Kapsamlı performans analizi ve testler:

1. **Unit test kullanılarak projenin testlerinin yazılması:**

import unittest

import numpy as np

from face\_detect import FaceDetect # FaceDetect sınıfının bulunduğu dosyadan import edin

from settings import GENDERS, EMOTIONS # Test için kullanılan sabitler

class TestFaceDetect(unittest.TestCase):

def setUp(self):  
 """Testler öncesi gerekli nesneler oluşturulur."""  
 self.face\_detect = FaceDetect()  
 self.empty\_frame = np.zeros((480, 640, 3), dtype=np.uint8)

# Boş bir test görüntüsü  
  
def test\_no\_faces(self):  
 """Hiç yüz tespit edilmediğinde doğru çıktı verildiğinden emin olun."""  
 result = self.face\_detect.run(self.empty\_frame)  
 self.assertEqual(result['label'], {}, "Boş görüntüde yüz algılanmamalı.")  
  
def test\_flip\_parameter(self):  
 """Flip parametresinin çalışıp çalışmadığını kontrol et."""  
 flipped\_result = self.face\_detect.run(self.empty\_frame, flip=True)  
 normal\_result = self.face\_detect.run(self.empty\_frame, flip=False)  
 np.testing.assert\_array\_equal(flipped\_result['frame'], np.flip(self.empty\_frame, axis=1), "Flip çalışmıyor.")  
  
def test\_gender\_detection(self):  
 """Cinsiyet tespitinin başarılı şekilde yapıldığını kontrol et."""  
 result = self.face\_detect.run(self.empty\_frame, options={'genders': [0, 1]})  
 if result['label']:  
 self.assertIn(result['label']['gender']['title'], GENDERS, "Cinsiyet yanlış tespit ediliyor.")  
  
def test\_emotion\_detection(self):  
 """Duygu durum tespitinin doğru çalıştığını kontrol et."""  
 result = self.face\_detect.run(self.empty\_frame, options={'emotions': list(range(len(EMOTIONS)))})  
 if result['label']:  
 self.assertIn(result['label']['emotion']['title'], EMOTIONS, "Duygu durumu yanlış tespit ediliyor.")

if **name** == "**main**":

unittest.main()

1. **Parametrelerin testi:**

import unittest import numpy as np

class TestFaceDetect(unittest.TestCase):

def setUp(self):

# Test için sahte modeller ve XML verileri kullanalım.

self.face\_detect = FaceDetect()

def test\_gender\_detection(self):  
 # Sahte bir yüz verisi oluşturup farklı cinsiyet modellerini test edelim  
 fake\_roi = np.random.random((1, 64, 64, 1))  
   
 # Örnek test: Kadın ve Erkek tespiti için  
 genders\_result = self.face\_detect.gender\_model.predict(fake\_roi)  
 self.assertEqual(len(genders\_result[0]), len(settings.GENDERS))  
 self.assertTrue(np.all(genders\_result[0] >= 0.0) and np.all(genders\_result[0] <= 1.0))  
   
def test\_emotion\_detection(self):  
 # Sahte bir yüz verisi oluşturup duygu tespitini test edelim  
 fake\_roi = np.random.random((1, 64, 64, 1))  
   
 # Örnek test: Duygu skorları  
 emotions\_result = self.face\_detect.emotion\_model.predict(fake\_roi)  
 self.assertEqual(len(emotions\_result[0]), len(settings.EMOTIONS))  
 self.assertTrue(np.all(emotions\_result[0] >= 0.0) and np.all(emotions\_result[0] <= 1.0))  
  
def test\_multiple\_faces(self):  
 # Bir çerçevede birden fazla yüz tespit edildiğinde işlem yapılıp yapılmadığını kontrol edelim  
 fake\_frame = np.random.random((480, 640, 3)).astype(np.uint8)  
 results = self.face\_detect.run(fake\_frame)  
 self.assertIsInstance(results['label'], dict)

if **name** == "**main**":

unittest.main()

**Parametreler:**

* flip (Görüntüyü ters çevirme),
* options (Cinsiyet ve duygu seçimi),
* \_eyes ve \_smiles (Göz ve gülümseme algılama) gibi değerler var.

