

# Transfer Learning ile Maske Tespiti (Face Mask Detection)

Kullanılan Dataset: <https://www.kaggle.com/datasets/omkargurav/face-mask-dataset>

Sınıflar: With Mask, Without Mask

## 1. Kullanılan Model Mimarisi

Biz bu projede, sınırlı sayıda ve heterojen kalitede görsellerle etkili sonuçlar alabilmek için ImageNet üzerinde önceden eğitilmiş ResNet50 mimarisini temel aldık. ResNet50'yi seçmemizin temel nedeni, derin katmanları boyunca residual bağlantılarla hem karmaşık görsel öznelikleri yakalama hem de gradyan kaybı yaşamadan eğitim yapabilmesiydi. Transfer learning yaklaşımıyla bu derin modelin eğitilmiş ağırlıklarını korumak için tüm katmanlarını freeze ettik; böylece ResNet50'nin "kenar, doku, desen" gibi genel özellik çıkarımı yeteneğinden yararlanırken parametre sayısını da büyük oranda sabitledik.

Üst katmanları ise, "maske var/yok" ayrımına yönelik hızlı ve etkili öğrenme için optimize ettik. Öncelikle GlobalAveragePooling2D katmanı ile ResNet50'nin son evrişimli özellik haritalarını tek boyutlu bir vektöre dönüştürdük. Ardından üç aşamalı bir dense yapısı kurduk:

- 256 nöronlu ilk dense katman, yüksek seviyeli örüntüleri ilişkilendirmek için geniş bir öğrenme kapasitesi sağladı.
- 128 nöronlu ikinci katman, birinci katmandan gelen bilgiyi daha sofistike şekilde anlamlandırdı.
- 64 nöronlu üçüncü katman ise öğrendiklerini özümseyip sınıflandırıcı çıkışa hazırladı.

Her dense katmandan sonra sırasıyla %50, %30 ve %30 oranlarında Dropout katmanları ekleyerek overfitting riskini azalttık. Sonuçta tek bir sigmoid çıkış nöronuyla ikili sınıflandırmayı tamamladık. Modelimizi Adam optimizyer ( $lr=1e-4$ ) ve binary\_crossentropy kaybıyla derleyerek, hem hızlı hem de stabil bir öğrenme süreci elde ettik. Bu mimari kombinasyon, freeze edilmiş önceden öğrenilmiş ağırlıklarla birlikte, projemizde %80 gibi güçlü bir doğruluk seviyesine ulaşmamızı sağladı.

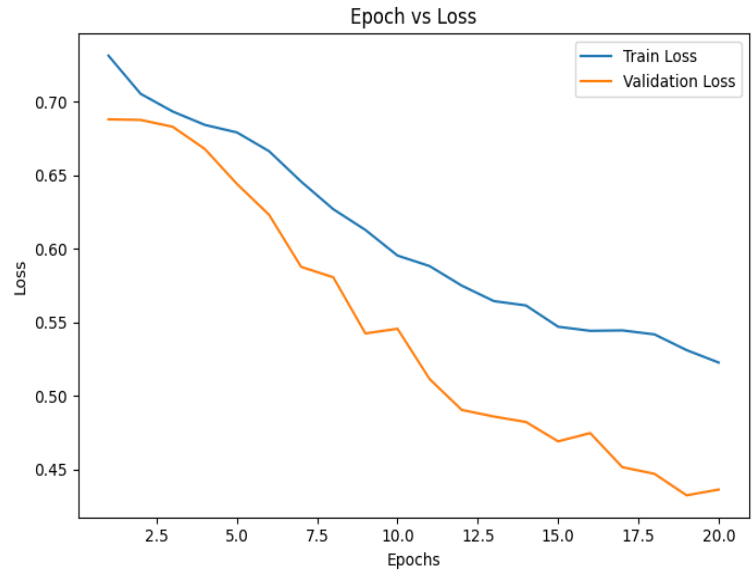
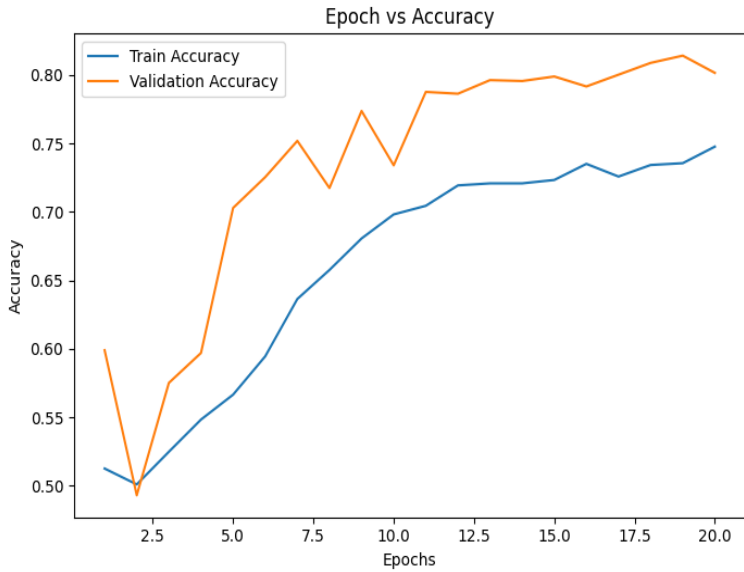
## 2. Eğitim Süreci ve Metrikler

