
Python İle Veri Bilimi Bitirme Projesi

Proje Raporu

Proje Adı

YOLOv8 ile Gerçek Zamanlı Sosyal Mesafe ve Maske Tespiti Sistemi

Proje Amacı

Bu projenin amacı, bir kameradan gelen gerçek zamanlı görüntüler üzerinde insanların sosyal mesafeye uyup uymadığını ve maske takıp takmadığını tespit eden bir sistem geliştirmektir. Sistem, YOLOv8 nesne algılama modeli ve OpenCV yardımıyla çalışır.

1. Giriş

Bu proje, COVID-19 pandemisi sürecinde halk sağlığını korumaya yönelik iki temel önlemin otomatik tespiti amacıyla geliştirilmiştir: **maske takma durumu tespiti** ve **sosyal mesafe ihlali algılama**.

Günümüzde, kalabalık ortamlarda bu tür tedbirlerin denetlenmesi insan gücüyle zor ve hataya açık olduğu için, yapay zeka tabanlı görüntü işleme yöntemleri ile bu kontrollerin otomatikleştirilmesi büyük önem taşımaktadır.

Bu çalışmada, YOLOv8 nesne algılama modeli kullanılarak maskeli, maskesiz ve hatalı maske takan kişiler algılanmış; aynı zamanda kişiler arasındaki sosyal mesafe mesafesi hesaplanarak ihlal durumları tespit edilmiştir.

2. Veri Seti ve Ön İşleme

2.1 Kullanılan Veri Seti

Proje kapsamında kullanılan veri seti, Kaggle platformunda bulunan **Mask Detection Dataset** olmuştur. Bu veri seti, farklı koşullarda çekilmiş binlerce yüz görüntüsünü ve üzerlerindeki maske durumlarını içermektedir.

Veri setindeki etiketler üç sınıfa ayrılmıştır:

- with_mask** (Maske takılı doğru şekilde)
- without_mask** (Maske takılı değil)
- mask_worn_incorrect** (Maske hatalı takılmış, örneğin burun açıkta)

2.2 Veri Dönüşümü

Orijinal veri setindeki XML formatındaki anotasyon dosyaları, YOLOv8 modelinin anlayabileceği formata dönüştürülmüştür. Bu amaçla Python ile yazılmış `convert_xml2yolo.py` scripti kullanılmış ve bounding box koordinatları normalize edilerek `.txt` formatına çevrilmiştir.

2.3 Eğitim-Doğrulama Ayrımı

Veri seti, %80 eğitim ve %20 doğrulama olacak şekilde `prepare_data.py` scripti ile rastgele ikiye bölünmüştür. Böylece modelin performansı doğrulama seti üzerinden gerçekçi biçimde değerlendirilebilmiştir.

3. Model Eğitimi

3.1 Model Seçimi

Nesne algılama için Ultralytics tarafından geliştirilmiş ve hız ile doğruluk dengesi açısından tercih edilen **YOLOv8 nano (yolov8n.pt)** modeli kullanılmıştır.

3.2 Eğitim Parametreleri

- Eğitim veri seti: `dataset/images/train`
- Doğrulama veri seti: `dataset/images/val`
- Sınıf sayısı: 3
- Görüntü boyutu: 640x640 piksel
- Eğitim tur sayısı (epochs): 20
- Cihaz: CUDA destekli GPU (device=0)

Model, transfer öğrenme yöntemi ile önceden eğitilmiş ağırlıklar üzerinden ince ayar (fine-tuning) yapılarak maskeli yüz tespiti için optimize edilmiştir.

3.3 Eğitim Komutu

Eğitim terminal komutu şu şekildedir:

```
yolo task=detect mode=train model=yolov8n.pt data=data.yaml epochs=20  
imgsz=640 device=0
```

4. Gerçek Zamanlı Maske ve Sosyal Mesafe Tespiti

4.1 Sistem Mimarisi ve İşleyişi

Modelin çıktılarına dayanarak gerçek zamanlı kamera görüntüsünden maske takma durumu tespiti yapılmakta, kişiler bounding box ile işaretlenmektedir.

Ayrıca kişiler arası sosyal mesafe, kutuların merkez koordinatları kullanılarak Öklid mesafe formülüyle hesaplanmaktadır. Sosyal mesafe ve maske durumu kuralları doğrultusunda, ihlal durumları kırmızı renkle uyarı olarak gösterilmektedir.

4.2 Uyarı ve Görselleştirme

- Maske doğru takılmış kişi: Yeşil kutu
- Hatalı takılmış maske: Sarı kutu
- Maske takmayan kişi: Kırmızı kutu
- Sosyal mesafe ihlali olan kişiler: Kutu rengi kırmızıya dönüşür, ayrıca ekranda “SOCIAL DISTANCE VIOLATION!” uyarısı gösterilir.

4.3 Kodun Ana İşlevi

`main.py` dosyasında OpenCV ve Ultralytics YOLO kütüphanesi kullanılarak, kameradan alınan her karede:

- Model ile tespit yapılır
- Maske durumuna göre kutu ve yazılar çizilir
- Kişiler arası mesafe hesaplanır ve risk durumu belirlenir
- Sonuçlar gerçek zamanlı olarak kullanıcıya gösterilir

5. Sonuç ve Değerlendirme

- Model, gerçek zamanlı kameradan alınan görüntülerde maskeli, maskesiz ve hatalı maske takan kişileri yüksek doğrulukla tespit edebilmiştir.
- Sosyal mesafe ihlalleri, kişinin maske durumu dikkate alınarak tutarlı biçimde belirlenmiştir.
- Uygulama, farklı mesafelerde ve ışık koşullarında gerçekçi sonuçlar vermiştir.
- Performans ve doğruluk arasında dengeli çalışan YOLOv8 nano modeli, düşük gecikmeyle gerçek zamanlı tespit sağlamıştır.
- Son olarak çıkardığım sonuç neticesinde, modelimizin performansı kullanılan veri setinin büyüklüğü ve çeşitliliği ile yakından ilişkilidir. Mevcut veri setindeki sınırlı örnek sayısı nedeniyle bazı tespitlerde çakışmalar ve hata oranları gözlemlenmiştir. Veri setinin daha geniş ve çeşitli olması durumunda, modelin öğrenme kapasitesi artacak ve bu tür çakışmaların önüne geçilerek daha sağlıklı ve güvenilir sonuçlar elde edilecektir.

6. Gelecek Çalışmalar

- Model ve sosyal mesafe tespiti için farklı kamera açıları ile daha hassas hale getirilmesi ve kamera iyileştirilmesi yapılabilir
- Kişi mesafe hesaplamasında derinlik algısı veya stereo kamera kullanımı ile daha kesin sonuçlar elde edilmesi.
- Modelin maske sınıflandırması dışında maske türü veya yüz ifadesi gibi ek sınıflandırmalarla zenginleştirilmesi.
- Daha geniş veri setiyle modelin öğrenme kapasitesini geliştirmek.

7. Kaynaklar

- [YOLOv8 - Ultralytics](#)
- [Ahmet Kaya – Veri Bilimi](#)
- [Kaggle Mask Detection Dataset](#)
- OpenCV Documentation
- Python Documentation