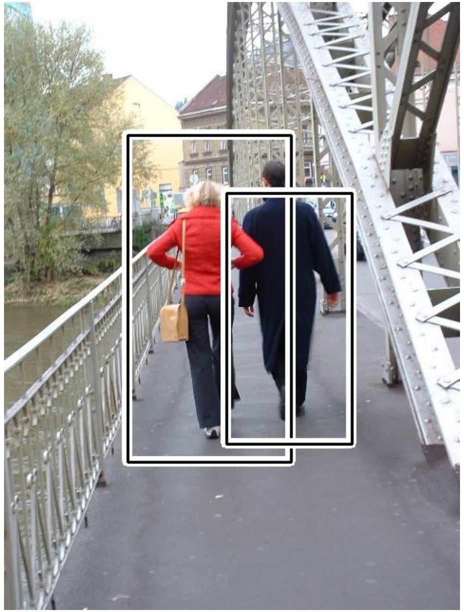
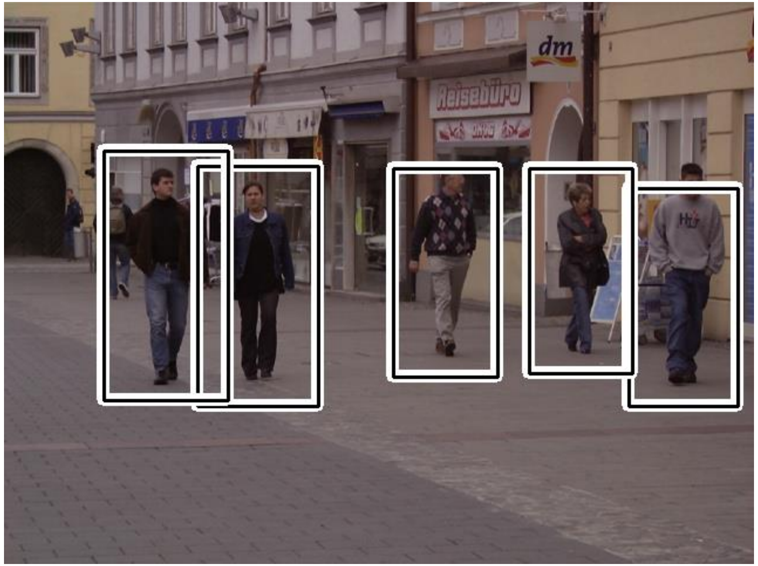
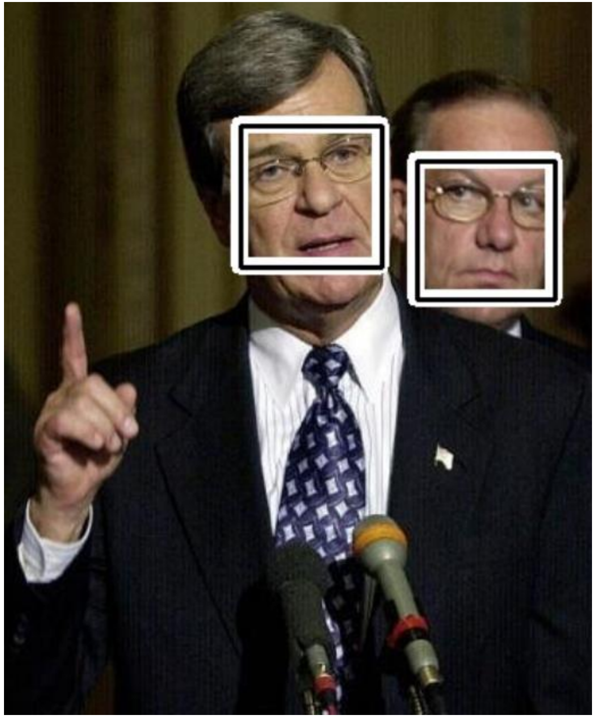
Object/Face Detection

What is the output of object detection methods?

- **Position of the object of interest**
- **Scale of the object of interest**







Object/Face Detection

- Haar

Cascade classifier in OpenCV

Paul Viola and Michael Jones
Rapid Object Detection using a Boosted Cascade of Simple Features

- HOG
- LBP

Traditional Approaches (slidindg window)

- SIFT, SURF

KeyPoints

- CNNs
- R-CNNs/YOLO/SSD

Deep Learning Approach

Face Detection

Intro into Face Detection

- **Sliding Window**

- In general, the sliding window technique represents the popular and successful approach for object detection. The main idea of this approach is that the input image is scanned by a rectangular window at multiple scales. The result of the scanning process is a large number of various sub-windows. A vector of features is extracted from each sub-window. The vector is then used as an input for the classifier (e.g. SVM classifer).

- During the classification process, some sub-windows are marked as the objects. Using the sliding window approach, the multiple positive detections may appear, especially around the objects of interest

Face Detection

Intro into Face Detection

- **Sliding Window**

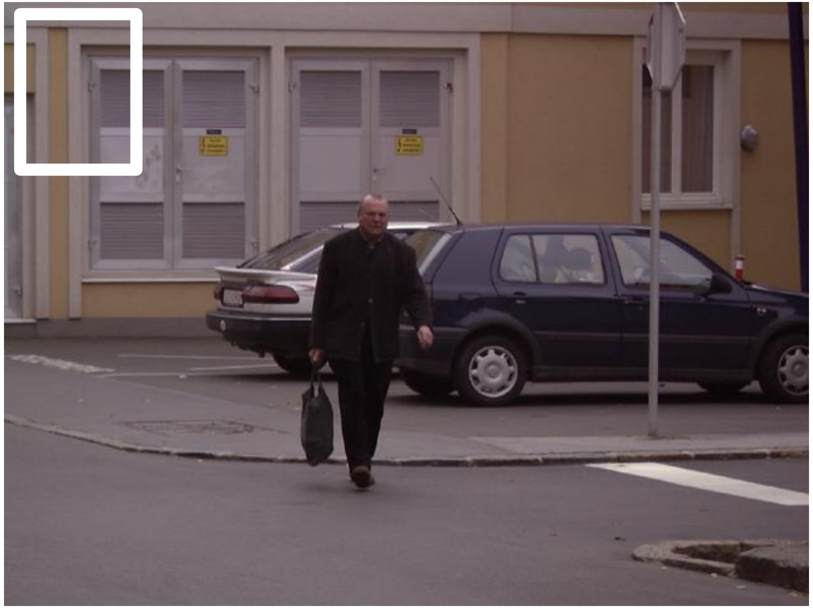
- These detections are merged to the final bounding box that represents the resulting detection.

