Facial Detection in Images

Cooper Yancey
EE 384-01 Fall 2021
Final Design Project

Running the Project

- Install computer vision toolbox, install image processing toolbox, install deep learning toolbox.
- Install `MTCNN.Face.Detection.mltbx` file provided or from the link: https://github.com/matlab-deep-learning/mtcnn-face-detection
- Run the code, and give the photo you want to run in the file dialog box. Please make sure it is a .png or .jpg file.

My Process

- I initially tried to use edge detection and Kekre transform, but that was becoming very cumbersome and complicated. I eventually found in the computer vision toolbox there was included an included computer vision program that would already find faces in an image using the Viola-Jones algorithm, which can detect faces, noses, eyes, mouth or upper body.

Voila-Jones Face Detection

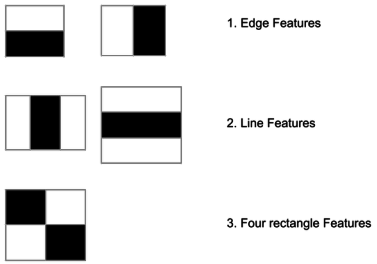
- I just had to create a face detector by typing `vj_face_detector = vision.CascadeObjectDetector;` and `vj_face_detector.MergeThreshold = 7;` to increase the accuracy of the face detection.
- I then created a shape inserting function from the vision toolbox by running `shape_inserter = vision.ShapeInserter('BorderColor','Custom','Custom BorderColor',[0 0 255]);` which put a blue border around each face detected in the Viola-Jones method. I then went to detect the faces and have the boxes around them into a variable called *bboxes* by running: `bboxes = step(vj_face_detector, img);`

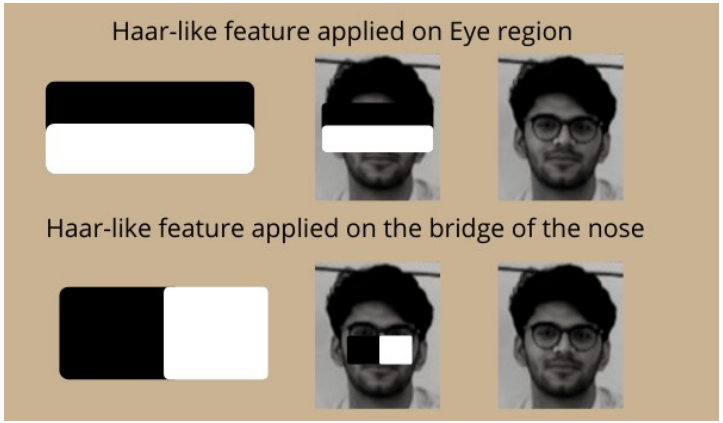
Voila-Jones Face Detection (cont.)

- Once I had the *bboxes*, I could now check if *bboxes* was empty in an if statement to determine if any faces were detected or not. If faces ***were not*** detected, it would only output that no faces were detected in the image. If faces ***were*** detected, it would output the image with the *bboxes* overlapped the image using the *shape_inserter* we instantiated.

The Issue with Voila-Jones

- Voila-Jones would not work on faces that were askew, tilted, or from the side. This is because of how Voila-Jones works, is that it requires more of the face to be visible, or it is ineffective at determining if there is a face present.





The Issue with Voila-Jones (cont.)

The Issue with Voila-Jones (cont.)

- The implementation of Voila-Jones as it is, just does not operate within enough of an acceptable margin for me, especially given the lack of complexity to the previous image.
- Thus, I had to find a better way to detect faces in images.

Multi-Task Cascaded Convolutional Neural Network (MTCNN)

- The way forward was with a Neural Network, the more I researched. But, I did not want to have to train an entire Machine Learning program, which could take weeks or months. So, I found someone who had done something similar, and already trained the Machine Learning program with lots of data, so all it needed to do was to receive an input image, much like the syntax of the functions used in my previous attempt.

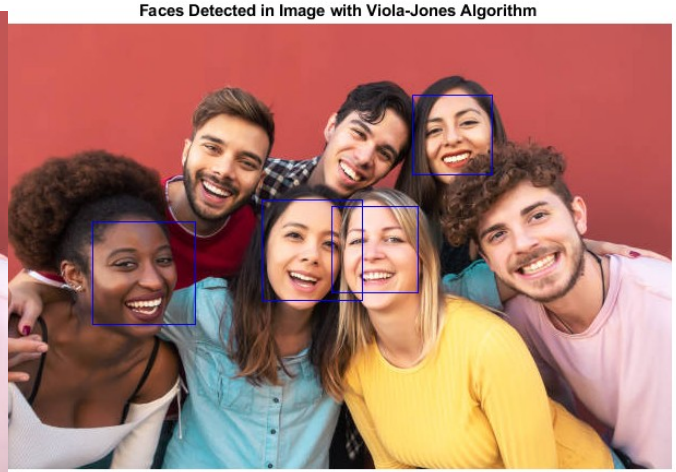
MTCNN Facial Detection

- Like the Voila-Jones method, I just needed to take in the image and store the output by running: `[bbox, scores, landmarks] = mtcnn.detectFaces(img);` The *bbox* is similar to the *bboxes* from before, the *scores* is a score from 0-1 on how confident the algorithm is, and *landmarks* is the facial features that lead the algorithm to decide it has found a human face, such as eyes, mouth, nose, etc.

