

Introduction to Image Recognition

@OsloMet Makerspace

Master's Universal Design of ICT

Abhijeet Sapkota



GitHub link:
(<https://github.com/makeriet/image-recognition>)

OSLOMET

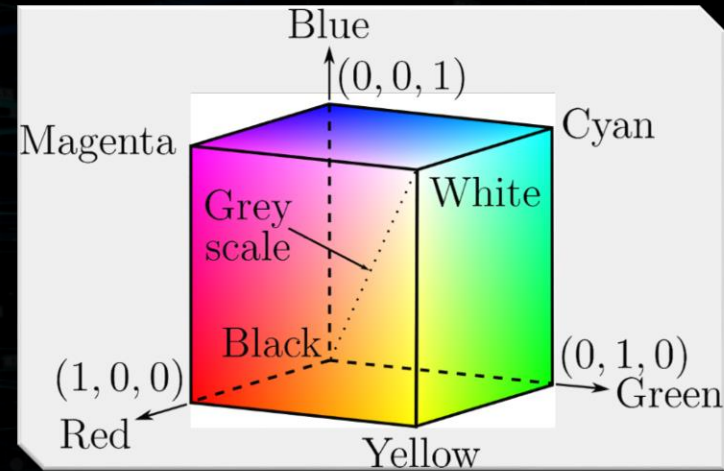
MAKERSPACE

Overview

- # Computer Vision
- # Viola-Jones Haar-Feature
- # Installing OpenCV
- # Face detection
- # Face tracking on camera
- # Template matching

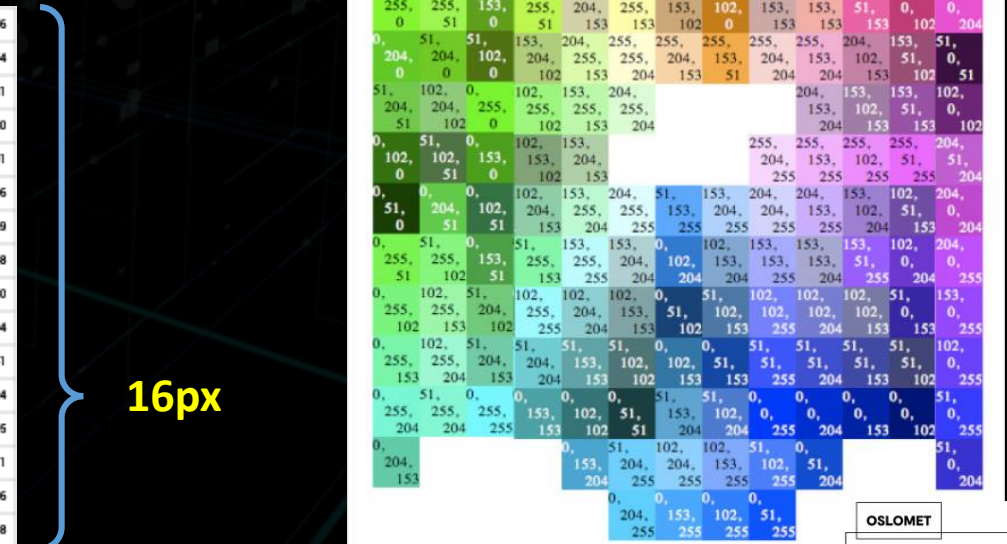
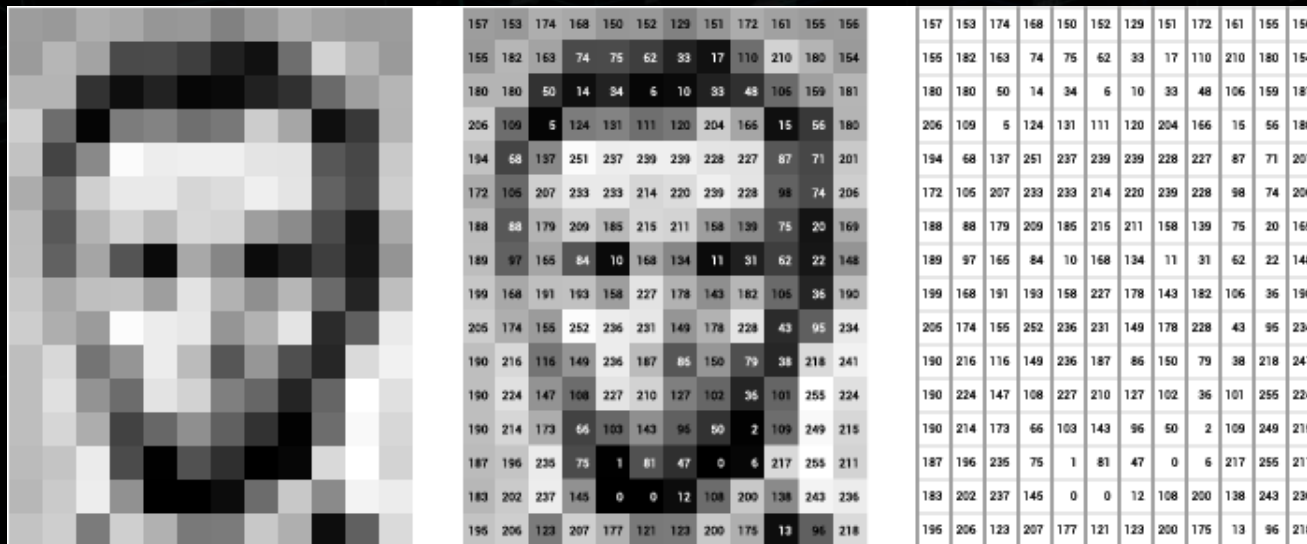
Computer Vision

