**ÜSKÜDAR ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPAY SİNİR AĞI-FİNAL ÖDEVİ**

**Face Mask Detection Using Deep Learning**

**1. GİRİŞ**

Bu projede, derin öğrenme teknikleri kullanılarak yüz maskesi tespiti yapan bir model geliştirilmiştir. COVID-19 pandemisi sürecinde maske kullanımının önemi göz önüne alındığında, otomatik maske tespiti yapan sistemlerin geliştirilmesi önem kazanmıştır. Bu çalışmada, transfer learning yaklaşımı kullanılarak DenseNet121 ve MobileNetV2 mimarilerine dayalı iki farklı model geliştirilmiş ve performansları karşılaştırılmıştır.

**2. YÖNTEM**

**2.1 Veri Seti**

Face Mask Detection veri seti kullanılmıştır. Veri seti, maskeli ve maskesiz yüz fotoğraflarından oluşmaktadır. Görüntüler 224x224 piksel boyutuna yeniden boyutlandırılmıştır.

A computer screen with text on it

Description automatically generated

**Fig.1. Veri Organizasyonu**

**2.2 Veri Ön İşleme ve Bölümleme**

Veri seti üzerinde aşağıdaki ön işleme adımları uygulanmıştır:

1. Veri Bölümleme:
   * Toplam verinin %80'i eğitim ve validasyon için ayrılmıştır
   * Kalan %20'si test seti olarak kullanılmıştır
   * Eğitim ve validasyon için ayrılan %80'lik kısım kendi içinde:
     + %80 eğitim (toplam verinin %64'ü)
     + %20 validasyon (toplam verinin %16'sı) olarak bölünmüştür

A computer screen shot of a program code

Description automatically generated

**Fig.2. Veri Bölümleme**

1. Veri Artırma (Data Augmentation):

Verilen kod parçası, **veri artırma (data augmentation)** işlemleri için kullanılır ve modelin farklı varyasyonlara dayanıklı hale gelmesini sağlar. Bu işlem, görüntülerin çeşitli şekillerde dönüştürülmesiyle (döndürme, kaydırma, yakınlaştırma, parlaklık değiştirme, aynalama vb.) veri çeşitliliğini artırır.

**Önemli parametreler:**

* **rescale**: Görüntü piksellerini normalize eder (0-255 → 0-1).
* **Döndürme, kaydırma, yakınlaştırma**: Modelin farklı pozisyonlara ve boyutlara duyarlılığını artırır.
* **Parlaklık ve renk değişiklikleri**: Farklı ışık koşullarına dayanıklılık sağlar.
* **Yatay aynalama**: Simetrik nesneler üzerinde genelleme yeteneğini artırır.

Bu işlemler, modelin overfitting'i önlemesine ve gerçek dünya verilerinde daha iyi performans göstermesine yardımcı olur.

A screenshot of a computer program

Description automatically generated

**Fig.3. Veri Arttırma**

Veri artırma işlemi yalnızca eğitim verisi üzerinde uygulanmıştır. Validasyon verisine veri artırma işlemi uygulanmamıştır, çünkü validasyon kümesi, modelin gerçek performansını değerlendirmek için kullanılan organik verilerden oluşmalıdır. Veri artırma, modelin farklı varyasyonlara karşı dayanıklılığını artırmak için eğitim verisi üzerinde uygulanır; ancak aynı işlem validasyon verisine uygulanırsa, modelin değerlendirmesi gerçek dünya koşullarını yansıtmayabilir. Bu, yanıltıcı performans sonuçlarına yol açabilir.

Ayrıca, validasyon kümesinde yalnızca **normalizasyon işlemi** uygulanmıştır, çünkü bu işlem verilerin ölçeklendirilerek modelin daha tutarlı bir şekilde değerlendirilmesini sağlar ve eğitim sırasında kullanılan veri işleme adımlarıyla uyumlu kalır. Bu yöntem, modelin genelleme yeteneğini doğru bir şekilde ölçmek için kritik öneme sahiptir.

