**445T.C**

**KONYA TEKNİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BİLGİSAYAR MÜHENDİSLİĞİ UYGULAMASI – 1**

**(BİTİRME PROJESİ-1) FİNAL RAPOR FORMU**

|  |  |
| --- | --- |
| Öğrencinin Adı- Soyadı | YAHYA KARAMAN |
| Numarası: | 211220021 |
| Danışmanı Adı Soyadı: | BETÜL UZBAŞ |
| Sınav Tarihi: | 10/01/2025 |
| Projenin Konusu:  Görme Engelliler İçin Market Reyonunda Temel Besin Gıdalarını ve Para Birimlerini Tanıyan Uygulama | |
| **Dönem içi Yapılan Çalışmaların Özeti:**   1. Proje kapsamı ve hedefleri netleştirildi. 2. Kullanılacak derin öğrenme algoritmaları (YOLOv8) araştırıldı ve öğrenildi. 3. İki ayrı veri seti (meyve-sebze ve para birimleri) toplandı ve etiketleme işlemleri   gerçekleştirildi.   1. Fotoğraf etiketleme işlemleri için **Make Sense** kullanıldı, ardından veri setleri **Roboflow** ile veri artırma işlemlerine tabi tutuldu. 2. Eğitim-veri-doğrulama test oranı %80-%10-%10 olarak belirlendi. 3. YOLOv8n modeli Google Colab üzerinde eğitildi ve optimize edildi. 4. Elde edilen model dosyaları PyCharm üzerinde test edilerek, fotoğraf ve video üzerinden doğruluk sonuçları analiz edildi. | |
| **Projede Kullanılan Materyal Ve Metotlar:**   1. **Materyal listesi**   **1. Veritabanı ve Veri Seti**   * **Veri Seti (meyve-sebze):** Bu projede kullanılan veri seti, 'Fruits and Vegetables Image Recognition Dataset' adlı, meyve ve sebze görsellerini içeren bir veri kümesidir. Veri seti toplamda 2 GB boyutundadır ve 17 farklı sınıfa ait 3000 görsel bulunmaktadır. * **Veri Seti (TL banknot):** Bu projede kullanılan veri seti, 5 TL , 10 TL , 20 TL , 50 TL , 100 TL , 200 TL banknotların görsellerini içeren bir veri kümesidir.6 farklı sınıfa ait 900 görsel bulunmaktadır. * **Veri Seti Yapısı ve Bölünmesi:** Veri seti eğitim, doğrulama ve test işlemleri için üç ana gruba ayrılmıştır. Görsellerin %80'i eğitim, %10'u doğrulama, %10'u ise test için kullanılmaktadır. Bu bölünme, modelin genelleme yeteneğini ve doğruluğunu değerlendirmek için yapılmıştır. * **Veri Yapısı:** Her bir sınıf, ilgili sınıf adını taşıyan bir dizin altında organize edilmiştir. Görseller, çeşitli açılardan ve farklı aydınlatma koşullarında çekilmiş olup, her bir sınıfta ait olduğu meyve veya sebzeyi temsil eden çeşitli örnekler bulunmaktadır. Bu yapı, modelin farklı varyasyonlardaki nesneleri tanıma yeteneğini artırmaya yöneliktir.   **2. Kodlama Ortamı**   * **Python**: 3.8 sürümü, model eğitimi ve veri işleme işlemleri için ana programlama dili olarak kullanılmıştır. * **Geliştirme Ortamı**: PyCharm, modelin geliştirilmesi ve hata ayıklaması için tercih edilmiştir. * **Jupyter Notebook**: Model prototipinin geliştirilmesinde ve test edilmesinde kullanılan etkileşimli bir Python geliştirme ortamıdır.   **3. Paket Programlar ve Kütüphaneler**   * **YOLOv8:** YOLO modeli için Ultralytics kütüphanesi kullanıldı. * **OpenCV:** Görüntü işleme ve canlı video analizi. * **Matplotlib ve Seaborn:** Eğitim verilerinin görselleştirilmesi.     **4. Donanım**   * **Google Colab GPU**: Model eğitimi için kullanıldı. * **İşlemci**: Ryzen 7 5800H * **Bellek**: 16 GB RAM. * **Grafik İşlemci (GPU)**: NVIDIA RTX 3050 ti , TensorFlow GPU desteği ile modelin daha hızlı eğitimini sağlamak amacıyla kullanılmıştır. * **Cihaz**: Xiaomi Redmi 9T – Son kullanıcı tarafında modelin mobil kamera ile görüntü alması için kullanılmıştır.   **5. Model Eğitimi**   * YOLOv8n modeli her veri seti için ayrı ayrı eğitildi. * Eğitim sırasında hiperparametre optimizasyonu yapıldı. * Elde edilen model dosyaları .pt formatında kaydedildi. | |

