Kapsamlı performans analizi ve testler:

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

### **Genel Yapı**

* **Test Çerçevesi:** unittest modülü kullanılarak bir test sınıfı (TestFaceDetect) oluşturulmuş.
* **Test Edilen Sınıf:** FaceDetect sınıfı, face\_detect modülünden içe aktarılmış ve testlerde kullanılıyor.
* **Kurulum:** setUp metodu, her testten önce çalıştırılan bir hazırlık metodudur.

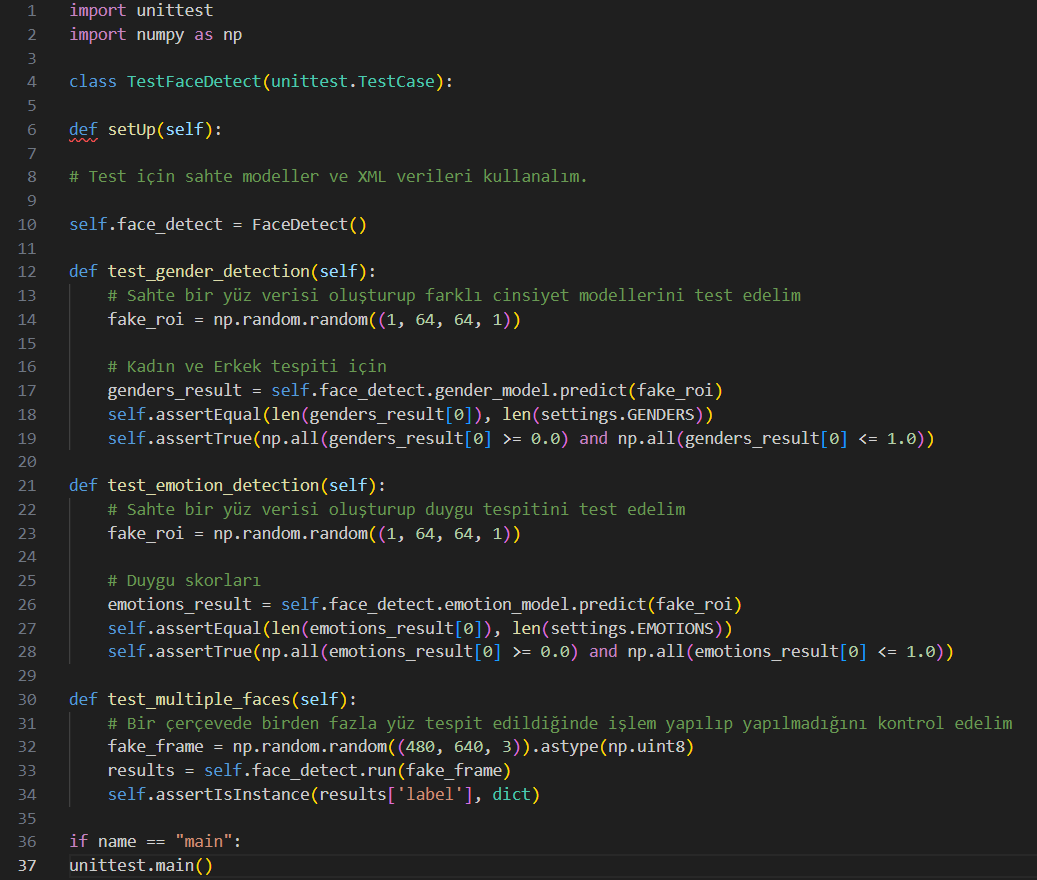
### **Testler ve Amaçları**

1. **setUp Metodu:**
   1. Testlerin çalışması için gerekli olan bir FaceDetect nesnesi (self.face\_detect) ve boş bir görüntü çerçevesi (self.empty\_frame) oluşturur.
   2. Boş çerçeve, (480, 640, 3) boyutlarında bir sıfır matrisidir (uint8 türünde), genelde bir boş video karesi simüle etmek için kullanılır.
2. **test\_no\_faces:**
   1. Amaç: Boş bir görüntü karesi verildiğinde, yüz algılama sisteminin doğru şekilde "yüz algılanmadı" sonucunu döndürdüğünü kontrol etmek.
   2. Kullanım: self.face\_detect.run(self.empty\_frame) metodu çalıştırılır ve sonucu kontrol edilir.
   3. Beklenen Çıktı: Etiket (label) olarak "Boş görüntüde yüz algılanmamalı." mesajıyla eşleşir.
3. **test\_flip\_parameter:**
   1. Amaç: Algılama sisteminde flip parametresinin doğru çalışıp çalışmadığını kontrol etmek.
   2. Kullanım: Görüntü ters çevrilerek (flip=True) ve normal şekilde işlenir (flip=False) ve sonuçlar kıyaslanır.
   3. Beklenen Çıktı: Ters çevrilmiş kare, NumPy'nin np.flip fonksiyonuyla alınan sonuçla eşleşmelidir.
4. **test\_gender\_detection:**
   1. Amaç: Yüz algılama sisteminin cinsiyet algılamasının doğru şekilde çalıştığını kontrol etmek.
   2. Kullanım: Bir çerçeve üzerinde cinsiyet algılama yapılarak çıkan sonuç, tanımlı GENDERS listesiyle kontrol edilir.
   3. Beklenen Çıktı: Algılanan cinsiyet (gender) değeri, önceden tanımlanmış cinsiyetler (GENDERS) arasında olmalıdır.
5. **test\_emotion\_detection:**
   1. Amaç: Duygu algılamasının doğru çalıştığını kontrol etmek.
   2. Kullanım: emotions parametresi ile verilen listeye göre algılanan duygu (emotion) değerinin önceden tanımlanmış duygular (EMOTIONS) arasında olup olmadığı kontrol edilir.
