIFT, SURF.

ÖNSÖZ

Hazırladığım bu tez çalışmasında, yardımlarını hiçbir zaman esirgemeyen tez danışmanım Doç. Dr. Levent Seyfi'ye, öğrenim hayatım boyunca emeği geçen tüm hocalarıma teşekkürü bir borç bilirim.

Ayrıca bugünlere gelmemde en büyük pay sahibi olan babama, beni her konuda daima destekleyen anneme ve tez yazım esnasında bana katlanan abimler ve ablalarıma teşekkürü bir borç bilirim.

Majeed ABDULMAJEED

KONYA-2019

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **ix**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **3**

3. PYTHON PROGRAMA DİLİ **8**

3.1. Python Özellikleri 8
3.2. Dünyada Python Kullanımı 10

4. ÖZNİTELİK ÇIKARMA ALGORITMALARI **12**

4.1. FAST (KÖŞE ALGILAMA) Hızlandırılmış Segment Testinden Nesneler 12
4.2. BRIEF (İkili Sağlam Bağımsız Temel Öznitelikler) 13
4.3. ORB (Yönlendirilmiş FAST ve Döndürülmüş BRIEF) 15
4.4. Brute-Force Algoritması 16
4.5. HAAR Cascade 16

5. MATERYAL VE YÖNTEM **17**

5.1. Materyal 17
5.1.1. Modül üzeri LattePanda bilgisayar 17
5.1.2. Kamera 18
5.1.3. ZUMO robot shield 19
5.1.4. LattePanda kumanda 19
5.1.5. Raspberry Pi 3 Model B+ 20
5.1.6. Raspberry Pi kamera V2 21
5.1.7. Raspberry Pi motor sürücü kartı 21
5.1.8. Power bank (Taşınabilir şarj aleti) 22
5.1.9. Normal kaynak gücü (AA kalem pil) 22
5.2. Yöntem 23
5.2.1. Birinci çalışmanın sistem tasarımı 23
5.2.2. İkinci çalışmanın sistem tasarımı 25

6. ARAŞTIRMA SONUÇLARI **29**

6.1. Geliştirilen Birinci Sistem 29

6.2. Geliştirilen İkinci Sistem 31
6.3. Birinci ve İkinci Uygulamanın Performansının Karşılaştırılması 33

7. SONUÇLAR VE ÖNERİLER 35

7.1. Sonuçlar 35
7.2. Öneriler 36

KAYNAKLAR 37

ÖZGEÇMİŞ 39

SİMGELER VE KISALTMALAR

Simgeler

θ	: Yamanın Yönelimi
$p(x)$	: Bir Düzleştirilmiş Yama p İçerisindeki Piksel Yoğunluğu
$f_{nd}(P)$	: BRIEF tanımlayıcı
$nd(x, y)$	: Konum Çiftleri Benzersiz Olarak Bir İkilik Değer Testleri Setini
$M_{p\,q}$	: Bir Yamanın Momentleri
C	: Ağırlık Merkezleri
τ	: Eşik değeri

Kısaltmalar

BRIEF	: İkili Sağlam Bağımsız Temel Özellikleri (Binary Robust Independent Elementary Features)
CPU	: Merkezi İşlem Birimi (Central Processing Unit)
DoG	: Gaussian Fark Dedektörü (The Difference of Gaussian Points)
FAST	: Hızlandırılmış Segment Testi Özellikleri (Features from Accelerated Segment Test)
ORB	: Yönlendirilmiş FAST ve Döndürülmüş BRIEF (Oriented FAST and Rotated BRIEF)
SIFT	: Ölçek Değişimsiz Özellik Dönüşümü (Scale Invariant Feature Transform)
SURF	: Hızlandırılmış Sağlam Özellikler (Speeded- Up Robust Features)
OPENCV	:Açık Kaynak Kodlu Bilgisayarla Görme Kütüphanesi
PWM	: Darbe Genişliği Modülasyonu (Pulse Width Modulation)

1. GİRİŞ

Gerçek dünya uygulamalarında nesne tanıma görevleri aydınlatma, rotasyon değişkeni, ölçek değişkeni ve çarpışma alanları gibi karmaşık konulara sahiptir. Bu zorluklar, yakın alandaki karışıklıklardan etkilenmeyecek özellik çıkarma tekniklerinin kullanılmasını gerektirir. Nesne tanıma sistemlerinin zorlukları, bu tür görüntü özelliklerinin başarılı bir şekilde alınamamasından kaynaklanmaktadır. Nesne tanıma zorlu bir bilgisayar görme görevidir, çünkü resim arama, resim otomatik açıklama, sahne algılama ve gerçek zamanlı nesne izleme gibi sayısız uygulamanın önemli bir parçasıdır. Son zamanlarda, model tabanlı, şekil tabanlı ve görünüm tabanlı birçok yöntem geliştirilmiştir. Nesne tanıma işlemlerinde, özellik çıkarma teknikleri ve yerel görünüm temelli yaklaşımlar önemli rol oynamaktadır.

Yerel görünüm tabanlı nesne tanıma sistemleri, görüntüdeki ayrı bölgelerde çalışır ve bu bölgeleri sabit bir şekilde tespit edebilmesi zorlu bir görevdir. Bir bölge detektörü görüntü içinde tam olarak bir pozisyon bilgisi verirse, onu ilgi noktası detektörü olarak da adlandırırız. İdeal bölge dedektörleri, ayrıca ilgilenilen bir bölgenin şeklini (ölçek) ve yönünü de verir. Bölge dedektörleri köşe tabanlı ve bölge tabanlı dedektörler olmak üzere iki kategoriye ayrılabilir. Köşe tabanlı dedektörler ilgi noktaları ve kenarlar gibi çok sayıda görüntü yapısı taşıyan bölgeler bulur, fakat düzgün değişimlere sahip tek biçimli alanlar ve bölgeler için tatmin edici sonuçlar verebilir. Bölge tabanlı dedektörler, görüntünün en dikkat çekici özellikleri ile benzer parlaklıkta yerel lekeleri algılar. Böylece farklı uygulamalarda kullanım için uyarlanabilir. Örneğin, bir bölgenin entropisini dikkate alarak insanın görsel dikkatini taklit etmeye çalışır.

Yeterli performans sağlayan popüler algoritmalar aşağıdaki gibi sıralanabilir:

- Harris veya Hessian nokta bazlı dedektörler (Harris, Harris-Laplace, Hessian Laplace)
- Gaussian Fark dedektörü (DoG),
- Harris veya Hessian Affin değişmez bölge detektörleri (Harris-Affine),
- Maksimum Kararlı Ekstrem Bölgeler (MSER),
- Entropi Tabanlı Dikkat Çeken Bölge dedektörü (EBSR) ve
- Yoğunluk Tabanlı Bölgeler ve Kenar Tabanlı Bölgeler (IBR, EBR).

(Rublee ve ark., 2011) yaptıkları çalışmada sundukları algoritma (ORB) SIFT (Ölçek Değişimsiz Özellik Dönüşümü) ve SURF'a (Hızlandırılmış Sağlam Özellikler)

hesaplama maliyetinde ve eşleştirme performansında mükemmel bir alternatiftir. OpenCV, bilgisayar görme kütüphanesinin bir parçası olarak sunulan ücretsiz ve açık kaynak algoritmasıdır. ORB performansı artırmak için yapılan bazı değişikliklerle FAST (Hızlandırılmış Segment Testi Özellikleri) anahtar noktaları dedektörü ve BRIEF (İkili Sağlam Bağımsız Temel Özellikleri) tanımlayıcısına dayanmaktadır.

Bu tez çalışmasında, özellik çıkarma yöntemi olarak köşe tabanlı temel nokta dedektör tekniği olan Yönlendirilmiş FAST (hızlandırılmış segment testi özellikleri) ve Döndürülmüş BRIEF (İkili Sağlam Bağımsız Temel Özellikleri) yöntemlerini içeren ORB algoritması kullanılmıştır. ORB algoritması bu çalışmada Brute force eşleştirme yöntemiyle beraber kullanılmıştır. Tasarlanan sistem, kameradan temin edilen video karelerinden gerçek zamanlı olarak ayarlanan anahtar noktaları algılar ve Brute Force algoritmasını kullanarak referans nesne görüntüsüyle eşleştirir (algılanan anahtar noktalar arasındaki mesafe oranının 0.65 olması ve diğer tüm anahtar noktaları atlaması gerekir).

2. KAYNAK ARAŞTIRMASI

Ramesh ve Mohan (2007) dağınık grup ortamı boyunca ilerleyen gevşek eşleşmiş aşamalar dizisi kullanan bir sistem tasarlamıştır. Bu sistemin en düşük seviyesi, beyinde var olan biyolojik özelliğe paraleldir ve üst seviyelerde ise bilişsel algılama aşamalarını en basit şekilde taklit etmeye çalışır.

Rane ve ark. (2013) yaptıkları çalışmada farklı nesne sınıflarını kategorize etmek için şekil bağlamı, renk histogramı ve tamamlanmış yerel ikili örüntü (CLBP) yaklaşımlarını tasarlamıştır. Şekil bağlamı, diğer yaklaşımlardan daha iyi olan ETH-80 veritabanı hakkında %93 doğruluk sağlar. Elde edilen sonuçlar, daha iyi özellik çıkarıcıları, şekil ve renk tanımlayıcı veya şekil ve doku tanımlayıcı kombinasyonu kullanılarak daha da iyileştirilebilir.

Kim ve Kweon (2007) yaptıkları çalışmada görsel sözcükler çerçevesindeki yüzey işaretlerine odaklanan bir nesne sınıflandırma yöntemi tasarlamıştır. Bu yöntem ile kod çizelgeleri entropisine dayalı yüzey işaretlerinin etkisi en aza indirilebilir. Sınıflar arası kod çizelgesindeki yüksek entropi, birinci öğrenme aşamasında yüzey işaretleme kısımlarını (düşük entropi) kaldırabilir. Ayrıca, kategoriye özgü kod çizelgesinden sınıflar arası kod çizelgesi entropisi ile yönlendirilen ayırt edici kod çizelgesi de seçilmiştir. Yüksek entropi kod çizelgesi isimsiz sınıf etiketleri verdiği için ilk önce kaldırılır. Sonra, farklı mesafeler ile En Yakın Komşu Sınıflandırıcı (NNC) ve Destek Vektör Makinesi (SVM) sınıflandırıcılarını kullanarak bu kod çizelgeleri değerlendirilir. Optimum özellikler, kod çizelgeleri ve sınıflandırıcı dizileri ile görsel sözcük çerçeve torbasında yükseltilmiş performans elde edebilirler. Bu kod çizelgesi ile seçim ve sınıflandırma çalışması diğer karmaşık kategorizasyon yöntemlerine de uygulanabilir.

Otoom ve ark. (2008) yaptıkları çalışmada terk edilmiş nesnelerin doğru sınıflandırmasını ortaya koyan en iyi özellik dizisini belirlemek amacıyla farklı özellik dizilerinin performansını değerlendirmiştir. Elde edilen deneysel sonuçlarla belirlenen geometrik temel özellik istatistikleri kullanılarak yapılan sınıflandırma sonuçlarının farklı sınıflandırma ve değerlendirme şemaları kullanan SIFT (Ölçek Sabiti Görüntü Dönüştürme) önemli noktalarına dayalı diğer iki yaklaşımı uyguladığı sonucuna varmışlardır. 10 katlı çapraz geçerlikli geometrik temel özellik istatistiklerine dayalı sınıflandırma, SIFT önemli nokta histogramlarına dayalı ikinci en iyi yaklaşımla

kıyaslandığında ortalama %22 daha yüksek tanıma doğruluğu ve %7 daha düşük yanlış alarm sağlar. Bu çalışmada sunulan örnek analiz, geometrik temel özellik istatistiklerinin sınıflar arasındaki ayrımı maksimuma çıkardığını ve sınıflandırma sürecini basitleştirdiğini kanıtlamıştır.

Shobha ve Mudgal (2012) yaptıkları çalışmada önemli çerçeveleri çıkarmak ve hareket eden bir nesneyi algılamak için yöntem tasarlamıştır. Çalışmalarında içerik bazlı video için hareket eden nesne izleme teknolojisini uyguladılar. Hareket eden pikselleri algılamak için de geri plan çıkarım algoritmasını kullandılar. Hareket eden nesne algılaması dışında videodaki gölgelerin giderilmesi şeklinde işlem de yapılabileceği belirtilmiş.

