

T.C.
Ege Üniversitesi
Mühendislik Fakültesi
Elektrik Elektronik Mühendisliği Bölümü

Lisans Bitirme Projesi Gelişme Raporu

GÖRÜNTÜ İŞLEME TABANLI GERÇEK ZAMANLI İNSAN KİTLESİ KONTROL SİSTEMİ

REAL-TIME CROWD CONTROL SYSTEM BASED ON IMAGE PROCESSING

Caner YILDIRIM
05140000548 caneryldrim@gmail.com 0 (507) 6658085

Emir Kaan YERLİ
05150000734 ekyerli@outlook.com 0 (534) 2880330

Proje Danışmanı: Dr. Öğr. Üyesi Erkan Zeki ENGİN

Bornova, İZMİR, Şubat 2020

İÇİNDEKİLER

1) KONU / AMAÇ.....	1
2) LİTERATÜR ÖZETİ	2
3) PROBLEM TANIMI	1
4) YÖNTEM / TEORİ	1
5) BİRİNCİ DÖNEM GERÇEKLEŞTİRİLENLER	1
6) ARA SONUÇLAR VE TARTIŞMA	1
7) BÜTÇE.....	1
8) REFERANSLAR	1
9) GELİŞME RAPORU ÖZ DEĞERLENDİRME FORMU.....	1

1) KONU / AMAÇ

Günümüzde her türlü veri ve bu verilerin doğru analiz edilip, kullanılması çok daha önem arz eden bir seviyeye ulaşmış durumdadır. Belirli bir alandaki giriş ve çıkışları sayan sistemler ise bu bağlamda önemi gittikçe artan bir veri sağlayıcı olarak görülebilir. Bu veriyi elde etmekte kullanılan kişi sayma kameraları farklı görüntü işleme teknikleri kullanılarak veya kızılötesi sensörler ile gerçekleştirilebilir. Projemizde ise düşük maliyet-düşük hata oranı gözetilerek en uygun yöntem seçilecek ve belirli bir alana giren-çıkan kişi sayısının anlık olarak takibini yapabilen, bu verileri hızlı, basit ve güvenilir şekilde sunan bir sistem tasarlanacaktır. Ayrıca elde edilen verilerin anlamlı hale getirilebilmesi için zamana ve sistemin kurulduğu bölgeye göre yoğunluk durumunu raporlayan web tabanlı bir arayüz tasarlanacaktır.

Belirli bir alana giren veya geçen insanların takibi ile deprem, yangın vb. acil durum senaryolarında güvenlik ve kurtarma operasyonlarının planlanmasına yardımcı olan bir sistem tasarlanması amaçlanmaktadır. Öyle ki her erişim noktasının üstüne kurulan kişi sayıcıları ile giren veya çıkan tüm insanların iki yönlü bir sayımı gerçekleştirilebilir. Tüm girişleri ve çıkışları gerçek zamanlı olarak karşılaştırarak, bir odanın, belirli bir katın veya tüm bir binanın doluluk derecesi belirlenebilir, o bölge için yoğunluk dağılım haritası oluşturulabilir. Bir binanın tahliyesinde veya bölgesel bir afet durumunda ise o sırada binadaki kişilerin tam sayısı hakkındaki bu değerli bilgiler acil durum ekiplerine sunulabilir. En kapsamlı haliyle ise akıllı şehirler için anlık yoğunluk bilgisi sağlayacak bu sistem ile ulaşım, aydınlatma, havalandırma, güç tüketimi gibi birçok farklı alanda optimizasyon sağlanabilir.

2) LİTERATÜR ÖZETİ

Açık ve/veya kapalı alanlarda insanları bulmak ve izlemek geniş bir araştırmanın hedefi olmuştur. Birçok teknik bu pazarın gerekliliğini az çok başarılı bir şekilde ele almaktadır. İzlemenin gerçekleştiği koşullar, başarısı için kritik öneme sahiptir. Gerçekleştirilen projeyle ilgili bazı çalışmalar aşağıdadır.

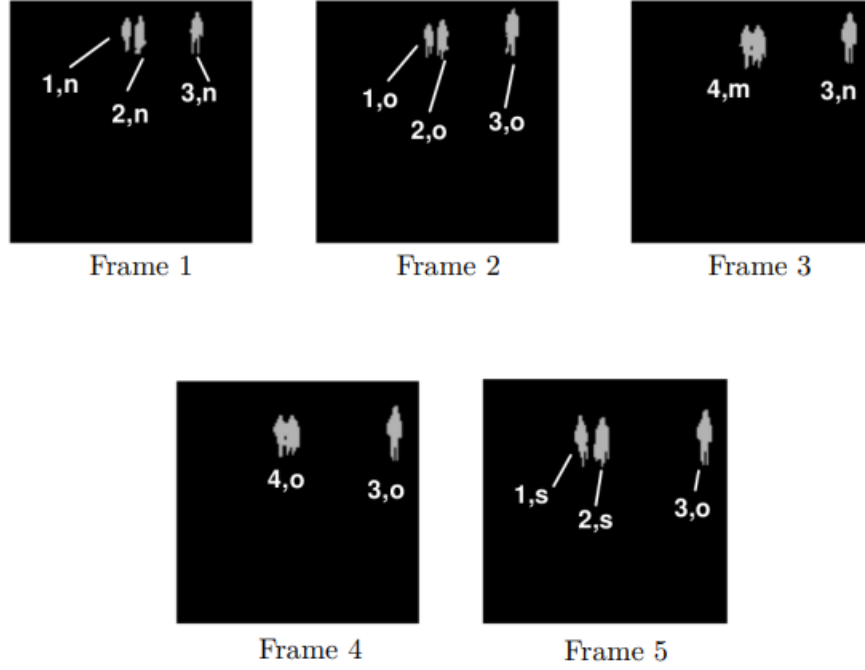
Tepegöz kamera görüntüsü kullanarak, insanların hareket yönlerini belirleyen ve sanal bir çizgiyi geçerken insanları sayan bir sistem tasarlanmıştır [1]. Tepeye yerleştirilen kamera ile, insan grupları kameranın görüş alanından geçtiğinde oluşabilecek tıkanma sorunu önlenir ve insanların yönünü belirlemek için bir uzay-zaman görüntüsü kullanılır.

Hashimoto sisteminde özel bir görüntüleme sistemi kullanarak (IR duyarlı seramikler, mekanik doğrama parçaları ve IR şeffaf lensler) insanları %95 oranında doğrulukla, 2 metrelik bir kapıdan geçerken sayabilen bir sistem geliştirdiler [2]. Sistem yan yana geçen iki ayrı kişiyi ayırt edebilmek için en az 10 cm aralığa ihtiyaç duymaktadır.

Tesei insanları izlemek ve oklüzyonları yönetmek için görüntü segmentasyonu, ilgilenilen bölgeleri çıkarmak için arka plan ayırt etme yöntemleri kullanır [3]. Damlaları izlemek için damla alanını, yüksekliği ve genişliği, sınırlayıcı kutu alanını, çevreyi ve ortalama gri seviyesini kullanırlar. Tüm bu özellikleri zaman içinde ezberleyerek, algoritma kutuları tıkanıklıktan kaynaklanan kabarcıkların birleştirilmesi ve ayrılması sorununu çözer. Kabarcıklar oklüzyon sırasında birleştiğinde, diğer özelliklerle yeni bir damla oluşturulur. Ancak, bu yeni damla, onu oluşturan kabarcıkların özelliklerinden gelen verileri saklar. Damlalar geri ayrıldığında, algoritma orijinal etiketleri orijinal damlalara atayabilir.

Shio'nun çalışması, insanın algısal gruplamasını simüle ederek arka plan segmentasyon algoritmasındaki insanların oklüzyonlarının saptanması üzerine odaklanmıştır [4]. İlk olarak, algoritma, çerçeve farkını kullanarak hareketin bir tahminini hesaplar ve bu verileri, arka plan çıkarma algoritmasının tıkanmış kişiler arasındaki sınırı belirlemesine yardımcı olmak için kullanır. Bu yöntem, genişlik, yükseklik, yön ve bir birleştirme/bölme adımı hakkında bilgiye sahip olan olasılıklı bir nesne modeli kullanır. Bir nesne modeli kullanmanın, segmentasyon için iyi bir gelişme ve oklüzyon problemini

özmenin olası bir yolu olduđu bulunmuştur. Ancak algısal gruplamanın kullanılması, örneğın, aynı yönde hareket eden bir grup insanın neredeyse eşit olduđu bazı durumlarda tamamen etkisizdir.



Şekil-1: Damla bölünmesi, birleşmesi ve yer değıştirmesinin kritik vakalarını gösteren bir dizi görüntü [3].

Basit ve etkili bir yaklaşım olarak Sexton basitleştirilmiş bir segmentasyon algoritması kullanır [5]. Sistemlerini Parisli bir tren istasyonunda test ettiler ve %1 ila %20 arasında bir hata aldılar. Sistemleri, insanları arka plandan izole etmek için basit bir arka plan çıkarma yöntemi kullanır ve çerçeveler arasındaki eşleşmeyi yapmak için sadece damlaların ağırlık merkezlerini alır. Sistemin sağlamlığını artırmak için arka plan modeli sürekli olarak güncellenerek çevresel değışikliklerin sistem üzerindeki etkisini azaltır. Kamera, oklüzyonları azaltan ve damla tespit problemini basitleştiren üstten bir konuma asılmıştır.

Segmentasyon sonrası görüntü işleme üzerine yoğunlaşılın bir başka yaklaşımda, damlaları basit bir arka plan çıkarma yöntemiyle ayıklar ve sonra çerçeveler arasındaki özelliklerini izlerler [6]. Her damlanın yolları saklanır ve sanal bir çizgiyi geçerken kesişim noktasını ve yönünü tespit etmek için kullanılır. Bu sistem oklüzyon problemleriyle

uğraşmaz, bu nedenle kalabalık ortamlarda performansı büyük ölçüde azaldı. Yol kesişme algoritması performansı kalabalık ortamlarda da olumsuz etkilendi.

Haritaoğlu insanların gerçek zamanlı izlenmesi sorunu ile başa çıkmak için farklı bir yöntem benimsemiştir [7]. Ön planı bölümlere ayırmak için renk ve piksel yoğunluğuna dayalı bir arka plan çıkarması kullanırlar. Pikseller üç farklı gruba ayrılır: ön plan, arka plan ve gölge. Ön plan bölgeleri, iki hareket kısıtlaması kullanılarak bireysel insanlara ayrılır: geçici ve küresel. Bireyleri izlemek için, algoritma renk, kenar yoğunlukları ve ortalama kaydırma izleyicisine dayanan bir görünüm modeli kullanır.

