

Gerçek Zamanlı Yüz Duygu Tanıma Sistemi

Hazırlayan: ROAA ALALI - 220404927

1. Giriş

Yapay zekâ teknolojilerinin günlük yaşama entegrasyonu hızla artmaktadır. Görüntü işleme, makine öğrenmesi ve derin öğrenme gibi alanların birleşimi sayesinde, makineler artık insan yüz ifadelerinden duyguları tanıyabilir hâle gelmiştir. Bu proje kapsamında geliştirilen sistem, gerçek zamanlı kamera görüntülerinden insan yüzünü tespit edip, ifadesine göre duygusunu tahmin etmektedir.

Sistem; insan-bilgisayar etkileşimi, güvenlik ve izleme uygulamaları gibi birçok alanda kullanılabilecek potansiyele sahiptir.

2. Kullanılan Teknolojiler ve Araçlar

- Python: Projenin temel programlama dili.
- OpenCV: Gerçek zamanlı yüz algılama ve görüntü işleme işlemleri için kullanıldı
- TensorFlow / Keras: Derin öğrenme modeli (CNN) oluşturmak ve eğitmek için.
- NumPy: Görüntü verileri üzerinde matematiksel işlemleri yapmak için.
- Haar Cascade: Kamera görüntülerinde hızlı ve etkili yüz tespiti sağladı.

3. Model ve Yöntem

→ Veri Girişi Sistem :

48x48 boyutunda gri tonlamalı yüz görselleri ile eğitildi. Bu görüntüler temel olarak FER-2013 veri setinden elde edilmiştir

→ CNN Eğitim:

7 temel duygu sınıfı (mutluluk, üzüntü, öfke, korku, tiksinti, şaşkınlık ve nötr) için bir Konvolüsyonel Sinir Ağı (CNN) eğitildi. Model, yüz özelliklerini otomatik olarak çıkarıp sınıflandırma yapabilecek şekilde tasarlandı.

→ Gerçek Zamanlı Tahmin:

Kameradan gelen görüntüde yüz algılandıktan sonra, bu yüz ROI (Region of Interest) olarak alınır. Ön işleme sonrasında eğitilmiş modele gönderilir ve duygu tahmini yapılır.

4. Stabilizasyon Yöntemi:

Gerçek zamanlı görüntüde yüz ifadeleri sıkça değişebilir. Bu nedenle sistemde stabilizasyon amacıyla son 8 duygu tahmini bir kuyrukta (deque) saklanır. En sık tekrar eden duygu, son tahmin olarak ekranda gösterilir. Bu yöntem ile modelin tahminlerinde oluşabilecek ani değişiklikler filtrelenmiş olur.

5. Kodun Temel İşleyişi (kamera_uygu.py) özetle:

- Kameradan görüntü alınır.
- Haar Cascade ile yüz tespiti yapılır.
- Görsel veriler işlenerek CNN modeline verilir.
- 7 duygudan biri tahmin edilir.
- Tahminler cv2.putText() ile ekranda gösterilir.
- Duygu değişimi algılanırsa sesli uyarı (bip) verilir.

6. Sonuçlar

Sistem; öfke, mutluluk, üzüntü, şaşkınlık gibi temel duyguları başarıyla ayırt edebilmektedir. Özellikle iyi aydınlatılmış ortamlarda yüksek doğrulukla çalışmaktadır.

Gerçek zamanlı uygulamada kullanıcı yüzüne kutu çizilerek ve tahmin edilen duygu ekrana yazdırılarak sezgisel bir arayüz sağlanmıştır.

7. Karşılaşılan Zorluklar

- Işık Koşulları: Aşırı karanlık ya da parlak ortamlar modelin tahmin doğruluğunu düşürdü.
- Kamera Kalitesi: Düşük çözünürlüklü kameralar yüz detaylarının kaybına neden oldu.
- Ani Mimik Değişimleri: Yüz ifadesindeki ani değişiklikler kararsız tahminlere yol açtı.

