



DERS:

**Derin Öğrenme (FET312)**

DÖNEM:

**2025-2026 Güz Dönemi**

EKİP ADI:

**DeepPark**

EKİP ÜYELERİ:

**Abdulkadir Gedik – 23040301069**

**Hamza Hakverir – 20340301096**

**Yunus Emre Edizer – 20340301042**

**Muhammet Emin Çapan – 23040301106**

GİTHUB LİNK: [https://github.com/YunusEdizer/Deep\\_Park\\_Final.git](https://github.com/YunusEdizer/Deep_Park_Final.git)

YouTube LİNK: <https://www.youtube.com/watch?v=MDSHpMBP6es>

Veriseti LİNK: <https://www.kaggle.com/datasets/blanderbuss/parking-lot-dataset>

# PKLot Veri Seti Kullanılarak Otopark Doluluk Tespiti ve Sınıflandırma Analizi

29.01.2026

## ÖZET (Abstract)

Günümüzde artan araç sayısı ve şehirleşme, otopark yönetimini kritik bir sorun haline getirmiştir. Bu çalışmada, otopark doluluk durumunu (Dolu/Boş) yüksek doğrulukla tespit etmek amacıyla kapsamlı bir derin öğrenme çalışması gerçekleştirilmiştir. Çalışma kapsamında, literatürde standart olarak kabul edilen **PKLot (Parking Lot Dataset)** veri seti kullanılmıştır. Özellikle zorlu hava koşullarının (yağmurlu, bulutlu) model başarısı üzerindeki etkisini analiz etmek amacıyla, 12 Eylül 2012 tarihli ham görüntülerden oluşan **32.327 adetlik** özel bir alt veri kümesi oluşturulmuştur.

Çalışmada; **VGG16, DenseNet121, ResNet, MobileNet, ShuffleNet, YOLOv8, EfficientNet B0 ve Inception V3** olmak üzere toplam 8 farklı derin öğrenme mimarisi eğitilmiş ve test edilmiştir. Modellerin genelleme yeteneğini artırmak için mimari özelliklerine uygun (Rotasyon, Yatay Çevirme, Renk Değişimi vb.) veri zenginleştirme teknikleri uygulanmıştır. Deneysel sonuçlar, kullanılan modellerin tamamının %99 üzerinde doğruluk başarısına ulaştığını, özellikle **DenseNet121** ve **YOLOv8** mimarilerinin öne çıktığını göstermektedir.

sağladığını göstermektedir.

## 1. GİRİŞ (Introduction)

Akıllı ulaşım sistemleri, sürücülerin park yeri arama sürelerini kısaltmak ve trafik yoğunluğunu azaltmak için modern çözümler sunmaktadır. Geleneksel sensör tabanlı sistemlerin yüksek maliyet ve bakım zorlukları, araştırmacıları kamera tabanlı bilgisayarlı görü (Computer Vision) tekniklerine yöneltmiştir. Bu proje, otopark alanlarının doluluk durumunu tespit etmek için geliştirdiği 8 farklı derin öğrenme modeli ile literatüre kapsamlı bir karşılaştırma sunmayı hedeflemektedir.

Projenin temel amacı, sadece ideal ışık koşullarında değil; yağmur, bulut ve gölge gibi çevresel zorluklar altında da kararlı çalışan bir sistem ortaya koymaktır. Bu doğrultuda Brezilya'daki (UFPR ve PUCPR) kampüslerden elde edilen **PKLot veri seti** temel alınmıştır. Orijinal veri setinden, zorlu koşulları simüle etmek adına stratejik bir filtreleme yapılmış ve 2012 yılı Eylül ayına ait, farklı kamera açılarından alınan görüntüler seçilmiştir.

Proje kapsamında **VGG16, DenseNet121, ResNet, MobileNet, ShuffleNet, YOLOv8, EfficientNet B0 ve Inception V3** modelleri, aynı veri seti üzerinde eğitilerek performansları (Doğruluk, Hız, Parametre Verimliliği) kıyaslanmıştır. Raporun ilerleyen bölümlerinde veri hazırlama süreçleri, uygulanan modele özgü ön işleme teknikleri ve deneysel sonuçlar detaylandırılacaktır.