Yang ve ark. (2008) görsel kod çizelgesi üretimini nesne sınıflandırıcı eğitimi ile birleştiren yeni bir çerçeve tasarlamıştır. Çalışmalarında geliştirdikleri iki önemli özellik, bu çalışmayı mevcut nesne sınıflama yaklaşımlarından ayırır. Birincisi, her bir görüntü özelliğini tek bir görsel sözcük ile ilişkilendiren gruplandırma yaklaşımının aksine, her bir görüntü özelliğini görsel bitler vektörü ile kodlamasıdır. İkincisi, görsel kod çizelgesi üretimi ve sınıflandırıcı eğitimi süreçlerini ayıran standart uygulamanın aksine, tasarlanan yaklaşımın bu iki süreci objektif bir işlev altında tek bir optimizasyon çerçevesinde birleştirmesidir. Optimum görsel bitler ile bunların ilişkilendirilmiş ağırlıklarını etkili bir şekilde tanımlamak için yinelemeli bir algoritma tasarlanmıştır. Tasarlanan birleştirilmiş yaklaşım önceki yaklaşımların kayda değer oranda üzerinde bir performans göstermiştir.

Karami ve ark. (2017) yaptıkları çalışmada, ölçekleme, döndürme, gürültü, balıkgözü bozulma ve kesme gibi farklı dönüşüm ve deformasyon türleri için üç farklı görüntü eşleştirme tekniğini (SIFT, SURF ve ORB) karşılaştırmıştır. Bu amaçla, orijinal görüntülere farklı dönüşüm türleri uygulanmıştır. Ayrıca, görüntüdeki kilit nokta sayısı, eşleşme oranı ve her algoritma için gereken yürütme süresi gibi eşleştirme değerlendirme parametreleri gösterilmiştir. ORB'nin en hızlı algoritma olduğu, SIFT'in ise çoğu senaryoda en iyisi sonucu elde ettiği gösterilmiştir. Dönme açısının 90 derece ile orantılı olduğu özel durumlar için, ORB ve SURF, SIFT'den daha iyi performans gösterirken gürültülü görüntülerde ORB ve SIFT neredeyse benzer performanslar gösterir. ORB'de, özellikler çoğunlukla görüntünün ortasındaki nesnelerde yoğunlaştırılırken, SURF, SIFT ve FAST anahtar nokta dedektörleri görüntünün üzerine dağıtılır.

Chang ve Krumm (1999) yaptıkları çalışmada görüntülerdeki nesneleri tanımanın etkili bir yöntemi olarak renk Histogramını (CH) tasarlamıştır. Bu yöntemde, Piksel çiftlerini takip ederek, normal renk histogramına değişken miktarda geometri eklenmesini sağlar. Algoritma yanlış alarm oranını düşürmek için analiz edebilir ve bu algoritmanın ayarlanabilir parametrelerinin bazılarını ayarlamak için kullanılabilir. Bu çalışmada algoritmanın karmaşık bir yapısı olmasına ve nesne bükülmesine rağmen çalıştığı gösterilmiştir.

Rosten ve ark. (2010) yaptıkları çalışmada, çok hızlı, yüksek kaliteli bir köşe dedektörü türetmek için makine öğrenmesini kullanmışlardır. Geliştirilen dedektörün avantajları bulunmaktadır. Diğer köşe dedektörlerinden çok daha hızlıdır ve büyük boyut değişikliklerinde ve farklı özelliklerde yüksek tekrarlanabilirlik seviyeleri sunar. Bununla birlikte, bazı dezavantajları da vardır. Yüksek seviyede gürültüye dayanıklı değildir. Ayrıca, dairenin niceliği çizgiyi kaçırdığında, belirli açılarda 1 piksel geniş çizgiye cevap verebilir. Bir diğer dezavantajı da bir eşiğe bağlı olmasıdır. Ayrıca köşelerin tekrarlanabilirliğini değerlendirmek için bir yöntem kullanarak diğer makalelerde yapılan birtakım iddiaları da doğrulanabilmiştir ve bu değerlendirmede sadece düzlemsel sahnelerden daha fazlasını kullanmanın önemi gösterilmiştir. Gerçek zamanlı kare hızı uygulamalarında özellik noktalarının kullanıldığı yerlerde, yüksek hızlı bir özellik dedektörü gereklidir. SIFT (DoG), Harris ve SUSAN gibi özellik dedektörleri, yüksek kaliteli özellikler sağlayan iyi yöntemlerdir, ancak herhangi bir karmaşıklıktaki gerçek zamanlı uygulamalarda kullanım için çok fazla hesaplama gerektirirler. Burada, makine öğreniminin, mevcut işlem süresinin %7'sinden daha azını kullanarak canlı PAL videoyu tam olarak işleyebilen bir özellik dedektörü türetmek için kullanılabileceği gösterilmiştir. Karşılaştırma yapıldığında, Harris detektörü (%120) ve SIFT'in (%300) tespit aşaması tam kare hızında çalışamaz. Açıkça görüldüğü gibi, eğer üretilen özellikler aşağı işlem için uygun değilse, yüksek hızlı bir dedektörün kullanımı sınırlıdır. Bu nedenle, bu makalenin diğer bir katkısı, 3D sahnelere uygulanan bu kriteri temel alan bir karşılaştırma köşesi dedektörünü sunmasıdır. Bu karşılaştırma, mevcut köşe dedektörleriyle ilgili olarak başka çalışmalarda sunulan ifadeleri de desteklemektedir. Ayrıca, ilk beklentilerin tersine, ilke olarak hız için yapılmış olmasına rağmen, geliştirilen dedektörün mevcut özellik dedektörlerini bu kritere göre önemli ölçüde geride bıraktığı görülmüştür.

Belongie ve ark. (2001) şekiller arasındaki benzerliği ölçmek için yeni bir yaklaşım sunmuştur ve bunu nesnenin tanınması için kullanmıştır. Çerçevede benzerlik

ölçümünden önce iki şekildeki noktalar arasındaki benzeşimi çözmek ve bir hizalama dönüşümü oluşturmak için benzeşimleri kullanmak gerekir. Benzeşme problemi çözüldükten sonra her noktaya "şekil bağlamı" tanımı eklenir. Bir referans noktadaki şekil bağlamı, kalan noktaların buna göre dağılımını gösterir. Benzer iki şekildeki benzeşen noktalar, benzer şekil bağlamlarına sahip olacaktır. Bu sayede benzeşmeler bir optimum atama problemi olarak çözülebilecektir. Noktaların benzeşiminde, ikisi şekli en iyi düzenleyen dönüşüm belirlenmiştir. Düzenlenmiş ince tabaka şeritler, bu amaç için esnek bir dönüşüm haritalama sınıfı sağlar. İki şekil arasındaki farklılık, karşılık gelen noktalar arasındaki eşleşen hataların toplamı olarak ve aynı zamanda hizalama dönüşümünün büyüklüğünü ölçen bir terim olarak hesaplanır. Tanıma işlemi sınıflandırma çerçevesine en yakın komşuluk bölgesinde ele alınmıştır. Sonuçlar silüetler, ticari markalar, el yazısı rakamları ve COIL veri kümesi için sunulmuştur.

Opelt ve ark. (2006) yaptıkları çalışmada uçaklar veya atlar gibi nesne sınıflarının tespitini amaçlamışlardır. Göze çarpan görüntü parçalarına dayalı bir model kullanmak yerine, yalnızca nesnenin sınırını kullanarak nesne sınıfı tespitinin de mümkün olduğunu göstermişlerdir. Bu amaçla, sınıf ayırıcı sınır parçaları çıkarmak için yeni bir öğrenme tekniği geliştirmişlerdir. Yükseltme, güçlü bir "sınır parçası modeli" (BFM) detektörü oluşturmak için sınır parçalarının (zayıf dedektörler) ayrımcı kombinasyonlarını seçmek için kullanılır. Modelin üretken yönü bölümleme yapmak için kullanılır. Çalışmanın sonucu olarak, BFM dedektörü görünüşlerinden ziyade esas olarak şekilleriyle tanımlanan nesne sınıflarını temsil edebildiği ve algılayabildiği ortaya konmuştur. Ayrıca, çeşitli nesne sınıflarında (uçaklar, arabalar, inekler) yayınlanan diğer sonuçlarla karşılaştırıldığında, BFM dedektörü önceki performansları aşabildiği ve bunu daha az denetimle (eğitim görüntülerinin sayısı gibi) başarabildiği gözlenmiştir. Nesne kategorilerinin örneklerini tespit etmek için bir Sınır Parça Modeli tanımlanmıştır. Yöntem, tipik olarak bir kenar detektörü ile geri kazanılan kısmi sınırlarla başa çıkabilir. Performansı, gelişmiş yöntemlere benzer veya daha üstündür. Dokunun çok değişken olduğu sınıflar için (örneğin şişeler, bardaklar) bir BFM tercih edilebilir. Diğer durumlarda görünüm ve sınırın bir arada kullanılması üstün performans sağlayacaktır.

Chaudhari ve Patil (2015) yaptıkları çalışmada bir nesnenin tanınması ve sınıflandırılması için dijital görüntü işleme ve makine öğrenme tekniklerini kullanmayı amaçlamışlardır. Bu makalede ayrıca farklı nesne tanıma yaklaşımları ve sınıflandırma

yöntemleri hakkında anket yapılmıştır. Bu anket farklı tanıma ve sınıflandırma yaklaşımları hakkında teorik bilgi sağlamıştır.

Song ve ark. (2011) yaptıkları çalışmada, hem nesne algılama hem de sınıflandırma görevlerinin performansını karşılıklı olarak artırmak için yinelemeli bir bağlamlaştırma düzeni önerilmiştir. İlk olarak genel sınıflandırma için dış bağlam özelliklerini ve konu özelliklerini sorunsuz bir şekilde entegre etmek için Bağlamlaştırılmış SVM önerilmiştir ve daha sonra bağlamsal-SVM, nesne belirleme ve sınıflandırma görevlerinin performansını yinelemeli ve karşılıklı olarak arttırmak için kullanılmıştır. Önerilen çözüm, hem PASCAL VOC 2007 hem de VOC 2010 veri setleri üzerinde kapsamlı bir şekilde değerlendirilmiş ve her iki görev için de yüksek performansa ulaşılmıştır.

Khurana ve Awasthi (2013) yaptıkları çalışmada bir görüntüdeki birden fazla nesnenin tespitine yönelik bir yöntem sunmuşlardır. Ayrıca nesne algılama tekniklerinden bahsetmişlerdir.

Leutenegger ve ark. (2011) yaptıkları çalışmada kilit nokta tespiti, açıklama ve eşleştirme için yeni bir yöntem olan BRISK'i önermişlerdir. Kıyaslama veri setleri üzerine yapılan kapsamlı bir değerlendirme, BRISK'in en gelişmiş algoritmalarda olduğu gibi uyarlanabilir, yüksek kaliteli performansını, hesaplama maliyetlerinde önemli ölçüde düşük olsa da ortaya koymaktadır (SURF'tan daha hızlıdır). Bu çalışmada BRİSK adlı yöntem, sahne ve kamera pozları hakkında yeterli bilgi sahibi olmayan durumlar için görüntü kilit noktalarını tespit etme, tanımlama ve eşleştirme ile klasik bilgisayar görme problemini çözmektedir. SIFT ve SURF gibi kanıtlanmış yüksek performansa sahip yerleşik algoritmaların aksine eldeki yöntem karşılaştırılabilir eşleştirme performansında, belirlenmiş bir çerçeveyi kullanarak kapsamlı bir değerlendirmeye dayanan bir ifade ile çarpıcı biçimde daha hızlı bir alternatif sunmaktadır. Yöntem ikili bir tanımlayıcı dizesi oluşturmak için parlaklık karşılaştırmalarının hesaplandığı kolayca yapılandırılabilir bir dairesel örnekleme desenine dayanır.

