

Gerçek Zamanlı Yüz Duygu Tanıma Sistemi

Hazırlayan: ROAA ALALI - 220404927

Proje Amacı



Gerçek Zamanlı Tanıma

Kamera görüntüleri
kullanılarak duyguların anlık
olarak analiz edilmesi.



Yüz İfadelerinin Tespiti

İnsanın temel duygularını
yansıtan yüz ifadelerinin
yüksek doğrulukla
sınıflandırılması.



Kullanım Alanları

İnsan-bilgisayar etkileşimi ve
güvenlik sistemlerinde geniş
uygulama olanakları.

Kullanılan Teknolojiler

- **Python programlama dili:** Yüksek seviyeli, çok yönlü ve yaygın olarak kullanılan bir programlama dili olarak sistemin temel yazılım altyapısını oluşturur.
- **OpenCV ile görüntü işleme:** Gerçek zamanlı video ve görüntü işleme için kullanılan açık kaynak kütüphane, yüz algılama ve ön işleme görevlerinde kullanılır.
- **TensorFlow/Keras kullanarak derin öğrenme:** Yüz ifadelerini tanımak için derin öğrenme modellerinin geliştirilmesi ve eğitilmesinde kullanılan popüler frameworklerdir.
- **NumPy ile veri işleme:** Çok boyutlu diziler ve matris işlemleri için optimize edilmiş kütüphane, model eğitiminde ve görüntü verilerinin hesaplanmasında destek sağlar.
- **Haar Cascade yüz tespiti algoritması:** Yüzlerin hızlı ve etkili bir şekilde tespit edilmesini sağlayan klasik bir makine öğrenmesi algoritmasıdır.

Model ve Yöntem

Giriş Verisi

48x48 boyutunda gri tonlamalı yüz görüntüleri kullanılır. Bu görüntüler, yüz ifadelerinin detaylarıyla birlikte modelin eğitimi için temel veri setini oluşturur.
+ FER-2013 veri setinden elde edilmiştir.

CNN Eğitimi

7 temel duygu sınıfını (mutluluk, üzüntü, şaşkınlık, korku, tiksinti, öfke, nötr) ayırt edebilen derin öğrenme modeli eğitilir. Konvolüsyonel Sinir Ağı (CNN) yapısı, yüz özelliklerini otomatik olarak çıkarır ve sınıflandırma yapar.

Tahmin

Gerçek zamanlı olarak kamera görüntüsü üzerinden yüzler tespit edilir ve önceden eğitilmiş CNN modeli yardımıyla duygular tahmin edilir. Bu tahminler anlık olarak sınıflandırmaya ve görsel çıktıya dönüştürülür.

Sistem Akışı

1

Kamera Görüntüsü alınır

Anlık olarak sistem kamerasından yüksek çözünürlüklü yüz görüntüleri yakalanır ve sonraki işlem adımları için alınan veriler anlık olarak iletilir.

2

Yüz Tespiti

Haar Cascade algoritması kullanılarak görüntü içerisindeki yüzler hızlı ve doğru bir şekilde belirlenir, gereksiz alanlardan filtrelenip yüz regionları tespit edilir.

3

ROI Normalize

Tespit edilen yüz bölgeleri ön işleme tabi tutularak boyut ve ışıklandırma açısından standart hale getirilir; böylece model tutarlı ve doğru sonuçlar verebilir.

4

Duygu Tahmini

Önceden eğitilmiş Konvolüsyonel Sinir Ağı (CNN) modeli, normalize edilmiş yüz görüntülerinden 7 temel duygu sınıfını ayırt ederek tahminler yapar.

5

Stabilizasyon

Modelin yaptığı son duygu tahminleri ortalanarak, sistemdeki dalgalanmalar minimize edilir ve tahmin sonuçlarında yüksek kararlılık sağlanır.

Stabilizasyon Yöntemi ne yapar ?



Duygu Geçmişi

Model, son 8 duygu tahminini takip ederek bu geçmiş veriler üzerinden analiz yapar. Böylece geçici ani değişiklikler etkisiz hale getirilir ve daha tutarlı sonuçlar elde edilir.