Color intensity	DEC	HEX	BIN
	0	0x00	00000000
	16	0x10	00010000
	32	0x20	00100000
	48	0x30	00110000
	64	0x40	01000000
	80	0x50	01010000
	96	0x60	01100000
	112	0x70	01110000
	128	0x80	10000000
	144	0x90	10010000
	160	0xA0	10100000
	176	0xB0	10110000
	192	0xC0	11000000
	208	0xD0	11010000
	224	0xE0	11100000
	255	0xFF	11111111



$$255 \times 255 \times 255 = 16581375$$

R G B

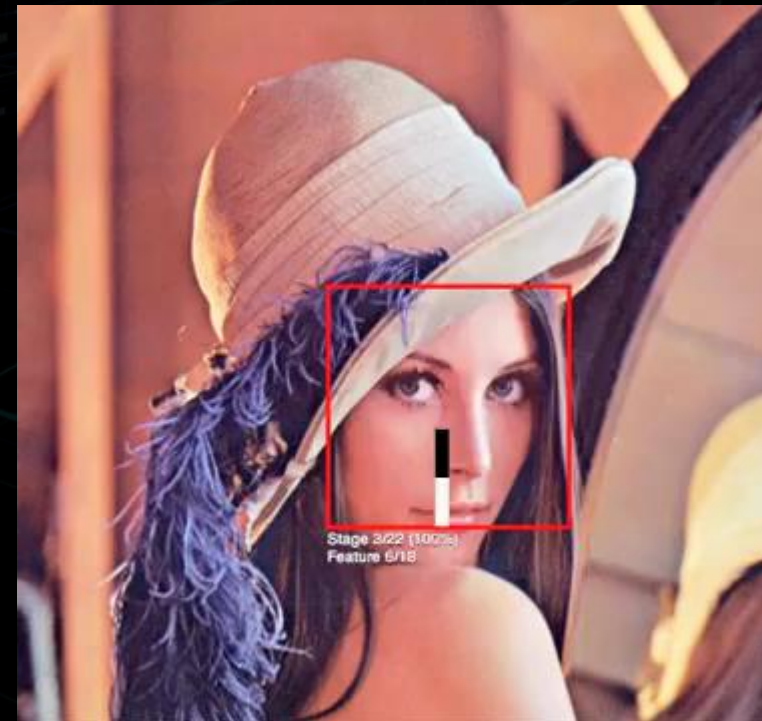
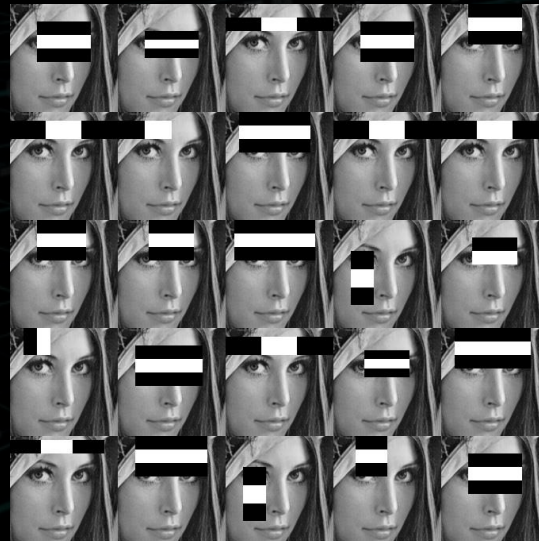
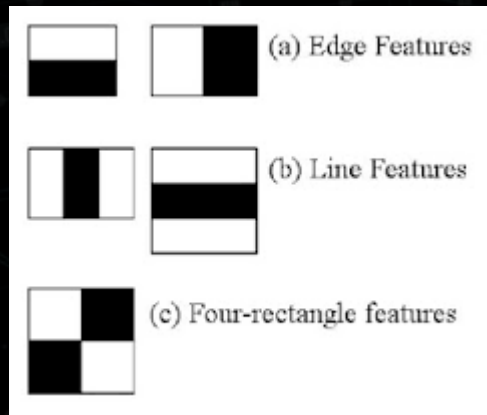


OSLOMET

MAKERSPACE

Viola-Jones Haar-Feature

- # algorithm used to identify objects in an image or video
- # proposed by Paul Viola and Michael Jones in 2001
- # powerful face/non-face classifiers can be constructed based on these features
- # they can be computed efficiently using the summed-area table or integral image technique.