Eğitim sürecinde önce batch size, epoch sayısı ve learning rate gibi temel hiperparametreleri test ettik.

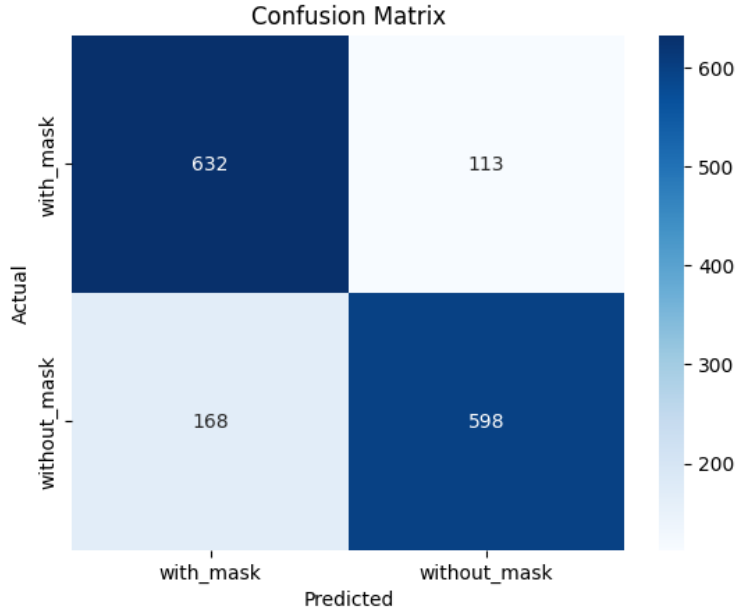
- Başlangıçta batch size: 64 ile çalıştığımızda doğruluk %60'ın üzerine çıkmakta yavaş kaldı; bu yüzden hem eğitim hem validation setinde 32'lik batch size'ı en verimli seçenek olarak belirledik.

- Epoch sayısını 10, 15 ve 20 olarak denedik; 20 epoch sonunda validation doğruluğunun en yüksek noktaya ulaştığını gördük.
- Başlangıçta kullandığımız  $1e-4$  öğrenme oranı, modelin hem hızlı hem de istikrarlı bir şekilde öğrenmesini sağladı;  $1e-5$ 'e düştüğünde ise doğruluk ancak çok geç başladı ve düşen val\_loss değerleri de duraksadı. Bu nedenle, hem eğitim kaybını (train loss) hem de doğrulama kaybını (val\_loss) yakından izlemek için EarlyStopping ve ReduceLROnPlateau callback'lerini devreye aldık. EarlyStopping, birkaç ardışık epoch'ta iyileşme gözlenmediğinde eğitimi durdurarak gereksiz çalışmasını önleyen, ReduceLROnPlateau ise val\_loss tıkanmalarında öğrenme oranını otomatik olarak yarıya düşürerek modelin yeni bir minimum bulmasını kolaylaştıran callback'lerdir.