Gary Conrad ve Richard Johnsonbaugh, oklüzyon problemini basitleştirmek için bir baş kamerası kullanmaktadır [8]. Aydınlatma değişikliği problemlerinin üstesinden gelmek için arka plan çıkarma kullanmak yerine ardışık kare farklılıkları kullanırlar. Sahip oldukları sınırlı hesaplama gücü nedeniyle, algoritmaları görüntünün yalnızca küçük bir dikdörtgen alanında hareket edecek şekilde tasarlanmıştır. 7491 kişide %95,6 doğruluk oranı elde edebildiler.

Toyama, arka plan bakımı için uyarlanabilir bir sistem sunmaktadır. Wiener filtresini gerçekleştiren piksel düzeyinde bir bileşen, beklenen arka planı tahmin eder. Ön plandaki nesnelerin homojen bölgelerini dolduran bölge düzeyinde bir bileşen ve görüntüdeki ani, global değişiklikleri algılayan çerçeve düzeyinde bir bileşen sistemin diğer algoritmalarla karşılaştırılması, beş temel bakım ilkesi oluşturmalarını sağladı [9].

Hoprasert toplanan görüntülerden elde edilen istatistikleri kullanarak parlak kısımları ve gölgeleri algılamayı önerdi [10]. Parlaklık ve renk sapması, pikselleri dört sınıfa ayırmak için dört eşik değeri kullanılır. Ortalama değeri [12] 'de referans görüntü olarak kullanan yöntem, dinamik arka plan için uygun değildir. Ayrıca, eşik değerleri, belirli bir algılama oranıyla parlaklık bozulması ve renk sapması histogramı temel alınarak tahmin edilir ve piksel değerlerinden bağımsız olarak tüm piksellere uygulanır. Bu nedenle, koyu piksel değerini gölge olarak sınıflandırmak mümkündür. Ayrıca, arka plan geçmişini kaydedemez.

Yeni bir 3D koni şekilli aydınlatma modeli önerilmiştir ve uzun vadeli renk tabanlı bir arka plan modeli ile kısa vadeli renk tabanlı bir arka plan modeli birleştirilmiştir [11].

Açık nesne bölümlleme veya izleme kullanmadan, farklı yönlerde seyahat eden yayalardan oluşan homojen olmayan kalabalıkların büyüklüğünü tahmin etmek için bir gizlilik koruma sistemi sunulmaktadır [13]. İlk olarak, kalabalık dinamik dokular hareket modeli karışımı kullanılarak homojen hareket bileşenlerine ayrılır. İkincisi, her bölümlenmiş bölgeden bir dizi basit bütünsel özellik çıkarılır ve özellikler ile bölüm başına kişi sayısı arasındaki yazışma Gauss Süreci regresyonu ile öğrenilir.

İzlenen nesneleri temsil etmek ve eşleştirmek için nicelleştirilmiş yerel özellik tanımlayıcılarının histogramı kullanılmıştır [14]. Bu yöntemin, tanımlayıcıların önceden çıkarılabildiği görüntü alma uygulamalarında nesne eşleştirme ve sınıflandırma için etkili olduğu kanıtlanmıştır. Bu sistem gerçek zamanlı gereksinimlere yaklaşır.

Satish, insan sayma sisteminin nerelerde kullanılacak olduğunu ayrıntılı bir şekilde anlatılmıştır [15]. Güvenlik uygulamaları için tahliye durumunda olduğu gibi, herhangi bir zamanda gözetleme alanında kaç kişinin bulunduğunu bilmek, yaya trafiğini yönetmek, turistlerin akış tahminlerini bilmek önemlidir. İki önemli unsur var, birincisi insanları tespit etme şimdi de yönlerini analiz etme. Sistem öncelikle insanları tespit eder, daha sonra kalman filtresi ile güzergahları takip ederek giriş mi çıkış mı yaptıklarını tahmin ederek sayacı günceller. Kalman filtresini ayrıntılı bir şekilde açıklanmıştır. İnsanların giriş mi çıkış mı yaptıklarını anlamak için iki yöntem vardır. Birincisi tek referans çizgisi kullanmak ikincisi ise iki referans çizgisi kullanmaktır. Sonuç olarak bu sistem kalabalık ortamlar için verimliliği kişi sayısı ile ters orantılıdır.

Yüz tanıma dayalı insan sayma sistemi, farklı ışık ve arka planlarda çalıştırılarak performansları gözlemlenmiştir [16]. NCC renk uzayına dayanarak, adayın cilt rengi ve boyutuna bakarak tespit eder. Yüz tanımadan sonra, kişinin yüzünü takip edip yüzünü sayarak izler. Kalabalık ortamlar için %80 doğruluk oranı korunmuştur. Görüntü işlemenin avantajları sıralanmıştır. Bu makalede insan tabanlı sayma sistemi kullanılmaktadır. Önerilen sistem dört bölümden oluşur. Hareketli kalabalığı bölümlendirme, ten rengi tespiti, yüz tanıma, izleme ve sayma şeklindedir. Bu bölümlendirmeler makale içinde ayrıntılı bir şekilde bölümler halinde anlatılmıştır. Örnekler verilmiştir. Sonuç olarak insan yüzüyle algılama yöntemi kapalı alanda %99 oranında doğruluk payı varken dışarıda %80 oranına

kadar düşmektedir. İnsan yüzü, maske ve saç gibi etkenler hataya sebep olmaktadır.

Stereo kamera kullanarak geçitten geçen insanları otomatik olarak sayma yöntemini önerilmiştir [17]. Önerilen yöntemde, stereo kamera kapının tavanından asılır ve kameranın optik eksenini, geçen insanların sadece yukarıdan gözlenebilmesi için ayarlanır. Çünkü bu sistem düzenlemesinde, eğer kapıda kalabalık bir insan grubu varsa, o zaman geçen insanların görüntü verileri, elde edilen görüntüler üzerinde birbiriyle örtüşmez. Ek olarak, geçen insanların her bir yüksekliği, piramitleştirmeye dayalı algoritma uygulanarak ölçülmüş. Stereo kamera kullanarak yapılan bu sistemin en büyük avantajı kamera tam tepede olduğu için birbirini kapatan görüntüler olmayacaktır. Bu sayede doğruluk payı daha yüksek ölçümler yapılmaktadır.

Kişi sayma sistemlerinde yeni metot olarak birden fazla kamera kullanmayı önermektedir [18]. Bu yöntemde tek bir kamera için uyarlanabilir kayan pencere (adaptive sliding window) algoritması tasarlanır ve eşyazımı (homography) teorisine göre birçok kamera arasındaki ortak görüntüler (aynı sahnenin farklı görünüşleri) hesaplanır. Tıkanma faktörünü göz önünde bulunduran nesne eşleştirme stratejisi, tek kameranın sonuçlarını iyileştirmek ve birden fazla kameranın sayımını elde etmek için tasarlanmıştır. Deney sonuçları, bu yöntemin karmaşık durumlarda iyi performans gösterdiğini göstermiştir. Ancak gerçek zamanlı performansı düşüktür.

Sensör olarak, derinlik verisi sağlayan Kinect kullanılan çalışmada [19], Kinect sensör, 11-bit derinliğe sahip VGA çözünürlüğünde (640×480 piksel) monokrom derinlik algılama video akışına sahiptir ve 2,048 hassasiyet seviyesi sağlar. Akış şemasını kısaca özetleyecek olursak ilk önce, derinlik verileri gürültüden temizlenir, ardından alakasız verileri ortadan kaldırmak için arka plan çıkarma yöntemi uygulanır. Bundan sonra su doldurma (water filling) yöntemi kullanılarak yerel maksimum noktalar bulunur ve bu görüntünün sonucu olarak filtreleme yapılır.

3) PROBLEM TANIMI

Kişi izleme problemine yaklaşım, üç kategoride sınıflandırılabilir farklı yöntemler kullanılarak yapılabilir:

- Bölge izleme özelliklerini kullanan yöntemler. Bu yöntemler piksel renklerine ve / veya dokulara dayalı bir sınıflandırma şeması kullanır.
- İnsanların 2B görünümünü kullanan yöntemler. Bu yöntemlerde, algoritmanın amacı belirli bir insan modelini görüntüdeki insan (lar) ile eşleştirmektir.
- Tam bir 3D model oluşturmak için birden fazla kamera kullanma yöntemleri. Bu yaklaşım, 3B referans çerçevesinde ilgi çekici noktaları oluşturmak için birden fazla kamera kullanır.

Mümkün olduğunca, mevcut güvenlik kameralarını kullanmak istediğimizden, bölge izleme (region tracking) kullanarak geleneksel bir yöntem kullanmayı tercih ettik. Bu yöntem, daha önce bahsedilen diğer yöntemlerden en esnek olmasının yanı sıra başlangıçta minimum kalibrasyon ve eğitim gerektirir.

➤ Hedefler

- Çalışma kapsamında binalarda bulunan kişi sayısını, anlık olarak yüksek doğrulukla verebilen, uygulanabilir bir sistem gerekmektedir.
- Sistemin mevcut güvenlik kameralarına entegre edilebilir ve mevcut internet altyapısını bağlanabilir olması hedeflenmektedir.
- Algoritma hızlı, basit ve güvenilir olmalıdır.
- İşlenen bilgilere acil durumlarda dahi internetten kolayca ulaşabilmek için sistemin bir Ethernet ağının parçası olması gerekmektedir.
- Sistem yüksek kullanılabilirlik özelliklerine sahip olmalıdır. Sistemin 24 saat / 7 gün iş başında gözetimsiz çalışması gerekiyor ve elektrik kesintileri, kendini yeniden başlatma ve tüm müdahaleleri ve kalibrasyonları insan müdahalesi olmadan yükleyebilmesi gerekiyor.
- Görüş alanı içerisinde veya sayım çizgisi yakınlarda uzun süre beklemeler, tekrarlı giriş-çıkışlar ihmal edilebilmelidir.

