Chapter: 4

Image Processing

Image Processing

- Image acquisition
- Image Representation
- Image Enhancement
- Image Compression
- Object Recognition

Image Acquisition

Camera + Scanner ->Digital Camera: Get images into computer

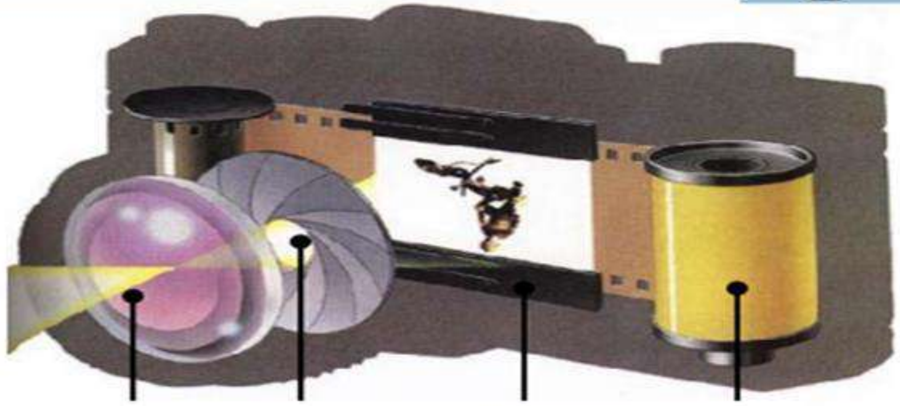


Image Representation

Discrete representation of images

- **We'll carve up image into a rectangular grid of pixels P[x,y]**
- **Each pixel p will store an intensity value in [0 1]**
- **0 ->black; 1 ->white; in-between ->gray**
- **Image size mxn ->(mn) pixels**

