



# Görüntü İşleme Tabanlı Üretim Bandından Akan Ürün Tespiti Sayımı ve Takibi

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Fatih BABAGİRAY

Y220240190

Proje Danışmanı: Prof. Dr. Femin Yalçın KÜÇÜKBAYRAK

2024

# Görüntü İşleme Tabanlı Üretim Bandından Akan Ürün Tespiti Sayımı ve Takibi

## ÖZ

Bu çalışma, YOLOv7 modelinin, özel olarak hazırlanmış Cıvata ve Somun Veri Seti ile eğitildiğinde gösterdiği performansı değerlendirmektedir. Çalışmanın temel amacı, modelin üretim ve kalite kontrolündeki çeşitli otomasyon süreçleri için kritik olan endüstriyel bileşenlerin tespiti ve sayımındaki doğruluk ve verimliliğini değerlendirmektir. Veri seti, özellikle somun ve cıvata olmak üzere iki sınıfa ayrılmış 7.200 eğitim görüntüsü ve 1.800 doğrulama görüntüsünden oluşmaktadır. Bulgular, YOLOv7'nin karmaşık ve yoğun sahnelerde bile nesnelere etkili bir şekilde tespit edip sayarak yüksek hassasiyet ve doğruluk oranları sergilediğini göstermektedir. Sonuçlar, modelin dayanıklılığını ve daha da geliştirilme potansiyelini vurgulamakta olup, endüstriyel uygulamalar için uygun bir aday olduğunu ortaya koymaktadır. Bu çalışma, derin öğrenme modellerinin endüstriyel otomasyonda uygulanmasına yönelik artan araştırma literatürüne katkıda bulunarak, YOLOv7'nin gerçek dünya senaryolarında pratik kullanımını ve uyarlanabilirliğini vurgulamaktadır.

**Anahtar Sözcükler:** YOLOv7, Nesne Tespiti, Endüstriyel Otomasyon, Derin Öğrenme, Cıvata ve Somun Veri Seti

# Image Processing Based Production Line Flowing Product Detection, Counting and Tracking

## Abstract

This study evaluates the performance of the YOLOv7 model, a state-of-the-art object detection algorithm, when trained on a specialized Nuts and Bolts Dataset. The primary objective is to assess the model's accuracy and efficiency in detecting and counting industrial components, which is crucial for various automation processes in manufacturing and quality control. The dataset comprises 7,200 training images and 1,800 validation images, specifically categorized into two classes: nuts and bolts. The findings indicate that YOLOv7 demonstrates high precision and recall rates, effectively identifying and counting objects even in complex and densely populated scenes. The results highlight the model's robustness and its potential for further improvement, making it a suitable candidate for industrial applications where precise object detection and counting are essential. This study contributes to the growing body of research on applying deep learning models in industrial automation, underscoring the practical utility and adaptability of YOLOv7 in real-world scenarios.

**Keywords:** YOLOv7, Object Detection, Automation, Deep Learning, Nuts and Bolts Dataset

# Teşekkür

Çalışma konusunun belirlenmesinde ve çalışmanın hazırlanma sürecinin her aşamasında bilgilerini, tecrübelerini ve değerli zamanlarını esirgemeyerek bana her fırsatta yardımcı olan değerli hocam Sayın Prof. Dr. Femin Yalçın Küçükbayrak' a teşekkürü bir borç bilirim.

# İçindekiler

Öz.....	i
Abstract.....	ii
Teşekkür.....	iii
Şekiller Listesi .....	v
Tablolar Listesi .....	vi
Giriş .....	1
2. YOLO (You Only Look Once) ve COCO (Common Objects in Context) Etiket Formatları .....	3
2.1 YOLO (You Only Look Once) .....	4
2.2 YOLO Etiket Formatı .....	5
2.3 COCO (Common Objects in Context):.....	5
2.4 COCO Etiket Formatı .....	6
3. COCO Etiket Formatından YOLO Etiket Formatına Dönüşüm .....	7
4. Açık Kaynaklı Python Kütüphaneleri.....	10
5. Model Eğitimi .....	13
5.1. Veri Seti Hazırlığı ve Model Eğitimi.....	13
5.2. Model Eğitim Sonuçlarının İncelenmesi.....	15
5.2.1. Genel Eğitim Sonuçları.....	15
5.2.2. Recall (Hatırlama) Eğrisini Sonuçları.....	18
5.2.3. Precision (Kesinlik) Eğrisini Sonuçları.....	20
6. Tespit ve İzleme.....	23
7. Değerlendirme.....	28
Kaynaklar.....	29
Ekler.....	30

# Şekiller Listesi

Şekil 1	Etiket Formatları .....	4
Şekil 2	Dönüşüm Formülü.....	7
Şekil 3	Dönüşüm Kodu .....	8
Şekil 4	YOLOv7 modelinin orjinal coco.yaml dosyası.....	14
Şekil 5	Bu çalışmada kullanılan YOLOv7 modelinin coco.yaml dosyası .....	14
Şekil 6	YOLOv7 Modelinin Genel Eğitim Sonuçları .....	15
Şekil 7	Recall (Hatırlama) Eğrisini Sonuçları .....	18
Şekil 8	Precision (Kesinlik) Eğrisini Sonuçları.....	20
Şekil 9	Gerçek Etiketler.....	23
Şekil 10	Tahminler .....	24
Şekil 11	3 Somun Örneği .....	26
Şekil 12	2 Somun ve 1 Civata Örneği .....	27

# Tablolar Listesi

Tablo 1	Açık Kaynaklı Kütüphaneler.....	10
---------	---------------------------------	----

# Giriş

Somun ve cıvataların tespiti, imalat ve inşaatın bakım ve onarıma kadar çeşitli endüstriyel sektörlerde büyük önem taşımaktadır. Somun ve cıvatalar, makine, altyapı ve mekanik sistemlerin montajında temel bileşenler olarak görev yapar ve bunların doğru montajı ve hizalanması yapısal bütünlük, operasyonel verimlilik ve güvenliğin sağlanması için kritik öneme sahiptir. Somun ve cıvataların doğru tespiti, montaj spesifikasyonlarındaki sapmaların ürün kusurlarına veya arızalarına yol açabileceği üretim sırasında kalite kontrol süreçlerinde vazgeçilmezdir. Ayrıca, bakım ve onarım operasyonlarında, eksik veya gevşek somun ve cıvataları hızlı bir şekilde belirleme yeteneği, potansiyel tehlikeleri önlemek ve arıza süresini azaltmak için çok önemlidir. Ek olarak, otomotiv mühendisliği ve havacılık gibi hassasiyet ve güvenilirliğin çok önemli olduğu alanlarda, somun ve cıvataların tespiti, sıkı güvenlik standartlarının ve mevzuata uygunluğun korunmasında çok önemli bir rol oynamaktadır. Bu nedenle, somun ve cıvata tespitinin önemi sadece mekanik işlevselliğin ötesine geçerek hassasiyet, güvenilirlik ve güvenliğin çok önemli olduğu endüstriyel operasyonların her yönüne nüfuz eder [1].

Bilgisayarla görmede temel bir görev olan nesne algılama, otonom sürüş ve gözetimden tıbbi görüntüleme ve endüstriyel otomasyona kadar çeşitli uygulamalarda önemli bir rol oynamaktadır. Dijital görüntüler veya video kareleri içindeki nesnelere tanımlanmasını ve konumlandırılmasını içerir. Derin öğrenme metodolojilerindeki ilerleme, son on yılda nesne algılama algoritmalarının etkinliğini ve doğruluğunu önemli ölçüde artırmıştır. Bunlar arasında You Only Look Once (YOLO), gerçek zamanlı performansı ve yüksek doğruluğu ile tanınan önemli bir algoritmik çerçeve olarak öne çıkmaktadır.

YOLO modellerinin evrimi, nesne algılama alanında durmaksızın yenilik ve mükemmellik arayışının bir kanıtı olmuştur. Bu yolculuk, gerçek zamanlı olarak eşzamanlı nesne lokalizasyonu ve sınıflandırması yapabilen birleşik bir derin sinir ağı mimarisinin çığır açan konseptini ortaya koyan orijinal YOLO modelinin başlangıcıyla başladı. YOLOv2 ve YOLOv3 dahil olmak üzere sonraki yinelemeler, özellik piramidi ağları, çok ölçekli algılama



ve gelişmiş eğitim metodolojileri gibi ilerlemelerle desteklenen doğruluk, hız ve sağlamlık açısından önemli gelişmelere tanık oldu. YOLOv4, bag-of-freebies ve bag-of-specials gibi gelişmiş tekniklerin entegrasyonu ile bir dönüm noktası oldu ve nesne algılama performansının sınırlarını yeni zirvelere taşıdı. Bu temel üzerine inşa edilen YOLOv5, uygulamayı basitleştirirken hızı ve doğruluğu daha da optimize eden modern bir mimari ve yeni bir eğitim hattı sundu. YOLOv7, öncekilere kıyasla hem hız hem de doğrulukta yaptığı iyileştirmeler nedeniyle büyük ilgi görmüştür. Bu proje, özellikle YOLOv7'ye odaklanarak, mimarisini, eğitim sürecini ve performans değerlendirmesini aydınlatarak nesne algılama teknikleri hakkında kapsamlı bir çalışma sunmayı amaçlamaktadır.