➤ **Zorluklar**

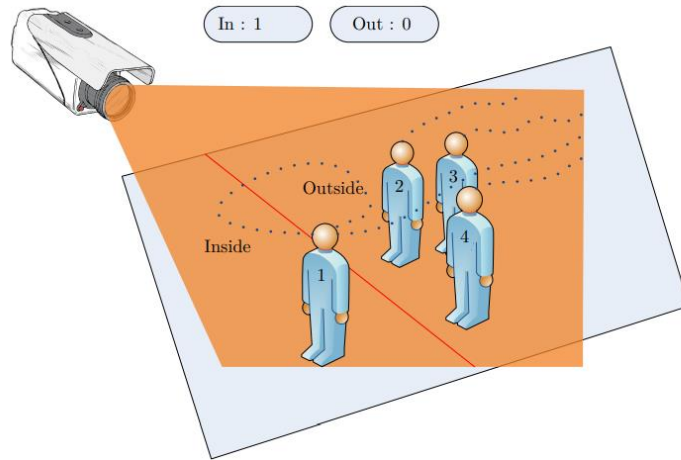
- Acil durum anlarında çok sayıda insan geçecektir.
- Sayım bölgesinin yanında çok sayıda insan olabilir. (Sayım bölgelerinde beklemler olacaktır)
- Ana girişteki aydınlatma koşulları ya güneşin gün içindeki konumunun değişmesi ya da binanın çalışma kuralları nedeniyle tutarsız olacaktır. Ana girişteki sayım kritik önem arz etmektedir. Bu alandaki ziyaretçileri saymak standart bir kapıdan veya düz bir koridordan geçen tek bir kişiyi tespit etmemiz gereken basit bir durumdan çok daha zordur. Acil durum senaryolarında bu girişlerdeki yoğun insan gruplarını saymak gerekir. Bu nedenle, birbirine yakın bir grup içerisindeki her bireyin izlediği yolları takip edebileceğimiz hassas bir izleme sistemi gerekiyordu.
- İzleyeceğimiz alanda çok parlak ve çok karanlık alanlar olabileceğinden. Görüntünün her pikselinde iyi bir piksel bilgisine sahip olmak için, kamera çok iyi bir dinamik aralığa sahip olmalı ve bu nedenle, her piksel için iyi bir renk bilgisine sahip olmalıdır.

4) YÖNTEM / TEORİ

Proje kapsamında 3.Bölümde belirtilen hedeflere ulaşabilmek ve zorlukları aşabilmek için kapsamlı literatür taramasının ardından bu bölümde bahsedilecek tasarım ve yöntemlerde karar kılındı.

4.1 Bilgisayarlı Görme Araçları

Kişi sayan bir sistem tasarlayabilmek için görüntü işleme yapabilen bilgisayar araçlarını öğrenerek, bu şekilde farklı görüntüleme algoritmalarının hızlı bir şekilde test edilmesi gerekiyordu. Literatür taramasının ardından Açık Kaynak Bilgisayarlı Görme (OpenCV) tercih edildi. OpenCV, Linux, Windows ve Mac OS X'te çalışan ve Python, Ruby, Matlab ve diğer diller için ara yüzlerde aktif gelişim sağlayan açık kaynaklı bir bilgisayarlı görme kütüphanesidir. OpenCV, hesaplama verimliliği ve gerçek zamanlı uygulamalara büyük önem verilerek tasarlanmıştır. Optimize edilmiş C ve C++ ile yazılmıştır ve çok çekirdekli işlemcilerin avantajlarından yararlanabilir. OpenCV'nin hedeflerinden biri, insanların hızlı bir şekilde görme uygulamaları oluşturmalarına yardımcı olan kullanımı kolay bir bilgisayar görme altyapısı sağlamaktır. OpenCV kütüphanesi ile, fabrikada ürün denetimi, tıbbi görüntüleme, güvenlik, kullanıcı arayüzü, kamera kalibrasyonu, stereo görüş ve robotik gibi birçok işlev başarılabılır. Bilgisayarla görme ve makine öğrenimi genellikle el ele gittiğinden, OpenCV genel amaçlı bir Makine Öğrenim Kütüphanesi de içerir.



Şekil 2- Bilgisayarlı görme uygulaması olarak kişi sayan kamera

4.2 OpenCV Kütüphaneleri

- **NumPy**: NumPy, Python ile bilimsel hesaplama için temel pakettir. Diğer şeylerin yanı sıra: C / C ++ ve Fortran kodunu kullanışlı lineer cebir, Fourier transformu ve rastgele sayı yeteneklerini entegre etmek için gelişmiş bir N-boyutlu dizi nesnesi sofistike (yayın) fonksiyon araçları. NumPy, açık bilimsel kullanımlarının yanı sıra verimli çok boyutlu jenerik veri konteyneri. Rasgele veri türleri tanımlanabilir. Bu, NumPy'nin çok çeşitli veri tabanlarıyla sorunsuz ve hızlı bir şekilde entegre olmasını sağlar.

- **Dlib**: Dlib, gerçek dünya sorunlarını çözmek için C ++ 'da karmaşık yazılımlar oluşturmak için makine öğrenme algoritmaları ve araçlar içeren modern bir C ++ araç takımıdır. Hem endüstride hem de akademik çevrede robotik, gömülü cihazlar, cep telefonları ve büyük yüksek performanslı bilgi işlem ortamları dahil olmak üzere çok çeşitli alanlarda kullanılır. Dlib'in açık kaynak lisansı, herhangi bir uygulamada ücretsiz olarak kullanmanızı sağlar.,

- **Imutils**: Çeviri, döndürme, yeniden boyutlandırma, iskeletleştirme ve Matplotlib görüntülerini OpenCV ile daha kolay görüntüleme gibi temel görüntü işleme işlevlerini kolaylaştıran bir dizi kullanışlılık kütüphanesi

- **Datetime**: Datetime modülü tarih ve saatleri işlemek için sınıflar sağlar. Tarih ve saat aritmetiği desteklenirken, uygulamanın odağı çıktı biçimlendirme ve manipülasyon için etkili öznitelik çıkarma üzerinedir.

- **Collections**: Bu modül, Python genel amaçlı yerleşik kaplara, dikte, liste, küme ve demetlere alternatifler sunan özel konteyner veri türlerini uygular.

- **Math**: Bu modül C standardı tarafından tanımlanan matematiksel fonksiyonlara erişim sağlar. Bu fonksiyonlar karmaşık sayılarla kullanılamaz;

- **Matplotlib**: Veri görselleştirmesinde kullandığımız temel python kütüphanesidir. 2 ve 3 boyutlu çizimler yapmamızı sağlar. Matplotlib genelde 2 boyutlu çizimlerde kullanılırken, 3 boyutlu çizimlerde başka kütüphanelerden yararlanılır

4.3 OpenCV ile Nesne Tespit Etme / Nesne İzleme

Kişi sayma sistemlerinden önce teoride nesne izleme ve nesne algılama arasındaki temel farklar ve beraber kullanımları anlaşılmalıdır.

Nesne algılama uyguladığımızda, bir nesnenin görüntüde / karede nerede olduğunu belirleriz. Bir nesne detektörü ayrıca bir nesne izleme algoritmasından hesaplama açısından daha fazla iş gerektirir ve bu nedenle daha yavaştır. Nesne algılama algoritmalarına örnek olarak Haar, YOLO ve Tek Atış Dedektörleri (SSD'ler) gibi derin öğrenme tabanlı nesne dedektörleri verilebilir. Sınıflandırıcılar için nesneler derin öğrenme yöntemleri ile öğretilir. Şablon eşleştirme yöntemi ise, aranan görüntü şablonunu kaynak görüntü üzerindeki tüm piksellerde dolaşarak eşleştirme işlemi yapar. Öte yandan, bir nesne izleyici, bir nesnenin görüntüde bulunduğu yerin (x, y) koordinatlarını kabul eder ve söz konusu nesneye benzersiz bir kimlik atar. Bir video akışında hareket ederken nesneyi, çerçevenin çeşitli niteliklerine (gradyan, optik akış vb.) Dayalı olarak bir sonraki karedeki yeni nesne konumunu tahmin ederek izler.

Son derece hassas nesne izleyiciler, nesne algılama ve nesne izleme kavramını, genellikle iki aşamaya bölünmüş tek bir algoritma halinde birleştirir.

➤ Aşama 1-Algılama: Algılama aşamasında, görüşümüze yeni nesnelerin girip girmediğini tespit etmek ve izleme aşamasında “kaybolan” nesneleri bulup bulunamayacağını görmek için nesne izleyici çalıştırılır. Algılanan her nesne için yeni sınırlayıcı kutu koordinatlarına sahip bir nesne izleyici oluşturur veya güncellenir. Nesne detektörümüz daha fazla hesap gerektirdiğinden, bu fazı her N karede bir kez çalıştırılır.

➤ Aşama 2-İzleme: “Tespit” aşamasında değilken “izleme” aşamasına geçilir. Tespit edilen nesnelerin her birine, nesneyi çerçeve etrafında hareket ederken izlemek için bir nesne izleyici oluşturulur. Nesne izleyici, nesne dedektöründen daha hızlı ve daha verimli olmalıdır. N'inci kareye ulaşıncaya kadar izlemeye devam edilir ve ardından nesne detektörü yeniden çalıştırılır.

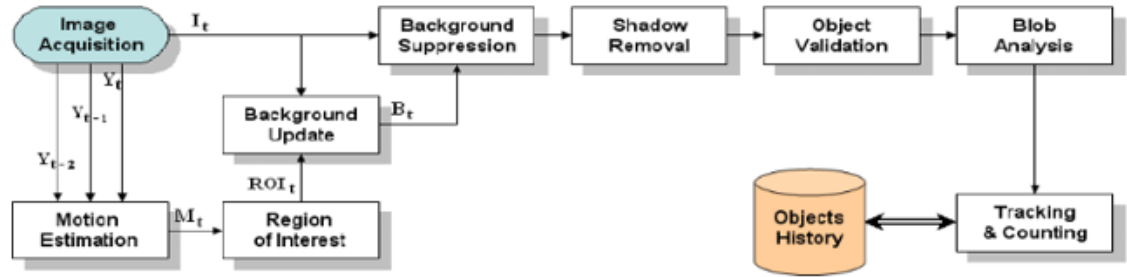
Bu melez yaklaşımın yararı, hesaplama yükünün büyük bir kısmı olmadan, yüksek oranda doğru nesne tespit yöntemleri uygulanabilmesidir.

4.4 Derin Öğrenme ve OpenCV

OpenCV 2.4 sürümü ile birlikte derin öğrenme tarafındaki birçok gelişmeye yer verilmeye başlanmıştır. Görüntü sınıflandırma için farklı kütüphaneler ile oluşturulmuş modellerin içe aktarılabilmesi, farklı sınıflandırıcı ağların oluşturulabilmesi, model oluşturabilme vb. OpenCV 3.3 sürümü ile ise neredeyse nihai bir dağıtım oluşturuldu ve Caffe, TensorFlow, Torch DarkNet gibi yapılar için destek sağlandı. Popüler; AlexNet, GoogLeNet , ResNet, SqueezeNet, , VGG, ENet , VGG-based SSD, MobileNet-based SSD gibi ağlar için ise destek sağlanmaktadır. Bu projede ise bugün için en güncel sürüm olan 3.4 sürümü kullanılmıştır.