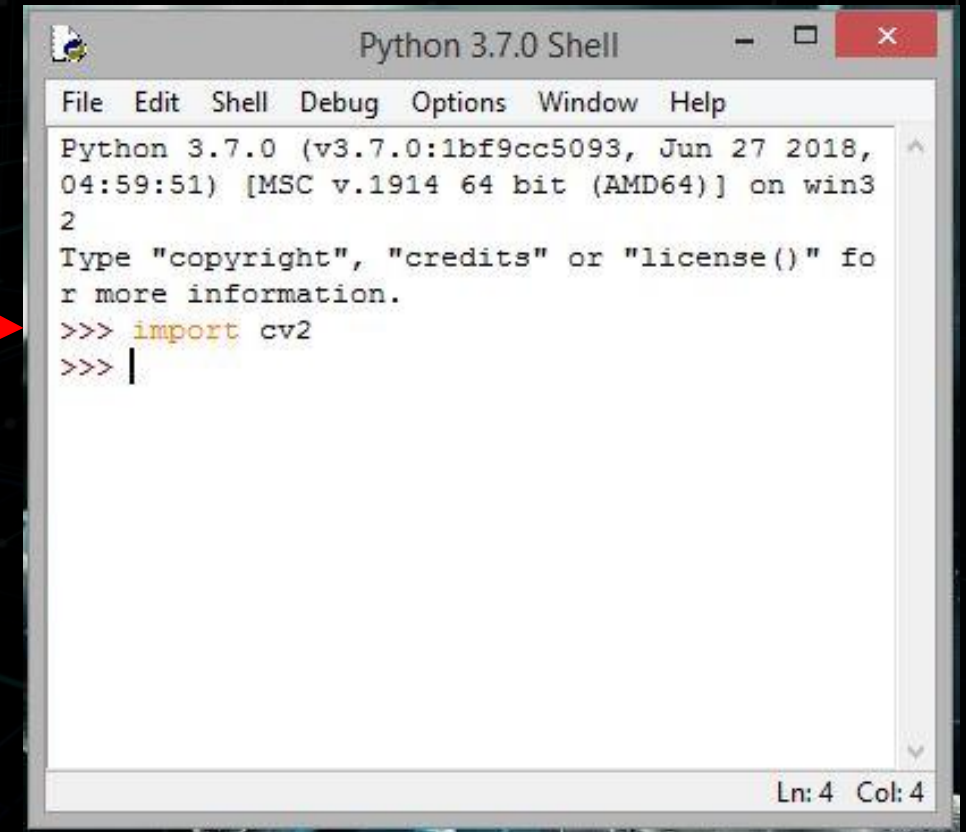
Installing OpenCV

1. Open “**Command Prompt**” in Windows or “**Terminal**” in Mac OS.
2. Enter the following command:

pip install opencv-python (for Python version 2)

-OR

pip3 install opencv-python (for Python version 3)



The screenshot shows a window titled "Python 3.7.0 Shell" with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The text inside the window reads: "Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32", "Type 'copyright', 'credits' or 'license()' for more information.", and the command prompt shows ">>> import cv2" followed by a new line ">>> |". The status bar at the bottom right indicates "Ln: 4 Col: 4".

OpenCV classifier

Location >>> C:\Program Files\Python37\Lib\site-packages\cv2\data

haarcascade_eye_tree_eyeglasses.xml
haarcascade_mcs_leftear.xml haarcascade_eye.xml
haarcascade_mcs_lefteye.xml
