

Face detection

Lab 1

Train a cascade object detector

- ◇ Reference: "Train A Stop Sign Detector" in the help document.
- ◇ Train a stop sign detector using the Viola-Jones algorithm

1. Load the positive samples data from a .mat file.

```
load('stopSigns.mat');
```

2. Add images location to the MATLAB path.

```
% $matlabroot/toolbox/vision/visiondata/stopSignImages  
imDir = fullfile(matlabroot, 'toolbox', 'vision', 'visiondata',  
'stopSignImages');  
addpath(imDir);
```

Train a cascade object detector

- ◇ Train a stop sign detector using the Viola-Jones algorithm

3. Specify the folder for negative images.

```
% $matlabroot/toolbox/vision/visiondata/nonStopSigns  
negativeFolder = fullfile(matlabroot,'toolbox','vision',  
    'visiondata','nonStopSigns');
```

3. Train a cascade object detector called 'stopSignDetector.xml'

```
trainCascadeObjectDetector('stopSignDetector.xml',data,negativeFo  
lder,'FalseAlarmRate',0.2,'NumCascadeStages',5);
```

Train a cascade object detector

◇ Train a stop sign detector using the Viola-Jones algorithm

5. Use the newly trained classifier to detect a stop sign in an image.

```
detector = vision.CascadeObjectDetector('stopSignDetector.xml');
```

6. Read the test image.

```
img = imread('stopSignTest.jpg');
```

Train a cascade object detector

◇ Train a stop sign detector using the Viola-Jones algorithm

7. Detect a stop sign.

```
bbox = step(detector,img); % or detector.step(img);
```

8. Insert bounding boxes and return marked image.

```
detectedImg = insertObjectAnnotation(img,'rectangle',bbox,'stop  
sign');
```

9. Display the detected stop sign.

```
figure;
```

```
imshow(detectedImg);
```

Train a cascade object detector

- ◆ Train a stop sign detector using the Viola-Jones algorithm

7. Detect a stop sign.

```
bbox = step(detector, img);
```

8. Insert bound

```
detectedImg =  
img(bbox, :);  
figure;  
imshow(detectedImg,  
'sign');
```

9. Display the d

```
figure;  
imshow(detectedImg,  
'sign');
```



Detecting face from an image

- ◇ `vision.CascadeObjectDetector()`

```
detector = vision.CascadeObjectDetector
```

```
detector = vision.CascadeObjectDetector(MODEL)
```

```
detector = vision.CascadeObjectDetector(XMLFile)
```

- ◇ `MODEL` is a string: 'FrontalFaceCART' (default), 'FrontalFaceLBP', 'UpperBody', 'EyePairBig', 'EyePairSmall', 'LeftEye', 'RightEye', 'ProfileFace', 'ProfileFace', 'Nose'.

Detecting face from an image

◇ Reference: "vision.CascadeObjectDetector System object"

