

# Beyin Tümörü MRI Sınıflandırma Projesi

## Yöntem ve Süreç Raporu

Hazırlayan: Berke Kutay Karakoç

NO:220707099

### 1. Projenin Amacı ve Önemi

Bu projenin amacı, beyin tümörü içeren MRI görüntülerini sınıflandırmak hastalık tespitine destek olabilecek bir yapay zeka sistemi geliştirmektir. Beyin tümörleri erken fark edildiğinde tedavi seçenekleri artabildiği için, görüntüler üzerinden otomatik sınıflandırma yapmak hem zaman kazandırabilir hem de doktorların karar verme sürecine yardımcı olabilir. Bu çalışmada hedefim, veri setini düzenli bir şekilde inceleyip (EDA), uygun ön işleme adımlarını uygulamak, farklı modelleri denemek ve en başarılı modeli seçip yeni gelen bir MRI görüntüsü için tahmin üretebilen bir sisteme dönüştürmek oldu.

### 2. Kullanılan Veri Seti

Çalışmada beyin tümörü sınıflandırması için MRI görüntülerinden oluşan bir veri seti kullanılmıştır. Veri seti, eğitim (train), doğrulama (validation) ve test (test) olarak üçe ayrılmıştır. Sınıflar, veri setinin yapısına göre "glioma", "meningioma", "pituitary" ve "no\_tumor" şeklindedir. Verilerin bu şekilde ayrılması, modelin öğrenme süreci sırasında genel performansın daha doğru ölçülmesini sağlar.

### 3. Keşifsel Veri Analizi (EDA)

Model eğitimi önce veri setini tanıtmak için keşifsel veri analizi (EDA) yapılmıştır. Bu kısımda aşağıdaki kontrolleri gerçekleştirdim:

- Klasör yapısının kontrolü: Train/validation/test ayrımı doğru mu, sınıf klasörleri eksiksiz mi?
- Sınıf başına görüntü sayılarının hesaplanması: Veri dengesizliği (class imbalance) var mı?
- Rastgele örnek görüntü inceleme: MRI görüntülerinde çözünürlük, kontrast ve gürültü farklılıklarını var mı?
- Etiket/sınıf isimlerinin doğrulanması: Sınıf isimleri ve sıralaması model çıktılarıyla tutarlı mı?

EDA sonucunda bazı sınıfların diğerlerine göre daha az görüntüye sahip olabileceği görüldü. Bu durum, modelin az örnek olan sınıflarda daha zor öğrenmesine neden olabildiği için ilerleyen aşamalarda veri artırma ve düzenleme adımlarına dikkat ettim.

## 4. Ön İşleme (Preprocessing)

MRI görüntülerini modele vermeden önce ortak bir formata getirmek için ön işleme adımları uyguladım. Temel olarak amaç, tüm görüntülerin aynı boyuta getirip piksel değerlerini daha uygun bir aralığa çekerek modelin daha stabil öğrenmesini sağlamaktır.

- Görüntü boyutlandırma (resize): Model giriş boyutuna uygun olacak şekilde (ör. 224x224).
- Normalize etme: Piksel değerlerini 0-1 aralığına ölçekleme.
- Gerekli durumlarda kanal uyumluluğu: Bazı MRI görüntülerini tek kanallı olabildiği için modele uygun hale getirme.

## 5. Uygulanan Yöntemler ve Model Yaklaşımları

Projede hem derin öğrenme tabanlı yöntemler hem de klasik makine öğrenmesi yöntemleri denenmiştir. Farklı yöntemleri denememin amacı, aynı veri üzerinde hangi yaklaşımın daha iyi genelleme yaptığına bakmaktır.

### 5.1 Derin Öğrenme (CNN) Modelleri

Derin öğrenme tarafından temel yaklaşım, MRI görüntülerinden özellikleri otomatik çıkarıran evrişimsel sinir ağlarını (CNN) kullanmaktadır. Bu kapsamda transfer öğrenme (pretrained modeller) ve/veya kendi kurduğum temel CNN yapıları denenmiştir. Eğitim sırasında doğrulama seti sonuçlarını takip ederek modelin aşırı öğrenmeye gitmediğini kontrol ettim.

### 5.2 Hibrit Yaklaşım: CNN Feature Extraction + Klasik Sınıflandırıcı

Kullandığım ana yöntemlerden biri hibrit yaklaşımıdır. Bu yöntemde CNN modeli doğrudan sınıflandırma yapmak için değil, görüntülerden anlamlı özellik vektörleri çıkarma (feature extraction) amacıyla kullanıldı. Daha sonra bu özellikler ile SVM gibi klasik sınıflandırıcılar eğitildi. Bu yaklaşımın avantajı, CNN'in güçlü temsil (feature) üretme yeteneğini, klasik modellerin daha kontrol edilebilir karar sınırlarıyla birleştirmesidir.

## 6. Aşırı Öğrenmeyi (Overfitting) Önleme Çalışmaları

Overfitting, modelin eğitim verisini ezberleyip doğrulama/test verisinde başarısının düşmesi durumudur. Bu projede overfitting'i önlemek için şu noktalara dikkat ettim:

- Eğitim ve doğrulama sonuçlarını birlikte takip ettim (accuracy/loss eğrileri üzerinden).
- Veri artırma (data augmentation) ile eğitim setini çeşitlendirdim (döndürme, yakınlaştırma, küçük kaydırırmalar gibi).
- Model karmaşıklığını kontrol ettim: Gerektiğinde daha küçük model/katman sayısı tercih ettim veya dropout gibi düzenlileştirme yöntemleri kullandım.
- Erken durdurma (early stopping) mantığı: Doğrulama başarısı artmıyorsa eğitim süresini uzatmamaya dikkat ettim.

## 7. Parametre Optimizasyonu (Hiperparametre Ayarı)

Parametre optimizasyonu, modelin ayarlarını (hiperparametrelerini) farklı değerlerle deneyerek en iyi sonucu veren kombinasyonu seçme sürecidir. Bu işlem, hem performansı artırmak hem de overfitting riskini azaltmak için önemlidir.

Bu projede parametre optimizasyonunu şu şekilde ele aldım:

- CNN tarafında: öğrenme oranı (learning rate), epoch sayısı, batch size gibi ayarları doğrulama sonuçlarına göre ayarladım.
- Klasik model tarafında (özellikle SVM): C, kernel ve gamma gibi hiperparametrelerin farklı değerlerini deneyerek en iyi sonucu veren ayarları seçtim (Grid Search/benzeri denemeler).

## 8. Model Karşılaştırması ve En İyi Modelin Seçilmesi

Farklı yöntemleri denedikten sonra her modelin performansını aynı ölçütlerle karşılaştırdım. Değerlendirme için doğruluk (accuracy) başta olmak üzere, sınıflar arası dengesizlik olabileceği için F1-Score ve Recall gibi metrikleri de dikkate aldım. Tüm sonuçlar düzenli şekilde bir tabloya/CSV dosyasına kaydedildi (ör. model\_results.csv). Karşılaştırma sonucunda en iyi performansı veren yaklaşım seçilerek son sistem bu model üzerine kuruldu.

