ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM 4062 PROJE RAPORU

Unmanned Aerial Vehicle Detection and Route Estimation

ERENCAN TEKİN

19290273

DOÇ. DR. BÜLENT TUĞRUL

Mayıs, 2024

ÖZET

Bu projede, insansız hava araçlarının gerçek zamanlı tespiti ve gelecekteki konumunun, yönünün tahmini için bir sistem geliştirilmiştir. Sistem, yüksek doğruluk ve güvenilirlikle insansız hava aracı tespiti yapmak ve gelecekteki konumunu, yönünü tahmin etmek için tasarlanmıştır. Sistemin temel bileşenleri arasında görüntü işleme, görüntü algılama/tespiti, Gaussian filtresi, Kalman filtresi kullanarak görüntü işlemeyle tespit edilen insansız hava araçlarının gelecekteki konumu ve yön tahmini yer almaktadır. Görüntü işleme sayesin görüntü yakalanır, ön işlemlerden geçirilerek insansız hava aracı tespiti yapılır, daha sonra Gaussian filtresiyle yakalanan insansız hava aracı görüntüsünün kenarları belirginleştirilerek tespiti kolaylaştırılır. Daha sonra hız ve mesafe hesaplamaları yapıldıktan sonra hedefin, yani insansız hava aracı, merkezi belirlenir ve hedef takibi başlatılır. En sonda ise insansız hava aracının gelecekteki konumu ve yönü tahmin edilir. Bu tahminler ile gerçek değerler karşılaştırılarak doğruluk hesabı yapılır.

Geliştirilen bu sistem gerçek zamanlı olarak algılanan ve izlenen insansız hava aracının konumunu, hızını ve kameraya olan mesafesini gerçek zamanlı olarak video üzerinde gösterir. Bu sayede, kötü niyetli insansız hava araçlarına karşı gerekli önlemler alınarak hem sivil hem de askeri alanda hava sahasının güvenliği sağlanmış olur. Güvenlik ve gözetim uygulamalarında kritik öneme sahip bir sistemdir.

İÇİNDEKİLER

1. GİRİŞ

İnsansız hava araçları, son yıllarda hızla gelişerek hem sivil hem de askeri alanlarda büyük etkiler yaratarak kendilerinden bahsettirdiler. Bu insansız hava araçları (İHA), gözetim, keşif, nakliyat gibi çeşitli amaçlarla kullanılmaktadırlar. Ancak, bu insansız hava araçlarının kullanımının yaygınlaşması özellikle güvenlik ve gizlilik açısından bazı önemli zorlukları da beraberinde getirmiştir. Yasak bölgelere izinsiz girişler, hassas alanlarda casusluk faaliyetleri ve hava sahasında yaşanan kazalar insansız hava araçlarının tespiti ve gelecekteki konum ve yönlerinin tahminine olan ihtiyacı arttırmaktadır.

Projemi esnek ve oldukça elverişli olan Python programlama dili ve OpenCV ve Kalman filtreleme kütüphanesi kullanılarak geliştirdim. Görüntüdeki insansız hava araçlarını algılamak için önceden eğitilmiş verilere (pretrained data) sahip olan YOLO'yu kullandım. Ayrıca, görüntü işleme teknikleriyle arka plan çıkarımı ve maskeleme gibi işlemler uyguladım. Bu sayede elde edilen görüntünün tespitinin daha güvenilir olmasını sağladım.

Bu çalışmada, öncelikle kullanıldığım algoritmalar ve yöntemler detaylı bir şekilde açıklanacaktır. Ardından projede kullandığım sistem tasarımını anlatılacaktır. İnsansız hava araçları tespiti ve gelecekteki konum ve yönlerinin nasıl tespit ve tahmin edileceği ve bu tahminin ne kadar doğru olduğu detaylı bir şekilde anlatılacaktır.

