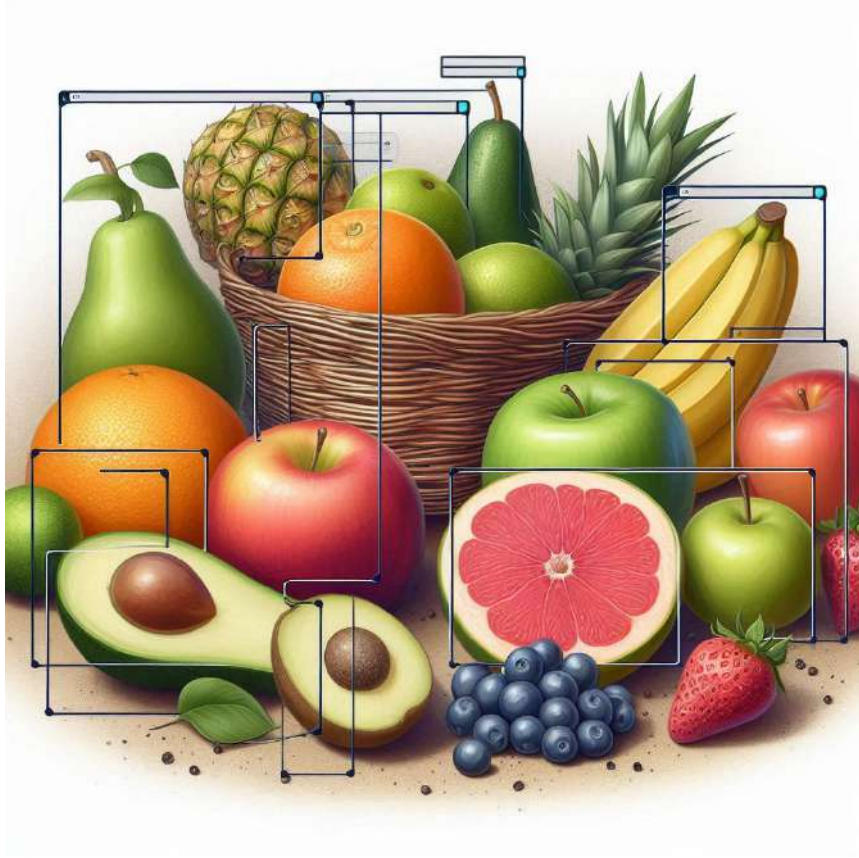


KÜTAHYA SAĞLIK BİLİMLERİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ



Şekil 1: [6]

GÖRÜNTÜ İŞLEME İLE KALORİ HESAPLAMA

Melisa Dursunoğulları
14.6.2024

Özet

Bu çalışma, meyvelerin tanınması ve boyutlarının hesaplanmasında görüntü işleme tekniklerinin kullanımını incelemektedir. Çalışmanın önemi, günlük hayatta meyvelerin kalori hesaplaması ve porsiyon kontrolü gibi alanlarda doğru bilgi sağlamaya yönelik olmasıdır. Çalışmanın amacı, yapay zeka ve görüntü işleme kullanarak meyvelerin doğru bir şekilde tanınmasını ve gerçek boyutlarının belirlenmesini sağlamaktır.

Bu çalışmada, YOLO algoritması ile oluşturulan bir veri seti kullanılarak meyveler tanınmış ve OpenCV kütüphanesi ile parmak boyutu referans alınarak gerçek boyutları hesaplanmıştır. Değişkenler arasında meyve türleri, meyve boyutları ve parmak uzunluğu bulunurken, çıktı olarak tanımlanan meyve türü ve hesaplanan gerçek boyutlar elde edilmiştir.

Kullanılan algoritma, YOLO (You Only Look Once) olup, yüksek doğruluk ve hızlı işlem süresi ile meyve tanımada etkin bir performans sergilemiştir. Çalışma, meyve tanıma ve boyut belirleme görevlerinde başarılı sonuçlar vermiştir. Sonuçlar gösterdi ki, geliştirilen sistem, meyveleri doğru bir şekilde tanıyıp boyutlarını başarılı bir şekilde hesaplayabilmektedir. Yapılan karşılaştırmalar, sistemin doğruluğunu ve güvenilirliğini göstermiştir. Sonuç olarak, bu çalışma, gıda endüstrisi ve sağlık alanında pratik ve etkili çözümler sunmaktadır.

1 GİRİŞ

Görüntü işleme ve yapay zeka tekniklerinin hızla gelişmesiyle birlikte, nesne algılama, sınıflandırma ve analiz alanlarında önemli adımlar atılmıştır. Bu teknolojilerin sunduğu olanaklar, sağlık, eğitim, sanayi ve daha birçok alanda çeşitli dönüşümleri beraberinde getirmiştir. Özellikle nesne tanıma ve kalori hesaplama gibi uygulamalar, günlük yaşamımızda sağlıklı beslenme alışkanlıklarını desteklemekten, ürünlerin kalitesini belirlemeye kadar geniş bir yelpazede etki göstermektedir. Bu rapor, nesne algılama yöntemlerini ele alarak özellikle Sadece Bir Kez Bak (YOLO-You Only Look Once) algoritmasının kullanımıyla meyvelerin algılanması ve sınıflandırılması üzerine bir projeyi içermektedir. Günümüzde, görüntü işleme ve derin öğrenme tekniklerinin hızla gelişmesiyle birlikte nesne algılama, nesne bölümlendirme, vücut pozunu tahmini ve nesne sınıflandırması gibi alanlarda büyük ilerlemeler kaydedilmiştir. Bu teknikler, geniş bir uygulama yelpazesine sahip olup endüstriyel çözümlerde yaygın olarak kullanılmaktadır.

Bu raporda, YOLO algoritması kullanılarak meyvelerin algılanması ve sınıflandırılması üzerine odaklanılmıştır. Proje kapsamında, bir Webcam kamerasına bağlanarak gerçek zamanlı görüntü işleme tekniklerinin kullanılması ve YOLO algoritmasının nesnelerin tespiti ve sınıflandırılmasında temel bir araç olarak kullanılması amaçlanmıştır. Meyvelerin gerçek boyutlarını belirlemek için el ve tırnak boyutları referans alınmıştır. Bu kapsamda antropometrik ölçümler incelenmiş ve parmak boyları arasındaki altın oran dikkate alınmıştır. Kameradan alınan görüntülerle el ve tırnak boyutları ölçülerek meyvelerin gerçek boyutları hesaplanmış ve bu hesaplamalar referans alınarak meyve ve sebzelerin gerçek boyutları belirlenmiştir. Ayrıca, projede veri artırma tekniklerinin kullanımı üzerinde durulmuştur. Veri artırma, mevcut veri çeşitliliğini artırmak için kullanılan bir strateji olup kırpma, doldurma ve yatay çevirme gibi tekniklerle büyük sinir ağlarını eğitmek için yaygın olarak kullanılmaktadır. Bu tekniklerin kullanımıyla elde edilen verilerin çeşitliliği artırılarak daha iyi sonuçlar elde edilmesi hedeflenmiştir.

Çalışmanın öne çıkan noktaları, nesne algılama ve derin öğrenme tekniklerinin meyve tanıma ve sınıflandırma gibi uygulamalarda potansiyelini vurgulamakta ve bu tekniklerin iş süreçlerini daha akıllı, verimli ve doğru bir şekilde yönetilebilir kılma kapasitesine dikkat çekmektedir.

2 LİTERATÜR ARAŞTIRMASI

2.1 Veri Toplanması Ve Hazırlaması

Bu proje için toplanan verileri etiketlemek için Roboflow kullanılacaktır. Roboflow platformunu tanımak adına bir kaç deneme yapılmış ve youtube 'dan örnek video bakılmıştır[16].

Cisimlerin bulunma ve doğru bulunma olasılığını arttırmak için Veri Arttırma(Data Augmentation) yöntemi kullanılmıştır[10].

2.2 Cisimlerin Tanınması

Etiketlenmiş veriler YOLOv8 ile tanımlanacaktır. YOLOv8 serisi gerçek zamanlı nesne dedektörlerinin en son yinelemesidir ve doğruluk ve hız açısından en üst düzeyde performans sunar. Bu modeller, nesne algılamadan örnek segmentasyonu, poz/anahtar nokta algılama, yönlendirilmiş nesne algılama ve sınıflandırma gibi daha karmaşık görevlere kadar çeşitli gereksinimleri karşılamak üzere tasarlanmıştır [12].

Kameradan alınan görüntülerden nesne tespiti yapmak için OpenCV den yararlanılmıştır. OpenCV: bilgisayarla görü, makine öğrenimi, görüntü

işleme, video analizi gibi uygulamalar için kullanılan devasa bir açık kaynak kodlu kütüphanedir. OpenCV kütüphanesi sayesinde web kameraları, video dosyaları veya diğer aygıt türleri tarafından bir bilgisayara bağlanan görsel bilgilerin yakalanmasını, analiz edilmesini ve değiştirilmesini destekleyen yüzlerce işlev içerir [4].

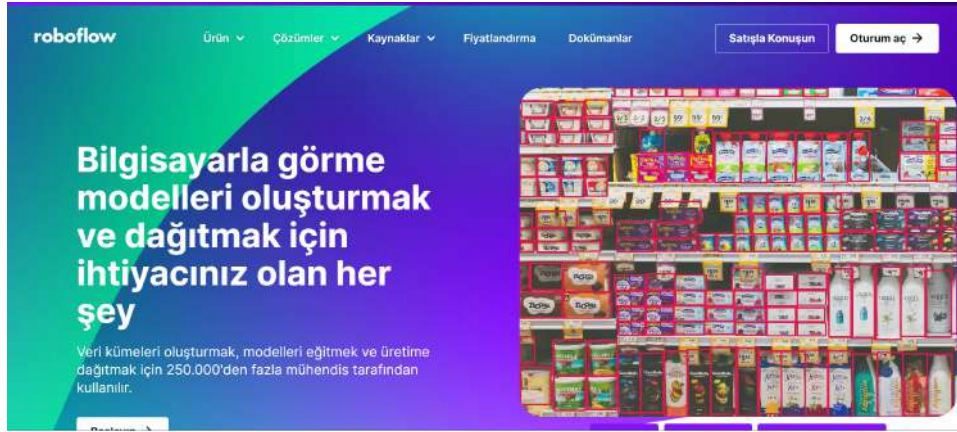
2.3 Boyut Bulunması

MediaPipe ile elin eklemleri algılanmış ve el ölçüsü referans alınarak gerçek boyut bulma hedeflenmiştir. MediaPipe Hands çözümü ile bir görüntü üzerinde ellerin ve el eklemlerinin algılanmasına olanak sağlıyor.

