

# YÜKSEK DÜZEY PROGRAMLAMA PROJE ÖDEVİ

## 1 – Seçilen Veriseti

**Cleaned vs Dirty:** <https://www.kaggle.com/competitions/platesv2/data>

İçerisinde temiz ve kirli olmak üzere iki sınıf barındırır. Her bir sınıfta 19 tane olmak üzere 38 tane eğitim verisi ve 744 adet test verisi bulunmaktadır. Az bir verisiyle eğitim ezberleme sorunu ortaya çıkarmakta ve test verilerindeki doğruluk tutarsızlığını meydana getirmekte. Ezberlemeyi ortadan kaldırmak için transform yöntemleri uygulanması gerekmektedir.

## 2 – Uygulanan Transform Yöntemleri

Rastgele Kontrast değişimi ve Normalizasyon işlemleri uygulanmıştır. Odaklanmamız gereken tabaklar resmin ortasına yakın olduğu için merkezden belirli bir oranda çember çizilecek ve korunacak geri kalan kısmı ise siyah arka plan olacak şekilde bir işlem uygulanmıştır.

## 3 – Model Kurgusu

Model olarak önceden eğitilmiş bir modeli değil kendi tasarladığım bir modeli kullandım. Model standart bir CNN modelidir. Doğruluğu ve stabiliteyi arttırmak amacıyla BatchNorm2d ve Dropout katmanları ile zenginleştirilmiştir.

### Dropout

Dropout, aşırı öğrenmeyi önlemek için eğitim sırasında giriş birimlerinin bir kısmını rastgele sıfırlayan bir düzenleme tekniğidir. Modelin belirli nöronlara aşırı bağımlılığını azaltarak daha iyi genelleştirme sağlar.

### Conv2d

Conv2d (Konvolüsyon Katmanı), görüntü gibi 2D giriş verisi üzerinde konvolüsyon işlemi gerçekleştirir. Giriş görüntüsüne filtreler uygulayarak kenar, doku ve desen gibi özellikleri algılar.

### MaxPool2d

MaxPool2d, belirtilen bir pencere boyunca maksimum değeri alarak girişin uzamsal boyutlarını azaltan bir havuzlama katmanıdır. Girişin boyutunu küçültmeye ve hesaplama maliyetini düşürmeye yardımcı olur.

### BatchNorm2d

BatchNorm2d (Batch Normalizasyon), önceki bir aktivasyon katmanının çıktısını, partinin ortalamasını ve varyansını ayarlayarak normalleştirir. Bu, öğrenme sürecini dengelemeye ve daha yüksek öğrenme oranları kullanılmasına olanak tanır.

### ReLU

ReLU (Düzeltilici Doğrusal Birim), girdideki tüm negatif değerleri sıfıra ayarlayarak ve pozitif değerleri olduğu gibi bırakarak doğrusal olmayan bir dönüşüm uygular. Ağın karmaşık desenleri öğrenmesine yardımcı olur.

## Linear

Linear katman (tam bağlantılı katman), giriş verisine bir ağırlık matrisi ve bir sapma vektörü uygulayarak doğrusal bir dönüşüm gerçekleştirir. Genellikle sınıflandırma veya regresyon görevleri için sinir ağının son katmanlarında kullanılır.

```
class PlateModel(nn.Module):
    def __init__(self, img_size=256):
        super(PlateModel, self).__init__()

        self.img_size = img_size

        self.conv1 = nn.Conv2d(3, 64, kernel_size=3, stride=1, padding=1)
        self.bn1 = nn.BatchNorm2d(64)
        self.conv2 = nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1)
        self.bn2 = nn.BatchNorm2d(128)
        self.conv3 = nn.Conv2d(128, 256, kernel_size=3, stride=1, padding=1)
        self.bn3 = nn.BatchNorm2d(256)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
        self.dropout = nn.Dropout(0.2)
        self.fc1 = nn.Linear(256 * (self.img_size // 8) * (self.img_size // 8), 128)
        self.dropout2 = nn.Dropout(0.1)
        self.fc2 = nn.Linear(128, 2) # İki sınıf için çıkış katmanı

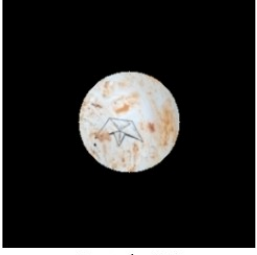
    def forward(self, x):
        x = self.pool(torch.relu(self.bn1(self.conv1(x))))
        x = self.pool(torch.relu(self.bn2(self.conv2(x))))
        x = self.pool(torch.relu(self.bn3(self.conv3(x))))
        x = x.view(-1, 256 * (self.img_size // 8) * (self.img_size // 8))
        x = torch.relu(self.dropout(x))
        x = self.fc1(x)
        x = torch.relu(self.dropout2(x))
        x = self.fc2(x)
        return x
```

## 4 – Model Eğitim ve Test Sonuçları

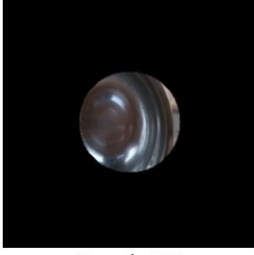
\* En iyi model eğitim sonuna kadar saklanmakta ve en son yüklenmektedir o baz alınmıştır.

Train Loss / Accuracy	<b>Loss: 0.6512, Accuracy: 0.7250 *</b>
Test Loss / Accuracy	<b>Loss: 0.6115, Accuracy: 0.8105</b>

Gerçek: 0.0  
Tahmin: 0



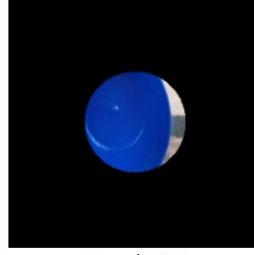
Gerçek: 0.0  
Tahmin: 0



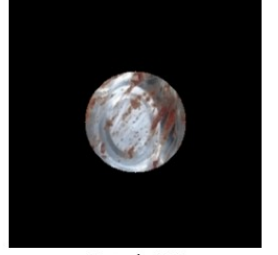
Gerçek: 0.0  
Tahmin: 0



Gerçek: 0.0  
Tahmin: 1



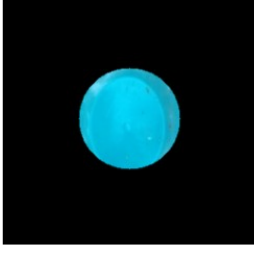
Gerçek: 0.0  
Tahmin: 0



Gerçek: 0.0  
Tahmin: 1



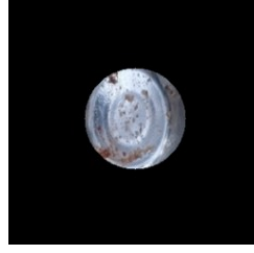
Gerçek: 0.0  
Tahmin: 1



Gerçek: 0.0  
Tahmin: 0



Gerçek: 0.0  
Tahmin: 0



Gerçek: 0.0  
Tahmin: 0



**HACI OSMAN KARABULUT**

**201913171052**