Yapılan Çalışmaya ait Genel Bilgiler:

**Meyve-Sebze Modeli**

1. **Veri Seti Hazırlığı:**
   * Meyve ve sebzelere ait görüntüler toplandı ve etiketleme işlemi **Make Sense** platformu kullanılarak yapıldı.
   * Görseller, sınıflandırma işlemleri için **Roboflow** platformunda artırılarak veri setinin boyutu yaklaşık üç katına çıkarıldı.
   * Veri seti **%80 eğitim**, **%10 doğrulama** ve **%10 test** oranlarıyla üç alt gruba ayrıldı. Bu oranlar, modelin performansını doğru bir şekilde değerlendirmek için kullanıldı.
2. **Model Eğitimi:**
   * **YOLOv8n** modeli seçildi ve Google Colab kullanılarak eğitim süreci hızlandırıldı.
   * Eğitim sırasında **100 epoch** boyunca model eğitildi. Eğitim sonucunda elde edilen model, meyve ve sebze sınıflarını başarıyla tanıyabilir hale geldi.

üretmek, mahsul, sebze, doğal gıdalar, gıda grubu içeren bir resim

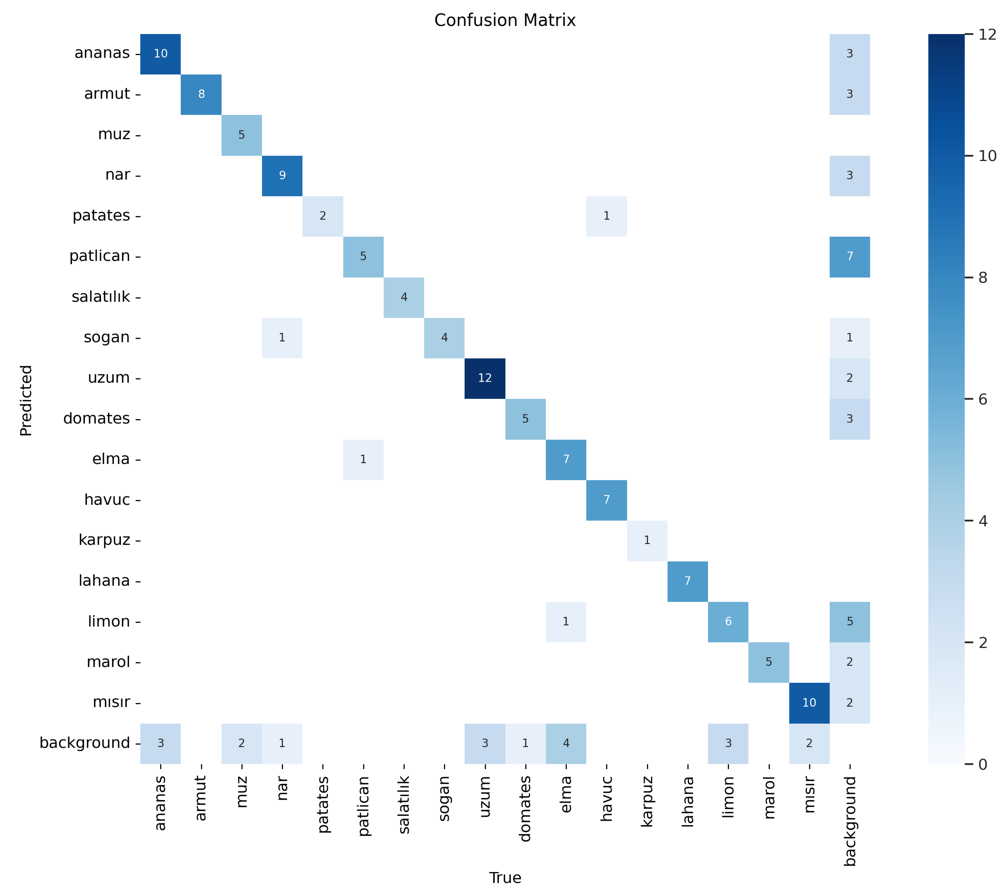
Açıklama otomatik olarak oluşturuldu

**val\_batch2\_pred Görseli:**

Modelin doğrulama setindeki resimlerde yaptığı tahminlerin sonuçlarını görselleştirir. Her bir görselde, modelin algıladığı nesneler üzerine yerleştirilen sınıf etiketleri ve tahmin doğruluk yüzdeleri bulunur. Bu, modelin performansını değerlendirmek için önemlidir, çünkü doğru ve yanlış tahminler görsel olarak analiz edilebilir.

1. **Konfüzyon Matrisi:**

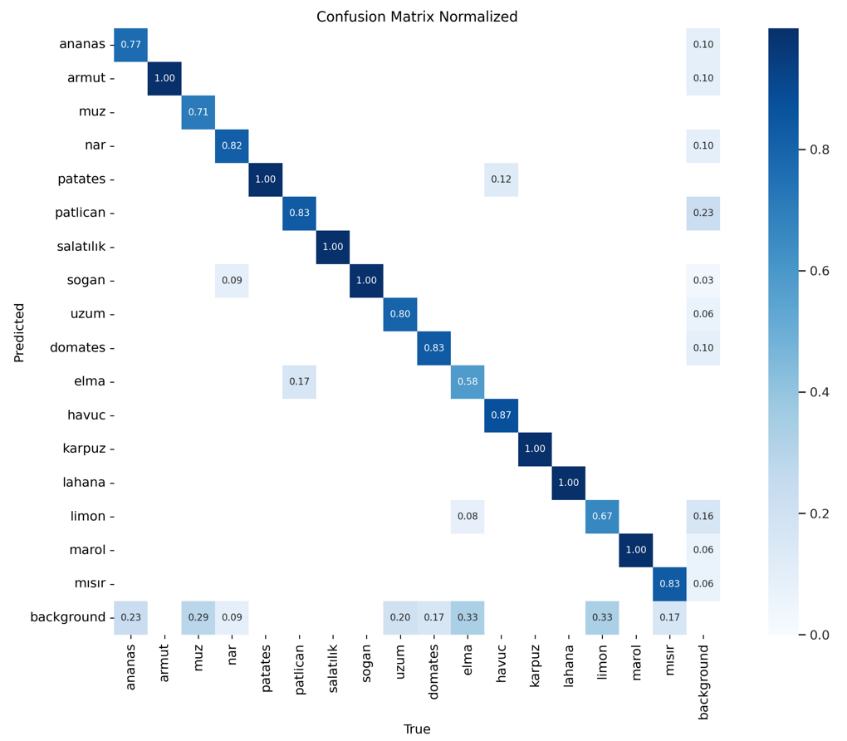
* Eğitim tamamlandıktan sonra modelin doğruluğunu değerlendirmek için konfüzyon matrisi oluşturuldu.
* Bu matriste, modelin doğru ve yanlış sınıflandırmaları görselleştirildi. Özellikle en sık karşılaşılan yanlış eşleştirmeler analiz edildi ve model performansını artırmak için bu bilgiler değerlendirildi.