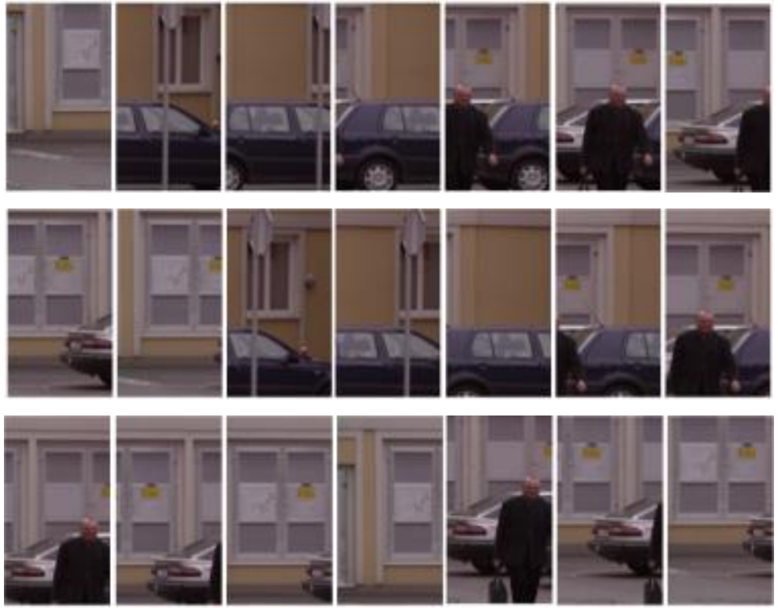
- The classifer that determines each sub-window is trained over the training set that consists of positive and negative images.

- The key point is to find what values (features) should be used to effectively encode the image inside the sliding window.