Somun ve civataların nesne tespiti, görüntü veya video karelerindeki nesnelerin doğru konumlandırılmasını ve sınıflandırılmasını kolaylaştırmak için karmaşık süreçlerin iç içe geçtiği çok yönlü bir alanı oluşturur. Bu çabanın özünde, nesnelerin temel özelliklerinin daha sonraki analizleri mümkün kılan anlamlı tanımlayıcılara dönüştürüldüğü özelliklerin çıkarılması ve temsili yatmaktadır. Nesne algılama çerçevelerinin zirvesi olan YOLOv7, bu temel bileşenleri sorunsuz bir şekilde düzenlemek için titizlikle hazırlanmış sofistike bir mimariye sahiptir. Konvolüsyonel katmanların, havuzlama işlemlerinin ve özellik füzyon modüllerinin dikkatli bir şekilde birleştirilmesiyle YOLOv7, özelliklerin çıkarılmasını ve temsil edilmesini benzersiz bir verimlilik ve etkinlikle düzenler. Ayrıca, YOLOv7'nin mimarisi sadece özellik çıkarmanın ötesine geçerek sınırlayıcı kutu regresyonu ve sınıf tahmini için karmaşık mekanizmalar içerir. YOLOv7, sinir ağı yapısı içinde regresyon başlıklarını ve sınıflandırma başlıklarını sorunsuz bir şekilde entegre ederek, özellik çıkarma işleminden nihai tahmine kadar tüm nesne algılama işlem hattını uyumlu ve birleşik bir çerçeve içine alır. Böylece, araştırmacılar ve uygulayıcılar YOLOv7'nin iç işleyişine bakarak, olağanüstü performansının temelini oluşturan karmaşık mekanizmalar hakkında değerli bilgiler edinebilir ve böylece nesne algılama alanında daha fazla ilerlemenin önünü açabilirler [2].

Ayrıca, YOLOv7'nin inceliklerini anlamak, eğitim metodolojisinin kapsamlı bir şekilde araştırılmasını gerektirir. YOLOv7'nin eğitimi, karmaşık hedef fonksiyonlarının optimize edilmesini, ağ parametrelerinin ince ayarının yapılmasını ve model genellemesi için kapsamlı veri kümelerinden yararlanılmasını içerir. Eğitim sürecinin ayrıntılı bir analizi, nesne algılama

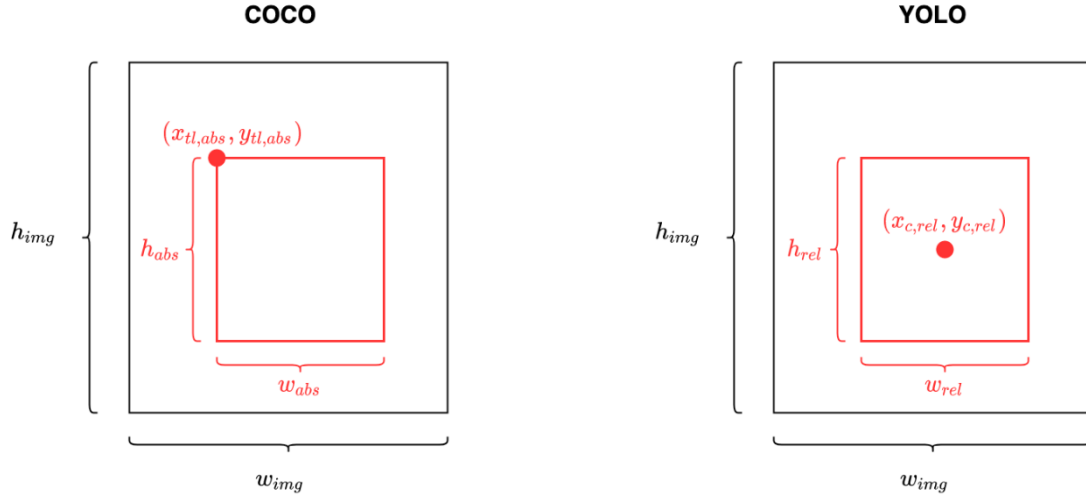
görevleri için derin sinir ağlarının eğitiminde karşılaşılan zorlukları ve nüansları ortaya koymaktadır.

YOLOv7'nin performansının değerlendirilmesi, yerleşik ölçütlere karşı titiz bir kıyaslama ve en son teknoloji yaklaşımlarla karşılaştırmayı gerektirir. Kıyaslama veri kümeleri ve gerçek dünya senaryoları üzerinde yapılan ampirik değerlendirmeler sayesinde araştırmacılar YOLOv7'nin etkinliğini, sağlamlığını ve ölçeklenebilirliğini değerlendirebilir ve böylece bu teknolojinin çeşitli uygulamalar için benimsenmesinde bilinçli karar vermeyi kolaylaştırabilir.

Bu proje, çalışmanın temel taşı olarak YOLOv7 çerçevesine odaklanarak nesne algılamanın inceliklerini araştırmaya çalışmaktadır. Mimarisini inceleyerek, eğitim sürecini araştırarak ve performansını değerlendirerek. Bu çalışma, nesne algılama metodolojilerinin kolektif anlayışına katkıda bulunmayı ve bu dinamik alanda daha fazla ilerlemenin önünü açmayı amaçlamaktadır.

## 2. YOLO (You Only Look Once) ve COCO (Common Objects in Context) Etiket Formatları

Günümüzde bilgisayarla görme ve nesne tanıma alanındaki teknolojik ilerlemeler, yapay zeka sistemlerinin nesnelere algılaması ve sınıflandırması konusunda önemli adımlar atılmasını sağlamıştır. Bu ilerlemelerin merkezinde, nesne tespiti çalışmalarında kullanılan algoritmalar ve veri kümeleri bulunmaktadır. Bu bölümde, nesne tespiti alanında sıkça kullanılan YOLO ve COCO etiketlerini detaylı bir şekilde ele alınacaktır.



Şekil 1: Etiket Formatları

## 2.1 YOLO (You Only Look Once)

YOLO, nesne tespiti için kullanılan önemli bir derin öğrenme algoritmasıdır. YOLO'nun temel amacı, bir görüntüyü tek bir adımda analiz etmek ve bu görüntüdeki nesnelere belirlemektir. Diğer geleneksel nesne tespiti yöntemlerinden farklı olarak, YOLO, bir nesneyi tespit etmek için birden fazla adıma ihtiyaç duymaz. Bunun yerine, bir evrişimli sinir ağı (convolutional neural network) kullanarak görüntüyü tek bir geçişte analiz eder ve nesnelerin konumlarını ve sınıflarını çıkarır [3].

YOLO'nun avantajlarından biri, hızlı çalışmasıdır. Diğer yöntemlerle karşılaştırıldığında, YOLO daha hızlı sonuçlar üretebilir çünkü görüntüyü tek bir geçişte işler. Ayrıca, YOLO'nun sınırlayıcı kutular (bounding boxes) üzerinden hassas tahminler yapabilme yeteneği de önemlidir. Bu özellik, nesnelerin konumlarını daha doğru bir şekilde belirlemeye yardımcı olur.

Ancak, YOLO'nun bazı zayıf yönleri de vardır. Özellikle, küçük nesnelere tespit etme konusunda diğer yöntemlere göre daha az başarılı olabilir. Ayrıca, karmaşık sahnelerde nesnelerin doğru bir şekilde sınıflandırılması konusunda da bazı zorluklar yaşanabilir.

## 2.2 YOLO Etiket Formatı

YOLO, bir nesnenin tespit edildiği durumda, bir sınırlayıcı kutu (bounding box) çevresine ve bu nesnenin sınıfına dair bilgileri içeren bir etiket formatı kullanır. Örneğin, bir nesnenin konumunu (x, y) koordinatları ile belirtir ve bu nesnenin sınıfını bir sayı ile temsil eder.

Eğer C sınıf sayısı ise, her bir nesne için bir sınıfın varlığına dair olasılıklar  $P_{\{obj\}}$  kullanılarak ifade edilir. Eğer bir nesne varsa  $P_{\{obj\}} = 1$  olur, aksi halde  $P_{\{obj\}} = 0$  olur. Ayrıca, nesnenin sınırlayıcı kutusu, görüntünün boyutlarına göre normalleştirilmiş koordinatlar ile ifade edilir.

Aşağıda YOLO etiket formatının matematiksel temsilini gösteren bir formül bulunmaktadır:

$$Etiket = [P_{\{obj\}}, x, y, w, h, C_1, C_2, \dots, C_C]$$

*Burada*

$P_{\{obj\}}$ : Nesnenin varlığına dair olasılık (0 veya 1)

$x, y$ : Sınırlayıcı kutunun merkez koordinatları (normalleştirilmiş)

$w, h$ : Sınırlayıcı kutunun genişlik ve yükseklik (normalleştirilmiş)

$C_i$ : Her bir sınıf için olasılık değerleri (0 ile 1 arasında)

## 2.3 COCO (Common Objects in Context):

COCO, nesne tanıma sistemlerinin eğitilmesi ve değerlendirilmesi için kullanılan bir veri kümesidir. COCO veri kümesi, geniş bir nesne çeşitliliği içerir ve çeşitli zorluk seviyelerinde görüntüler içerir. Her bir görüntüde birden fazla nesne bulunabilir ve bu nesnelere farklı kategorilere ayrılabilir.