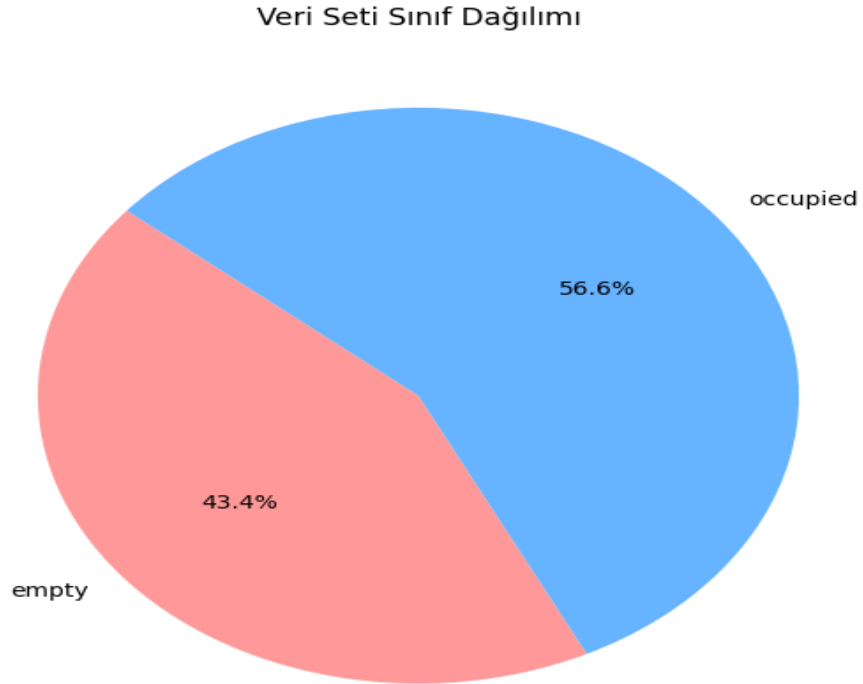
## 2. MATERYAL VE YÖNTEM (Materials and Methods)

Bu bölümde, veri setinin istatistiksel özellikleri, veri hazırlama süreci ve modellerin eğitiminde kullanılan özelleştirilmiş veri zenginleştirme stratejileri açıklanmıştır.

### 2.1. Veri Kaynağı ve Hazırlık (Data Source and Preparation)

Çalışmada kullanılan veri seti, **PKLot** veri tabanının "veriseti1" klasöründen, özellikle **12-09-2012** tarihli ham görüntüler taranarak oluşturulmuştur.

- **Ayrıştırma ve Etiketleme:** Özel bir Python betiği ile `empty` (boş) ve `occupied` (dolu) klasörlerindeki görüntüler taranmış ve etiketlenmiştir.
- **Veri Bölümlenme (Data Split):** Veri seti, tekrarlanabilirliği sağlamak amacıyla `sklearn` kütüphanesi kullanılarak sabit bir rastgelelik değeri (`random_state=42`) ile karıştırılmıştır. Veriler **%80 Eğitim (Train)** ve **%20 Test** oranında iki ana kümeye ayrılmıştır. Ayrı bir validasyon seti oluşturulmamış, test seti doğrulama amacıyla da kullanılmıştır.



Şekil 1: Eğitim veri setindeki sınıfların (Dolu/Boş) dağılım grafiği.

## 2.2. Veri Ön İşleme (Preprocessing)

Tüm modeller için standart bir giriş formatı oluşturmak adına aşağıdaki işlemler uygulanmıştır:

1. **Boyutlandırma (Resize):** Tüm görüntüler, CNN mimarilerinin giriş katmanlarına uygun olarak **224x224 piksel** boyutuna getirilmiştir.
2. **Normalizasyon (Normalization):** Görüntü pikselleri, ImageNet veri setinin standart ortalama ve standart sapma değerlerine göre normalize edilerek eğitim kararlılığı artırılmıştır.