**A screen shot of a computer program

Description automatically generated**

**Fig.4. Validasyon ve Eğitim Kümesinin Hazırlanması**

Eğitim kümesinde toplam 6042 görüntü ve 2 sınıf olduğunu, validasyon kümesinde ise 754 görüntü ve 2 sınıf olduğu gözlemlenmiştir.

**2.3 Model Mimarisi**

İki farklı temel model kullanılmıştır:

1. DenseNet121:
   * ImageNet ağırlıkları ile transfer learning uygulanmıştır
   * Global Average Pooling katmanı eklenmiştir
   * İki yoğun katman (256 ve 128 nöron) eklenmiştir
   * %50 dropout oranı kullanılmıştır
   * Son katmanda 2 nöronlu softmax aktivasyonu

A computer screen with colorful text

Description automatically generated

**Fig.5. DenseNet Model Tanımlanması**

1. MobileNetV2:
   * ImageNet ağırlıkları ile transfer learning uygulanmıştır
   * Global Average Pooling katmanı eklenmiştir
   * İki yoğun katman (256 ve 128 nöron) eklenmiştir
   * %50 dropout oranı kullanılmıştır
   * Son katmanda 2 nöronlu softmax aktivasyonu

A screen shot of a computer program

Description automatically generated

**Fig.6. MobileNet Model Tanımlanması**

**2.4 Eğitim Stratejisi**

Eğitim iki aşamada gerçekleştirilmiştir:

1. İlk Aşama:
   * Sadece yeni eklenen katmanlar eğitilmiştir
   * Başlangıç öğrenme oranı: 0.001
   * Optimizer: Adam
   * Early Stopping (patience=10)
   * ReduceLROnPlateau kullanılmıştır (factor=0.1, patience=5)

A computer screen with text on it

Description automatically generated

**Fig.7. Model Eğitimi İlk Aşama**

Bu kod, transfer öğrenmenin ilk aşamasında modelin üst katmanlarını eğitmek için tasarlanmıştır. **Adam** optimizasyon algoritması, başlangıç öğrenme oranı **0.001** ile kullanılmıştır. Eğitim sırasında, doğrulama kaybı (**val\_loss**) izlenir. Eğer 10 epoch boyunca iyileşme olmazsa (**EarlyStopping**), eğitim durdurulur ve en iyi ağırlıklar yüklenir. Ayrıca, 5 epoch iyileşme olmadığında öğrenme oranı 10 kat azaltılır (**ReduceLROnPlateau**). Model, **categorical\_crossentropy** kayıp fonksiyonu ve **accuracy** metriğiyle derlenmiştir. Eğitim, maksimum 50 epoch sürecek şekilde, eğitim ve doğrulama veri generator ile gerçekleştirilir. Eğitim sırasında **history1** nesnesi, kayıp ve metrik değişimlerini saklar. Bu yöntem, overfitting'i önlemeyi ve daha iyi genelleme sağlamayı amaçlar.

1. İkinci Aşama (Fine-tuning):
   * Base modelin son 50 katmanı eğitime dahil edilmiştir
   * Öğrenme oranı: 1e-5
   * Diğer parametreler aynı tutulmuştur

A computer screen with text

Description automatically generated

**Fig.8. Model Eğitimi Fine-Tune Adımı**

Bu kod, transfer öğrenmenin ikinci aşamasında, temel modelin ince ayarını (**fine-tuning**) gerçekleştirir. **base\_model.trainable = True** ile temel modelin tüm katmanları eğitilebilir hale getirilir; ancak ilk 50 katman dondurulur, yalnızca üst katmanlar eğitilir.

Model, düşük bir öğrenme oranı (**1e-5**) ile derlenerek ağırlıkların fazla değişmeden yeni verilere uyum sağlaması hedeflenir. Eğitim, **EarlyStopping** ve **ReduceLROnPlateau** geri çağırmalarıyla desteklenir. Sonuçlar **history2** değişkeninde kaydedilir. Bu aşama, önceden öğrenilen bilgileri koruyarak yeni verilerle performansı optimize eder.