3. PYTHON PROGRAMA DİLİ

Python, Guido Rossum tarafından 1991'da geliştirilen nesne yönelimli bir programlama dilidir. Karmaşık uygulamaların hızla gerçekleştirilebilmesi amaçlanarak oluşturulmuştur. Pek çok işletim sistemi ile birleştirilebilir arayüzleri bulunmaktadır. Ayrıca, C veya C ++ 'a genişletilebilir bir programlama aracıdır. NASA, Google, YouTube, BitTorrent gibi birçok büyük firma Python programlama dilini projelerinde aktif olarak kullanmaktadır. Python, yapay zeka, yapay sinir ağları ve diğer bilgisayar bilimi konularında yaygın olarak kullanılmaktadır. Python'da yazılmış programların çalıştırılması için diğer programlama dillerinden farklı olarak önceden derlenmesi gerekmez. Bu durum küçük kod parçacıklarını test etmeyi ve başka platformlara taşınmasını kolaylaştırır.

3.1. Python Özellikleri

Python dilinin en önemli özellikleri aşağıdaki gibi sıralanabilir:

- Acemi kullanıcılara uygun,
- Basit içerikli ve öğrenmesi kolay,
- Yorumlanmış bir dil,
- Çapraz platform dili,
- Kullanımı ücretsiz ve açık kaynak olarak sunulur,
- Nesne yönelimli bir dil,
- Kapsamlı kütüphaneleri bulundurur,
- Veritabanları ile bağlantısı bulunur.

Python dilini öğrenmeye yeni başlayanlar, döngüler ve prosedürler gibi temel kavramları hızlı bir şekilde kavrayabilir. Python'un söz dizimindeki basitliği, yeni kullanıcıların bile gelişmiş programlama kavramlarını kullanmasına olanak tanır, örneğin ekranda çıkış göstermesi için aşağıdaki gibi bir satırlık kod yeterli olur:

```
>>> print ("Merhaba dünya ")
Merhaba dünya
>>>
```

Basit kuruluma ve web araçları oluşturmada birçok gelişmiş uygulamaya sahiptir. Diğer dillere kıyasla yazılacak kodu daha kısa hale getirebilmek için kullanılabilecek bir çok modül kullanabilir. Python'da özellikle interaktif tercüman ile çalışmak için mükemmel ve basit araçlar vardır.

Python kullanmaya başlamak için bir derleme sistemi, IDE, özel metin editörü veya başka bir şey öğrenilmesi gerekmemektedir. Python'da kod çalıştırılırken dahili olarak kaynak kodu, "bytecodes" adı verilen bir ara forma dönüştürülür ve daha sonra bilgisayarın ana diline çevirilerek çalıştırılır. Bu aşamada kütüphanelerle bağlantı kurmak veya kütüphaneleri yüklemek gerekliliği bulunmadan sadece programın çalıştırılması yeterli olmaktadır.

Python, Windows, Linux, Unix, Macintosh gibi farklı sistemlerde aynı performansla çalışabilmektedir. Örneğin, Macintosh bir bilgisayarda yazılmış Python programı Linux kurulu bir bilgisayarda rahatlıkla çalışabilecektir. Bu durumun tersi de geçerlidir. Bu sayede, Python taşınabilir bir özelliğe sahip olabilmektedir. Nesne yönelimli özellikleri destekler. Diğer programlama dilleriyle mukayese edildiğinde, Python'un sınıf mekanizması C ++ ve Modula-3'te bulunan sınıf mekanizmalarının bir karışımıdır ve en az yeni söz dizimi ve anlambilim içeren sınıfları içermektedir. Python sınıfları, nesne yönelimli programlama dilinin standart özelliklerinin neredeyse tümünü sağlar.

Python standart kütüphanesi çok büyüktür. Diğer programlama dillerinde erişilemeyecek olan dosya ve I/O gibi sistem işlevlerine erişim sağlayan yerleşik modüller (C ile yazılmış) içermektedir. Ayrıca, birçok problem için Python'da standartlaştırılmış çözümler sunan modülleri içermektedir. Python programlama dili C, C ++, JAVA gibi dillerle kolayca entegre edilebilir ve tüm ticari veritabanlarına arayüz sağlayabilir.

Standart kütüphane, ihtiyaçlarımıza göre çok çeşitli modüller arasından seçim yapmamızı sağlamaktadır. Her modül, ek kod yazmadan Python uygulamasına işlev eklememizi sağlamaktadır. Örneğin, Python'da bir web uygulaması yazarken, web servislerini uygulamak, string işlemleri yapmak, işletim sistemi arayüzünü yönetmek veya internet protokolleriyle çalışmak için belirli modüller kullanılabilir. Python Standart Kütüphanesi belgeleri incelendiğinde kullanılabilecek modüller görülebilir. Açık kaynak kodlu bir programlama dili olan Python, yazılım geliştirme maliyetini önemli ölçüde azaltmaktadır. İhtiyaca göre çok çeşitli açık kaynak Python çerçeveleri ve geliştirme araçları seçilebilmektedir. Örneğin, Django, Flask, Pyramid, Bottle ve

Cherrypy gibi Python web çerçeveleri ile web uygulaması geliştirme hızlandırılabilir. Ayrıca, Python GUI çerçeveleri ve PyQT, PyJs, PyGUI, Kivy, PyGTK, WxPython gibi araç takımları kullanılarak GUI uygulama geliştirmesi hızlandırılabilmektedir.

3.2. Dünyada Python Kullanımı

PYPL (PopularitY of Programming Language) programlama dili sıralamasına göre Python Şekil 3.1'de görüldüğü üzere 1. sırada bulunmaktadır. PYPL'de puanlama, Google'da yapılan arama sayısına bağlı olarak yapılmaktadır. Bir dil ne kadar çok aranırsa, dilin o kadar popüler olduğu varsayılmaktadır.

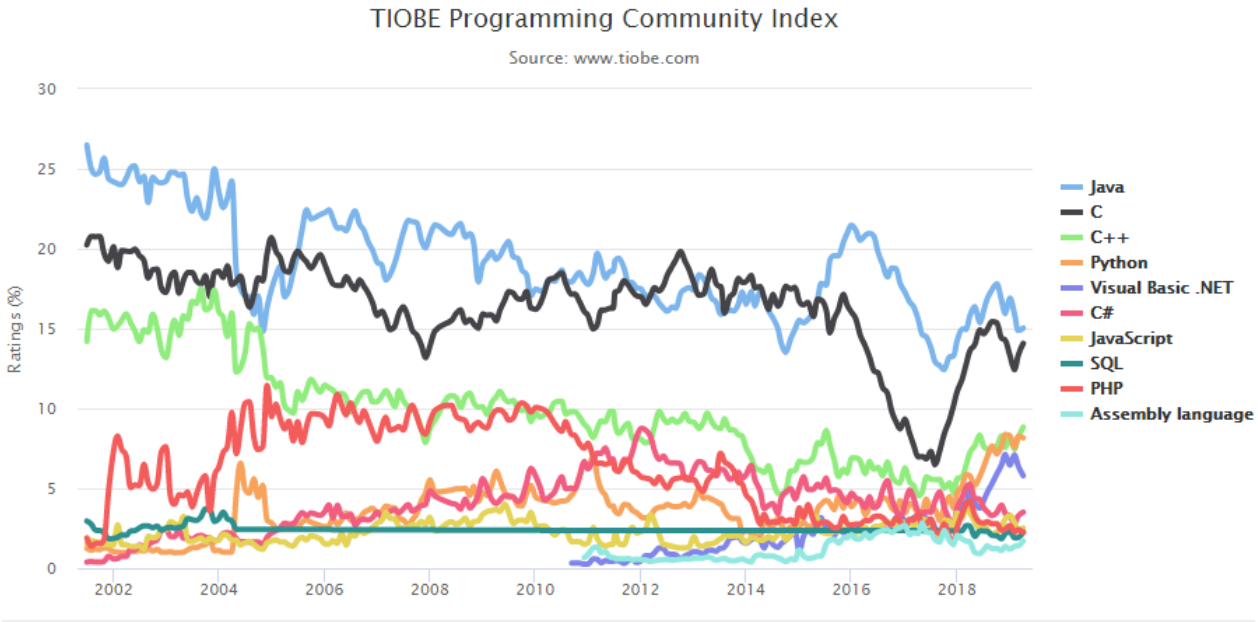
Worldwide, Apr 2019 compared to a year ago:

Rank	Change	Language	Share	Trend
1	↑	Python	27.35 %	+5.2 %
2	↓	Java	20.64 %	-1.8 %
3		Javascript	8.4 %	-0.2 %
4	↑	C#	7.45 %	-0.5 %
5	↓	PHP	7.18 %	-1.1 %
6		C/C++	6.08 %	-0.3 %
7		R	4.18 %	-0.1 %
8		Objective-C	2.86 %	-0.8 %
9		Swift	2.47 %	-0.4 %
10		Matlab	2.06 %	-0.3 %
11	↑	TypeScript	1.64 %	+0.2 %
12	↓	Ruby	1.39 %	-0.3 %
13		VBA	1.39 %	+0.0 %
14	↑↑	Kotlin	1.18 %	+0.3 %
15		Scala	1.16 %	-0.0 %

Şekil 3.1. (PYPL) Programlama dilinin popülerliği (Anonymous, 2019a)

TIOBE ("The Importance Of Being Earnest") tarafından belirlenen listeye göre ise Python, 2019 yılının en popüler 10 programlama dilinden biridir. TIOBE, programlama dillerinin kalitesini ölçen bir kuruluştur. Oluşturulan liste düzenli olarak güncellenir. Puanlamalar yapılırken ilgili programlama dillerinin dünya genelindeki yetenekli mühendislerinin sayısı, kurslar ve üçüncü taraf satıcılara bakılmaktadır. Google, Bing, Yahoo, Wikipedia, Amazon, YouTube ve Baidu gibi popüler web

uygulamaları da puanlamaları hesaplamak için kullanılır. Şekil 3.2'de TIOBE tarafından yapılan programlama dillerinin yıllara göre derecelendirilmesi görülebilmektedir.



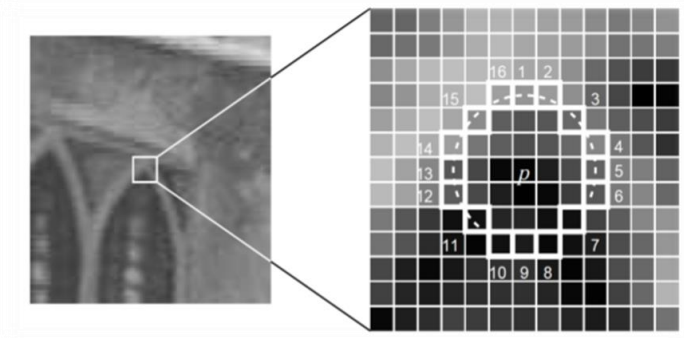
Şekil 3.2. Programlama dillerinin yıllara göre popülerliği (Anonymous, 2019b)

4. ÖZNİTELİK ÇIKARMA ALGORITMALARI

Bu bölümde, öznitelik çıkarma için kullanılan FAST, BRIEF, ORB, Brute Force ve HAAR Cascade algoritmalarından bahsedilecektir.

4.1. FAST (KÖŞE ALGILAMA) Hızlandırılmış Segment Testinden Nesneler

FAST, ilk defa Rosten ve Drummond tarafından tanıtılmıştır. Ana amacı görüntüdeki belirli noktaları bulmaktır. Bu noktalar köşelerdir, fakat bu köşeler görüntüdeki pek çok köşe arasında en güçlü olanlardır. En güçlü köşe noktaları, ölçek değişiminden etkilenmeyecekleri anlamına gelir. Yönelim ve aydınlatmadan bağımsız olduğu ve uygulama kolaylığına sahip olduğu için fast algoritmasını kullanarak köşeleri bulma yöntemi son derece verimlidir. Sadece pikseller seviyesinde çalışır. Bu yöntem aşağıdaki adımlarla özetlenebilir: Herhangi bir görüntüde (p) adlı bir piksel test edilen noktadır. Eşik değerinin (T) örneğin %30 olarak ayarlanması (Kullanım alanı daha sonra açıklanacaktır. Şekil 4.1'de gösterildiği üzere test edilen pikseli (p) çevreleyen daireler şeklinde (16) pikseller seçilirken pikselin (p) diğer piksellerle arasındaki mesafe (3) piksel olmalıdır (bu düzenleme Bresenham dairesi olarak adlandırılır). Çevreleyen pikseller (p-T) veya (p+T) değerinin üzerinde veya altında (N)komşu piksel değerine sahipse, noktanın (p) bir ilgi noktası olduğu söylenebilir. Bir köşenin varlığını daha hızlı bir şekilde algılayabilmek için, noktanın kontrol mü edileceği yoksa göz ardı mı edileceği kararını hemen vermek için bir başlangıç testi yapılmalı ve bu test çevreleyen dairedeki (1), (5), (9), (13) noktaları kontrol edilerek gerçekleştirilmelidir. En az 3 nokta (p-T) veya (p+T) altında veya üzerinde değilse, nokta bir köşe değildir ve ayrıntılı testi yapmadan önce göz ardı edilecektir. Bu prosedür aşağıda Şekil 4.1 'de gösterilmektedir (Rosten ve ark., 2010):



Şekil 4.1. İlgi noktasını algılamak için FAST köşe algılama

Komşu piksel sayısının (N) seçilmesi bir ölçüde kafa karıştırıcıdır, çünkü 12'den az olduğu takdirde bu durum çok sayıda pikselin seçileceği anlamına gelir. Bazı çalışmalarda N=9 değerinin köşe algılama için optimum değer olduğu söylenmesine rağmen bu algoritmanın doğruluğunu etkileyebilmektedir.