****

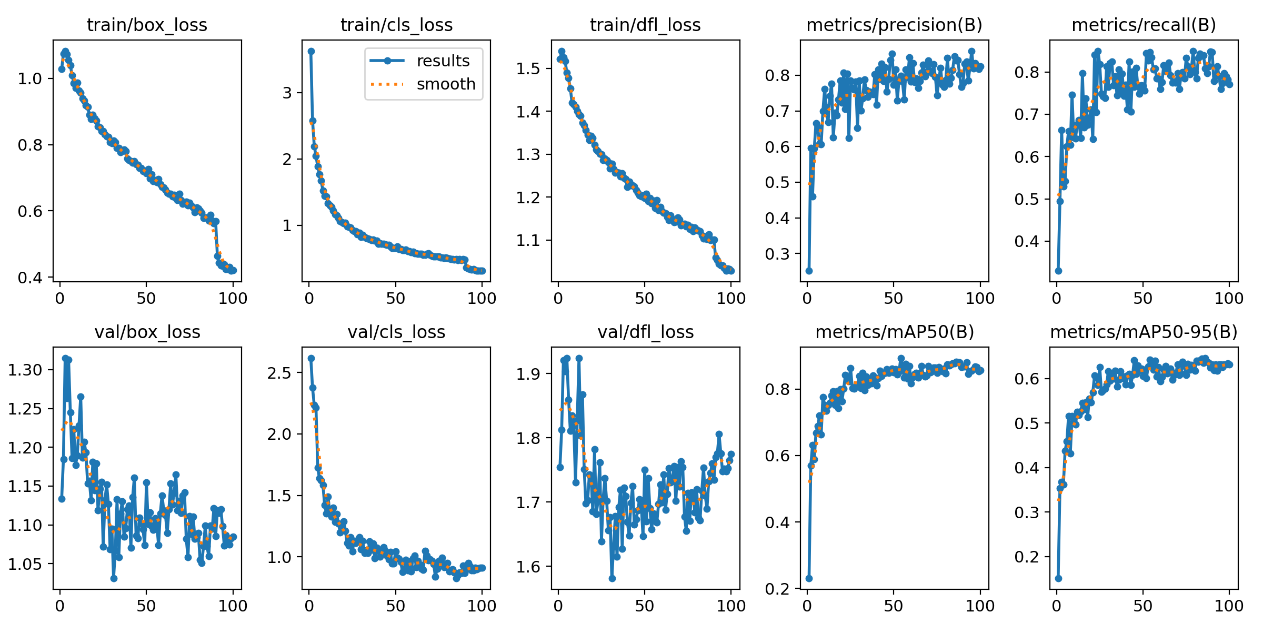
**Görselden Çıkarımlar:**

* Model genellikle doğru tahmin yapıyor, çünkü diyagonal hücrelerdeki değerler görece yüksek.
* Ancak bazı sınıflar arasında karışıklıklar mevcut (örneğin, patlıcan ve limon sınıflarında).

**Normalize Edilmiş Confizyon Matrisi:**  
 Bu matris, tahminlerin yüzdesel olarak dağılımını gösterir. Her bir hücre, ilgili sınıf için yapılan tahminlerin toplamına oranlanarak normalize edilmiştir. Bu, veri dengesizliği olan durumlarda modelin performansını değerlendirmek için daha adil bir karşılaştırma sağlar. Matrisin diyagonalindeki yüksek oranlar, modelin doğru sınıflandırma yaptığını, diyagonal dışındaki oranlar ise hata oranını belirtir.