Image Enhancement

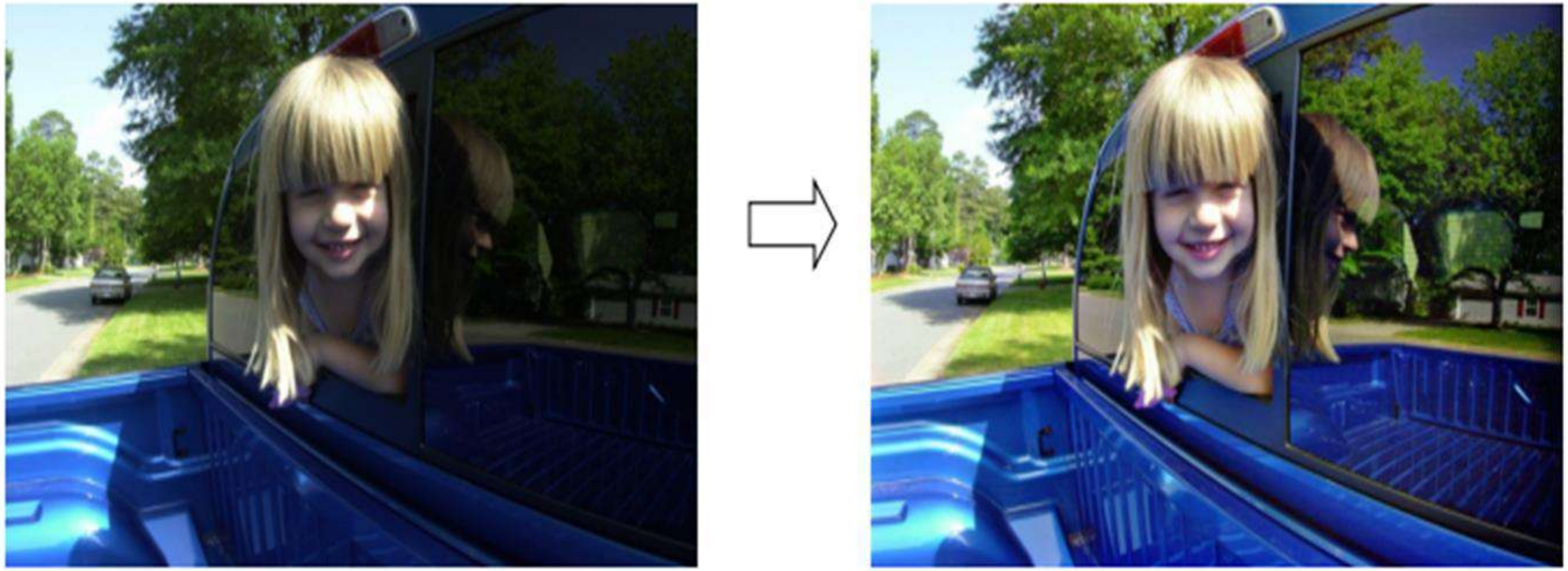
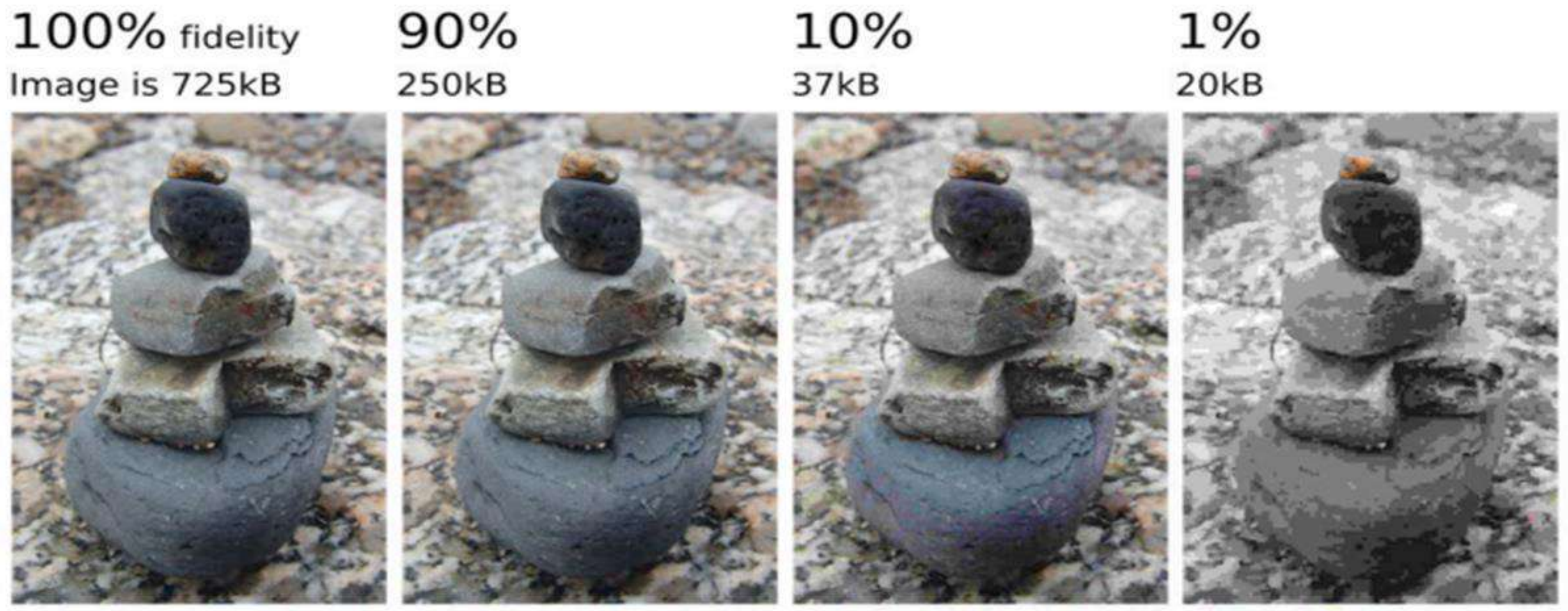
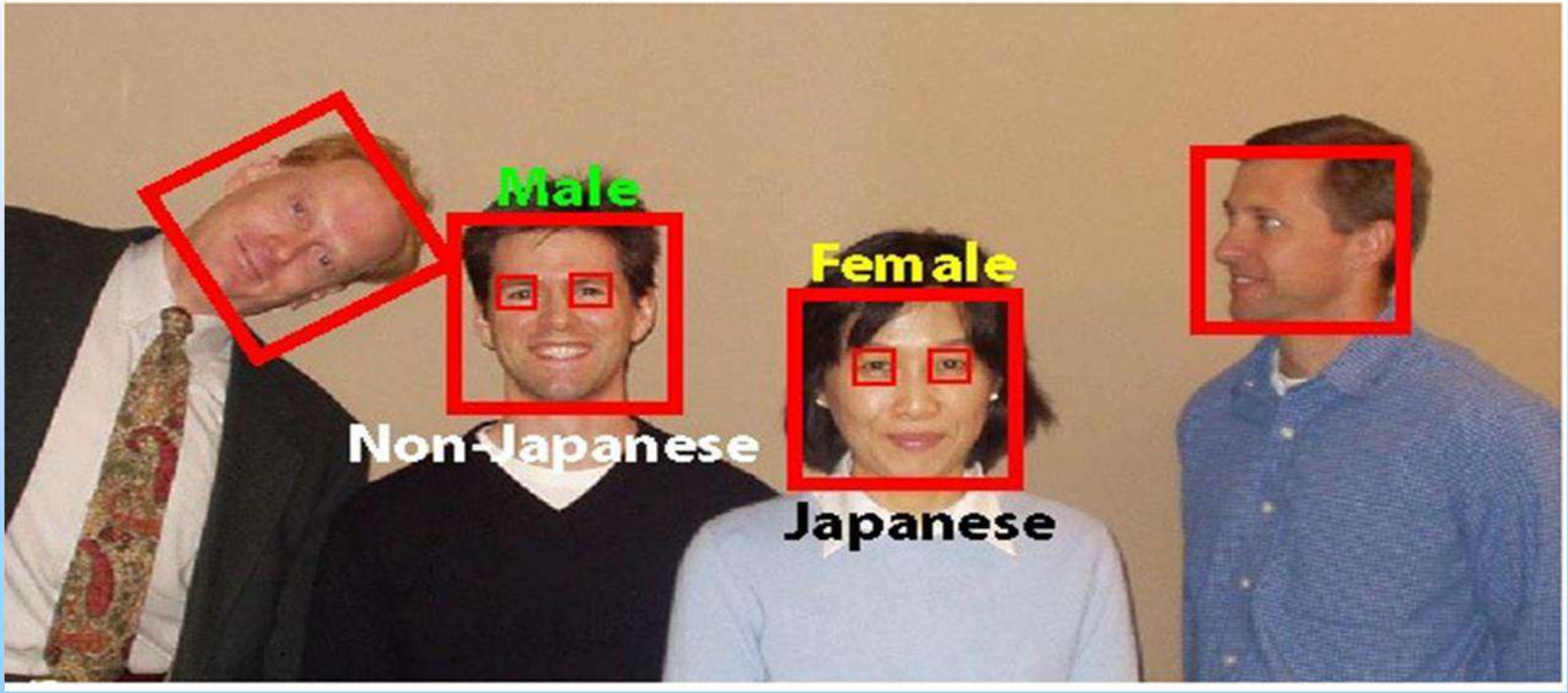


Image Compression



Object Detection / Recognition



Python & Image Processing

Python provides lots of libraries for image processing, including:

- OpenCV:

Image processing library mainly focused on computer vision with application in the features of 2D and 3D images , facial recognition, Human-computer interaction, Mobile robotics, Object identification and others.

