***faceDetect.py*  
  
options**

* **Tanım**: Kullanıcının filtreleme yapabileceği parametreleri tutar.
* **Özellikler**:
  + genders: Analizde dikkate alınacak cinsiyet kategorileri.
  + emotions: Analizde dikkate alınacak duygu kategorileri.

**flip**

* **Tanım**: Görüntüyü yatay eksende ters çevirmek için kullanılan bir parametre.
* **Varsayılan**: False.
* **Kullanım**: Aynalı görüntü oluşturmak için True yapılabilir.

**face\_cascade, eye\_cascade, smile\_cascade**

* **Tanım**: Haar Cascade sınıflandırıcılarıdır.
* **İşlev**:
  + face\_cascade: Yüz tespiti.
  + eye\_cascade: Göz tespiti.
  + smile\_cascade: Gülümseme tespiti.

**emotion\_model ve gender\_model**

* **Tanım**: TensorFlow ile yüklenen eğitimli modellerdir.
* **Kullanım**:
  + emotion\_model: Duygu analizi yapmak için kullanılır.
  + gender\_model: Cinsiyet analizi yapmak için kullanılır.

**frame**

* **Tanım**: Üzerinde işlem yapılacak görüntü karesi (OpenCV tarafından alınan bir görüntü).
* **İşlev**:
  + Yüz, cinsiyet ve duygu analizlerinin yapıldığı ana veri.

**label**

* **Tanım**: Analiz sonuçlarını tutan bir sözlük.

**ROI (Region of Interest)**

* **Tanım**: Analiz için kullanılan yüzün belirli bir alanını temsil eder.
* **İşlev**:
  + roi\_gray: Grayscale görüntü.
  + roi\_color: Renkli görüntü

***faceDetectVideo.py***

**1. capture**

* **Tanım**: OpenCV ile kamera görüntüsünü almak için kullanılan cv2.VideoCapture nesnesidir.
* **İşlev**:
  + capture = cv2.VideoCapture(0): Kamera açılır (0, varsayılan olarak bağlı olan ilk kamerayı ifade eder).
  + Bu nesne, kameradan alınan görüntüleri her bir döngüde alır.

**face\_detect (FaceDetect sınıfı)**

* **Tanım**: Yüz, cinsiyet ve duygu analizi yapabilen sınıfın bir nesnesi.
* **İşlev**:
  + face\_detect = FaceDetect(): FaceDetect sınıfından bir nesne oluşturuluyor.
  + Bu nesne, her bir kare üzerinde yüz, cinsiyet ve duygu analizi yapmak için kullanılır.

**cv2.imshow('frame', detect)**

* **Tanım**: OpenCV kullanarak işlemden geçirilen görüntüyü ekranda gösterir.
* **İşlev**:
  + Bu satır, işlenmiş görüntüyü "frame" adlı pencere içerisinde ekranda gösterir.
  + detect: FaceDetect.run() fonksiyonundan dönen işlenmiş görüntüdür.
* **Kullanım**:
  + Ekranda sürekli olarak yüz tespiti, cinsiyet ve duygu analizi eklenmiş görüntüyü görmek için kullanılır.

**cv2.VideoCapture(0)**

* **Tanım**: Web kameradan video almayı sağlayan OpenCV fonksiyonudur.
* **İşlev**:
  + cv2.VideoCapture(0) fonksiyonu, bilgisayarınıza bağlı ilk kamerayı açar.
  + 0 parametresi, genellikle varsayılan web kamerasını ifade eder. Eğer birden fazla kamera varsa, 1, 2 gibi sayılarla diğer kameralar seçilebilir.

**ret, frame**

* **Tanım**: cv2.VideoCapture.read() fonksiyonundan dönen iki parametredir.
* **İşlev**:
  + ret: Kamera görüntüsü başarıyla alınmışsa True, aksi takdirde False döner.
  + frame: Kameradan alınan görüntü karesi.

**while True: (sonsuz döngü)**

* **Tanım**: Kameradan sürekli olarak görüntü alınmasını sağlar.
* **İşlev**:
  + Bu döngü, kamera görüntüsünü sürekli olarak okur ve her bir görüntü üzerinde analiz yapar.
  + True olduğunda sonsuza kadar çalışmaya devam eder.
* **Kullanım**:
  + Sonsuz döngü sayesinde, kullanıcı kameradan çıkana kadar sürekli olarak yüz tespiti ve diğer analizler yapılır.

**capture.release()**

* **Tanım**: Kamerayı serbest bırakır.
* **İşlev**:
  + Kamerayı kapatır ve kaynakları serbest bırakır.
  + Bu, kameranın düzgün bir şekilde kapanması ve başka programlar tarafından kullanılabilmesi için önemlidir.