8. Uygulanan İyileştirmeler

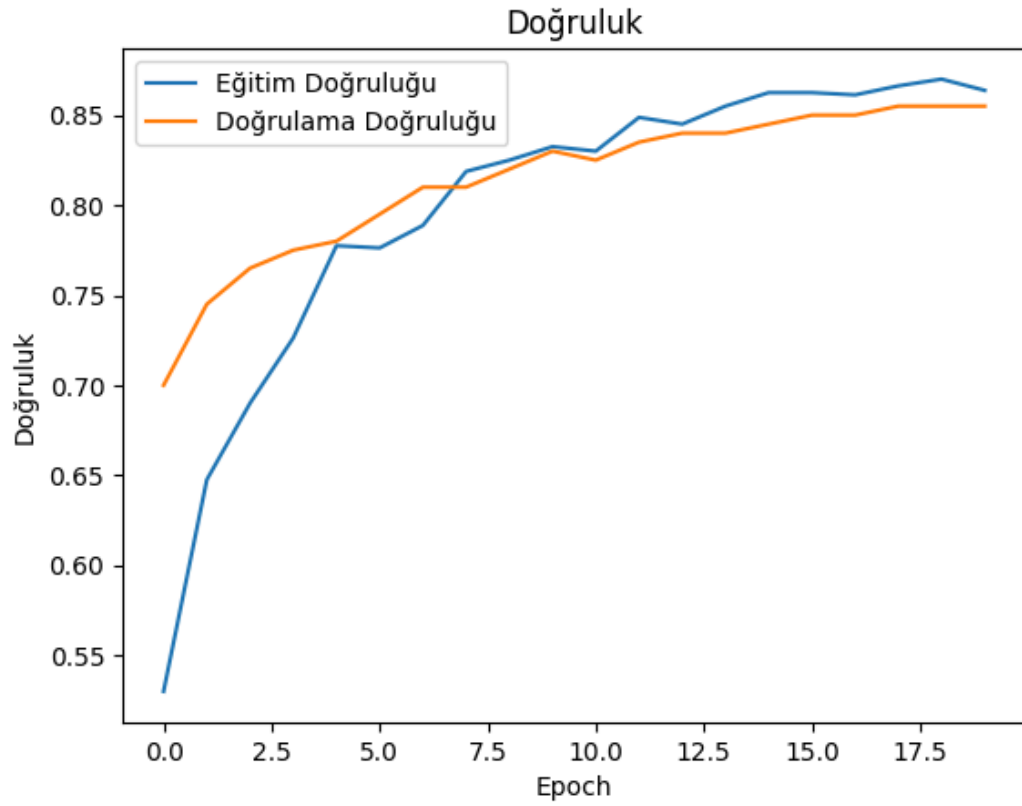
- Histogram Dengeleme: cv2.equalizeHist() yöntemiyle görüntü kalitesi artırıldı.
- Tahmin Stabilizasyonu: deque yapısı ile en sık tekrar eden duygu seçildi.
- Sesli Uyarı: Duygu değişimlerinde sesli bildirim özelliği eklendi.

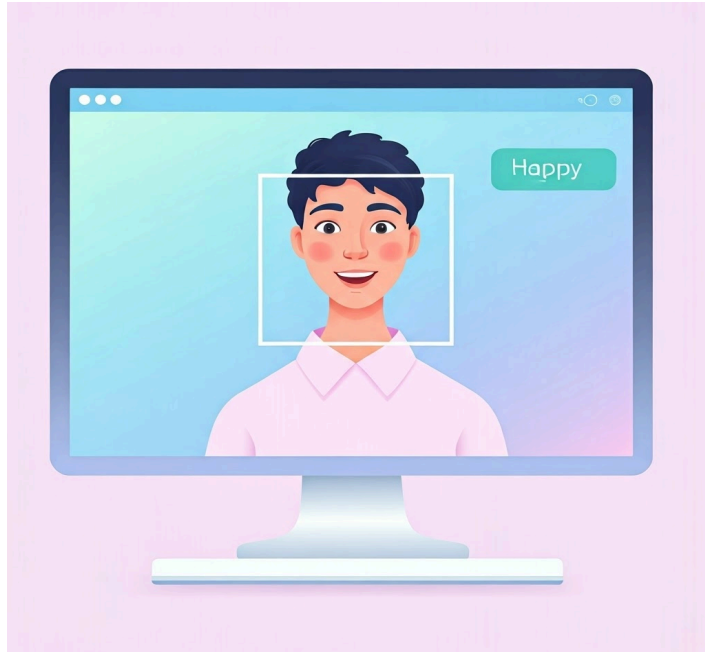
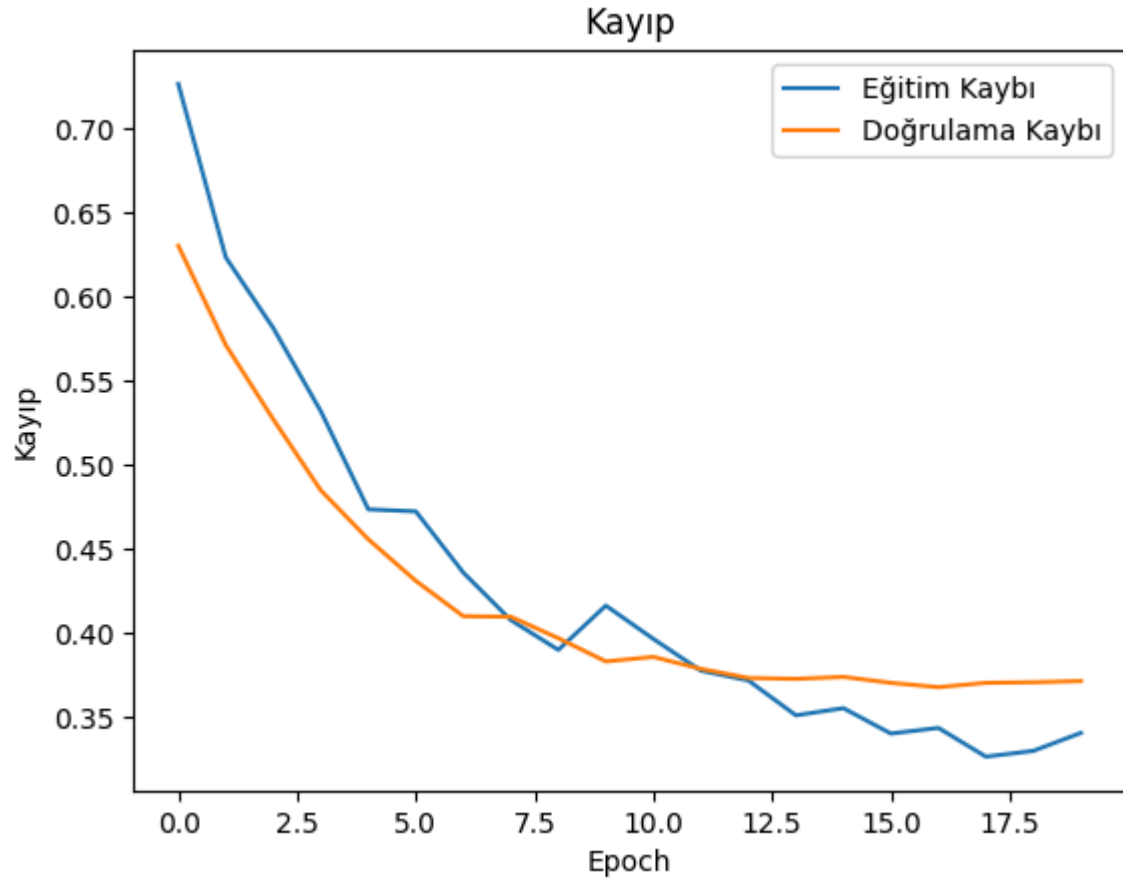
9. Akademik Temeller ve Uygulamalar