4.5 Sistem Tasarımı

İnsan sayımı için önerilen sistem esas olarak, Şekil-1’de gösterilen adımları içermektedir. Birinci adımda karelerin elde edilmesi ve arka plan kestirimi yapılır, ikinci adım da insanların tespiti yapılır, üçüncü adımda izleme yapılır ve sonunda insanlar sayılır. İnsanların yörüngelerini oluşturmak için izleme yapılır. Yörüngelerin tarifleri sayesinde sayaçlar güncellenir. Bu algoritmalar OPENCV programı üzerinde derlenmiştir.



Şekil-3: Sistemin Akış Şeması

Saymanın doğru çalışması için sistemin arka planı doğru şekilde güncellenmeli ve insanlar başarıyla tanıtılmalıdır. Sistemimizde arka plan modellemesi karşılaştırma aşaması ve güncelleme aşaması olarak iki aşamaya ayrılmıştır.

Belli sayıda kare üzerinde karşılaştırma aşamasında, görüntünün her pikselinin minimum/maksimum yoğunluk değerleri elde edilir ve güncelleme aşamasında arka plan karşılaştırma aşaması ve güncelleme aşaması denklemleriyle güncellenir.

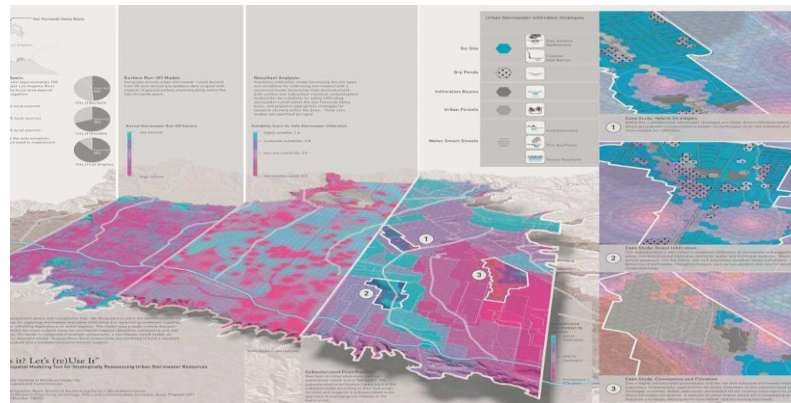
Daha iyi ön plan elde etmek için, arka planın mevcut çerçeveden çıkarılmasından elde edilen fark çerçevesine, morfolojik işlemler uygulanır. İkili ön plan görüntüsünü elde etmek için ilk fark görüntüsü eşiklenir. Eşiklendikten sonra, kendisi tarafından üretilen küçük delikleri doldurmak için Dilation (Yayma- Genişletme) uygulanır.

Tespit edilen kişilerin takibi ile farklı yollar inşa edilir. Bu durumda, kamera yakalama hızı, insanların takibi için zaman içinde doğrusal olarak kabul edilen ardışık karelerdeki bireysel hareketlerin gerçekleşmesi için yeterli olmalıdır. Burada, insanların hareketi doğrusal olarak kabul edilir, böylece Kalman filtresi izleme için çok uygundur. Bu yüzden kalman kanalı tarafından sağlanan tahminlerin kullanıldı. İnsanların durumlarını inşa etmek için Kalman filtresi için durum değişkenleri aşağıdaki gibidir.

Sayma sistemimizde denetlenen alan yaklaşık 3 ila 4 metre uzunluğundadır, böylece model sabit ivmelenme olarak kabul edilir ve izleme için sadece hız dikkate alınır. Ayrıca, kişinin gerçek konumunu ve hızını bilmek gerekmez, kişi görüntüdeki konumu ve hızına göre izlenebilir.

Kalman Filtresi, iki aşamaya ayrılan tahmin ve değiştirilmiş doğrusal ve özyinelemeli yöntemdir. Tahmin aşaması hareketi tahmin ederken düzeltme aşaması tahmin aşaması tarafından öngörülen hareketteki hataları düzeltir.

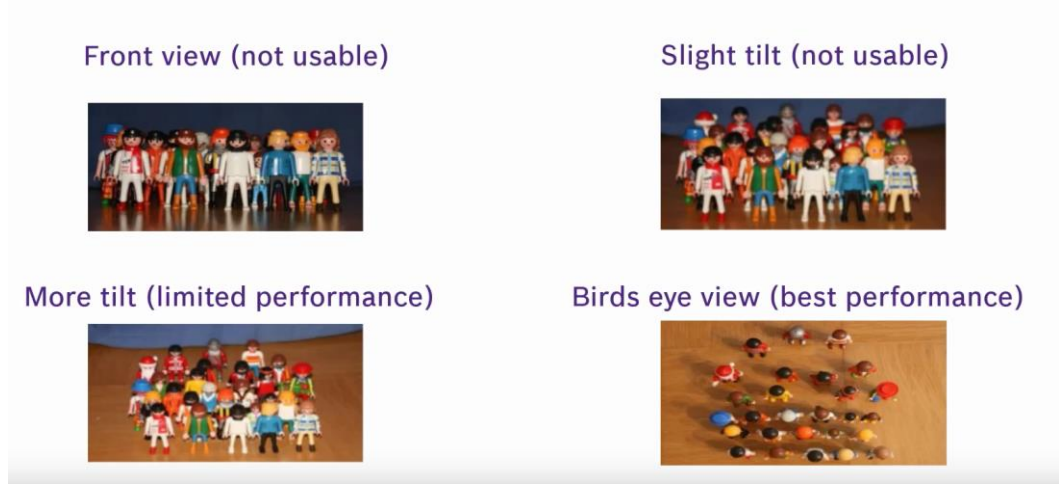
Denetlenen alan içerisinde girip çıkan insanları saymak için farklı algoritmalar önerilmiştir. İlk önce tek referans çizgisi, ikincisi ise çok referans çizgisi olmak üzere iki tür teknik vardır. Bu projede tek referans çizgi yöntemi kullanılacaktır. Kişi denetlenen alana girdiğinde, o kişinin yörüngeleri üretildi ve yönlendirilerek sayıldı. Daha sonra bu bilgiler işlemci tarafından internet aracılığıyla Şekil-2’de bir taslağı gösterilen arayüz sistemi ile son kullanıcılara iletilecektir.



ŞEKİL-4 [11]

4.6 Kamera Yerleşimi

İnsan grupları kameranın görüş alanından geçtiğinde tıkanıklıklardan kaçınmak için genel kamera kullanımı genellikle en iyi seçenektir. Bu yerleşim mümkün olduğunca tercih edilmelidir.



ŞEKİL-5

4.7 Arka Plan Bakımı

Piksel seviyesi bileşeni, beklenen arka planın olasılıklı tahminlerini yapmak için Wiener filtrelemesi gerçekleştirir; Bölge düzeyinde bileşen ön plandaki nesnelerin homojen bölgelerini doldurur; Çerçeve düzeyindeki bileşen görüntüdeki ani, küresel değişiklikleri algılar ve arka planın daha iyi tahminleriyle takas eder.

Wiener filtresi, yakın tarih geçmişine dayanan doğrusal bir tahmin edicidir. Öngörülen değerinden önemli ölçüde sapan herhangi bir piksel ön plan olarak bildirilir. Belirli bir piksel için doğrusal tahmin şu şekilde verilir:

$$s_t = - \sum_{k=1}^p a_k s_{t-k}$$

s_t , t karesindeki öngörülen piksel değeridir, s_{t-k} pikselin geçmiş bir değeridir, a_k tahmin katsayılarıdır. Böylece beklenen kare tahmin hatası;

$$E[e_t^2] = E[s_t^2] + \sum_{k=1}^p a_k E[s_t s_{t-k}]$$

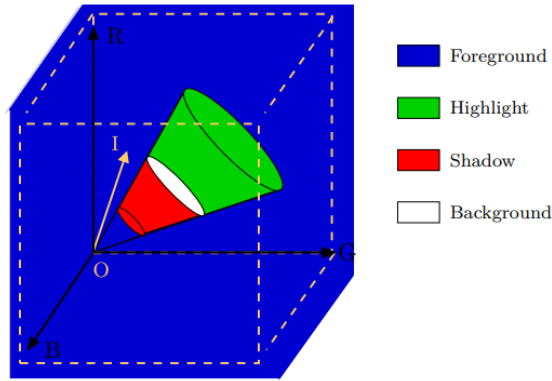
Burada tarif edilen çerçeve seviyesi bileşeni, daha önce gözlenen durumlara hızlı

adaptasyon için bir yöntem gösterdiği için de dikkate değer bir öneme sahiptir. Sahne arka plan modellerinin temsili setleri tutulur ve daha sonra aralarında otomatik bir mekanizma değişir. Modellerin ne zaman değiştirileceğine karar vermek, görüntüdeki ön plan piksellerinin oranını izleyerek yapılır. Bu kesir 0,7'yi aşarsa, yeni bir modelin değerlendirilmesi tetiklenir. Seçilen model, geçerli görüntüdeki ön plan piksellerinin oranını en aza indiren modeldir. Modül, ışık anahtarı senaryosuna karşı başarıyla test edildi.

4.8 Gölge Segmentasyonu

Bu süreç sistemimizde kritik öneme sahiptir, çünkü bu süreç olmadan gölgeleri sahiplerinden ayırt edemeyiz.

RGB renk uzayında parlaklığı renklilikten ayıran bir renk modeli önerilmektedir [13]. Bu modelde (Şekil 4.2) arka plan görüntüsü önce istatistiksel olarak piksel olarak modellenmiştir. Sonra her bir i piksel beklenen renk E_i , standart sapma s_i ve mevcut renk I_i ile temsil edilebilir. I_i ve E_i arasındaki fark parlaklık (α_i) ve kromatiklik CD_i bileşenlerine ayrılır.



Şekil-6: RGB renk uzayında önerilen 3D koni modeli.

5) BİRİNCİ DÖNEM GERÇEKLEŞTİRİLENLER

Bu bölümde Ekim 2019 – Şubat 2020 tarihleri arasında projeye ilgili gelişmelerden, öğrenilen yazılım programlarından, karşılaşılan bazı problemlerden ve bunların nasıl aşıldığından bahsedilmektedir.