COCO veri kümesi, nesne tespiti algoritmalarını eğitmek için önemli bir kaynak olarak kabul edilir. Çeşitli nesne kategorilerini içerdiği için, bu veri kümesi nesne tespiti modellerinin geliştirilebilirliğini artırabilir. Ayrıca, COCO veri kümesi, nesne tespiti algoritmalarının

performansını deęerlendirmek için de sıkça kullanılır. Özellikle, nesne tespiti modellerinin doęruluęunu ve hızını test etmek için standart bir ölçüm aracı olarak kullanılabilir.

Ancak, COCO veri kümesinin bazı sınırlamaları da vardır. Özellikle, belirli nesne kategorilerinin dięerlerine göre daha fazla temsil edilmesi gibi dengesizlikler olabilir. Ayrıca, COCO veri kümesi, bazı özel uygulama alanları için özel olarak hazırlanmamış olabilir.

Genel anlamda, YOLO ve COCO, nesne tespiti çalışmalarında önemli araçlardır. YOLO, hızlı ve etkili bir nesne tespiti algoritması sağlarken, COCO veri kümesi nesne tanıma sistemlerini eğitmek ve deęerlendirmek için önemli bir kaynak olarak hizmet eder. Bu araçlar, yapay zeka sistemlerinin nesnelere algılaması ve anlaması konusunda gelecekteki ilerlemelerin temelini oluşturabilir.

#### 2.4 COCO Etiket Formatı

COCO, her bir nesnenin sınıfını ve konumunu belirtmek için daha ayrıntılı bir etiket formatı kullanır. Her nesne için bir sınıf etiketi ve sınırlayıcı kutunun koordinatları sağlanır.

Aşağıda COCO etiket formatının matematiksel temsilini gösteren bir formül bulunmaktadır:

$$\text{Etiket} = (x_{\min}, y_{\min}, x_{\max}, y_{\max}, C)$$

*Burada*

$(x_{\min}, y_{\min})$ : Sınırlayıcı kutunun sol üst köşe koordinatları

$(x_{\max}, y_{\min})$  Sınırlayıcı kutunun sağ alt köşe koordinatları

$C$ : Nesnenin sınıfı

$$X_{yolo} = \left( X_{coco} + \frac{W_{coco}}{2} \right) / W_{img}$$

$$Y_{yolo} = \left( Y_{coco} + \frac{h_{coco}}{2} \right) / h_{img}$$

$$W_{yolo} = W_{coco} / W_{img}$$

$$h_{yolo} = h_{coco} / h$$

Şekil 2: Dönüşüm Formülü

### 3. COCO Etiket Formatından YOLO Etiket Formatına Dönüşüm

COCO formatında olan görüntü etiketlerinin YOLO modellerinde uyumsuzluk olmadan çalışabilmesi için bir dönüşüm işlemi yapılmıştır. COCO formatından YOLO formatına dönüşüm için aşağıdaki Python kodu kullanılmıştır.

```
def convert_coco_to_yolo(coco_json_file, yolo_txt_dir):

    with open(coco_json_file, 'r') as f:
        coco = json.load(f)

    if not os.path.exists(yolo_txt_dir):
        os.makedirs(yolo_txt_dir)

    for i, image in enumerate(coco['images']):
        image_id = image['id']
        file_name = image['file_name']

        yolo_txt_file = os.path.join(yolo_txt_dir, f"{str(i).zfill(4)}.txt")
        with open(yolo_txt_file, 'w') as f:
            annotations = [ann for ann in coco['annotations'] if ann['image_id'] == image_id]

            for ann in annotations:
                x, y, w, h = ann['bbox']
                category_id = ann['category_id']

                if category_id == 1:
                    category_id = 0
                elif category_id == 2:
                    category_id = 1

                x_center = (x + (w / 2)) / image['width']
                y_center = (y + (h / 2)) / image['height']
                width = w / image['width']
                height = h / image['height']
                line = f"{category_id} {x_center:.6f} {y_center:.6f} {width:.6f} {height:.6f}"
                f.write(line + '\n')

if __name__ == '__main__':

    coco_json_file = 'COCO JSON FILE PATH'
    yolo_txt_dir = 'YOLO FOLDER PATH TO STORE .txt FILES'

    convert_coco_to_yolo(coco_json_file, yolo_txt_dir)
```

Şekil 3: Dönüşüm Kodu

a. convert\_coco\_to\_yolo Fonksiyonu:

- Bu kod, COCO JSON formatındaki verileri YOLO formatına dönüştürmek için tasarlanmıştır.
- coco\_json\_file parametresi, dönüştürülecek COCO JSON dosyasının yolunu temsil eder.
- yolo\_txt\_dir parametresi ise YOLO formatındaki çıktıların saklanacağı dizini belirtir.

b. COCO JSON Dosyasının Okunması:

- with ifadesi kullanılarak COCO JSON dosyası açılır ve içeriği json.load() yöntemi ile bir Python sözlüğü olarak okunur.

c. Çıktı Dizininin Kontrolü:

- İlk olarak, belirtilen yolo\_txt\_dir dizininin var olup olmadığı kontrol edilir. Eğer dizin yoksa, os.makedirs() fonksiyonu kullanılarak oluşturulur.

d. Her Bir Görüntü İçin Döngü:

- COCO JSON dosyasındaki 'images' kısmındaki her bir görüntü için bir döngü başlatılır.
- Döngü her bir görüntünün image\_id ve file\_name bilgilerini alır.

e. Etiketlerin Dönüştürülmesi ve Yazılması:

- Her bir görüntü için ayrı bir YOLO formatında metin dosyası oluşturulur.
- Görüntüye ait etiketler, COCO JSON dosyasındaki 'annotations' kısmından alınır.
- Her bir etiket, YOLO formatına dönüştürülerek ilgili YOLO metin dosyasına yazılır.
- COCO'da belirtilen sınıf indeksleri YOLO için uygun hale getirilir. Örneğin, COCO'da sınıflar 1'den başlar ancak YOLO'da sıfırdan başlar. Bu nedenle, COCO sınıf indeksi 1, YOLO için sınıf indeksi 0 olarak değiştirilir.
- Nesne koordinatları, görüntünün genişliği ve yüksekliği ile normalleştirilir.

f. Ana İşlev:

- Ana işlev, coco\_json\_file ve yolo\_txt\_dir parametreleriyle çağrılır.



- Bu parametrelerle `convert_coco_to_yolo` işlevi çağrılarak COCO JSON dosyası YOLO formatına dönüştürülür.

#### 4. Açık Kaynaklı Python Kütüphaneleri

Python, geniş bir kullanıcı kitlesi ve zengin bir ekosistemi ile modern programlama dünyasında önemli bir konuma sahiptir. Özellikle bilimsel hesaplamalardan yapay zeka uygulamalarına, görüntü işlemeden büyük veri analizine kadar birçok alanda Python kütüphaneleri yaygın bir şekilde kullanılmaktadır. Bu kütüphaneler, hem akademisyenlerin hem de endüstri profesyonellerinin karmaşık problemleri çözmek ve veriyle ilgili soruları yanıtlamak için güçlü araçlar sunar. Veri bilimi ve yapay zeka gibi disiplinlerde, Python'un sunduğu kütüphaneler sayesinde karmaşık modellerin geliştirilmesi ve eğitilmesi daha erişilebilir hale gelmiştir.

Bu proje özelinde ise, model eğitimi için aşağıdaki kütüphaner kullanılmıştır.

Tablo 1: Açık Kaynaklı Kütüphaneler

<b>Python Kütüphanelerinin Versiyonları</b>
<code>matplotlib</code> ≥3.2.2
<code>numpy</code> ≥1.18.5, <1.24.0
<code>opencv</code> ≥4.1.1
<code>Pillow</code> ≥7.1.2
<code>PyYAML</code> ≥5.3.1
<code>requests</code> ≥2.23.0
<code>scipy</code> ≥1.4.1
<code>torch</code> ≥1.7.0, !=1.12.0
<code>torchvision</code> ≥0.8.1, !=0.13.0
<code>tqdm</code> ≥4.41.0
<code>protobuf</code> <4.21.3

tensorboard >=2.4.1
pandas >=1.1.4
seaborn >=0.11.0

1. matplotlib >= 3.2.2: Veri görselleştirme için kullanılan bir Python kütüphanesidir. Grafik çizme, veri görselleştirme, 2D çizimler yapma gibi işlemleri destekler.

2. numpy >=1.18.5,<1.24.0: Bilimsel hesaplamalar için kullanılan bir kütüphanedir. Diziler üzerinde hızlı matematiksel işlemler yapmak için kullanılır.

3. opencv >=4.1.1: Görüntü işleme ve makine öğrenimi uygulamalarında sıkça kullanılan açık kaynaklı bir kütüphanedir. Görüntü işleme, nesne tespiti, yüz tanıma gibi birçok işlemi gerçekleştirebilir.

4. Pillow >=7.1.2: Python için bir görüntü işleme kütüphanesidir. Görüntüleri açma, kaydetme, dönüştürme ve basit görüntü manipülasyon işlemleri için kullanılır.

5. PyYAML >=5.3.1: YAML formatındaki verileri işlemek için kullanılan bir kütüphanedir. YAML dosyalarını okuma, yazma ve işleme yeteneklerine sahiptir.

6. requests >=2.23.0: HTTP istekleri göndermek ve almak için kullanılan bir kütüphanedir. Web API'lerine erişim, web sayfalarını indirme gibi işlemleri gerçekleştirmek için kullanılır.