****

1. **Model Performans İzleme Grafiği:**



**4.1. Eğitim Kayıpları (Üst Sıra - Soldan Sağa)**

* **train/box\_loss**: Modelin nesne algılaması sırasında tahmin edilen kutuların (bounding boxes) yerleşim hatalarını ifade eder. Eğitimin ilerlemesiyle bu kaybın azalması beklenir.
* **train/cls\_loss**: Modelin nesnelerin sınıfını doğru tahmin etme başarısına odaklanır. Düşük bir cls\_loss, sınıflandırma doğruluğunun arttığını gösterir.
* **train/dfl\_loss**: "Distribution Focal Loss" (DFL), YOLO modellerinde kullanılan bir kayıp fonksiyonudur. Nesne sınırlarının daha iyi tahmin edilmesini sağlar. Azalan bir dfl\_loss, modelin nesne sınırlarını daha iyi tahmin ettiğini gösterir.
* **metrics/precision(B)**: Modelin doğruluğunu ölçer; modelin yaptığı pozitif tahminlerin ne kadarının doğru olduğunu ifade eder. Artan bir değer, modelin doğru tahminler yapmada daha iyi olduğunu gösterir.
* **metrics/recall(B)**: Modelin hassasiyetini ölçer; modelin gerçek pozitifleri ne kadar iyi bulduğunu ifade eder. Değerin artması, modelin gerçek nesneleri daha iyi algıladığını gösterir.

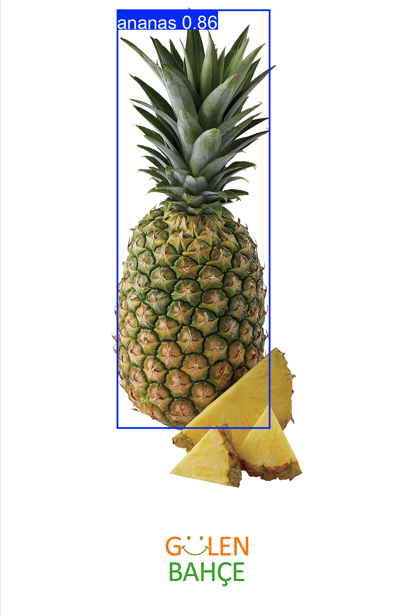
**4.2. Doğrulama Kayıpları (Alt Sıra - Soldan Sağa)**

* **val/box\_loss**: Doğrulama veri seti üzerindeki kutu yerleşim hatalarını ifade eder. Eğitim sürecinde bu kaybın da azalması beklenir.
* **val/cls\_loss**: Doğrulama veri setinde sınıflandırma başarısını ifade eder. Eğitimle birlikte bu kaybın da azalması gerekir.
* **val/dfl\_loss**: Doğrulama veri setinde nesne sınırlarının tahmin başarısı. Azalan bir değer, doğrulamada daha iyi tahminler anlamına gelir.
* **metrics/mAP50(B)**: Ortalama Doğruluk (Mean Average Precision) metriği, modelin tüm tahminlerinin doğruluğunu değerlendirir. mAP@50\text{mAP@50}mAP@50, Intersection over Union (IoU) eşik değeri %50 olan tahminleri içerir. Artış, model performansının arttığını gösterir.
* **metrics/mAP50-95(B)**: IoU eşik değerlerini %50'den %95'e kadar kapsayan daha geniş bir doğruluk değerlendirmesidir. Bu metriğin artması, modelin genel performansının arttığını gösterir.

**Görselden Çıkarımlar:**

1. **Eğitim Süreci**: Tüm eğitim kayıpları (train/box\_loss, train/cls\_loss, train/dfl\_loss) azalmış ve modelin öğrenme sürecinin başarılı olduğunu gösteriyor.
2. **Doğrulama Süreci**: Doğrulama kayıplarında da azalma gözlemleniyor, ancak bazı dalgalanmalar mevcut (özellikle val/dfl\_loss grafiğinde). Bu, modelin doğrulama veri setinde zaman zaman istikrarsız tahminler yaptığını gösterebilir.
3. **Performans Metrikleri**: Precision, Recall, mAP@50 ve mAP@50-95 değerlerinin tamamı artış göstermiş. Bu, modelin nesne algılama ve sınıflandırma görevlerinde başarılı olduğunu işaret eder
4. **PyCharm Üzerinden Test:**
   * Eğitimden sonra model, PyCharm üzerinde test edildi.
   * Test sürecinde bir fotoğraf, modele gönderildi ve model, fotoğraftaki meyve ve sebzeleri başarıyla algılayarak sınıflandırdı.
   * **Ek: PyCharm Üzerinden Test Edilmiş Görseller** (Bir fotoğrafın üzerindeki sınıflandırma sonuçlarıyla birlikte rapora eklenmeli).

