

ELM463 – DÖNEM PROJESİ

GÖRSELLERDE İNSAN TESPİTİ VE SOSYAL MESAFE UYUMLULUĞU

Ogün Uygur YILDIRIM, Alperen KARATAŞ
1801022091, 1801022022
ouyildirim@gtu.edu.tr, alperen.karatas2018@gtu.edu.tr,

ABSTRACT (ÖZET)

Yapılan projede Python Programlama Diline ait olan OpenCV kütüphanesi kullanılarak pandemi döneminde sosyal mesafe konusu üzerinde çalışılmıştır. Farklı görüntülerde yer alan insanların, sosyal mesafe kuralını ihlal edilip edilmediği kontrol edilmiştir.

ANAHTAR KELİMELER

Sosyal Mesafe, Kontur, Eşikleme, Morfolojik Operatörler, Ayırıt Saptama, Filtreleme, Öznetelik Çıkartımı

1. Giriş

Yapılan projenin genel amacı, çeşitli algoritmalar yardımı ile farklı görsellerdeki insanların tespit edilip sosyal mesafe kuralına uygunluğunu göstermek olmuştur. Proje iki farklı adımla yapılmıştır. İlk adım olarak görüntüdeki insanların tespiti gerçekleştirilmiştir, sonrasında tespit edilen insanlar arasındaki sosyal mesafenin uygunluğu kontrol edilmiştir.

2. Deneyler ve Analiz

Sosyal mesafe analizi için Python Programlama Dili ve **Jupyter Notebook** arayüzü kullanılmıştır. Belirtilen işlemin gerçekleşmesi amacı ile gereken kod satırları yazılarak görüntü dosyaları arayüze yüklenmiştir.

Görüntülerdeki insanları bulmak için görüntü Gauss Filtresi yardımı ile bulanıklaştırılıp gürültüler bastırılmıştır ve kenar bulma algoritmalarından Canny filtresi kullanılarak görüntüdeki kenarlar elde edilmiştir. Canny fonksiyonundaki dx ve dy değerleri görüntü üzerinde deneme-yanılma yolu ile bulunup, en optimize değerler projede kullanılmıştır. Bulunan kenarlar kullanılarak kapama işlemi gerçekleştirilmiştir. Kapama işlemi; genişletme (dilasyon) işlemi uygulanarak birbirine yakın alanlar birleştirilip aradaki gürültülü alanlar kaybolmuştur, fakat yapılan işlem genel alanın büyümesine yol açmıştır. Eski haline getirilmesi için aşındırma (erozyon) işlemi uygulanıp görüntü kenarlardan kırılmıştır. Ortadaki kapanan bölgeler büyük alana birleştiği için tekrar ayrılmamıştır. Kapama işlemi sonrasında elde edilen görüntüye tekrardan genişletme (dilasyon) işlemi uygulanarak kenarlar daha fazla vurgulanmıştır. Elde edilen görüntü Otsu Methodu ile eşiklenmiştir ve bitwise_not fonksiyonu ile görüntünün negatifi alınmıştır.

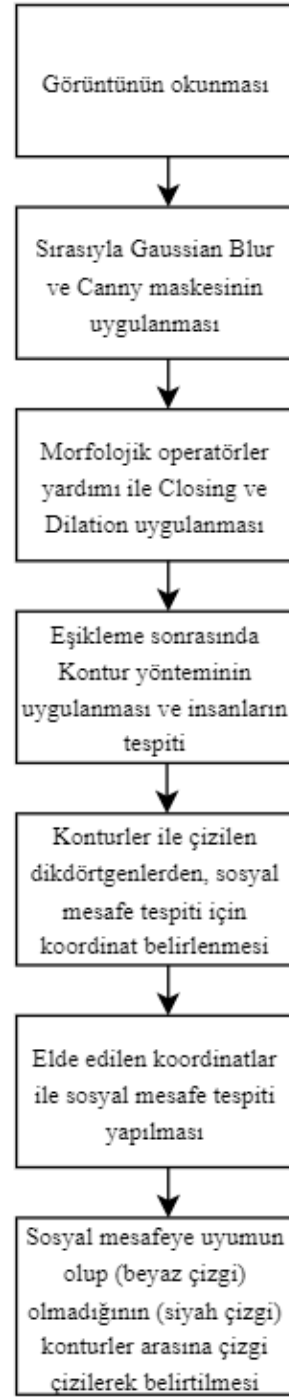
Sonrasında görüntüdeki insanları bulmak için kontur işlemi yapılmıştır. Kontur işlemi ile aynı renk ve yoğunluğa sahip olan tüm kesintisiz noktalar sınır boyunca birleştirilmiştir. Kontur işleminde sadece başlangıç ve bitiş noktalarını bilmek yeterli olunacağı için cv.CHAIN_APPROX_SIMPLE komutu kullanılmıştır. Kontur değeri (contour.area) 1000'den küçük olarak belirtilerek görüntüde sadece insan tespiti yapılmıştır. Bahsedilen değer, görüntüler üzerinde deneme yanılma yolu ile bulunup, en optimize değer kullanılmıştır.