Şekil 4.2. FAST köşe algılama

Şekil 4.2'de görüntüde tüm köşelerin değil sadece algoritmanın koşullarını karşılayan köşelerin seçildiğini görebiliriz. FAST köşe algılamanın önemi uygulama kolaylığından gelir, dolayısıyla çevrimiçi uygulamalar için performansı çok yüksek olacaktır.

4.2. BRIEF (İkili Sağlam Bağımsız Temel Öznitelikler)

Öznitelik dönüştürme yöntemleri ana ilgi noktalarını tarif etmek için son derece önemli yöntemler olup en önemli yöntemler SURF ve SIFT'tir. Bunlar önemli ilgi noktalarının konumunu belirleme konusunda neredeyse mükemmeldir, fakat SIFT'in tanımlayıcılar için 128-dim vektör kullanması şeklinde önemli bir probleme sahiptir. Yüzen nokta sayıları kullandığı bilinmesinden ötürü temel olarak 512 bayt kaplayacaktır. Benzer şekilde SURF de minimum 256 bayt kaplamaktadır (64-dim için). Binlerce öznitelik için çok büyük bir vektör yaratmak çok fazla hafıza kaplar ve çok fazla zaman alacağı için bu durum tümleşik sistemler için verimli değildir. BRIEF yöntemi, göreceli olarak az sayıda ikili yoğunluk karşılaştırmalarını alarak görüntü yamalarının verimli bir şekilde sınıflandırılabileceğini göstermektedir. Bu testlerin sonuçları rastgele sınıflandırma ağaçlarını veya bir Naive Bayes sınıflandırıcıyı çoklu

bakış açılarından görülen yamaları tanımak üzere eğitmek için kullanılmıştır. BRIEF yönteminde hem sınıflandırıcı hem de ağaçlar kullanılmaktadır ve basit olarak test yanıtlarından, görüntü yamasını düzleştirdikten sonra hesaplayacağı bir vektör yaratmaktadır (Calonder ve ark., 2010).

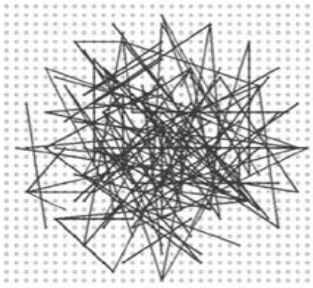
Test, S × S boyutlarına sahip yama p üzerinde τ değerini şu şekilde tanımlamaktadır:

$$\tau(P; x, y) := \begin{cases} 1, & P(x) < P(y) \text{ ise} \\ 0, & \text{Aksi halde} \end{cases} \quad (4.1)$$

Burada P(x), x = (u, v)'te düzleştirilmiş yama p'deki piksel yoğunluğudur. Bir nd (x, y)- seti seçildiğinde konum çiftleri benzersiz olarak bir ikilik değer testleri setini tanımlar. BRIEF tanımlayıcının nd- boyutsal bit dizgisi olarak düşünülmüştür**.**

$$f_{nd}(P) := \sum_{1 \le i \le nd} 2^{i-1} \tau(P; x, y) \quad (4.2)$$

Araştırmada nd = 128, 256 ve 512 olarak düşünülmüş olup bunun hız, depolama verimliliği ve tanıma oranı bakış açısından iyi sonuçlar vereceği görülmektedir. Bazen BRIEF tanımlayıcılar, k = nd/8 değerinin tanımlayıcıyı depolamak için gereken bayt sayısını temsil ettiği BRIEF-k şeklinde de gösterilebilir. Aşağıda yer alan Şekil 4.3 BRIEF örnekleme düzenini göstermektedir**.**



Şekil 4.3. Köşe noktaları tanımlayacak BRIEF örnekleme düzeni

4.3. ORB (Yönlendirilmiş FAST ve Döndürülmüş BRIEF)

Yönelimli brief (ORB), (BREIF) tabanında bir tanımlayıcıdır. İyi bilinenleri (SIFT ve SURF) karşılaştırma konusunda son derece hızlıdır. Döndürmeden bağımsızdır ve gürültüye karşı dirençlidir. Deneyler ORB'nin (SIFT ve SURF)'ten daha hızlı iki büyüklük kertesine sahip olduğunu göstermektedir. ORB'nin temel avantajı, GPU kullanma ihtiyacı olmadan tümleşik sistemler ve cep telefonları üzerindeki gerçek zamanlı uygulama için kullanılabilmesidir. ORB'nin ana fikri, FAST köşe algılamayı bir yönelim bileşeniyle birlikte sağlamaktır. Öncelikle, 9 sayısının komşu pikseli (N=9) ifade ettiği FAST-9 şeklindeki belirli bir FAST türü ele alındığında, deney sonuçları bu değerin çok iyi performans sonuçları verdiğini göstermiştir. Kilit noktaları seçme konusundaki temel husus, köşelerin (Nval) değeri elde edilmek istendiğinde, FAST eşik değerini (Nval) değerini aşan değerler elde edene dek düşürmemiz gerektiğidir. Ardından, alınan köşeleri sıralamak için Harris köşesi ölçeği kullanılmalıdır ve sonrasında da bir eşit köşeler sayısı (Nval) seçilmelidir. FAST çok ölçekli öznitelikler sağlamaz. Bu nedenle, ölçek piramidi için köşe algılama sağlamak amacıyla çok ölçekli alana yönelik olarak ORB için FAST gerçekleştirilmelidir. ORB, yönelim açısını hesaplamak için (yoğunluk ağırlık merkezi) kavramını kullanmaktadır. Yoğunluk ağırlık merkezi, bir köşeye ait yoğunluğun merkezinden kaymış olduğunu varsaymaktadır. Dolayısıyla bu vektör yönelimi hesaplamak için kullanılabilir. Rosin bir yamanın momentlerini şu şekilde tanımlamaktadır (Rublee ve ark., 2011):

$$M_{p\,q}=\sum_{x,\,y} \quad x^{p}\, y^{p}\, I(x, y) \tag{4.3}$$

Ve bu momentleri kullanarak aşağıdaki ağırlık merkezlerini elde edilebilir**:**

$$C= ((m10/m00), (m01/m00)) \tag{4.4}$$

Yönelim açısını bulmak için köşelerin merkezinden (O) ağırlık merkezine (C) olan vektör bulunmalıdır. Yamanın OC yönelimi şu şekildedir**:**

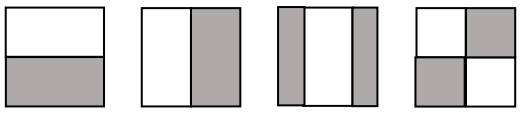
$$\varphi = \text{atan2}\ (m01, m10) \tag{4.5}$$

4.4. Brute-Force Algoritması

Brute-Force, genellikle doğrudan sorunun ifadesine ve ilgili kavramların tanımlarına dayanarak, Problemi çözmek için basit bir yaklaşımdır. Genel olarak, geçerli bir çözüm bulunana kadar olası tüm çözümleri gözden geçirmeyi içermektedir. Tutarsız gibi görünse de, çoğu durumda Brute-Force, en iyi yoldur, çünkü sorunu çözmek için bilgisayarın hızına güvenebiliriz. Brute-Force algoritmaları, daha karmaşık algoritmalarımızı karşılaştırmamız için güzel bir temel sunmaktadır. Basit bir örnek olarak, bazı hedefler için sıralanan öğeler listesinde arama yapmaktadır. Brute-Force basitçe ilk maddeden başlayacak, hedef olup olmadığını görecek veya hedefi bulmayacak ya da listenin sonuna vurana kadar sırayla ilerlemektedir. Küçük listeler için bu sorun değildir (ve aslında tercih edilen çözüm olacaktır), ancak aşırı büyük listeler için daha verimli teknikler kullanabiliriz. Brute-Force tanımlayıcı eşleştiricisi, özellik eşleştirmesi için Brute-Force yaklaşımı kullanır. İlk görüntüdeki bir özelliğin tanımlayıcısını alır ve bazı uzaklık hesaplamaları kullanarak ikinci görüntüdeki tüm özelliklerin tanımlayıcıları ile karşılaştırır. Sonra en yakın olan sonuçta elde edilen çiftle geri dönmektedir. Çok sayıda kıyaslama nedeniyle, Brute-Force algoritması bazen daha fazla zaman alır ancak kesindir. Özel parametreleri ayarlayarak performansı ve aykırı değerler kaldırma yeteneği geliştirilebilir (Jakubović ve Velagić, 2018).

4.5. HAAR Cascade

Alfred HAAR adında bir bilim adamı HAAR-cascade sistemini geliştirmiştir. Bu yöntem görüntü üzerindeki nesneyi bulmak için kullanılmaktadır. İlk olarak yüz tanıma amacıyla kullanılmıştır. Nesneyi uygun hassasiyette tanımlaması için çok sayıda özellik tanımlaması yapılması gerekmektedir. Algılama penceresindeki her bölgenin piksel yoğunluğunun toplamını hesaplar ve bu toplamlar arasındaki farkı hesaplar bu sayede alt kategorileri sınıflandırır. Çalışması esnasında Şekil 4.4'te gösterilen beyaz dikdörtgenlerin içinde kalan piksellerin toplamı gri dikdörtgenlerin piksellerinin toplamından çıkarılır.



Şekil 4.4. (a), (b) İki dikdörtgen özellikli, (c) üç dikdörtgen özellikli, (d) 4 dikdörtgen özellikli

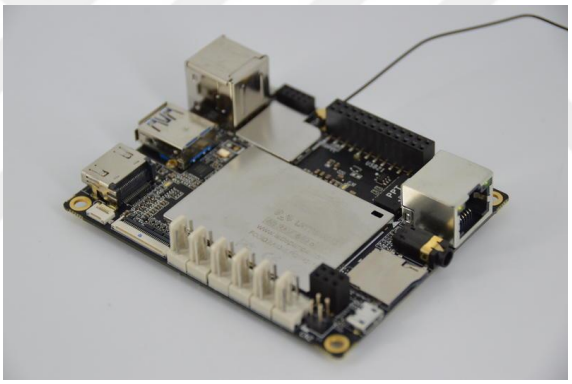
5. MATERYAL VE YÖNTEM

Bu bölümde, tez çalışmasında kullanılan Nesne Tanıma yöntemleri ile ORB (Yönlendirilmiş FAST ve Döndürülmüş BRIEF), kullanılan elektronik ve mekanik parçalar tanıtılmıştır.

5.1. Materyal

5.1.1. Modül üzeri LattePanda bilgisayar

LattePanda bir modül üzeri bilgisayar geliştirme kartıdır. Intel Atom dört çekirdekli işlemcisini ve 4 GB düşük güçlü DDR3 RAM içermektedir. Ayrıca 64 GB gömülü bellek ve Arduino Leonardo mikrodenetleyici ile gelmektedir.