Intro into Face Detection

- Sliding Window

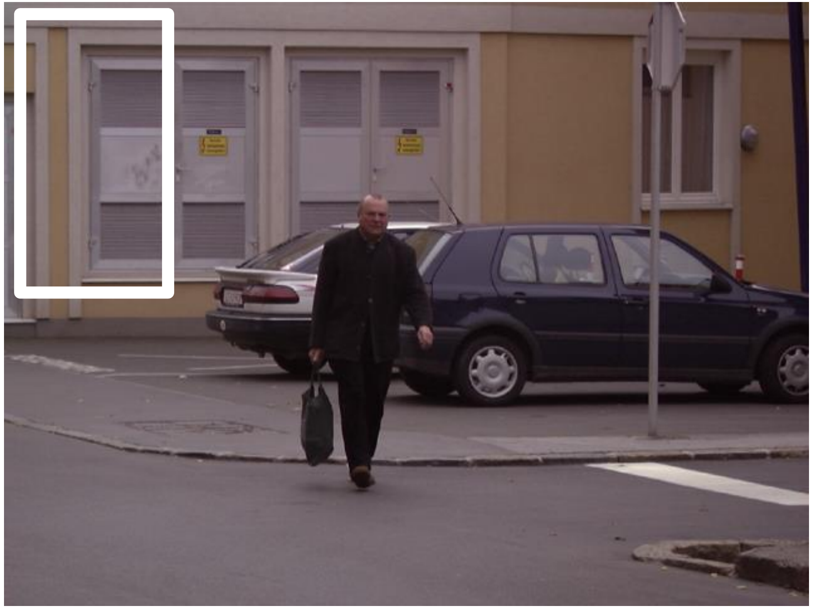


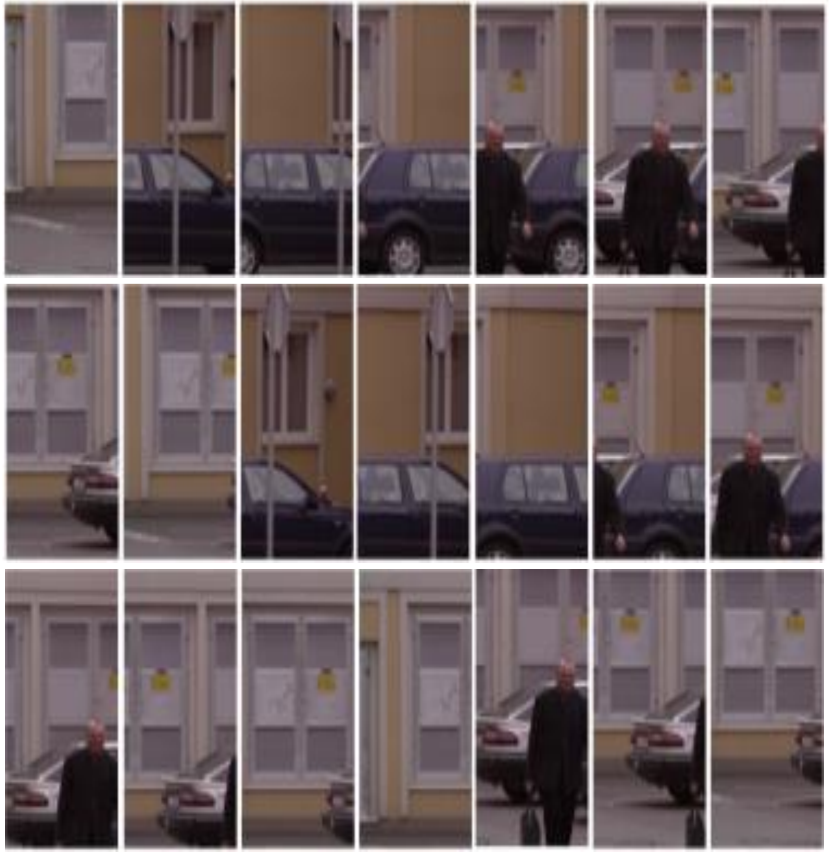


Face Detection

Intro into Face Detection

- Sliding Window

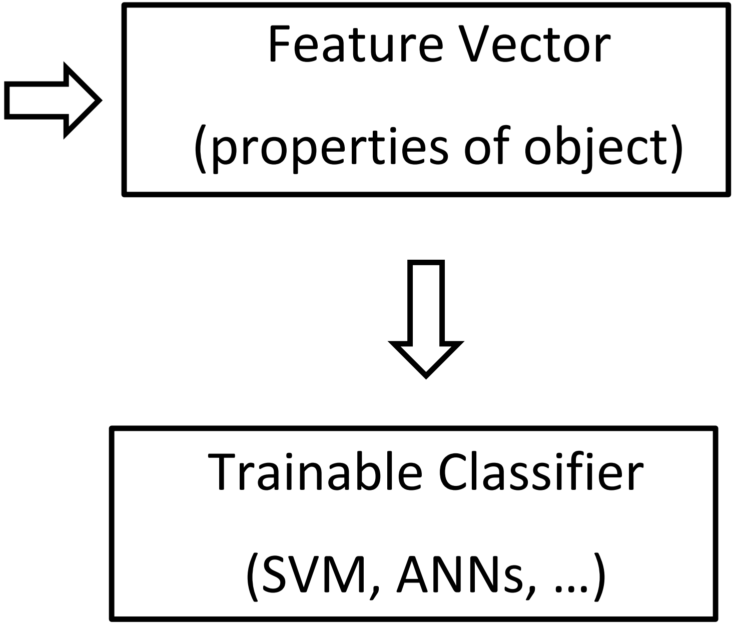




Face Detection

Intro into Face Detection

- Sliding Window

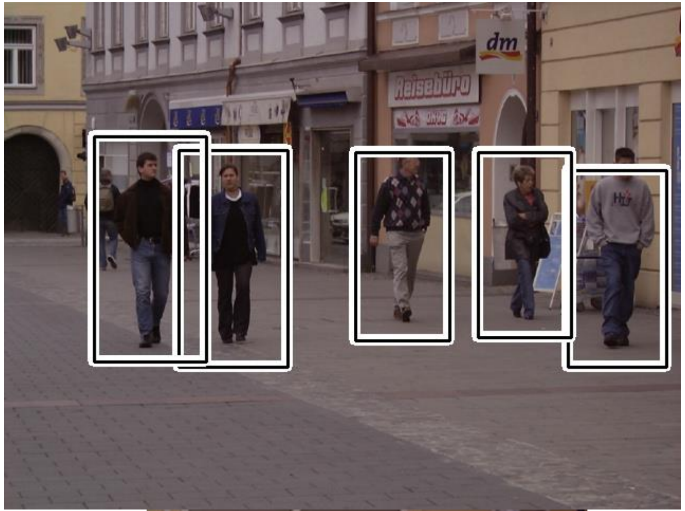


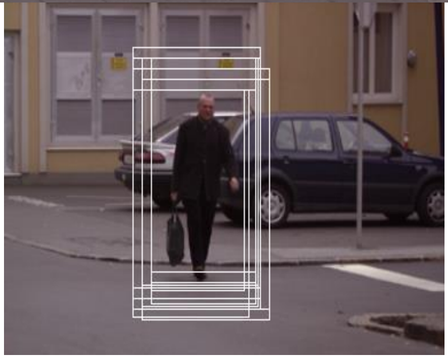
Feature Vector

(properties of object)

Trainable Classifier

(SVM, ANNs, …)





Face Detection

Face Detection in OpenCV using cascade classifier

1. **Rectangle (Haar features):**

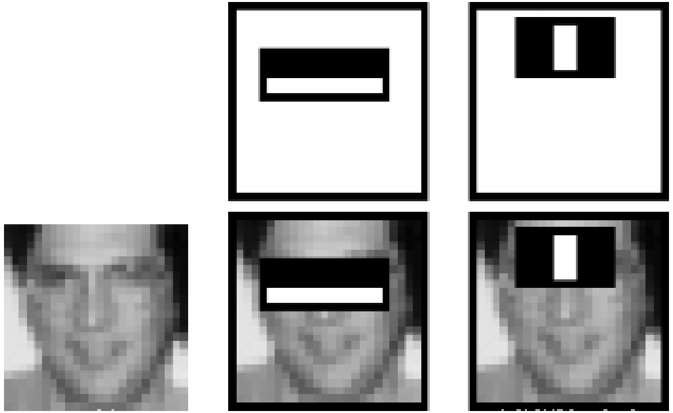
- faces have similar properties
- eye regions are darker than the upper-cheeks
- the nose bridge region is brighter than the eyes
- thousands possible

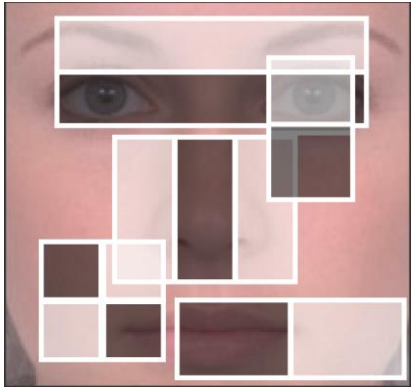
2. **Integral Image**

- speed the computational process

3. **Cascade Classifier + AdaBoost**

- in an image, most of the image is non-face region
- reject the non-face region as soon as possible

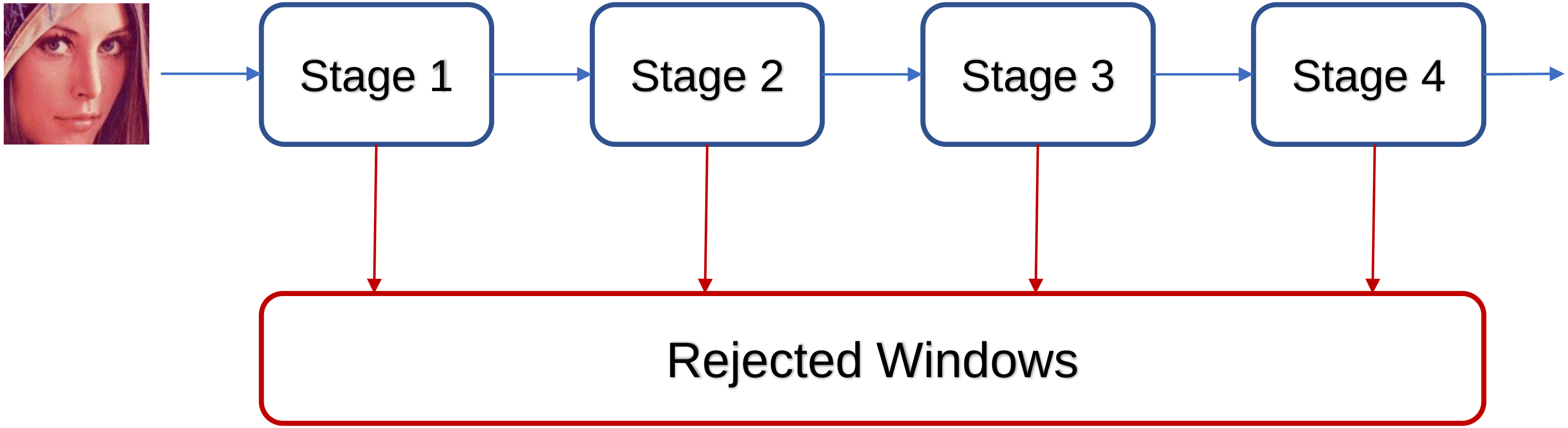




David Gerónimo: Haar-like Features and Integral Image Representation, Master in Computer Vision and Artificial Intelligence, 18th December 2009
https://docs.opencv.org/4.5.5/db/d28/tutorial_cascade_classifier.html

Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



Stage 1

Stage 2

Stage 3

Stage 4

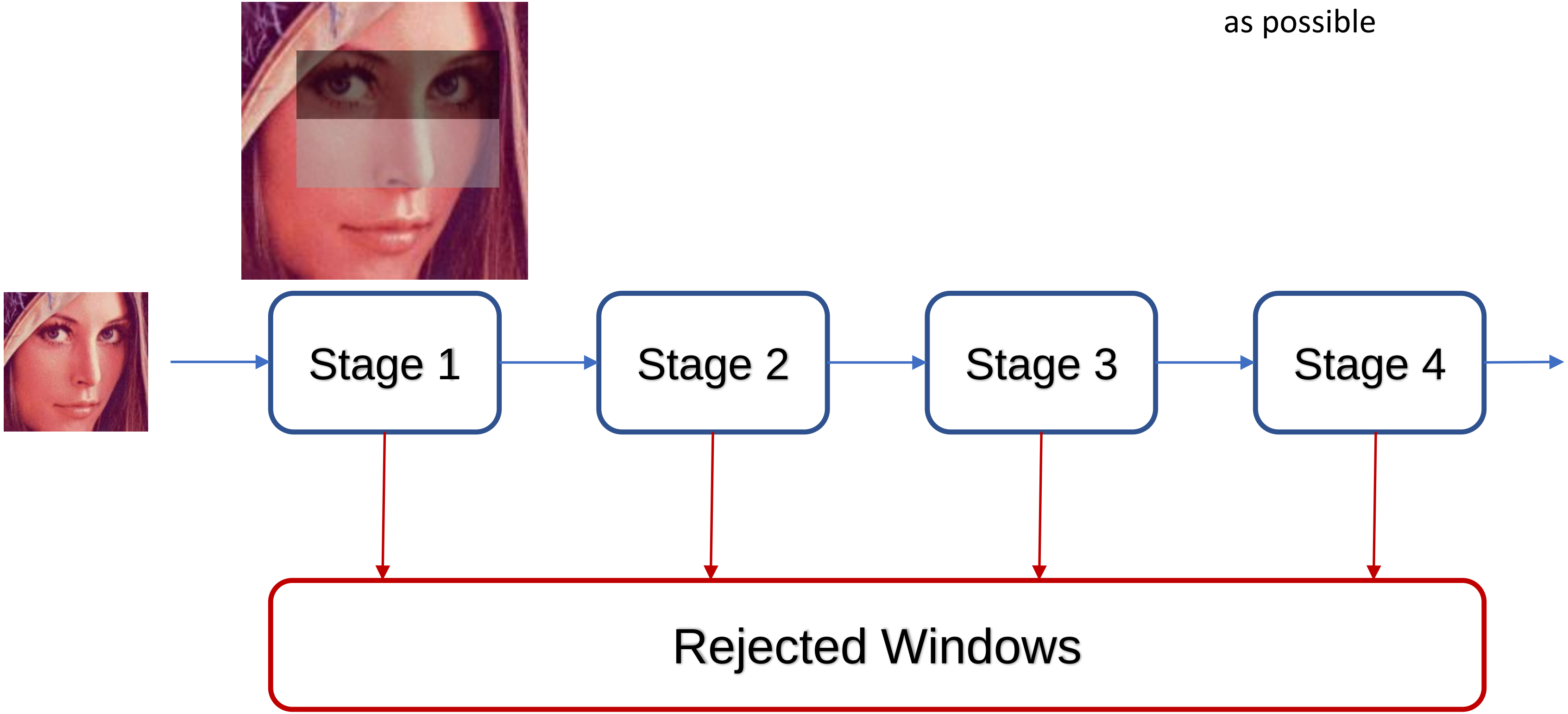
Rejected Windows

https://docs.opencv.org/4.5.5/db/d28/tutorial_cascade_classifier.html

Face Detection

The idea of cascade classifier is reject the non-face region as soon as possible

Cascade of Classifier



Stage 1

Stage 2

Stage 3

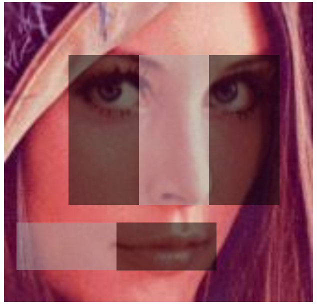
Stage 4

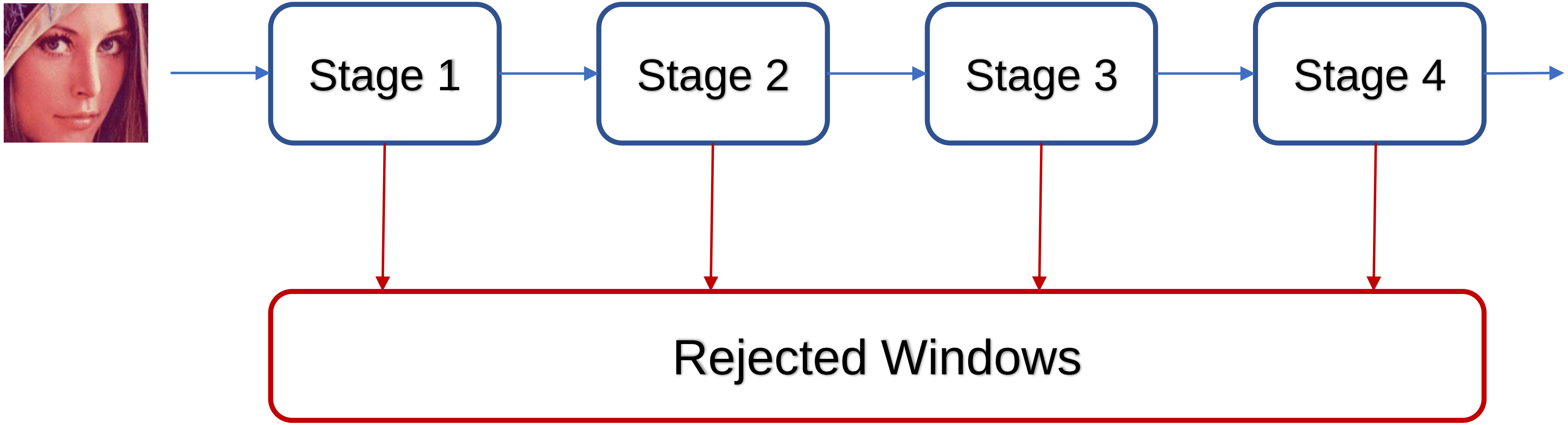
Rejected Windows

Face Detection

The idea of cascade classifier is reject the non-face region as soon as possible

Cascade of Classifier