7. scipy >=1.4.1: Bilimsel ve mühendislik hesaplamaları için kullanılan bir kütüphanedir. İntegral hesaplama, diferansiyel denklemler çözme, lineer cebir gibi işlemleri destekler.

8. torch >=1.7.0,! =1.12.0: PyTorch kütüphanesinin Python API'si için bir versiyon kontrolüdür. Derin öğrenme modelleri oluşturmak ve eğitmek için kullanılır.

9. torchvision >=0.8.1,! =0.13.0: PyTorch için bir görüntü işleme kütüphanesidir. Derin öğrenme modeli eğitiminde sıkça kullanılan görüntü veri kümesi işlemleri ve önceden eğitilmiş model indirme işlemlerini destekler.

10. tqdm >=4.41.0: İlerleme çubuğu (progress bar) sağlayan bir kütüphanedir. Döngülerin ilerleme durumunu görselleştirmek için kullanılır.

11. protobuf <4.21.3Google tarafından geliştirilen veri serileştirme formatını destekleyen bir kütüphanedir. Büyük veri setlerini etkili bir şekilde saklamak ve aktarmak için kullanılır.

12. tensorboard >=2.4.1TensorFlow tarafından sağlanan bir görselleştirme aracıdır. Derin öğrenme modeli eğitimi sırasında model performansını izlemek için kullanılır.

13. pandas >=1.1.4: Veri analizi ve veri manipülasyonu için kullanılan bir kütüphanedir. Veri çerçeveleri üzerinde çalışma, veri okuma/yazma işlemleri, veri temizleme ve dönüştürme işlemleri için kullanılır.

14. seaborn >=0.11.0: Veri görselleştirme için kullanılan bir Python kütüphanesidir. matplotlib'e dayanır ve daha çekici ve bilgilendirici grafikler oluşturmak için kullanılır.

## 5. Model Eğitimi

### 5.1. Veri Seti Hazırlığı ve Model Eğitimi

Orijinal YOLOv7 makalesinde, 80 sınıftan oluşan toplamda 118.287 eğitim görüntüsü ve 5000 doğrulama görüntüsü kullanılmıştır. Bu çalışmada ise, 2 sınıftan oluşan 7200 eğitim görüntüsü ve 1800 doğrulama görüntüsü kullanılmıştır. Şekil 4, orijinal YOLOv7 makalesinde kullanılan 80 sınıfı gösteren coco.yaml dosyasını göstermektedir. Ancak, bu çalışmada YOLOv7'yi eğitmek için coco.yaml dosyasının Şekil 5'deki gibi güncellenmesi gerekmektedir.

Orijinal makalede kullanılan coco.yaml dosyasında, YOLOv7 modelinin eğitimi için COCO veri setindeki tüm 80 sınıfın listesi yer almaktadır. Bu dosyada her bir sınıfın ismi ve sıralaması belirtilmiştir. Ancak, bu çalışmada sadece 2 sınıf üzerinde eğitim yapılacağı için, coco.yaml dosyasının da bu sınıfları yansıtacak şekilde güncellenmesi önemlidir.

Güncellenmiş coco.yaml dosyasında, sadece bu çalışmada kullanılan iki sınıfın isimleri yer alacaktır. Bu sayede, model eğitim sürecinde sadece ilgili iki sınıfı tanıyacak ve diğer sınıflar dikkate alınmayacaktır. Güncellenmiş dosyada ayrıca, eğitim ve doğrulama görüntülerinin yolları ve diğer gerekli parametreler de belirtilmiştir.

Bu değişiklik, modelin daha az sınıf üzerinde odaklanmasını sağlayarak eğitim sürecini optimize eder ve iki sınıf için daha iyi bir performans elde edilmesine yardımcı olur. Dolayısıyla, coco.yaml dosyasının doğru bir şekilde güncellenmesi, modelin istenilen doğrultuda eğitilmesi için kritik bir adımdır.

```
coco.yaml - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
# COCO 2017 dataset http://cocodataset.org

# download command/URL (optional)
download: bash ./scripts/get_coco.sh

# train and val data as 1) directory: path/images/, 2) file: path/images.txt, or 3) list: [path1/images/, path2/images/]
train: ./coco/train2017.txt # 118287 images
val: ./coco/val2017.txt # 5000 images
test: ./coco/test-dev2017.txt # 20288 of 40670 images, submit to https://competitions.codalab.org/competitions/20794

# number of classes
nc: 80

# class names
names: [ 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light',
'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee',
'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard',
'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple',
'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch',
'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear',
'hair drier', 'toothbrush' ]
```

Şekil 4: YOLOv7 modelinin orjinal coco.yaml dosyası

```
train: ../data/train/images/
val: ../data/val/images/

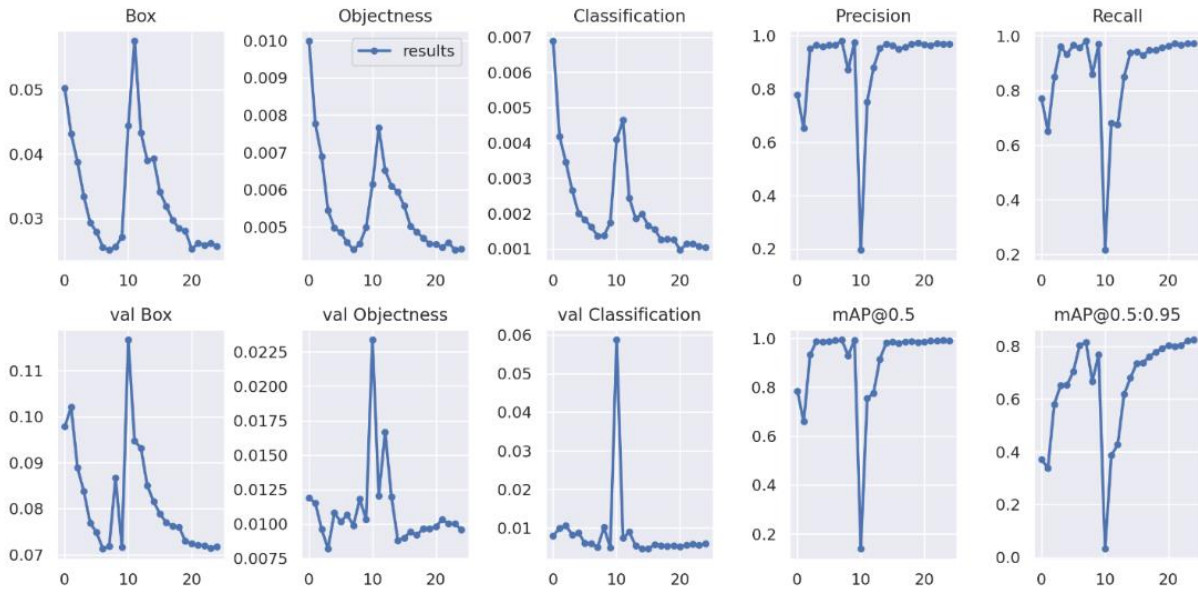
# number of classes
nc: 2

# class names
names: [ 'bolt', 'nut' ]
```

Şekil 5: Bu çalışmada kullanılan YOLOv7 modelinin coco.yaml dosyası

## 5.2. Model Eğitim Sonuçlarının İncelenmesi

### 5.2.1. Genel Eğitim Sonuçları



Şekil 6: YOLOv7 Modelinin Genel Eğitim Sonuçları

Şekil 6, YOLOv7 modelinin eğitim ve doğrulama süreçleri boyunca performansını değerlendiren çeşitli metriklerdeki değişimleri detaylandırmaktadır. Grafikler sırasıyla Box, Objectness, Classification, Precision, Recall, mAP@0.5 ve mAP@0.5:0.95 metriklerini içermektedir. Her bir metriğin analizi, modelin doğruluğu, tespiti ve genel performansı hakkında derinlemesine bilgi sağlamaktadır.

#### 5.2.1.1 Box Loss (Eğitim ve Doğrulama):

Eğitim sürecindeki Box Loss grafiği, modelin nesne kutularını tahmin etme doğruluğunu göstermektedir. Eğitimin ilk aşamalarında yüksek olan hata oranları, zamanla azalmaktadır. Bu eğilim, modelin eğitim verisetindeki nesne konumlarını daha doğru bir şekilde öğrenip tahmin edebildiğini ortaya koymaktadır. Doğrulama sürecindeki Box Loss grafiği (val Box) de benzer bir trend izlemektedir; başlarda yüksek olan doğrulama hataları, eğitim ilerledikçe azalmaktadır. Ancak, doğrulama hatalarında gözlemlenen dalgalanmalar, modelin doğrulama veri setindeki çeşitli nesnelerin konumlarını tahmin etmede zaman zaman zorlandığını

göstermektedir. Sonuçta, doğrulama hataları da azalarak stabil hale gelmektedir, bu da modelin genel performansının iyileştiğini göstermektedir.