1.1. Proje Amacı ve Kapsamı

Bu projenin amacı, gelişmiş görüntü işleme teknikleri ve Kalman filtresi kullanarak insansız hava aracı tespiti ve bu hava araçlarının gelecekteki konum ve yönlerinin daha pratik ve yenilikçi bir sistem kullanılarak tahmininin gerçekleştirilmesidir. Proje kapsamında, gerçek zamanlı video verilerini analiz ederek insansız hava araçlarını algılayabilen ve hareketlerinden bir sonraki konum ve yönlerinin tahminini yüksek doğrulukta tespit edebildiğimiz bir algoritma tasarladım. Bu sistem, özellikle güvenlik ve savunma uygulamaları için yüksek doğruluk oranına sahip bir çözüm sunmayı

hedeflemektedir. Proje, güvenlik açıklarını kapatmada ve izinsiz, kötü niyetli insansız hava araçlarının girişlerini engellemede etkili bir araç olmasını hedeflenmektedir.

1.2. Havacılık ve Savunma Sektöründe (İHA) Tespiti ve İzlenilebilirliği

Son yıllarda insansız hava araçlarındaki (İHA) hızlı gelişme, havacılık ve savunam sektörlerinde büyük bir etki yaratmıştır. İnsansız hava araçları, gözetim, keşif, haritalama alanları önde gelmek üzere birçok alanda etkin bir şekilde kullanılmaktadır. Bu gelişmeler maalesef hava sahasında dolaşan izinsiz, kötü niyetli insansız hava araçlarının dolaşmasına da sebep olmuştur. Bu durum hassas bölgeler üzerinde casusluğa sebep olabilir ve askeri operasyonları tehlikeye atabilir. Bu nedenle, havacılık ve savunma sektöründe insansız hava aracı tespiti ve izlenmesi, kritik bir öneme sahip olmuştur. Bu proje de bu soruna çözüm oluşturmak için yapılmıştır.

2. PROJEYE HAZIRLIK

2.1. Python Programlama Dili

Kodumu yazarken Python programlama dilini tercih etmemin nedenlerinden biri geniş bir kütüphaneye sahip olmasıdır. Özellikle görüntü işleme ve görüntü manipülasyonları işleminde kullanılabilecek OpenCV gibi kütüphanelerinin de olması Python'ı bu proje için oldukça uygun bir programlama dili yaptı.

Kolay bir sözdizimine (" syntax ") sahip olması da komplex algoritmaları yazarken oldukça pratiklik ve kolaylık sağladı. Dinamik değişkenlere izin vermesi Python'a esneklik kazandıran bir diğer özellik oldu. Ayrıca açık ve sade sözdizimi (" syntax ") yapısı sayesinde okunabilirliği arttırmakta ve daha sonraki yapılacak olan modifikasyon işlemleri için daha rahat bir yapıda olmasına olanak sağladı.

Python'ı seçmemdeki bir diğer neden ise oldukça yardımsever bir topluluğa sahip olması, geniş dokümantasyonu olması ve bu sayede herhangi bir sorun ile karşılaştığımda kolaylıkla bu sorunu halledebildim.

Bir başka özelliği ise interpreter (" yorumlayıcı ") bir dildir. Kod zaman kaybı olmaksızın anında çalıştırılır ve bu sayede hatalar daha hızlı bir şekilde tespit edilebilir. Kodlar platformdan bağımsız olarak rahatlıkla yürütülebilir.

Python'un bu etkili ve güçlü yönlerinden dolayı, en başarılı ve uygun bir şekilde insansız hava aracı tespiti ve bir sonraki konum, yön tahmini yapabilmek için en uygun programlama dili olduğunu düşündüm.

2.2. Görüntü İşleme

Görüntü işleme, dijital görüntülerin analiz edilerek işlenmesi ve anlamlandırılması işlemidir. Bu işlem, bilgisayarla görme tekniklerini kullanarak, görüntülerden bilgiyi çıkartarak görüntü verilini daha anlamlı hale getirmeyi amaçlar. Görüntü işleme uygulamaları çeşitli şekillerde kullanılmaktadır. Bu nedenle birçok endüstride önemli bir rol oynamaktadır. Örnek olarak güvenlik ve robotik verilebilir.

Nesne tanıma (“ Object Detection “), bir görüntü içindeki belirli nesnelerin tanınması ve sınıflandırılmasıdır. Hareket algılama (“ Motion Detection “), bir nesnenin veya sahnenin hareketini tespit etme ve izleme işlemidir. Nesne takibi (“ Object Tracing “), bir nesnenin hareketini takip etme ve belirli bir süre boyunca izleme işlemidir. Görüntü iyileştirme ve filtreleme (“ Image Enhancement and Filtering “), bir görüntüdeki gürültüyü azaltma ve görüntü kalitesini artırma işlemlerini içeren işlemler topluluğudur.