## 9. En İyi Modelin Kaydedilmesi ve Yeni Örnek Üzerinde Tahmin

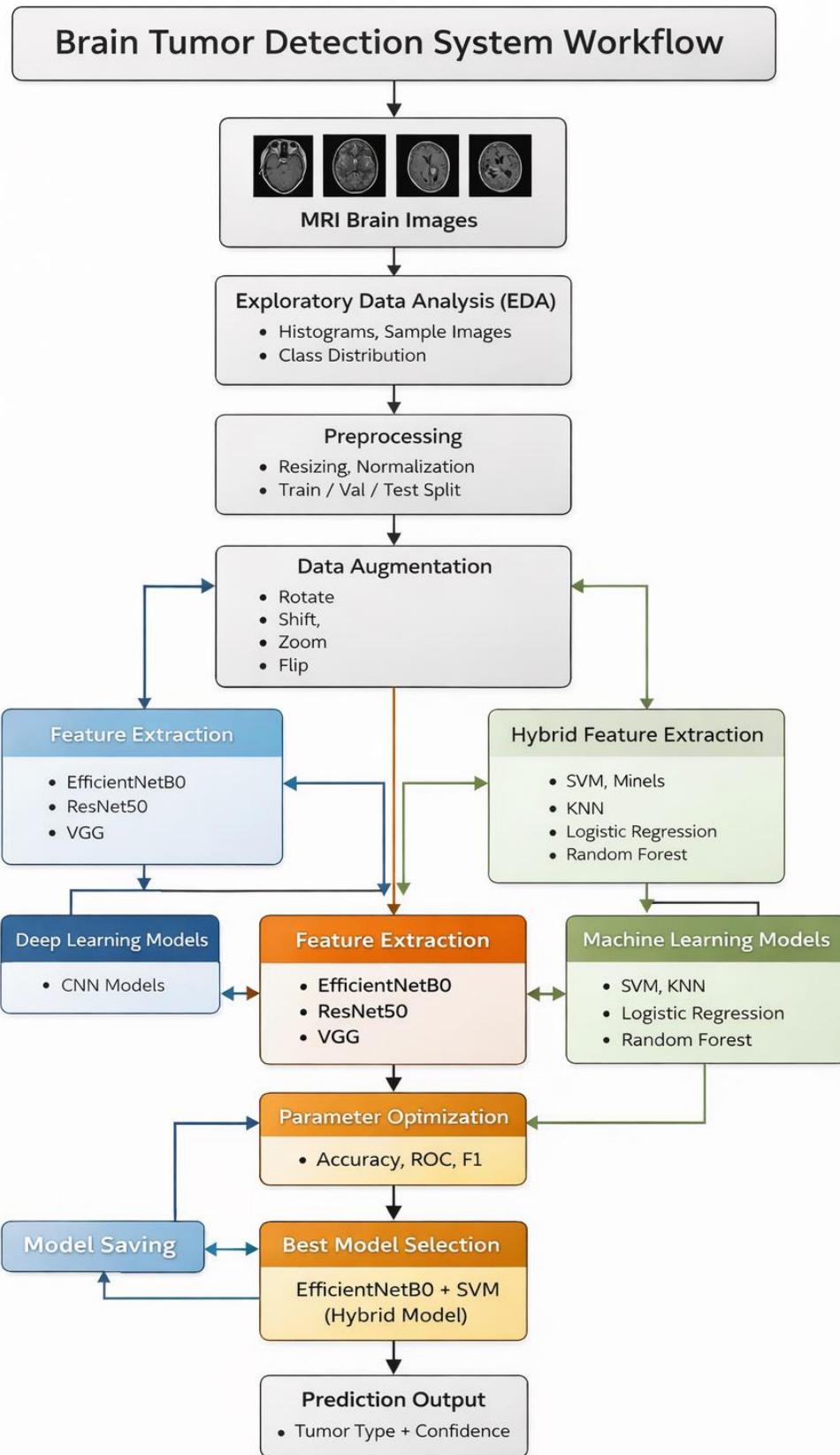
Projenin teslim kısmı için en önemli adımlardan biri, en iyi modelin kaydedilip daha sonra tekrar yüklenmesidir. Hibrit yaklaşım kullanıldığında iki parça ayrı ayrı düşünülmelidir: (1) özellik çıkarıcı CNN, (2) sınıflandırıcı (SVM). Bu nedenle seçilen yöntemde her iki bileşeni de kaydedilecek şekilde tasarladım.

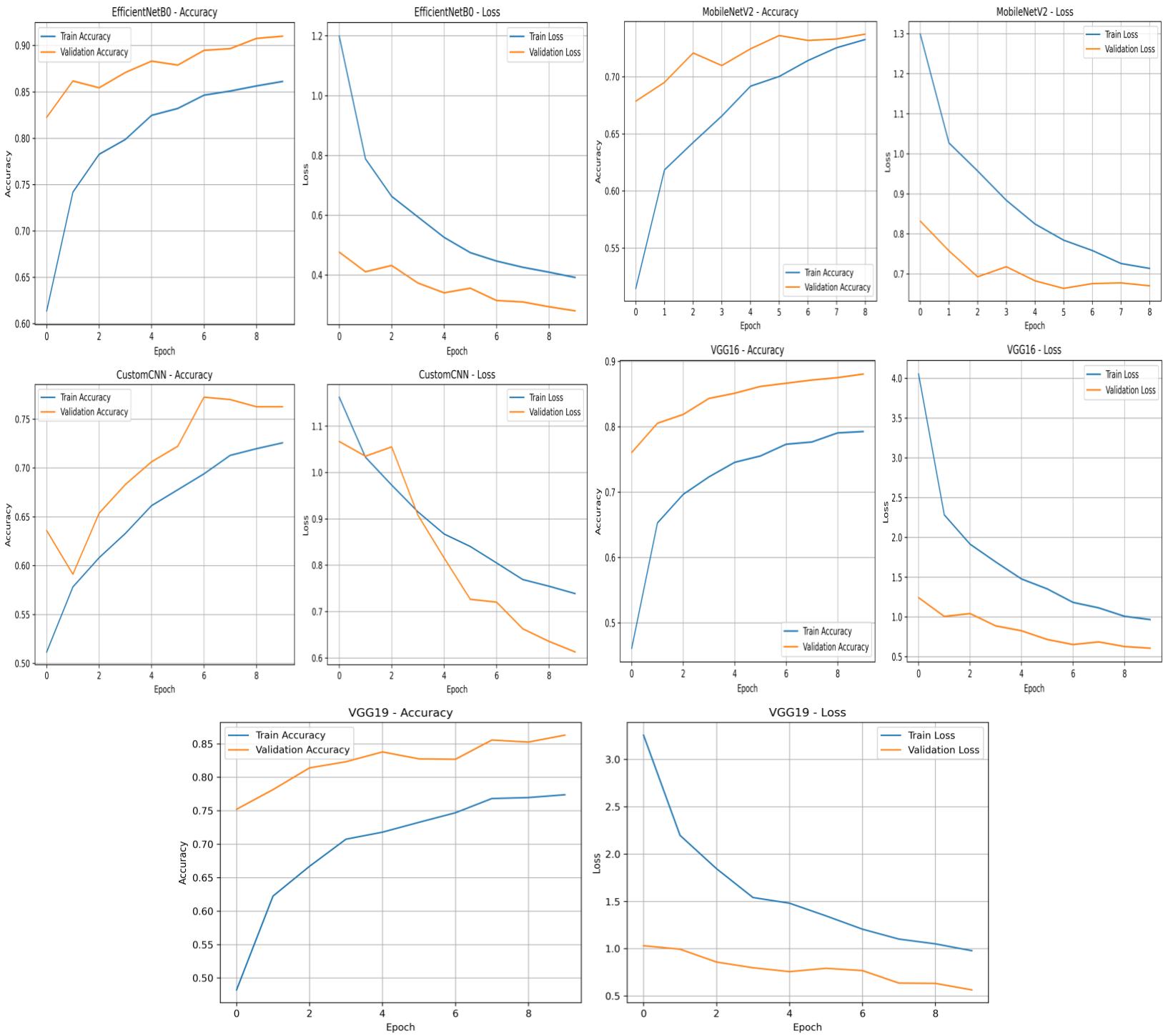
Yeni bir MRI görüntüsü geldiğinde tahmin süreci şu sırayla çalışır:

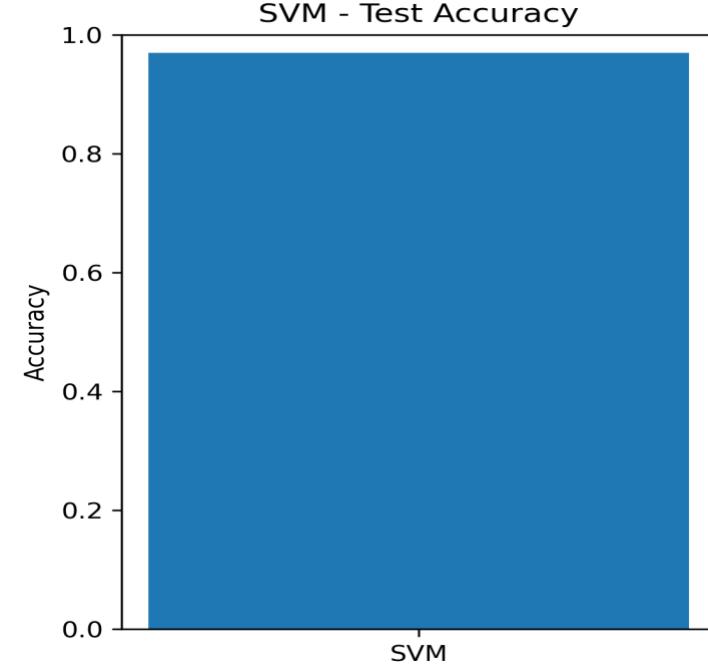
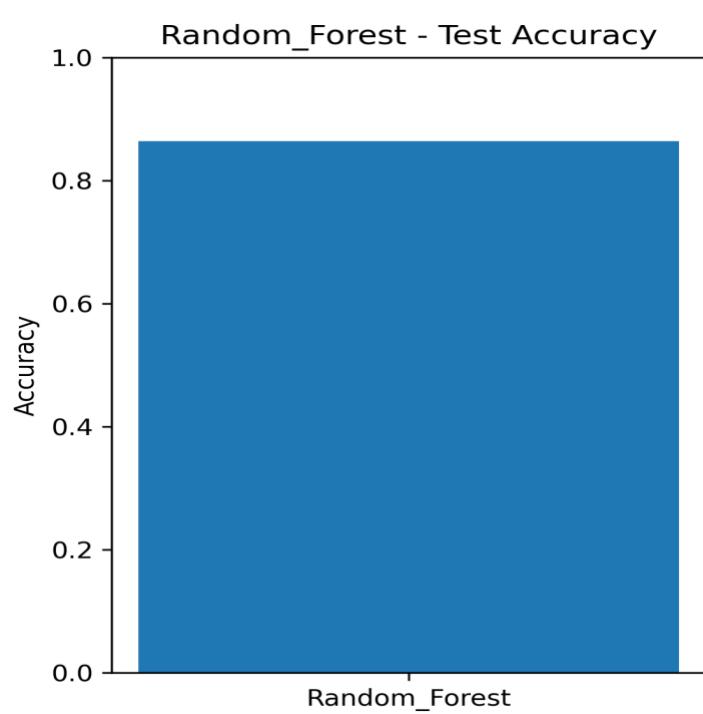
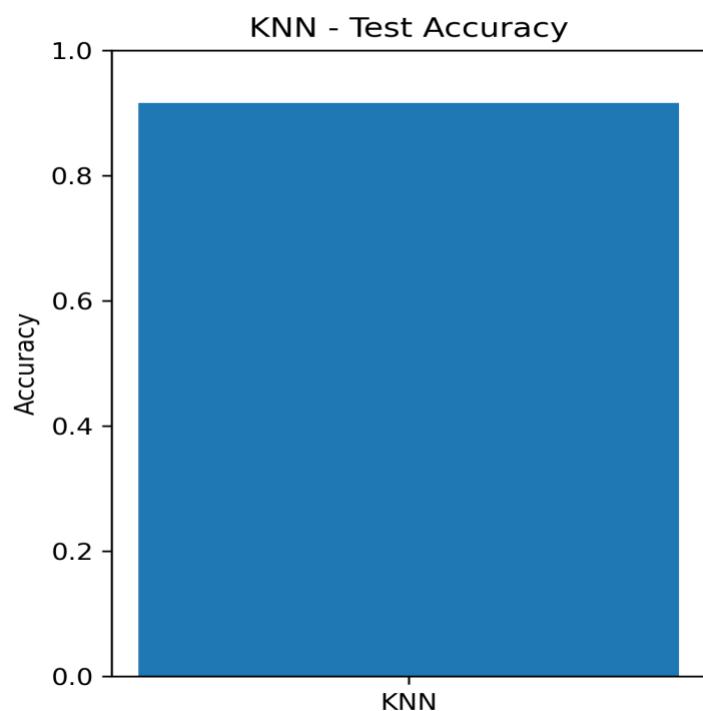
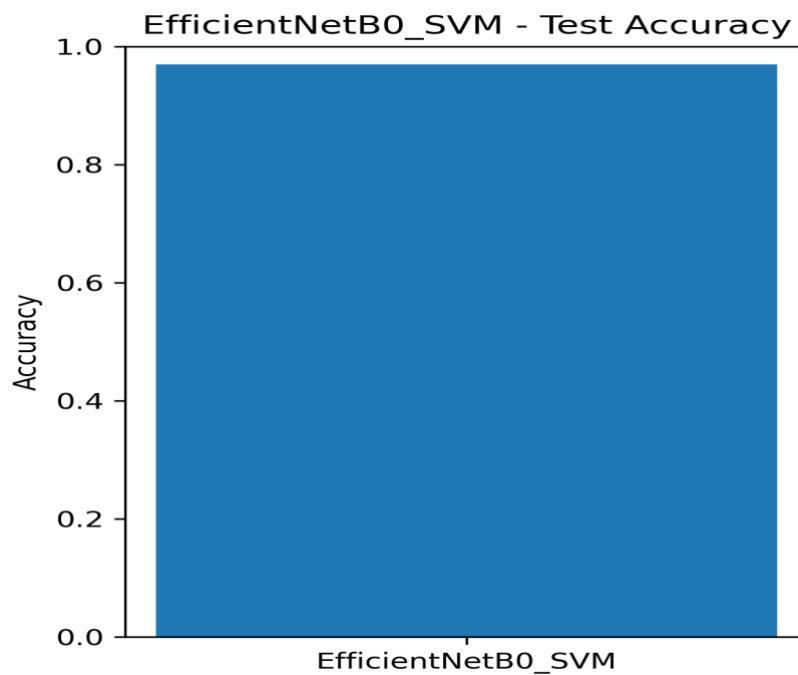
1. Görüntü ön işleme: boyutlandırma ve normalize etme.
2. CNN ile özellik çıkarma: görüntünün sayısal bir feature vektörüne dönüştürülmesi.
3. SVM ile sınıflandırma: feature vektörü üzerinden sınıf tahmini ve güven skoru üretme.
4. Sonucun kullanıcıya sunulması: tahmin edilen sınıf (ör. glioma) ve güven değeri.