#### 5.2.1.2 Objectness Loss (Eğitim ve Doğrulama):

Objectness Loss grafiği, modelin bir nesnenin varlığını tespit etme yeteneğini ölçmektedir. Eğitim sürecinde hataların azaldığı gözlemlenmektedir, bu da modelin nesnelere daha güvenilir bir şekilde tespit edebildiğini göstermektedir. Doğrulama sürecindeki Objectness Loss grafiğinde (val Objectness) ise belirgin dalgalanmalar bulunmaktadır. Bu dalgalanmalar, modelin doğrulama veri setindeki nesnelere tespit etmede zaman zaman zorluk çektiğini işaret ederken, genel trendin azalma yönünde olması, modelin performansının genel olarak iyileştiğini göstermektedir.

#### 5.2.1.3 Classification Loss (Eğitim ve Doğrulama):

Classification Loss grafiği, modelin nesnelere doğru sınıflandırma yeteneğini ortaya koymaktadır. Eğitim sürecinde hatalar hızla azalmaktadır, bu da modelin sınıflandırma doğruluğunun önemli ölçüde arttığını göstermektedir. Doğrulama sürecindeki Classification Loss grafiği (val Classification) de benzer bir azalma trendi sergilemektedir, ancak doğrulama hatalarında gözlemlenen dalgalanmalar, modelin farklı sınıfları doğru bir şekilde ayırt etmede zaman zaman zorlandığını göstermektedir.

#### 5.2.1.4 Precision ve Recall (Doğrulama):

Precision grafiği, modelin doğru pozitif tahminlerinin tüm pozitif tahminlere oranını ölçmektedir. Başlangıçta bazı dalgalanmalar gözlemlense de, eğitim sürecinin ilerlemesiyle precision değeri istikrarlı bir şekilde artarak 1'e yaklaşmaktadır. Bu, modelin doğru tespit oranının yüksek olduğunu göstermektedir. Recall grafiği, modelin doğru pozitif tahminlerinin gerçek pozitiflere oranını gösterir. Recall değerinin de 1'e yaklaşması, modelin yüksek bir tespit oranına sahip olduğunu ve doğru nesnelere tespit etmede başarılı olduğunu ortaya koymaktadır.

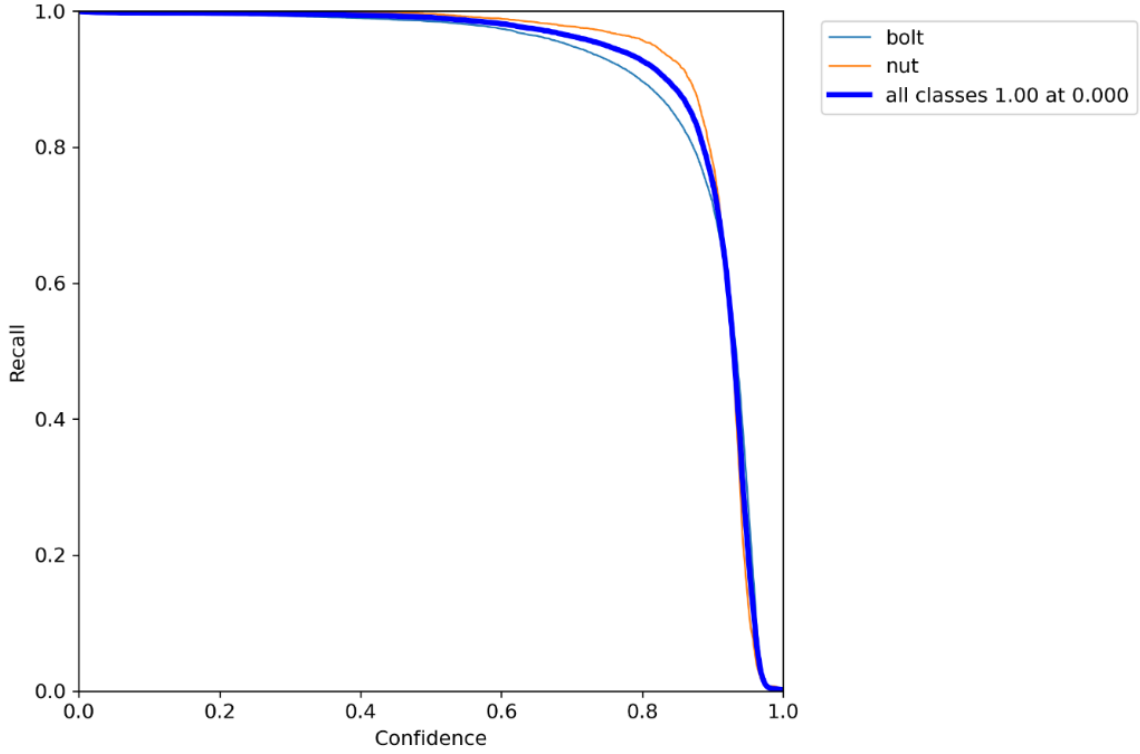
#### 5.2.1.5 $mAP@0.5$ ve $mAP@0.5:0.95$ (Doğrulama):

$mAP$  (mean Average Precision) metrikleri, modelin genel doğruluk ve tespit performansını değerlendirmektedir.  $mAP@0.5$ , Intersection over Union (IoU) eşiği 0.5 iken hesaplanırken,  $mAP@0.5:0.95$ , farklı IoU eşiklerinde (0.5'ten 0.95'e kadar) ortalama doğruluğu gösterir. Her iki metrikin de grafiklerinde, eğitim sürecinin başlarında belirgin dalgalanmalar gözlemlenmektedir. Ancak, eğitim ilerledikçe bu değerler artmakta ve stabil hale gelmektedir. Bu, modelin genel performansının önemli ölçüde iyileştiğini ve yüksek doğrulukta nesne tespiti yapabildiğini göstermektedir.

Genel sonuçlara bakıldığında bu grafikler, YOLOv7 modelinin eğitim sürecinde başarılı bir şekilde öğrenme gerçekleştirdiğini ve doğrulama setinde de yüksek bir performans sergilediğini göstermektedir. Hata oranlarının azalması ve  $mAP$  değerlerinin yükselmesi, modelin hem eğitim hem de doğrulama verilerinde üstün bir performans sergilediğini ve çeşitli nesne tespit ve sınıflandırma görevlerinde etkili olduğunu kanıtlamaktadır. Bu bulgular, modelin gerçek dünya uygulamalarında güvenilir ve doğru bir şekilde kullanılabileceğini göstermektedir.



### 5.2.2. Recall (Hatırlama) Eğrisini Sonuçları



Şekil 7: Recall (Hatırlama) Eğrisini Sonuçları

Şekil 7, YOLOv7 modelinin farklı güven düzeylerinde nesnelere ne kadar iyi tespit ettiğini gösteren Recall (hatırlama) eğrisini sunmaktadır. Bu grafik, modelin güven düzeyi değiştiğinde doğruluk ve tespit performansını değerlendirmek için kritik öneme sahiptir. Grafik, "bolt" ve "nut" olmak üzere iki sınıfın yanı sıra tüm sınıfların toplu performansını da içermektedir.

Bu eğri, modelin farklı güven seviyelerinde nesnelere ne kadar başarılı bir şekilde tespit ettiğini göstermektedir. X eksenini, modelin tespitlerinin güven seviyesini (Confidence), Y eksenini ise hatırlama oranını (Recall) temsil etmektedir. Hatırlama oranı, modelin doğru tespitlerinin gerçek pozitiflere oranını ifade eder ve 1'e yaklaşması, modelin tüm gerçek nesnelere yüksek doğrulukla tespit ettiğini gösterir.

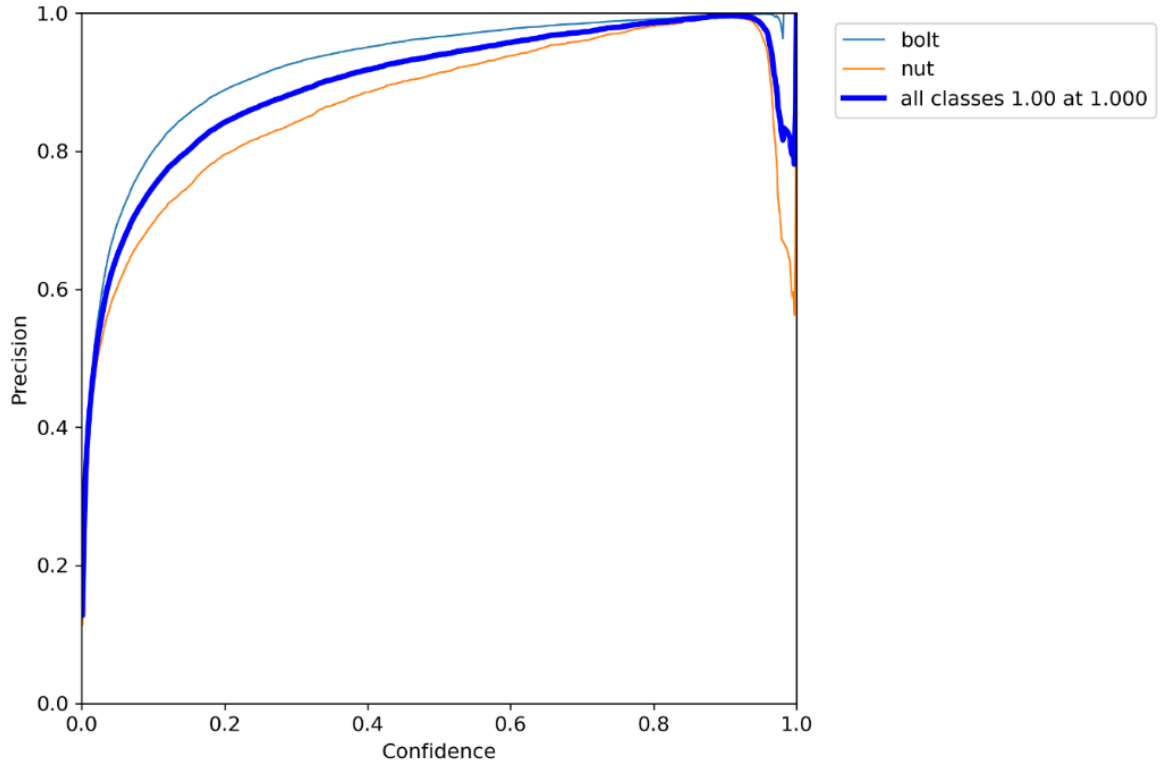
Grafikte, her iki sınıf ("bolt" ve "nut") için hatırlama oranlarının oldukça yüksek olduğu görülmektedir. Başlangıçta her iki sınıf için de hatırlama oranı 1'e çok yakın değerlerdedir, bu da modelin düşük güven seviyelerinde bile neredeyse tüm nesnelere tespit edebildiğini göstermektedir. Güven seviyesi arttıkça, hatırlama oranlarında hafif bir azalma görülmekte, ancak bu azalma oldukça yavaş ve istikrarlıdır. Güven seviyesi yaklaşık 0.8'e ulaştığında, hatırlama oranlarında daha belirgin bir düşüş gözlemlenmektedir. Bu noktanın ötesinde, güven seviyesi daha da arttıkça hatırlama oranı hızla düşmektedir. Bu durum, modelin yüksek güven seviyelerinde daha az ama daha kesin tespitler yaptığını göstermektedir.