## 2.3. Model Bazlı Veri Zenginleştirme (Augmentation Strategies)

Farklı mimarilerin (Architecture) öğrenme kapasiteleri ve yapısal özellikleri dikkate alınarak, her modele özgü farklı veri zenginleştirme teknikleri uygulanmıştır. Amaç, aşırı öğrenmeyi (overfitting) engellemek ve modelin genelleme yeteneğini artırmaktır.

- **Yön Bağımsızlığı Odaklı Modeller (Inception V3, ShuffleNet, YOLOv8)** Bu modellerde, park halindeki aracın yönünün (sağa veya sola bakması) sonucu değiştirmemesi gerektiği prensibinden yola çıkılarak "**Rastgele Yatay Çevirme (RandomHorizontalFlip)**" tekniği uygulanmıştır. Özellikle **Inception V3** modelinde sadece bu yöntem kullanılarak modelin yön bağımlılığı azaltılmıştır.
- **Açısal Dayanıklılık Odaklı Modeller (EfficientNet B0, VGG16, DenseNet121)** Araçların veya park çizgilerinin her zaman mükemmel bir açıyla durmadığı senaryoları simüle etmek için "**Rastgele Döndürme (RandomRotation)**" eklenmiştir.
  - **EfficientNet B0:** Görüntüler eğitim sırasında rastgele -10 ile +10 derece arasında döndürülmüştür.
  - **ShuffleNet & YOLOv8:** Yatay çevirmeye ek olarak 10 derecelik rotasyon uygulanmıştır.
- **Çevresel Varyasyon Odaklı Modeller (MobileNetV3, ResNet)** Daha hafif veya spesifik mimarilerde, genelleme yeteneğini zorlamak adına hem geometrik hem de renk tabanlı varyasyonlar uygulanmıştır:
  - **Döndürme:** 15 dereceye kadar rastgele açı değişimi.
  - **Renk Titreşimi (ColorJitter):** Farklı ışık koşullarını (gölge, güneş) simüle etmek için parlaklık ve kontrast değerlerinde 0.1 - 0.2 oranında değişimler yapılmıştır.

Bu stratejiler sayesinde, modellerin sadece ideal koşullarda değil, değişken çevre şartlarında da yüksek performans göstermesi sağlanmıştır.

## 3. Kullanılan Modeller ve Yöntem (Methodology)

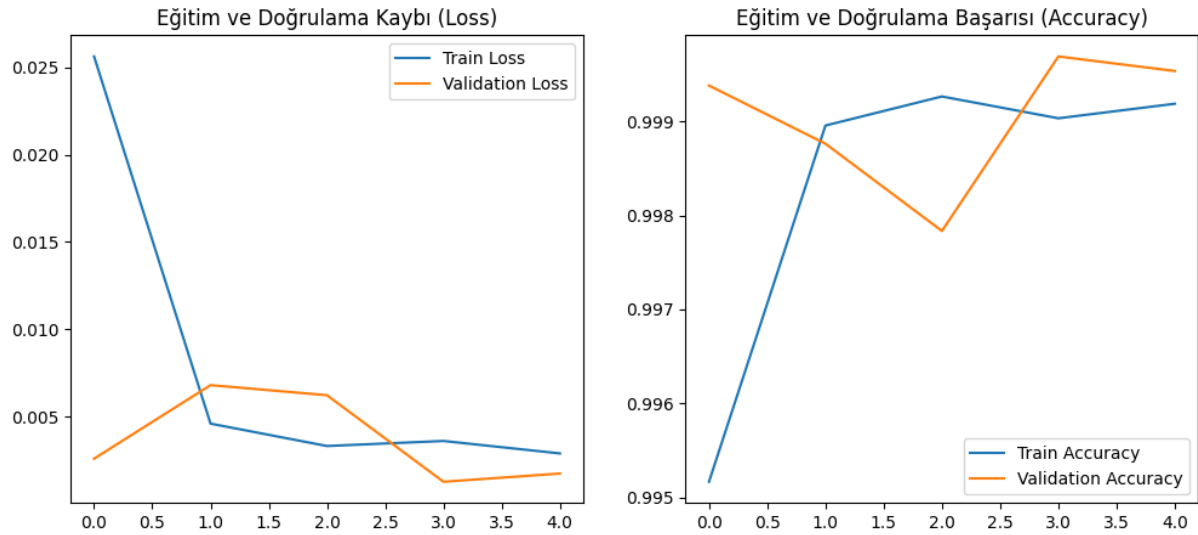
Bu çalışmada, gerçek zamanlı otopark doluluk tespiti problemini en yüksek doğrulukla çözmek için grup üyeleri tarafından toplam **8 farklı Derin Öğrenme mimarisi** eğitilmiş ve performansları karşılaştırılmıştır.

