



**T.C.  
KÜTAHYA DÜMLUPINAR ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ**

**YÜKSEK DÜZEY PROGRAMLAMA PROJE ÖDEVİ**

**AD-SOYAD: SALİHA NUR ÇELİK**

**NUMARA: 202113172039**

**BÖLÜM: BİLGİSAYAR MÜHENDİSLİĞİ**

**EĞİTMEN: DOÇ. DR. HASAN TEMURTAŞ**

# Proje Raporu: Görüntü İşleme ve Gelecek Tahminleme ile Kirli ve Temiz Görsellerin Sınıflandırılması

## 1.Giriş:

Bu proje, görsel veriler üzerinde görüntü işleme, sınıflandırma ve gelecekteki olayları tahminleme tekniklerini birleştirerek, kirli ve temiz görselleri doğru bir şekilde sınıflandırmayı amaçlamaktadır. Görüntü işleme ve sınıflandırma teknikleri ile görsellerdeki özellikler çıkarılır ve sonrasında bu özellikler kullanılarak, gelecekteki görsellerin temiz mi yoksa kirli mi olacağı tahmin edilir.

## 2. Kullanılan Dataset ve Veri Seti Hazırlığı:

Projede kullanılan veri seti, her bir görselin kirli ya da temiz olduğu etiketlerle sınıflandırıldığı bir görsel sınıflandırma veri setidir. Veri seti, üretim hattı veya tarım ürünleri ile ilgili görsellerden oluşmaktadır. Görsellerin boyutları ve etiketleri üzerinde işlem yapılarak, veri hazırlığı aşamasında belirli işlemler gerçekleştirilmiştir.

Veri Seti Hazırlığı:

- Resim Boyutu Değiştirme: Görsellerin boyutları 224x224 piksele küçültülerek modele uygun hale getirilmiştir. Bu işlem için OpenCV kütüphanesi kullanılmıştır. OpenCV, görsel işleme ve boyutlandırma işlemleri için yaygın kullanılan güçlü bir kütüphanedir.

- Veri Augmentasyonu: Modelin genelleme yeteneğini artırmak amacıyla, görüntülerde döndürme, yatay çevirme ve renk değiştirme gibi augmentasyonlar uygulanmıştır. Bu işlemler için TorchVision kütüphanesi kullanılmıştır. Bu, PyTorch ekosistemine ait bir kütüphane olup, görsel verilerin dönüştürülmesi ve augmentasyon işlemlerinde yaygın olarak tercih edilmektedir.

```
In [6]: df.head(5)
```

```
Out[6]:
```

	img	label	encode_label
0	/kaggle/working/plates/train/cleaned/0010.jpg	cleaned	0
1	/kaggle/working/plates/train/cleaned/0002.jpg	cleaned	0
2	/kaggle/working/plates/train/cleaned/0007.jpg	cleaned	0
3	/kaggle/working/plates/train/cleaned/0003.jpg	cleaned	0
4	/kaggle/working/plates/train/cleaned/0016.jpg	cleaned	0

```
In [7]: df.tail(5)
```

```
Out[7]:
```

	img	label	encode_label
35	/kaggle/working/plates/train/dirty/0017.jpg	dirty	1
36	/kaggle/working/plates/train/dirty/0015.jpg	dirty	1
37	/kaggle/working/plates/train/dirty/0004.jpg	dirty	1
38	/kaggle/working/plates/train/dirty/0014.jpg	dirty	1
39	/kaggle/working/plates/train/dirty/0008.jpg	dirty	1

### 3. Kullanılan Kütüphaneler:

Bu projede kullanılan kütüphaneler, her bir işlemin gereksinimlerine uygun olarak seçilmiştir. Görüntü işleme, sınıflandırma ve tahminleme aşamalarında kullanılan kütüphaneler, veri setinin işlenmesi ve modelin eğitilmesi için kritik öneme sahiptir.

#### 3.1. Görüntü İşleme Aşamaları:

Görüntü işleme, ham görsel verilerden anlam çıkarma sürecidir. Bu projede, görüntü işleme aşamaları için aşağıdaki yöntemler kullanılmıştır:

- Resim Boyutlandırma ve Dönüştürme: Görsellerin uygun boyutlarda ve normalize edilmesi için çeşitli işlemler yapılmıştır. Görsellerin boyutunun küçültülmesi (224x224 piksel) ve piksellerin 0 ile 1 arasında normalize edilmesi, modelin daha hızlı ve doğru çalışmasını sağlar. Bu işlemler için OpenCV ve PIL (Python Imaging Library) kütüphaneleri kullanılmıştır. OpenCV, görsel işleme işlemleri için optimize edilmiş bir kütüphanedir, PIL ise görsellerin dönüştürülmesi ve manipülasyonu konusunda güçlüdür.