3 METODOLOJİ

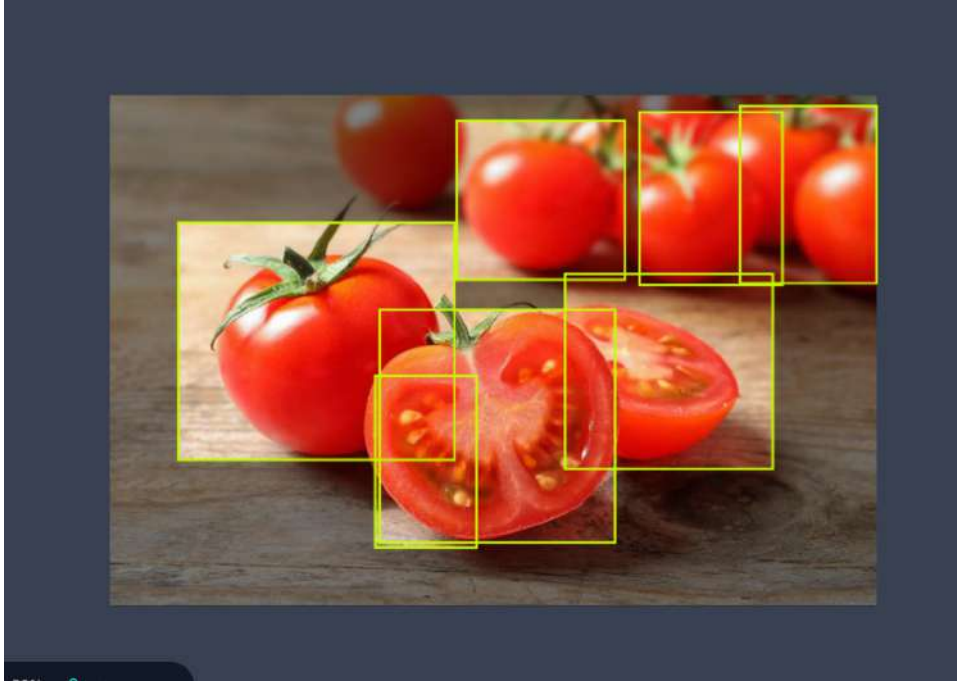
3.1 ROBOFLOW

Görüntü işleme ile kalori hesaplama projesi için özel olarak toplanan veriler roboflow [1] yardımıyla etiketleme yapılarak train-valid-test olarak 3 gruba ayrılacaktır. Bu sayede nesnenin tanınması için yapay zeka eğitilecek ve test edilebilecek konuma gelecektir.



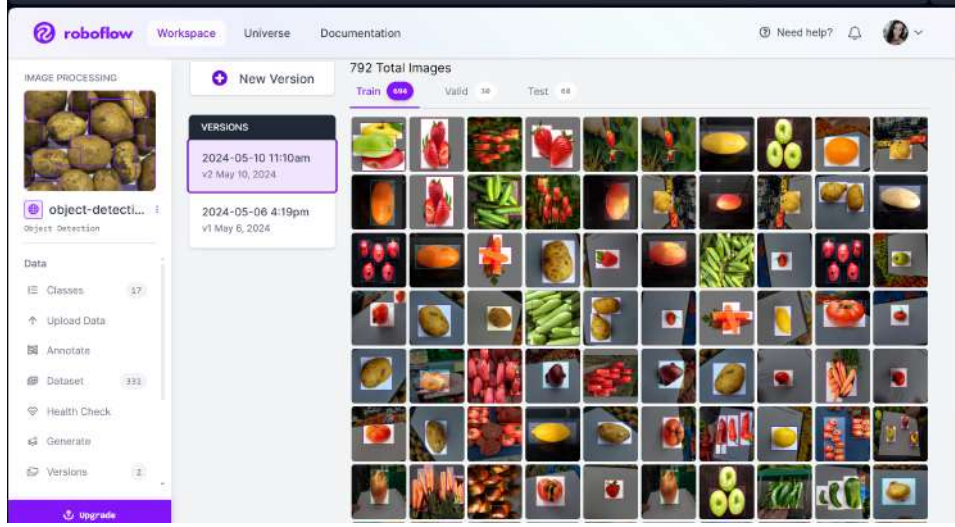
Şekil 2: Roboflow

Görüntü etiketleme, bilgisayarlı görü (Computer Vision) modellerinin geliştirilmesinde çok önemli bir adımdır. Görüntü etiketleme, görüntülerin bilgisayarlı görü algoritmaları tarafından daha kolay anlaşılmasını sağlamak için bir görüntüye ilgili bilgilerin eklenmesidir. Etiketleme aşamasında sınırlayıcı kutular ve hatta görüntüdeki farklı nesnelerin ayrıntılı segmentasyonları yer alabilir.



Şekil 3: Roboflow : Etiketlemeye örnek

Şekil 3 de etiketleme işlemine örnek verilmiştir.

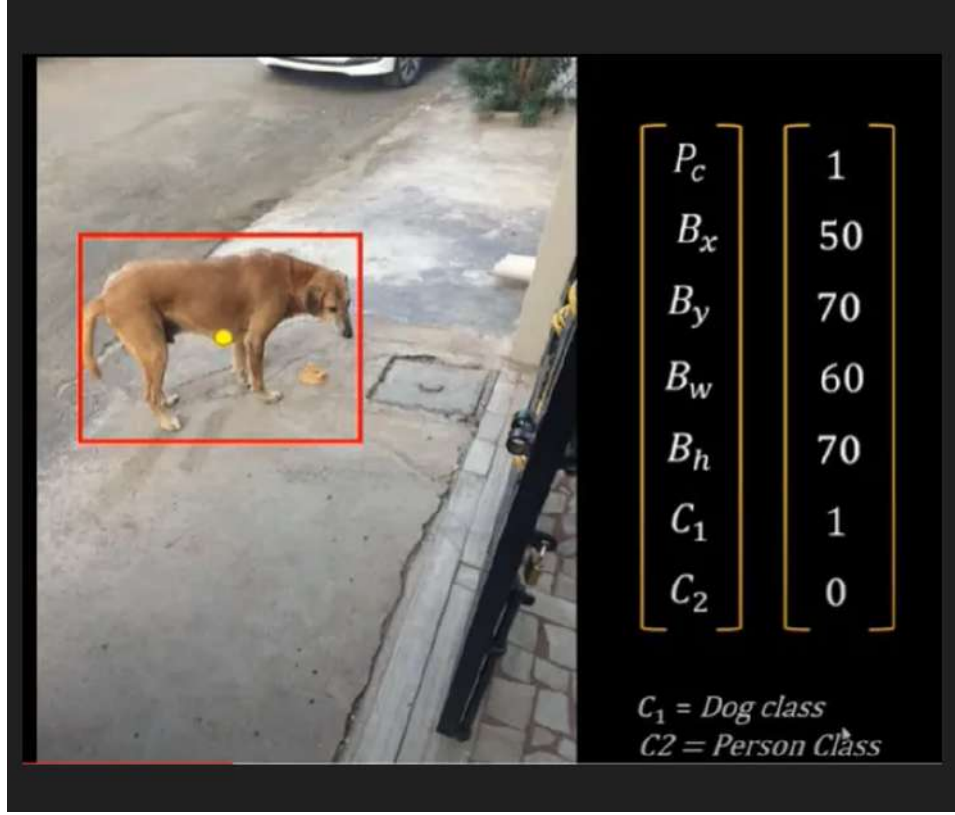


Şekil 4: Roboflow

3.2 YOLO

YOLO (You only look once) türkçe karşılığı ‘Yalnızca bir kez bak ‘ olan, gerçek zamanlı nesne takibi için CNN (Convolutional Neural Network - Evrişimli Sinir Ağı) kullanan en yaygın algoritmadır. Gerçek zamanlı nesne takibi için RCNN (Regions with Convolutional Neural Networks), Fast R CNN (daha hızlı ve daha verimli) , Faster R CNN (daha hızlı daha doğru sonuç- nesne tespiti ve bölge önerisi (region proposal) aşamalarını tek bir ağ içinde birleştirir.) gibi uygulamalar 2015 yılında YOLO piyasaya sürülene kadar kullanılan popüler uygulamalardı. YOLO’yu diğer uygulamalardan ayıran en önemli özelliği çok hızlı olmasıdır. YOLO’nun çalışma mekanizmasına bakıldığında diğer nesne algılama yöntemlerinden farklı ve hızlı kılan birçok yönü bulunmaktadır.