Proje kapsamında akıllı şehirlere hizmet edebilecek, acil durum anlarında anlamlı veriler oluşturabilecek bir sistem tasarlamayı hedeflerken, projedeki en önemli unsur sistemin uygulanabilirliği ve maliyeti olarak ön plana çıkmaktaydı. Öyle ki sistemin kullanıldığı bina ve mekân sayısı ne kadar fazla ise, buradan elde edilen veriler o kadar anlamlı ve etkili hale gelecektir. Bu nedenle öneri raporunda bahsedilen sistem yerine binalardaki güvenlik kameralarını kullanabilen bir sistemde karar kılındı.

Sistemdeki bir diğer önemli kriter ise güvenilir veri sağlayabilmek ve bunu acil durum anlarında raporlayabilmektir. Bu bağlamda ise oldukça kapsamlı bir literatür taraması yapılarak bölüm 5’te bahsedilen derin öğrenme tabanlı bir kişi sayma algoritması geliştirildi. Bu süreçte OpenCV ile görüntü işleme konusu detaylı olarak araştırıldı ve gerekli yetenekler kazanıldı.

Binaların çok büyük kapıları ve salonları ile karakterize edilen benzersiz mimari tasarımı olabilir ve binada meydana gelen farklı faaliyetler için hareket içerisinde olan çok sayıda insan nedeniyle karmaşık bir görevi temsil eder. Bu zorluğun teknik çözümünü elde etmek için, ilk olarak video kameraları ile algılanan insanlar için çeşitli görüntü işleme yöntemleri incelenmiş, kapsamlı literatür taraması yapılmıştır. Bir sonraki adım, bireyleri istenen doğrulukta saymak için OpenCV kütüphanelerini ve bilgisayar görme kavramlarını kullanarak gerçek zamanlı bir algoritmanın geliştirilmesiydi. Bu nedenle OpenCV ile görüntü işleme algoritması yazabilmek üzerine gerekli yetenekler kazanıldı. Bu algoritma, önceki yöntemlerin incelenmesinde elde edilen bilimsel ve teknik bilgileri içerir. Bu tezde geliştirilen temalar, arka plan bakımı, gölge ve açık ton algılama ve damla algılama ve izleme alanlarını içermektedir. Bir sonraki adımda ise kameranın konumu ve açısına karar verilip, uygun kamera araştırması yapıldı. Ayrıca, önerilen sistemin geliştirilmesine, test edilmesine ve düzeltilmesine yardımcı olmak için, çalışmayı tamamlayan bir grafik arayüz de inşa

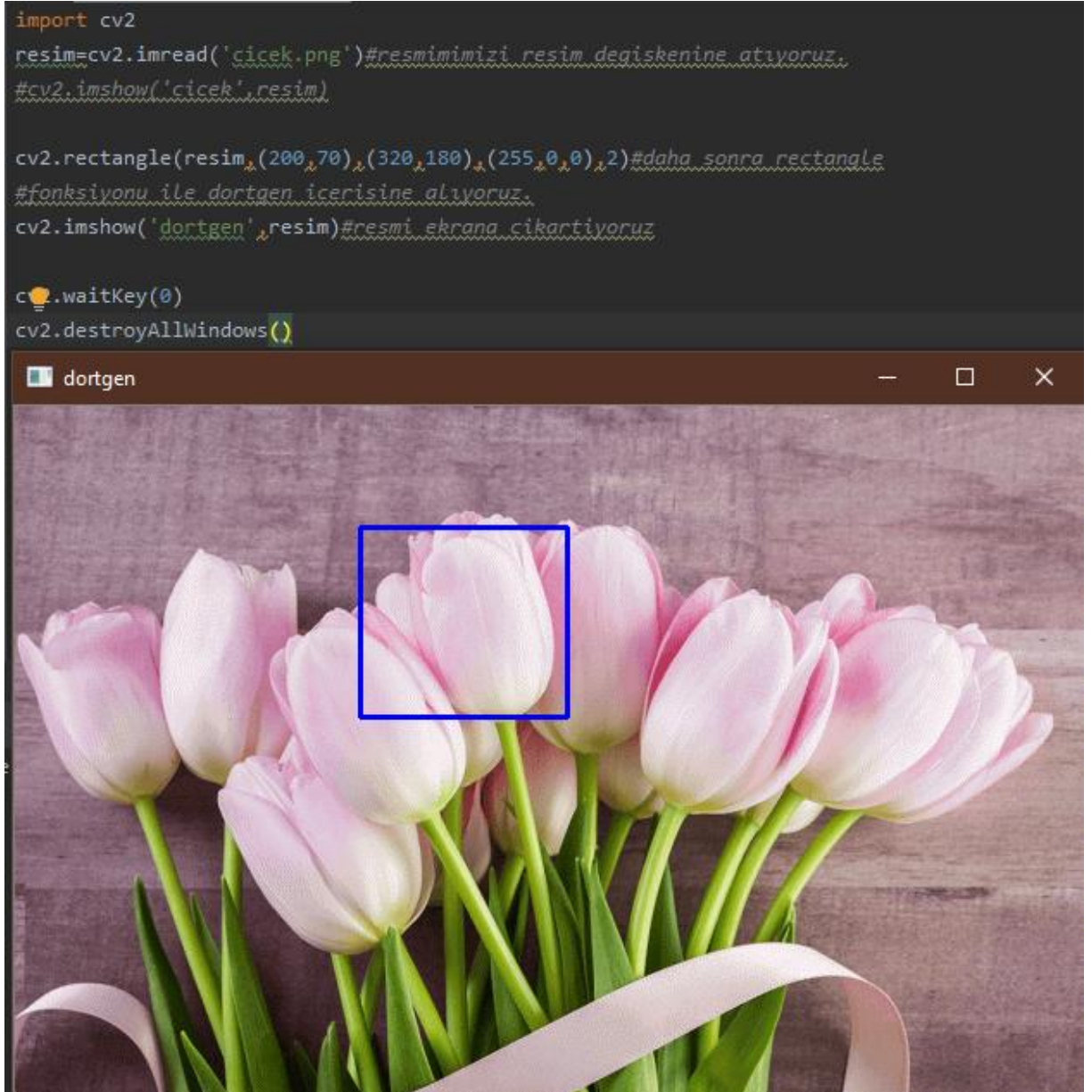
edilecektir ve bir dizi sınırlı kořula karşı sertifikalandırmak için sisteme yönelik testler yapılacaktır. Elde edilen sonuçların, algoritmanın karmaşık ortamlardaki insan sayısını güvenilir bir doğrulukla saymak için başarıyla uygulandığını ortaya koyması hedeflenmektedir.

Bu kısmın devamında ise geçtiğimiz dönem içerisinde öğrenilen ve kullanılan algoritmalarından bahsedilmektedir.

5.1 OpenCV ile Görüntü İşleme

5.1.1 Resim Okuma ve Yazma İşlemleri

Dışarıdan png, jpeg dosyalarımızı PyCharm ortamına aktarmayı ve istediğimiz alanı dörtgen içerisine alındı.

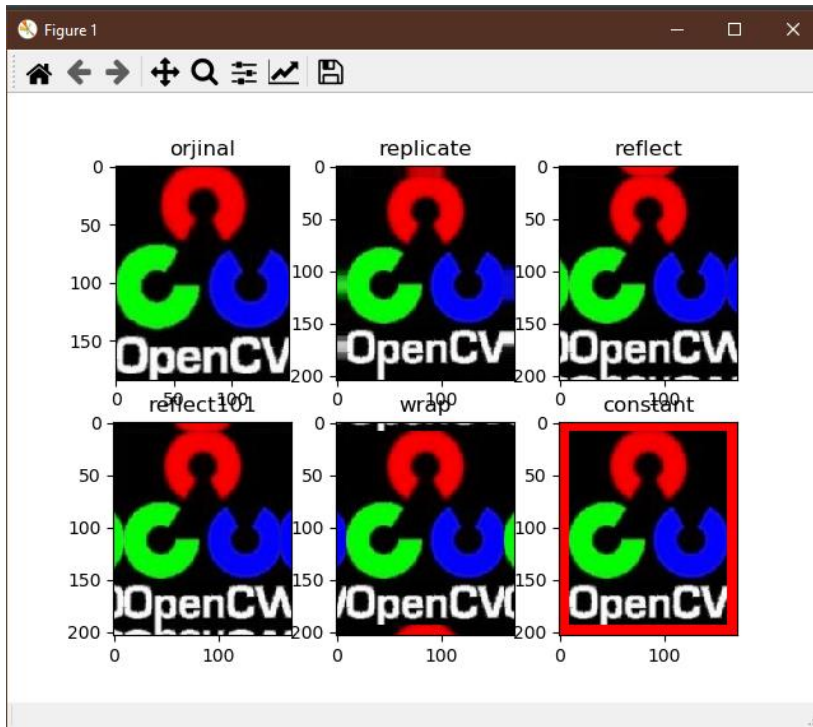


ŞEKİL-7 : Resim okuma ve yazma işlemleri

5.1.2 Çerçeveleme

Dışarıdan png, jpeg dosyalarımızı PyCharm ortamına aktardıktan sonra resmin çerçevesiyle ilgili değişiklikler yapıldı.

```
img=cv2.imread('opencv.JPG')
replicate=cv2.copyMakeBorder(img,10,10,10,10,cv2.BORDER_REPLICATE)
reflect=cv2.copyMakeBorder(img,10,10,10,10,cv2.BORDER_REFLECT)
reflect101=cv2.copyMakeBorder(img,10,10,10,10,cv2.BORDER_REFLECT101)
wrap=cv2.copyMakeBorder(img,10,10,10,10,cv2.BORDER_WRAP)
constant=cv2.copyMakeBorder(img,10,10,10,10,cv2.BORDER_CONSTANT,value=mavi)
```



ŞEKİL-8 :Çerçeveleme

5.1.3 Aritmetik İşlemler

Bir görüntü çeşitli işlemlere sokularak diğer görüntünün üzerine eklenildi.

```
im2gray=cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
cv2.imshow('im2gray',im2gray)
ret, mask=cv2.threshold(im2gray,10,255,cv2.THRESH_BINARY)
mask_inv=cv2.bitwise_not(mask)
cv2.imshow('mask',mask)
cv2.imshow('mask_inv',mask_inv)
im1_bg=cv2.bitwise_and(roi,roi,mask=mask_inv)
cv2.imshow('im1_bg',im1_bg)
im2_fg=cv2.bitwise_and(img2,img2,mask=mask)
cv2.imshow('im2_fg',im2_fg)
son_resim=cv2.add(im1_bg,im2_fg)
img1[0:satir,0:sutun]=son_resim
cv2.imshow('son_resim',img1)
```