haarcascade_frontalface_alt2.xml
haarcascade_mcs_mouth.xml
haarcascade_frontalface_alt_tree.xml
haarcascade_mcs_nose.xml
haarcascade_frontalface_alt.xml
haarcascade_mcs_rightear.xml
haarcascade_frontalface_default.xml
haarcascade_mcs_righteye.xml haarcascade_fullbody.xml
haarcascade_mcs_upperbody.xml

haarcascade_lefteye_2splits.xml
haarcascade_profileface.xml haarcascade_lowerbody.xml
haarcascade_righteye_2splits.xml
haarcascade_mcs_eyepair_big.xml haarcascade_smile.xml
haarcascade_mcs_eyepair_small.xml
haarcascade_upperbody.xml

Face detection

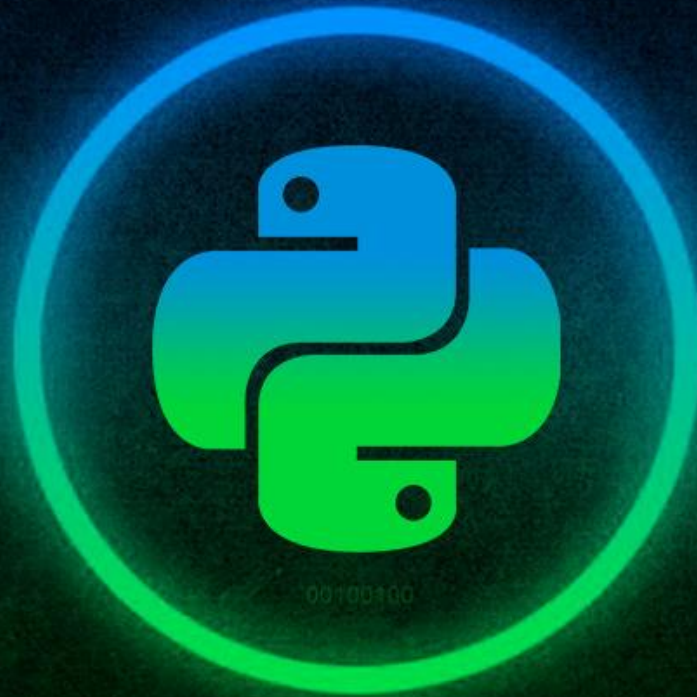
- # Import OpenCv
- # Read the image file
- # Resize picture if required
- # Convert image into grayscale
- # Initialize haar cascade feature for frontalface
- # Detect faces on image
- # Draw the rectangle around the image
- # Display the output image with faces detected

