Project 3 Task1: Face Detection

Stella Liao

50321222

In this face detection project, I applied the OpenCV Haar feature-based cascade classifier, proposed by Pau Viola and Michael Jones. This cascade classifier is more of a general object detection method, and here I used it as a face detector.

The Haar feature-based cascade classifier is originally provided by the OpenCV cv2.CascadeClassifier module, and I adopted this module and tried 3 different pre-trained models

- haarcascade_frontalface_alt.xml:
  a stump-based 20x20 gentle adaboost frontal face detector.
- haarcascade_frontalface_alt2.xml:
  a tree-based 20x20 gentle adaboost frontal face detector
- haarcascade_frontalface_default.xml:
  a stump-based 24x24 discrete adaboost frontal face detector

The *cv2.CascadeClassifier* module provides an interface *detectMultiScale* which takes a gray-scale image as input and returns a list of detected faces in rectangles encoded by (x, y, w, h). The interface *detectMultiScale* can also be tuned with several parameters, e.g.,

- scaleFactor:
  Parameter specifying how much the image size is reduced at each image scale.
- minNeighbors:
  Parameter specifying how many neighbors each candidate rectangle should have to retain it.

In the implementation, I set scaleFactor to 1.1, and minNeighbors to 3. And every image fed into the detector, I convert it to a gray-scaled image.

Model	F1 Score
haarcascade_frontalface_alt	0.8113207547169811
haarcascade_frontalface_alt2	0.7987421383647799
haarcascade_frontalface_default	0.7764705882352942

Table 1. F1 scores of the three models

According to table 1, I chose haarcascade_frontalface_alt as the final model to detect faces in this case, since it has better performance than the other two models.