MTCNN Facial Detection (cont.)

- Once I had my 3 variables, I needed to run in the *if* statement to check if *bbox* was empty. If not, I could show the image, and put the *bboxes* over a face to show the area it detected, the confidence would be shown by *scores* and the *landmarks* would show as dots over the detected landmarks.

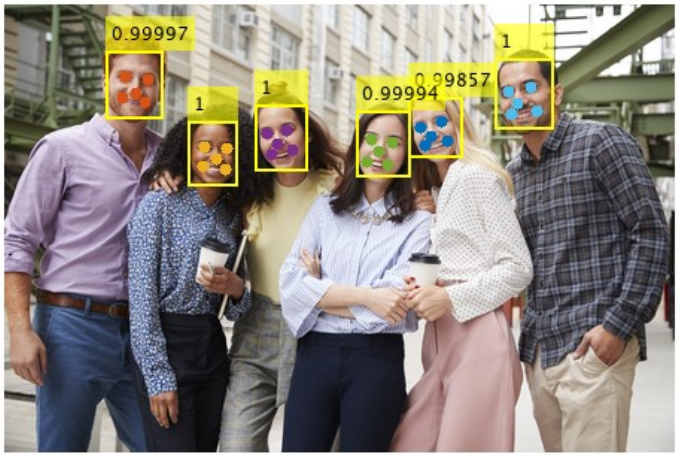
MTCNN Facial Detection (cont.)



Examples (cont.)

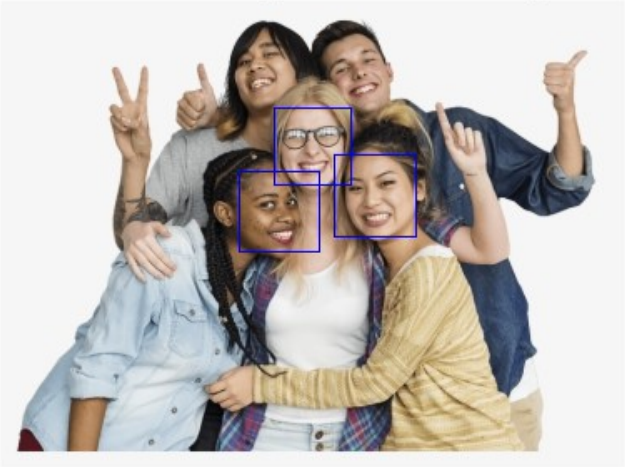
Faces Detected in Image with Viola-Jones Algorithm

Faces Detected in Image with Confidence and Features Shown with MTCNN



Examples (cont.)

Faces Detected in Image with Viola-Jones Algorithm



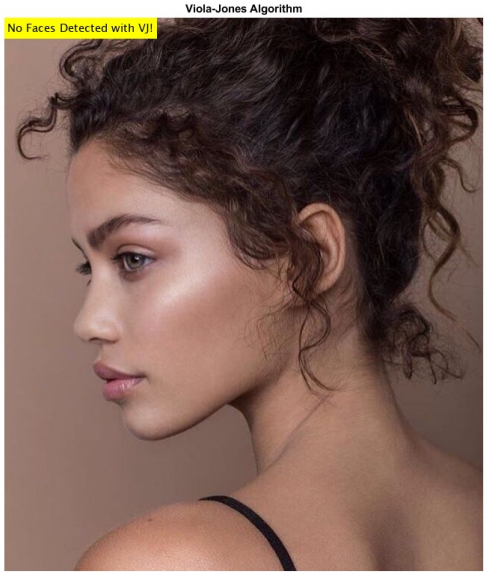
Faces Detected in Image with Confidence and Features Shown with MTCNN

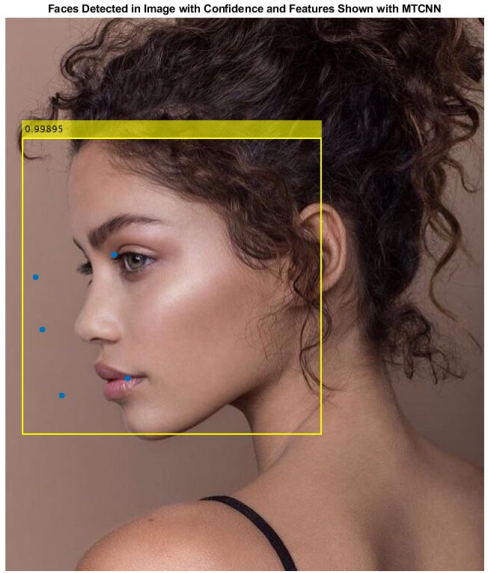
Examples

Viola-Jones Algorithm

MTCNN Algorithm

Examples (cont.)





Takeaways

- Voila-Jones can still work well once properly trained, but is limited to straight-on faces, and can produce many false negatives.
- MTCNNs are great once trained, and are very quick, can give false-positives, but at least assigns a confidence score to each, to assist in accuracy.

Source

- Source code is available and is GPLv3 at: https://github.com/vimtomk/matlab-facial-detection-ee384

- Source for the other project is available at: https://github.com/matlab-deep-learning/mtcnn-face-detection

Questions?