   3. Beklenen Çıktı: Sonuç, tanımlı duygu listesiyle eşleşmelidir.
6. **if \_\_name\_\_ == "\_\_main\_\_":**
   1. Bu bölüm, testlerin doğrudan dosya çalıştırıldığında (python dosya.py) çalışmasını sağlar.
   2. unittest.main() çağrısıyla, test çerçevesi tüm testleri otomatik olarak çalıştırır.

### **Kullanılan Modüller**

* **numpy:** NumPy, burada boş çerçeve oluşturma ve ters çevirme işlemlerinde kullanılmış.
* **unittest:** Test çerçevesi için kullanılır, testlerin organize edilmesini ve çalıştırılmasını sağlar.
* **GENDERS ve EMOTIONS:** Muhtemelen cinsiyet ve duyguların önceden tanımlandığı listeler (örneğin: ["male", "female"] ve ["happy", "sad"] gibi).

1. **Parametrelerin testi:**



### **Genel Yapı**

* **Kütüphaneler:**
  + unittest: Testleri organize ve çalıştırmak için kullanılıyor.
  + numpy: Rastgele veri oluşturmak ve matris işlemleri için kullanılıyor.
* **Test Sınıfı:** TestFaceDetect sınıfı, unittest.TestCase sınıfını miras alarak çeşitli test fonksiyonları içeriyor.
* **Kurulum:** setUp metodu, her bir testten önce çağrılır ve testler için gerekli nesneleri hazırlar.

### **Test Fonksiyonları**

#### **1. setUp Metodu:**

* Testlerden önce bir FaceDetect nesnesi oluşturuluyor ve bu nesne test fonksiyonlarında kullanılıyor.
* **Not:** FaceDetect sınıfı, burada test edilen yüz algılama modelini temsil ediyor.

#### **2. test\_gender\_detection:**

* **Amaç:** Cinsiyet algılama modelinin doğruluğunu test eder.
* **Açıklama:**
  + Sahte bir yüz bölgesi (fake\_roi) oluşturuluyor. Bu, (1, 64, 64, 1) boyutlarında rastgele değerlerden oluşan bir matristir ve bir yüz görüntüsünü simüle eder.
  + gender\_model.predict(fake\_roi) ile cinsiyet algılama modeli çalıştırılır ve sonucu genders\_result değişkenine kaydedilir.
  + **Testler:**
    - len(genders\_result[0]), cinsiyet kategorilerinin sayısına eşit olmalıdır (len(settings.GENDERS)).
    - Tüm cinsiyet skorları [0.0, 1.0] aralığında olmalıdır. Bu, np.all(genders\_result[0] >= 0.0) ve np.all(genders\_result[0] <= 1.0) ile doğrulanır.