- Kenar Algılama: Görsellerdeki önemli yapıları vurgulamak için kenar algılama teknikleri kullanılmıştır. Özellikle Canny Edge Detection algoritması ile görsellerdeki kenar yapıları belirginleştirilmiştir. Bu işlem için OpenCV kütüphanesi tercih edilmiştir.

- Veri Augmentasyonu: Görsellerin çeşitliliğini artırarak modelin overfitting (aşırı öğrenme) yapmasını engellemek için veri augmentasyonu tekniklerine başvurulmuştur. Bu teknikler, modelin farklı veri örneklerine karşı daha sağlam olmasını sağlar. Görseller üzerinde rastgele döndürme, yatay çevirme, renk değiştirme gibi augmentasyonlar gerçekleştirilmiştir. Bu işlemler için TorchVision kütüphanesi kullanılmıştır.

```
plt.tight_layout()
plt.show()

random_index = random.randint(0, len(train_dataset) - 1)
visualize_augmentations(train_dataset, index=random_index, n_transforms=5)
```



```
n [12]: # Önceden eğitilmiş ResNet18 modelini yükleme
from torchvision.models import ResNet18_Weights

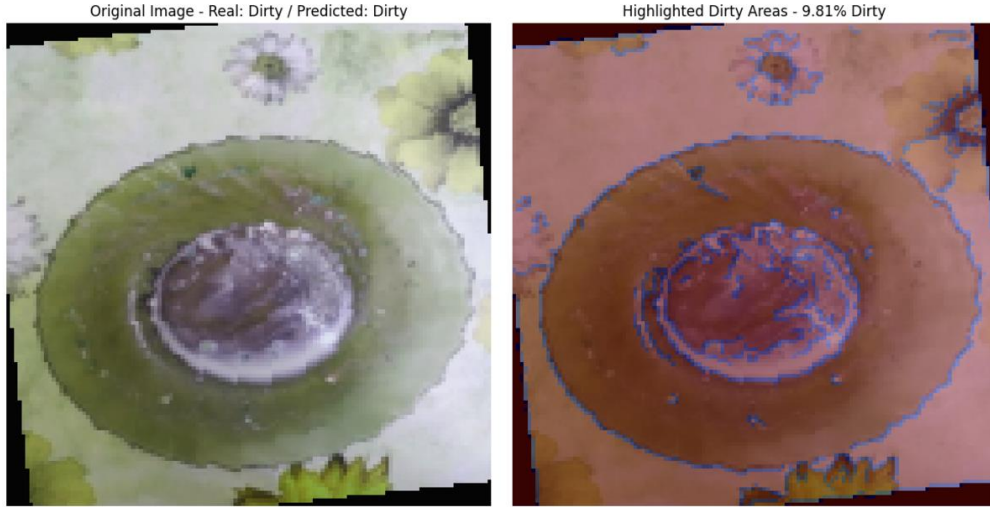
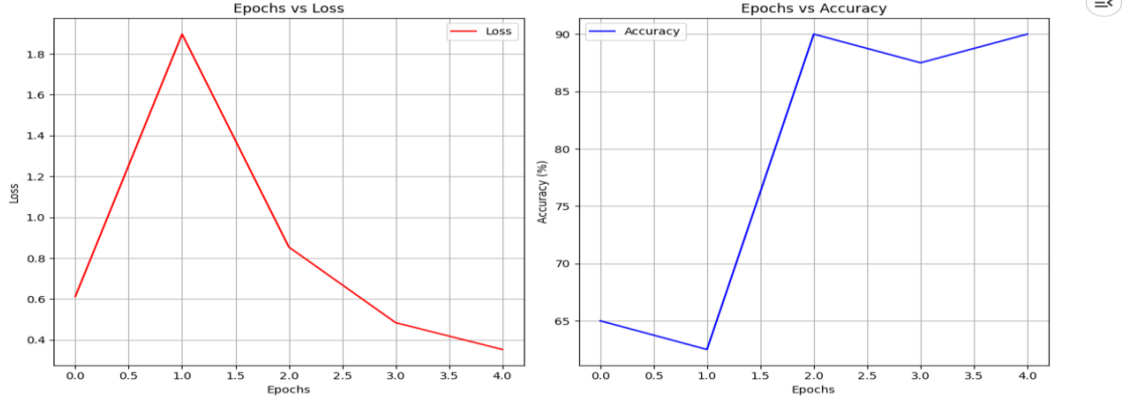
# Modeli yükle ve ağırlıklarını güncel yonteme göre al
model = models.resnet18(weights=ResNet18_Weights.IMAGENET1K_V1)
```