Python & Image Processing

- **Numpy and Scipy libraries:**

  For image manipulation and processing.

- **Sckikit:**

  Provides lots of algorithms for image processing.

- **Python Imaging Library (PIL):**

  To perform basic operations on images like create thumbnails, resize, rotation, convert between different file formats etc.

Python & Image Processing

The Requirements to use Python in image processing

1. Python Download from the following link:

   https://www.python.org/downloads/

2. Pycharm Download from the following link:

   https://www.jetbrains.com/pycharm/download/#section=windows

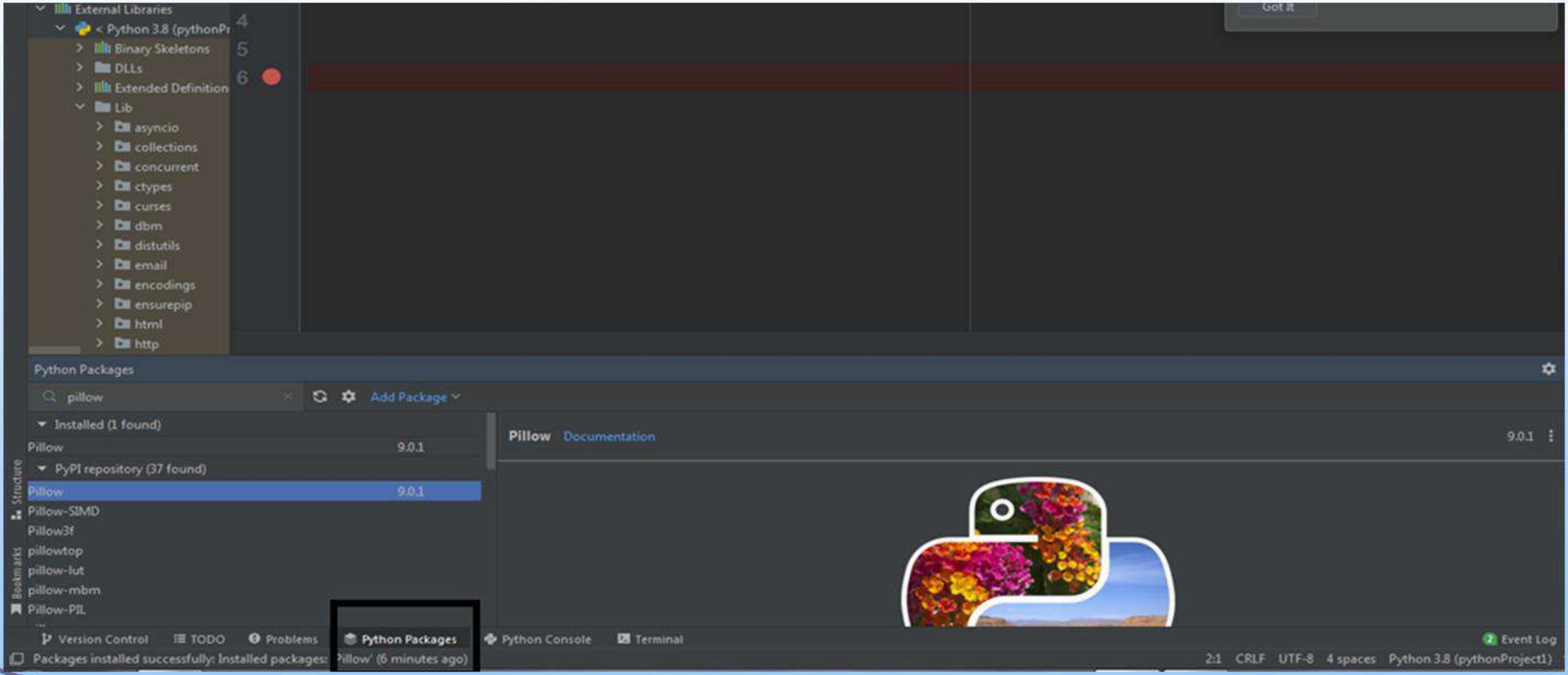
Python & Image Processing

3. Install required library:

Our first step will be to install the required library, like openCV, pillow or other which we wants to use for image processing. We can use pip to install the required library, like:−

- Install pip
- Install pillow

Python & Image Processing



```
Installed (1 found)
Pillow                9.0.1
PyPI repository (37 found)
Pillow                9.0.1
Pillow-SIMD
Pillow3f
pillowtop
pillow-lut
pillow-mbm
Pillow-PIL
```

Pillow 9.0.1

Packages installed successfully: Installed packages: 'Pillow' (6 minutes ago)

2:1 CRLF UTF-8 4 spaces Python 3.8 (pythonProject1)