1. Model Eğitimi ve Değerlendirme:

* Kod, birden fazla modelin eğitim ve değerlendirmesini yönetir.
* Eğitim süreci, yeni katmanların eğitimi ve ince ayar (fine-tuning) aşamalarını içerir.
* Test verisi, **valid\_datagen.flow\_from\_directory** ile hazırlanır.
* Tahminler **model.predict** ile yapılır, tahmin edilen sınıflar (**y\_pred\_classes**) ve gerçek sınıflar (**y\_true**) karşılaştırılır.
* **ROC ve AUC:** ROC eğrisi ve AUC değeri.
* **Confusion Matrix (CM):** Gerçek ve tahmin edilen sınıflar karşılaştırılır.
* **Classification Report:** Doğruluk, hatırlama, F1 skoru gibi metrikler.
* Sonuçlar model adıyla ilişkilendirilmiş bir sözlükte saklanır.
* Kod, birden fazla modelin eğitimini ve performans karşılaştırmasını sağlar.

A screen shot of a computer program

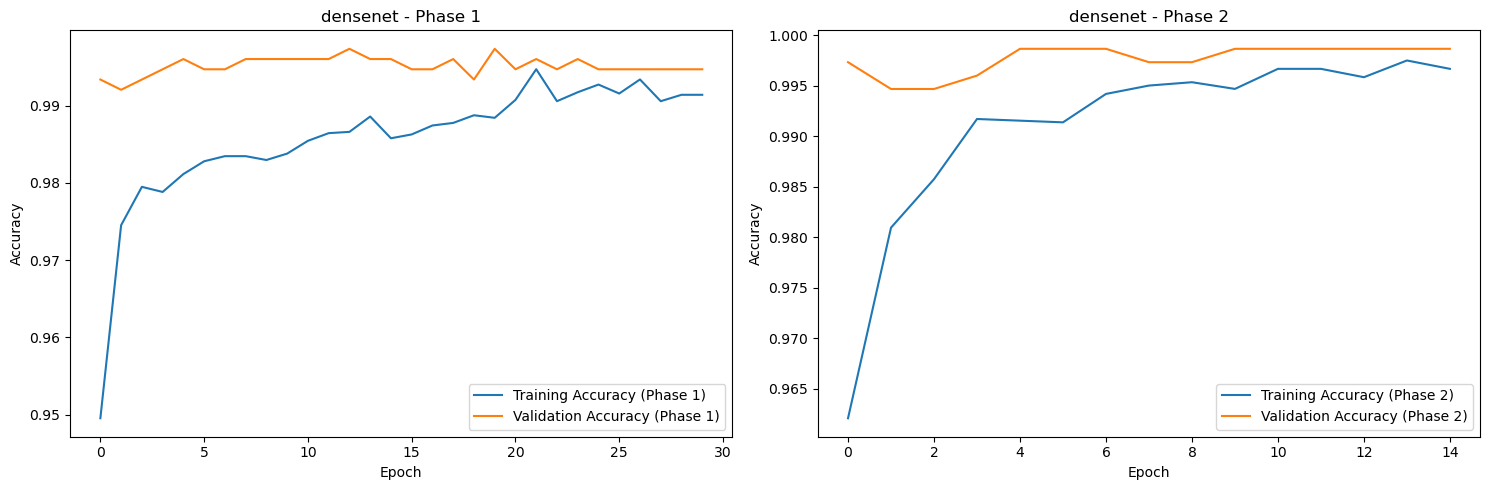
