

# Face Mask Detection

Ahmet Kaan Tırhış

231101060

# Proje Tanıtımı

## Face Mask Detection

- Proje en basit şekliyle insan yüzlerini alıp bu insanların maskeyi düzgün takıp takmadığını takıyor mu takmıyor mu sorusuna cevap veriyor.
- Olasılıklar Maske Takması ve Maske Takmaması. Yanlış Maske Takması gibi ve bunların hepsi sadece 2 class şeklinde sınıflandırılıyor.
- Metod olarak CNN ve HOG+SVM kullandım.
- Kullandığım database yaklaşık 15k dosyalık ve iki farklı kaynaktan alındı. Üzerine eklediğim demo için google imagesden alınan resimler var. Dataset kaynaklarını ileride belirtiyorum.

# Veriseti Bilgileri

- <https://www.kaggle.com/datasets/omkargurav/face-mask-dataset>
- Yaklaşık 7000 veri dengeli
- <https://github.com/cabani/MaskedFace-Net>
- Yaklaşık 8000 veri dengeli
- 2 Sınıftan oluşuyor: Düzgün Maskeli ve Düzgün Maskesiz(Maskesiz dahil bu sınıfa)
- **Toplam Görüntü: 15065**
- Eğitim: 12053
- Doğrulama: 3012 %20

CNN

# CNN Model: 3 Katmanlı TensorFlow/Keras

- CNN hiperparametre ayarlaması yaptıktan ve overlearning engellemek için learning rate ayarlayıp early stopping ekledikten sonra yaklaşık 1 saat içinde eğitimini benim bilgisayarımda tamamlayabiliyor.
- Hiperparametreler:
  - Batch Boyutu: 32
  - Epoch Sayısı: 10
  - Validation/Test %80
  - Dropout: 0.5 (Dropout validation accuracy'yi etkiledi)
  - Image Size: 255x255
  - Learning rate: 0.0003 ancak durgunlukta LR'yi düşüren kısım bulunmakta.CNN katmanları sonraki slaytta.

# CNN Veriseti İşlemleri

- **Görsel boyutu:**  $256 \times 256$
- **Augmentasyon:**
  - Döndürme:  $\pm 15^\circ$
  - Zoom: %10
  - Yatay çevirme
  - Normalizasyon: 1./255 rescale

# CNN Katmanları

```
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(256, 256, 3)),
    MaxPooling2D(pool_size=(2,2)),

    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(pool_size=(2,2)),

    Conv2D(128, (3,3), activation='relu'),
    MaxPooling2D(pool_size=(2,2)),

    Flatten(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')
])
```

# Eğitimden Hemen Sonraki Performans Sonuçları

```
F1 Skorları:  
F1-Score (Weighted): 0.9927  
F1-Score (Macro): 0.9927  
✓ F1-Score (Sınıf Bazında):  
| correct_mask: 0.9927  
| incorrect_mask: 0.9927  
  
✓ Validation Set - Classification Report:  
| | | | | | | | precision | recall | f1-score | support  
| | | | | | | |-----|-----|-----|-----|  
| | correct_mask | 0.99 | 1.00 | 0.99 | 1504  
✓ | incorrect_mask | 1.00 | 0.99 | 0.99 | 1508  
| | | | | | | |-----|-----|-----|-----|  
| | | | | | | | accuracy | 0.99 | 3012  
| | | | | | | | macro avg | 0.99 | 0.99 | 0.99 | 3012  
| | | | | | | | weighted avg | 0.99 | 0.99 | 0.99 | 3012  
| | | | | | | |-----|-----|-----|-----|  
  
Validation Set - Confusion Matrix:  
✓ [[1503 1]  
| [ 21 1487]]  
=====
```



HOG + Linear SVM

# HOG SVM

- Histogram of Oriented Gradients (HOG) + Destek Vektör Makineleri (SVM)
- **Ön İşleme Aşamaları:**
  - Gri seviyeye çevirme
  - CLAHE (lokal kontrast artırımı)
  - Bilateral filtreleme (kenar koruyucu)
  - Histogram stretching
  - Yeniden boyutlandırma:  $128 \times 128$

# HOG SVM

- Yöntem: Histogram of Oriented Gradients (HOG)
- Parametreler:
  - orientations=12
  - pixels\_per\_cell=(6, 6)
  - cells\_per\_block=(3, 3)
  - block\_norm='L2-Hys'
- Amaç: Kenar ve yönelime dayalı öznitelik çıkarımı
- Boyut: ~8100 öznitelik / görüntü
- Boyut Azaltımı (PCA): %95 varyans korunara

# HOG SVM

- LinearSVC, C=1.0
  - Maksimum iterasyon: 2000
  - C: [0.1, 1.0, 10.0]
  - loss: ['hinge', 'squared\_hinge']
  - tol: [1e-4, 1e-3]
- 
- Pipeline:
  - StandardScaler → LinearSVC

# HOG SVM

```
=====
Toplam örnek: 15065
Eğitim kümesi: 12052 örnek
Test kümesi: 3013 örnek
Model: HOG + LinearSVC
Doğruluk (accuracy): 0.8765
Classification Report:
              precision    recall  f1-score   support

 correct_mask      0.88      0.87      0.87      1498
incorrect_mask      0.87      0.89      0.88      1515

      accuracy              0.88      3013
      macro avg      0.88      0.88      0.88      3013
      weighted avg      0.88      0.88      0.88      3013

Confusion Matrix:
[[1300  198]
 [ 174 1341]]
=====
```