Şekil 5.1. Modül üzeri LattePanda

LattePanda yeni bir geliştirme kartıdır. Windows 10 ile tam uyumla çalışmaktadır. Düzenli bir PC'nin sahip olduğu her şeyi içermekte ve yapabildiği her şeyi yapabilmektedir. Bilinen hemen her küçük aygıta uyumludur. Yazıcılar, kumanda kolları, kameralar örnek olarak verilebilir. PC üzerinde çalışan herhangi bir çevrebirimi LattePanda üzerinde de çalışmaktadır. LattePanda Windows 10'un tam sürümü kurulu bir şekilde gelmektedir. Benzer şekilde üzerinde Visual Studio, NodeJS, Java, Processing gibi uygulamalar da içermektedir. Mevcut uygulama arayüzlerini kullanarak normal bir PC'de yapılabildiği gibi yazılı ve donanım projeleri LattePanda üzerinde başarıyla geliştirilebilmektedir. LattePanda, sadece düşük maliyetli düzenli bir Windows PC olarak kullanılmakla kalmayıp aynı zamanda Arduino mikrodenetleyici ile de tasarlanıp kullanılabilmektedir.

Çizelge 5.1. LattePanda özellikleri

İşlemci	Intel® Cherry Trail Z8350 Quad Core
İşletim Sistemi	Windows 10
RAM	4 GB
Bellek	64 GB eMMC
GPU	Intel HD Graphics, 12 Eus, 200-500 MHz, tek kanallı bellek.
Portlar	1 adet USB 3.0 ve 2 adet USB 2.0
İletişim	Wi-Fi 802.11n 2.4G, Bluetooth 4.0
Diğer Özellikler	Yerleşik Arduino Leonardo

5.1.2. Kamera

Şekil 5.2'de görülen kamera modeli web kameralarının 1080p Tam-HD çeşididir. Bu web kamerasının yüksek duyarlılıkta bir sensörü vardır. Web kamerası doğal sesi aktarabilen bir gömülü yüksek kalite mikrofonu da içermektedir. Ayrıca 360 derece döndürülebilmektedir.

Şekil 5.2. A4TECH web kamera.

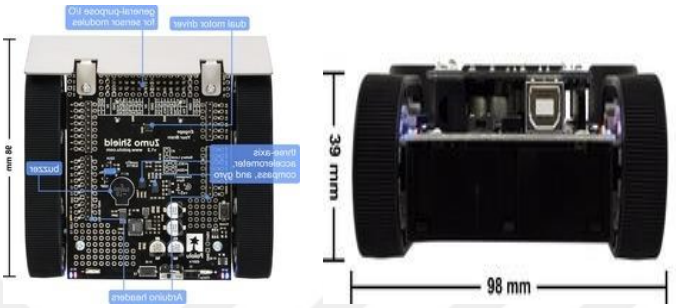
Kompakt A4TECH web kamerası ultra-keskin video kaydı alabilmek için yüksek duyarlılıkta Tam HD sensörü kullanmaktadır. Düşük ortam ışığında bile 30 fps'de ve 16 Mp'e kadar görüntü kaydedebilmektedir. Kameranın diğer teknik özellikleri Çizelge 5.2'de görülebilmektedir.

Çizelge 5.2. A4TECH webcam özellikleri

Sensör	1080p Tam HD
Görüntü Kalitesi	En fazla 16Mpx (4608×3456)
Ortam Işığı	30fps
Bağlanabilirlik	USB 2.0
Mikrofon	Gömülü

5.1.3. ZUMO robot shield

Bir Arduino kontrollü Zumo robot, batarya ve hareket ettiriciler, kart üzerindeki Arduino girişlerine bağlanmaktadır. Zumo Robot Shield ikili hareket ettiricileri, buzzer, bir 3-eksenli ivme-ölçer ve pusula içermektedir. Hareketini paletler üzerinden gerçekleştiren 10 cm genişlikli Zumo robot Şekil 5.3'de görülebilmektedir.



Şekil 5.3. Zumo robot shield

Zumo robot paletlerini hareket ettirebilmek için metal redüktör dişlileri içeren iki motoru sürmek için DRV8835 motor sürücüsü ve 3 boyutlu ivme-ölçer ve manyetik alan ölçer olarak LSM303DLHC bulundurmaktadır.

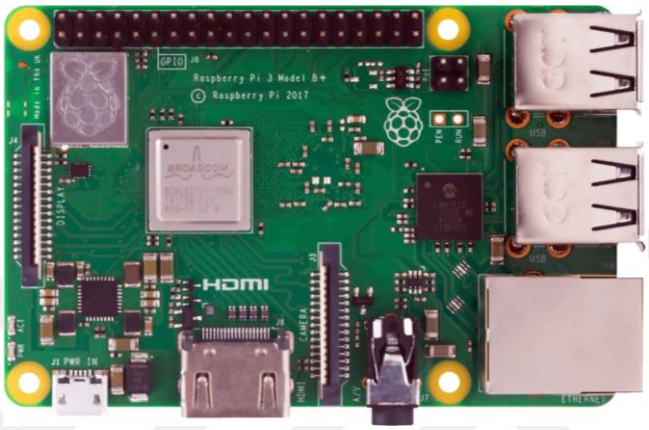
5.1.4. LattePanda kumanda

Çalışmada kullanılan LattePanda kumandasının üzerindeki oklar robotu uzaktan kontrol etmek için kullanılmıştır. Şekil 5.4'te görülebilen bu kumanda ile robota sağ-sol ve ileri-geri komutları verilebilmektedir.

Şekil 5.4. LattePanda kumanda.

5.1.5. Raspberry Pi 3 Model B+

Şekil 5.5'de görülen Raspberry Pi 3 B+ modeli yenilenen özellikleriyle işlem gücü ve bağlantı hızları arttırılmış bir ürün olarak Raspberry Pi 3 ailesinin en son ürünüdür. Önceki sürümlerine göre daha hızlı bir işlemciye sahip olan bu ürünün işlemcisinin üzerine yerleştirilen metal kapak ile daha iyi bir soğutma sağlanmıştır.



Şekil 5.5. Raspberry Pi 3 B+ Modeli

Raspberry Pi 3 B+ için detaylı teknik özellikler aşağıdaki gibi sıralanabilir:

- Dört Çekirdek 64-bit 1.4GHz A53/ARMv8 İşlemci,
- 1 GB LPDDR2 SD-RAM,
- Çift bant 2.4 GHz + 5 GHz 802.11.b/g/n/ac Wireless LAN,
- 300Mbps Gigabit, PoE HAT uyumlu Ethernet,
- Düşük Enerjili Bluetooth 4.2,
- 40 pinli GPIO giriş,
- Micro SD kart desteği,
- HDMI, DSI ekran portu ve CSI kamera portu,
- 4 kutuplu 3.5mm ses ve kompozit video portu,
- 4 adet USB 2.0 ve 1 adet MikroUSB 5V/2.5A güç girişi,
- H.264, MPEG-4 1080p (30 fps), OpenGL 2.0 Multimedya.

5.1.6. Raspberry Pi kamera V2

Raspberry Pi için özel olarak tasarlanan ve Şekil 5.6'da fotoğrafı görülebilen Raspberry Pi Kamera V2, sabit odaklı bir objektife sahiptir. Kamera, yüksek kalitede 8 megapiksel çözünürlüklü Sony IMX219 görüntü sensörü içermektedir.

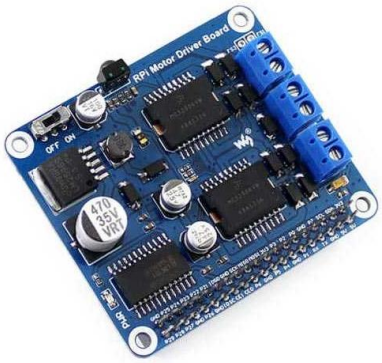
Şekil 5.6. Raspberry Pi kamera V2

Raspberry Pi Kamera V2'nin detaylı teknik özellikleri aşağıdaki gibi sıralanabilir:

- Sabit Odaklı Lens,
- 8 Mp doğal çözünürlüklü sensör (3280 x 2464 piksel fotoğraf),
- 1080p (30fps), 720p (60 fps) ve 640x480p (90 fps) video,
- 5 mm x 23 mm x 9 mm, 3 gram,
- Raspberry Pi'nin en son sürüm işletim sistemi (Raspbian) ile uyumlu,
- Kısa GPIO çoklu kablo (ribbon) ile Raspberry Pi'ye doğrudan bağlanabilmektedir.

5.1.7. Raspberry Pi motor sürücü kartı

Şekil 5.7'de görülen Raspberry Pi Motor Sürücü Kartı, Raspberry Pi 3 B+ ile uyumlu küçük boyutlu bir DC motor sürücü kartıdır. Kart üzerinden 2 adet DC motoru birbirinden bağımsız olarak sürebilmektedir. 1.5V-11V gibi geniş bir gerilim aralığında çalışabilmektedir. Böylece birçok projede rahatlıkla kullanılabilmektedir.



Şekil 5.7. Raspberry Pi motor sürücü kartı

Raspberry Pi motor sürücü kartının detaylı teknik özellikleri aşağıdaki gibi sıralanabilir:

- 1.5-11 V arası motor gerilimi,
- 2-7 V arası lojik 1 sinyali,
- Kanal başına maksimum 1.2 A çıkış akımı,
- 250 kHz maksimum PWM frekansı,
- Ters voltaj koruması.

5.1.8. Power bank (Taşınabilir şarj aleti)

Şekil 5.8'de görülen power bank, Raspberry Pi için kaynak gücü olarak kullanılmakta ve 10.000 mAh enerji kapasitesine sahiptir.

Şekil 5.8. Power bank (Taşınabilir şarj aleti)

5.1.9. Normal kaynak gücü (AA kalem pil)

Şekil 5.9'da görülen 8 adet AA kalem pil 4 adet DC motora enerji sağlamak için kullanılmıştır.

Şekil 5.9. Normal kaynak gücü

5.2. Yöntem

Bu çalışmada ORB algoritmasının uygulaması ve Brute-Force eşleme yöntemi kullanılmıştır. Sistem gerçek zamanlı kamera video çerçevelerinden interset ana noktaları algılamakta ve Brute-Force kullanarak referans nesne görüntüsü tanımlayıcısı ile eşlemektedir

5.2.1. Birinci çalışmanın sistem tasarımı

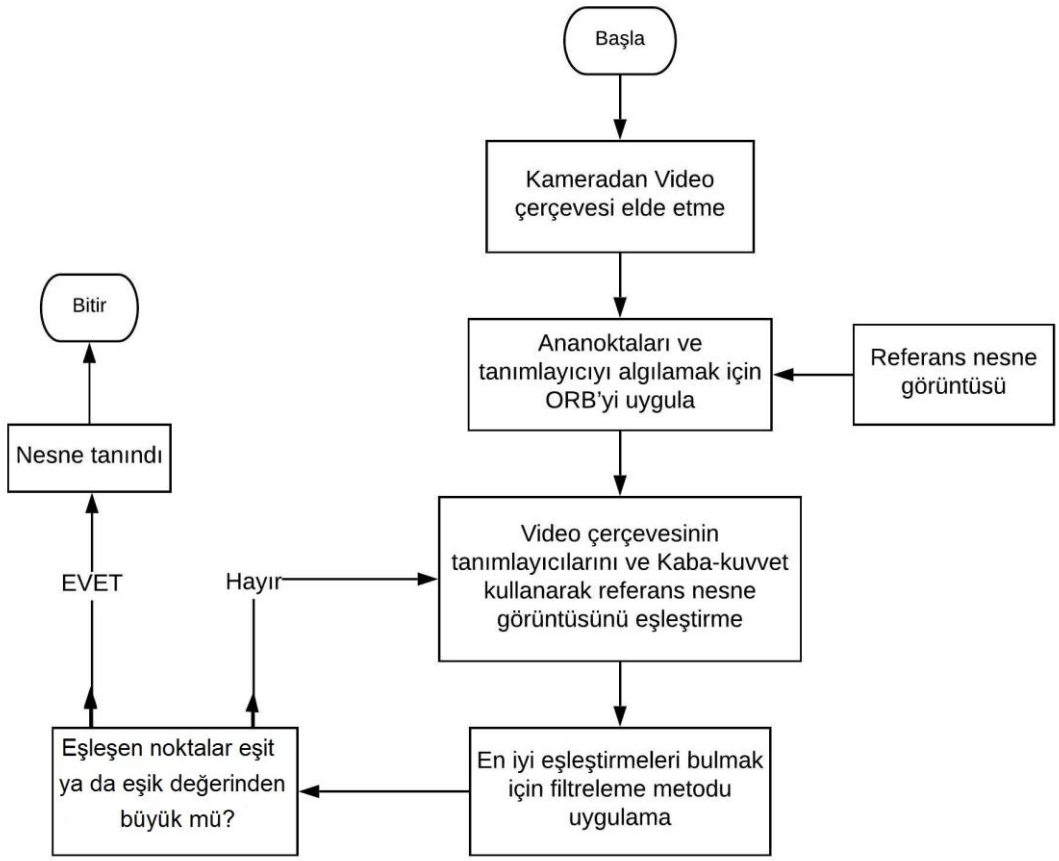
Bu kapsamda gerçekleştirilen sistem, Şekil 5.10'da görülebilen uzaktan kumanda (RC) ile çalışabilen bir robottur. Kontrol için kullanılan LattePanda uzaktan kumandası Bluetooth aracılığıyla robot ile bağlantı kurarak çalışmaktadır.

