Notes on Face Anonymiser

12 December 2024 12:19

What is a Haar Cascade Classifier?

A **Haar Cascade Classifier** is a machine learning object detection method used to identify objects in images or video. In the context of OpenCV, it's typically used for detecting faces, eyes, or other objects. The term "Haar" comes from Haar-like features, which are simple features based on the intensity of pixel values in rectangular regions of an image.

Key Concepts of Haar Cascade Classifier:

- **Haar Features**: Haar features are essentially rectangular patterns of pixel intensities, which are calculated by comparing sums of pixel intensities in different regions of an image. These features are used to distinguish between object and non-object regions. They are inspired by the work of Haar wavelets in image processing.

- **Cascade Classifier**: A classifier is a trained machine learning model that can classify objects (for example, detecting faces). The "cascade" refers to a series of increasingly complex stages where simpler, less computationally expensive tests are done first, and more detailed tests are only applied when necessary. The idea is to quickly eliminate non-object regions of the image.

- **Training**: The Haar Cascade Classifier needs to be trained using positive and negative images. In OpenCV, however, you don't need to train it yourself since OpenCV provides pre-trained classifiers (like the frontal face classifier).

How Does cv2.CascadeClassifier Work?

1. **Pre-trained Model**: OpenCV provides several pre-trained Haar Cascade classifiers that are used for face detection, eye detection, etc. These models are trained using thousands of positive (containing faces) and negative (non-faces) images.
2. **Feature Calculation**: For each region of interest (such as a window of an image), Haar features are calculated and classified by a series of weak classifiers. If the region is likely to contain a face, it proceeds to the next stage of the cascade.
3. **Detection Process**: The classifier is applied to different regions of an image. The cascade classifier slides over the image in steps, checking different regions of varying sizes, and each time, it applies the classifier to determine if it contains a face or not.

How to Use cv2.CascadeClassifier:

1. **Loading the Classifier:** import cv2
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
(code snippet)

2. **Reading the Image : img= cv2.imread(image path)**

3. Converting to Grayscale : gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

4. **Detecting Faces :** faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30)

5. **Drawing Rectangles Around Faces :** cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 2)

6. **Displaying the Image :** cv2.imshow('Face Detection', image)