Görüntü işleme genellikle filtreleme, kenar tespiti, dönüşümler ve desen tanıma gibi yöntemleri içermektedir. Filtreleme, gürültüyü azaltma veya belirli özellikleri ön plana çıkarmak amacıyla uygulanır. Kenar tespiti görüntüdeki nesnelerin sınırlarını belirlemek için kullanılır. Dönüşümler ise görüntülerin boyutunu değiştirmek veya belirli bir açıyla döndürmek gibi işlemleri içerisinde barındırır. Desen tanıma ise görüntülerdeki belirli şekilleri veya nesneleri tanımlama sürecine verilen addır. Projemde gürültüyü azaltmak için Gaussian filtresini, daha iyi tespitlerde bulunmak için kenar tespiti özelliklerinden yararlandım. Aynı zamanda nesne tanımlama ile insansız hava aracı tespitinde bulundum.

2.3. YOLO

YOLO (You Only Look Once), nesne tespiti için kullanılan bir derin öğrenme algoritmasıdır. Bu algoritma, bir görüntüyü tek bir geçişte inceleyerek nesnelerin konumlarını ve sınıflarını belirler. Geleneksel nesne tespiti yöntemlerine kıyasla daha hızlıdır, çünkü bir görüntüyü sadece bir kez işler, bu da hız kazandırır. YOLO’nun temel mantığı, görüntünün tümünü tek bir CNN (“ Convolutional Neural Network ”) geçişinde işleyerek nesne tespiti yapmaktır. Bu, bir ağım tam bir görüntüyü bir kerede analiz etmesine olanak tanır. Bu da görüntünün hızlıca yakalanmasını ve işlenmesine olanak sağlamaktadır. YOLO’nun avantajlarını söylemek gerekirse, hızlı tespit yapabilmesi, yüksek doğruluk oranına sahip olması, nesne tespiti ve takibi gibi alanlarda yüksek başarılar tespit edebilmesi söylenebilir. Bu özelliklerinden dolayı projemde insansız hava araçları tespitinde YOLO ‘yu kullandım. Bu sayede video akışı başlatıldığında zaman kaybı olmaksızın insansız hava aracı tespitlerini başarıyla yapabildim.

2.4. Normalizasyon

Normalizasyon, verilen belirli bir aralık veya standart bir dağılım içinde olmasını sağlayan bir veri işlemesidir. Veri normalizasyonu, veri setindeki değerleri ölçeklendirerek, farklı özellikler arasındaki ilişkilerin daha doğru bir şekilde analiz edilmesini sağlar. Normalizasyon işlemi, veri setindeki değerleri genellikle [0, 1] aralığında veya belirli bir aralıkta sınırlar. Böylece veri setindeki değerlerin ölçeklerine göre yanıltıcı sonuçlar üretmesi önlenmiş olur.

3. PROJE ÇALIŞMALARI

3.1. Gerekli Kütüphane ve Modüllerin Seçimi ve Kullanım Alanları

Görüntü işlemede (“ Computer Vision “) önemli olan görüntünün analiz edilerek üzerlerinde işlem yapılabilirliğini sağlamaktır. Görüntünün yakalama da ve yakalanan görüntü üzerinde çeşitli işlemler uygulamaya elverişli olması ve oldukça esnek bir programlama dili olduğu için Python programlama dilini seçtim. Bu esnekliği ve sağladığı kolaylıkları en doğru bir şekilde kullanmak doğru kütüphaneleri seçmekle olur. Kütüphane (“ Library “), bir alanda özelleşmiş, belirli bir görevi yerine getirmek için bir araya gelen kod parçalarının birleşmesiyle oluşan paketler (“ modül “) demektir.

3.1.1. OpenCV (Açık Kaynak Bilgisayar Görüntüsü Kütüphanesi)

Görüntü işleme ve bilgisayarla görme alanında kullanılan açık kaynaklı bir kütüphanedir. Çok geniş bir fonksiyon yelpazesine sahiptir. Bu kütüphane, nesne tanıma, hareket takibi, görüntü filtrasyonu gibi çeşitli işlemleri gerçekleştirmek için optimize edilmiş elverişli bir kütüphanedir.