Bu parametreleri farklı kombinasyonlarla test etmek, fonksiyonun beklenen şekilde çalışıp çalışmadığını kontrol eder.

**Test Senaryoları:**

1. **flip=True ile Test**: Görüntü aynalandığında sistemin doğru çalışıp çalışmadığını kontrol edersiniz.

result = face\_detect.run(frame, flip=True)

1. **options Parametresiyle Test**: Belirli cinsiyet ve duygu ayarlarında çalışıp çalışmadığını kontrol edersiniz.

result = face\_detect.run(frame, options={'genders': [0], 'emotions': [3]})

1. **\_eyes ve \_smiles Parametreleriyle Test**: Göz veya gülümseme algılamanın etkin olduğu senaryoları test edersiniz.

result = face\_detect.run(frame, \_eyes=True, \_smiles=False)

1. **Hiçbir Ek Parametre Verilmeden Test**: Varsayılan ayarlarla çalıştığından emin olursunuz.

result = face\_detect.run(frame)

**3) Analiz sonuçlarının özetlenmesi:**

def summarize\_analysis\_results(results):

""" Analiz sonuçlarını özetleyen bir metot :param results: run() metodundan dönen sonuçlar """

labels = results.get('label', {})

gender\_info = labels.get('gender', {}).get('title', 'Belirlenemedi') gender\_confidence = labels.get('gender', {}).get('percent', 0) emotion\_info = labels.get('emotion', {}).get('title', 'Belirlenemedi')

emotion\_confidence = labels.get('emotion', {}).get('percent', 0)

summary = (  
 f"Cinsiyet: {gender\_info} (%{gender\_confidence}), "  
 f"Duygu: {emotion\_info} (%{emotion\_confidence})."  
)  
return summary

Şöyle açıklanabilir:

**Algılanan Özelliklerin Belirlenmesi:** Sistem, çerçevede bulunan yüzlerin analizini tamamladıktan sonra cinsiyet ve duygular gibi özellikleri tanımlar. Her bir yüz için belirlenen cinsiyet (örneğin "Erkek" veya "Kadın") ve duygu (örneğin "Mutlu" veya "Üzgün") gibi bilgiler bir listeye eklenir.

**Özelliklerin Skorları ile Birlikte Gösterilmesi:** Tespit edilen her özelliğin güven skorları da belirtilir. Örneğin, sistem bir yüzün "Erkek" olduğunu %85 doğrulukla belirlerse, bu bilgi analiz sonuçlarına eklenir. Bu skorlar, sistemin ne kadar emin olduğunu göstermesi açısından kritik bir öneme sahiptir.

**Verilerin Gruplandırılması ve Birleştirilmesi:** Eğer çerçevede birden fazla yüz algılanmışsa, her yüz için ayrı ayrı analiz yapılır ve sonuçlar bir araya getirilir. Örneğin, "Birinci yüz: Erkek (%85), Mutlu (%90). İkinci yüz: Kadın (%70), Üzgün (%80)." gibi bir özet hazırlanır.

**Anlaşılır ve Kullanıcı Dostu Bir Format:** Elde edilen tüm bilgiler, teknik detaylara girmeden, kullanıcı için anlamlı ve kolay anlaşılır bir paragraf halinde sunulur. Örneğin, "Çerçevede iki yüz algılandı. İlk yüz için analiz sonuçları: Erkek (%85) ve Mutlu (%90). İkinci yüz için analiz sonuçları: Kadın (%70) ve Üzgün (%80)." gibi bir açıklama yapılır.

**İstatistiksel Özet:** Eğer analiz bir video veya birden fazla çerçeve üzerinden yapıldıysa, toplam analiz edilen çerçeve sayısı, her bir cinsiyetin ve duygunun genel dağılımı gibi istatistikler de eklenebilir. Bu, kullanıcının büyük veriler arasında genel bir perspektif kazanmasını sağlar.