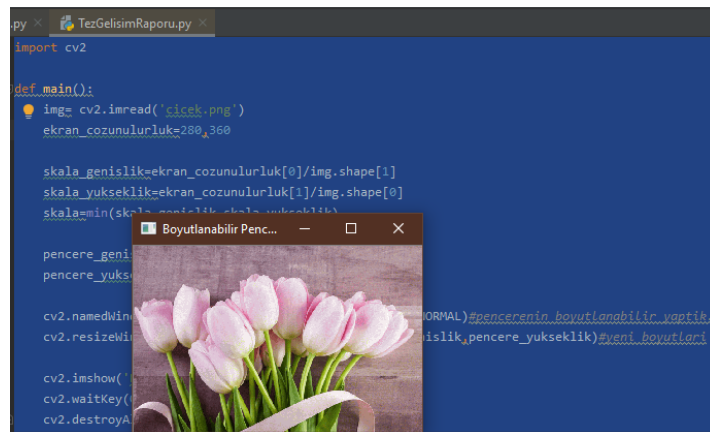


ŞEKİL-9: Aritmetik işlemler

5.1.4 Pencereyi Yeniden Boyutlandırma

Dışarıdan aldığımız görüntünün boyutları istenen boyutlara dönüştürülebilir.

```
cv2.namedWindow('Boyutlanabilir Pencere',cv2.WINDOW_NORMAL)#pencerenin
boyutlanabilir yaptik.
cv2.resizeWindow('Boyutlanabilir
Pencere',pencere_genislik,pencere_yukseklik)#yeni boyutlari
```

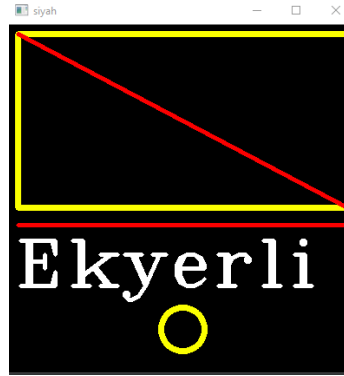


ŞEKİL-10: Pencereyi yeniden boyutlandırma

5.1.5 Görüntüye Metin Ekleme

Görüntünün üzerine istenen metin eklendi.

```
resim=np.zeros((400,400,3),dtype='uint8')
cv2.rectangle(resim,(10,10),(390,210),(0,255,251),5)
cv2.line(resim,(10,10),(390,210),(0,0,251),3)
cv2.line(resim,(10,230),(390,230),(0,0,251),3)
cv2.circle(resim,(200,350),25,(0,255,251),5)
cv2.putText(resim,'Ekyerli',(5,300),cv2.FONT_HERSHEY_COMPLEX_SMALL,4,(255,255,255),4,cv2.LINE_4)
```



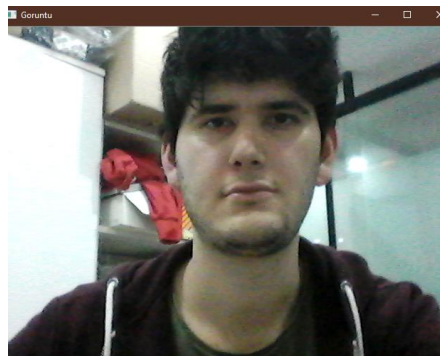
ŞEKİL-11: Görüntüye metin ekleme

5.1.6 Kameradan görüntü alma ve video görüntüleme

Kamera görüntüsü ve dışarıdan mp4 uzantılı dosya PyCharm ortamına aktarıldı.

```
kamera=cv2.VideoCapture(0)
while True:
    ret, goruntu=kamera.read()
    griton=cv2.cvtColor(goruntu,cv2.COLOR_BGR2GRAY)

    cv2.imshow('Goruntu',goruntu)
    cv2.imshow('Gri Ton',griton)
    if cv2.waitKey(25) & 0xFF==ord('q'):
        break
kamera.release()
cv2.destroyAllWindows()
```



ŞEKİL-12: Kameradan görüntü alma ve video görüntüleme

5.1.7 Video Kaydetme ve Döndürme

Videolar istenen açıyla döndürüldü ve harici bir mp4 uzantılı dosya olarak kaydedildi.

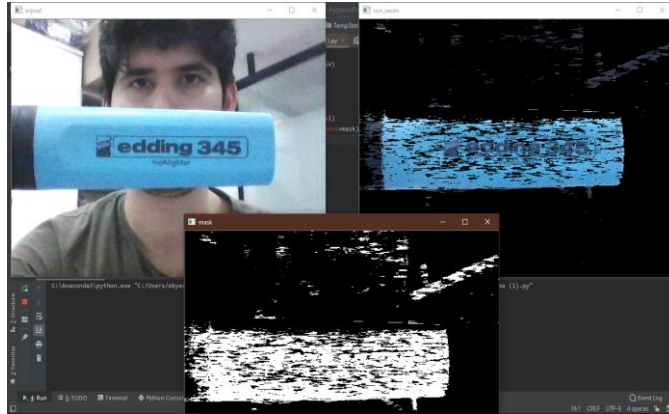
```
import cv2
kamera= cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
kayit=cv2.VideoWriter('kayit.avi',fourcc,20.0,(640,480))
while(1):
    ret, video=kamera.read()
    if ret ==True:
        #video=cv2.flip(video,0)
        kayit.write(video)
        cv2.imshow('kayit',video)
        if cv2.waitKey(0) & 0xFF == ord('q'):
            break
    else:
        break
kamera.release()
kayit.release()
cv2.destroyAllWindows()
```


5.2 Filtreleme İşlemleri

5.2.1 Renk Filtreleme

İstenen renkte filtre uygulanarak görüntü üzerinden o rengin geçirilmesi sağlandı.

```
kamera= cv2.VideoCapture(0)
while(1):
    ret, frame=kamera.read()
    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)#renk filtreleri için hsv'ye
    çeviriyoruz
    dusuk_mavi=np.array([100,60,60])#Mavi için eşik değerlerini veriyoruz.
    ust_mavi=np.array([140,255,255])#Mavi için eşik değerlerini veriyoruz.
    mask=cv2.inRange(hsv,dusuk_mavi,ust_mavi)
    son_resim=cv2.bitwise_and(frame,frame,mask=mask)
    cv2.imshow('orjinal',frame)
    cv2.imshow('mask', mask)
    cv2.imshow('son_resim', son_resim)
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break
```



ŞEKİL-13: Renk Filtreleme

5.2.2 Bulanıklaştırma

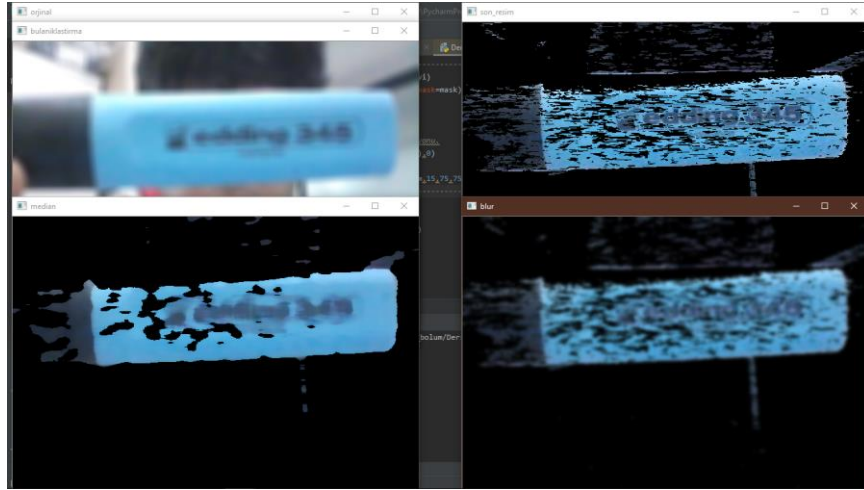
Görüntü bulanıklaştırma düşük geçişli filtreler ile resmi eriştilererek yapılır. Bu görüntü kirliliğini kaldırmak için iyi bir yöntemdir. Aslında yüksek frekanslı içeriği (ör. Gürültü, kenar) görüntüden kaldırır ve böylece filtre uygulandığında kenarları bulanık hale getirir. OpenCV, esas olarak dört tür bulanıklaştırma tekniği sunar.

```
while(1):
    ret, frame=kamera.read()
    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    dusuk_mavi=np.array([100,60,60])
    ust_mavi=np.array([140,255,255])
    mask=cv2.inRange(hsv,dusuk_mavi,ust_mavi)
    son_resim=cv2.bitwise_and(frame,frame,mask=mask)
    #en neti bu
```

```

kernel=np.ones((15,15),np.float32)/225
smoothed=cv2.filter2D(frame,-1,kernel)
#alttakilerin hepsi bulaniklastirma fonksiyonu.
blur=cv2.GaussianBlur(son_resim,(15,15),0)
median=cv2.medianBlur(son_resim,15)
bileteral=cv2.bilateralFilter(son_resim,15,75,75)
cv2.imshow('orjinal',frame)
cv2.imshow('son_resim', son_resim)
cv2.imshow('bulaniklastirma', smoothed)
cv2.imshow('median', median)
cv2.imshow('blur', blur)

```



ŞEKİL-14: Bulanıklaştırma

5.2.3 Canny Kenar Tespiti

Kenar tespiti için kullanılacak olan Canny için ise eşik-1 ve eşik-2 değerleri metot içerisine girildi ve şekilde gibi gösterdiğim nesne üzerinde kenar tespiti yapılmaktadır.

```

#kenar tespiti 3 yontem var laplacian
import cv2
import numpy as np

kamera= cv2.VideoCapture(0)
while(1):
    ret, frame=kamera.read()
    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    dusuk_mavi=np.array([100,60,60])
    ust_mavi=np.array([140,255,255])

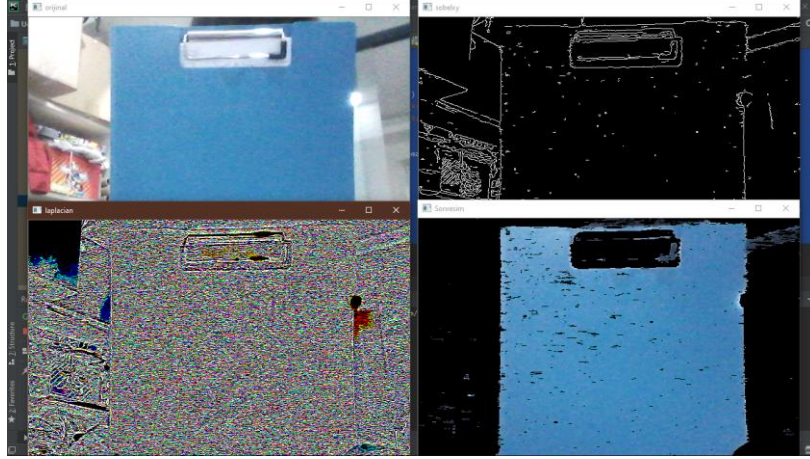
    laplacian = cv2.Laplacian(frame, cv2.CV_64F)
    sobelx = cv2.Sobel(frame, cv2.CV_64F, 1, 0, ksize=5)
    sobely = cv2.Sobel(frame, cv2.CV_64F, 0, 1, ksize=5)
    kenarlar=cv2.Canny(frame,115,175)
    mask=cv2.inRange(hsv,dusuk_mavi,ust_mavi)
    son_resim=cv2.bitwise_and(frame,frame,mask=mask)
    cv2.imshow('laplacian',laplacian)
    cv2.imshow('sobelxy', kenarlar)