Sunduğu bu geniş işlevsellik ve esneklikten dolayı, insansız hava araçlarını yakalamada nesne tespiti özellikleri sayesinde projemde kullandım.

3.1.2. NumPy

NumPy, diğer adıyla “ Numerical Python “, bilimsel hesaplamalar için kullanılan açık kaynaklı bir Python kütüphanesidir. Çok boyutlu dizi (array) nesnelerini ve bu diziler üzerinde yüksek performanslı matematiksel işlemler yapmayı sağlayan fonksiyonları yapısında barındırır. Özellikle büyük veri setleri ve matris işlemleri ile çalışırken etkinliği arttırır ve diğer bilimsel kütüphaneler için temel yapı taşı olarak hizmet eder. Python’un standart listelerinden daha hızlı ve verimli işlemleri sunar ve bilimsel hesaplamalar, veri analizi gibi alanlarda yaygın olarak kullanılır.

Projemde video akışı başlatıldığı sırada çektiğim görüntü verilerinden maskeleme işlemleri sonucunda elde ettiğim büyük dizileri kolaylıkla modifiye etmemi sağladığı için NumPy kütüphanesini kullanmayı tercih ettim.

3.1.3. PIL

PIL (Python Imaging Library), Python için popüler bir görüntü işleme kütüphanesidir. Görüntü dosyalarını açmak, işlemek ve kaydetmek için geniş bir işlevsel yelpazeye sahiptir. Kütüphanenin önemli bir parçası olan “ Image “ modülü görüntüleri açmak, oluşturmak, dönüştürmek gibi çeşitli adımları optimize bir şekilde gerçekleştirmek için kullanılır.

“ Image “ modülünü, görüntü işleme görevlerini Python ile kolay ve etkili bir şekilde gerçekleştirmek için güçlü ve etkili araçlar sağladığı için projemde kullanmayı tercih ettim.

3.1.4. Torch

Torch, PyTorch olarak bilinen popüler bir açık kaynaklı makine öğrenimi kütüphanesinin temelini oluşturan bir derin öğrenme kütüphanesidir. Derin öğrenme modellerinin geliştirilmesi, eğitimi ve dağıtımı için optimize edilmiştir.

Esnekliği ve Python ile uyumu sayesinde karmaşık modellerin kolayca oluşturulup, modifiye ve test edilebilinmesine olanak sağlar.

Bir diğer özelliği de YOLO veri setini ağırlıklarıyla birlikte yüklemeyi sağlar.

3.1.5. Math

Math kütüphanesi, Python’un standart kütüphanelerinden biridir ve temel matematiksel işlemleri gerçekleştirmek için kullanılır. Trigonometrik fonksiyonlar, üstel fonksiyonlar ve logaritmik fonksiyonlar gibi çok sayıda oldukça karmaşık işlemleri yerine getirebilecek matematiksel işlemleri barındırmaktadır. İşlemlerin hızlı ve verimli bir şekilde gerçekleşmesine olanak sağlar. Matematik kütüphanesi, bilimsel hesaplamalar, mühendislik uygulamaları ve istatistiksel analizler gibi birçok matematiksel işlemlerin gerçekleşmesinde yaygın olarak kullanılır.

3.1.6. Kalman Filtresi

Kalman filtresi, sistem durumunu ve bu durumun tahminlerini sürekli olarak güncelleyerek, sistemin gelecek durumlarını en iyi ve optimize bir şekilde tahmin

etmeye çalışan bir filtreleme türüdür. Bu filtre, özellikle sensör verilerinin gürültülü olduğu ve sistem dinamiğinin karmaşık olduğu durumlarda etkin sonuçlar ortaya koyabilmektedir.

Kalman filtresi, iki temel aşamadan oluşur: tahmin ve güncelleme. Tahmin aşamasında, sistem koşulları kullanılarak gelecekteki durumu tahmin edilir. Güncelleme aşamasında da tahmin edilen durum gerçek değerlerle karşılaştırılarak güncellenir. Bu sayede daha iyi sonuçlar ortaya konur.