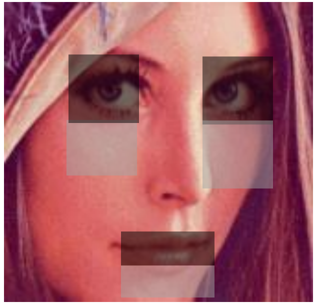


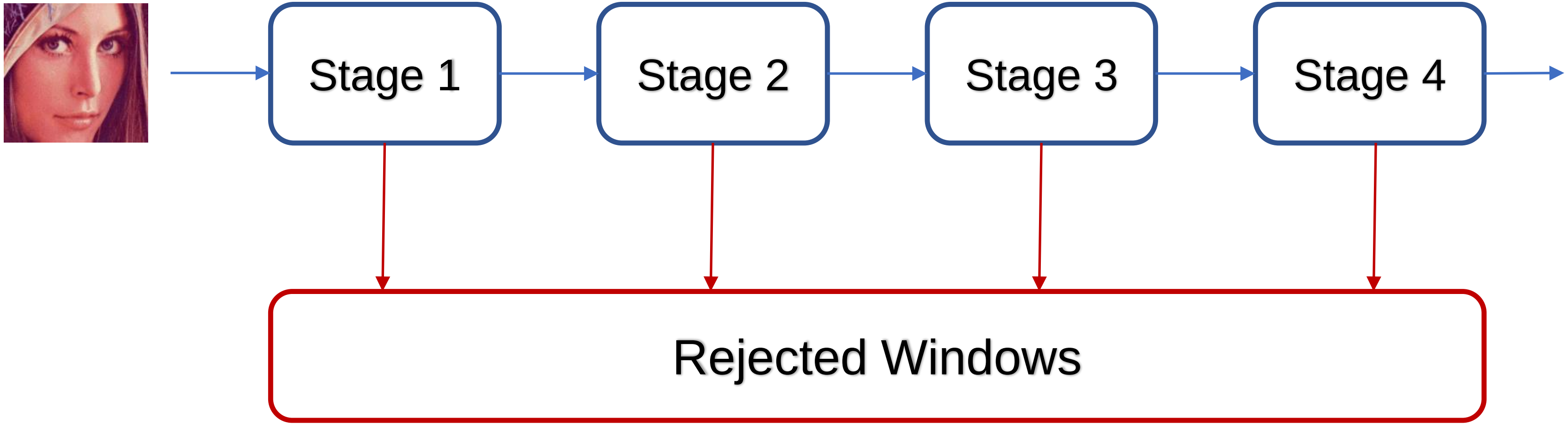


Face Detection

Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible





Stage 1

Stage 2

Stage 3

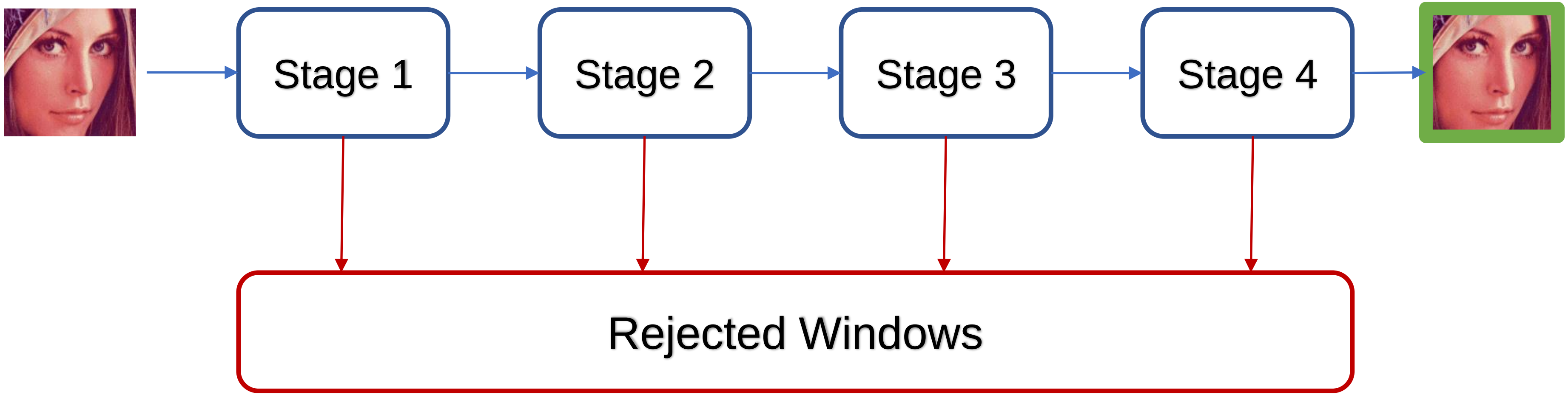
Stage 4

Rejected Windows

Face Detection

The idea of cascade classifier is reject the non-face region as soon as possible

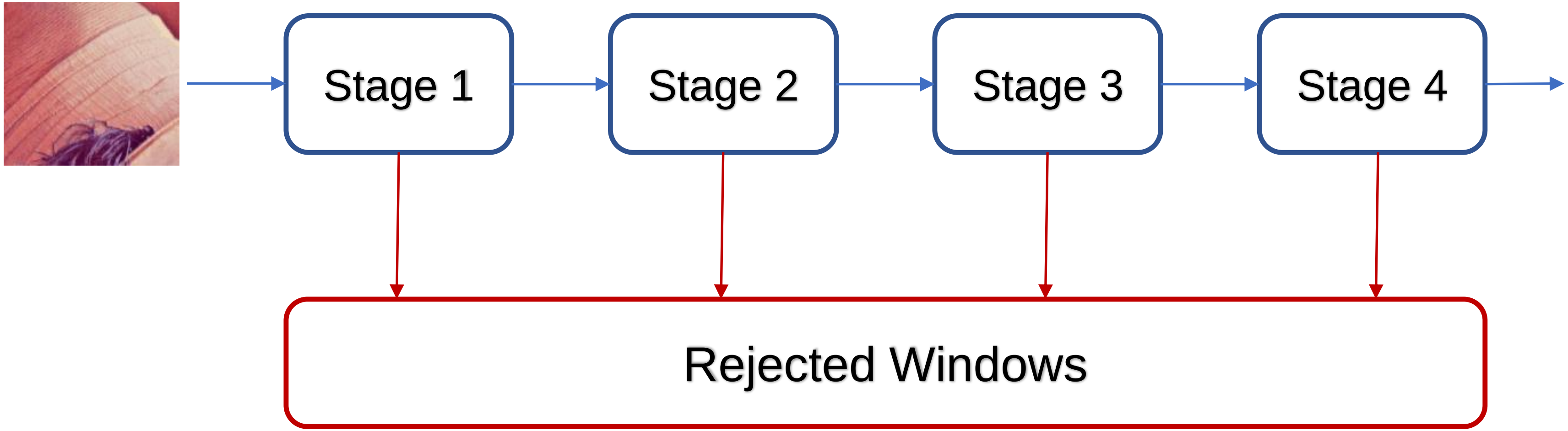
Cascade of Classifier



Face Detection

Cascade of Classifier

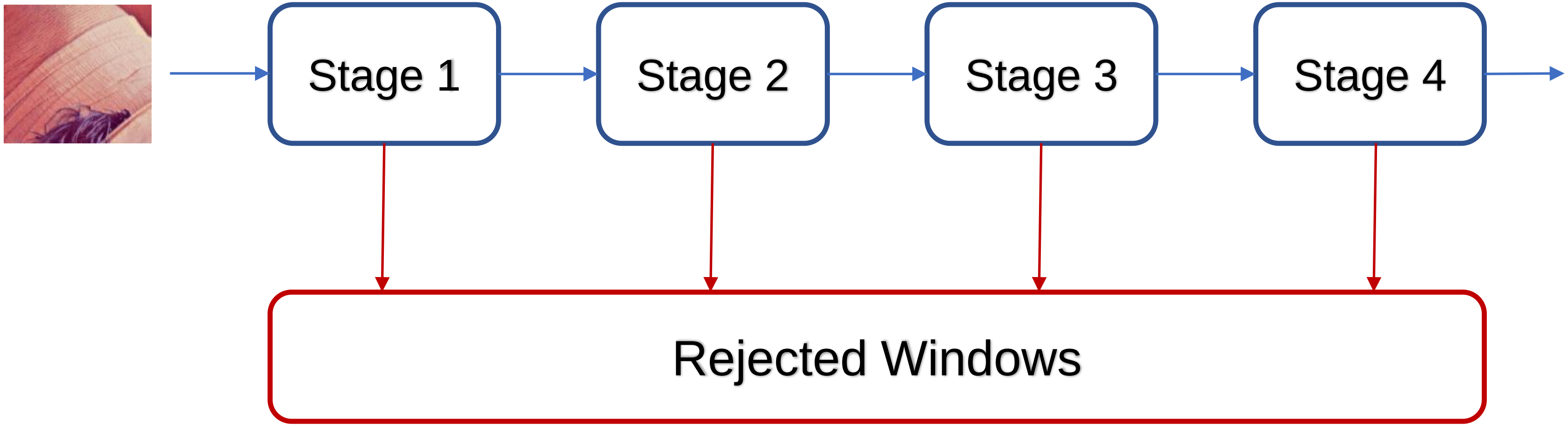
The idea of cascade classifier is reject the non-face region as soon as possible