Kontur işlemi sonucunda bulunan insanlar dikdörtgen (box) içerisine alınmıştır. Bu işlemde öncelikle minAreaRect() fonksiyonu ile insanı içine alacak en küçük dikdörtgen bulunmuştur, sonrasında boxPoints() fonksiyonu yardımıyla dikdörtgenin köşeleri elde edilmiştir. Bu köşeler drawContours() fonksiyonunda kullanılarak insanlar görüntü üzerinde dikdörtgen içerisine alınmıştır. Sonrasında dikdörtgenin köşelerine cv.circle() fonksiyonu yardımıyla beyaz renkte daireler eklenmiştir. Sonuç olarak projenin ilk basamağı olan insan tespiti gerçekleştirilmiştir.

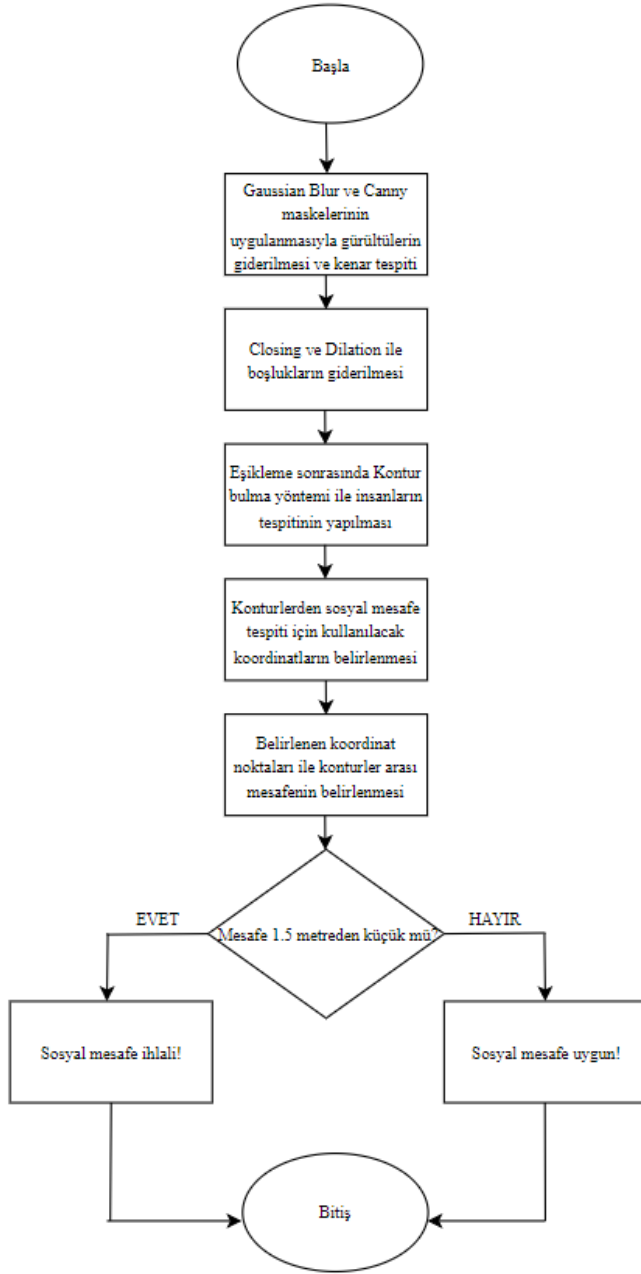
Dikdörtgenlerin köşeleri, koordinat düzleminde noktaları yani koordinatları ifade etmektedir. Görüntülerdeki insanların sosyal mesafelerini bulmak için öncelikle insan tespiti yapan dikdörtgenlerin üst kenarının orta noktasına yine cv.circle() fonksiyonundan faydalanarak siyah daireler eklenmiştir. Sosyal mesafe tespitinde bu siyah daireler kullanılmıştır. Dikdörtgenlerin kenarlarının orta noktaları (siyah daireler) birer listede (list_of_values) tutulmuştur. Bu liste, socialDistance fonksiyonuna gönderilerek her bir noktanın birbirine olan uzaklığı hesaplanmıştır. Bu hesaplama sırasında, aynı değere ve 0 değerine sahip uzaklıklar removeZerosAndSame fonksiyonu ile temizlenerek elde edilen nihai sonuçlar yeni tanımlanan listeye (new_dist_list) atanmıştır. Bu listeden elde edilen değerler sayesinde, görüntüler üzerinde incelemelerde bulunularak sosyal mesafe tespitinin yapılması kolaylaşmıştır.