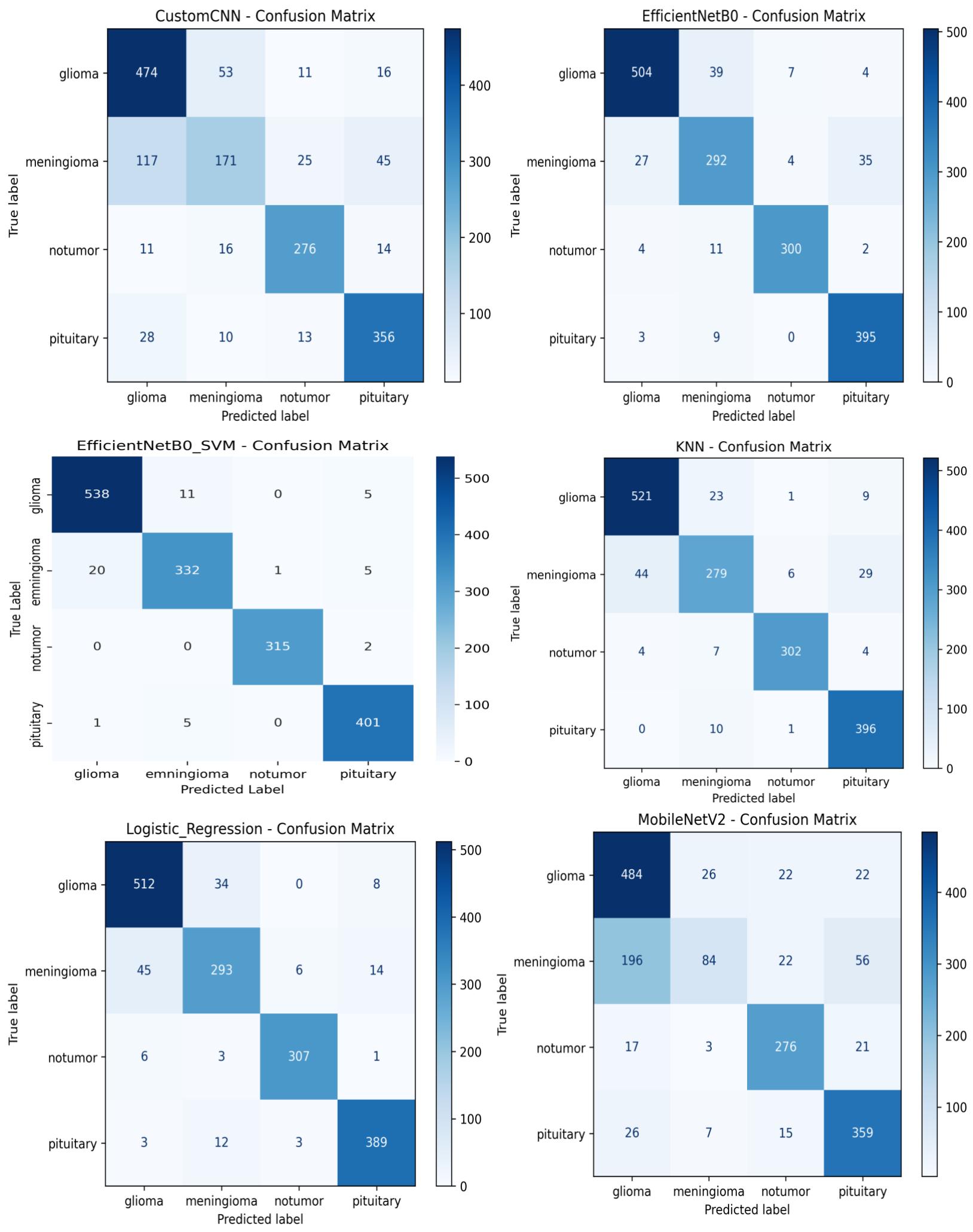
## 10. Sonuç ve Kısa Değerlendirme

Bu projede beyin tümörü MRI görüntülerini sınıflandırmak için veri analizi, ön işleme, model geliştirme, overfitting önleme ve parametre optimizasyonu adımlarını sırayla uyguladım. Farklı modelleri denedikten sonra en iyi performansı veren yöntemi seçip kaydederek, yeni görüntüler üzerinde tahmin yapabilen bir pipeline oluşturdum. İleride veri seti büyütülerek ve sınıflar daha dengeli hale getirilerek modelin özellikle zor sınıflarda daha iyi sonuç vermesi hedeflenebilir. Ayrıca açıklanabilirlik yöntemleri (ör. Grad-CAM) eklenerek modelin hangi bölgelere bakarak karar verdiği de gösterilebilir.

