**Mango Yaprak Hastalıkları Sınıflandırma Modeli: Teknik Rapor**

**1. Proje Amacı**

Bu proje, mango yapraklarındaki hastalıkları tespit etmek için **Vision Transformer (ViT)** modelini kullanarak bir sınıflandırma modeli geliştirmeyi amaçlamaktadır. Model, mango yapraklarının görüntülerini analiz ederek, hangi hastalığa sahip olduğunu belirlemek için eğitilmiştir.

**2. Kullanılan Teknolojiler**

* **Python**: Veri işleme ve model eğitimi için ana programlama dili.
* **PyTorch**: Derin öğrenme modeli geliştirmek için kullanılan framework.
* **TIMM (PyTorch Image Models)**: Vision Transformer modeli oluşturmak için kullanılan kütüphane.
* **PIL (Pillow)**: Görüntü işlemleri için kullanıldı.
* **Torchvision**: Görüntü dönüşümleri ve veri işleme için kullanıldı.
* **tqdm**: Model test sürecinde ilerleme durumunu göstermek için kullanıldı.

**3. Model Seçimi**

Model olarak **Vision Transformer (ViT)** kullanıldı. ViT, son yıllarda **CNN'lerin yerini alabilecek güçlü bir alternatif** olarak gösterilen bir modeldir.

**Seçilen model:** vit\_base\_patch16\_224

* Transformer tabanlıdır.
* Görüntüyü küçük parçalara (patch) böler ve her parçayı bir diziye dönüştürerek işler.
* CNN'lerden farklı olarak küresel ilişkilere daha iyi odaklanır ve dikkat mekanizmasını kullanır.

**4. Veri Kümesi**

Veri kümesi, mango yapraklarına ait **hastalık sınıflarını içeren** bir görüntü veri setinden oluşmaktadır. Veri seti her bir hastalık için ayrı klasörlere ayrılmıştır.

**Hastalık Sınıfları:**

* **Anthracnose**
* **Bacterial Canker**
* **Healthy (Sağlıklı)**
* **Cutting Weevil**
* **Die back**
* **Gall Midge**
* **Powdery Mildew**
* **Sooty Mould**

Veri seti, modelin her bir görüntüye uygun bir etiket atayarak eğitilmesine olanak sağlamıştır.

**5. Modelin Eğitimi**

Model eğitimi için aşağıdaki adımlar izlenmiştir:

**5.1 Veri Ön İşleme**

Görüntüleri modelin anlayabileceği formata getirmek için aşağıdaki dönüşümler uygulandı:

* **Resize (224x224)**: Tüm görüntüler **224x224** boyutuna getirildi.
* **ToTensor()**: Görüntüler PyTorch’un anlayabileceği tensör formatına dönüştürüldü.
* **Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])**: Modelin daha verimli öğrenmesi için normalizasyon yapıldı.

Kod:

transform = transforms.Compose([

transforms.Resize((224, 224)),

transforms.ToTensor(),

transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])

])

**5.2 Modelin Yüklenmesi ve Eğitilmesi**

Model, **TIMM kütüphanesi** kullanılarak yüklendi:

model = timm.create\_model("vit\_base\_patch16\_224", pretrained=True, num\_classes=8)

model.to(device)

* **pretrained=True**: Önceden eğitilmiş ağırlıklar kullanıldı.
* **num\_classes=8**: Veri setindeki sınıf sayısı kadar çıktı üretildi.

Daha sonra model **eğitim verileri** ile eğitildi ve test verileri ile doğrulandı.

**5.3 Modelin Test Edilmesi**

Eğitilen modelin doğruluğunu görmek için test veri seti ile tahminler yapıldı. Modelin doğruluğunu ölçmek için aşağıdaki yöntem kullanıldı:

correct = 0

total = 0

for class\_name in class\_names:

class\_path = os.path.join(dataset\_path, class\_name)

if not os.path.isdir(class\_path): # Eğer dosya değilse atla

continue

for img\_name in tqdm(os.listdir(class\_path), desc=f"Testing {class\_name}"):

img\_path = os.path.join(class\_path, img\_name)

image = Image.open(img\_path).convert("RGB")

image = transform(image).unsqueeze(0).to(device)

with torch.no\_grad():

output = model(image)

predicted\_class = torch.argmax(output, dim=1).item()

predicted\_label = class\_names[predicted\_class]

if predicted\_label == class\_name:

correct += 1

total += 1

accuracy = (correct / total) \* 100 if total > 0 else 0

print(f"✅ Modelin Genel Doğruluğu: {accuracy:.2f}%")

Bu kod, test verilerinin her biri için tahminde bulunarak **modelin genel doğruluk oranını hesapladı**.

**5.4 Sonuçlar**

* Model, test verileri üzerinde **yüksek doğrulukla çalıştı**.
* Vision Transformer (ViT) tabanlı model, CNN modellerine kıyasla **daha iyi küresel özellik öğrenme yeteneği** sundu.
* Bazı durumlarda modelin **yanlış tahmin yaptığı gözlemlendi**, ancak genel başarı oranı tatmin edici bulundu.

**6. Sonuç ve Değerlendirme**

Bu çalışmada **Vision Transformer (ViT) kullanarak mango yaprak hastalıklarını sınıflandıran bir model geliştirdik**. Model, eğitim verisi üzerinde başarılı sonuçlar verdi ve test aşamasında da yüksek doğruluk oranlarına ulaştı.

İleri aşamalarda, **veri setinin genişletilmesi** ve **hiperparametre optimizasyonu** yapılarak modelin daha iyi hale getirilmesi mümkündür.