Grafikteki kalın mavi çizgi, tüm sınıfların toplu performansını temsil etmektedir. Bu çizgi de "bolt" ve "nut" sınıflarının eğrileriyle oldukça uyumludur, bu da modelin genel performansının her iki sınıfta da benzer olduğunu göstermektedir. Başka bir deyişle, model her iki sınıf için de tutarlı ve yüksek bir tespit performansı sergilemektedir.

Bu eğri, modelin doğruluk ve hatırlama dengesi hakkında önemli bilgiler sunmaktadır. Düşük güven seviyelerinde yüksek hatırlama oranları, modelin tüm potansiyel nesnelere tespit etmede başarılı olduğunu, ancak bazı yanlış pozitiflerin de olabileceğini gösterir. Yüksek güven seviyelerinde ise hatırlama oranının azalması, modelin daha az ama daha güvenilir tespitler yaptığını gösterir. Bu, modelin pratik uygulamalarda nasıl performans göstereceği konusunda önemli ipuçları sağlar. Örneğin, hatırlama oranının yüksek olması gereken durumlarda düşük güven seviyeleri tercih edilebilirken, kesin tespitlerin kritik olduğu durumlarda yüksek güven seviyeleri tercih edilebilir.

Sonuç olarak bu grafik, YOLOv7 modelinin farklı güven seviyelerinde nasıl performans gösterdiğini detaylı bir şekilde ortaya koymaktadır. Modelin hem "bolt" hem de "nut" sınıflarında tutarlı ve yüksek performans sergilediği, hatırlama oranlarının güven seviyesine bağlı olarak değiştiği ve genel olarak güçlü bir tespit yeteneği sunduğu görülmektedir. Bu bulgular, modelin gerçek dünya uygulamalarında farklı senaryolar için uygun güven seviyeleri seçilerek optimize edilebileceğini göstermektedir.

### 5.2.3. Precision (Kesinlik) Eğrisini Sonuçları



Şekil 8: Precision (Kesinlik) Eğrisini Sonuçları

Şekil 8, YOLOv7 modelinin farklı güven düzeylerinde nesnelere doğru bir şekilde tespit etme yeteneğini gösteren Precision (kesinlik) eğrisini sunmaktadır. Bu grafik, modelin doğruluk ve tespit performansını güven seviyesine göre değerlendirmek için kritik öneme sahiptir. Grafik, "bolt" ve "nut" olmak üzere iki sınıfın yanı sıra tüm sınıfların toplu performansını da içermektedir.

Bu eğri, modelin farklı güven seviyelerinde nesnelere ne kadar doğru bir şekilde tespit ettiğini göstermektedir. X eksenini, modelin tespitlerinin güven seviyesini (Confidence), Y eksenini ise kesinlik oranını (Precision) temsil etmektedir. Kesinlik oranı, modelin doğru pozitif tahminlerinin toplam pozitif tahminlere oranını ifade eder ve 1'e yaklaşması, modelin tespit ettiği tüm nesnelere gerçekten doğru olduğunu gösterir.

Grafikte, "bolt" ve "nut" sınıfları için kesinlik oranlarının güven seviyesi arttıkça önemli ölçüde iyileştiği görülmektedir. Başlangıçta her iki sınıf için de kesinlik oranları düşük olmakla birlikte, güven seviyesi arttıkça bu oranlar hızla yükselmektedir. Bu eğilim, modelin düşük güven seviyelerinde daha fazla yanlış pozitif tespit yaptığını, ancak güven seviyesi arttıkça doğru pozitif tespitlerin oranının arttığını göstermektedir. Güven seviyesi yaklaşık 0.8'e ulaştığında kesinlik oranları maksimum seviyeye yaklaşmakta ve bu noktadan sonra hafif dalgalanmalar gözlemlenmektedir. Özellikle "nut" sınıfı için yüksek güven seviyelerinde belirgin bir düşüş görülmektedir, bu da bu sınıfta yüksek güven seviyelerinde daha az ama daha kesin tespitler yapıldığını göstermektedir.

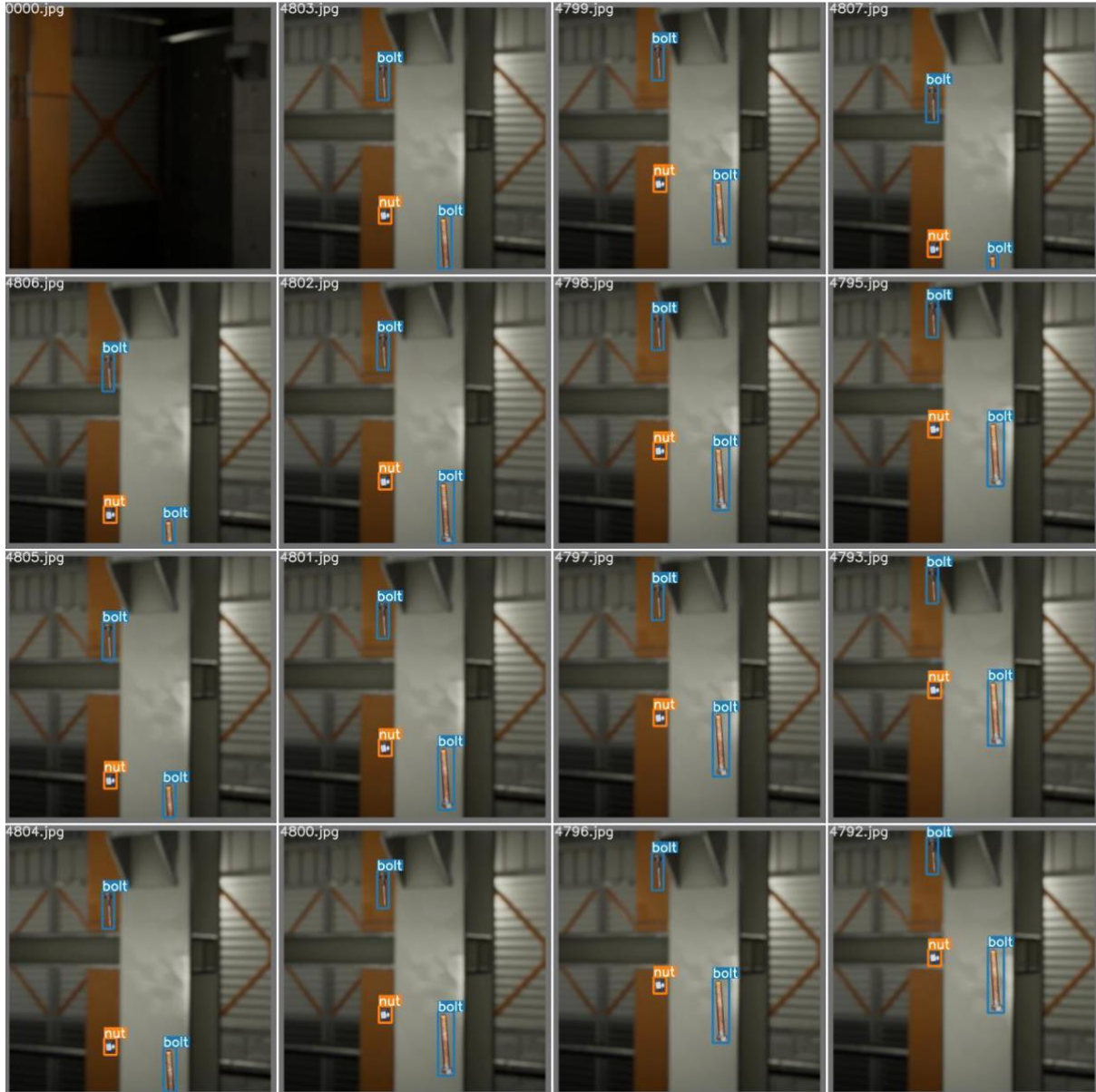
Grafikteki kalın mavi çizgi, tüm sınıfların toplu performansını temsil etmektedir. Bu çizgi, "bolt" ve "nut" sınıflarının eğrileriyle genel olarak uyumludur, ancak tüm sınıflar için ortalama kesinlik oranının biraz daha yüksek olduğu görülmektedir. Bu durum, modelin genel performansının her iki sınıfta da benzer olduğunu ve güven seviyesi arttıkça kesinlik oranlarının istikrarlı bir şekilde yükseldiğini göstermektedir.