YOLO, görseli tek seferde bir sinir ağından geçirerek resimdeki nesnelerin koordinatlarını ve sınıfını tahmin eder. Bu, ağın bir resmi tek bir geçişte tarayarak ağır hesaplamalara gerek kalmadan nesneleri tespit edebilmesini sağlar. Bu tanımlamayı yaparken görseli 3×3 , 4×4 , 19×19 ızgaralara (grids) ayırır. Izgaraların sayısını belirlemek için belirli bir koşul yoktur. $N \times N$ formatında olması yeterlidir. Her ızgara kendi içerisinde nesne olup olmadığını ve nesne var olduğunu düşünüyorsa merkez noktasının kendi alanında olup olmadığını düşünür. Resim sinir ağından geçtikten sonra çıktı olarak bir vektör meydana gelir.[3]



Şekil 5: Photo on Codebasics Youtube Channel by Dhaval Pate

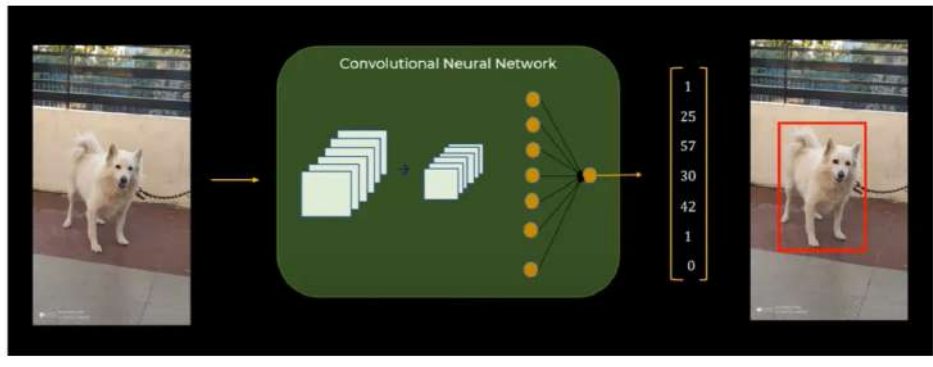
Şekil 5’de görüldüğü gibi bir nesne için bir vektör meydana gelmektedir.

- P_c (Probability of class): Sınıf olasılığıdır. Burada nesnenin yer alıp almasına göre değerlendirme yapılır. Nesne olma olasılığı 1 ile , nesne olmama olasılığı 0 ile belli edilir. Bu görselde çerçevenin içerisinde köpek olduğu için P_c değeri 1’dir.
- B_x : Köpeğin merkez noktasının (sarı nokta ile gösterilmiştir) x koordinatıdır.
- B_y : Köpeğin merkez noktasının (sarı nokta ile gösterilmiştir) y koordinatıdır.
- B_w : Kırmızı çerçevenin enidir .(Nesnenin eni, buradaki nesne köpektir.)
- B_h : Kırmızı çerçevenin yüksekliğidir. (Nesnenin yüksekliğidir, buradaki nesne köpektir.)
- C_1 : Köpek sınıfını ifade eder.

- C2: İnsan sınıfını ifade eder.

Nesne tespitine iki ana yaklaşım vardır:

- İki aşamalı dedektörler Daha Hızlı R-CNN gibi, önce bölge önerileri üretir, ardından her bölgenin sınırlarını sınıflandırır ve hassaslaştırır. Daha doğru olma eğilimindedirler ancak daha yavaşırlar.
- Tek aşamalı dedektörler YOLO gibi, tek geçişte doğrudan görüntünün üzerine bir model uygulayın. Çok hızlı çıkarım süreleri için doğruluktan ödün veriyorlar.

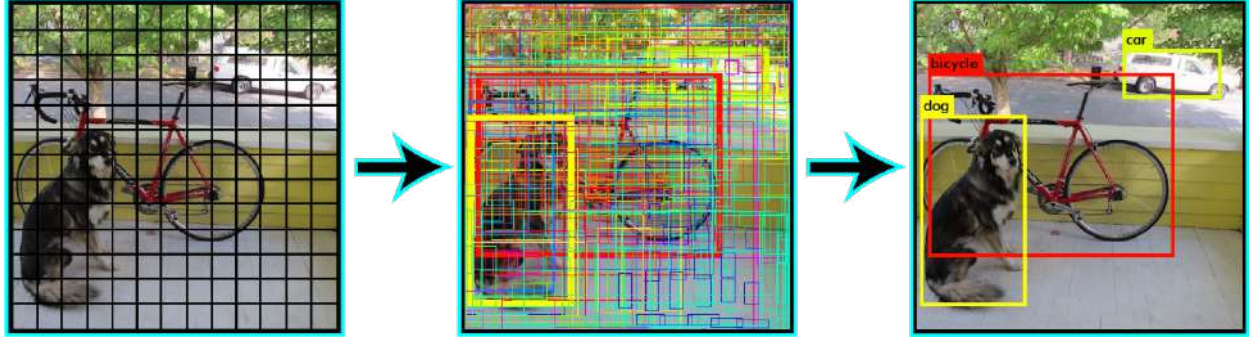


Şekil 6: Photo on Codebasics Youtube Channel by Dhaval Pate



Şekil 7: Nesne tanıma işlemi yapılacak fotoğraf

Şekil 7'deki resime YOLO ile nesne tanıma yaptırılacaktır.



Şekil 8: detection işlemi yapılacak fotoğraf

İlk önce girdi resmini $S \times S$ 'lik ızgaralara bölüyor. Her bir ızgara kendi içinde, alanda nesnenin olup olmadığını, varsa orta noktasının içinde olup olmadığını, orta noktası da içindeyse uzunluğunu, yüksekliğini ve hangi sınıftan olduğunu bulmakla sorumlu. Resimde köpeğin orta noktasının olduğu ızgara köpeğin tespit edilmesinden/etrafına kutucuk çizmesinden sorumlu.

Buna göre YOLO her ızgara için ayrı bir tahmin vektörü oluşturur. Her bir ızgara sadece 1 tane nesne tanımlayabiliyor.

Güven skoru: Bu skor modelin geçerli ızgara içinde nesne bulunup bulunmadığından ne kadar emin olduğunu gösterir. (0 ise kesinlikle yok 1 ise kesinlikle var) Eğer nesne olduğunu düşünürse de bu nesnenin gerçekten o nesne olup olmadığından ve etrafındaki kutunun koordinatlarından ne kadar emin olduğunu gösterir.

$$\text{Güven Skoru} = \text{Pr}(\text{obj}) \times \text{IoU}$$

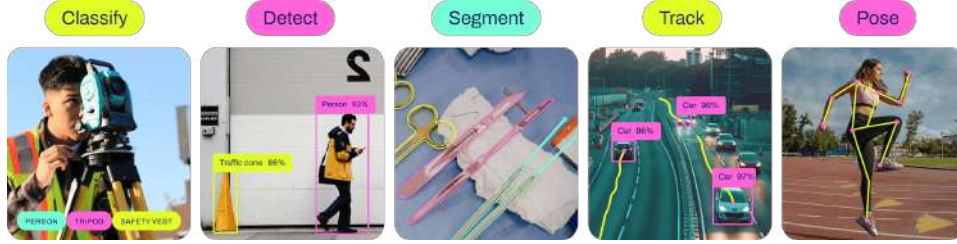
$\text{Pr}(\text{obj})$: ızgara içinde nesnenin bulunma olasılığı

IoU : gerçekte nesnenin bulunduğu kutu ile tahmin edilen kutunun kesişimi

YOLO'nun çalışma mantığını özetlersek,

- Algoritma görseli öncelikle ızgaralara ayırır.
- Her bir bölgedeki nesnelerin etrafına çerçeve (bounding box) çizer.
- Her bir bölgede nesne bulunma olasılığını hesaplar.
- Her bir çerçeve (bounding box) için güven skoru hesaplar.

Diğer geleneksel nesne algılama yöntemlerine kıyasla YOLO, oldukça hızlı ve gerçek zamanlı uygulamalarda etkili bir şekilde çalışabilir. Bu özelliği, özellikle canlı video akışı gibi anlık işlemlerde ve gerçek zamanlı uygulamalarda avantaj sağlar.



Şekil 9: ultralytics

Detection (COCO)

COCO, nesne tanıma, nesne tespiti ve nesne segmentasyonu için popüler bir veri kümesidir. İçinde 80 farklı nesne kategorisi barındırır ve her bir resimde bu nesnelerin detaylı etiketleri bulunur. Özellikle nesne segmentasyonu için ayrıntılı piksel bazında segmentasyon maskeleri sağlar.

Detection (Open Image V7)

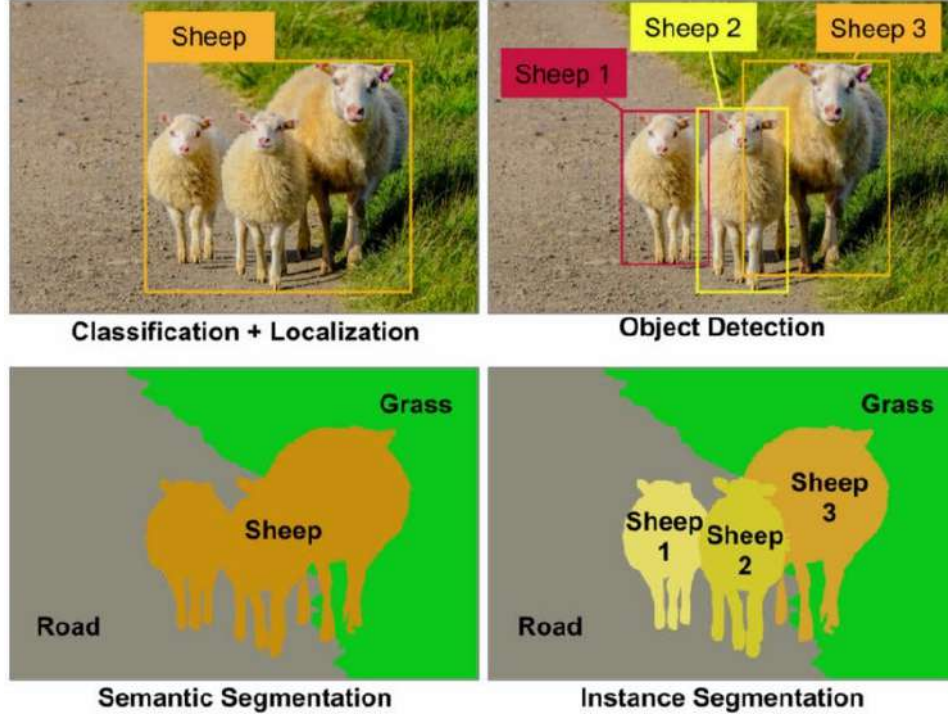
Google tarafından sunulan büyük bir resim veri kümesidir. COCO'dan daha geniş bir etiket kategorisi yelpazesi sunar ve 9 milyondan fazla etiketlenmiş resme sahiptir. Sınırlayıcı kutular ve sınıf etiketleriyle nesne tespiti ve sınıflandırma görevlerinde kullanılır. Open Images, COCO gibi ayrıntılı piksel bazında segmentasyon maskeleri sağlamaz. Ancak nesnelerin sınırlayıcı kutuları ve sınıf etiketleri mevcuttur.