ananas, meyve, Ananas, doğal gıdalar içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Para Tanıma Modeli**

1. **Veri Seti Hazırlığı:**
   * Türk Lirası banknotlarının görüntüleri toplandı ve Make Sense platformunda etiketleme işlemi yapıldı.
   * Görseller, Roboflow platformu kullanılarak artırıldı ve veri seti üç katına çıkarıldı.
   * Veri seti yine **%80 eğitim**, **%10 doğrulama** ve **%10 test** olarak ayrıldı.
2. **Model Eğitimi:**
   * Para tanıma işlemi için de **YOLOv8n** modeli seçildi.
   * Eğitim, Google Colab üzerinde gerçekleştirildi ve model **60 epoch** boyunca eğitildi.
   * Elde edilen model, farklı banknotları başarıyla tanıyabilir hale geldi.



**val\_batch2\_pred Görseli:**

Modelin doğrulama setindeki resimlerde yaptığı tahminlerin sonuçlarını görselleştirir. Her bir görselde, modelin algıladığı nesneler üzerine yerleştirilen sınıf etiketleri ve tahmin doğruluk yüzdeleri bulunur. Bu, modelin performansını değerlendirmek için önemlidir, çünkü doğru ve yanlış tahminler görsel olarak analiz edilebilir.

1. **Konfüzyon Matrisi:**
   * Para tanıma modeli için konfüzyon matrisi oluşturularak doğru ve yanlış sınıflandırmalar incelendi.
   * Matriste, benzer görünüme sahip banknotlarda modelin hata yapma olasılıkları gözlemlendi ve gerekirse eğitim verisi üzerinde iyileştirmeler planlandı.

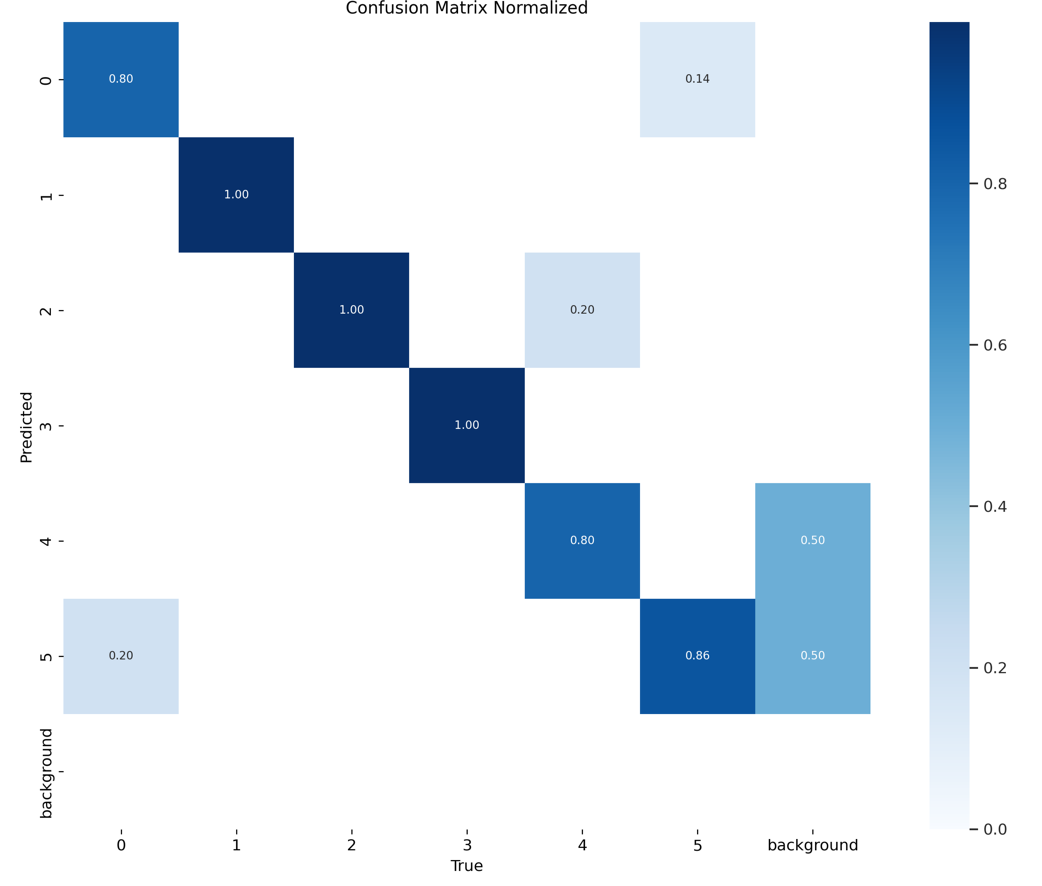
ekran görüntüsü, diyagram, dikdörtgen, kare içeren bir resim