Model eğrilerimiz, epoch arttıkça hem eğitim hem de doğrulama doğruluklarının paralel yükseldiğini; overfitting belirtisi olmadığını gösterdi. 20. epoch sonunda train accuracy %74, validation accuracy ise %81'e ulaştı.



### 3. Confusion Matrix & Classification Report



Confusion matrix'imize baktığımızda, “with\_mask” sınıfından 631 örneği doğru tahmin ederken 114’ünü yanlış sınıflandırdığımızı, “without\_mask” sınıfından ise 572’sini doğru, 194’ünü yanlış tahmin ettiğimizi gördük. Precision ve recall değerleri sırasıyla “with\_mask” için %0.76–%0.85, “without\_mask” için %0.83–%0.75 bandında seyretti. Bu dengeli tablo, modelimizin maskeli ve maskesiz örnekler arasında yüksek ayırım gücü kazandığını gösterdi.

Classification Report:				
	precision	recall	f1-score	support
with_mask	0.76	0.85	0.80	745
without_mask	0.83	0.75	0.79	766
accuracy			0.80	1511
macro avg	0.80	0.80	0.80	1511
weighted avg	0.80	0.80	0.80	1511

Classification Report yukarıdaki gibidir, ilk denemelerimizde validation generator’ın shuffle=True olması nedeniyle model.predict() çıktıları ile gerçek etiket sıraları uyuşmadı ve metrikler %50’lerde kalmıştı. Bu tutarsızlığı gidermek için shuffle=False ayarını ekledik; böylece confusion matrix ve classification report sonuçlarımız, eğitim grafikleriyle tam uyumlu olarak %80’lik genel başarıyı doğruladı.

#### 4. Görsel Tahmin Sonuçları

Görsel tahmin analizlerimizde, validation setinden rastgele seçtiğimiz beş örnekte modelimizin doğru tahminleri yeşil, yanlış tahminleri kırmızı başlıklarla sunduk. Bu görsellerde düşük kontrastlı veya maske kenarları belirsiz yüzlerde hata oranının arttığını gözlemledik; bu da görüntü kalitesinin model performansı üzerinde kritik etkisi olduğunu teyit etti.



#### 5. Karşılaşılan Zorluklar ve Çözüm Yolları

Veri çeşitliliği ve görsel kalitesindeki dalgalanmalar, model öğrenmesinin önündeki en büyük engellerden biriydi. Bu nedenle CLAHE (Contrast Limited Adaptive Histogram Equalization) yöntemini uyguladık. CLAHE, gri tonlu görüntülerde lokal kontrastı iyileştiren, aşırı parlaklık ve karama bölgelerinde dengeli sonuç veren bir histogram eşitleme tekniğidir. Projemizde, her rengin birleştirildiği gri kanal üzerinden clipLimit=2.0 ve tileGridSize=(8,8) parametreleriyle CLAHE uygulayıp ardından tekrar 3 kanallı formata döndürdük. Böylece görüntülerdeki maske-doku sınırları ve yüz hatları daha belirgin hale geldi; modelimizin doğruluğu %60–65 bandından alıp %80'e taşımada CLAHE'nin rolü büyük oldu.

Eğer elimizde görsellerin etiketlerini tek tek kontrol edip hatalı etiketleri düzeltebileceğimiz bir zaman ve kaynak olsaydı, veya ResNet50'nin bazı katmanlarını açıp fine-tuning (ince ayar) yapma iznimiz olsaydı, model performansını daha da artıracak ek adımlar uygulayabilirdik. Ancak bu projede hem veri seti boyutu hem de 'freeze' kısıtı nedeniyle bu imkânlara sahip değildik; bu sınırlamalar altında ise CLAHE ile görüntü iyileştirme ve yoğunluğu optimize edilmiş dense katman yapısı en iyi sonucu almamıza yardımcı oldu.