Farkları

COCO, daha az etiket kategorisine sahip olmasına rağmen, nesne segmentasyonu için detaylı maske bilgileri sağlar. Open Images V7, çok daha geniş bir etiket kategorisi yelpazesi sunar ve çok daha fazla sayıda resme sahiptir. COCO'nun daha spesifik görevler için daha ayrıntılı veri sağladığı söylenebilirken, Open Images V7 daha genel ve çeşitli veri sağlar. Hangi veri kümesinin kullanılacağı, kullanım senaryosuna ve ihtiyaca bağlıdır. Örneğin, daha geniş bir nesne kategorisi yelpazesi gerekiyorsa Open Images V7 tercih edilebilirken, nesne segmentasyonu gibi piksel bazında ayrıntılı bilgiler gerekiyorsa COCO daha uygun olabilir.[2]

Segmentation

Temelde, bir görüntüdeki nesnelerin veya bölümlerin (segmentlerin) belirlenmesi işlemidir. Bu işlem, görüntüdeki pikselleri farklı nesnelere veya bölümlere ayrılmış gruplara ayırarak gerçekleştirilir. Segmentasyon, nesne tanıma, nesne tespiti, nesne izleme, medikal görüntüleme ve daha birçok alanda kullanılır.



Şekil 10: segmentation

Semantic (Anlamsal) Segmentasyon: "Semantic Segmentasyon" veya "Anlamsal Segmentasyon", bir görüntüdeki her bir pikselin ait olduğu nesne veya bölge sınıfını belirlemeyi amaçlayan bir görüntü işleme ve derin öğrenme görevidir. Temel amacı, görüntüdeki her pikseli, o pikselin ait olduğu nesne veya bölgenin genel sınıfına atamaktır. Bu yöntem, pikselleri ayrı ayrı nesnelere veya bölümlere sınıflandırarak görüntüyü anlamlandırmayı hedefler.

Instance Segmentation: nesne algılama ve nesne segmentasyonu yöntemlerinin bir kombinasyonudur. Bu yöntem, bir görüntüdeki her bir nesneyi ayrı tanımlamayı ve sınıflandırmayı, aynı zamanda her bir nesnenin piksel bazında segmentasyonunu gerçekleştirmeyi amaçlar. Yani, görüntüdeki her bir nesneyi belirlerken, nesnelerin sınırlarını (bounding box) çizerken aynı zamanda her bir nesnenin piksellerini de ayrı tanımlar.

Classification

"Classification" (Sınıflandırma), makine öğrenimi ve derin öğrenme alanlarında önemli bir konsepttir. Temelde, bir girdiyi belirli bir kategoriye veya sınıfa atama işlemidir. Bu işlem genellikle önceden belirlenmiş bir sınıf listesinden birine karar vermek için yapılır. Örneğin, bir resmi "köpek", "kedi", "araba" gibi belirli bir kategoriye sınıflandırmak gibi.[15]



Şekil 11: segmentation

3.3 OPENCV

OpenCV, çeşitli formatlardaki görüntüleri ve videoları yüklemek, görüntülemek ve kaydetmek için kolay kullanımlı fonksiyonlar sağlamaktadır. Görüntülerin ve videoların keskinleştirilmesi, bulanıklaştırılması, gri tonlamaya dönüştürülmesi gibi çeşitli ön işleme adımlarını gerçekleştirmektedir. OpenCV, derin öğrenme modellerini (örneğin, YOLO, SSD, Faster R-CNN) kullanarak nesne tanıma gerçekleştirmek için DNN (Deep Neural Network) modülünü içermektedir.

OpenCV, gerçek zamanlı video akışını (örneğin, bir web kamerasından gelen video) işleyebilmekte ve bu akış üzerinde nesne tanıma algoritmalarını çalıştırabilmektedir. Hareketli nesnelerin algılanması ve izlenmesi için kullanılabilir. Nesnelerin daha hassas bir şekilde tanımlanması için görüntü segmentasyonu teknikleri kullanılmaktadır. Bu projede OpenCV, kendi verilerimizi kullanarak makine öğrenimi modelleri eğitmek ve bu modelleri nesne tanıma için kullanmak için araçlar sağlamaktadır.

OpenCV’de Canny Fonksiyonu ile Kenar Tespiti

Canny kenar tespit fonksiyonu, eşik değerleri arasındaki piksel değerlerini kullanarak görüntüdeki kenarları tespit eder. Görüntü parlaklığının keskin bir şekilde değiştiği noktaları tespit etmeye dayanan bir algoritmadır.

- Threshold1: Minimum eşik değeri. Bu değer altındaki piksel değerleri kenar olarak tanımlanmaz. (Düşük eşik değeri, daha fazla gürültü ve fazla sayıda kenar tespiti anlamına gelir.)
- Threshold2 : Maksimum eşik değeri. Bu değer üzerindeki piksel değerleri kenar olarak tanımlanır. (Yüksek eşik değeri, daha az gürültü ve daha az sayıda kenar tespiti anlamına gelir.)

Canny kenar tespiti fonksiyonunun minimum eşik değeri “0” ve maksimum eşik değeri “255” olarak ayarlanmıştır. Eşik değerleri arasındaki piksel değerleri, görüntüdeki kenarları tanımlamak için kullanılır ve sonuç binary bir görüntü olarak döndürülür.

Threshold1 ve Threshold2 değerlerinin seçimi kullanılan görüntüye göre değişkenlik gösterir. Canny fonksiyonunu kullanmadan önce görüntüye bazı filtreler uygulanabilir.

Adım 1: Görüntünün Yüklmesi ve Ölçeklendirilmesi

İlk adımda, bir görüntü yüklenir ve isteğe bağlı olarak belirtilen bir ölçekte yeniden boyutlandırılır. Ölçeklendirme, görüntünün işlem süresini azaltmak ve bellek kullanımını kontrol etmek için kullanışlıdır. Örneğin, büyük bir görüntüyü işlerken bilgisayar kaynaklarına daha az yük bindirmek için ölçeklendirme yapılabilir.

Adım 2: Ön işleme (Preprocessing)

Ön işleme adımı, görüntü üzerinde bazı ön işlemlerin gerçekleştirildiği aşamadır. Bu adımlar şunları içerir:

- Gri Tonlama (Grayscale): Renkli bir görüntüyü gri tonlamaya dönüştürmek, işlemleri basitleştirmek için yaygın bir adımdır. Gri tonlamalı görüntü, her pikseldeki renk bilgisini tek bir yoğunluk değerinde temsil eder.
- Gaussian Bulanıklaştırma (Gaussian Blur): Gürültüyü azaltmak ve daha iyi kenar algılaması için görüntüye bir Gaussian filtresi uygulanır. Bu, kenarların daha belirgin olmasını sağlar.
- Kenar Algılama (Canny Edge Detection): Kenar algılama, görüntüdeki önemli kenarları tespit etmek için kullanılır. Canny kenar algılama, diğer yöntemlere göre daha hassas ve doğru sonuçlar verir.

- Morfolojik İşlemler (Dilation ve Closing): Dilation, kenarların kalınlığını artırarak görüntüdeki boşlukları doldurur. Closing, kenarları birleştirerek nesnelerin daha düzgün bir şekilde algılanmasını sağlar.

Adım 3: Kontur Bulma ve Köşe Noktalarının Tespiti

Kontur bulma, görüntüdeki nesnelerin sınırlarını belirlemek için kullanılır. Burada, nesnenin dış sınırlarını çevreleyen bir kontur elde edilir. Bu kontur, nesnenin şeklini ve boyutunu belirlemek için kullanılacaktır.

Kontur tespiti yapıldıktan sonra, nesnenin köşe noktalarını tespit etmek için bir dizi işlem yapılır. Bu köşe noktaları, nesnenin dörtgen bir şekil olduğunu düşünerek, daha sonra perspektif dönüşümü için kullanılacaktır.

Adım 4: Perspektif Dönüşümü (Warping)

Perspektif dönüşümü adımı, nesnenin görsel olarak düz bir açıdan görülebilmesi için görüntü üzerinde bir dönüşüm yapılır. Bu adım, nesnenin gerçek boyutunu doğru bir şekilde belirleyebilmek için önemlidir. Örneğin, bir A4 kağıdının köşe noktaları belirlenerek bu köşe noktalarından bir dörtgen oluşturulur ve bu dörtgen, görüntüdeki nesnenin yerini temsil eder. Daha sonra bu dörtgen, A4 kağıdının gerçek boyutlarına göre yeniden boyutlandırılır.

Adım 5: Boyutların Hesaplanması

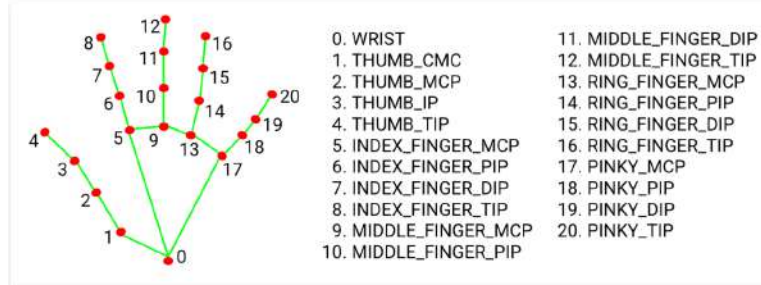
Perspektif dönüşümü sonrasında, dönüştürülen nesnenin genişliği ve yüksekliği piksel cinsinden hesaplanır. Bu adım, nesnenin görüntü üzerindeki piksel boyutlarını belirler.