```

```

cv2.imshow('Sonresim', son_resim)
cv2.imshow('orijinal', frame)
if cv2.waitKey(25) & 0xFF == ord('q'):
    break
kamera.release()
cv2.destroyAllWindows()

```



ŞEKİL-15: Canny Kenar Tespiti

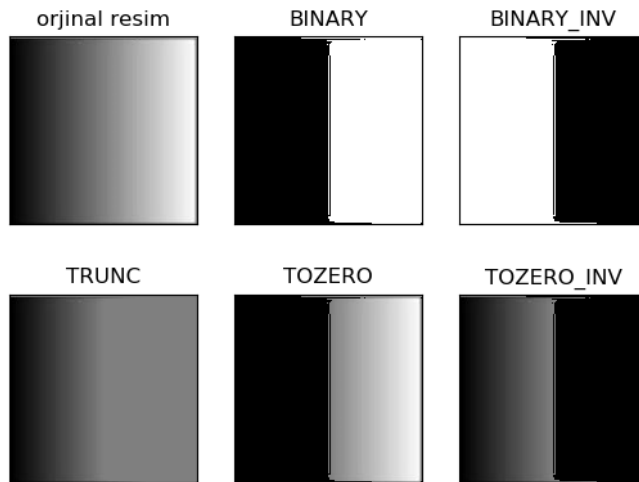
5.2.4 Basit Eşik Filtreleme

Belli eşik değerleri verilerek görüntümüzün siyah ve beyaz olması sağlandı.

```

ret, thresh1=cv2.threshold(resim,127,255,cv2.THRESH_BINARY)
ret, thresh2=cv2.threshold(resim,127,255,cv2.THRESH_BINARY_INV)
ret, thresh3=cv2.threshold(resim,127,255,cv2.THRESH_TRUNC)
ret, thresh4=cv2.threshold(resim,127,255,cv2.THRESH_TOZERO)
ret, thresh5=cv2.threshold(resim,127,255,cv2.THRESH_TOZERO_INV)

```

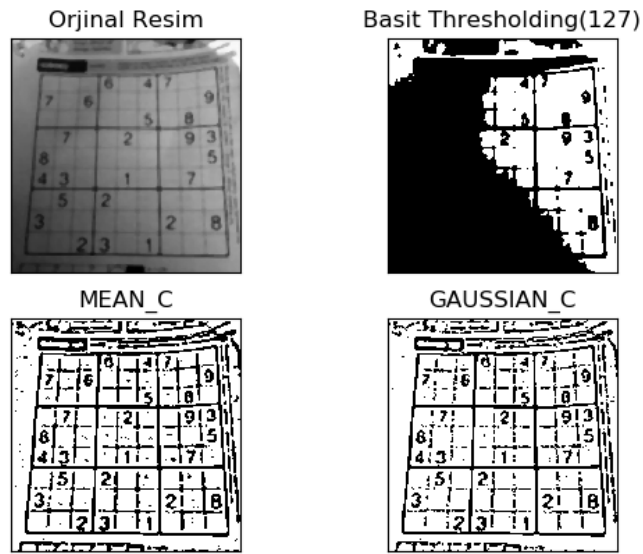


ŞEKİL-16: Basit Eşik Filtreleme

5.2.5 Gauss ve Ortalama Filtreleme

Eşik değeri kullanılarak görüntü üzerinde düzenlemeler yapıldı.

```
ret, th1=cv2.threshold(img,127,255,cv2.THRESH_BINARY)
th2=cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,
9,2)
th3=cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,
9,2)
```

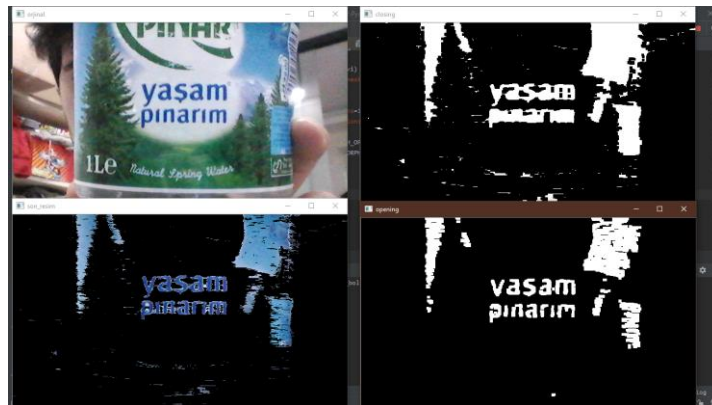


ŞEKİL-17: Gauss Filtreleme

5.2.6 Morfolojik İşlemler

Opencv de erosion(aşındırma), dilation(yayma genişletme), opening(açınım) ve closing(kapanım) olmak üzere 4 tane morfolojik işlem vardır.

```
opening=cv2.morphologyEx(mask,cv2.MORPH_OPEN,kernel)
closing = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
```



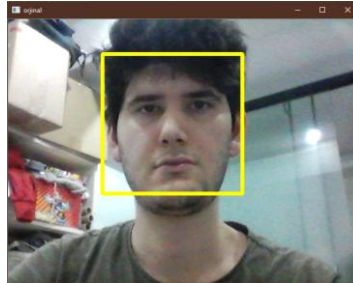
ŞEKİL-18: Morfolojik işlemler

5.3 Haar-cascade

Bu özellik ile görüntü üzerinde çizgi, kenar, yüz, göz, araç vb. birçok nesneyi tespit edebiliriz. Aşağıda bulunan bir insan yüzü görseli üzerinde Haar-cascade uygulanmıştır. Haarcascade metodu önceden birçok kez eğitilerek yüz şeklinin nasıl bir yapıda olduğunu bilmektedir. Bir çiçek tanıma veya plaka tanıma işlemi yaparken önceden plakayı birçok kez tanıtarak nasıl bir yapıda olduğunu tanıtlıyoruz. Örneğin bir yüz arayan bir sistem ilk olarak iki gözü aramaktadır. Göz var ise burun var mı diye bakar. Burun var ise kaş var mı diye kontrol ederek istenen sonuçları veren yapıdır.

5.3.1 Haar-cascade ile Resim Üzerinde Yüz Tespiti

```
kamera= cv2.VideoCapture(0)
yuz_cascade=cv2.CascadeClassifier('haarcascade-frontalface-default.xml')
while(1):
    ret, frame=kamera.read()
    griton=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    yuzler = yuz_cascade.detectMultiScale(griton,1.1,4)
    for(x,y,w,h) in yuzler:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,255),6)
    cv2.imshow('original',frame)
```



ŞEKİL-19: Haarcascade ile resim üzerinde yüz tespiti

5.3.2 Haar-cascade ile Vücut Tespiti

```
img=cv2.imread('r2.jpg')
body_cascade=cv2.CascadeClassifier('haarcascade_fullbody.xml')

griton=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
bodies=body_cascade.detectMultiScale(griton,1.8,2)

for (x,y,w,h) in bodies:
    cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,255),3)
cv2.imshow('body',img)
```



ŞEKİL-20: Haarcascade ile vücut tespiti

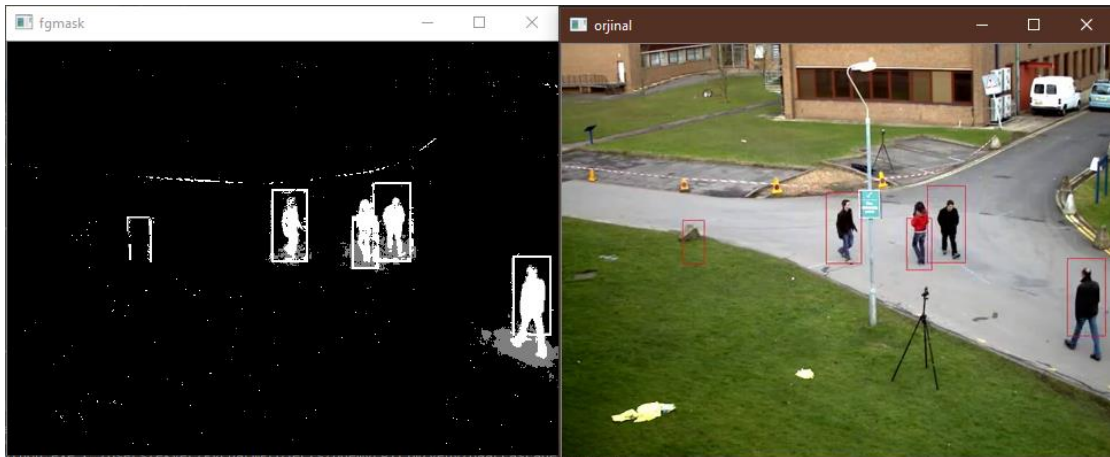
5.3.3 Arka Plan Filtreleme

Arka plandan insanları ayırmak için fark alıcı bir metot kullanıldı.

```
import numpy as np
import cv2

cap=cv2.VideoCapture('video1.avi')
fgbg=cv2.createBackgroundSubtractorMOG2()

while True:
    ret,frame=cap.read()
    fgmask=fgbg.apply(frame)
    cv2.imshow('fgmask',fgmask)
    cv2.imshow('orjinal',frame)
    k=cv2.waitKey(25) & 0xff
    if k==27:
        break
cap.release()
cv2.destroyAllWindows()
```



ŞEKİL-22: Arka plan filtreleme

6) ARA SONUÇLAR VE TARTIŞMA

Sonuç olarak bir binada bulunan kişi sayısını takip etme probleminin giriş ve çıkışlara yerleştirilen kameralar yardımıyla veya güvenlik kameralarıyla çözüme ulaştırabilir olduğu anlaşılmıştır. Bu bilgiyi kullanılabilir hale getirme problemi ve bir bölgenin yoğunluk haritasını oluşturabilme problemi ise projenin geliştirme sürecinde devam etmektedir.