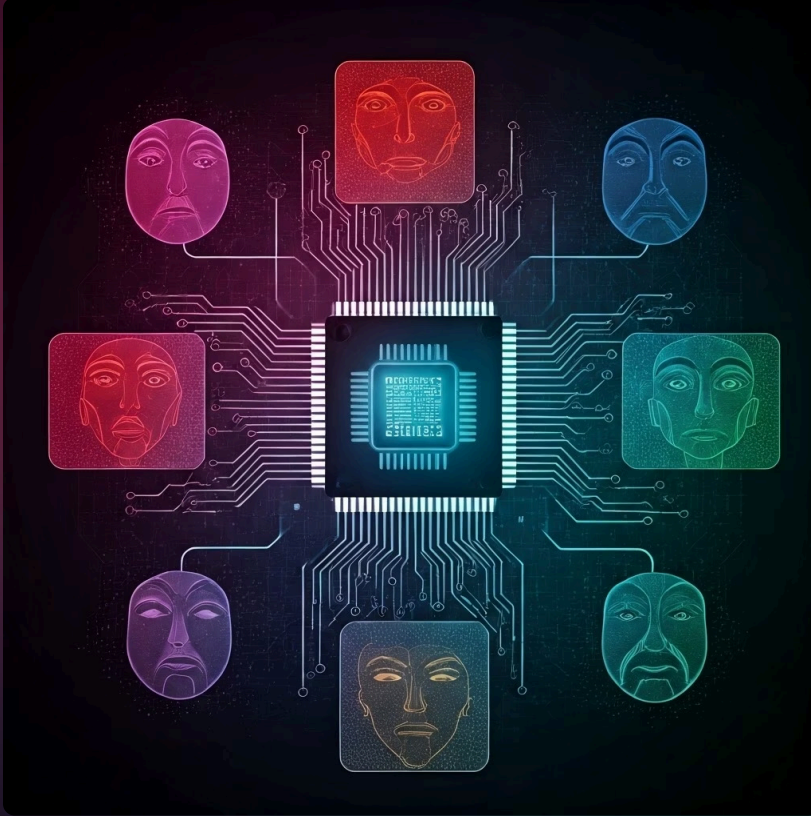
Mode Hesaplama

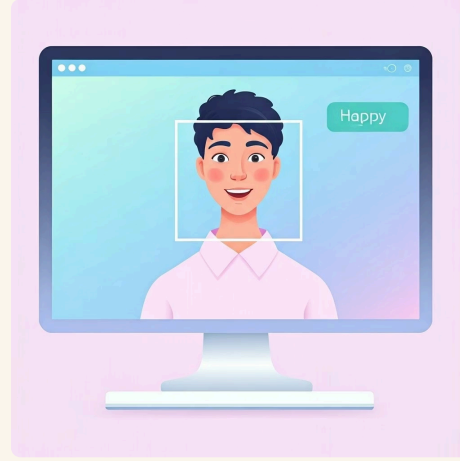
Sistemdeki en sık tekrar eden duygu hesaplanarak, bu duygu stabil ve sabit bir tahmin olarak kullanıcıya gösterilir. Bu yöntem rastgele ya da anlık yanlış tahminlerin önüne geçer.



Kararlılık Artışı

Ani ve sık duygu değişimlerinin etkisi azaltılarak, gerçek zamanlı duygusal analiz sonuçları daha dengeli ve güvenilir hale getirilir. Böylece sistem çıktılarında dalgalanmalar minimize edilir.





🧩 **Kodun Görevi (kamera_duygu.py)** : Bu Python kodu, gerçek zamanlı kamera görüntüsünden insan yüzünü algılar ve ifadesine göre duygusunu tahmin eder. Temel adımlar şunlardır:

- 📷 **Kameradan görüntü alınır.**
- 🧠 **Haar Cascade yöntemiyle yüz tespiti yapılır.**
- 🖋️ **Yüz görüntüsü işlenir ve önceden eğitilmiş modele gönderilir.**
- 📊 **Model, 7 temel duygudan birini tahmin eder: *Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral***
- ♻️ **Tahminler stabilize edilerek ani değişiklikler filtrelenir.**
- 🔔 **Duygu değiştiğinde sesli uyarı (bip) verilir.**
- 🖼️ **Yüzün üzerine kutu çizilir ve tahmin edilen duygu ekranda gösterilir.**

(kamera_duygu.py)

```
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
40 # == ANA DÖNGÜ ==
41 while True:
42     ret, frame = cap.read() # Kameradan bir kare al
43     if not ret:
44         break
45
46     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Gri formata dönüştür (model böyle bekliyor)
47     faces = face_cascade.detectMultiScale(gray, 1.3, 5) # Yüzleri tespit et
48
49     for (x, y, w, h) in faces:
50         roi = gray[y:y+h, x:x+w] # Yüz bölgesini al (region of interest)
51         roi = cv2.resize(roi, (48, 48)) # Modelin giriş boyutuna küçült
52         roi = cv2.equalizeHist(roi) # Işık farklarını azaltmak için kontrast artır
53         roi = roi.astype("float32") / 255.0 # Normalize (0-1 arası)
54         roi = np.expand_dims(roi, axis=-1) # (48, 48) → (48, 48, 1)
55         roi = np.expand_dims(roi, axis=0) # (48, 48, 1) → (1, 48, 48, 1)
56
57         prediction = model.predict(roi, verbose=0) # Duygu tahmini yap
58
59         if prediction.shape[1] == len(emotion_labels): # Model çıktısı bekleniyorsa
60             emotion = emotion_labels[np.argmax(prediction)] # En yüksek skorlu duygu etiketi
61         else:
62             print("⚠ Model çıktısı beklenenden farklı!") # Hatalı çıktı varsa uyar
63             emotion = "Bilinmiyor"
64
65         emotion_history.append(emotion) # Tahmini kuyrukta sakla
66         stabilized_emotion = Counter(emotion_history).most_common(1)[0][0] # En sık tahmini al
67
68         if stabilized_emotion != last_stabilized_emotion: # Yeni bir duygu tespit edildiyse
69             print(f"⚠ Yeni Duygu: {stabilized_emotion}")
70             play_beep()
71             last_stabilized_emotion = stabilized_emotion
72
73         cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2) # Yüzün etrafına kutu çiz
74         cv2.putText(frame, stabilized_emotion, (x, y-10), # Üstüne sabitlenmiş duyguyu yaz
75                     cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
76
77     cv2.imshow("Duygu Tanıma", frame) # Kameradaki görüntüyü ekranda göster
78
79     if cv2.waitKey(1) & 0xFF == ord('q'): # q tuşuna basıldığında çık
80         break
81
82 cap.release() # Kamerayı serbest bırak
83 cv2.destroyAllWindows() # OpenCV pencerelerini kapat
84
```