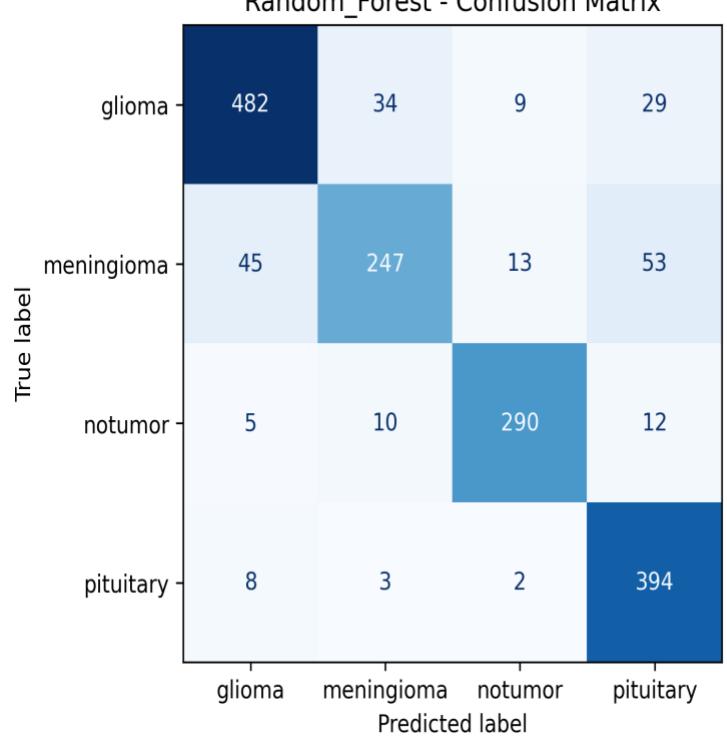




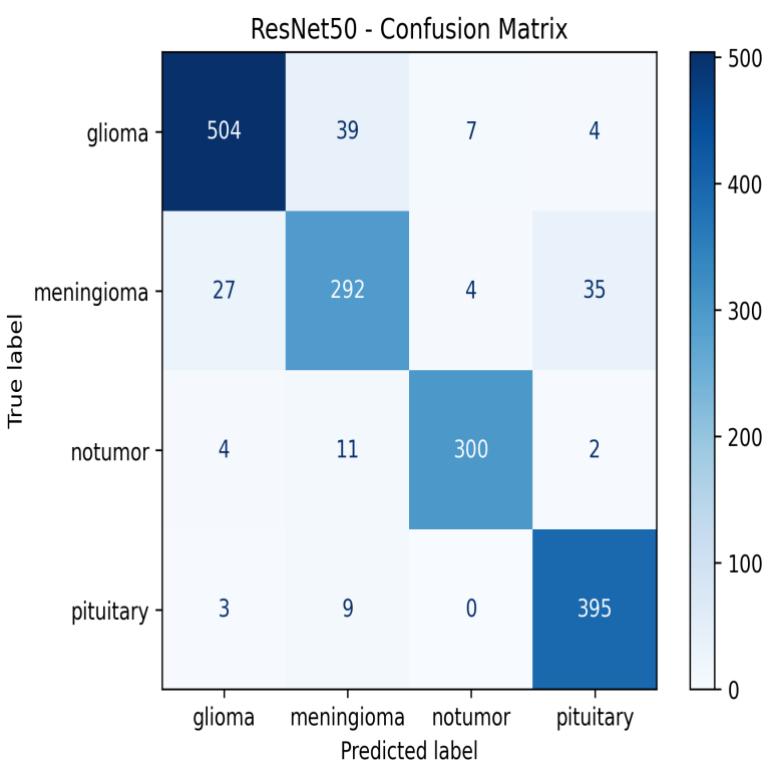




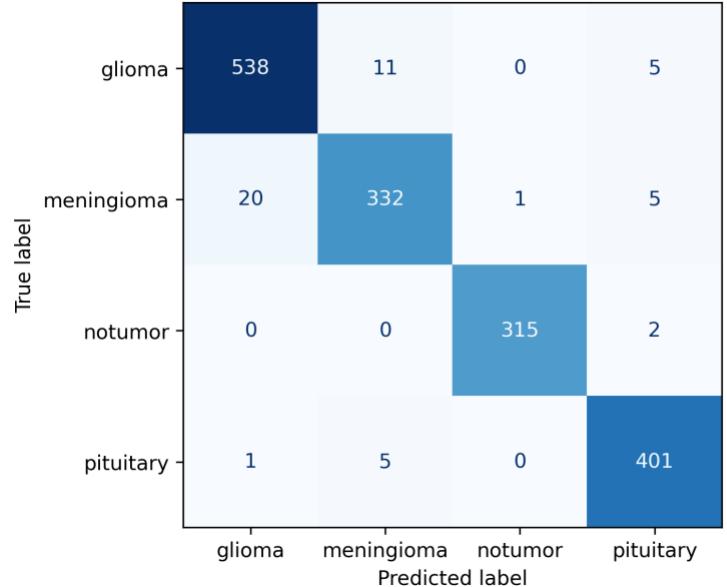
Random\_Forest - Confusion Matrix



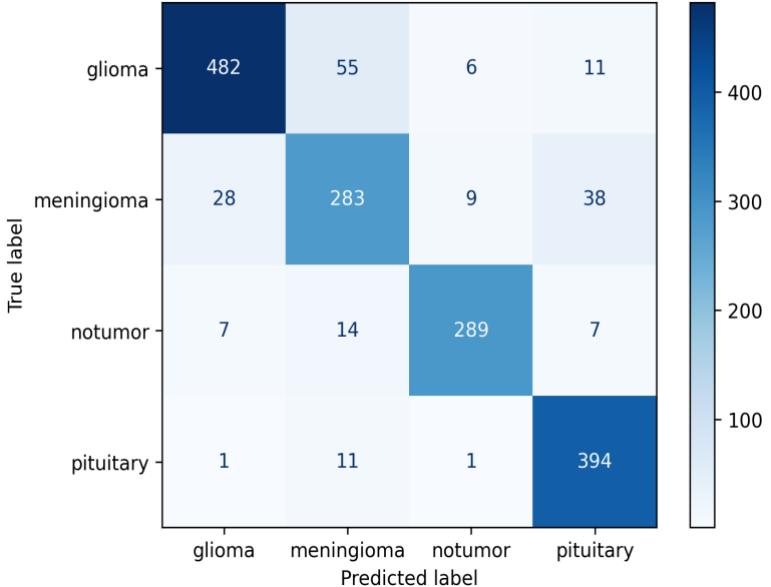
ResNet50 - Confusion Matrix



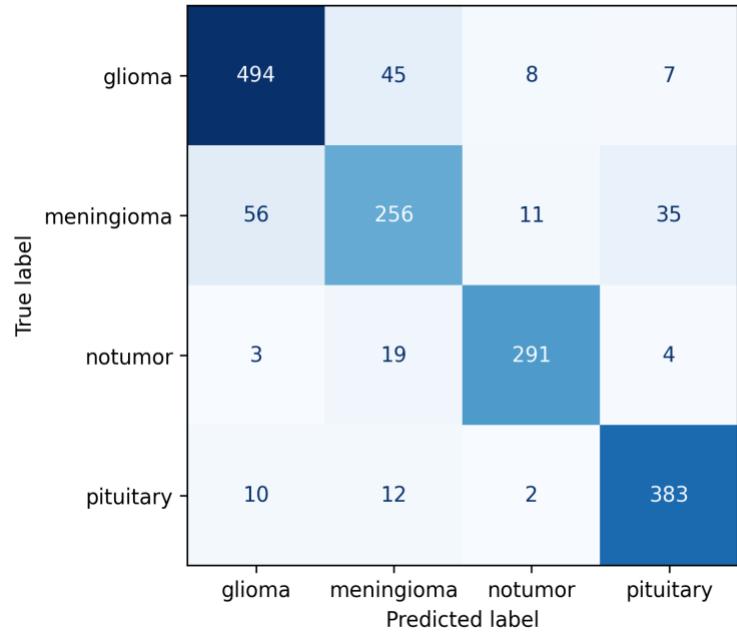
SVM - Confusion Matrix



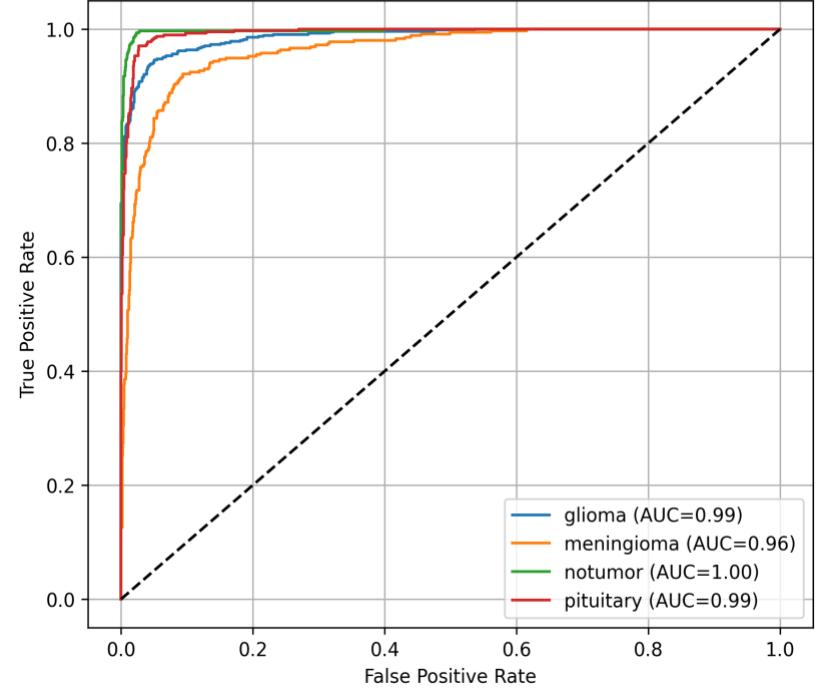
VGG16 - Confusion Matrix



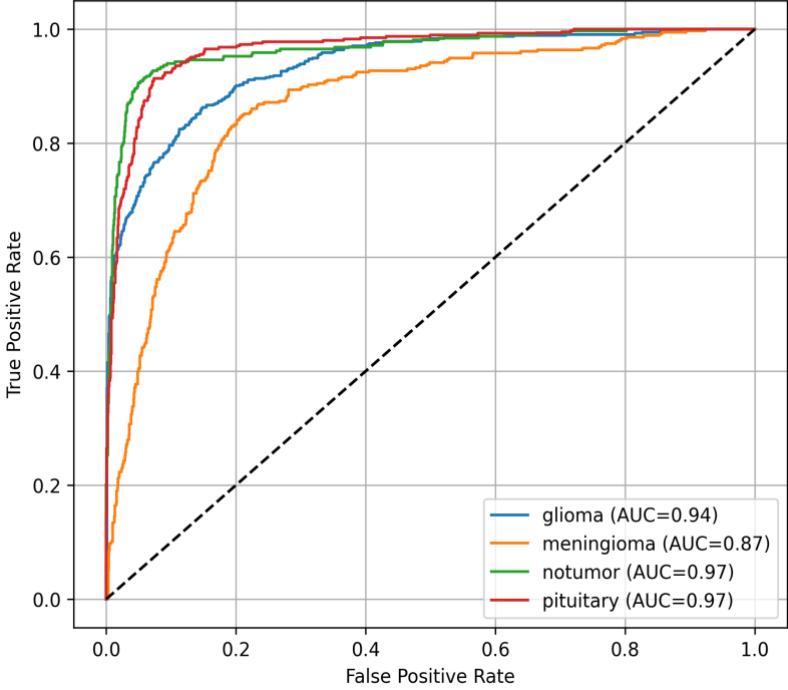
VGG19 - Confusion Matrix



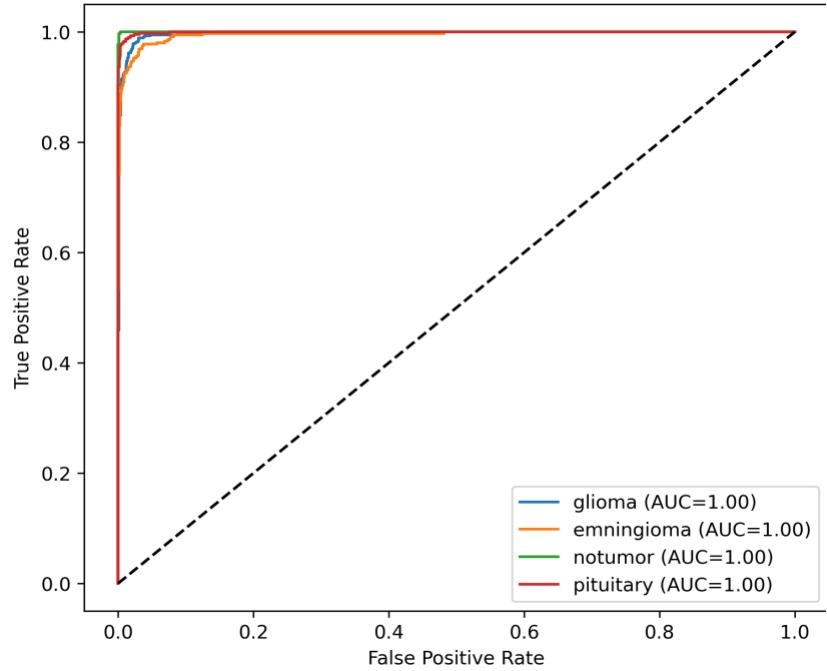
EfficientNetB0 - ROC Curves



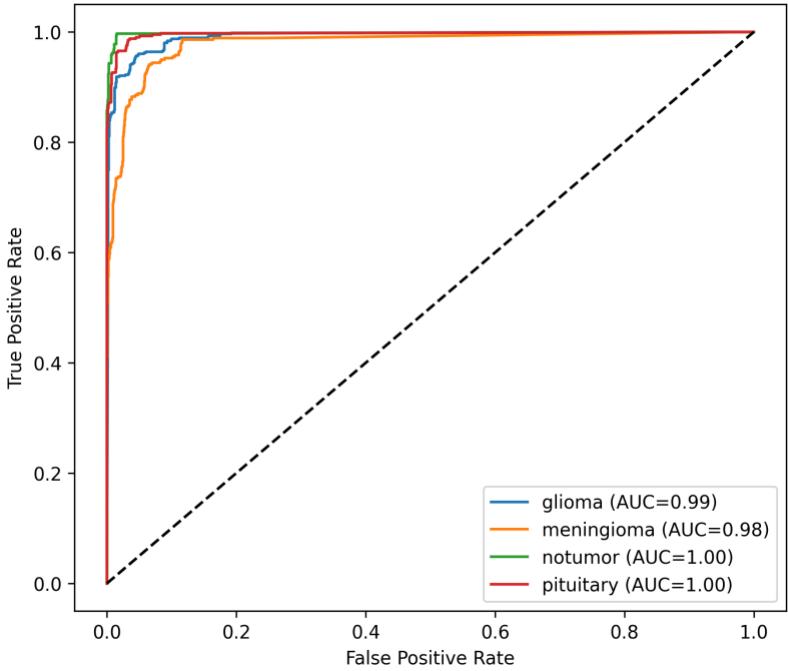
CustomCNN - ROC Curves



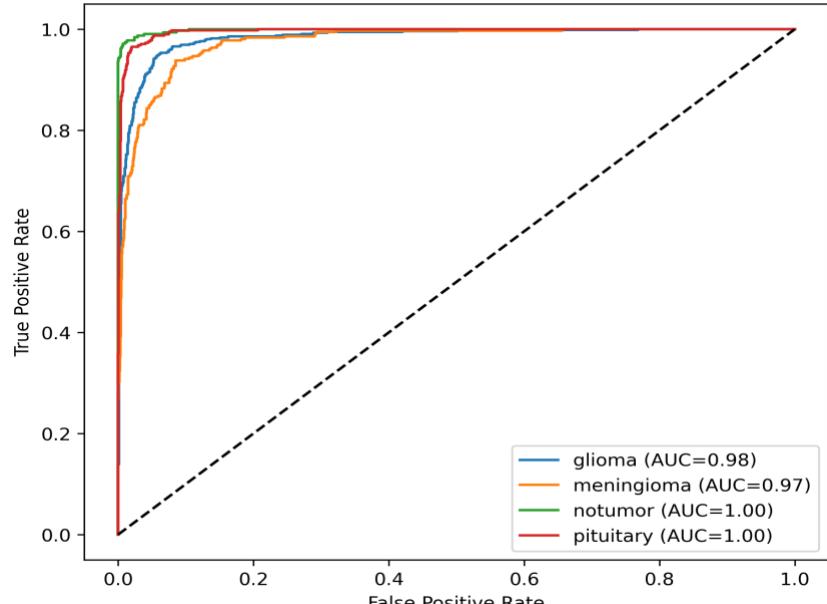
EfficientNetB0\_SVM - ROC Curves



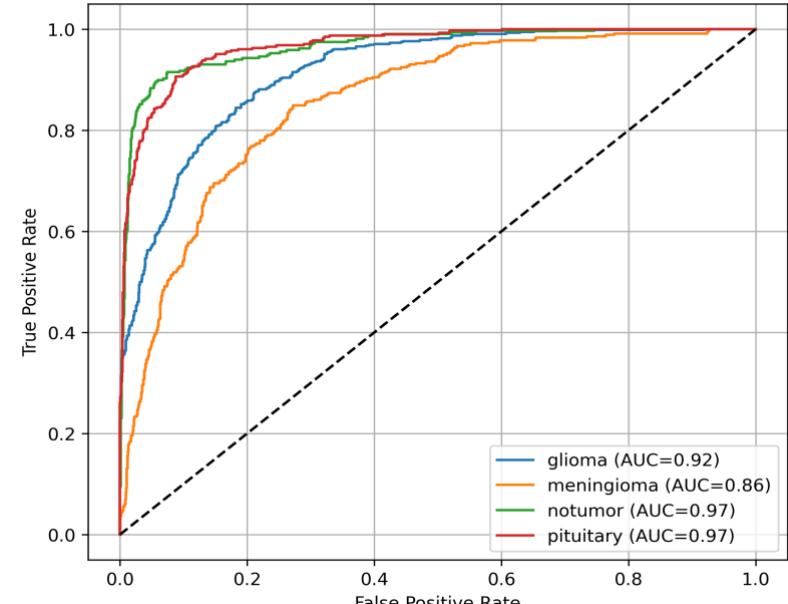
KNN - ROC Curves



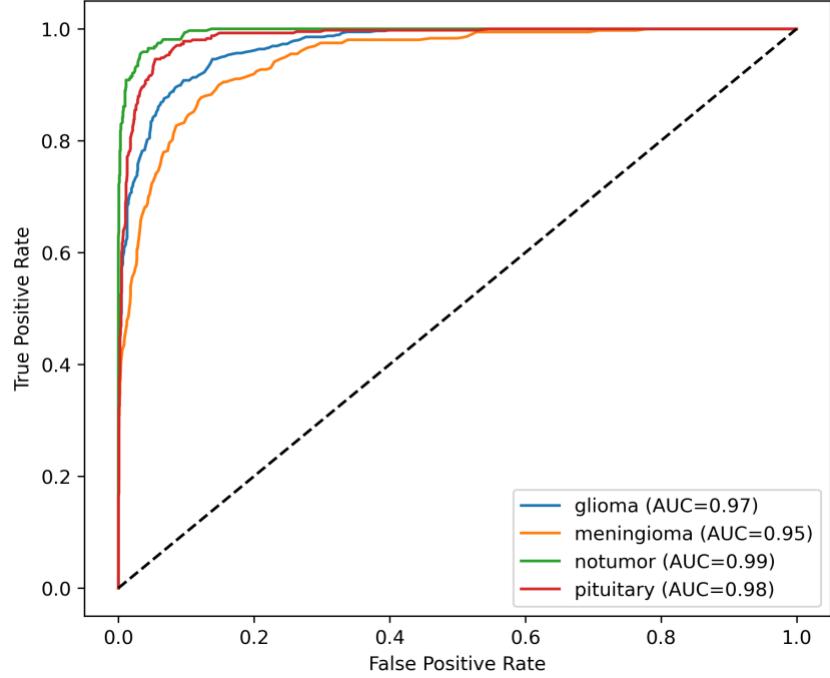