Geliştirilen sistemi değerlendirmek için, internet ortamından elde edilen bir dizi görüntü üzerinde bazı analizler yapıldı. Elde edilen doğruluk sonuçları, istenen son hassasiyete yakın oldukları için bir ilk prototip için çok tatmin edicidir. Ancak sistemin daha önceden tanımlanmış tüm durumlarda doğru bir şekilde sayabilmesi için ele alınması gereken bazı sorunlar olduğundan test süreci devam etmektedir. Çözülmesi gereken ana sorun çeşitli ışık kaynakları ve gölgeler nedeniyle arka plan piksellerinin yanlış algılanabilmesidir. Bir dakikalık test videosunda sistem yüzde 10-12 hata oranları ile çalışmaktadır.

7) BÜTÇE

Pratik tasarım gerçekleştirilmediği için bütçemiz bulunmamaktadır. Yani hiç kaynak harcanmamıştır. Zaten gerekli harcamalar; kamera ve işlemci(bilgisayar) olduğundan ve bu gereksinimler bizim sistemimizin kurulmak istendiği yerlerde halihazırda olduğu için ekstra bir harcama yapılmasına henüz gerek duyulmamıştır.

8) REFERANSLAR

- [1] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi, "A method of counting the passing people by using the stereo images," in Proc. Int. Conf. Image Processing ICIP 99, vol. 2, 1999, pp. 338–342.
- [2] K. Hashimoto, K. Morinaka, N. Yoshiike, and S. Kawaguchi, C.and Matsueda, "People count system using multi-sensing application," in Proc. Int Solid State Sensors and Actuators TRANSDUCERS '97 Chicago.Conf, vol. 2, 1997, pp. 1291– 1294.
- [3] A. G. Vernazza, "Long-memory matching of interacting complex objects from real image sequences," in no book, 1996.
- [4] A. Shio and J. Sklansky, "Segmentation of people in motion," in Proc. IEEE Workshop Visual Motion, 1991, pp. 325–332.
- [5] G. Sexton, X. Zhang, G. Redpath, and D. Greaves, "Advances in automated pedestrian counting," in Proc. European Convention Security and Detection, 1995, pp. 106–110.
- [6] J. Segen, "A camera-based system for tracking people in real time," in Proc. 13th Int Pattern Recognition Conf, vol. 3, 1996, pp. 63–67.
- [7] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2001, vol. 1, 2001.
- [8] G. Conrad and R. Johnsonbaugh, "A real-time people counter," in Proceedings of the 1994 ACM symposium on Applied computing, ser. SAC '94. New York, NY, USA: ACM, 1994, pp. 20–24. [Online]. Available: <http://doi.acm.org/10.1145/326619.326649>
- [9] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in Proc. Seventh IEEE Int Computer Vision Conf. The, vol. 1, 1999, pp. 255–261.
- [10] A statistical approach for real-time robust background subtraction and shadow detection, 1999.
- [11] J.-S. Hu, T.-M. Su, and S.-C. Jeng, "Robust background subtraction with shadow and highlight removal for indoor surveillance," in Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf, 2006, pp. 4545–4550.
- [12] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in Proc. IEEE Conf. Computer Vision and Pattern Recognition

- [13] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," vol. 30, no. 5, pp. 909–926, 2008.
- [14] G. Monteiro, J. Marcos, M. Ribeiro, and J. Batista, "Robust segmentation for outdoor traffic surveillance," in Proc. 15th IEEE Int. Conf. Image Processing ICIP 2008, 2008, pp. 2652–2655.
- [15] Satish D. Pore, B. F. Momin, "Bidirectional People Counting System in Video Surveillance"
- [16] Chen T., Chen C., Wang D., Kuo Y., "A People Counting System Based on Face-Detection"
- [17] Terada K., Yoshida D., Oe S., Yamaguchi J., "A Counting Method of the Number of Passing People Using a Stereo Camera"
- [18] Chang Q., Song Z., Shi R., Xu J., "A People counting method based on multiple cameras information fusion"
- [19] Coskun A., Kara A., Parlaktuna M., Ozkan M., Parlaktuna O., "People Counting System by Using Kinect Sensor"

9) GELİŞME RAPORU ÖZ DEĞERLENDİRME FORMU

<p>1. Literatür taraması ve araştırma tamamen yapıldı mı? Hayır ise nedeni açıklayınız.</p> <p>Literatür taraması kapsamlı bir şekilde yapıldı, kişi sayma yöntemleri üzerine elde edilebilen birçok makaleye ulaşıldı ve incelendi.</p>
<p>2. Yapacağınız çalışmanın daha önce yapılan çalışmalardan farklıları ve yeniliklerini belirledi mi? Açıklayınız.</p> <p>Çalışmamız ve piyasadaki kişi sayma kameraları karşılaştırıldı. Bosch Security tarafından geliştirilen bir kişi sayma uygulaması olarak benzer şekilde kuş bakışı bir pozisyona yerleştirilmiş kamera ile sayma işlemi yapmaktadır. Ayrıca sektörde derinlik algılayan sensörler ile 3D görüntüleme ile kişi sayabilen kameralar bulunmaktadır. Çalışmamız ise kişi sayacı kameralarından elde edilen verinin acil durumlarda kullanılabilecek şekilde raporlanmasını sağlayarak ve sistemin büyük ölçekli kullanım alanlarını araştırarak, bu anlamda ülkemizde ve dünyada yenilikçi bir bakış açısı getirmekte ve büyük ölçekli sistemlerin kullanılmasını teşvik eden ilk çalışmalar olma özelliği göstermektedir.</p>
<p>3. Malzeme alımları gerçekleştirildi mi? Hayır ise nedeni açıklayınız.</p> <p>Güvenlik kamerası bulunan her bina da sistemimiz ek bir gereksinime ihtiyaç olmaksızın verimli bir şekilde çalışmaktadır. Ancak proje sürecinde testlerin daha rahat yapılabilmesi için bir kameraya ihtiyaç duyulması durumunda temin edilmesi planlanmaktadır.</p>
<p>4. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? (Ekonomi, Çevre sorunları, Sürdürülebilirlik, Üretilebilirlik, Etik, Sağlık, Güvenlik, Sosyal ve Politik sorunlar)</p> <p>Projede en önemli kısıt olarak uygulanabilirlik ön plana çıkmaktadır. Uygulanabilirliğin sağlanması ile hedeflenen bölgedeki bütün binalardaki kişi sayısının elde edilmesi ve bölgenin acil durum anlarında tahliye, kurtarma vb. operasyonlarında kullanılmak üzere bu verilerin</p>

korunması ile hem ekonomik açıdan hem de toplumsal açıdan fayda sağlayan bir proje gerçekleştirilmesi hedeflenmiştir. Etik açıdan kişileri kimlikleri göz ardı edilerek yalnızca birey olarak sayımları sağlanmıştır. Sistemin kolayca uygulanabilir olması ekstra maliyet gerektirmemesi ve acil durum anlarında doğru planlamaya yardımcı olması, böylece büyük kazançlar sağlayabilecek olması ekonomik açıdan da somut kısıtlara sahip olduğunu gösterir. Projemizin insan sağlığını tehdit eden bir durum bulunmamaktadır.

5. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir?

IEEE, SONET, ITU standartlarına uygun tasarım yapılmıştır.

6. Proje çalışma takımınız belirtiniz.

Caner Yıldırım

25.09.1995 tarihinde Konak/İzmir’de doğdu. İlk, ortaokul ve lise öğrenimi yine İzmir’de tamamladı. 2014 yılında Ege Üniversitesi Elektrik Elektronik Mühendisliği Bölümü’nde lisans eğitimine başladı. Lisans eğitimini yurtdışı programları, girişimcilik eğitimleri, gönüllü çalışmalar ile geçirdi.

Emir Kaan Yerli

12.02.1997 tarihinde Antalya’da doğdu. İlk, ortaokul ve lise öğrenimi yine Antalya’da tamamladı. 2015 yılında Ege Üniversitesi Elektrik Elektronik Mühendisliği Bölümü’nde lisans eğitimine başladı. Yurtiçi mühendislik yarışmalarında yazılım geliştirici olarak görevler edindi. Kendisi ayrıca lisanlı olarak briç hakemliği yapmaktadır.

7. Proje yönetimi nasıl gerçekleştirdiniz açıklayınız. İş yükü ve zamanı nasıl paylaştınız.

Proje yönetiminde genel olarak agile(çevik) yaklaşım önemsenmiştir. İnsan odağını kaybetmeden, iç ya da dış tüm değişkenlere uyum sağlayıp hedefe uygun, yaratıcı ve basit çözümler üretmeyi esas aldık. Süreç içerisinde en büyük zorluk zamandıydı. Proje çalışanları daha çok ilgi duydukları alanları belirleyerek araştırmalarını gerçekleştirdi. Projenin taslağı oluşturulduktan sonra yetkinlikler doğrultusunda görev dağılımı yapıldı ve mevcut riskler

gözetilerek zaman çizelgesi planlandı. Ekip üyeleri sürekli olarak bilgi paylaşımında bulunarak tutarlı ilerlemeye ayrıca önem verdi.

8. Öneri raporunda yapılan takvime ve/veya konuya göre değişiklik/sapma oldu mu? Evet ise nedeni açıklayınız.

Öneri raporundan yazdığımız takvim ve konu sıralamasına genellikle paralel gidilmiştir.

9. Proje çalışması sırasında ne tür problemler ve aksiliklerle karşılaştınız?

Proje için literatür araştırması yapıldıkça zaman çok çeşitli çözümler yaklaşımlar olduğu görüldü. Bu çeşitlilik içerisinde projede gerekli hedef ve kriterlerin sağlanması adına en uygun çözümler arasında karar vermek bir problem olarak tanımlanabilir. Ayrıca kendi kullandığımız bilgisayarların donanımsal olarak yetersiz olması bizi bir nebze yavaşlatmıştır.

10. Projede ne tür risklerle karşılaştınız ve azaltmak için ne yaptınız?

Proje sürecindeki en büyük risk proje çalışanlarının bir yandan lisans eğitimlerini başarıyla tamamlayabilmesi gerekliliği idi. Bu risk, doğru çalışma takvimi planlanması/uygulanması ile en aza indirgenmiştir.