Face Detection

The idea of cascade classifier is reject the non-face region as soon as possible

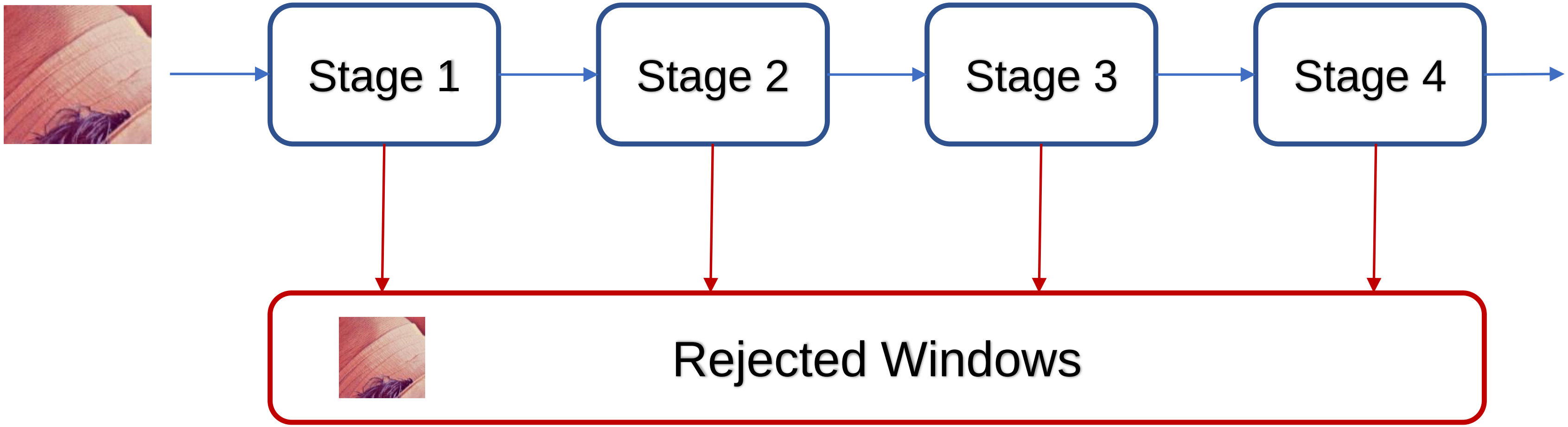
Cascade of Classifier



Face Detection

Cascade of Classifier

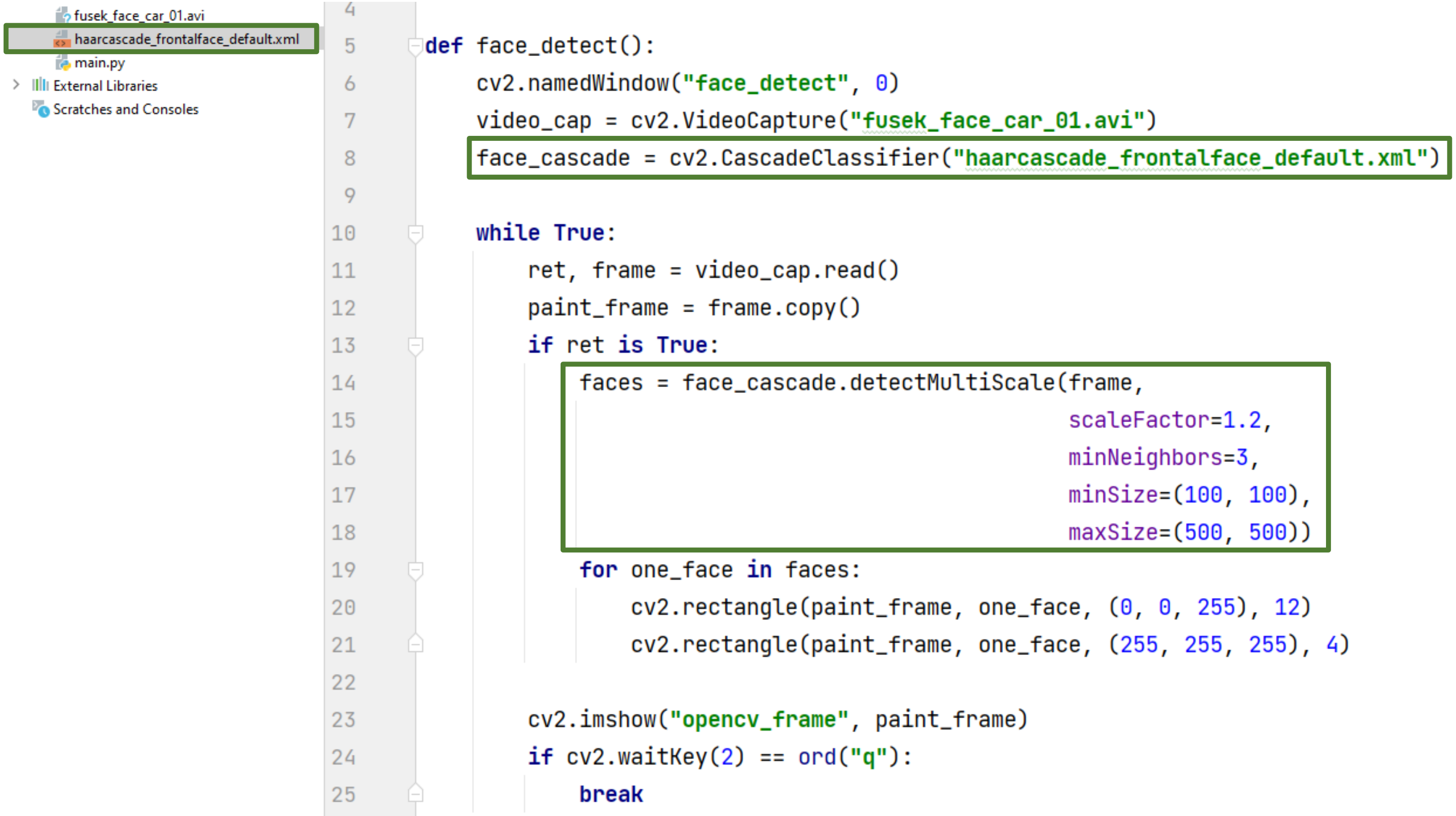
The idea of cascade classifier is reject the non-face region as soon as possible