Bu proje, derin öğrenme ve bilgisayarla görme alanındaki akademik araştırmalara dayanmaktadır. İnsan-bilgisayar etkileşimi, güvenlik sistemleri ve kullanıcı odaklı arayüzler gibi birçok gerçek dünya uygulamasında kullanılabilir. Empati kurabilen dijital sistemler tasarlamak adına önemli bir adımdır.

10. Toplam Çalışma Süresi

Bu proje, bireysel olarak yürütülmüş ve yaklaşık 20 gün süren planlama, modelleme, kodlama ve test aşamalarından oluşmuştur.





```
1 import cv2 # Görüntü işleme için OpenCV
2 import numpy as np # Numpy dizileri ve matris işlemleri
3 from tensorflow.keras.models import load_model # Eğitilmiş modeli yüklemek için
4 from collections import deque, Counter # Stabilizasyon için kuyruk ve sayım
5 import os, platform # İşletim sistemi ve bip sesi işlemleri
6
7 # == MODELİ YÜKLE ==
8 try:
9     model = load_model("duygu_modeli.keras") # Eğitilmiş duygu tanıma modelini yükler
10 except:
11     print("❌ Model yüklenemedi! Dosya mevcut mu?")
12     exit()
13
14 # == DUYGU ETİKETLERİ ==
15 emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral'] # Modelin sınıf çıktıları
16
17 # == YÜZ ALGILAMA ==
18 face_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_frontalface_default.xml') # Haar cascade ile yüz tespiti
19
20 # == DUYGU STABİLİZASYONU ==
21 emotion_history = deque(maxlen=8) # Son 8 tahmini saklayan kuyruk (ani değişimi önlemek için)
22 last_stabilized_emotion = None # En son sabitlenen duygu
23
24 # == SESLİ BİLDİRİM (DUYGU DEĞİŞİNCE BİP) ==
25 def play_beep():
26     if platform.system() == "Windows": # Windows sistemlerde
27         import winsound
28         winsound.Beep(1000, 200) # 1000 Hz frekans, 200 ms
29     else:
30         os.system('printf "\a"') # Mac/Linux sistemlerde terminal bip sesi
31
32 # == KAMERA BAŞLAT ==
33 cap = cv2.VideoCapture(0) # Bilgisayarın varsayılan kamerasını aç
34 if not cap.isOpened(): # Kamera açılmazsa uyarı ver
35     print("❌ Kamera açılmadı!")
36     exit()
37
38 print("👋 Sistem başlatıldı. Gerçek zamanlı duygu tanıma çalışıyor...")
39 elerini kapat
40
```