Python'da, Kalman filtresini kullanarak gerçek zamanlı nesne izleme, hata düzeltme, gelecek konum, yön tahmini uygulamalar yapılabilir. Bu projede de bu özelliklerinden yararlanarak insansız hava aracı tespiti ve sonraki konum, yön tahminini yaptım.

3.1.7. Gaussian Filtresi

Gaussian filtesi, görüntü işleme alanında sıklıkla kullanılan bir filtreleme tekniğidir. Temel olarak, bir görüntü üzerindeki gürültüyü azaltmak ve görüntüyü daha yumuşak bir hale getirmek için kullanılır. Gaussian filtresi, Gauss dağılımı olarak bilinen bir olasılık dağılım sistemi ile uygulanır.

Bu filtre, bir pikselin değerini, ona yakın olan diğer piksellerin değerleriyle ağırlıklı bir şekilde ortalamak suretiyle belirler. Ağırlıklar, piksellerin birbirlerine olan uzaklıklarına göre Gauss fonksiyonu tarafından belirlenir. Bu yöntem sayesinde merkez pikselin etrafındaki piksellerin katkısı merkeze daha yakın olan piksellerden daha fazla olur.

Gaussian filtresi, yakalanan görüntü üzerindeki keskin kenarları yumuşatırken gürültüyü azaltır. Aynı zamanda bu filtre görüntüdeki detayların korunmasına yardımcı olur. Görüntüde bulunan gereksiz detaylar azaltılarak görüntü işleme algoritmalarının daha hızlı ve daha doğru sonuçlar elde etmesini sağlar. Daha pürüzsüz, doğru ve etkili sonuçlar almak için projemde Gaussian filtresini kullanmayı tercih ettim.

Aynı zamanda projemi yaparken görüntülerdeki insansız hava araçlarının her bir sonraki döngüde daha da net bir şekilde tespit edilebilip sonraki aşamada daha sağlıklı bir Kalman filtresiyle insansız hava araçlarının gelecekteki yön ve konumunu tahmin edebilmem için Gaussian filtresine ihtiyaç duydum.

```
gaussian_filter( girisDizisi, standartSapma, mode)
```

Şekil 1 Gaussian filter fonksiyonu

3.2. Görüntü Yakalama ve Ön İşleme

Görüntünün yakalanması projenin en önemli aşamalarından biridir. Görüntünün yakalanması Python programlama dilinin sağladığı önemli kütüphanelerden biri olan OpenCV kütüphanesini kullanarak video akışını yakaladım. Aşağıda Şekil 1'de görüntüyü yakalamak için kullandığım kod verilmiştir.

```
import cv2 as cv
cap = cv.VideoCapture("UnmannedAerialVehicle.mp4")
```

Şekil 2 OpenCV kütüphanesi kullanarak video akışı yakalama

Görüntüyü aldıktan sonra birtakım ön işlemlerden geçiririm. Örnek verilmek gerekilirse, arka plan çıkarımı, erozyon, medyan bulanıklaştırma işlemi ve genişletme işlemi verilebilir. Bu özellik sahnede hareket halindeki, değişen nesneleri tespit etmek için arka planı kaldırır. Aşağıda Şekil 2'de arka plan çıkarıcı prototipi görülmektedir.

```
backgroundSubtractor = cv.createBackgroundSubtractorMOG2(history, threshold, shadows)
backgroundS = backgroundSubtractor.apply(frame)
```

Şekil 3 Arka plan çıkarıcı

Burada " createBackgroundSubtractorMOG2() " bu fonksiyon MOG2 (" Gaussian Mixture-based Background / Foreground Segmentation) algoritmasını kullanan bir arka plan çıkarıcı nesnesi oluşturur.

Şekil 2'de gösterilen arka plan çıkarıcının parametrelerini incelemek gerekirse;

1. History

Arka plan modellemesi için kullanılacak önceki kare sayısıdır.

2. Threshold

Bir pikselin arka plan ait bir piksel olup olmadığını belirlemek için belirlenen eşik değerini belirtir. Ne kadar düşükse o kadar hassas bir algılama yapılır.