Şekil 5.10. Geliştirilen RC robot ve kumanda aygıtı

Geliştirilen sistemde mikro işlemci olarak LattePanda kullanılmıştır. Robot kiti olarak ise Zumo robot shield kullanılmıştır. Uzaktan kumanda ile sağ-sol ve ileri-geri komutları ile robot, kullanıcı tarafından kontrol edilebilmektedir. Ayrıca kumanda üzerinden robotun hız kontrolü de yapılabilmektedir. A4Tech HD web kamera, hareket

eden robotun üzerine yerleştirilmiştir. Kameradan elde edilen video görüntüsü USB bağlantısı yolu ile LattePanda'ya aktarılmaktadır.

Yapılan çalışmada özellik çıkarma yöntemi olarak ORB (Yönlendirilmiş FAST ve Döndürülmüş BRIEF) kullanan bir tanıma sistemi sunulmakta, böylece daha önceden kaydedilmiş nesneleri algılama ve tanıma görevlerini yerine getiren bir robot sistemi oluşturulmuş olmaktadır. ORB gerçek zamanlı kameradan elde edilen video çerçevesinde mevcut özellikleri algılamak için kullanılmakta ve sonra Brute-Force eşleştirici kullanılmaktadır. Böylece kameranın görüntü kapsamına giren nesne görüntüsünü referans görüntü ile eşleştirmektedir. Kamera ile elde edilen canlı görüntü üzerinden sürekli olarak nesnelerin işlenmesi gerçekleştirilerek nesneleri algılamaya çalışmaktadır. Eşleştirme aşamasında eşik değere eşit ya da daha büyük ölçüm elde edilebilirse nesne tanınacaktır. Uygulamanın çalıştırılması için Python ile de programlama yapılmıştır. Sistemin çalışma diagramı Şekil 5.11'de görülmektedir.

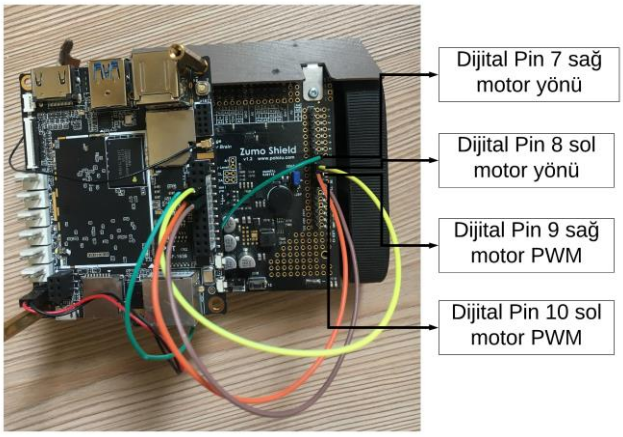


Şekil 5.11. Geliştirilen birinci sistemin akış diyagramı

LattePanda entegreli Arduino Leonardo mikrodenetleyici pinout'larına bağlı Zumo robot Shield Çizelge 5.3 ve Şekil 5.12'de görüldüğü gibi kullanıldı.

Çizelge 5.3. Arduino pinout ve Zumo shield işlevleri

LattePanda pin çıkışları	Zumo Shield İşlevi
Pin 7	Sağ motor yönü kontrol hattı
Pin 8	Sol motor yönü kontrol hattı
Pin 9	Sağ motor PWM kontrol hattı
Pin 10	Sol motor PWM kontrol hattı



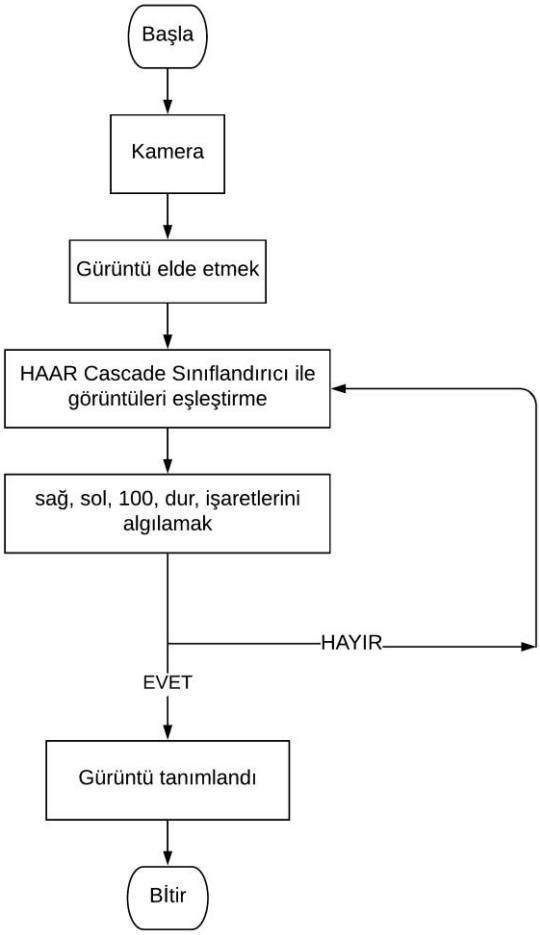
Şekil 5.12. LattePanda pin çıkışları ve Zumo shield'in bağlantısı

5.2.2. İkinci çalışmanın sistem tasarımı

Geliştirilen ikinci sistemde Raspberry Pi kullanılmıştır. İşletim sistemi üzerinde Python ile programlama yapılmıştır. Raspberry Pi'nin üzerine motor sürücüsü yerleştirilmiştir. Raspberry Pi'ye uyumlu bir kamera eklenmiştir. Bu şekilde oluşturulan Robot, HAAR Cascade Sınıflandırıcı ve OpenCV kütüphanesi kullanmaktadır. Aynı zamanda VNC Viewer programı araçları ve yazılımlar ile robot kontrol edilmektedir. HAAR Cascade temel olarak, kaynaktan eğitildiği nesneyi tespit etmek için kullanılan bir sınıflandırıcıdır. HAAR Cascade, pozitif görüntüyü bir dizi negatif görüntü üzerinde üst üste ilave ederek eğitilir. Eğitim genellikle bir sunucuda ve çeşitli aşamalarda yapılır. Yüksek kaliteli görüntüler kullanılarak ve sınıflandırıcının eğitildiği aşamaların miktarı arttırılarak daha iyi sonuçlar elde edilir.

Yapılan ikinci çalışmada VNC viewer programı ile robota "başla" komutu verilerek Raspberry Pi ve kamera çalışılmaktadır. Robot çalıştıktan sonra Şekil 5.13'da akış diyagramında belirtildiği gibi görüntüleri elde ederek HAAR Cascade Sınıflandırıcı ile görüntüleri eşleştirilecektir. Eşleştirilen görüntüler sağ, sol, 100, dur, işaretlerini algılayarak robot işaretlere göre yönlenmektedir. Robot algıladığı görüntüler arasında sağ, sol, 100, dur işaretlerini bulamadığı zaman sabit hız ve düz doğrultuda devam

edecektir. Robot sağ yöne dönerken sağ tekerlekler tamamen durmaktadır ve sağ yöne devam etmektedir. Sola dönüş de benzer şekilde gerçekleşmektedir.

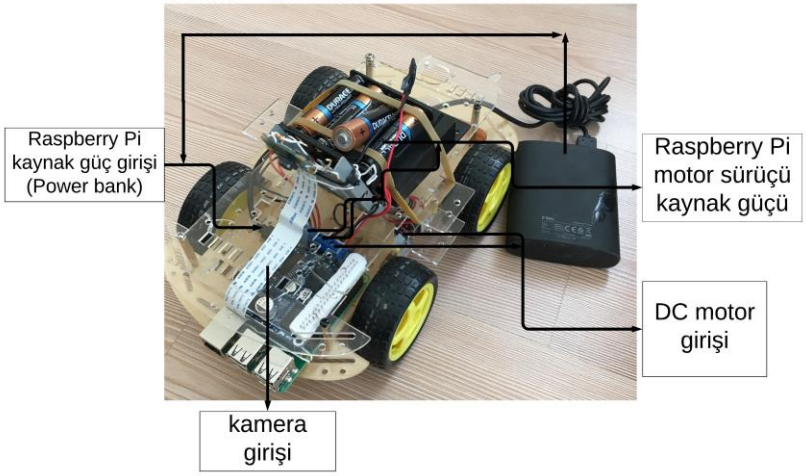


Şekil 5.13. İkinci sistemin akış diyagramı.

Gerçekleştirilen sistemin bağlantıları Çizelge 5.4'de ifade edildiği gibi yapılmıştır. Gerçekleştirilen sistemin görüntüsü Şekil 5.14'de görülebilmektedir.

Çizelge 5.4. 4WD Çift Kat Mobil Araç Kiti Elektronik bağlantısı

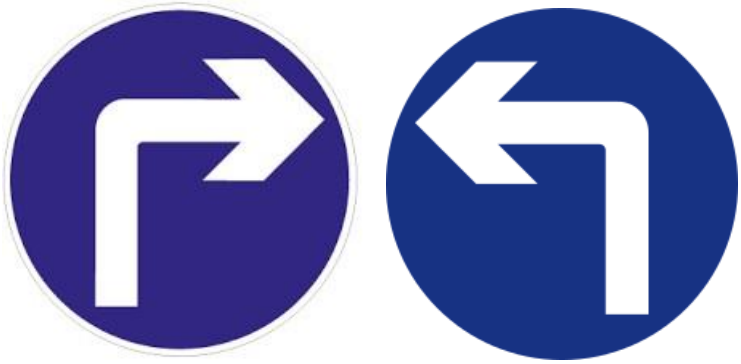
Robot Parçaları	Raspberry Pi girişleri
Kamera	Raspberry Pi soket
Güç kaynağı (power bank)	Raspberry Pi kaynak güç girişi
DC motor	Raspberry Pi motor sürücü kartı
Güç kaynağı (batarya)	Raspberry Pi motor sürücü kartı



Şekil 5.14. Geliştirilen ikinci sistemin elektronik bağlantısı

Yapılan ikinci çalışmada, tasarlanan ve uygulaması yapılan birinci robot sisteminin daha da geliştirilebilmesi amaçlanmıştır. Birinci uygulamada sistemin görevi belirli bir nesneyi tanıma ve bu nesneyi tanıdıktan sonra nesneye doğru hareket etme ve bu nesnenin önüne belirli bir mesafe kala vardığında otomatik olarak durmaktır. İkinci uygulamada sisteme yüklenen ilave görevler aşağıdaki şekilde sıralanabilir:

- Önündeki alanda Şekil 5.15'de görülen 'SAĞA DÖN' ve 'SOLA DÖN" trafik işaretlerini gördüğünde yeterli şekilde dönme imkanına sahiptir.
- Şekil 5.16'da görülen '100 HIZ' işaretini gördüğünde -kontrollü- hızlanma performansı eklenmiştir.
- Şekil 5.17'de görülen 'DUR' işaretini gördüğünde durmak.