Her bir dikdörtgenin birbirine olan uzaklığı, bahsi geçen siyah dairelerin x eksenini boyunca birbirlerine olan uzaklığı referans alınarak hesaplanmıştır. Bu referansa göre, uzaklığı 240.0'dan küçük olanlar, yani 1.5 metre sosyal mesafe kuralına uymayanlar sosyal mesafe ihlali olarak tespit edilmiştir. Bu şartı sağlayan uzaklık değerleri ise sosyal mesafenin uygunluğunu ifade etmektedir. Belirlenen sosyal mesafe tespitine göre, konturler (insanlar) arasında cv.line fonksiyonundan faydalanarak x eksenini boyunca bir çizgi çekilmiştir. Bu çizginin siyah olması sosyal mesafe ihlali yapıldığı (mesafe 1.5 metreden küçük), beyaz olması ise sosyal mesafeye uygunluğunu belirtmiştir.

Blok Diyagram:

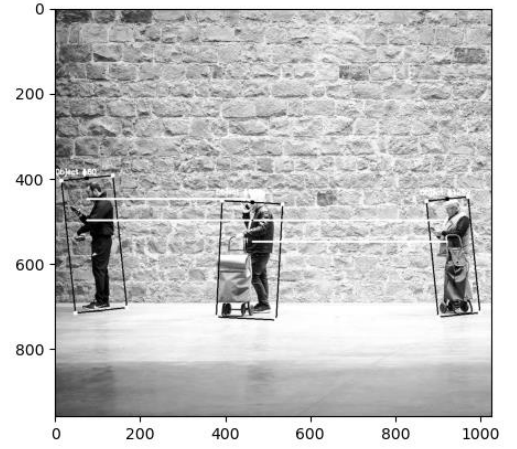


Akış Diyagramı:



2.1. Görseller ve Tablolar

2.1.1. SocialDistance.pgm görüntüsü



Şekil 2.1.1. SocialDistance.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenini boyunca)		
75.5	464.0	922
Tablo 2.1.1		

Dikdörtgenler arası mesafe		
458.0	388.5	846.5
Tablo 2.1.1b		

2.1.2. SocialDistance2.pgm görüntüsü



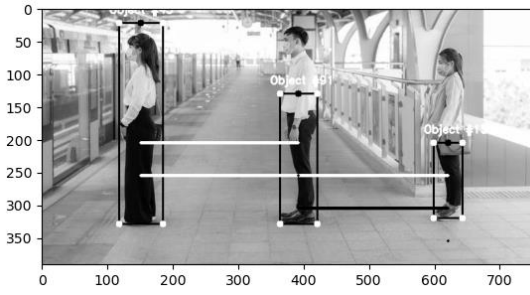
Şekil 2.1.2. SocialDistance.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenı boyunca)	
894.5	2453.0
Tablo 2.1.2	