Face tracking

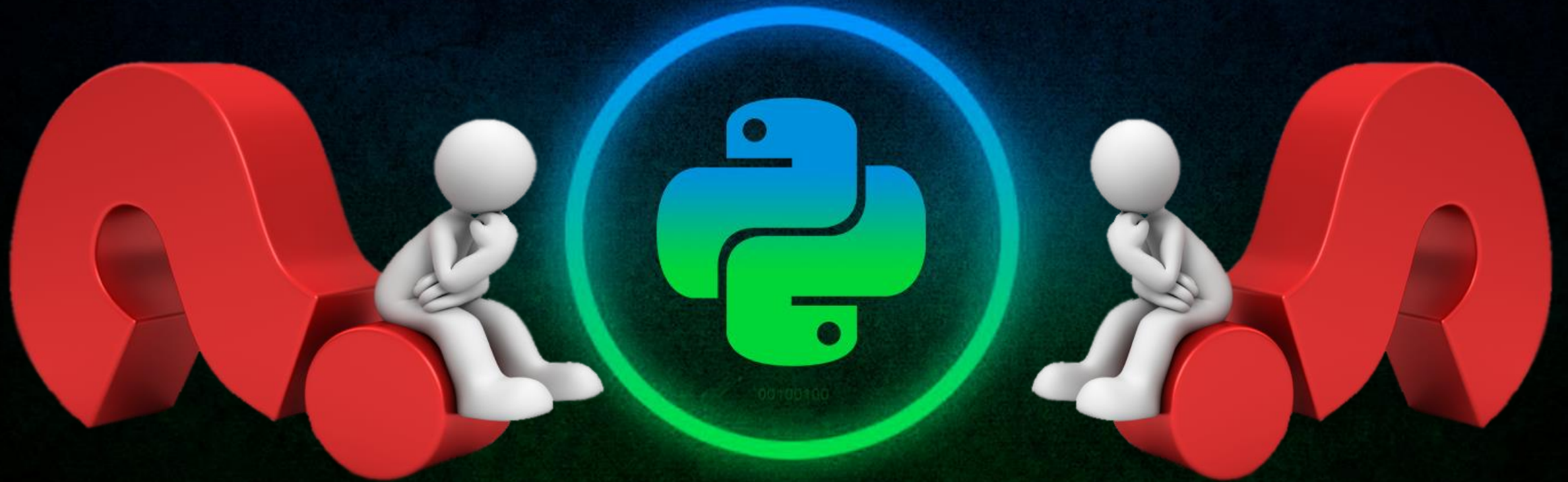
- # Import OpenCv
- # Initialize webcam
- # Initialize haar cascade feature for frontalface
- # Create a loop to continuously read the video
 - # Resize or flip the video if required
 - # Convert the video into grayscale and also blur it
 - # Detect faces in the video
 - # Draw the rectangle around the faces in the video
 - # Display the video with the rectangle
- # Release the webcam when you close the program

Template matching

- # Import OpenCv
- # Initialize webcam
- # Open a template image and get its width and height
- # Create a loop to continuously read the video
 - # Resize or flip the video if required
 - # Convert the video into grayscale and also blur it
 - # Match image template with the blurred video
 - # Find the area where the template is matched for maximum time
 - # Draw the rectangle around the area in the video, where the template is matching
 - # Display the video with the rectangle
 - # Release the webcam when you close the program



```
print ("Lets start coding")
```

```
print ("Any Questions?")
```

```
print("Provide us quick feedback")
```



Link to feedback form:

https://docs.google.com/forms/d/e/1FAIpQLSf6B4QaFleIVZ-uv4zo796EXN5_cBUAd9yh90a_tsiC3paO7A/viewform



```
print ("Thank You")
```