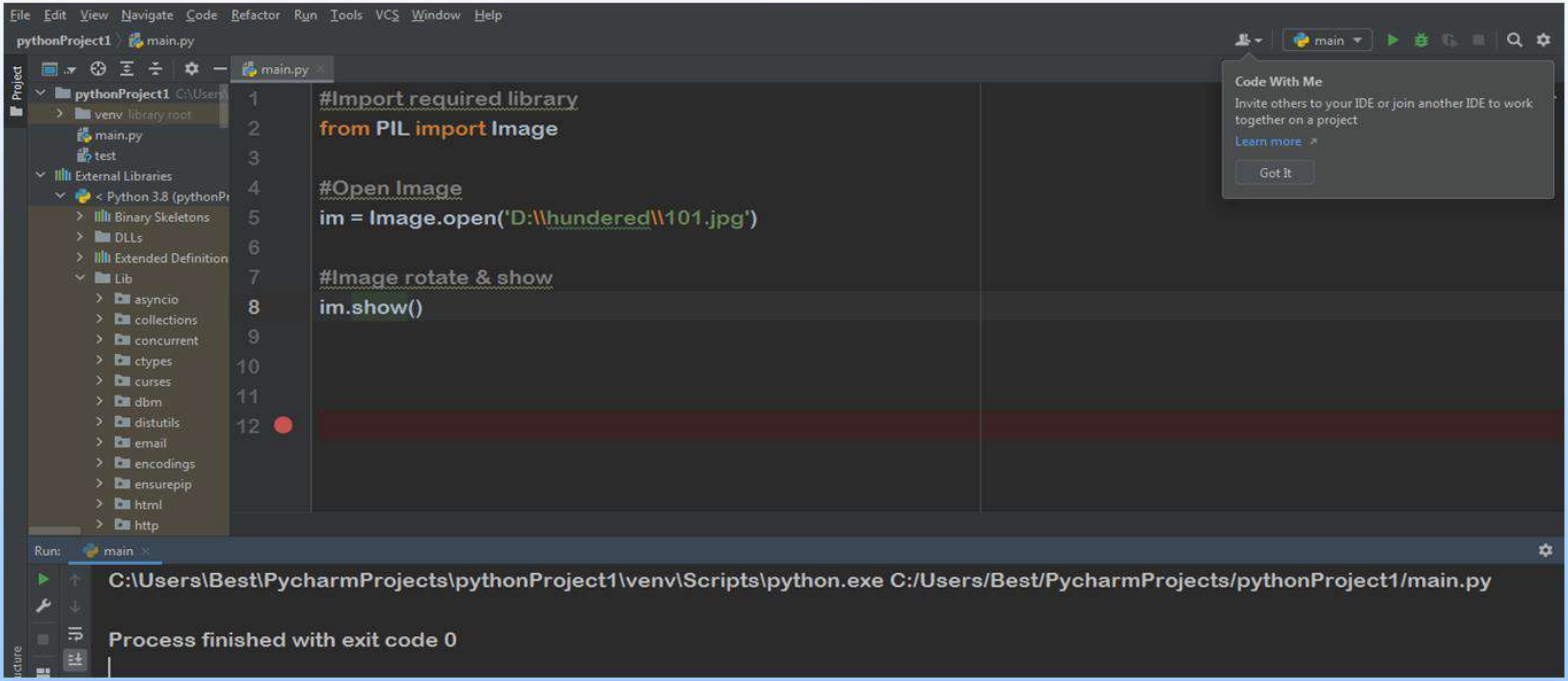
Python & Image Processing

- Image: Open() and show()

First, open the file/image and show. You can rotate the image while showing like below:

```
#Import required library
from PIL import Image

#Open Image
im = Image.open(“xxx.jpg")

#Image rotate & show
im.rotate(45).show()
```

Python & Image Processing



```python
#Import required library
from PIL import Image

#Open Image
im = Image.open('D:\\hundered\\101.jpg')

#Image rotate & show
im.show()
```

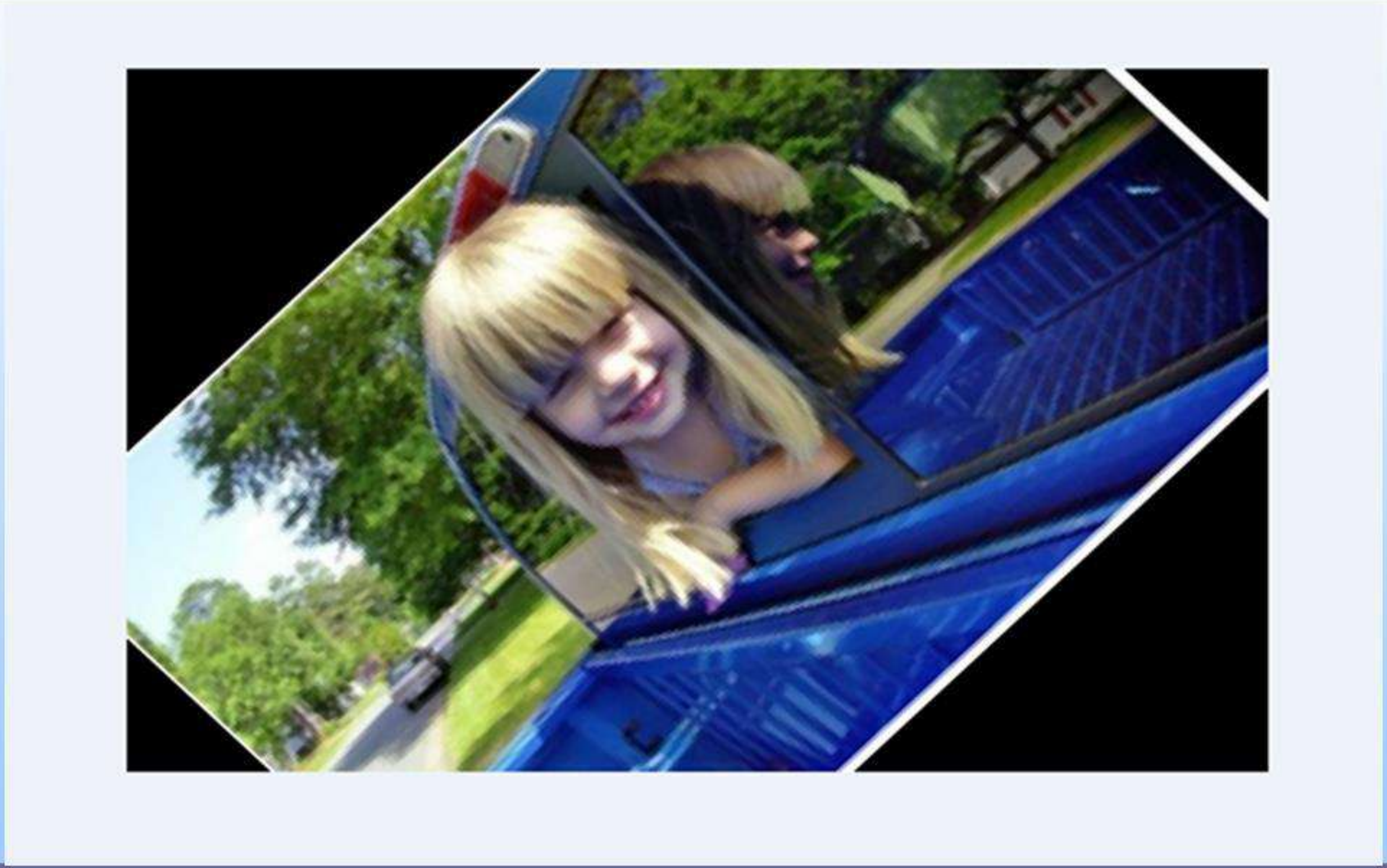
```
C:\Users\Best\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/Best/PycharmProjects/pythonProject1/main.py

Process finished with exit code 0
```