Dikdörtgenler arası mesafe	
1558.8	
Tablo 2.1.2b	

2.1.3. SocialDistance3.pgm görüntüsü



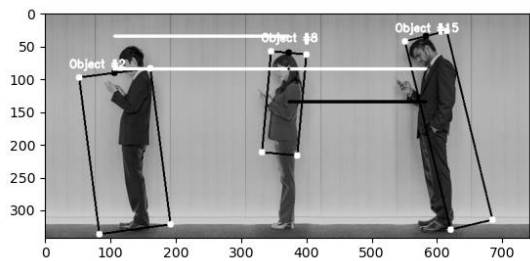
Şekil 2.1.3. SocialDistance.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenı boyunca)		
151.5	392.5	621.0
Tablo 2.1.3		

Dikdörtgenler arası mesafe		
241.0	228.5	469.5
Tablo 2.1.3b		

2.1.4. SocialDistance4.pgm görüntüsü



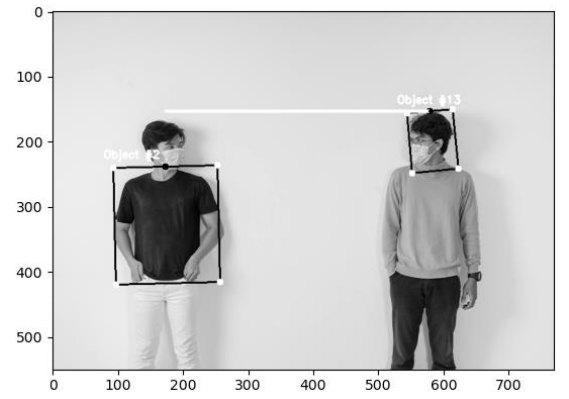
Şekil 2.1.4. SocialDistance.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenı boyunca)		
106.5	373.0	583.0
Tablo 2.1.4		

Dikdörtgenler arası mesafe		
210.0	266.5	476.5
Tablo 2.1.4b		

2.1.5. SocialDistance5.pgm görüntüsü



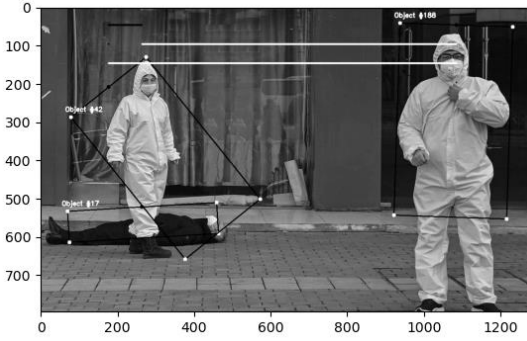
Şekil 2.1.5. SocialDistance.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenı boyunca)	
173.0	579.0
Tablo 2.1.5	

Dikdörtgenler arası mesafe	
406.0	
Tablo 2.1.5b	

2.1.6. SocialDistance6.pgm görüntüsü

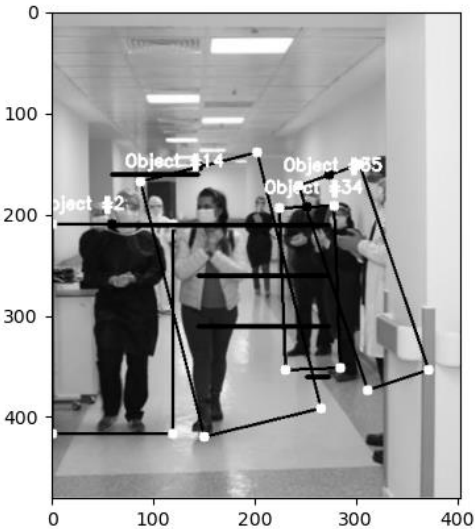


Şekil 2.1.6. SocialDistance6.pgm

Siyah daire koordinatları (soldan sağa, x eksenı boyunca)		
246.5	177.0	1085.0
Tablo 2.1.6		