Açıklama otomatik olarak oluşturuldu

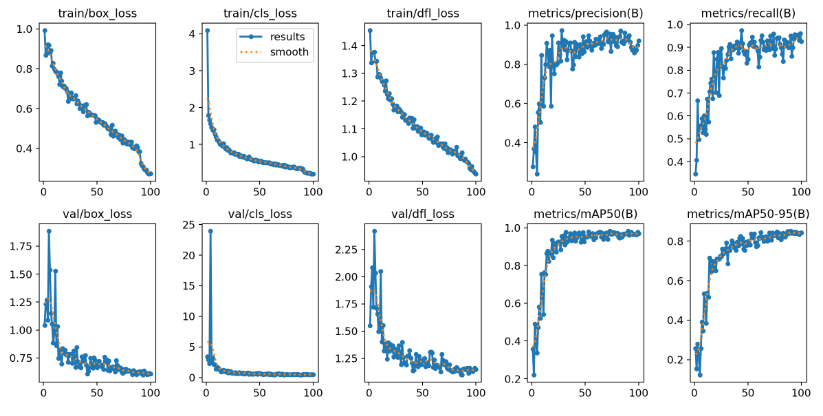
**Görselden Çıkarımlar:**

* Model genellikle doğru tahmin yapıyor, çünkü diyagonal hücrelerdeki değerler görece yüksek.

**Normalize Edilmiş Confizyon Matrisi:**  
 Bu matris, tahminlerin yüzdesel olarak dağılımını gösterir. Her bir hücre, ilgili sınıf için yapılan tahminlerin toplamına oranlanarak normalize edilmiştir. Bu, veri dengesizliği olan durumlarda modelin performansını değerlendirmek için daha adil bir karşılaştırma sağlar. Matrisin diyagonalindeki yüksek oranlar, modelin doğru sınıflandırma yaptığını, diyagonal dışındaki oranlar ise hata oranını belirtir.



1. **Model Performans İzleme Grafiği:**



**1. Eğitim Kayıpları (Üst Sıra - Soldan Sağa)**

* **train/box\_loss**: Modelin tahmin edilen kutuların (bounding boxes) yerleşim hatalarını ölçen kayıp fonksiyonu. Eğitimin ilerlemesiyle bu kaybın azalması beklenir.
* **train/cls\_loss**: Modelin nesne sınıflarını doğru tahmin etme başarısına odaklanan kayıp. Azalan cls\_loss, sınıflandırma doğruluğunu gösterir.
* **train/dfl\_loss**: "Distribution Focal Loss", nesne sınırlarını daha iyi tahmin etmeye yönelik bir kayıp fonksiyonudur. Azalan bir dfl\_loss değeri, daha doğru sınır tahminlerini ifade eder.
* **metrics/precision(B)**: Modelin doğruluk metriğidir; pozitif tahminlerin ne kadarının doğru olduğunu gösterir. Artan değer, modelin pozitif tahminlerde daha isabetli olduğunu belirtir.
* **metrics/recall(B)**: Modelin hassasiyet metriğidir; gerçek pozitifleri ne kadar iyi bulduğunu gösterir. Değerin artması, modelin gerçek nesneleri algılamada daha başarılı olduğunu ifade eder.