Python & Image Processing

Output

Python & Image Processing

Output



Python & Image Processing

- Image.size

It returns the tuple consist of height & weight of the image

```
>>> im.size()
(1000, 667)
```

Python & Image Processing

- Image.format

This function returns file format of the image file like ‘JPEG’, ‘BMP’, ‘PNG’, etc.

```
>>> im.format
'JPEG'
```

Python & Image Processing

- Image.width

  It returns only the width of the image.

```
>>> im.width
1280
```

- Image.height

  It returns only the height of the image.

```
>>> im.height
721
```

Python & Image Processing

- Image.info

It returns a dictionary holding data associated with the image

```
>>>im.info
{'jfif': 257, 'jfif_version': (1, 1), 'dpi': (300, 300), 'jfif_unit': 1,
'jfif_density': (300, 300), 'exif': b"Exif\x00\x00MM\x00*\x00\x00\x00
....
....
\xeb\x00\x00'\x10\x00\x00\xd7\xb3\x00\x00\x03\xe8"}
```

Python & Image Processing

- **Convert and Save() Image**

  We can change the format of image from one form to another, like below:

```
>>> im.save('TajMahal.png')
```

- **Resize-thumbnails()**

  We can change the size of image using thumbnail() method of pillow:

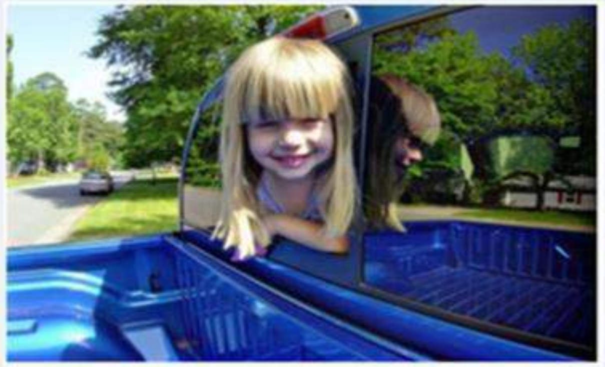
```
>>> im.thumbnail ((300, 300))
>>> im.show()
```

Python & Image Processing

```
#Import required library
from PIL import Image

#Open Image
im = Image.open('D:\\hundered\\101.jpg')

#Resize-thumbnails()
im.thumbnail ((300, 300))
im.show()
```

Python & Image Processing

- The image will change as follows:



Python & Image Processing

- Converting to grayscale image – convert()