Dikdörtgenler arası mesafe		
908.0	820.5	87.5
Tablo 2.1.6b		

2.1.7. SocialDistance7.pgm görüntüsü



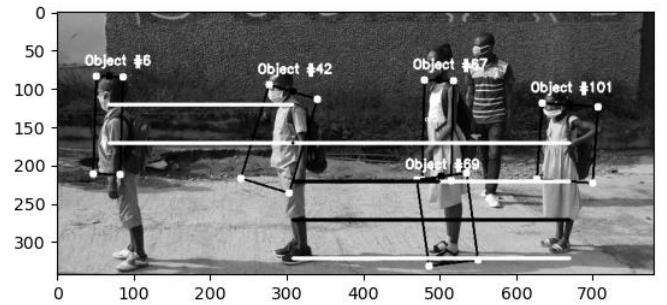
Şekil 2.1.7. SocialDistance7.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenı boyunca)			
59.5	144.5	251.0	273.0
Tablo 2.1.7			

Dikdörtgenler arası mesafe					
128.5	106.5	213.5	85.0	22.0	191.5
Tablo 2.1.7b					

2.1.8. SocialDistance8.pgm görüntüsü



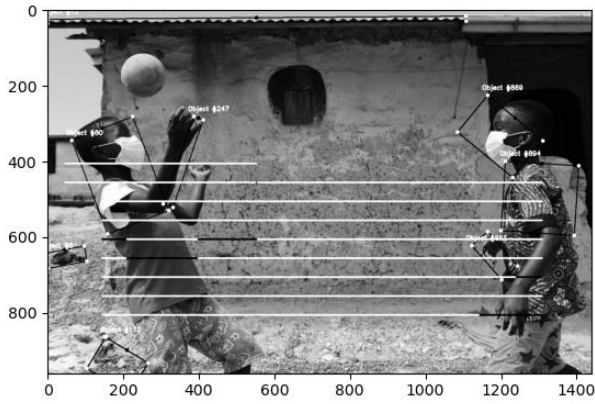
Şekil 2.1.8. SocialDistance8.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenı boyunca)				
68.5	308.5	499.0	502.5	670.5
Tablo 2.1.8				

Dikdörtgenler arası mesafe								
194.0	3.5	168.0	362.0	171.5	430.5	240.0	434.0	602.0
Tablo 2.1.8b								

2.1.9. SocialDistance9.pgm görüntüsü



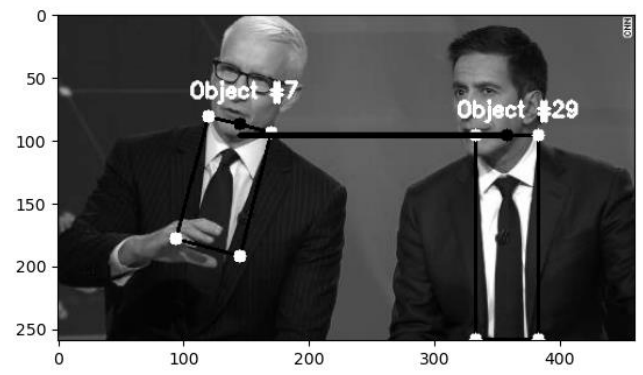
Şekil 2.1.9. SocialDistance9.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenine boyunca)	
210.0	553.5
Tablo 2.1.9	

Dikdörtgenler arası mesafe
343.5
Tablo 2.1.9b

2.1.10. SocialDistance10.pgm



Şekil 2.1.10. SocialDistance10.pgm

Tablo:

Siyah daire koordinatları (soldan sağa, x eksenine boyunca)	
145.0	358.0
Tablo 2.1.10	

Dikdörtgenler arası mesafe
213.0
Tablo 2.1.10b

2.2 Grafik Analizi

2.2.1- SocialDistance.pgm görüntüsü Analizi

Şekil 2.1.1'de yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür. Algoritma, insan tespitini doğru bir şekilde yapmıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olduğunu beyaz çizgi çekerek göstermiştir. Algoritma belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.1b'de yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