Şekil 5.15. 'SAĞA DÖN' ve 'SOLA DÖN' levhaları

Şekil 5.16. ‘100 HIZ’ levhası

Şekil 5.17. ‘DUR’ levhası

Birinci çalışmadan farklı olarak ilerletme çalışmasında kullanılan donanım parçaları aşağıda belirtilmiştir:

- Raspberry Pi 3 Model B+
- Raspberry Pi Kamera V2
- Raspberry Pi Motor Sürücü Kartı
- 4 adet DC motor
- 8 adet pil
- Power bank

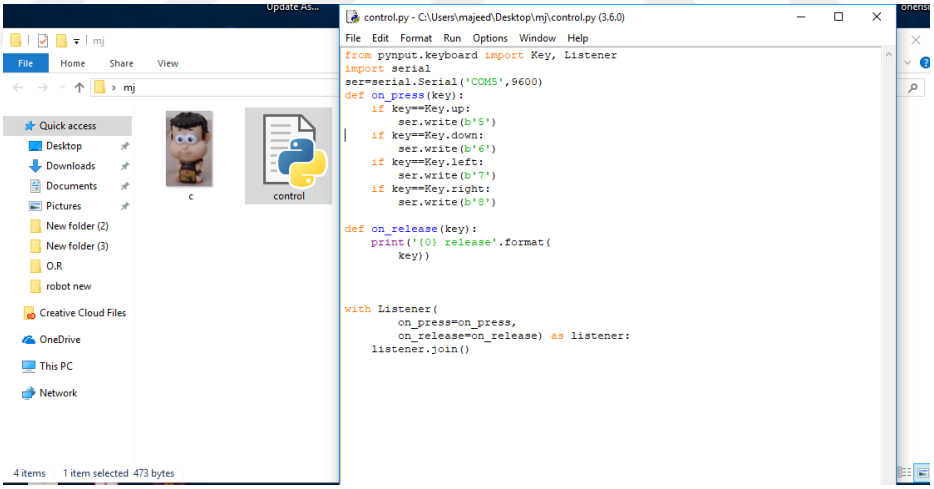
Yukarıdaki donanım parçalarının kullanımının tercih edilmesinin nedeni robot sistemin görevlerini daha yüksek performansla yerine getirmesini sağlamaktır. Nitekim birinci çalışmaya oranla ikinci çalışmadaki robotun bu donanımla belli görevlerde daha iyi performans sergilediği gözlenmiştir.

6. ARAŞTIRMA SONUÇLARI

Yapılan çalışmada önce birinci robot sistemin tasarımı ve uygulaması yapılmıştır. Daha sonra yapılan ilk sisteme yeni görevlerin eklendiği ikinci bir çalışma daha gerçekleştirilmiştir.

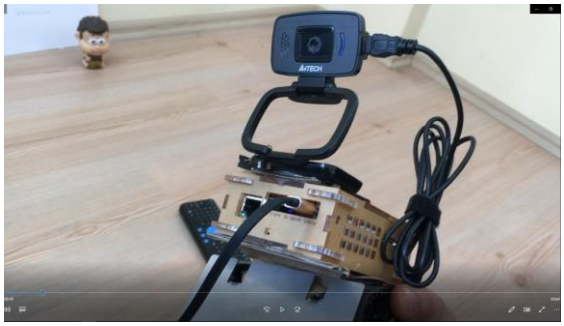
6.1. Geliştirilen Birinci Sistem

Sistemin tasarımında kullanılan Zumo robot shield ile modül-üzeri-LattePanda'ya bağlantı kurulmaktadır. LattePanda'ya entegre bir Arduino Leonardo mikrodenetleyici bulunmaktadır. Ayrıca sistemde kullanılan LattePanda bilgisayar USB portlara sahip olup, bu sayede normal PC ile de bağlantısı sağlanabilmektedir. Arduino mikrodenetleyicisi için Arduino yazılımı ve nesne tanıma yazılımı uygulamasında Python programlama dili kullanılmıştır. Şekil 6.1'de görüldüğü gibi Python programlama dili, openCv kütüphanesini ve robot hareket ettiricilerini kontrol etmek için ilgili bir arayüzü kullanmaktadır.



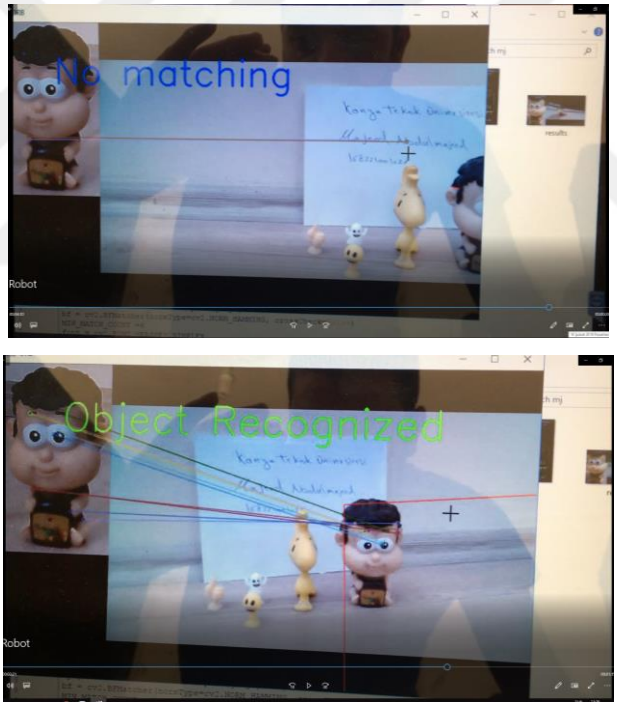
Şekil 6.1. Python kontrol dosyası ve ilgili yazılımlar.

Şekil 6.2'deki gibi kamera ile donatılmış robot, kamera açısına gelen nesneyi algılamakta ve referans olarak kaydedilen görüntüyle eşleştirme yapmaktadır.



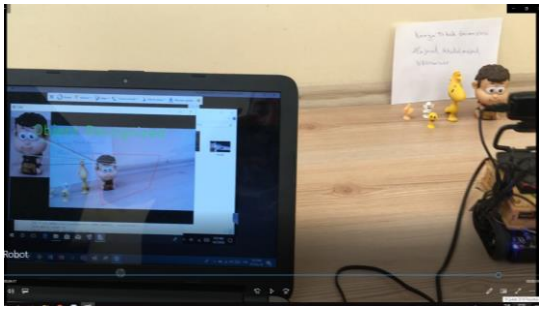
Şekil 6.2. Robot sistemi üzerinde monte edilmiş kameranın görünümü

Sistemin çalışması esnasında önce kameradan video görüntüsü elde edilmektedir. Ardından Şekil 6.3'de görüleceği üzere kameradan elde edilen her bir video çerçevesi üzerindeki görüntülerden algoritma ana noktaları çekip çıkarmakta ve tanımlayıcısını bir referans nesne görüntüsü tanımlayıcısı ile eşleştirmeye çalışmaktadır.



Şekil 6.3. Nesne eşleştirme ve tanıma süreçleri

Video çerçevesindeki nesneler ve referans nesneler arasında en iyi eşleştirme özelliklerini bulmak için Brute-Force eşleyici kullanılarak tanımlayıcılar eşlenmektedir. Eşleştirmeler bir eşik (uzaklık) değeri kullanarak filtrelenmektedir. Referans nesne görüntüsü ve kamera görüntüsündeki nesneler arasında daha iyi eşleştirmeleri elde etmek için 0.65 uzaklık oranı kullanılmaktadır. Teori ve pratikte en iyi eşleştirmeler eşik seviyesinden daha az bir uzaklığa sahip olacaktır.



Şekil 6.4. Nesnenin tanınması sonucunda robotun nesneye doğru hareketlenmesi

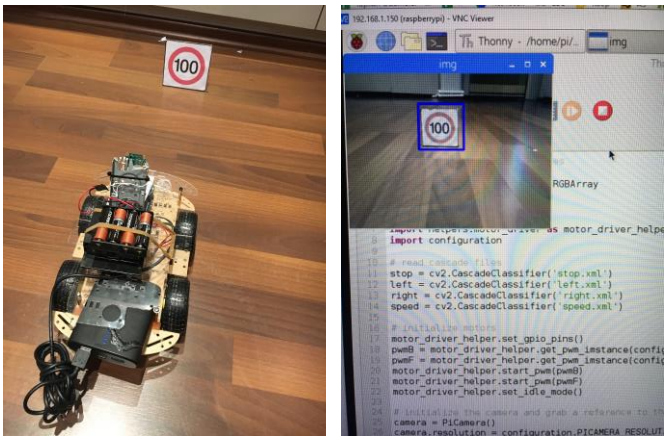
Şekil 6.4'de görüldüğü üzere sistemin çalışması esnasında yazılım yardımıyla tanınmakta olan nesneye doğru robot hareketlenmekte ve önüne gelip durmaktadır.

Bu sistem üzerinde birden fazla nesnenin hafızada kaydedilmesi ile bu nesnelerin ayrı ayrı tanınmasının gerçekleştirilmesi üzerine yapılan çalışmada başarılı sonuçlar elde edilememiştir. Çünkü ORB tek nesnenin tanınabilmesi için elverişli bir algoritmadır. Bu yüzden geliştirilecek ikinci sistemde ORB kullanılmamıştır, onun yerine HAAR Cascade sınıflandırıcısı kullanılmıştır. Ayrıca HAAR Cascade sınıflandırıcısının LattePanda işlemcisi üzerinde kullanımında takılmalar yaptığı gözlenmiştir. Bundan dolayı ikinci sistemde LattePanda yerine Rasperry Pi 3 Model B+ tercih edilmiş ve başarılı sonuç elde edilmiştir.

6.2. Geliştirilen İkinci Sistem

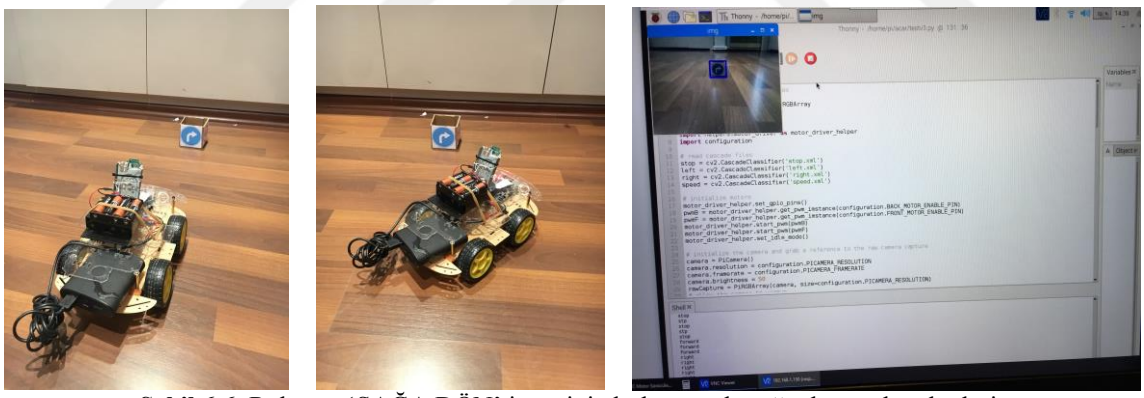
Birinci robot sisteminde daha önce kaydedilmiş bir nesnenin tanınması gerçekleştirilmesine rağmen birden fazla nesnenin kaydedilip ayrı ayrı tanınmasının gerçekleştirilememesi nedeniyle ikinci sistemin ayrı donanımlar ile gerçekleştirilmesi gerekmiştir. İkinci sistem birden fazla nesnenin kaydedilebilmesine ve alınan video görüntülerinde tespit edilen nesnelerin referans nesneleriyle eşleştirmelerini gerçekleştirip buna bağlı olarak tanımlanmış komutların robot tarafından gerçekleştirilmesi sağlanmıştır.

Şekil 6.5'de görüldüğü üzere "100 Hız" işaretini gördüğünde robot, -kontrollü- hızlanmaktadır. Robot güzergahta ilerlerken yaklaşık 20 cm kala bu işareti algılayarak hızlanma yapmaktadır.