Adım 6: Piksel Boyutlarının Gerçek Boyutlara Dönüştürülmesi

Son olarak, nesnenin piksel boyutları gerçek dünya boyutlarına dönüştürülür. Bu adım, bir referans nesne kullanarak gerçek dünya boyutlarına göre piksel boyutlarının hesaplanmasını içerir. Örneğin, bir A4 kağıdının gerçek boyutlarını biliyorsak ve görüntüdeki A4 kağıdının piksel boyutlarını hesapladıysak, diğer nesnelerin boyutlarını da bu oranlar kullanılarak hesaplayabiliriz.

Bu adımlar, nesnenin boyutunu belirlemek için tipik olarak izlenen adımlardır ve bu sürecin bir parçası olarak siyah-beyaz yapma, kenar algılama, kontur bulma ve perspektif dönüşüm gibi işlemler kullanılır. Bu işlemler, nesnenin görüntü üzerindeki belirginliğini artırarak ve perspektif etkisini düzelterek doğru boyutları elde etmeyi sağlar.[11]

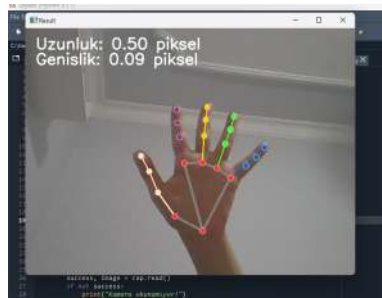
3.4 MEDIAPIPE



Şekil 12: MediaPipe-El Eklemleri

Meyve ve sebzelerin gerçek boyutlarını belirlemek için el ve tırnak boyutunu referans alacak. Kameradan alınan görüntülerle el ve tırnak boyutları ölçülerek meyvelerin gerçek boyutları hesaplanacaktır. Kameradan alınan görüntülerle elin boyutları ölçülecektir. En geniş, en uzun ve en dar noktaların ölçüleri kaydedilecektir. Tırnak boyutları, işaret, orta, yüzük, serçe ve küçük parmakların tırnaklarından ayrı ayrı ölçülecektir. Her bir parmak için tırnak uzunluğu belirlenecektir.

Mediapipe ve OpenCV gibi görüntü işleme kütüphaneleri kullanılarak, kameradan alınan görüntülerde el ve tırnaklar tanımlanacaktır. Bu adımda, elin genel yapısı ve tırnakların konumu tespit edilecektir. El ve tırnak boyutları piksel cinsinden ölçüldükten sonra, kameranın sabit bir uzaklığına göre gerçek dünya ölçülerine dönüştürülecektir. El ve tırnak boyutları referans alınarak, meyve ve sebzelerin görüntülerinde el ile karşılaştırma yapılacaktır. El ve tırnak boyutlarına göre meyve ve sebzelerin gerçek boyutları hesaplanacaktır.¹



Şekil 13: MediaPipe Webcamden El Tespiti

¹ Bu yöntem, meyve ve sebzelerin gerçek boyutlarını belirlemek için kullanılan bir referans ölçüm yöntemidir.

3.5 Antropometri Nedir?

Antropometri tanımı, “insan” ve “ölçüm” anlamına gelen antropos ve metrikos Yunanca kelimelerine dayanmaktadır. İlgili alanlarda kullanılmak üzere insan vücudu parçalarının ölçüm verilerini organize etme, türetme ve analiz etme bilimidir. Antropometri, bireysel farklılıklara odaklanan ve çeşitli tasarım disiplinleri için önemli referanslar ve girdiler sağlayan ergonomi uygulamalarının hayati bir parçasıdır. Koltuklar, konsollar, ATM, araç iç hacimleri, cep telefonları, gamepadlar gibi insan vücudu ölçüleri içeren ürünler için ana tasarım çalışma konularından biridir. Mekanik tasarım yapacak mühendisler için de antropometri çalışmaları zaman zaman önem kazanmaktadır.[9]

Vücut ölçülerinin elde edilmesine yönelik, statik ve dinamik (fonksiyonel) antropometri olmak üzere iki farklı metot geliştirilmiştir.

Statik antropometri

İnsanların statik duruş ve oturuşlarında ölçülen boyutları ele alan bir uğraş alanıdır. Antropometrik ölçüler ayakta durma ve düz bir zeminde oturma durumlarına bağlı olarak özel aletlerin kullanımıyla alınmakta ve farklı ergonomik tasarımlarda kullanılmaktadır. Statik boyutlar, dirsek ve bilek arası ölçümler ile eklem merkezleri arasında ölçümler gibi insan iskeleti boyutları yanı sıra baş çevresi, cilt yüzeyi çevre ölçüleri gibi dış hat boyutlarını içermektedir. Çeşitli yaş gruplarındaki okul çocuklarının oturacağı sıraların boyutlarını saptamanın yanı sıra, bir gaz maskesinin yüz ölçülerine uygun bir şekilde ve boyutlarda imali için ihtiyaç duyulan antropometri ölçümleri de statik antropometri yaklaşımı ile elde edilir.

Dinamik Antropometri

Endüstri ve iş ortamında iş görenler sürekli devinim hâlinde dirler. Bir iş gören işini yaparken çeşitli yönlerde uzanması, kol, bacak ve gövdesini değişik boyutlarda ve devamlı hareket ettirmesi nedeni ile çeşitli dinamik ölçülerin bilinmesine ihtiyaç duyulur. Fonksiyonel antropometri olarak da bilinen dinamik antropometri yaklaşımı ile elde edilen boyutlar, bazı fiziksel aktivitelerde bulunan insan vücudundan belli şartlar altında elde edilirler. İnsanların ayakta dururken ya da otururken çevrelerindeki malzemelere, kontrol sistemlerine ve çeşitli işlem noktalarına uzanabilmeleri için; eğilme, uzanma ve dönme gibi hareketlerinin hudutlarını ölçmek de iş düzeni ve insan-tezgâh, insan-makine gibi arakesitlerin tasarımında optimizasyon açısından önemlidir. Ancak çalışma ortamında insanların, sekreterin masasında bulunan telefona erişmesi, masanın çekmecesinden kâğıt almak için eğilmesi örneklerinde olduğu gibi, hareketlerde bulunmaları nedeniyle çeşitli dinamik boyutların ölçülmesine ihtiyaç duyulmuştur. İnsanların ayakta dururken ya da otururken çevrelerindeki malzemelere, kontrol araçlarına ve çeşitli işlem nok-

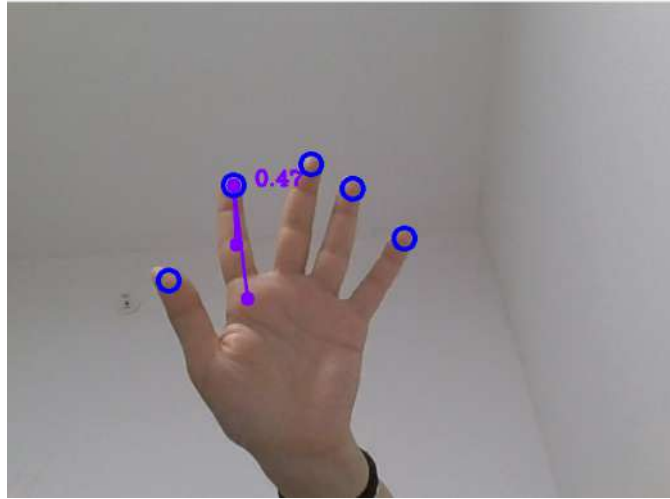
talarına eğilme, dönme, uzanma gibi hareketlerle erişebilecekleri sınırlar dinamik antropometri ile ölçülür.

İnsan Bedeninde Altın Oran

İnsan vücudunda altın orana verilebilecek ilk örnek, göbek ile ayak arasındaki mesafe bir birim olarak kabul edildiğinde, insan boyunun 1,618'e denk gelmesidir. Bunun dışında vücudumuzda yer alan diğer bazı altın oranlar şöyledir:

- Parmak ucu ile dirsek arası / El bileği ile dirsek arası
- Omuz hizasından başucuna olan mesafe / Kafa boyu
- Göbek ile başucu arası mesafe / Omuz hizasından başucuna olan mesafe
- Göbek ile diz arası / Diz ile ayakucu arası mesafe

Altın oran ile uyumlu tüm yapıların insan gözüne estetik açıdan ve uzunlukların birbirlerine uyumu açısından en güzel algılanan yapılar olduğu ve tercih sebebi olduğu gösterilmiştir. Altın oran parçalar veya uzuvlar arasında içsel bir uyum ve güzelliği bulundurmaktadır. Bu orana sahip bütünün tüm öğeleri insan gözüne çekici, estetik ve güzel gelmektedir. Altın orana diğer bir örnek ise Fibonacci dizisine tam bir uygunluk gösteren insan elidir. Parmaklarımız üç boğumludur. Parmağın tam boyunun ilk iki boğuma oranı altın oranı verir (başparmak dışındaki parmaklar için). Ayrıca orta parmağın serçe parmağına oranında da altın oran olduğunu fark edilebilmektedir.



Şekil 14: Parmaktaki Altın Oran

Şekil 14 de her ardışık bölüm arasındaki oran altın oran sabiti olan 0,518... sayısıdır. Ayrıca her bölüm 2, 3, 5, 8'e yani ardışık Fibonacci sayılarına karşılık gelmektedir.

3.6 DATA AUGMENTATION - VERİ ARTTIRMA

Veri artırma, uygulayıcıların, aslında yeni veriler toplamadan, eğitim modelleri için mevcut olan veri çeşitliliğini önemli ölçüde artırmaya olanak tanıyan bir stratejidir. Kırpma, doldurma ve yatay çevirme gibi veri artırma teknikleri, büyük sinir ağlarını eğitmek için yaygın olarak kullanılır [10].