Face Detection - OpenCV

https://vimeo.com/12774628

Face Detection - OpenCV

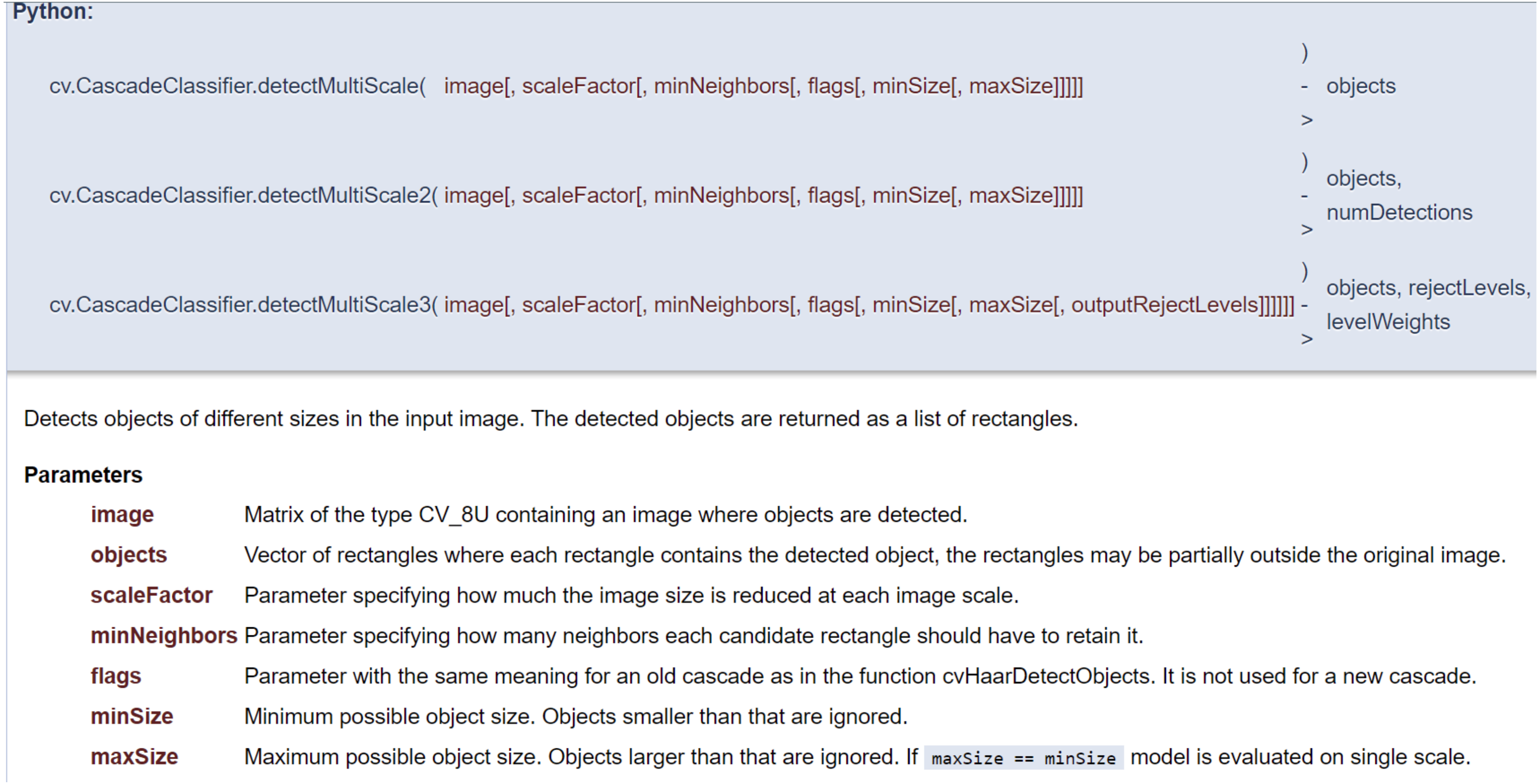


```python
def face_detect():
    cv2.namedWindow("face_detect", 0)
    video_cap = cv2.VideoCapture("fusek_face_car_01.avi")
    face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

    while True:
        ret, frame = video_cap.read()
        paint_frame = frame.copy()
        if ret is True:
            faces = face_cascade.detectMultiScale(frame,
                                                  scaleFactor=1.2,
                                                  minNeighbors=3,
                                                  minSize=(100, 100),
                                                  maxSize=(500, 500))
            for one_face in faces:
                cv2.rectangle(paint_frame, one_face, (0, 0, 255), 12)
                cv2.rectangle(paint_frame, one_face, (255, 255, 255), 4)

        cv2.imshow("opencv_frame", paint_frame)
        if cv2.waitKey(2) == ord("q"):
            break
```

Face Detection - OpenCV



Python:

```
cv.CascadeClassifier.detectMultiScale(  image[, scaleFactor[, minNeighbors[, flags[, minSize[, maxSize]]]]]        ) -> objects
cv.CascadeClassifier.detectMultiScale2( image[, scaleFactor[, minNeighbors[, flags[, minSize[, maxSize]]]]]        ) -> objects, numDetections
cv.CascadeClassifier.detectMultiScale3( image[, scaleFactor[, minNeighbors[, flags[, minSize[, maxSize[, outputRejectLevels]]]]]] ) -> objects, rejectLevels, levelWeights
```

Detects objects of different sizes in the input image. The detected objects are returned as a list of rectangles.

Parameters

image Matrix of the type CV_8U containing an image where objects are detected.

objects Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.

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.

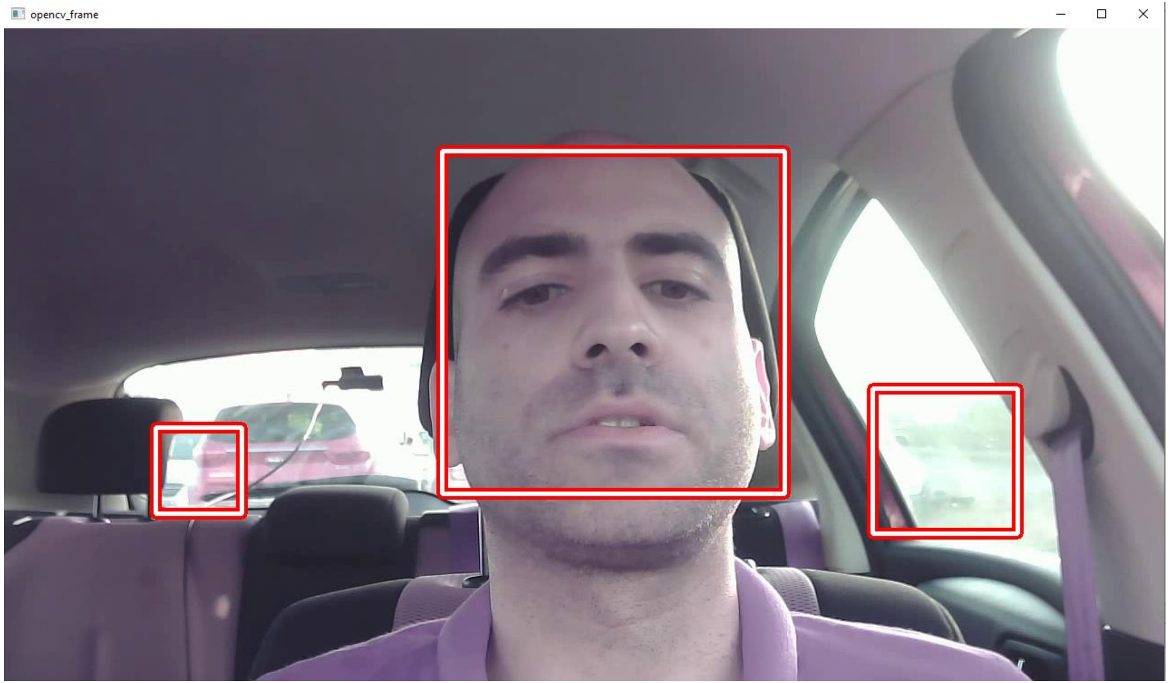
flags Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade.