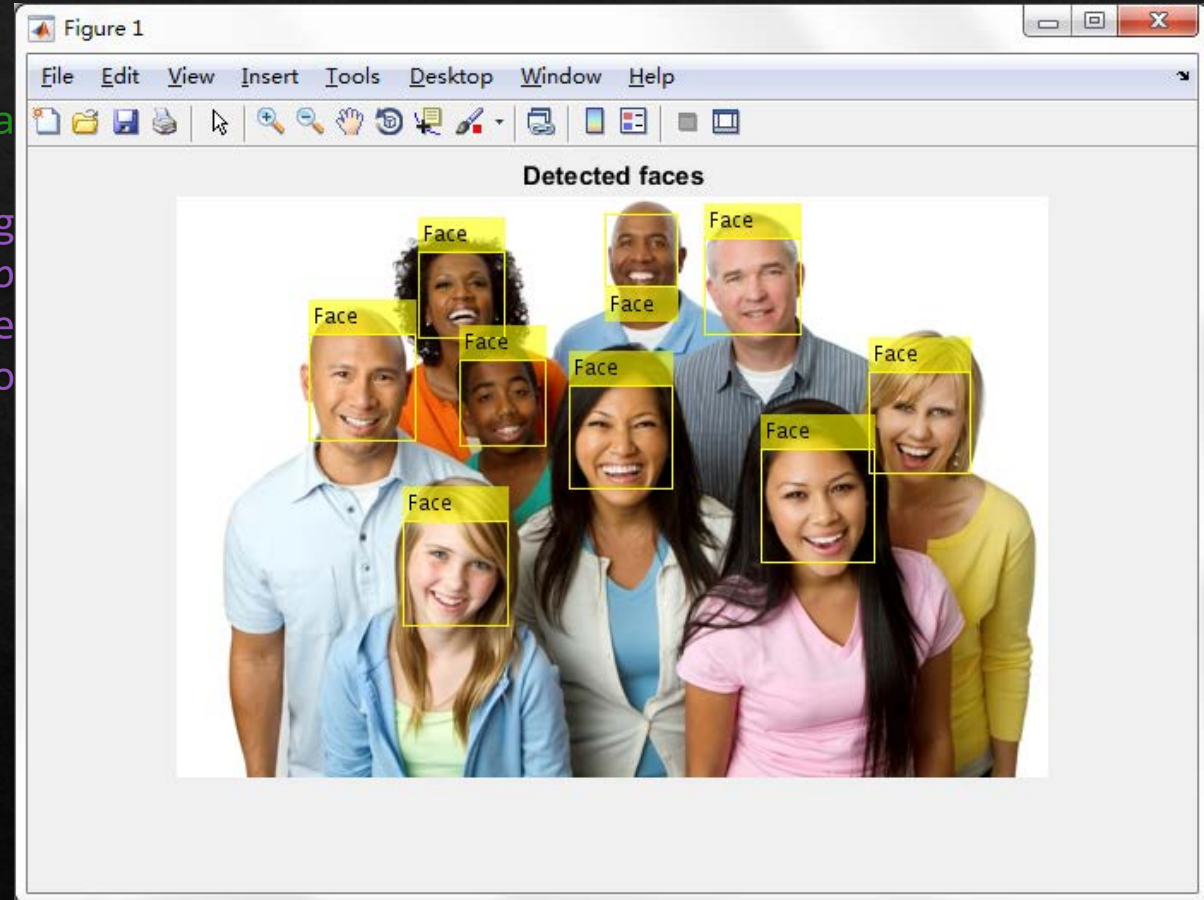
◇ Example

```
% Create a cascade detector object.  
faceDetector = vision.CascadeObjectDetector();  
I = imread('faces2.jpg');  
bboxes = step(faceDetector, I); % faceDetector.step(I);  
IFaces = insertObjectAnnotation(I, 'rectangle', bboxes, 'Face');  
figure, imshow(IFaces), title('Detected faces');
```


Detecting face from an image

◇ Example

```
% Create a ca  
faceDetector  
I = imread('g  
bboxes = step  
IFaces = inse  
figure, imsho
```



Detecting face from a video

◇ Example

```
% Create a cascade detector object.  
faceDetector = vision.CascadeObjectDetector();  
% Read a video frame and run the face detector.  
videoFileReader = vision.VideoFileReader('visionface.avi');  
videoInfo      = info(videoFileReader);  
videoPlayer    = vision.VideoPlayer('Position',[300 300  
videoInfo.VideoSize+30]);
```

Detecting face from a video

◇ Example (Cont.)

```
while ~isDone(videoFileReader)
    % Extract the next video frame
    videoFrame = step(videoFileReader);
    bbox = step(faceDetector, videoFrame);
    % Insert a bounding box around the object being tracked
    videoOut = insertObjectAnnotation(videoFrame, 'rectangle', bbox, 'Face');
    step(videoPlayer, videoOut);
end
% Release resources
release(videoFileReader);
release(videoPlayer);
```

Detecting face from the camera

- ◇ Reference: "Face Detection and Tracking Using Live Video Acquisition" in MATLAB help document

Exercise

- ◇ Run previous examples
- ◇ Finish the following task
 - ◇ Mask the detected face in the video "test.mov" using a cartoon image ("test.mov" can be found in Lab1_example);
 - ◇ The size of cartoon image should change according to the size of the detected face;
- ◇ Submit your work to TA
 - ◇ Compress the whole project folder into a zip file("ID_name_lab1.zip")
 - ◇ 1350588-1452737: js_ilab@163.com
 - ◇ 1452741-1452844: 0628yulu@tongji.edu.cn

Notes

- ◇ `videoFrame = step(videoFileReader)`
 - ◇ Detect the face from `videoFrame`
- ◇ Resize the cartoon image
 - ◇ `imresize()`
- ◇ Cover the detected face using the resized cartoon image
 - ◇ Set the values of detected face by those of resized cartoon image
 - ◇ `videoFrame` is **SINGLE** type (from 0~1)
 - ◇ The resized image is **UINT8** type
 - ◇ Use `im2single()` to convert `uint8` to `single`
- ◇ When you finish the exercise, you can try to detect your fist from the camera using "aGest.xml".