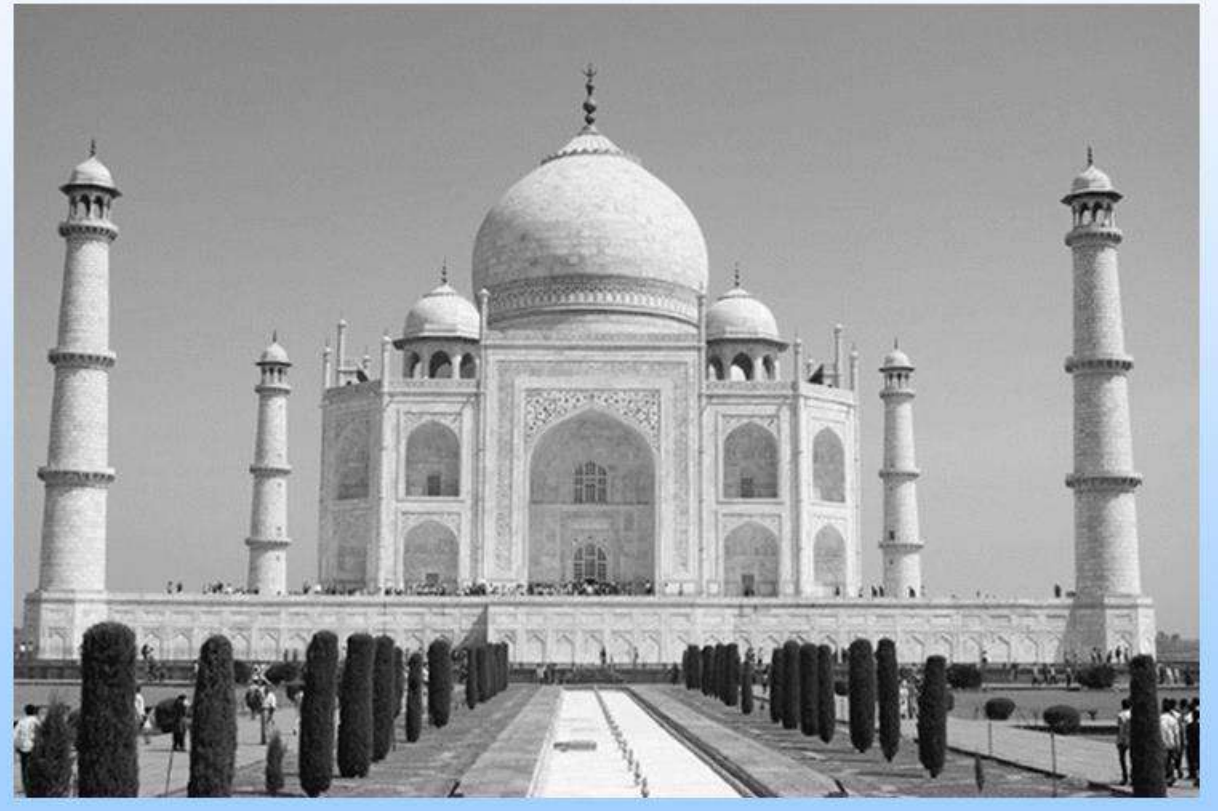
  We can make the grayscale image from our original colored image.

```
>>> im = Image.open('TajMahal.jpg').convert('L')
>>> im.show()
```

Where "L" stands for 'luminous'.

Python & Image Processing

Output



Python & Image Processing

- Image.filename

This function is used to get the file name or the path of the image.

```
>>>im = Image.open('beach1.jpg')
>>> im.filename
'beach1.jpg'
```

Python & Image Processing

- Image.mode

  It is used to get the pixel format used by the image. Typical values are “1”, “L”, “RGB” or “CMYK”.

```
>>> image.mode
'RGB'
```

Python & Image Processing

- Merging two images

In the same way, to merge two different images, you need to:

- Create image object for the required images using the open() function.
- While merging two images, you need to make sure that both images are of same size. Therefore, get each sizes of both images and if required, resize them accordingly.
- Create an empty image using the Image.new() function.
- Paste the images using the paste() function.
- Save and display the resultant image using the save() and show() functions

Python & Image Processing

```
from PIL import Image
#Read the two images
im1=Image.open('D:\\hundered\\10.jpg')
im1.show();
im2=Image.open('D:\\hundered\\11.jpg')
im2.show()
#resize, first image
im1 = im1.resize((250,250))
im2 = im2.resize((250,250))
```

Python & Image Processing

```
im1size = im1.size
im2size = im2.size
newimage = Image.new('RGB',(2*im1size[0],im1size[1]))
newimage.paste(im1,(0,0))
newimage.paste(im2,(im1size[0],0))
newimage.save('D:\\hundered\\newimage.jpg')
newimage.show()
```

Python & Image Processing

- Input im1

Python & Image Processing

Input im2

Python & Image Processing

- Merged image

Python & Image Processing

- Blur an Image

There are various techniques used to blur images and we are going to discuss the below mentioned techniques.