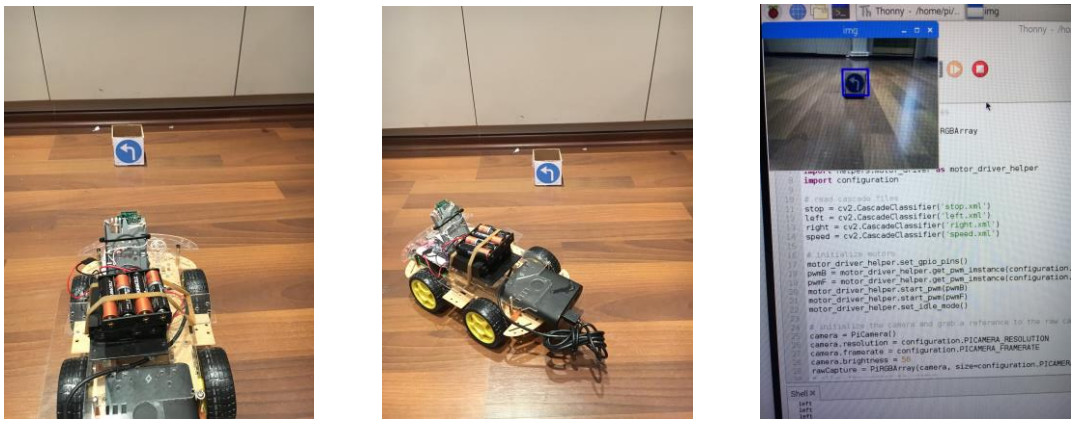


Şekil 6.5. Robotun "100 Hız" işaretini algılamasıyla belirli bir hızlanma göstermesi

Robotun sağa ve sola dönüş levhalarını algılayıp dönüş işlemini gerçekleştirme aşamaları Şekil 6.6 ve 6.7'de görülebilmektedir. Robotun sağa ve sola dönüş performansına bakıldığında, sağa ve sola dönüşlerde belli bir miktar yavaşlayarak dönüşlerini yapmaktadır. Dönüşlerden sonra tekrar hızlanmaktadır. Robot yolda giderken sağa ve sola dönüş işaretlerini yaklaşık 30 cm kala algılayarak dönüşlerini yapmaktadır.



Şekil 6.6. Robotun 'SAĞA DÖN' işaretini algılamasıyla sağa dönme hareketleri.



Şekil 6.7. Robotun ‘SOLA DÖN’ işaretini algılamasıyla sola dönme hareketleri.

Robot ‘DUR’ işaretini algıladığında, ‘DUR’ işaretinin önüne –belli bir mesafe kala- geldiğinde Şekil 6.8’de görüldüğü gibi durmaktadır. Bu mesafe yaklaşık 35 cm olarak ölçülmüştür.

Şekil 6.8. Robotun ‘DUR’ işaretini algılamasıyla durma anı

6.3. Birinci ve İkinci Uygulamanın Performansının Karşılaştırılması

Geliştirilen birinci sistemde karşılaşılan kısıtlar nedeniyle ikinci sistemin gerçekleştirilmesi gerekmiştir. Bu bağlamda ikinci sistemin geliştirilmesi ile amaçlanan hedefe ulaşılmıştır. Çizelge 6.1’de gerçekleştirilen iki ayrı uygulamanın kabiliyetleri karşılaştırılmıştır.

Çizelge 6.1. Birinci ve ikinci uygulamanın performanslarının karşılaştırma tablosu

Kabiliyetler	Birinci Sistem	İkinci Sistem
Nesne tanıma	Tanıma	Tanıma ve hareket
Toplam sistem görevlerini uygulama	Daha basit (Sistemin basit görevlerini yerini getirmede zorlanma ve aksaklıklar gözlendi).	İleri seviye (Sistemin ileri görevlerini kolay ve aksaksız yerine getirdiği gözlendi).
Hızlanma	Ortalama	İleri seviye (Daha kontrol edilebilir hızlanma ve yavaşlama)
Dönüşler	Sağ ve sol dönüşler normal	Sağ ve sol dönüşler daha ileri
Durma	Bireysel durma	Algılayarak durma
Özerk hareket	Kısıtlı	İleri seviye

7. SONUÇLAR VE ÖNERİLER

7.1. Sonuçlar

Nesne tanıma, bilgisayar bilimi alanıyla ilişkili önemli bir uygulamadır ve birçok farklı alanda da uygulamaları görülmektedir. Günümüzde nesne tanıma amacıyla kullanılan muhtelif algoritmaların birbirine göre üstün ya da zayıf olduğu yanları vardır. Bu tez çalışmasında kullanılan ORB ve HAAR Cascade bu algoritmalardadır. ORB hızlı ve sağlam bir yerel özellik bulucusudur. Ayrıca eşleştirmeleri de hızlı yaptığı görülmektedir. Bu algoritmayı geliştirenler tarafından sonradan onun üzerinde yapılan bazı değişikliklerle ORB'nin özellikleri daha da iyileştirilmiştir. ORB, özetle FAST köşe noktaları bulucu algoritması ve BRIEF tanımlayıcısı bileşenlerinden oluşmaktadır. Algoritmanın geliştiriciler tarafından sonradan yönlendirme (orientation) ve döndürme (rotation) ile performansını iyileştirmek için değişiklikler yapılmıştır. Görüntü özelliklerini çıkartma yöntemleri, yapılan bu çalışmada da görülmüştür ki, geliştirilen uygulamalı sistemlerin performansında önemli rol oynamaktadır. HAAR Cascade temel olarak, kaynaktan eğitildiği nesneyi tespit etmek için kullanılan bir sınıflandırıcıdır. HAAR Cascade, pozitif görüntüyü bir dizi negatif görüntü üzerinde üst üste ilave ederek eğitilir. Eğitim genellikle bir sunucuda ve çeşitli aşamalarda yapılır. Yüksek kaliteli görüntüler kullanılarak ve sınıflandırıcının eğitildiği aşamaların miktarı arttırılarak daha iyi sonuçlar elde edilir.

Tez çalışması kapsamında çevredeki nesnelerin üzerindeki mikrodenetleyicili bir mikro bilgisayar , bir kamera gibi gerekli donanım parçaları ve ilgili yazılımlar yoluyla otomatik tanıyabilme kabiliyetli bir robot geliştirilerek, bu robotun yürütülmesi sağlanmıştır. Bu amaçla iki ayrı sistem geliştirilmiştir. Birinci sistemde LattePanda, Zumo robot ve web kamera kullanılmıştır. Sistemin üzerine yerleştirilen kamera tarafından bilgisayara gerçek-zamanlı görüntüler iletilerek nesnelerin özellik noktalarının çıkartılması gerçekleştirilmiştir. Ardından referans nesne görüntüsü tanımlayıcısıyla eşleştirme işlemi sonucunda nesneleri tanımasıyla robot bu nesnelere doğru başarıyla yürütülmüş ve önlerine geldiğinde otomatik olarak durmuştur. Bu çalışma için tasarlanan robota sınırlı görevler tanımlanmıştır. Geliştirilen birinci sistem üzerinde birden fazla nesnenin hafızada kaydedilmesi ile bu nesnelerin ayrı ayrı tanınmasının gerçekleştirilmesi amacıyla yapılan ilave çalışmada başarılı sonuçlar elde edilememiştir. Çünkü ORB tek nesnenin tanınabilmesi için elverişli bir algoritmadır. Bu

yüzden geliştirilen ikinci sistemde HAAR Cascade sınıflandırıcısı tercih edilmiştir. Ayrıca HAAR Cascade sınıflandırıcısının LattePanda işlemcisi üzerinde kullanımında takılmalar yaptığı gözlenmiştir. Bundan dolayı ikinci sistemde LattePanda yerine Rasperry Pi 3 Model B+ tercih edilmiş ve başarılı sonuç elde edilmiştir. Birinci sistemin uygulamasında sınırlı olan görevler, ikinci sistemin uygulamasında çoğaltılmış ve robotun beklenen yeni görevlerini başarıyla yapabildiği tespit edilmiştir. İkinci çalışmada robot sistemi, sağa ve sola dönüş işaretlerini gördüğünde uygun (yeterli) bir şekilde dönebilme, hız işaretini gördüğünde hızlanma, ayrıca dur işaretini gördüğünde durma görevlerini başarmıştır. Bu işlemler özerk (otonom) olarak gerçekleşmektedir.

7.2. Öneriler

Yapılan çalışmada ORB'nin performansı iyi bulunarak diğer uygulayıcıların da gelecekteki çalışmalarında kullanmaları için önerildiği gibi eğer daha üstün sonuçlar elde edilmek istenirse ORB algoritması yerine makine öğrenmesi kullanarak daha iyi sonuçlar da elde edilebilir. Bu çalışmada belirli bir uygulama için nesne algılama ve tanıma yeteneğine sahip iki ayrı robot sistemi gerçekleştirilmiştir. Geliştiriciler farklı amaçlar için de benzer sistemleri gerçekleştirebilir.

KAYNAKLAR

Anonymous, 2019a, Popularity of Programming Languages, http://pypl.github.io/PYPL.html:

Anonymous, 2019b, Programming Community Index with respect to Years, https://www.tiobe.com/tiobe-index/:

Belongie, S. J., Malik, J. ve Puzicha, J., 2001, Matching Shapes, *iccv*, 454-461.

Calonder, M., Lepetit, V., Strecha, C. ve Fua, P., 2010, Brief: Binary robust independent elementary features, *European conference on computer vision*, 778-792.

Chang, P. ve Krumm, J., 1999, Object recognition with color cooccurrence histograms, *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 498-504.

Chaudhari, K. C. ve Patil, A. M., 2015, A Survey on Object Recognition and Classification, *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 3 (Issue 10, October 2015).

Jakubović, A. ve Velagić, J., 2018, Image Feature Matching and Object Detection Using Brute-Force Matchers, *2018 International Symposium ELMAR*, 83-86.

Karami, E., Prasad, S. ve Shehata, M., 2017, Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images, *arXiv preprint arXiv:1710.02726*.

Khurana, K. ve Awasthi, R., 2013, Techniques for object recognition in images and multi-object detection, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2 (4), 1383-1388.

Kim, S. ve Kweon, I. S., 2007, Object categorization robust to surface markings using entropy-guided codebook, *2007 IEEE Workshop on Applications of Computer Vision (WACV'07)*, 22-22.

Leutenegger, S., Chli, M. ve Siegwart, R., 2011, BRISK: Binary robust invariant scalable keypoints, *2011 IEEE international conference on computer vision (ICCV)*, 2548-2555.

Opelt, A., Pinz, A. ve Zisserman, A., 2006, A boundary-fragment-model for object detection, *European conference on computer vision*, 575-588.

Otoom, A. F., Gunes, H. ve Piccardi, M., 2008, Comparative performance analysis of feature sets for abandoned object classification, *2008 IEEE International Conference on Systems, Man and Cybernetics*, 1020-1025.

Ramesh, G. ve Mohan, R. M., 2007, Automated object recognition-an intelligent systems approach, *IET-UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES 2007)*, 574-580.

Rane, M. R. D., Khadse, B. K. ve Suralkar, R., 2013, A review of Object Recognition Using Visual Codebook, *International Journal of Computer Science and Mobile Computing*, 2 (2), 74-79.

Rosten, E., Porter, R. ve Drummond, T., 2010, Faster and better: A machine learning approach to corner detection, *IEEE transactions on pattern analysis and machine intelligence*, 32 (1), 105-119.

Rublee, E., Rabaud, V., Konolige, K. ve Bradski, G. R., 2011, ORB: An efficient alternative to SIFT or SURF, *ICCV*, 2.

Shobha, G. ve Mudgal, V., 2012, Moving Object Detection in the real world video, *ISOR Journal of Computer Engineering, ISSN*, 2278-0661.

Song, Z., Chen, Q., Huang, Z., Hua, Y. ve Yan, S., 2011, Contextualizing object detection and classification, *CVPR 2011*, 1585-1592.

Yang, L., Jin, R., Sukthankar, R. ve Jurie, F., 2008, Unifying discriminative visual codebook generation with classifier training for object category recognition, *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1-8.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	**Majeed ABDULMAJEED**
Uyruğu	**:**	**Irak**
Doğum Yeri ve Tarihi	**:**	**Irak / Kerkük**
Telefon	**:**	**05310133332**
Faks	**:**	
e-mail	**:**	**mj.abdalmajeed@gmail.com**

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Kerkük Erkek Lisesi	2008-2009
Üniversite	:	Al Mamon Üniversitesi	2012-2013
Yüksek Lisans	:		
Doktora	:		

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**

UZMANLIK ALANI

YABANCI DİLLER

İngilizce, Arapça

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR

- Majeed A. Mohsin Abdulmajeed, Levent Seyfi, Object Recognition System Based on Oriented FAST and Rotated BRIEF, *2nd International Symposium on Innovative Approaches in Scientific Studies (ISAS 2018-Winter)*, pp. 179-181, Samsun, Turkey, November 30- December 2, 2018 **(Yüksek Lisans tez çalışmasından yapılmıştır)**.