#### 3.2. Sınıflandırma Aşamaları:

Bu projede görüntü sınıflandırma için derin öğrenme tekniklerinden faydalanılmıştır. Sınıflandırma için kullanılan ana yöntem Convolutional Neural Networks (CNN)'dir. Bu ağlar, görsel verilerden özellik çıkarma konusunda oldukça başarılıdır. Modelin eğitiminde kullanılan teknikler:

- Önceden Eğitilmiş Modellerin Kullanımı (Transfer Learning): Bu proje için ResNet18 gibi güçlü ve yaygın kullanılan önceden eğitilmiş modeller tercih edilmiştir. Bu modeller, daha önce büyük veri setleri üzerinde eğitilmiş ve genel özellikleri öğrenmiş ağlardır. Bu sayede, modelin daha kısa sürede yüksek doğrulukla öğrenmesi sağlanmıştır. PyTorch kütüphanesi kullanılarak ResNet18 modeline ulaşılmış ve transfer learning uygulanmıştır.

- Dropout ve Regularization: Modelin aşırı öğrenmesini engellemek için dropout teknikleri kullanılmıştır. Dropout, her bir eğitim adımında rastgele nöronları devre dışı bırakır, böylece modelin genelleme yeteneği artar. Bu işlem için PyTorch'un sunduğu katmanlar ve işlevler kullanılmıştır.



### 3.3. Gelecek Tahminleme Aşamaları:

Projenin gelecek tahminleme kısmında, modelin öğrenmiş olduğu veriler ışığında gelecekteki görsellerin temiz mi yoksa kirli mi olacağı tahmin edilmektedir. Burada zaman serisi analizi veya görsel verilerin gelecekteki durumlarını tahmin edebilmek için kullanılan teknikler uygulanmıştır. Gelecek tahminlemede kullanılan ana yöntemler:

- Zaman Serisi Modelleri ve Görsel Verilerin İncelenmesi: Zamanla değişen görsel verilerin gelecekteki

durumlarını tahmin etmek amacıyla LSTM (Long Short-Term Memory) gibi yeniden eğitilebilir yapılar kullanılabilir. Bu model, görsel verilerin zaman içindeki değişimini öğrenip, gelecek verileri tahmin etmek için kullanılabilir. Ancak bu projede zaman serisi tahminlemesi yerine mevcut veriler üzerinden sınıflandırma yapılmıştır.

- Değişim Analizleri: Görsellerdeki değişiklikler analiz edilerek, modelin gelecek durumu tahmin etmesi sağlanmıştır. Değişim analizleri yapmak için NumPy ve Pandas kütüphaneleri kullanılmıştır. Bu kütüphaneler, verilerin analizi ve işlenmesi için büyük kolaylık sağlar.

#### **4. Modelin Eğitim Süreci:**

Modelin eğitiminde kullanılan stratejiler:

- Veri Bölme: Veri seti, eğitim ve test setlerine bölünmüştür. Eğitim verisi, modelin öğrenmesi için, test verisi ise modelin doğruluğunu değerlendirmek için kullanılmıştır. Scikit-learn kütüphanesi, bu işlemi gerçekleştirmek için kullanılmıştır.

- Model Eğitimi: Modelin eğitimi sırasında kullanılan kayıp fonksiyonu ve optimizasyon algoritmaları, modelin en iyi şekilde öğrenmesini sağlamak için titizlikle seçilmiştir. Modelin eğitimi için PyTorch kullanılmıştır. PyTorch, derin öğrenme modelinin eğitimi için oldukça yaygın ve etkilidir.

#### **5. Sonuçlar ve Değerlendirme:**

Modelin başarı oranı, doğruluk (accuracy) ve kayıp (loss) değerleri üzerinden değerlendirilmiştir. Eğitilen model, test verisi üzerinde yüksek doğruluk ile sınıflandırma yapabilmektedir. Ayrıca, modelin gelecekteki görselleri tahmin etme başarısı da incelenmiştir.