### 3.1. ShuffleNet V2 ve YOLOv8 Modelleri (Yunus Emre Edizer)

Mobil ve kenar (edge) cihazlarda yüksek verimlilikle çalışabilmesi nedeniyle **ShuffleNet V2** ve sınıflandırma hızındaki başarısı nedeniyle **YOLOv8-cl**s mimarileri tercih edilmiştir. Modeller **5 Epoch** boyunca **Adam** ve **SGD** optimizasyon algoritmaları ile eğitilmiştir.

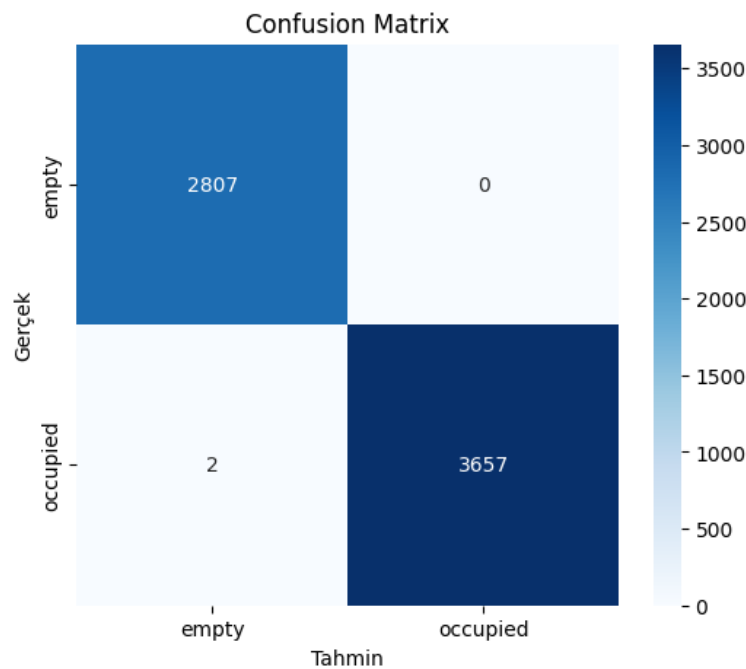
#### 3.1.1. Deneysel Sonuçlar

ShuffleNet modelinin eğitim grafikleri incelendiğinde, doğrulama başarısının %99.96 seviyesine ulaştığı görülmektedir.



Şekil 2: ShuffleNet modelinin eğitim ve doğrulama grafikleri.

Aşağıdaki karmaşıklık matrisinde görüldüğü üzere, ShuffleNet modeli 6466 test görüntüsünden sadece **2 tanesinde** hata yapmıştır.