Description automatically generated

**Fig.9. Model Eğitimleri ve Değerlendirme**

**3. DENEYSEL SONUÇLAR**

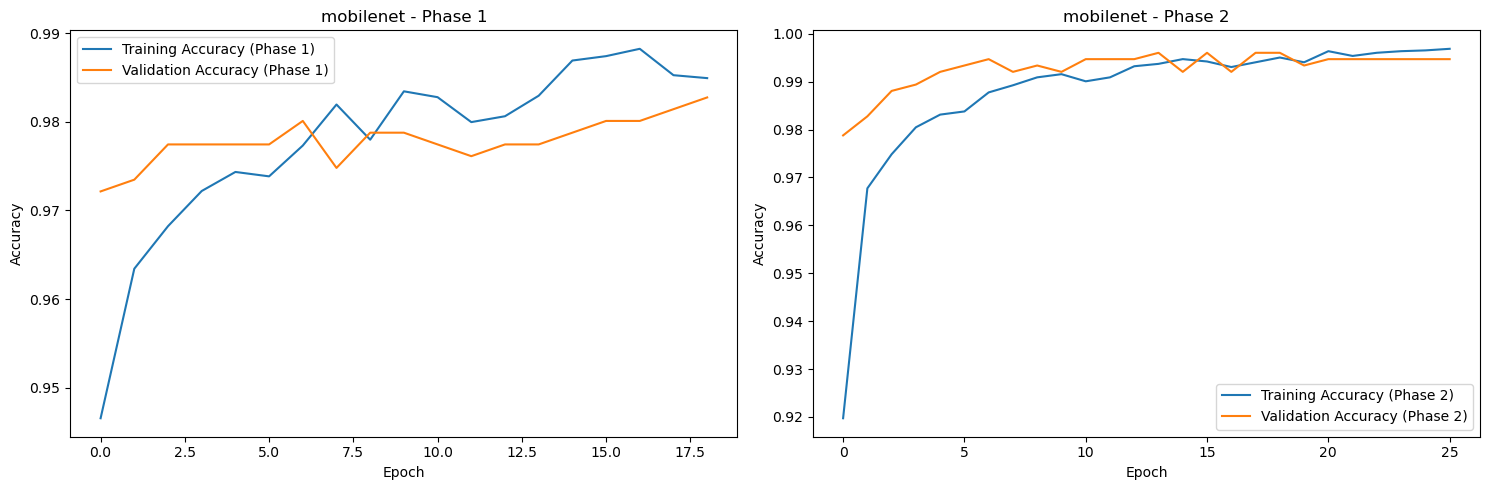
**3.1 Eğitim Grafikleri**

DenseNet Phase 1 çalışmasında pre-trained modelin eğitim ve validasyon grafiği ve DenseNet Phase 2 fine-tune edilmiş eğitim ve validasyon grafiği aşağıdaki gibidir.



**Fig.10. DenseNet Phase1 ve Phase2 Grafikleri**

MobileNetv2 Phase 1 çalışmasında pre-trained modelin eğitim ve validasyon grafiği ve MobileNetv2 Phase 2 fine-tune edilmiş eğitim ve validasyon grafiği aşağıdaki gibidir.

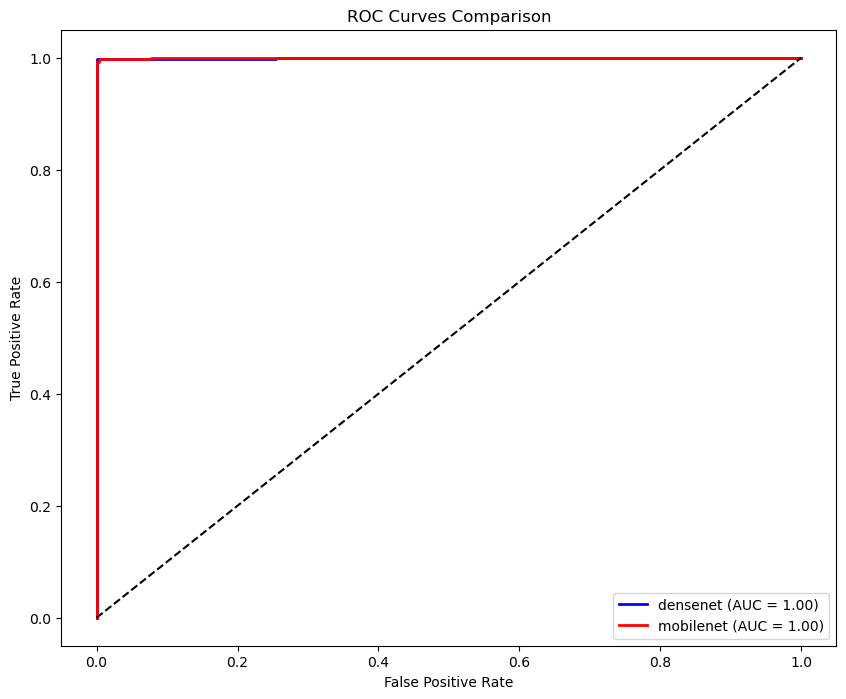


**Fig.11. MobileNet Phase1 ve Phase2 Grafikleri**

Grafikler incelendiğinde, her iki modelin de hızlı bir şekilde yüksek doğruluk değerlerine ulaştığı görülmektedir. Fine-tuning aşamasında performansta ek iyileşme gözlenmiştir.

