## **1. Giriş 🎯**

Bu projenin temel amacı, bilgisayarla görü ve makine öğrenmesi yöntemlerini kullanarak dört farklı temel yüz ifadesini (mutluluk, üzüntü, kızgınlık, şaşkınlık) gerçek zamanlı olarak tanıyabilen uçtan uca bir sistem geliştirmektir. Geliştirilen sistem, insan-bilgisayar etkileşimini zenginleştirme, duygusal analiz uygulamalarına temel oluşturma ve erişilebilirlik teknolojilerine katkı sağlama potansiyeline sahiptir. Proje, veri toplama, model eğitimi ve gerçek zamanlı test aşamalarını kapsamaktadır.

## **2. Yöntem 🛠️**

Projenin gerçekleştirilmesinde izlenen adımlar aşağıda özetlenmiştir:

### **2.1. Veri Seti Oluşturma**

* **Araçlar**: Veri toplama işlemi için **Python** programlama dili, **OpenCV** kütüphanesi ile kamera erişimi ve **MediaPipe** kütüphanesinin FaceLandmarker aracı kullanılmıştır.
* **Süreç**: "Mutlu", "üzgün", "kızgın" ve "şaşkın" olmak üzere dört temel yüz ifadesi için veri toplanmıştır. Her bir ifade için kullanıcıdan ilgili yüz ifadesini yapması istenmiş ve bu sırada yüzündeki 478 adet 3B ilgin noktasının (landmark) 2B (x, y) koordinatları MediaPipe aracılığıyla çıkarılmıştır.
* **Veri Formatı**: Her bir ifade için en az 200 örnek toplanması hedeflenmiştir. Toplanan ilgin noktası koordinatları, ait oldukları ifade etiketiyle birlikte veriseti.csv adlı bir dosyada satır satır saklanmıştır. İlk satır, ilgin noktası koordinatlarını (x1,y1,x2,y2,...,x478,y478) ve "Etiket" sütununu içeren başlık satırıdır.
* **Uygulama**: Bu aşama, kullanıcıya hangi ifadeyi toplaması gerektiğini ve toplanan örnek sayısını bildiren arayüze sahip yuz\_algila.py betiği ile gerçekleştirilmiştir.

### **2.2. Model Eğitimi**

* **Veri Hazırlığı**: veriseti.csv dosyasındaki veriler, **Pandas** kütüphanesi kullanılarak yüklenmiştir. İlgin noktası koordinatları özellik (X) olarak, ifade etiketleri ise hedef değişken (y) olarak ayrılmıştır.
* **Ölçeklendirme ve Model Seçimi**: Özellikler, scikit-learn kütüphanesindeki StandardScaler ile ölçeklendirilerek modelin daha iyi performans göstermesi amaçlanmıştır. Sınıflandırma için çok sınıflı Lojistik Regresyon modeli tercih edilmiştir. Bu adımlar bir Pipeline içerisinde birleştirilmiştir.
* **Eğitim ve Değerlendirme**: Veri seti, %80 eğitim ve %20 test verisi olacak şekilde ayrılmıştır. Model, eğitim verisi üzerinde eğitilmiş ve performansı test verisi üzerindeki **doğruluk (accuracy)** oranı ile değerlendirilmiştir.
* **Model Kaydı**: Eğitilen Pipeline (StandardScaler ve Lojistik Regresyon modelini içeren) pickle modülü kullanılarak model.pkl dosyasına kaydedilmiştir.
* **Uygulama**: Bu aşama, egitim.py betiği ile yürütülmüştür.