3. Shadows

Gölge piksellerini algılamak için kullanılan bir özelliktir. Eğer “ True “ olarak belirlenirse, algoritma gölge piksellerini tespit etmeye çalışır. “ False “ olarak belirlenmişse, gölge pikselleri tespit edilmeden devam edilir.

“ BackgroundSubtractor.apply(frame) “ kodu ile belirtilen frame üzerinde arka plan çıkarımı uygular. Başka bir deyişle, arka plana ait olmayan pikseller çıkartılır ve sadece ön plandaki nesneler bulunur.

Bir sonraki uyguladığım ön işlem ise erozyon işlemidir. Bu işlem ile görüntü üzerindeki küçük ve gereksiz noktaları kaldırarak gürültüyü azaltmış oluyorum.

```
cv.erode( girdiGörüntüsü, çekirdekMatris, iterasyon )
```

Şekil 4 Erozyon işlemi

Ayrıca, daha iyi bir görüntü için medyan bulanıklığı uyguluyorum. Bunu uygulamamdaki amaç da kalan gürültüyü daha da azaltarak minimuma indirmektir.

```
cv.medianBlur(girdiGörüntüsü, çekirdekMatrisi)
```

Şekil 5 Medyan bulanıklaştırma işlemi

Ayrıca “ cv.dilate() “ fonksiyonu sayesinde görüntüdeki tespit edilen nesnelerin kenarları genişletilerek büyütülür. Bu işlem, nesne tespiti ve nesne kenarlarını iyileştirmek için kullanılır.

```
cv.dilate(girdiGörüntüsü, çekirdekMatris, iterasyon)
```

Şekil 6 Genişletme İşlemi

Bu aşama görüntüyü sonraki aşamalarda kullanacağım hız ve mesafe hesabı, nesne takibi ve Kalman filtresi gibi adımlar için hazır duruma getirmektedir. Bu aşama sayesinde görüntüdeki gürültü büyük oranda azalır ve yüksek başarı oranlarıyla (projemde %85 ila % 95 arası) sonuçlar bulmamı sağladı.

3.3. Hız ve Mesafe Hesaplaması

Görüntü işlemede nesnelerin hızı ve kameraya olan mesafelerinin hesaplanması oldukça önemlidir. Özellikle, nesnelerin hızı, izlenen nesnenin hareket durumu hakkında değerli bir bilgi kaynağı sağlar. Belirli bir noktaya ne kadar sürede varabileceği hakkında bilgi verir. Örneğin bir hava sahasına izinsiz girmiş bir insansız hava aracının hangi hızda hangi yöne doğru yöneldiğini ve daha sonra hangi yöne doğru yöneleceğini bilmek büyük öneme sahiptir.

Aynı zamanda kameraya olan mesafesi de en az hız tespiti kadar hayati öneme sahiptir. Mesafe bilgisi, nesnelerin konumunun doğru bir şekilde belirlenmesinde önemli rol oynar. Ayrıca, güvenlik sistemlerinde tehdit tespiti ve acil durum müdahalelerinde hızlı tepki vermede büyük pay sahibidir.

Projemde hız hesabını yaparken belirli bir zaman aralığında, tespit ettiğim insansız hava aracının değiştirdiği mesafeyi hesapladım. Daha sonra yer değişimi miktarını zamana bölerek hızı bulmuş oldum. Ancak bulduğum bu değerleri bir listede saklıyor ve ortalama değerlerini alarak daha stabil ve doğru bir sonuç elde ediyorum.

Mesafeyi ise bulduğum gerçek genişlik ve hesapladığım odak uzaklığı çarpıyor ve tespit ettiğim insansız hava aracının genişliğine bölerek buluyorum.

3.4. Nesne Takibi

Nesne takibi, bir nesnenin video akışı boyunca tespit edilip konumunun sürekli olarak izlenmesi işlemine verilen addır. Bu işlem nesnenin her karede algılanmasını,

konumunun tespit edilmesini ve hareketinin analiz edilmesiyle gerçekleşen bir işlemdir.

Projemde nesne takibini ilk olarak her görüntünün her karesinde insansız hava aracının konumunu tespit ediyorum. Daha sonra tespit ettiğim bu insansız hava aracının merkezini buluyor ve ona odaklanıyorum. Bu sayede insansız hava aracı gerçek zamanlı olarak hareket ederken ben her karede aslında onun merkezi noktasına odaklı bir şekilde onu takip ediyorum.