Kod'tan bazı Örnekler:

Duygu Geçmişi

emotion_history = deque(maxlen=8) : ifadesi, sistemin son 8 duygu tahminini tutmak için kullanılan bir veri yapısını tanımlar. Bu sayede anlık değişimlerden kaynaklanan dalgalanmalar azaltılır ve daha stabil bir duygu analizi yapılır.

En Yaygın Duygu

most_common = Counter(emotion_history).most_common(1)[0][0] : kodu, kaydedilen duygu geçmişindeki en sık rastlanan duyguyu belirler. Bu, sistemin geçici hataları görmezden gelerek kullanıcının genel ruh halini daha doğru şekilde yansıtmasını sağlar.

Ekran Yazısı

cv2.putText() : fonksiyonu, tespit edilen duygu bilgisini kamera görüntüsü üzerinde kullanıcıya görsel olarak sunar. Yüzün çevresine kutu çizilir ve tahmin edilen duygu metin olarak ekranda gösterilir..

Akademik Temeller ve Gerçek Hayat Uygulamaları

Bu yüz duygu tanıma sistemi, hem akademik çalışmalardan beslenen sağlam bir altyapıya sahiptir hem de gerçek hayatta geniş uygulama alanlarına sahiptir. Sistem; insan davranışını anlama, güvenliği artırma ve kullanıcı odaklı teknolojiler geliştirme açısından kritik rol oynamaktadır.



Bilimsel Altyapı

bu Projede , yüz ifadelerinden duygu çıkarımı üzerine yapılmış derin öğrenme ve görüntü işleme temelli güncel akademik araştırmalarla desteklenmektedir.



İnsan-Bilgisayar Etkileşimi

Uyarlanabilir ve duygusal farkındalığa sahip sistemler geliştirerek daha doğal, empati kurabilen kullanıcı arayüzleri oluşturmayı mümkün kılar.



Güvenlik ve İzleme Sistemleri

Kameradan alınan görsel verilerin analiz edilmesiyle potansiyel tehlikeli durumlar (örn. öfke, korku) gerçek zamanlı olarak tespit edilebilir. Bu sayede güvenlik sistemlerine yapay zeka destekli yenilikçi çözümler sunar.

Karşılaşılan Zorluklar ⚠️

💡 Işık Koşulları

Gerçek zamanlı yüz tanıma sistemlerinde ışıklandırma, model doğruluğunu doğrudan etkiler.

Aşırı karanlık veya çok parlak ortamlarda yüz hatları netliğini kaybedebilir. Bu durum, CNN modelinin doğru duygu sınıflandırması yapmasını zorlaştırır.

📹 Kamera Kalitesi ve Açıları

Farklı çözünürlüklerdeki ve düşük kaliteli kameralar görüntüde gürültü oluşturur.

Yüzün kameraya olan açısı değiştikçe, modelin eğitildiği pozlardan sapmalar olur ve bu da tespit oranını azaltır.

😬 Yüz Hareketleri ve Anlık İfadeler

İnsan yüzü saniyeler içinde mimik değişikliklerine uğrar.

Bu ani değişimler, modelin her karede farklı duygu tahmini yapmasına neden olabilir ve sonuçların kararsız ve sürekli değişken olmasına yol açar.



Çözüm ve İyileştirme Yöntemleri



Kontrast İyileştirme (Görüntü Kalitesi Artırımı)

Düşük ışıklı veya bozuk görüntüler duygu tanıma doğruluğunu olumsuz etkiler. Bu nedenle, gri tonlamalı yüz görüntülerine `cv2.equalizeHist()` uygulanarak yüz detayları netleştirildi ve modelin daha doğru tahmin yapması sağlandı.



Tahmin Stabilizasyonu (Tutarlılık Artırımı)

Modelin her karede farklı duygu tahmini yapması engellendi. Son 8 tahmin, `deque` yapısı ile bellekte tutuldu ve en sık tekrar eden duygu (mod) ekranda sabit gösterildi. Bu yöntem, anlık ifade değişimlerine bağlı kararsız tahminleri azalttı.



Kullanıcı Bilgilendirmesi (Sesli Uyarı Sistemi)

Duygu değişimi algılandığında sistem, kullanıcıyı uyarmak için **bip sesi** verir. Bu özellik, sistemin aktif olduğunu gösterirken ani duygu değişimlerine de dikkat çeker.

✨ *Dinlediğiniz için Teşekkürler* ✨