2.2.2- SocialDistance2.pgm görüntüsü Analizi

Şekil 2.1.2'de yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür. Algoritma, insan tespitini doğru bir şekilde yapmıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olduğunu beyaz çizgi çekerek göstermiştir. Algoritma belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.2b'de yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

2.2.3- SocialDistance3.pgm görüntüsü Analizi

Şekil 2.1.3'te yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür lakin algoritma insan tespitini tam anlamıyla doğru bir şekilde yapamamıştır çünkü arka plan tam anlamıyla yok edilirken insana ait kenarlar da kısmen yok edilmiştir. Bu problem sosyal mesafenin hesaplanmasında herhangi bir sorun oluşturmamıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olup olmadığını çizgi çekerek göstermiştir. Sosyal mesafe uyumluluğu algoritması belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.3b'de yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

2.2.4- SocialDistance4.pgm görüntüsü Analizi

Şekil 2.1.4'te yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür lakin algoritma insan tespitini tam anlamıyla doğru bir şekilde yapamamıştır çünkü arka plan tam anlamıyla yok edilirken insana ait kenarlar da kısmen yok edilmiştir. Bu problem sosyal mesafenin hesaplanmasında herhangi bir sorun oluşturmamıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olup olmadığını çizgi çekerek göstermiştir. Sosyal mesafe uyumluluğu algoritması belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.4b'de yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

2.2.5- SocialDistance5.pgm görüntüsü Analizi

Şekil 2.1.5'te yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür lakin algoritma insan tespitini tam anlamıyla doğru bir şekilde yapamamıştır çünkü arka plan tam anlamıyla yok edilirken insana ait kenarlar da kısmen yok edilmiştir. Bu problem sosyal mesafenin hesaplanmasında herhangi bir sorun oluşturmamıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olduğunu beyaz çizgi çekerek göstermiştir. Sosyal mesafe uyumluluğu algoritması belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.5b'de yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

2.2.6- SocialDistance6.pgm görüntüsü Analizi

Şekil 2.1.6'da yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür. Algoritma, insan tespitini doğru bir şekilde yapmıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olup olmadığını çizgi çekerek göstermiştir. Algoritma belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.6b'da yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

2.2.7- SocialDistance7.pgm görüntüsü Analizi

Şekil 2.1.7'de yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür. Algoritma, insan tespitini doğru bir şekilde yapmıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olmadığını siyah çizgi çekerek göstermiştir. Algoritma belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.7b'de yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

2.2.8- SocialDistance8.pgm görüntüsü Analizi

Şekil 2.1.8'de yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür lakin algoritma insan tespitini tam anlamıyla doğru bir şekilde yapamamıştır çünkü insan tespiti yapılırken arka plan tam anlamıyla yok edilememiştir. Bu problem sosyal mesafenin hesaplanmasında herhangi bir sorun oluşturmamıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olup olmadığını çizgi çekerek göstermiştir. Sosyal mesafe uyumluluğu algoritması belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.8b'de yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

2.2.9- SocialDistance8.pgm görüntüsü Analizi

Şekil 2.1.9'da yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür lakin algoritma insan tespitini tam anlamıyla doğru bir şekilde yapamamıştır çünkü insan tespiti yapılırken arka plan tam anlamıyla yok edilememiştir. Bu problem sosyal mesafenin hesaplanmasında belirli bir problem oluşturmuştur ve birden fazla çizgi ile sosyal mesafenin kontrolü sağlanmıştır lakin Tablo 2.1.9b'de yer alan sonuçlara da bakıldığında algoritmanın sosyal mesafe uyumluluğu tespitinde başarılı olduğu görülmüştür.

2.2.10- SocialDistance7.pgm görüntüsü Analizi

Şekil 2.1.10'da yer alan görüntüye bakıldığında yazılan algoritmanın belirtilen görüntüde istenildiği şekilde çalıştığı görülmüştür lakin algoritma insan tespitini tam anlamıyla doğru bir şekilde yapamamıştır çünkü arka plan tam anlamıyla yok edilirken insana ait kenarlar da kısmen yok edilmiştir. Bu problem sosyal mesafenin hesaplanmasında herhangi bir sorun oluşturmamıştır ve aralarındaki mesafenin sosyal mesafe kuralına uygun olmadığını siyah çizgi çekerek göstermiştir. Sosyal mesafe uyumluluğu algoritması belirtilen görüntüde olumlu sonuç vermiştir. Tablo 2.1.10b'de yer alan sonuçlara da bakıldığında algoritmanın doğru çalıştığı anlaşılmıştır.