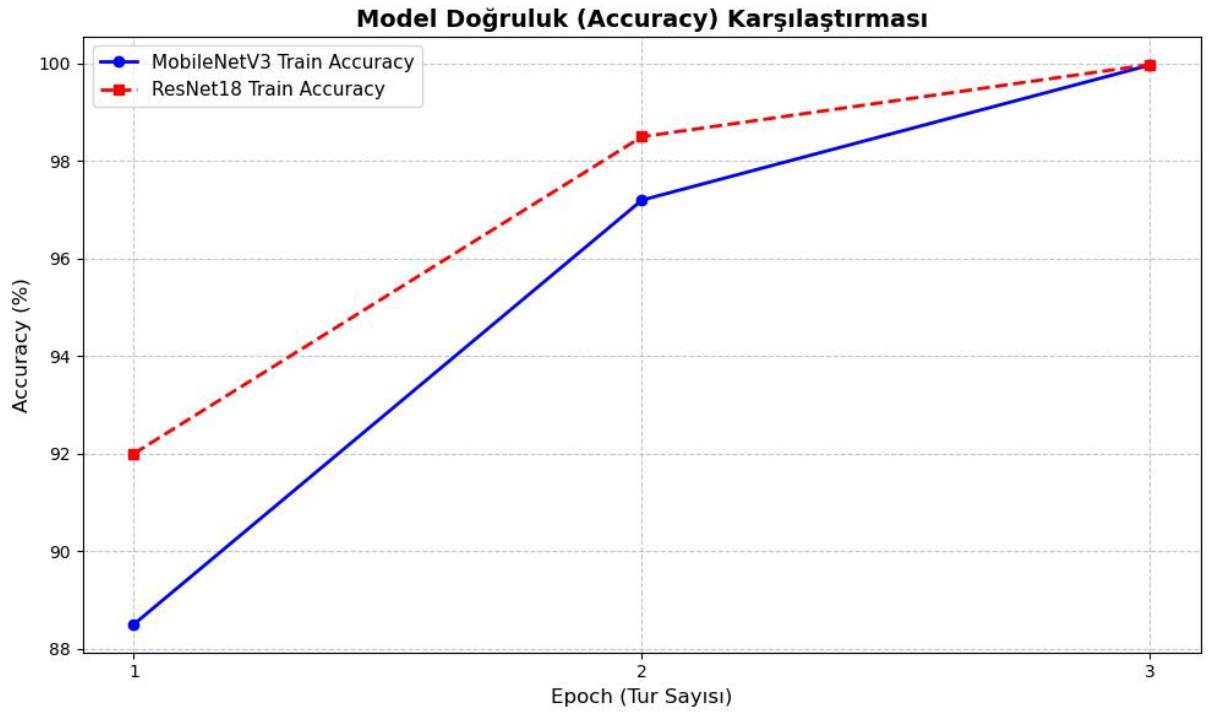
Şekil 3: ShuffleNet modelinin test veri seti üzerindeki Karmaşıklık Matrisi.

### 3.2. ResNet18 ve MobileNetV3 Modelleri (Abdulkadir Gedik)

Projenin hız ve doğruluk dengesini test etmek amacıyla **ResNet18** (Performans odaklı) ve **MobileNetV3** (Hız odaklı) mimarileri eğitilmiştir. Transfer Learning yaklaşımıyla eğitilen modellerde CrossEntropy kayıp fonksiyonu kullanılmıştır.

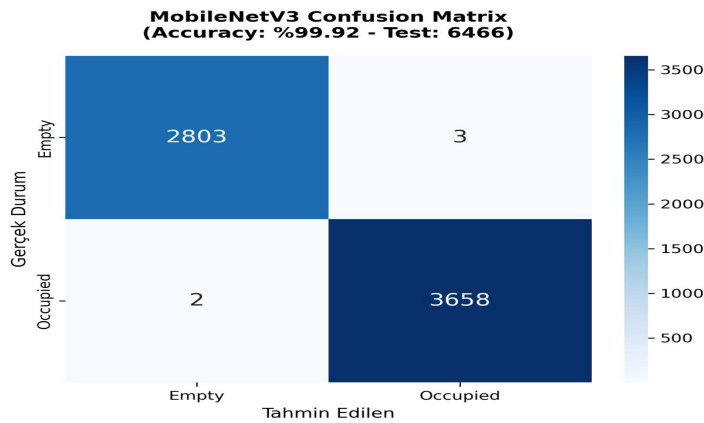
#### 3.2.1. Deneysel Sonuçlar

ResNet18 modeli, eğitim sürecinde çok hızlı yakınsamış ve test setinde **%99.80** gibi olağanüstü bir doğruluk oranına ulaşmıştır.