- Simple blur
- Gaussian blur

Python & Image Processing

- Simple blur

Syntax

```
filter(ImageFilter.BLUR)
```

Python & Image Processing

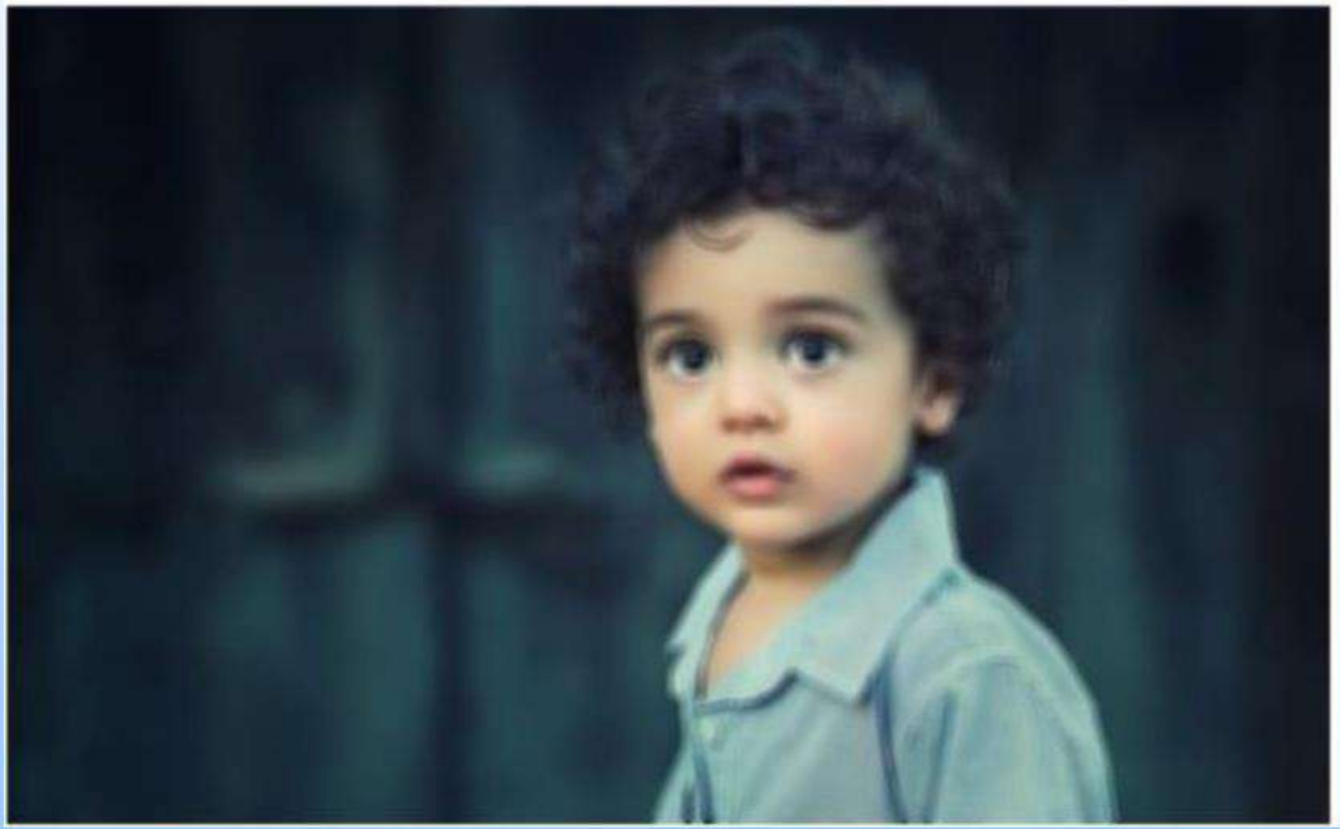
```
#Import required Image library
from PIL import Image, ImageFilter
#Open existing image
im1 = Image.open('D:\\hundered\\10.jpg')
im1.show()
blurIm1 = im1.filter(ImageFilter.BLUR)
blurIm1.show()
#Save blurImage
blurIm1.save('D:\\hundered\\blur10.jpg')
```

Python & Image Processing

Original image(10.jpg)

Python & Image Processing

- Blurred image(blur10.jpg)



Python & Image Processing

- Gaussian Blur

<u>Syntax</u>

```
ImageFilter.GaussianBlur(radius=2)
```

Where:

Radius = Blur radius

Python & Image Processing

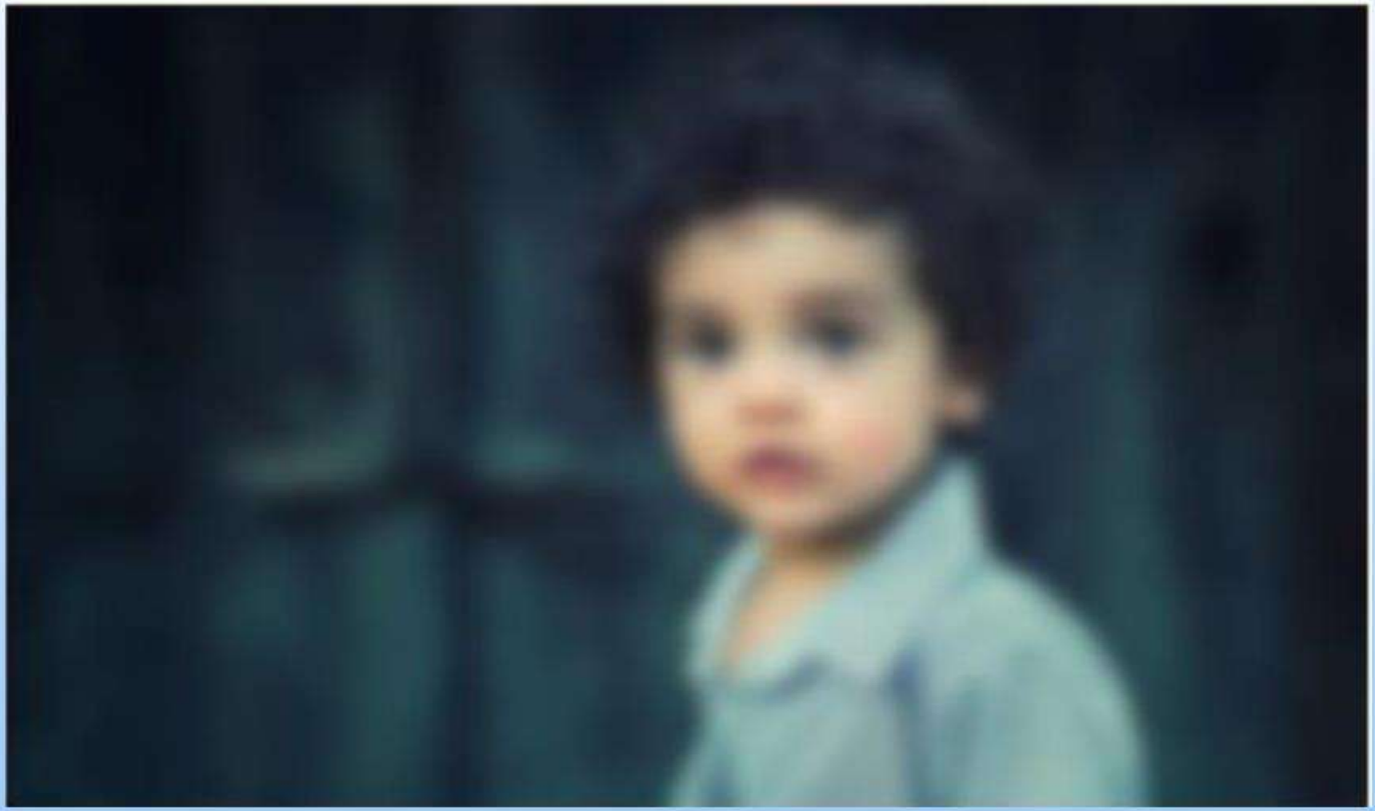
```
#Import required Image library
from PIL import Image, ImageFilter
#Open existing image
im1 = Image.open("D:\\hundered\\10.jpg")
im1.show()
#Applying GaussianBlur filter
gaussIm1 = im1.filter(ImageFilter.GaussianBlur(5))
gaussIm1.show()
#Save Gaussian Blur Image
gaussIm1.save(‘'D:\\hundered\\gauss10.jpg")
```