Bu teknik , makine öğrenimi modellerini eğitirken aşırı uyumu azaltmak için makine öğreniminde yaygın olarak kullanılır, modelleri biraz değiştirilmiş birkaç kopya üzerinde eğiterek elde edilir. Veri büyütme, model genellemesini ve performansını iyileştirmek için eğitim veri seti çeşitliliğini zenginleştirerek görüntü sınıflandırmada temel hale geldi . Bu uygulamanın gelişimi, geometrik dönüşümler, renk alanı ayarlamaları ve gürültü enjeksiyonu dahil olmak üzere geniş bir teknik yelpazesini ortaya çıkarmıştır.

3.7 Geometrik Dönüşümler

Geometrik dönüşümler, farklı perspektifleri, yönelimleri ve ölçekleri simüle etmek için görüntülerin uzamsal özelliklerini değiştirir.

Yaygın teknikler şunları içerir:

Döndürme: Modellerin nesneleri çeşitli açılardan tanımasına yardımcı olmak için görüntüleri belirli bir dereceye kadar döndürme.

Çevirme: Yönlendirmede değişkenlik sağlamak için görüntüleri yatay veya dikey olarak yansıtmak.

Kırpma: Belirli özelliklere odaklanmak veya daha yakın görünümünü simüle etmek için görüntünün bazı bölümlerini kaldırmak.

Çeviri: Modellere konumsal değişmezliği öğretmek için görüntüleri farklı yönlerde kaydırmak.

3.8 Renk Uzayı Dönüşümleri

Renk alanı dönüşümleri, aydınlatma, renk doygunluğu ve kontrasttaki farklılıklara değinerek görüntülerin renk özelliklerini değıştirir.

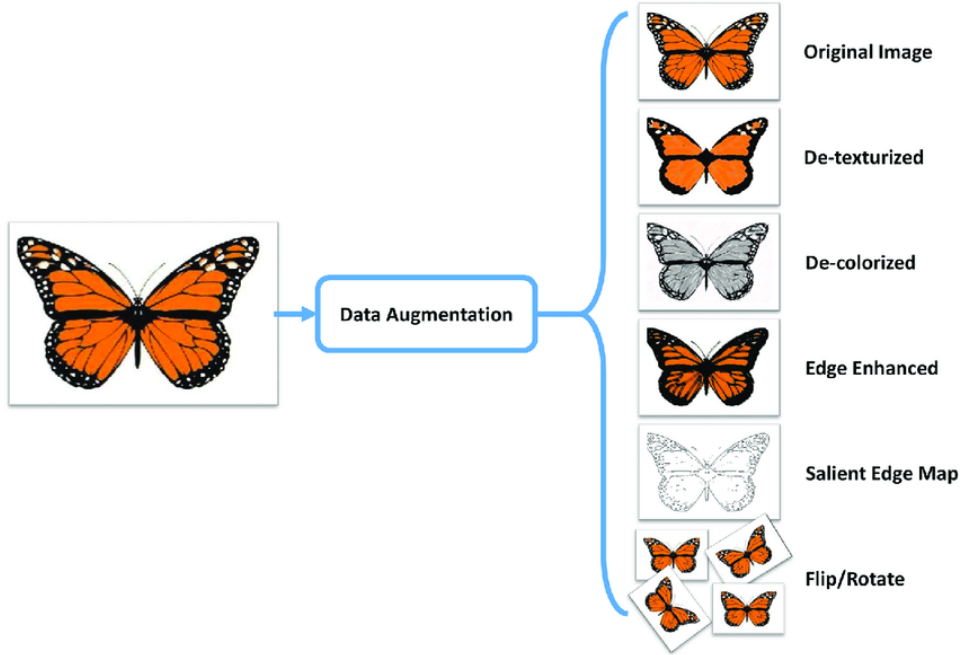
Teknikler şunları içerir:

Parlaklık Ayarı: Farklı aydınlatma koşullarını simüle etmek için görüntünün parlaklığını değıştirme.

Kontrast Ayarı: Modellerin çeşitli netlik seviyelerindeki nesneleri tanımasına yardımcı olmak için kontrastın değıştirilmesi.

Doygunluk Ayarı: Farklı renk yoğunluklarına sahip görüntülere yönelik modeller hazırlamak amacıyla doygunluğun değıştirilmesi.

Renk Değışimi: Renk değışkenliği sağlamak için parlaklık, kontrast, doygunluk ve renk tonunun rastgele ayarlanması [14].



Şekil 15: Augmentation [10]

4 Veri oęaltma



(16a) Orijinal Grsel



(16b) Contrast Uygulanmıř Grsel



(16c) Rotation Edilmiř Grsel

řekil 16: Augmentation rneęi

5 PIL Kütüphanesi

Python Imaging Library (kısaca PIL), Python programlama dili için geliştirilen, açık kaynak kodlu grafik işleme kütüphanesidir. Bu kütüphane, içinde barındırdığı hazır fonksiyonlar sayesinde programcıya üstün bir grafik işleme imkânı sunar. Birçok grafik türünü açıp kaydetme yeteneği ile birlikte çizim, düzenleme, filtreleme gibi işlemlerde kullanılabilecek fonksiyonlara sahiptir[13].

6 Sequential Modeli:

Sequential modeli, katmanları ardışık olarak ekleyerek oluşturulan bir model türüdür. Her bir katman, kendisinden önceki katmanın çıktısını girdi olarak alır.

AUGMENTATION KOD PARÇASI

```
num_classes = 5

model = Sequential([
    data_augmentation,
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

model.fit(X_train_scaled, y_train, epochs=30)
```

AUGMENTATION KODUNUN TERMİNOLOJİSİ

- Sequential: Katmanları sırayla eklememizi sağlar.
- Conv2D: İki boyutlu konvolüsyon katmanıdır. Görüntü verilerini işlemede kullanılır.
- 16, 32, 64: Filtre sayısını belirtir.
- 3: Çekirdek boyutunu (3x3) belirtir.
- padding='same': Çıktının giriş ile aynı boyutta olmasını sağlar.
- MaxPooling2D: Maksimum havuzlama katmanıdır. Uzaysal boyutları küçültmek için kullanılır.
- Flatten: Çok boyutlu girişleri tek boyutlu vektöre dönüştürür.
- Dense: Tam bağlantılı katmandır. Her nöron bir önceki katmandaki tüm nöronlara bağlıdır.
- 128: Nöron sayısını belirtir. activation='relu': ReLU aktivasyon fonksiyonunu kullanır.
- model.fit(X-train-scaled, y-train, epochs=30): Modeli eğitim verisi (X-train-scaled) ve etiketleri (y-train) kullanarak 30 dönem boyunca eğitir.

7 Veri Artırmanın Faydaları

- **Geliştirilmiş genelleme:** Veri artırma, bir modelin verilerdeki farklı varyasyonlara karşı daha dayanıklı olmasını sağlayarak genelleme yeteneğinin geliştirilmesine yardımcı olabilir. Bunun nedeni, modelin eğitim sırasında daha geniş bir veri çeşitliliğine maruz kalmasıdır.
- **Aşırı uyumun azaltılması:** Veri artırma, makine öğreniminde yaygın bir sorun olan aşırı uyumun azaltılmasına da yardımcı olabilir. Aşırı uyum, bir model eğitim verilerini çok iyi öğrendiğinde ve yeni verilere genelleştiremediğinde ortaya çıkar. Veri artırma, eğitim verilerini daha çeşitli hale getirir ve bu da modelin aşırı uyumunu önlemeye yardımcı olur.
- **Artan doğruluk:** Verilerin artırılması genellikle hem eğitim hem de test setlerinde doğruluğun iyileştirilmesine yol açabilir. Bunun nedeni, modelin daha büyük ve daha çeşitli bir veri kümesinden daha sağlam özellikleri öğrenebilmesidir.

- **Azalan maliyet:**Veri artırma, özellikle yeni veri toplamının pahalı veya zaman alıcı olduğu görevler için, eğitim veri kümesinin boyutunu ve çeşitliliğini artırmanın uygun maliyetli bir yolu olabilir [7].

8 BULGU VE TARTIŞMA

8.1 ÇALIŞMA-1 NESNE TANIMA

Bu projede, YOLO (You Only Look Once) nesne algılama algoritmasının farklı meyveleri tanıma ve sınıflandırma yeteneğini incelenmiştir. Özellikle, aynı renklere sahip meyvelerin birbirine benzetilip benzetilmediği ve YOLO'nun bu meyveleri nasıl sınıflandırdığına odaklanılmıştır.

Hazırlanan veri seti üzerinde YOLOv8 algoritması kullanılarak bir nesne algılama modeli eğitilmiştir. Eğitim sürecinde 70 epoch (dönem) ve imgs=640 (çözünürlük) kullanıldı.

İlk olarak, aynı renklere sahip meyvelerin birbirine benzetilip benzetilmediği test edilmiştir. Özellikle, farklı renklerdeki elma örneklerini ve sarı renkteki muz ve limon gibi meyveleri sınıflandırma yeteneği test edilmiştir. Model, genel olarak bu meyveleri sınıflandırmada ortalama bir başarı elde etti.