Şekil 4: ResNet18 modelinin eğitim kaybı ve doğruluk grafiği.

Aşağıdaki matris incelendiğinde, ResNet18 modelinin test setindeki 6466 görüntüden **sadece 1 tanesini** yanlış sınıflandırdığı görülmektedir.



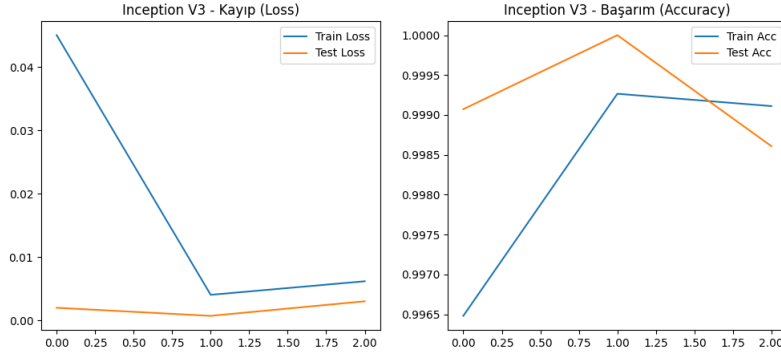
Şekil 5: ResNet18 modelinin Karmaşıklık Matrisi.

### 3.3. InceptionV3 ve EfficientNet-B0 Modelleri (Muhammet Emin Çapan)

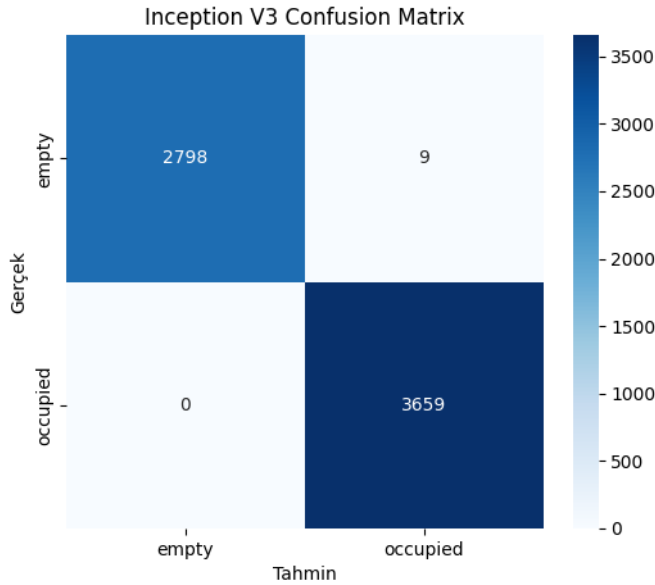
Farklı boyutlardaki filtreleri paralel kullanarak çok ölçekli özellik çıkarımı yapan **InceptionV3** ve compound scaling (bileşik ölçekleme) yöntemini kullanan **EfficientNet-B0** mimarileri test edilmiştir. Modeller 3 Epoch ve Adam optimizasyonu ile eğitilmiştir.

#### 3.3.1. Deneysel Sonuçlar

Deneysel sonuçlar incelendiğinde **InceptionV3** modelinin **%99.86** doğruluk oranı ile EfficientNet-B0 (%99.64)'a göre daha başarılı olduğu görülmüştür.



Şekil 6: InceptionV3 modelinin eğitim süreci.