Logistic\_Regression - ROC Curves



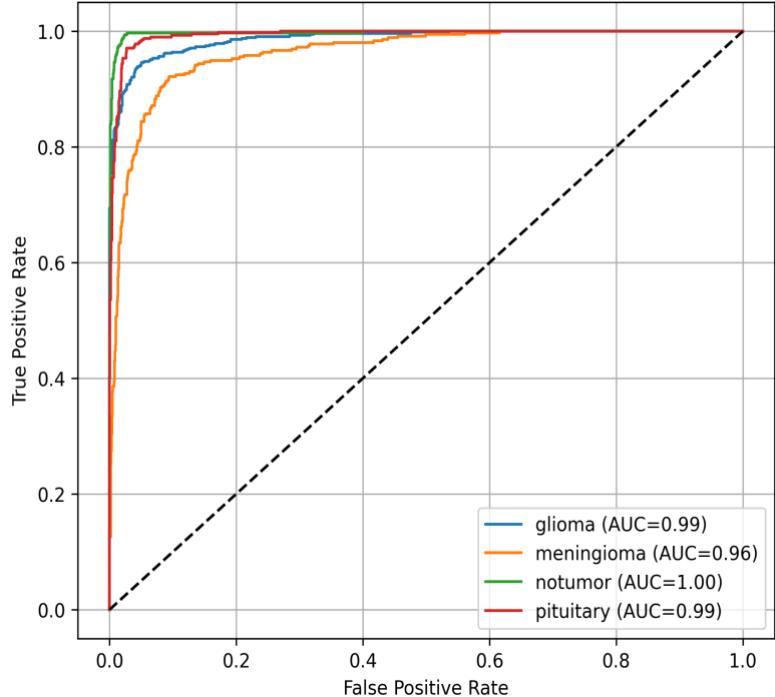
MobileNetV2 - ROC Curves



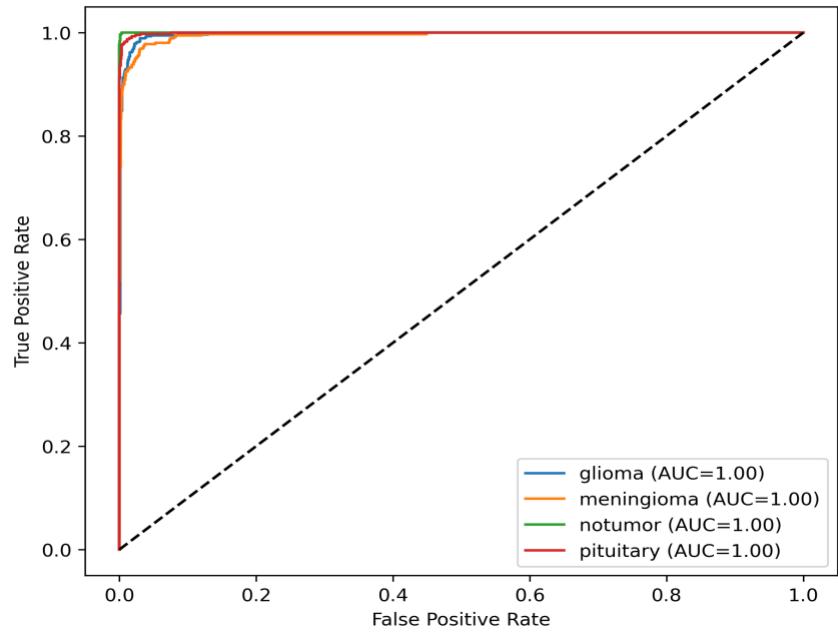
Random\_Forest - ROC Curves



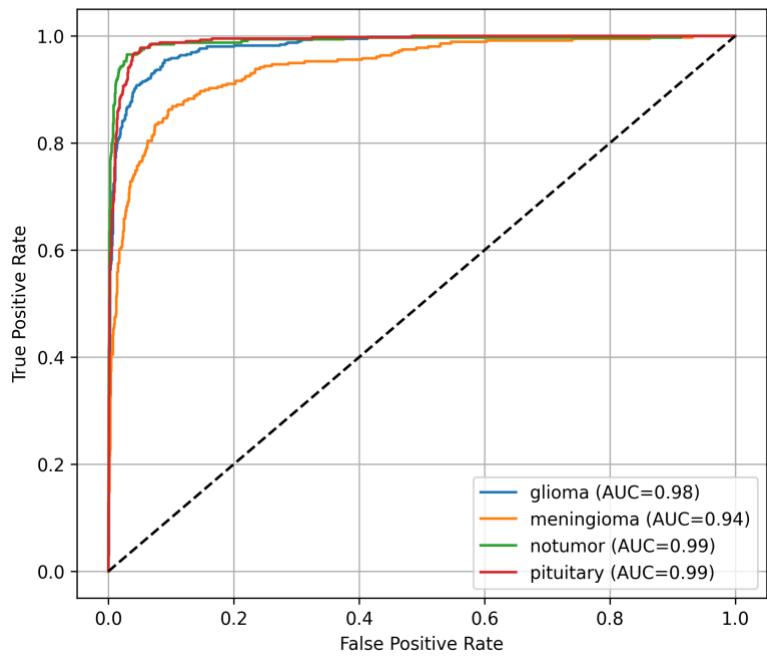
ResNet50 - ROC Curves



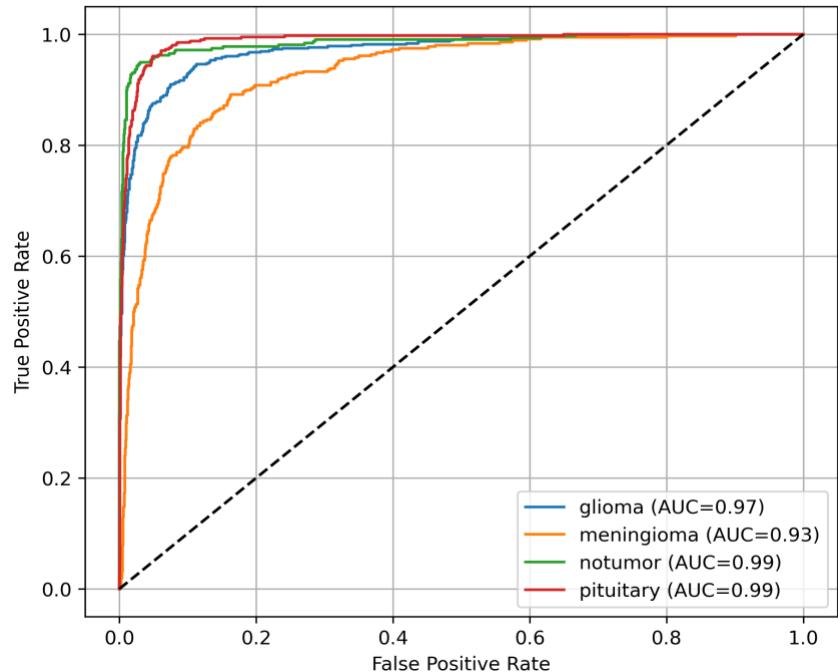
SVM - ROC Curves



VGG16 - ROC Curves



VGG19 - ROC Curves



Classifier	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Kappa	
0	EfficientNetB0	88.536696	88.630807	88.519648	88.630807	0.846043
1	ResNet50	91.156910	91.136919	91.109133	91.136919	0.879995
2	VGG16	91.156910	91.136919	91.109133	91.136919	0.879995
3	VGG19	86.868805	87.041565	86.913223	87.041565	0.824142
4	MobileNetV2	73.451100	73.533007	70.414726	73.533007	0.634926
5	CustomCNN	77.449483	78.056235	77.219705	78.056235	0.700274
6	Logistic Regression	91.693345	91.748166	91.706698	91.748166	0.888038
7	KNN	91.513879	91.564792	91.439923	91.564792	0.885428
8	SVM	96.937606	96.943765	96.934544	96.943765	0.958561
9	Random Forest	86.591575	86.369193	86.132901	86.369193	0.815177
10	EfficientNetB0_SVM	96.937606	96.943765	96.934544	96.943765	0.958561

## **Accurcay Loss Grafikleri Yorumlanması**

### **EfficientNetV0**

EfficientNetV0 modeli eğitim ve doğrulama sürecinde istikrarlı bir performans sergilemiştir. Doğruluk ve kayıp değerlerinin dengeli ilerlemesi, modelin veriyi iyi genelleştirdiğini göstermektedir.

---

### **MobileNetV2**

MobileNetV2 modeli hafif yapısına rağmen doğrulama verisi üzerinde başarılı sonuçlar üretmiştir. Eğitim süreci genel olarak stabil olup belirgin bir aşırı öğrenme gözlemlenmemiştir.