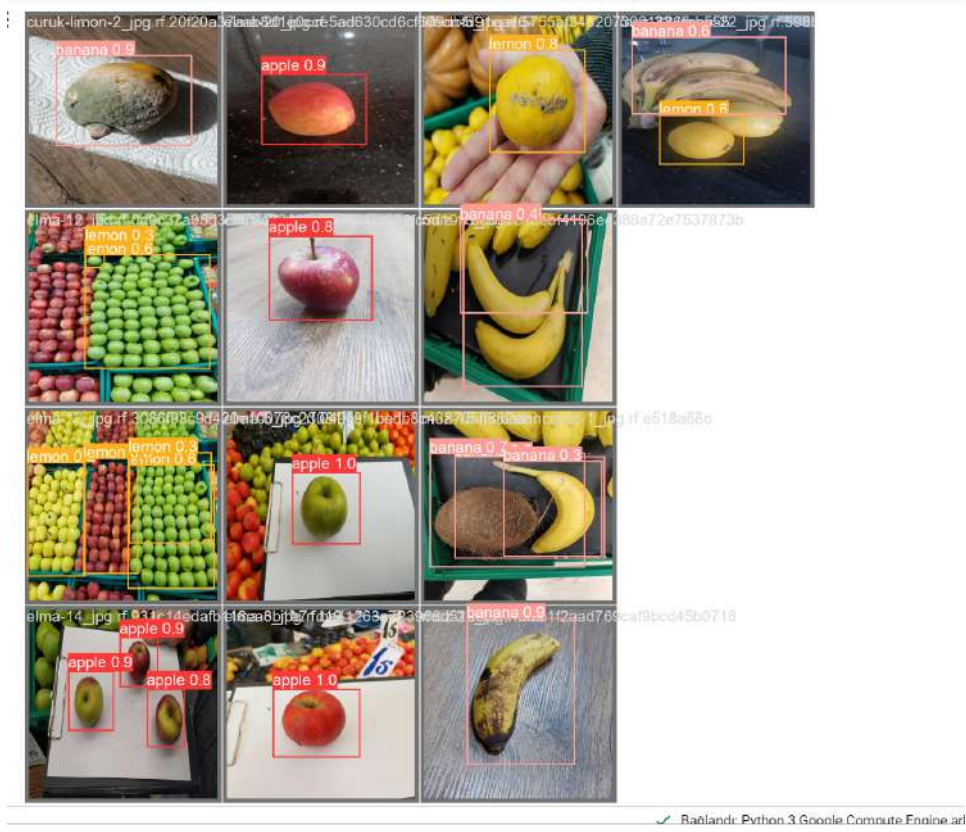
Elma Sınıflandırması: Model, farklı renklerdeki elma örneklerini genellikle doğru şekilde sınıflandıramadı. Kırmızı, yeşil ve sarı gibi farklı renklere sahip elmalara çoğunlukla doğru etiketler verilemedi. Model, bazı sarı renkli elma örneklerini limon olarak etiketledi. Bu durum, modelin renk bazında sınıflandırma yeteneğinde belirli bir hassasiyetsizlik olduğunu gösteriyor.

Muz ve Limon Sınıflandırması: Sarı renkteki muz ve limon gibi meyveleri ayırt etme konusunda genel olarak iyi bir performans sergilemiştir. Model, muz ve limonları doğru şekilde sınıflandırdı ve genellikle bu meyveleri ayırt etmiştir. Ancak bazı durumlarda, özellikle çürük veya hasarlı limonların tanınmasında zorlanma yaşandı. Model, çürük limonları genellikle muz olarak sınıflandırdı.

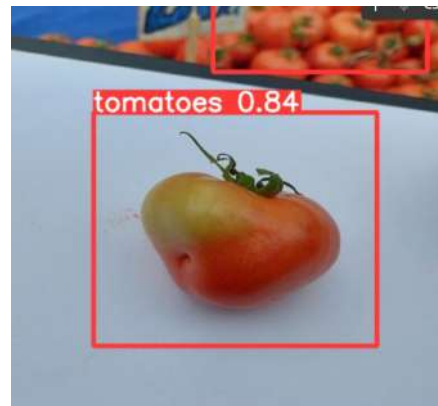
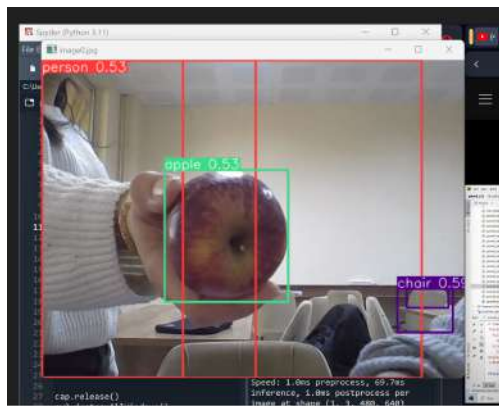
8.2 ÇALIŞMA-2 WEBCAM İLE NESNE TANIMA

Webcam'e bağlanarak gerçek zamanlı görüntü almak, nesne tanıma projemizin ilk adımıdır. Bu adımda, Spyder IDE'si kullanılarak Webcam'e bağlanma işlemi gerçekleştirilecektir. Webcam'den alınan görüntüleri daha sonra nesne tanıma algoritmalarıyla işlenecektir.

OpenCV (Açık Kaynak Bilgisayar Görüntü İşleme Kütüphanesi): Webcam'e bağlanmak ve görüntü işlemek için kullanılacak olan temel kütüphanedir.



Şekil 17: Nesne tespiti ile Güven Skoru



Şekil 18: Nesne Tanıma Görselleri

8.3 ÇALIŞMA-3 CANNY FONKSİYONU İLE KENAR TESPİTİ

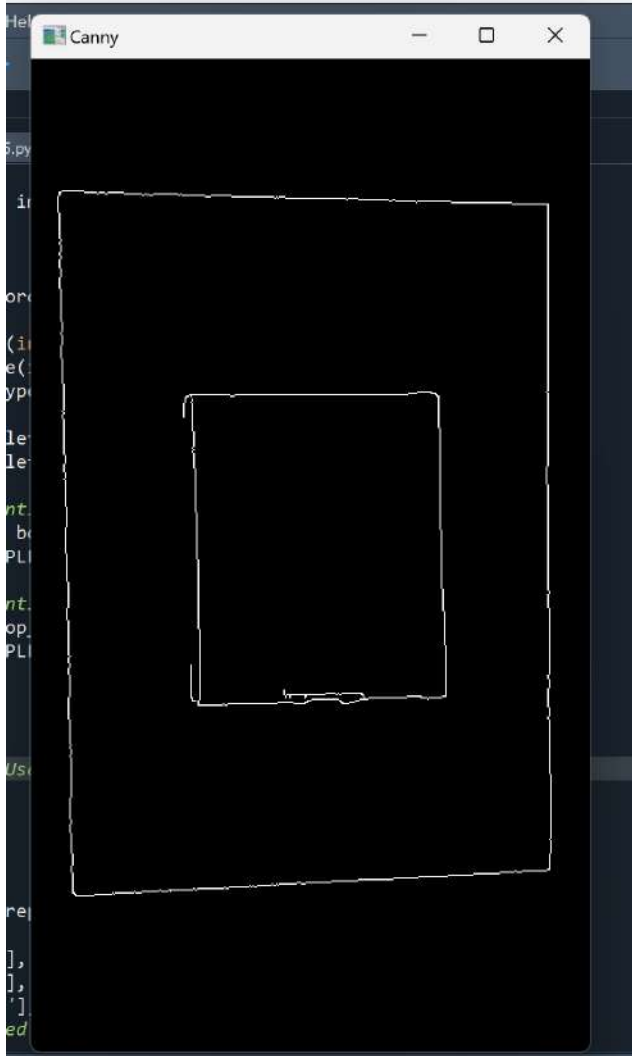
CANNY algoritması ile tespit edilen kenarların uzunluklarının hesaplanarak cismin boyutunun bulunması hedeflenmiştir. İlk olarak, farklı cisimlerde denenen algoritma sonrasında, meyvelerin tam sınırlarından çizilen bounding box'ların boyutunun bulunması ile meyvelerin yaklaşık boyutları belirlenmiştir. Bu süreçte, CANNY kenar tespiti algoritması ile meyvelerin sınırları belirlenmiş, ardından bu sınırlar kullanılarak bounding box'lar çizilmiştir. Çizilen bounding box'ların genişlik ve yükseklik değerleri hesaplanarak meyvelerin boyutları tahmin edilmiştir. Bu yöntem, meyvelerin yaklaşık boyutlarını belirlemek için pratik bir çözüm sunar ve farklı cisimlerde de uygulanabilirliği test edilmiştir.



(19a) Nesne'nin orjinal hali



(19b) Siyah beyaz hali



(20a) Kenarlarının çizilmiş hali



(20b) Referansın kenarlarının çizilmesi



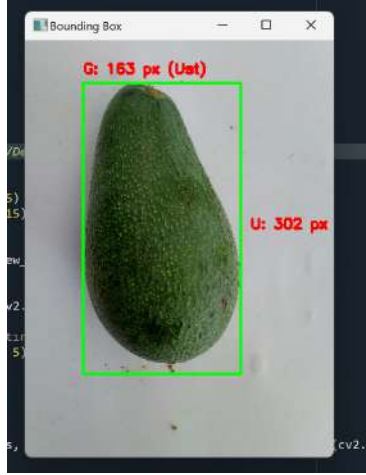
(21a) Nesne'nin çevresinin çizilmesi



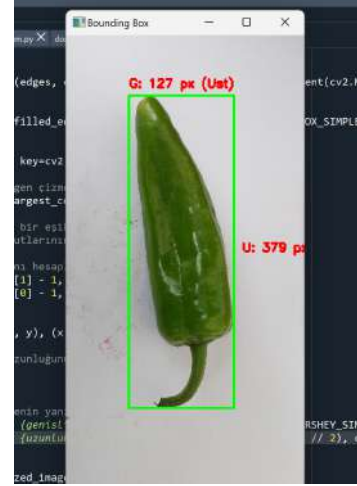
(21b) Nesnenin Boyutları

8.4 Veri Seti Üzerinde Boyutlandırma Çalışmaları

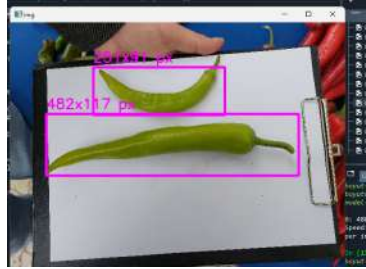
Canny algortiması ile tespit edilen kenarlardan çizilen bouding box ile nesnenin yaklaşık boyutunun bulunması hedeflenmiştir.