**2. Doğrulama Kayıpları (Alt Sıra - Soldan Sağa)**

* **val/box\_loss**: Doğrulama veri seti üzerindeki kutu yerleşim hatalarını ifade eder. Azalan bir değer, doğrulama performansının arttığını gösterir.
* **val/cls\_loss**: Doğrulama veri setinde sınıflandırma kaybını ölçer. Azalan değer, sınıflandırmada doğruluğun arttığını ifade eder.
* **val/dfl\_loss**: Doğrulama veri setindeki nesne sınırı tahminlerinin başarısını ölçer. Dalgalanmalara rağmen genel azalma, modelin doğrulama performansında iyileşmeyi işaret eder.
* **metrics/mAP50(B)**: Ortalama Doğruluk (mAP), Intersection over Union (IoU) eşik değeri %50 olan tahminlerin doğruluğunu ölçer. Artan değer, modelin performansının iyileştiğini gösterir.
* **metrics/mAP50-95(B)**: IoU eşik değeri %50'den %95'e kadar olan bir doğruluk değerlendirmesidir. Artış, modelin genel doğrulukta iyileştiğini ifade eder.

**Görselden Çıkarımlar:**

1. **Eğitim Performansı**: Eğitim kayıplarının (train/box\_loss, train/cls\_loss, train/dfl\_loss) tümü azalmış, bu da modelin öğrenme sürecinin başarılı olduğunu gösterir.
2. **Doğrulama Performansı**: Doğrulama kayıplarında azalma gözlemleniyor, ancak özellikle val/dfl\_loss grafiğinde bazı dalgalanmalar mevcut. Bu, doğrulama setinde zaman zaman tutarsızlıkların olabileceğini gösterir.
3. **Metrikler**: Precision, Recall, mAP50 ve mAP50-95 değerlerinin sürekli artışı, modelin hem eğitim hem de doğrulama setlerinde başarılı olduğunu ve daha iyi performans gösterdiğini işaret etti.
4. **PyCharm Üzerinden Test:**
   * Model, PyCharm üzerinde test edildi.
   * Bir banknot görüntüsü modeli test etmek için kullanıldı ve model, doğru sınıflandırma sonuçlarını verdi.

metin, Banknot, kağıt, kağıt ürünü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Dönem Sonu Hedeflerinin Değerlendirilmesi:

|  |  |
| --- | --- |
| **Gerçekleştirilen Hedefler:** | **2. Dönem Hedefleri:** |
| **1.** Veri seti araştırılıp etiketlendi. | **1.**  Kamera entegrasyonu ve temel bir arayüz oluşturulması. |
| **2.**Yolo algoritmasının çalışma mantığı anlaşılması. | **2.**  Model performansının optimize edilmesi ve doğruluğun artırılması. |
| **3.**Model optimize edildi. | **3.**  Text-to-Speech entegrasyonu ile sesli geri bildirim özelliğinin eklenmesi. |
| **4.** Model eğitildi ve test edildi. | 4. Uygulamanın son haliyle sunuma ve kullanıma hazır hale getirilmesi. |

Kaynaklar:

<https://docs.ultralytics.com/tr>

<https://roboflow.com/>

<https://www.makesense.ai/>

<https://opencv.org/>

<https://colab.research.google.com/>

<https://www.python.org/>

<https://www.kaggle.com/>

ÖĞRENCİNİN DANIŞMAN

Adı ve Soyadı : Yahya KARAMAN Adı ve Soyadı : Betül UZBAŞ

İmzası : İmzası :