#### **3. test\_emotion\_detection:**

* **Amaç:** Duygu algılama modelinin doğruluğunu test eder.
* **Açıklama:**
  + Sahte bir yüz bölgesi (fake\_roi) oluşturuluyor. Bu da (1, 64, 64, 1) boyutunda rastgele değerlerden oluşur.
  + emotion\_model.predict(fake\_roi) ile duygu algılama modeli çalıştırılır ve sonuç emotions\_result değişkenine kaydedilir.
  + **Testler:**
    - len(emotions\_result[0]), duygu kategorilerinin sayısına eşit olmalıdır (len(settings.EMOTIONS)).
    - Tüm duygu skorları [0.0, 1.0] aralığında olmalıdır. Bu da np.all(emotions\_result[0] >= 0.0) ve np.all(emotions\_result[0] <= 1.0) ile kontrol edilir.

#### **4. test\_multiple\_faces:**

* **Amaç:** Aynı çerçevede birden fazla yüz algılama senaryosunu test eder.
* **Açıklama:**
  + Rastgele bir görüntü çerçevesi (fake\_frame) oluşturuluyor. Bu, (480, 640, 3) boyutlarında ve uint8 veri türünde bir matristir.
  + face\_detect.run(fake\_frame) çalıştırılarak çerçevede yüz algılama yapılır.
  + **Test:** Algılama sonucu (results['label']) bir Python sözlüğü (dict) olmalıdır. Bu, self.assertIsInstance(results['label'], dict) ile doğrulanır.

### **Testlerin Çalıştırılması**

* **if \_\_name\_\_ == "\_\_main\_\_":**: Test dosyası doğrudan çalıştırıldığında (python test\_file.py), tüm testler unittest.main() çağrısıyla çalıştırılır.

### **Özet**

Bu test senaryoları, bir yüz algılama modelinin farklı özelliklerini sahte verilerle test ediyor:

1. **Cinsiyet Algılama:** Modelin çıktısının doğru uzunlukta ve geçerli aralıkta olduğunu doğrular.
2. **Duygu Algılama:** Duygular için modelin skorlarının doğru aralıkta olduğunu kontrol eder.
3. **Birden Fazla Yüz Algılama:** Çerçevede birden fazla yüzün algılanmasını ve sonuç tipinin doğruluğunu test eder.

**Diğer 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]})

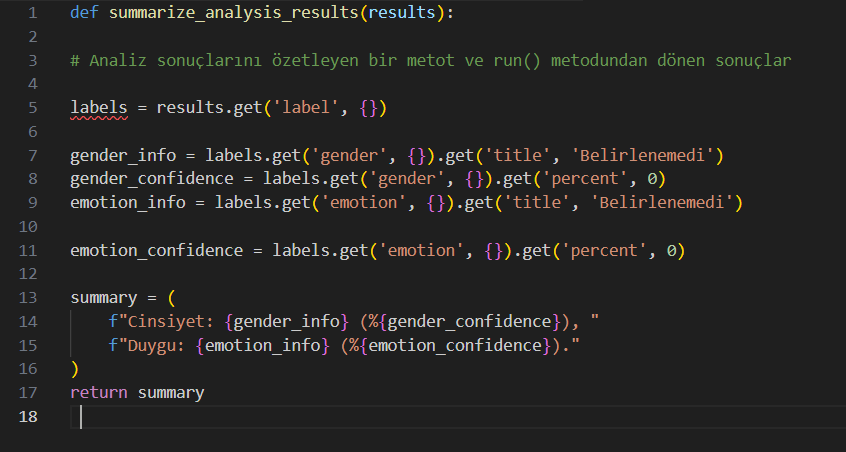
1. **\_eyes ve \_smiles Parametreleriyle Test**: Göz veya gülümseme algılamanın etkin olduğu senaryoları test edersiniz.

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:**



**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.