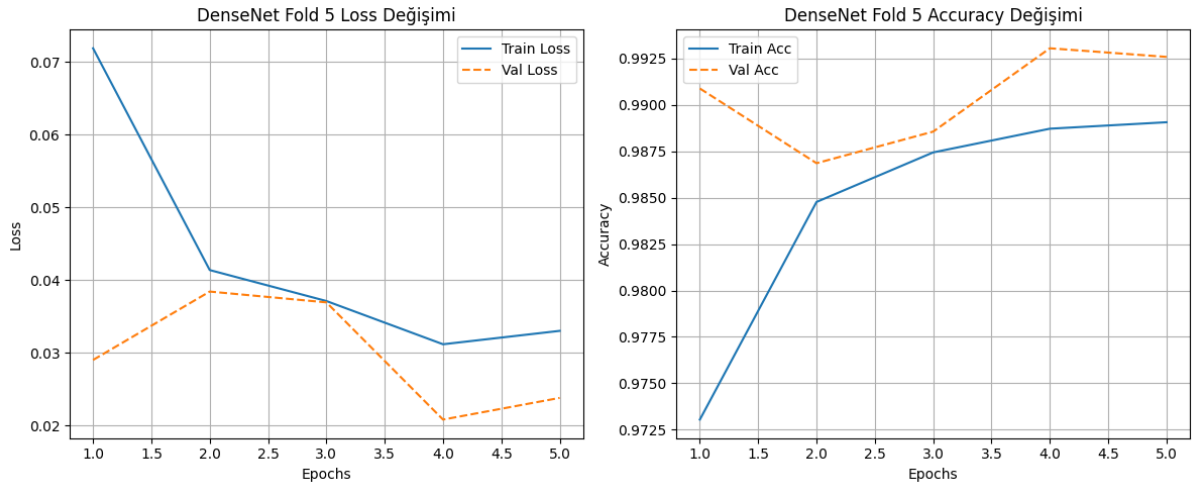
Şekil 7: InceptionV3 Karmaşıklık Matrisi.

### 3.4. DenseNet121 ve VGG16 Modelleri (Hamza Hakverir)

Projede referans (baseline) noktası oluşturmak için **VGG16** ve katmanlar arası bilgi akışını maksimize eden yapısı nedeniyle **DenseNet121** mimarileri kullanılmıştır. Modellerin güvenilirliğini artırmak için **5-Katlı Çapraz Doğrulama (5-Fold Cross Validation)** yöntemi ile testler gerçekleştirilmiştir.

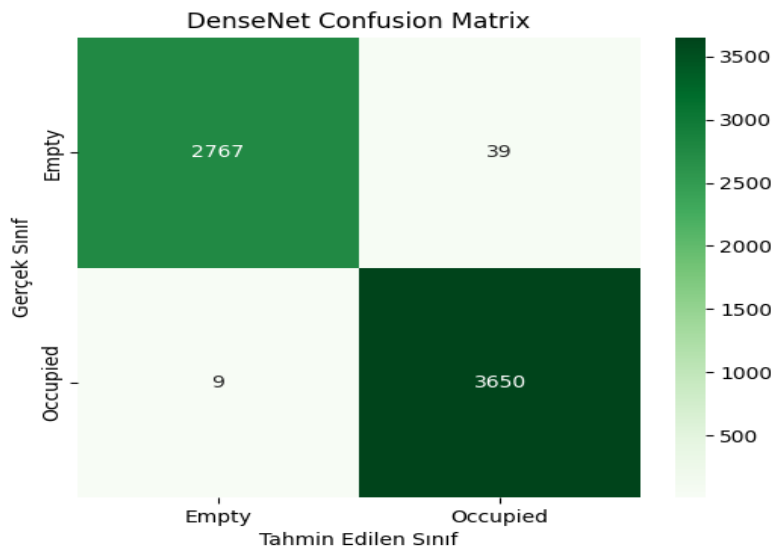
### 3.4.1. Deneyisel Sonuçlar

Deneyisel sonuçlar incelendiğinde, **DenseNet121** modelinin **%99.34** doğruluk oranı ile VGG16 (%99.10)'ya göre daha kararlı ve yüksek bir performans sergilediği görülmüştür.