**3.2 ROC Eğrisi**

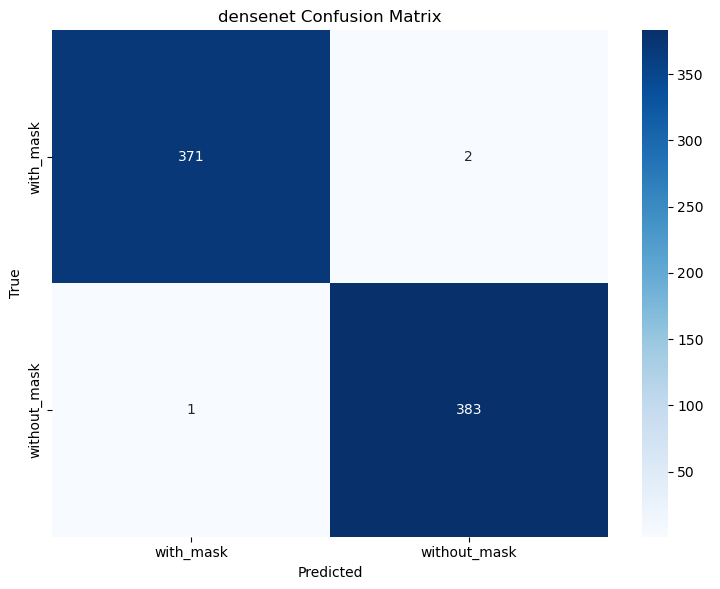
ROC eğrileri, her iki modelin de yüksek ayırt edici performansa sahip olduğunu göstermektedir.



**Fig.12. ROC Eğrisi**

**3.3 Karmaşıklık Matrisi**

Karmaşıklık matrisleri, modellerin (DenseNet ve MobileNet) hem maskeli hem de maskesiz yüzleri yüksek doğrulukla sınıflandırabildiğini göstermektedir.



**Fig.13. DenseNet Karmaşıklık Matrisi**

**A blue squares with white text

Description automatically generated**

**Fig.14. MobileNet Karmaşıklık Matrisi**

**3.4 Performans Metrikleri**

Fine-tune edilmiş DenseNet modeli uygulanmış modelin sınıflandırma raporu incelendiğinde başarıları gayet kabul edilebilir bir nokta olduğu gözlenmiştir. Modelin aşırı öğrenme ihtimali olduğu düşünülmüş, bundan dolayı karmaşıklık matrisi incelenmiş ve aşırı öğrenme durumunun söz konusu olmadığı kanıtlanmıştır.

**A screenshot of a computer screen

Description automatically generated**

**Fig.15. DenseNet Sınıflandırma Raporu**

**A screenshot of a computer screen

Description automatically generated**

**Fig.16. MobileNet Sınıflandırma Raporu**

**4. SONUÇ VE DEĞERLENDİRME**

Bu çalışmada, transfer learning yaklaşımı kullanılarak maske tespiti yapan iki farklı derin öğrenme modeli geliştirilmiştir. Uygulanan iki aşamalı eğitim stratejisi ve veri artırma teknikleri sayesinde yüksek performans elde edilmiştir.

Önemli gözlemler:

1. Her iki model de yüksek doğruluk değerlerine ulaşmıştır
2. Fine-tuning aşaması modellerin performansını artırmıştır
3. Veri artırma teknikleri overfitting'i önlemede etkili olmuştur
4. Early stopping ve learning rate scheduling modellerin eğitimini optimize etmiştir

Gelecek çalışmalarda:

* Farklı model mimarileri denenebilir
* Daha büyük ve çeşitli veri setleri kullanılabilir
* Gerçek zamanlı tespit için model optimizasyonu yapılabilir
* Farklı maske tipleri için çoklu sınıflandırma denenebilir

**KAYNAKLAR**

1. 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 (pp. 4700-4708).
2. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).