minSize Minimum possible object size. Objects smaller than that are ignored.

maxSize Maximum possible object size. Objects larger than that are ignored. If `maxSize == minSize` model is evaluated on single scale.

https://docs.opencv.org/4.5.5/d1/de5/classcv_1_1CascadeClassifier.html#aaf8181cb63968136476ec4204ffca498

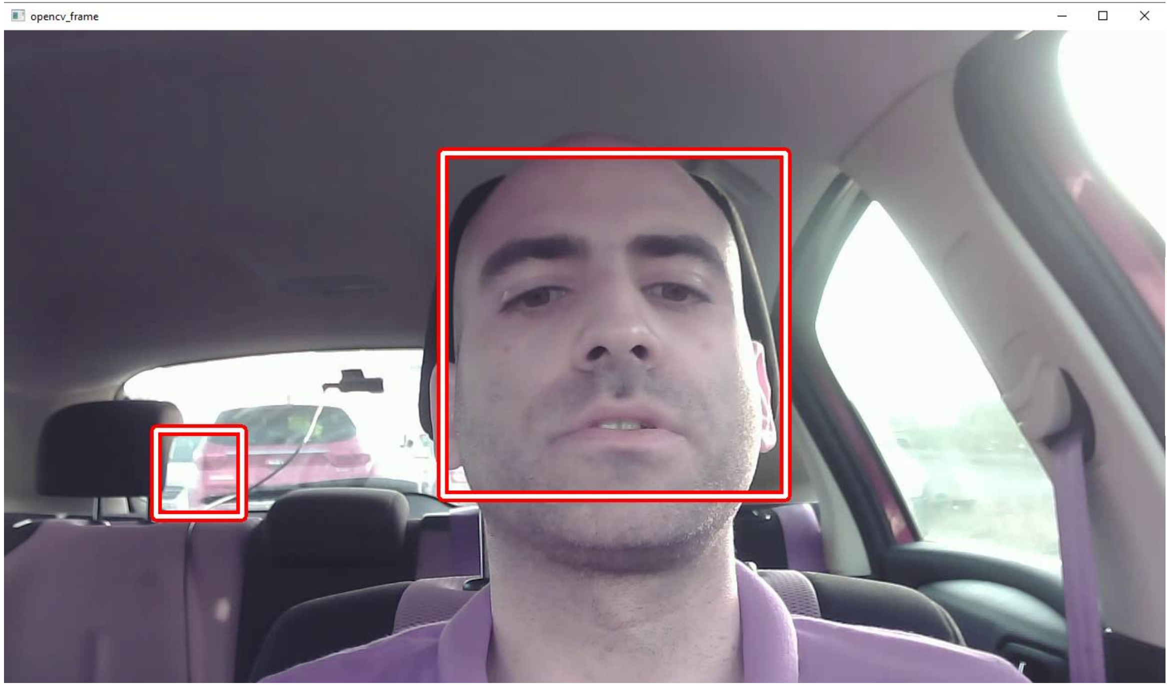
Face Detection - OpenCV



```
faces = face_cascade.detectMultiScale(frame,
                                      scaleFactor=1.1,
                                      minNeighbors=1,
                                      minSize=(50, 50),
                                      maxSize=(500, 500))
```

https://docs.opencv.org/4.5.5/d1/de5/classcv_1_1CascadeClassifier.html#aaf8181cb63968136476ec4204ffca498

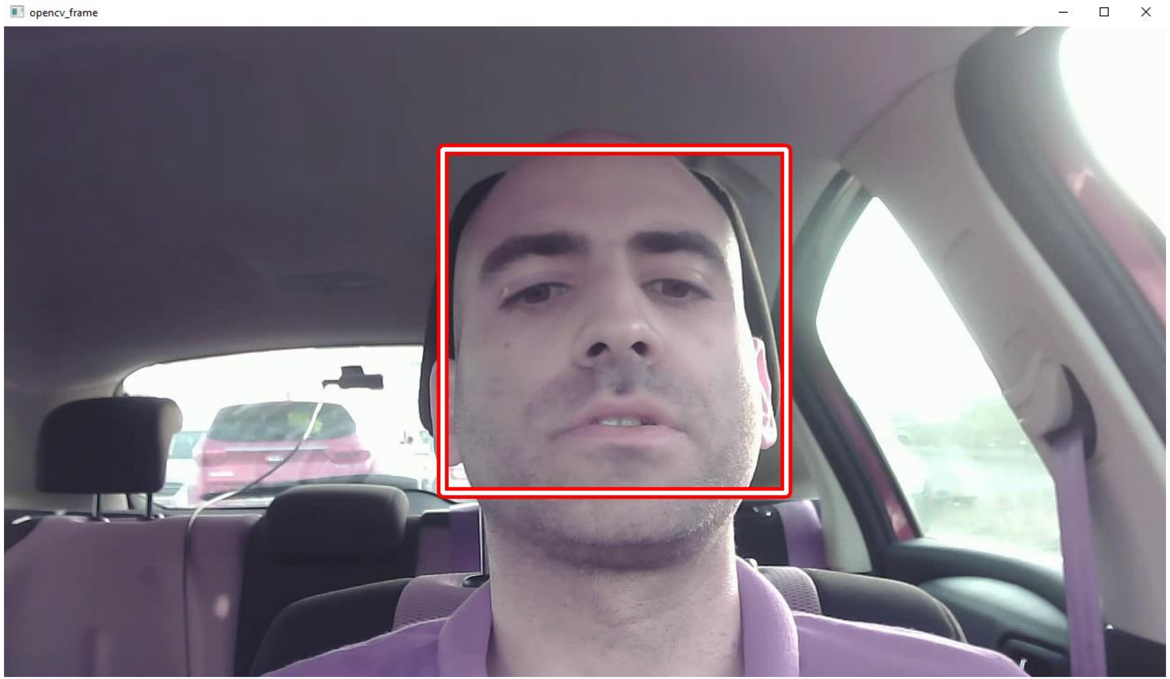
Face Detection - OpenCV



```
faces = face_cascade.detectMultiScale(frame,
                                      scaleFactor=1.1,
                                      minNeighbors=3,
                                      minSize=(50, 50),
                                      maxSize=(500, 500))
```

https://docs.opencv.org/4.5.5/d1/de5/classcv_1_1CascadeClassifier.html#aaf8181cb63968136476ec4204ffca498

Face Detection - OpenCV



```
faces = face_cascade.detectMultiScale(frame,
                                      scaleFactor=1.1,
                                      minNeighbors=3,
                                      minSize=(100, 100),
                                      maxSize=(500, 500))
```

https://docs.opencv.org/4.5.5/d1/de5/classcv_1_1CascadeClassifier.html#aaf8181cb63968136476ec4204ffca498