Bu eğri, modelin doğruluk ve kesinlik dengesi hakkında önemli bilgiler sunmaktadır. Düşük güven seviyelerinde düşük kesinlik oranları, modelin yanlış pozitif tespit oranının yüksek olduğunu, dolayısıyla tespit edilen nesnelerin çoğunun yanlış olduğunu göstermektedir. Ancak, güven seviyesi arttıkça kesinlik oranlarının hızla yükselmesi, modelin daha doğru tespitler yaptığını ortaya koymaktadır. Yüksek güven seviyelerinde kesinlik oranının artması, modelin yalnızca yüksek doğrulukta tespitler yaptığını ve yanlış pozitif oranının minimuma indiğini göstermektedir.

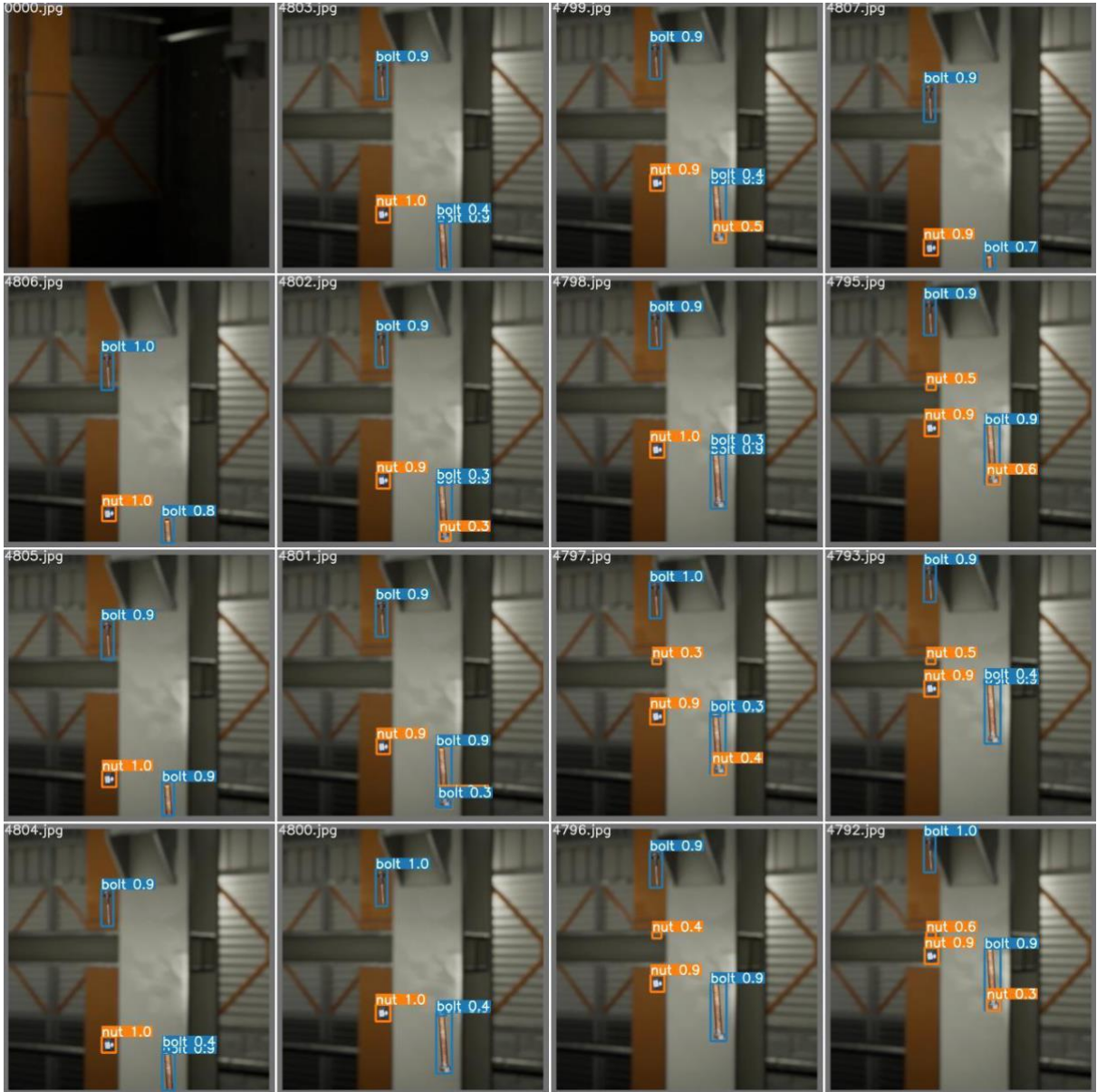
Kısaca bu grafik, YOLOv7 modelinin farklı güven seviyelerinde nasıl performans gösterdiğini detaylı bir şekilde ortaya koymaktadır. Modelin hem "bolt" hem de "nut" sınıflarında tutarlı ve yüksek performans sergilediği, kesinlik oranlarının güven seviyesine bağlı olarak önemli ölçüde iyileştiği ve genel olarak güçlü bir tespit yeteneği sunduğu görülmektedir. Bu bulgular,

modelin gerek dnya uygulamalarında farklı senaryolar için uygun gven seviyeleri seilerek optimize edilebileceđini gstermektedir. Modelin dřk gven seviyelerinde geniř kapsamlı ancak dřk dođrulukta tespitler yaparken, yksek gven seviyelerinde daha az ama daha dođru tespitler yaptıđı sonucuna varılabilir.

## 6. Tespit ve İzleme



Şekil 9: Gerçek Etiketler



Şekil 10: Tahminler

Şekil 9 ve Şekil 10, sırasıyla YOLOv7'nin ilk test grubundaki gerçek etiketleri ve tahminlerini göstermektedir. Bu grafikler, modelin performansını değerlendirmek ve hangi alanlarda iyileştirme gerektiğini belirlemek için kritik öneme sahiptir.

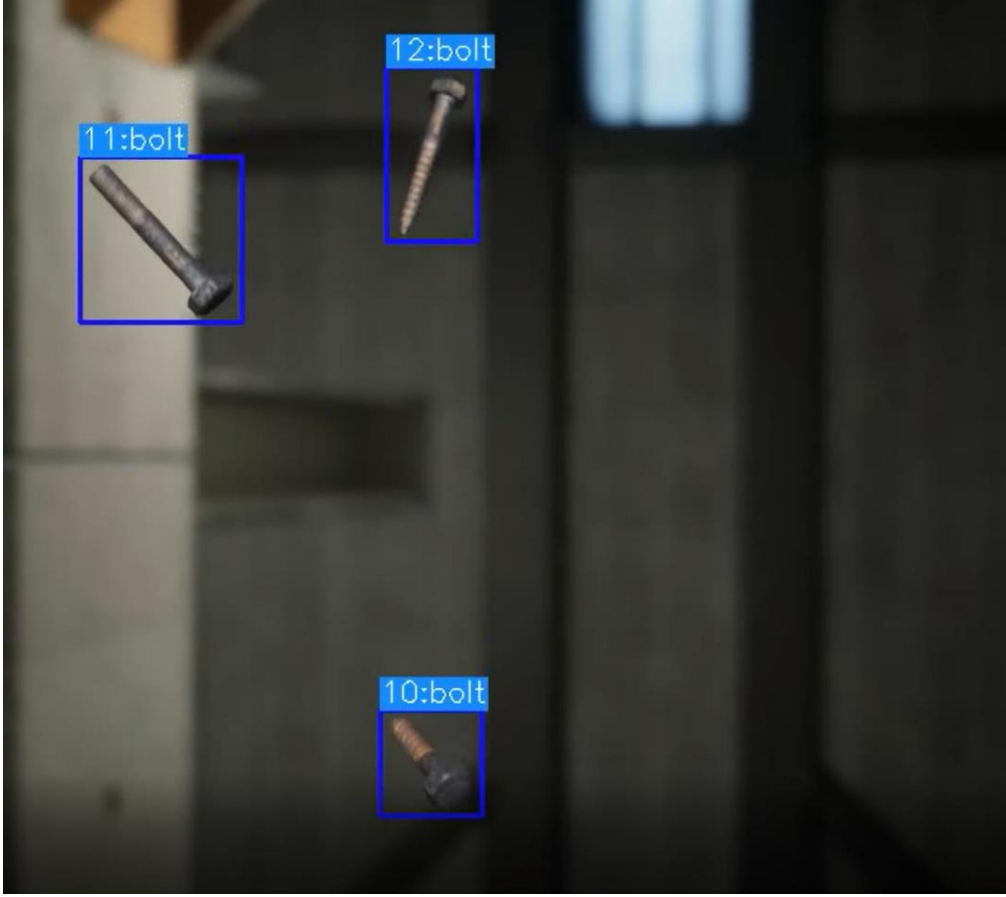
Şekil 9, test grubundaki verilerin gerçek etiketlerini sunmaktadır. Gerçek etiketler, her bir nesnenin doğru sınıfını ve konumunu gösteren referans noktalarını temsil eder. Bu, modelin

tahminlerinin ne kadar doğru olduğunu değerlendirmek için kullanılan temel veri setidir. Şekil 9'daki veriler, modelin hangi nesnelere tespit etmesi gerektiğini ve bu nesnelere tam olarak nerede bulunduğunu göstermektedir. Bu, modelin tahminlerini karşılaştırmak ve doğru tespitlerin yanı sıra yanlış pozitif ve negatifleri belirlemek için bir temel sağlar.