3.5. Kalman Filtresi Kullanarak Tahminde Bulunma

Kalman filtresi, gerçek zamanlı olarak değişen sistem ve nesnelerin durumunu tahmin etmek ve ölçün hatalarını azaltmak için kullanılan bir algoritmadır. Bu filtre, doğrusal dinamik sistemlerde, geçmiş ölçümlerden yararlanarak gelecekteki durumu tahmin etmeyi sağlar. Kalman filtresinin gürültülü ve belirsiz ortamlarda bile yüksek doğruluk oranlarına ulaşabildiği için projemde kullanmayı seçtim ve başarı oranının yaklaşık olarak %85 ila %95 arasında değiştiğini gördüm.

Projemde ilk olarak başlangıç durumunu ve önceden tanımlamış olduğum gürültü kovaryans matrislerini kullanarak bir sonraki tahmini konumu hesapladım. Daha sonra ise her döngü de yeni ölçümlerden ve kovaryans matrislerinden yararlanarak tahmin edilen durumu düzelttim. Bunu başarı oranının zaman içerisinde artmasıyla da gördüm.

```
KalmanFilter().predict(xCoord, yCoord)
```

Şekil 7 Kalman filtresiyle sonraki konum tahmini

```
KalmanFilter().kal( oncekiTahmin, belirsizlikKovaryansMatrisi, kontrolMatrisi, kontrolVektoru, gozlemVektoru )
```

Şekil 8 Kalman filtresinin bir sonraki durumunu güncellemesi

Şekil 8 parametreleri;

1. Önceki Durumun Tahmini
2. Önceki Durum Tahmininin belirsizlik kovaryans matrisi

3. Durum denklemindeki kontrol matrisi
4. Sonradan gelen bir kontrol girdisinin filtreleme işlemine etkisini belirten parametre
5. Gerçek ölçümlerin filtreleme işlemine entegrasyonu, gerçek ölçümlerin olmadığı durumlarda sonraki durum tahmin edilir

3.6. Başarı Oranı Hesaplaması

Projemde oluşturduğum sistemin doğruluk oranını, tahmin ettiğim ve gerçek değerlerin Öklidyen mesafelerinden yararlanarak buldum. Temel olarak anlatmak gerekilirse, tahmin ettiğim ve gerçekte olan konum koordinatlarını (x, y) bir listede sakladım. Daha sonra bu listedeki her tahmini değeri karşılığı olan gerçek değerle karşılaştırdım ve ikisi arasındaki Öklidyen mesafelerini hesaplayarak toplam mesafeyi belirledim ve son olarak bulduğum bu mesafelerin ortalamasını alarak stabil ve doğru bir sonuç elde etmeye çalıştım.

4. SONUÇ

Hızla gelişen teknolojik dünyada adıyla kendinden bahsettiren görüntü işleme diğer adıyla “ Computer Vision “, görüntünün analizi ve istenilen çıktı için verileri çeşitli işlemlere tabi tutabilen gelişmiş bir sistematik teknolojidir. Aynı zamanda gerçek zamanlı olarak nesnelerin şu an ki konumlarından gelecekteki bir sonraki konumlarını ve yönlerini yüksek doğruluk oranlarıyla tahminde bulunabilecek kadar gelişmiş algoritmaları içinde barındıran bir sistemdir.

Hızla gelişen bu teknoloji birçok yeniliği beraberinde getirirken aynı zamanda beraberinde güvenlik açıklarını da beraberinde getirmiştir. Örneğin, yüksek öncelikli hava sahalarına giren kötü niyetli insansız hava araçları bilgi açığına ve hatta geri dönülemez hasarlar meydana getirebilir. Özellikle askeri alanda böyle bir durum hem sivil hem askeri alanda çok büyük bir tehdit barındırmaktadır. Bu proje de bu güvenlik açıklarının ve tehditlerin minimize edebilmeyi ve mümkünse ortadan kaldırmayı hedefledim.