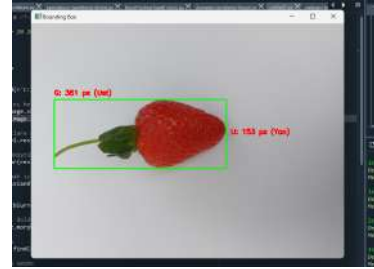
(22a) Avokado Pixel Ölçümü



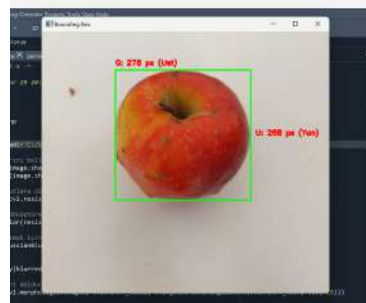
(22b) Biber Pixel Ölçümü



(22c) Domates Pixel Ölçümü



(22d) Çilek Pixel Ölçümü

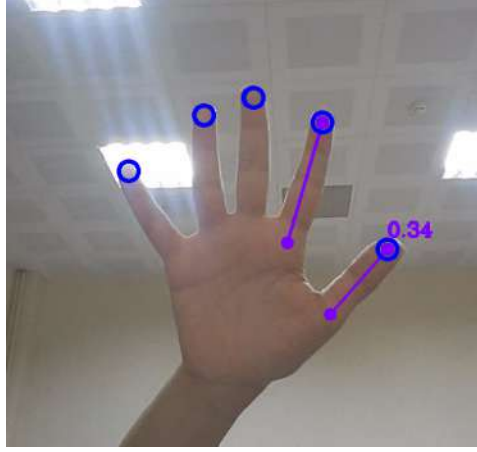


(22e) Elma Pixel Ölçümü

Şekil 22: Pixel Ölçümleri

Şekil -2222a,-2222b,-2222d,-2222e,-2222c de bulanıklaştırma yöntemi ile sınırları belirlenen cisim için tam kenarlarından geçen bouding box çizimi gösterilmektedir.Bouding box ile cisimlerin yaklaşık boyutu hesaplanmıştır.[8][5]

8.5 BařParmak Oran Bulunması

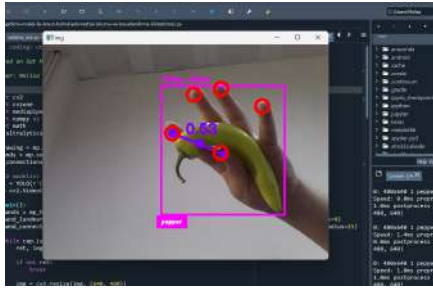


řekil 23: BařParmak Oranı

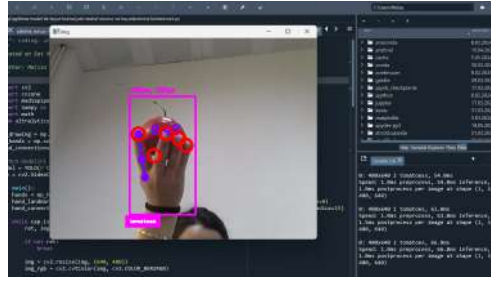
řekil -23 de işaret parmağının altın oranından yola çıkarak işaret parmağın pixel boyutunun altın oranına olan oranından başparmak için bir oran bulunmuştur. Bu oran cisimlerin gerçek boyutunu bulmada kullanılacaktır.

8.6 El Boyutuna Oranla Gerçek Boyut Bulma

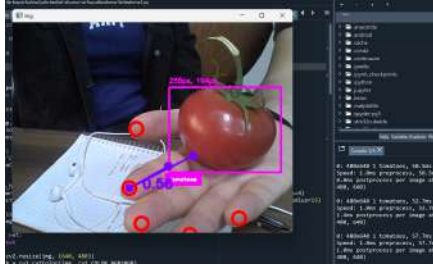
Bu kısımda önceden ayrı ayrı yapılan 2 projenin birleştirilip birlikte çalışması sağlanmıştır. Eldeki işaret parmağının referans olması için piksel boyutunun hesaplandığı proje ile cisimlerin tespit edilip piksel olarak uzunluk ve genişliğinin bulunduğu ve cismin ne olduğunun tespit edildiği projeler birleştirilmiştir.



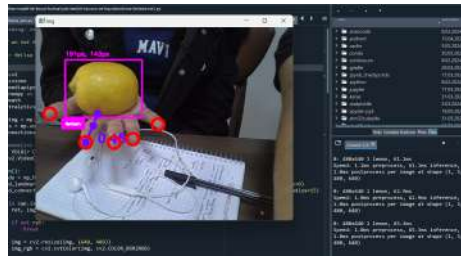
(24a) Biber Pixel Ölçümü



(24b) Domates Pixel Ölçümü



(24c) Domates Pixel Ölçümü



(24d) Limon Pixel Ölçümü

Şekil 24: Cisimlerin Pixel Boyutları Ve El Tanınması

9 SONUÇ

Bu çalışmada, webcam kamerası ile algılanan nesnelerin YOLO algoritması kullanılarak tespiti ve daha sonrasında MediaPipe ile el ölçümünden yola çıkarak gerçek ölçülerinin bulunması hedeflenmiştir. Yapılan işlemler sonucunda, nesnelerin boyutları piksel cinsinden bulunmuş ve bu sonuçlar el ölçüsü kullanılarak gerçek boyutlara dönüştürülmek üzere bir aşamaya getirilmiştir.

Nesne tanıma işlemi için YOLO algoritması kullanılarak başarılı bir şekilde nesneler tespit edilmiştir. Tespit edilen nesnelerin sınırları, bounding box kullanılarak belirlenmiş ve bu sınırlar içerisindeki nesnelerin yükseklik ve genişlik boyutları piksel cinsinden ölçülmüştür. Bu piksel ölçümleri, daha sonra MediaPipe kullanılarak el ölçümleri ile entegre edilmiştir.

Bu entegrasyon, nesne boyutlarının doğru bir şekilde gerçek dünya ölçümlerine dönüştürülmesini sağlamayı amaçlamaktadır. Şu ana kadar, nesnelerin piksel cinsinden boyutları belirlenmiş ve el ölçüsü verileri ile oranlanarak gerçek boyutların bulunması aşamasına gelinmiştir. Son adım olarak, bu oranlama işlemi tamamlandığında, elde edilen sonuçlar sayesinde nesnelerin gerçek boyutları yüksek doğrulukla hesaplanabilecektir.

Sonuç olarak, bu çalışma, nesne tanıma ve boyut belirleme konularında önemli bir ilerleme kaydetmiş olup, uygulamanın son aşaması olan oranlama işlemi ile gerçek boyutların hesaplanmasına çok yakındır. Bu yaklaşım, çeşitli alanlarda pratik uygulamalar için kullanılabilir ve nesnelerin doğru boyutlarının tespit edilmesinde faydalı olabilir.

Kaynakça

- [1] Roboflow nedir? <https://roboflow.com/>, 2024.
- [2] abirami vina and glenn jocher. Documentation. <https://medium.com/deep-learning-turkiye/yolo-algoritmasÄŒnÄŒ-anlamak-290f2152808f>, 2019.
- [3] Ece Akdağlı. Yolo algoritması nedir? <https://ece-akdagli.medium.com/yolo-algoritmasÄŒ-nedir-859862c7d5a6#:~:text=YOLO%20algoritmasÄŒ%20nasÄŒl%20ÄŒalÄŒsÄŒsr%3F,iÄŒin%20belirli%20bir%20koÅŒul%20yoktur.>, 2021.
- [4] Betül Aksu. Opencv nedir? <https://www.siberegitmen.com/opencv-nedir-neler-yapilabilir/>, 2022.
- [5] AziMez. Get number of pixels of each object in image. <https://stackoverflow.com/questions/74855384/get-number-of-pixels-of-each-object-in-image/74856225#74856225>, 2022.
- [6] bing. yapayzeka. <https://www.bing.com/images/create/recognizing-objects-and-finding-their-actual-size-/1-664e2694d0f84d9ea6437a3191f47728?darkschemeovr=1&FORM=GUH2CR/>.
- [7] Pouya Hallaj. Data Augmentation: Benefits and Disadvantages. <https://medium.com/@pouyahallaj/data-augmentation-benefits-and-disadvantages-38d8201aead/>, 2023. [Online; accessed 25-mayıs-2024].
- [8] Innat. Drawing bounding rectangles around multiple objects in binary image in python. <https://stackoverflow.com/questions/63923800/drawing-bounding-rectangles-around-multiple-objects-in-binary-image-in-python/>, 2022.
- [9] Serdar Korkut. Antropometri nedir? <https://www.serdarkorkut.com/2017/08/08/antropometri-nedir/>, 8 Ağustos 2017.

- [10] Sourav Kumar. Data Augmentation Increases Accuracy of your model. <https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722/>, 2022.
- [11] muratgölcan. Nesne Boyut. <https://medium.com/@muratgulcan/opencv-kullanarak-görüntüleri-temel-işlemler-da90b59ba2ef/>, 2023. [Online; accessed 03-nisan-2024].
- [12] Ultralytics. Ultralytics yolo dokümanlar. <https://docs.ultralytics.com/tr/models/yolov8/>, 2024.
- [13] wikipedia. wikipedia. [https://tr.wikipedia.org/wiki/Python_Imaging_Library#:~:text=Python%20Imaging%20Library%20\(k%C4%B1saca%20PIL,bir%20grafik%20i%C5%9Flene%20imk%C3%A2n%C4%B1%20sunar./](https://tr.wikipedia.org/wiki/Python_Imaging_Library#:~:text=Python%20Imaging%20Library%20(k%C4%B1saca%20PIL,bir%20grafik%20i%C5%9Flene%20imk%C3%A2n%C4%B1%20sunar./).
- [14] wikipedia. Data augmentation. https://en.wikipedia.org/wiki/Data_augmentation/, 2022. [Online; accessed 25-mayıs-2024].
- [15] yentur. Görüntü segmentasyonu. <https://medium.com/move-on-ai/g%C3%B6r%C3%BCnt%C3%BCsegmentasyonu-ba193b168a1b>, 2021.
- [16] Özgür kan. Roboflow İle veri seti hazırlama. https://www.youtube.com/watch?v=lz_MITUBie8/, 2023.