Şekil 8: DenseNet121 modelinin eğitim ve 5-Fold doğrulama grafikleri.

Aşağıdaki karmaşıklık matrisinde görüldüğü üzere, DenseNet121 modeli binlerce test görüntüsü arasından sadece **33 hata** yapmıştır.



Şekil 9: DenseNet121 modelinin Karmaşıklık Matrisi.



## 4. BÖLÜM: PERFORMANS KARŞILAŞTIRMASI

Modellerin tüm test seti üzerindeki performansları aşağıdaki tablolarda karşılaştırılmıştır.

**Tablo 1: Genel Model Performans Karşılaştırması**

<i>Model Adı</i>	<i>Öğrenci</i>	<i>Accuracy</i>	<i>F1-Score</i>	<i>Hata Sayısı</i>
<i>ShuffleNet V2</i>	<b>Yunus Emre Edizer</b>	<b>%99.96</b>	<b>0.999</b>	<b>2</b>
<i>YOLOv8-cls</i>	Yunus Emre Edizer	%99.94	0.999	~4
<i>ResNet18</i>	Abdulkadir Gedik	%99.92	0.9992	~5
<i>InceptionV3</i>	M. Emin Çapan	%99.86	0.9986	~9
<i>MobileNetV3</i>	Abdulkadir Gedik	%99.80	0.9980	~13
<i>EfficientNet-B0</i>	M. Emin Çapan	%99.64	0.9964	~23
<i>DenseNet121</i>	Hamza Hakverir	%99.34	0.9934	33
<i>VGG16</i>	Hamza Hakverir	%99.10	0.9910	~58

**Tablo 2: Sınıf Bazlı Performans Tablosu (En İyi 4 Model)**

<i>Model</i>	<i>Sınıf</i>	<i>Precision</i>	<i>Recall</i>
<i>ShuffleNet V2</i>	Empty / Occupied	0.999 / 1.000	1.000 / 0.999
<i>ResNet18</i>	Empty / Occupied	0.9992 / 0.9992	0.9992 / 0.9992
<i>InceptionV3</i>	Empty / Occupied	1.0000 / 0.9975	0.9968 / 1.0000
<i>DenseNet121</i>	Empty / Occupied	0.9982 / 0.9924	0.9900 / 0.9986

## 5. BÖLÜM: SONUÇ VE TARTIŞMA

**En İyi Model:** Yapılan analizler sonucunda **ShuffleNet V2**, %99.96 doğruluk ve sadece 2 hata ile projenin **en iyi modeli** seçilmiştir. Bu başarının temel sebebi, modelin düşük parametre sayısına rağmen otopark alanlarındaki ayırt edici özellikleri yüksek hassasiyetle yakalayabilmesidir. Özellikle hafif mimarisi, gerçek zamanlı kenar cihaz uygulamaları için bu modeli rakipsiz kılmaktadır.

## İkinci ve Üçüncü Modeller:

- **En İyi 2. Model: YOLOv8-cls (%99.94).** Modern mimarisi sayesinde çok hızlı yakınsamış ve test verilerinde yüksek güven skorları üretmiştir.
- **En İyi 3. Model: ResNet18 (%99.92).** Derin artık bağlantıları sayesinde eğitim sürecinde yüksek stabilite sergilemiş ve her iki sınıfta dengeli bir başarımla sonuçlanmıştır.

**Hata Analizi:** Hataların çoğunlukla gölgeli alanlardaki "Boş" park yerlerinin "Dolu" olarak algılanmasından (Yanlış Pozitif) kaynaklandığı gözlemlenmiştir.

## 6. BÖLÜM: KAYNAKÇA

Almeida, P. R. L., Oliveira, L. S., Britto Jr, A. S., Silva, E. J., & Koerich, A. L. (2015). PKLot – A robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11), 4937-4949.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700-4708.

Jocher, G., et al. (2023). *Ultralytics YOLOv8 Docs*. Retrieved from <https://docs.ultralytics.com>.