kodun devamı :

```
2 # === ANA DÖNGÜ ===
3 while True:
4     ret, frame = cap.read() # Kameradan bir kare al
5     if not ret:
6         break
7
8     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Gri formata dönüştür (model böyle bekliyor)
9     faces = face_cascade.detectMultiScale(gray, 1.3, 5) # Yüzleri tespit et
10
11     for (x, y, w, h) in faces:
12         roi = gray[y:y+h, x:x+w] # Yüz bölgesini al (region of interest)
13         roi = cv2.resize(roi, (48, 48)) # Modelin giriş boyutuna küçült
14         roi = cv2.equalizeHist(roi) # Işık farklarını azaltmak için kontrast artır
15         roi = roi.astype("float32") / 255.0 # Normalize (0-1 arası)
16         roi = np.expand_dims(roi, axis=-1) # (48, 48) → (48, 48, 1)
17         roi = np.expand_dims(roi, axis=0) # (48, 48, 1) → (1, 48, 48, 1)
18
19         prediction = model.predict(roi, verbose=0) # Duygu tahmini yap
20
21         if prediction.shape[1] == len(emotion_labels): # Model çıktısı bekleniyorsa
22             emotion = emotion_labels[np.argmax(prediction)] # En yüksek skorlu duygu etiketi
23         else:
24             print("⚠ Model çıktısı beklenenden farklı!") # Hatalı çıktı varsa uyar
25             emotion = "Bilinmiyor"
26
27         emotion_history.append(emotion) # Tahmini kuyrukta sakla
28         stabilized_emotion = Counter(emotion_history).most_common(1)[0][0] # En sık tahmini al
29
30         if stabilized_emotion != last_stabilized_emotion: # Yeni bir duygu tespit edildiyse
31             print(f"⚠ Yeni Duygu: {stabilized_emotion}")
32             play_beep()
33             last_stabilized_emotion = stabilized_emotion
34
35         cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2) # Yüzün etrafına kutu çiz
36         cv2.putText(frame, stabilized_emotion, (x, y-10), # Üstüne sabitlenmiş duyguyu yaz
37                     cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
38
39     cv2.imshow("Duygu Tanıma", frame) # Kameradaki görüntüyü ekranda göster
40
41     if cv2.waitKey(1) & 0xFF == ord('q'): # q tuşuna basıldığında çık
42         break
43
44 cap.release() # Kamerayı serbest bırak
45 cv2.destroyAllWindows() # OpenCV pencer
```

kod sayesinde:

- Bilgisayar kamerası ile **canlı video akışı** alınır.
- Görüntüdeki **yüz otomatik olarak tespit** edilir.
- Yüz görüntüsü, önceden eğitilmiş bir yapay zeka modeline verilir.
- Model, bu görüntüyü analiz ederek **hangi duygunun gösterildiğini tahmin eder**:

● Avantajlar

◆ Gerçek Zamanlı Etkileşim:

Kullanıcı tepkilerine anlık yanıt vererek daha etkileşimli sistemler oluşturulabilir. (eğitim, terapi, oyun)

◆ Otomatik Gözlem:

İnsan müdahalesi olmadan yüz ifadeleri analiz edilir. (örneğin güvenlik veya sürücü takip sistemleri)

◆ Veriye Dayalı Karar:

Kullanıcı memnuniyeti, stres seviyesi gibi duygusal veriler toplanarak daha iyi kararlar alınabilir.

◆ Çok Yönlü Kullanım Alanı:

Eğitim teknolojileri, psikoloji uygulamaları, müşteri hizmetleri, sürücü izleme ve güvenlik sistemlerinde kullanılabilir.

● Dezavantajlar

⚠ Gizlilik ve Etik Sorunlar:

Yüz görüntüsü ve duygular kişisel veri sayılır. Kullanıcı izni olmadan kullanmak etik ve yasal sorunlara yol açabilir.

⚠ Model Hataları / Yanıltıcı Tahminler:

Kültürel farklılıklar veya bireysel ifadeler sonucu tahminler doğru olmayabilir.
(Not: Mutlu görünmek her zaman mutlu olmak değildir.)

⚠ Donanım Bağımlılığı:

Kamera kalitesi, aydınlatma ve işlemci gücü sistemin doğruluğunu doğrudan etkiler.

⚠ Gerçek Zamanlı Yük:

Sistem her kareyi analiz ettiği için işlemciye yük bindirir. Zayıf cihazlarda gecikme olabilir.