**cv2.destroyAllWindows()**

* **Tanım**: Tüm OpenCV pencerelerini kapatır.
* **İşlev**:
  + cv2.imshow() fonksiyonu ile açılan pencereyi sonlandırmak için kullanılır.
  + Bu, uygulama kapatıldığında ekranda açılan pencerelerin temizlenmesini sağlar.

***pictureFileDialog.py***

**selected\_file**

* **Tanım**: Kullanıcının seçtiği resim dosyasının yolunu saklar.
* **İşlev**:
  + Bu parametre, kullanıcı resim dosyasını seçtiğinde dosyanın tam yolunu kaydeder. Bu yol, daha sonra resim işleme işlemi için kullanılır.

**pictureThread**

* **Tanım**: Fotoğraf işleme işlemini başlatan ve yöneten iş parçacığı (PictureThread).
* **İşlev**:
  + PictureThread sınıfı, resim işleme (yüz tespiti, cinsiyet ve duygu analizi) işlemleri için bir iş parçacığıdır.
  + pictureThread.changePixmap.connect(self.setImage) ile işlenmiş resmin GUI'ye aktarılması sağlanır.
  + Bu iş parçacığı başlatıldığında, seçilen resim üzerinde işlem yapılır ve işlem sonuçları GUI'ye yansıtılır.

**output\_name**

* **Tanım**: Kullanıcının seçtiği resim dosyasının adı.
* **İşlev**:
  + Kullanıcı bir dosya seçtiğinde, output\_name dosyanın adını alır.
  + Bu parametre, kullanıcıya seçilen dosyanın adını göstermek için kullanılır ve aynı zamanda kaydetme işlemi için gereklidir.

**option\_genders**

* **Tanım**: Kullanıcının seçtiği cinsiyet analiz seçeneklerini saklar.
* **İşlev**:
  + Kullanıcı, resimde cinsiyet analizi yapılmasını isterse, option\_genders listesi seçilen cinsiyetleri (Kadın ve/veya Erkek) saklar.
  + Bu seçenek, resim işleme işlemi sırasında geçerli olacak cinsiyet analizini belirler.

**option\_emotions**

* **Tanım**: Kullanıcının seçtiği duygu analiz seçeneklerini saklar.
* **İşlev**:
  + Kullanıcı, resimde duygu analizi yapılmasını isterse, option\_emotions listesi seçilen duyguları (Öfkeli, Tiksinmiş, Korkmuş, Mutlu, Üzgün, Şaşkın, Duygusuz) saklar.
  + Bu seçenek, resim işleme işlemi sırasında geçerli olacak duygu analizini belirler.

**setImage(QImage)**

* **Tanım**: İşlenmiş resmi GUI'ye aktaran fonksiyon.
* **İşlev**:
  + QImage tipindeki işlenmiş resim, QPixmap.fromImage(image) kullanılarak GUI'deki labelPictureFrame'e aktarılır.
  + Resim, pencere boyutuna uygun şekilde ölçeklenir (Qt.KeepAspectRatio).

***pictureThread.py***

**path**

* **Tanım:** Resim dosyasının dosya yolu.
* **İşlev:** path parametresi, işlenecek olan resmin dosya yolunu tutar. getCapture() fonksiyonu, bu yolu kullanarak resmi okumak için kullanılır.

**options**

* **Tanım:** Kullanıcı tarafından seçilen analiz seçeneklerini içeren bir sözlük.
* **İşlev:** Bu parametre, yüz, cinsiyet ve duygu analizi seçeneklerini içerir. options['genders'] ve options['emotions'] dizileri, kullanıcı tarafından belirlenen cinsiyet ve duygu kategorilerini tutar. Bu parametre, yüz analizinin nasıl yapılacağını belirler.

**run()**

* **Parametreler:** self (class instance).
* **İşlev:** run() fonksiyonu, PictureThread sınıfının çalıştırılmasını sağlar. Bu metodda, yüz tanıma işlemleri yapılır, görüntü işlenir ve sonuç PyQt5 arayüzüne aktarılır. face\_detect.run() fonksiyonu kullanılarak yüz tanıma işlemi yapılır ve elde edilen sonuç changePixmap sinyaliyle GUI'ye aktarılır.

**getCapture()**

* **Parametreler:** self (class instance).
* **İşlev:** Bu fonksiyon, path ile belirtilen dosya yolunu kullanarak resmi okur ve capture parametresine atar. Bu, resmin işlenebilmesi için gereklidir.

**onvertToQtFormat**

* **Tanım:** OpenCV görüntüsünü Qt formatına dönüştüren QImage objesi.
* **İşlev:** frame (OpenCV formatında bir görüntü) PyQt5 tarafından kullanılabilecek bir QImage formatına dönüştürülür. Bu format, PyQt5 GUI üzerinde gösterilebilir.