---

### **Custom CNN**

Özel olarak geliştirilen CNN modeli, eğitim boyunca doğruluk değerlerini artırmış ve doğrulama verisiyle uyumlu sonuçlar elde etmiştir. Modelin öğrenme süreci dengeli ilerlemiştir.

---

### **VGG16**

VGG16 modeli doğrulama verisi üzerinde yüksek ve stabil bir performans göstermiştir. Ancak modelin hesaplama maliyeti diğer modellere göre daha yüksektir.

---

### **VGG19**

VGG19 modeli derin yapısı nedeniyle daha yavaş öğrenmiş, ancak genel olarak kabul edilebilir doğruluk değerleri elde etmiştir. Bu veri seti için VGG16'ya göre belirgin bir üstünlük sağlamamıştır.

---

### **EfficientNetB0 + SVM**

EfficientNetB0'dan çıkarılan özellikler ile eğitilen SVM modeli test verisi üzerinde yüksek doğruluk elde etmiştir. Bu sonuç, derin özelliklerin klasik makine öğrenmesi algoritmalarıyla birlikte etkili şekilde kullanılabildiğini göstermektedir.

---

## KNN

KNN modeli test verisi üzerinde kabul edilebilir bir doğruluk sağlamıştır. Ancak performansı, derin özellik kullanan diğer modellere kıyasla biraz daha sınırlı kalmıştır.

---

## Logistic Regression

Logistic Regression modeli test doğruluğu açısından dengeli bir sonuç üretmiştir. Modelin doğrusal yapısına rağmen özellik çıkarımı sonrası makul bir performans gösterdiği görülmüştür.

---

## Random Forest

Random Forest modeli test verisi üzerinde diğer klasik modellere göre biraz daha düşük bir doğruluk elde etmiştir. Bu durum, modelin veri dağılımına daha duyarlı olmasından kaynaklanmış olabilir.

---

## SVM

SVM modeli test doğruluğu açısından güçlü sonuçlar üretmiştir. Özellikle yüksek boyutlu özellikler üzerinde başarılı bir sınıflandırma performansı sergilemiştir.

## **Confusion Matrix Yorumlanması**

### **Custom CNN – Confusion Matrix**

Custom CNN modeli tüm sınıflarda genel olarak doğru tahminler üretmiştir. Hatalar çoğunlukla birbirine benzer tümör türleri arasında gerçekleşmiştir.

---

### **EfficientNetB0 – Confusion Matrix**

EfficientNetB0 modeli sınıflar arasında yüksek ayırt edicilik göstermiştir. Yanlış sınıflandırma sayısı düşüktür ve sınıflar dengeli şekilde tahmin edilmiştir.

---

### **EfficientNetB0 + SVM – Confusion Matrix**

Derin özellikler ile eğitilen SVM modeli sınıfların büyük kısmını doğru şekilde ayırmıştır. Özellikle glioma ve pituitary sınıflarında yüksek başarı elde edilmiştir.

---

### **KNN – Confusion Matrix**

KNN modelinde doğru sınıflandırmalar çoğunlukta olsa da bazı sınıflar arasında karışmalar gözlemlenmiştir. Bu durum, yöntemin komşu örneklerle duyarlı olmasından kaynaklanmaktadır.

---

### **Logistic Regression – Confusion Matrix**

Logistic Regression modeli sınıflar arasında genel olarak tutarlı sonuçlar üretmiştir. Ancak doğrusal yapısı nedeniyle bazı sınıflarda karışmalar görülmüştür.

---

### **MobileNetV2 – Confusion Matrix**

MobileNetV2 modeli bazı sınıflarda başarılı tahminler yaparken, meningioma sınıfında görece daha fazla hata üretmiştir. Genel performansı kabul edilebilir düzeydedir.

---

### **Random Forest – Confusion Matrix**

Random Forest modeli sınıfların çoğunu doğru tahmin etmiş ancak bazı sınıflar arasında belirgin karışmalar oluşmuştur. Bu durum modelin veri dağılımına duyarlığını göstermektedir.

---

### **ResNet50 – Confusion Matrix**

ResNet50 modeli sınıflar arasında yüksek doğrulukla ayrımlı yapmıştır. Yanlış sınıflandırmalar sınırlı düzeydedir ve model dengeli bir performans sergilemiştir.

---

### **SVM – Confusion Matrix**

SVM modeli tüm sınıflarda oldukça net bir ayrımlı yapmıştır. Özellikle yüksek boyutlu özellikler üzerinde güçlü bir sınıflandırma başarısı göstermiştir.

---

### **VGG16 – Confusion Matrix**

VGG16 modeli çoğu sınıfı doğru tahmin etmiş, ancak bazı sınıflar arasında sınırlı düzeyde karışmalar oluşmuştur. Genel performansı stabildir.

---

### **VGG19 – Confusion Matrix**

VGG19 modeli doğru sınıflandırma oranı yüksek olsa da, derin yapısı nedeniyle bazı sınıflarda hata oranı artmıştır. Bu veri seti için VGG16'ya kıyasla belirgin bir üstünlük sağlamamıştır.

## ROC Eğrileri Yorumlanması

### EfficientNetB0 – ROC Eğrileri

EfficientNetB0 modeli tüm sınıflar için ROC eğrilerinde yüksek AUC değerleri elde etmiştir. Bu durum modelin sınıfları ayırt etme başarısının oldukça yüksek olduğunu göstermektedir.

---

### Custom CNN – ROC Eğrileri

Custom CNN modelinde ROC eğrileri genel olarak başarılıdır. Bazı sınıflarda ayırm gücünden diğerlerine göre biraz daha düşük olsa da genel performans yeterlidir.

---

### EfficientNetB0 + SVM – ROC Eğrileri

EfficientNetB0'dan çıkarılan özellikler ile eğitilen SVM modeli, ROC eğrilerinde neredeyse ideal sonuçlar üretmiştir. Sınıflar arası ayırt edicilik oldukça yüksektir.

---

### KNN – ROC Eğrileri

KNN modelinde ROC eğrileri genel olarak iyi seviyededir. Ancak bazı sınıflarda eğrinin daha yavaş yükseldiği görülmekte, bu da ayırm gücünün nispeten daha düşük olduğunu göstermektedir.

---

### Logistic Regression – ROC Eğrileri

Logistic Regression modeli ROC eğrilerinde dengeli bir performans sergilemiştir. Modelin doğrusal yapısına rağmen sınıfları ayırt edebilme yeteneği kabul edilebilir düzeydedir.

---

### MobileNetV2 – ROC Eğrileri

MobileNetV2 modelinde bazı sınıflar için ROC eğrileri diğerlerine göre daha düşüktür. Buna rağmen genel olarak modelin ayırt ediciliği yeterli seviyededir.

---

### **Random Forest – ROC Eğrileri**

Random Forest modeli ROC eğrilerinde sınıflar arasında farklılıklar göstermiştir. Bazı sınıflarda yüksek ayırt edicilik sağlanırken, bazı sınıflarda performans düşmüştür.

---

### **ResNet50 – ROC Eğrileri**

ResNet50 modeli ROC eğrilerinde yüksek AUC değerleri elde etmiştir. Bu sonuç, modelin sınıfları ayırt etmede güçlü bir performans sergilediğini göstermektedir.

---

### **SVM – ROC Eğrileri**

SVM modeli ROC eğrilerinde neredeyse kusursuz bir ayırm gücü sergilemiştir. Özellikle derin özellikler kullanıldığında oldukça başarılı sonuçlar elde edilmiştir.

---

### **VGG16 – ROC Eğrileri**

VGG16 modeli ROC eğrilerinde genel olarak yüksek bir performans göstermiştir. Bazı sınıflarda küçük düşüşler olsa da modelin ayırt ediciliği iyidir.

---

### **VGG19 – ROC Eğrileri**

VGG19 modeli ROC eğrilerinde yüksek AUC değerleri üretmiştir. Ancak VGG16'ya kıyasla belirgin bir performans artışı sağlamamıştır.