Projem, insansız hava araçlarının tespitine ve şu an bulundukları konumlar göz önünde bulundurularak gelecekte gidebilecekleri konum ve yönlerin tahminine dayanmaktadır. Bunun için öncelikle bir görüntü girişi alınır. Bu görüntü girişi sağlıklı bir şekilde alındıysa birtakım ön işlemlerden geçirilir. Bu ön işlemlerin amacı görüntüyü gereksiz arka plan gürültülerinden kurtarmak ve görüntü içerisindeki insansız hava aracı tespitini başarıyla yapmaktır. Daha sonra tespit başlatıldığında görüntüde insansız hava aracı varsa başarılı bir şekilde onu tespit eder, hız ve kameraya olan mesafesini hesaplar. En sonunda ise nesne takibi ve Kalman filtresi uygulayarak bir sonraki konum ve yönünü bularak hedefime ulaşmış oluyorum.

Bu projenin en büyük avantajlarından biri bilgi casusluğunu önlemesidir. Çünkü bu proje sayesinde hava sahasına izinsiz giren kötü niyetli insansız hava araçları en kısa zamanda etkisiz hale getirilebilir. Aynı zamanda sivil ve askeri güvenliği sağlar. Bu gibi güvenlik açıklarının yanı sıra keşif alanlarında da kullanılabilen çok avantajlı bir sistemdir. Her alanda da Kalman filtresiyle insansız hava araçlarının gelecekteki sonraki konum ve yönlerinin tespiti daha güvenilir ve etkili bir şekilde sağlanmış olmaktadır.

Sonuç olarak görüntü işleme oldukça gelişmiş bir teknolojidir. Bu gelişim sürecinde birçok yeniliğin yanı sıra güvenlik açıkları gibi tehditleri de beraberinde gelmiştir. Projem, Kalman filtresini oldukça verimli kullanarak bu tehditleri ortadan kaldırmak, daha elverişli ve güvenli bir ortam sağlamak için çok iyi bir çözüm sunmaktadır. Projem, ayrıca hedefin hız ve kameraya olan mesafesi gibi özelliklerini de hesaplayabilen optimal bir sistemdir. En önemlisi de bunların hepsinin optimal bir şekilde gerçek zamanlı olarak olabilmesidir. Bu nedenlerden dolayı sivil ve askeri alanda oluşabilecek güvenlik açıklarını kapatabilecek, hedefin gelecekteki konum ve yönünü %85 ila %95 gibi yüksek doğruluk oranlarıyla tespit edebilecek olmasından dolayı optimal ve güvenli bir sistem sunmaktadır.

5. KAYNAKÇA

Akkose, Onur. (2020, Kasım 19). Object Detection

https://medium.com/deep-learning-turkiye/tagged/object-detection

Kerimova, Firengiz. (2023, Ağustos 16). YOLOv8.

https://medium.com/@firengizz099/yolov8-984b93c93c01

Krampah, Prince. (2022, Haziran, 1). INTRODUCTION TO OCMPUTER VISION WITH OPENCV PYTHON.

https://medium.com/@princekrampah/introduction-to-computer-vision-with-opencv-python-63662736fab9

Krishna, Rohit. (2023, Ocak 4). Coding: Gaussian Blue Algorithm from scratch in Python.

https://medium.com/@rohit-krishna/coding-gaussian-blur-operation-from-scratch-in-python-f5a9af0a0c0f

Laaraiedh, Mohamed. (2012, Nisan). Implementation of Kalman Filter with Python Language.

https://arxiv.org/pdf/1204.0375

Labbe, Roger. (2020). Kalman and Bayesian Filters in Python.

https://elec3004.uqcloud.net/2015/tutes/Kalman_and_Bayesian_Filters_in_Python.pdf

Munawar, Muhammad Rizwasn. (2023, Ekim 7). Ultralytics YOLOv8 Vital Features You Should Know!

https://muhammadrizwanmunawar.medium.com/ultralytics-yolov8-vital-features-you-should-know-d472eeeb8812

Pham, Khang. (2022, Kasım 17). YOLOv5 PyTorch Tutorial.

https://medium.com/@khang.pham.exxact/yolov5-pytorch-tutorial-533a043e8c8f

Soyubelli, Ayça. (2024, Mart 14). Görüntü İşleme Nedir, Nasıl Yapılır?

https://gelecekbilimde.net/goruntu-isleme-nedir-nasil-yapilir/