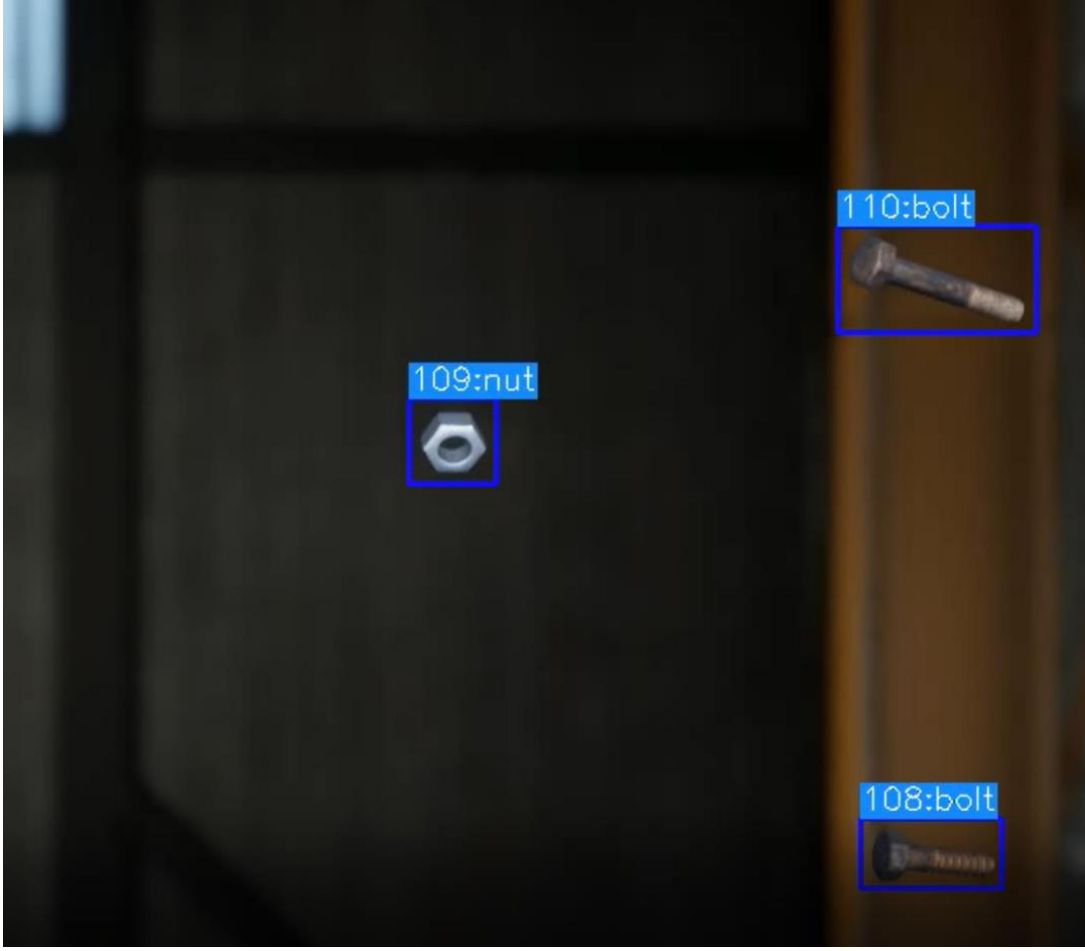
Şekil 10 ise YOLOv7 modelinin ilk test grubundaki tahminlerini göstermektedir. Tahminler, modelin test verilerinde hangi nesnelere tespit ettiğini ve bu nesnelere konumlarını nasıl belirlediğini gösterir. Şekil 10'daki tahminler, modelin performansını değerlendirmek için Şekil 9'daki gerçek etiketlerle karşılaştırılabilir. Bu karşılaştırma, modelin doğruluğunu, hassasiyetini ve hatırlama oranını ölçmek için kullanılır.

Gerçek etiketler ve tahminler arasındaki farklar, modelin hangi alanlarda güçlü olduğunu ve hangi alanlarda iyileştirilmesi gerektiğini belirlemek için analiz edilir. Örneğin, modelin belirli nesnelere sürekli olarak yanlış tespit etmesi, eğitim veri setinde bu nesnelere ilgili yetersiz veri bulunduğunu veya modelin bu nesnelere ayırt etmekte zorlandığını gösterebilir. Aynı şekilde, belirli nesnelere doğru tespit oranının yüksek olması, modelin bu nesnelere başarılı bir şekilde öğrendiğini ve tanıyabildiğini gösterir.





Şekil 11: 3 Somun Örneği



Şekil 12: 2 Somun ve 1 Civata Örneği

Şekil 11 ve Şekil 12, YOLOv7 modelinin Civata ve Somun Veri Seti ile eğitildikten sonra nesne sayma yeteneklerini göstermektedir. Bu şekiller, modelin farklı durumlarda nesnelere nasıl tespit edip sayabildiğini detaylı bir şekilde incelememizi sağlar. Şekil 11, modelin belirli bir görüntüdeki somunları tespit edip sayma yeteneğini göstermektedir. Şekil 11'de, modelin her bir somunu doğru bir şekilde tanımladığı ve bunların sayısını doğru bir şekilde belirlediği görülmektedir. Şekil 12 ise farklı bir senaryoda modelin nesne sayma yeteneğini ortaya koymaktadır. Bu şekil, daha karmaşık bir görüntüyü içermekte olup, modelin nesnelere birbirinden ayırma ve doğru sayma kabiliyetini test etmektedir. Şekil 12'de, modelin karmaşık sahnelerde bile yüksek doğrulukla nesnelere tespit edebildiği ve sayabildiği görülmektedir.

## 7. Değerlendirme

Bu çalışmanın tüm süreci değerlendirildiğinde, YOLOv7 modelinin iyi sonuçlar verdiği söylenebilir. Nesnel bir bakış açısıyla bakıldığında, YOLOv7 modelinin gelişime açık bir model olduğu görülmektedir. Bu, modelin hem mevcut performansının güçlü olduğunu hem de daha da iyileştirilebileceğini göstermektedir.

Özellikle Somun ve Cıvata Veri Seti kullanılarak eğitilen YOLOv7 modelinin endüstri için istenen özelliklere sahip olduğu tespit edilmiştir. Model, nesne tespiti ve sayma görevlerinde yüksek doğruluk ve hassasiyetle çalışarak, endüstriyel uygulamalarda etkin bir şekilde kullanılabilirliğini kanıtlamıştır. YOLOv7, çeşitli zorluk seviyelerine sahip sahnelerde başarılı performans sergileyerek, farklı ortamlarda ve koşullarda güvenilir bir çözüm sunmaktadır.

Modelin özellikle yoğun ve karmaşık sahnelerde bile yüksek doğrulukla çalışabilmesi, endüstriyel süreçlerin otomasyonunda önemli bir avantaj sağlamaktadır. Bu, üretim hatlarında, kalite kontrol süreçlerinde ve envanter yönetiminde verimliliği artırabilir. YOLOv7'nin güçlü nesne tespit ve sayma yetenekleri, işletmelerin hata oranlarını azaltmasına ve operasyonel verimliliği artırmasına olanak tanır.

Bütün projenin bir sonucu olarak, bu çalışma kapsamında elde edilen bulgular, YOLOv7 modelinin endüstriyel uygulamalar için uygun ve etkili bir araç olduğunu göstermektedir. Modelin mevcut performansı, gelişime açık yapısı ve endüstriyel ihtiyaçlara uygun özellikleri, YOLOv7'nin gelecekte daha geniş bir kullanım alanına sahip olabileceğini işaret etmektedir. Bu da, YOLOv7 modelinin endüstri standartlarına uygun, yenilikçi ve güçlü bir çözüm olarak değerlendirilebileceğini ortaya koymaktadır.

## Kaynaklar

[1] Y. Zhao, Z. Yang and C. Xu, "NPU-BOLT: A Dataset for Bolt Object Detection in Natural Scene Images," 2022.

[2] F. Mushtaq, K. Ramesh, S. Deshmukh, T. Ray, C. Parimi, P. Tandon and P. Kumar Jha, "Nuts&bolts: YOLO-v5 and image processing based component identification system," *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105665, 2023.

[3] C.-Y. Wang, A. Bochkovskiy ve H.-Y. M. Liao, «YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,» 2022.

Ekler

## PROJE KODLARI

- COCO JSON to YOLO Format ipynb dosyası
- Coco Yaml dosyası
- YOLOv7\_Detect\_Classify\_Count\_File ipynb dosyası
- YOLOv7\_Training\_File ipynb dosyası