## **3. Sistem Mimarisi ve Uygulama Akışı ⚙️**

Geliştirilen sistem üç ana Python betiğinden oluşmaktadır:

1. **yuz\_algila.py**:
   1. Kameradan görüntü alır.
   2. MediaPipe ile yüz ilgin noktalarını tespit eder.
   3. Kullanıcının belirlediği ifadeye göre (mutlu, üzgün, kızgın, şaşkın) ilgin noktalarını etiketleyerek veriseti.csv dosyasına kaydeder.
   4. Kullanıcıya hangi ifade için veri toplandığı ve kaç örnek alındığı hakkında görsel geri bildirim sağlar.
2. **egitim.py**:
   1. veriseti.csv dosyasını okur.
   2. Veriyi eğitim ve test setlerine böler.
   3. Belirlenen sınıflandırıcıyı (Lojistik Regresyon) eğitir.
   4. Modelin doğruluk oranını ekrana yazdırır.
   5. Eğitilmiş modeli model.pkl olarak diske kaydeder.
3. **yuz\_algila\_test.py**:
   1. Kaydedilmiş model.pkl dosyasını yükler.
   2. Kameradan gerçek zamanlı görüntü alır.
   3. Her karede yüz ilgin noktalarını tespit eder.
   4. Bu ilgin noktalarını kullanarak eğitilmiş model ile yüz ifadesini tahmin eder.
   5. Tahmin edilen ifadeyi (metin olarak) canlı video akışının üzerinde gösterir.

## **4. Elde Edilen Sonuçlar ve Değerlendirme (Örnek) 📈**

*(Bu bölüm, projenin fiili çalıştırılması sonucu elde edilecek spesifik doğruluk değerleri ve gözlemlerle doldurulmalıdır. Aşağısı genel bir ifadedir.)*

Proje kapsamında geliştirilen sistem, dört temel yüz ifadesini gerçek zamanlı olarak tanıma yeteneğini başarıyla sergilemiştir. egitim.py betiği çalıştırıldığında, test seti üzerinde elde edilen doğruluk oranı [Örn: %85] olarak ölçülmüştür. Bu oran, seçilen özelliklerin ve modelin yüz ifadelerini ayırt etmede belirli bir başarıya ulaştığını göstermektedir. yuz\_algila\_test.py ile yapılan gerçek zamanlı testlerde, sistemin farklı ifadeler için tutarlı tahminler ürettiği ve kullanıcıya anlık geri bildirim sağladığı gözlemlenmiştir. Tanıma hızı, gerçek zamanlı uygulamalar için kabul edilebilir düzeydedir.

## **5. Sonuç ve Gelecek Çalışmalar 🏁**

Bu proje ile MediaPipe ve temel makine öğrenmesi teknikleri kullanılarak, dört farklı yüz ifadesini gerçek zamanlı olarak tanıyabilen fonksiyonel bir sistem geliştirilmiştir. Veri toplama, model eğitimi ve gerçek zamanlı çıkarım adımları başarıyla tamamlanmıştır.

Gelecekte yapılabilecek potansiyel geliştirmeler şunlardır:

* **Daha Fazla İfade**: Tanınan ifade sayısının artırılması (örn: korku, iğrenme, nötr).
* **Model Çeşitliliği**: Destek Vektör Makineleri (SVM), Rastgele Orman (Random Forest) veya Derin Öğrenme (CNN, RNN) gibi daha karmaşık modellerin denenerek performans karşılaştırması yapılması.
* **Veri Zenginleştirme**: Daha çeşitli ve büyük bir veri seti kullanarak modelin genelleme yeteneğinin artırılması.
* **Gelişmiş Özellikler**: İlgin noktalarına ek olarak baş pozisyonu, göz bebeği hareketleri gibi ek özelliklerin dahil edilmesi.
* **Kullanıcı Arayüzü**: Daha kullanıcı dostu bir grafik arayüz (GUI) geliştirilmesi.

## **6. Yeniden Üretilebilirlik ♻️**

Projenin yeniden üretilebilirliğini sağlamak amacıyla, tüm Python kodları (yuz\_algila.py, egitim.py, yuz\_algila\_test.py), örnek veriseti.csv dosyası, eğitilmiş model.pkl dosyası ve bu rapor, bir GitHub deposunda paylaşılacaktır. Bu sayede, projenin farklı ortamlarda çalıştırılması ve sonuçların doğrulanması mümkün olacaktır.