3. Sonuç ve Yorum

Yapılan projede genel olarak, belirlenen görüntülerde yer alan insanın tespiti ve bu insanların sosyal mesafe uyumluluğu kontrol edilmiştir. Proje iki aşamada ilerlemiştir. İlk aşamada insan tespiti yapılırken, ikinci aşamada ise tespit edilen insanların sosyal mesafeye uyumluluğuna bakılmıştır. Bu işlemler yapılırken görüntü işleme algoritmaları hakkında daha fazla fikir sahibi olunmuştur. Farklı görüntülerde en doğru sonucu elde etmek için parametrelerin en optimize değerleri bulunmuştur. Şekil 2.1.1, Şekil 2.1.2, Şekil 2.1.3, Şekil 2.1.4 ve Şekil 2.1.5'te yer alan görüntüler eğitim görüntüleri olarak kullanılmıştır ve algoritma geliştirilmiştir. Diğer görüntüler ise test görüntüleri olarak kullanılıp algoritmanın performansı hakkında fikir sahibi olunmuştur.

Genel olarak algoritmanın performansından bahsedilir ise farklı görüntülerde insan tespitinde küçük hatalar olsa dahi iyi sonuçlar verdiği görülmüştür. Bahsedilen küçük hataların, arka planları tamamen yok edilememesi veya arka plan yok edilirken insan kenarlarının da yok edilmesinden kaynaklandığı anlaşılmıştır. Sosyal mesafe uyumluluğu tespitinde ise algoritma istenildiği şekilde çalışarak doğru sonuçlar vermiştir. Sosyal mesafeye uygun mesafe beyaz çizgi olarak, uygun olmayan mesafe ise siyah çizgi olarak belirtilmiştir.

Projede kenar tespitinde ilk olarak Sobel metodu denenmiştir lakin kenarlar istenildiği şekilde bulunamadığı için Canny metodu kullanılmıştır. Canny metodunun Sobel'e göre daha iyi sonuç verdiği anlaşılmıştır. Canny metodundaki dx ve dy parametreleri artırıldığında istenilen görüntünün tamamen yok olduğu görülmüştür.

Morfolojik operatörler yardımı ile projede görüntüler engellenmiştir ve insan kenarları birbirlerine yaklaşarak tespit edilmesi daha kolay olmuştur.

Projede kontur işlemi sayesinde insanların tespiti gerçekleştirilmiştir ve insanlar dikdörtgen içerisine alınmıştır. Proje sayesinde kontur kullanımı öğrenilip nesnelerin dikdörtgen içerisine alınması hakkında fikir sahibi olunmuştur.

Sonuç olarak projede istenilen sonuçlar elde edilmiştir ve herhangi şaşırtıcı bir problemle karşılaşılmamıştır. Beklenen sonuçlar çıkış görüntülerinde görülmüştür. Proje sayesinde nesne tespiti hakkında fikir sahibi olunup gelecekte görüntü işleme hakkında daha fazla bilgi sahibi olmak hedeflenmiştir.

4. Kaynaklar

[1] GTÜ, ELM463 ders materyalleri

[2]<https://www.pyimagesearch.com/2016/04/04/measuring-distance-between-objects-in-an-image-with-opencv/>

[3]https://www.pyimagesearch.com/2016/03/21/ordering-coordinates-clockwise-with-python-and-opencv/?_ga=2.198775343.1471408995.1639594310-1118873955.1638619128

[4] <https://theailearner.com/tag/cv2-minarearect/>

[5]https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html

[6]https://docs.opencv.org/4.x/dc/da5/tutorial_py_drawing_functions.html

[7]https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

[8]<https://www.geeksforgeeks.org/python-opencv-cv2-puttext-method/>