Python & Image Processing

Original image (10.jpg)

Python & Image Processing

- Blurred image(gauss10.jpg)



Python & Image Processing

- ImageDraw Module

1. <u>Line</u>

Following is, the syntax to draw a line using python pillow:

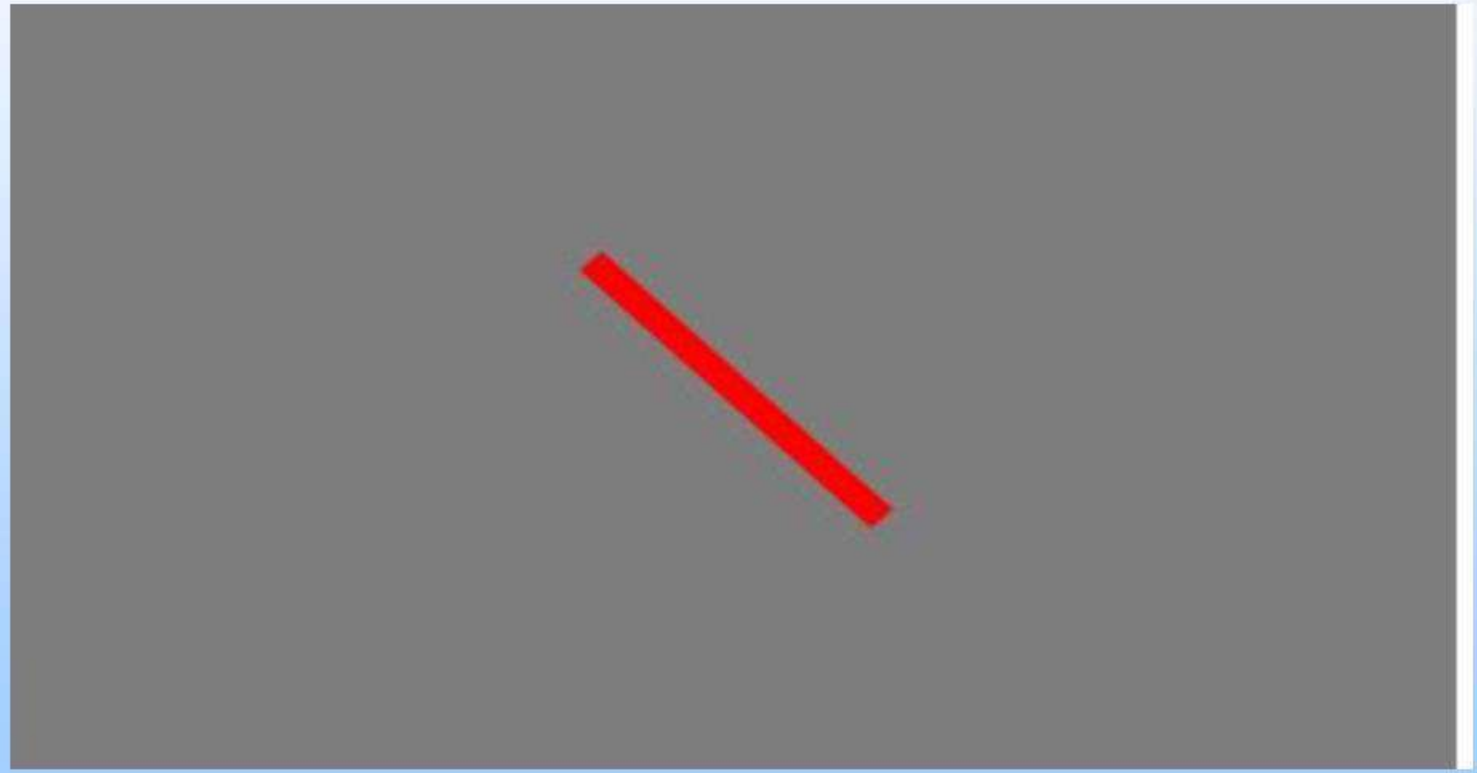
```
draw.line(xy, fill=None, width=0)
```

Python & Image Processing

```
from PIL import Image, ImageDraw
im1 = Image.new('RGB', (500, 300), (125, 125, 125))
draw = ImageDraw.Draw(im1)
draw.line((200, 100, 300, 200), fill=(0, 0, 0), width=10)
im1.show()
```

Python & Image Processing

Output



Python & Image Processing

- Ellipse

Following is, the syntax to draw an ellipse using python pillow:

```
draw.ellipse(xy, fill=None, outline=None)
```

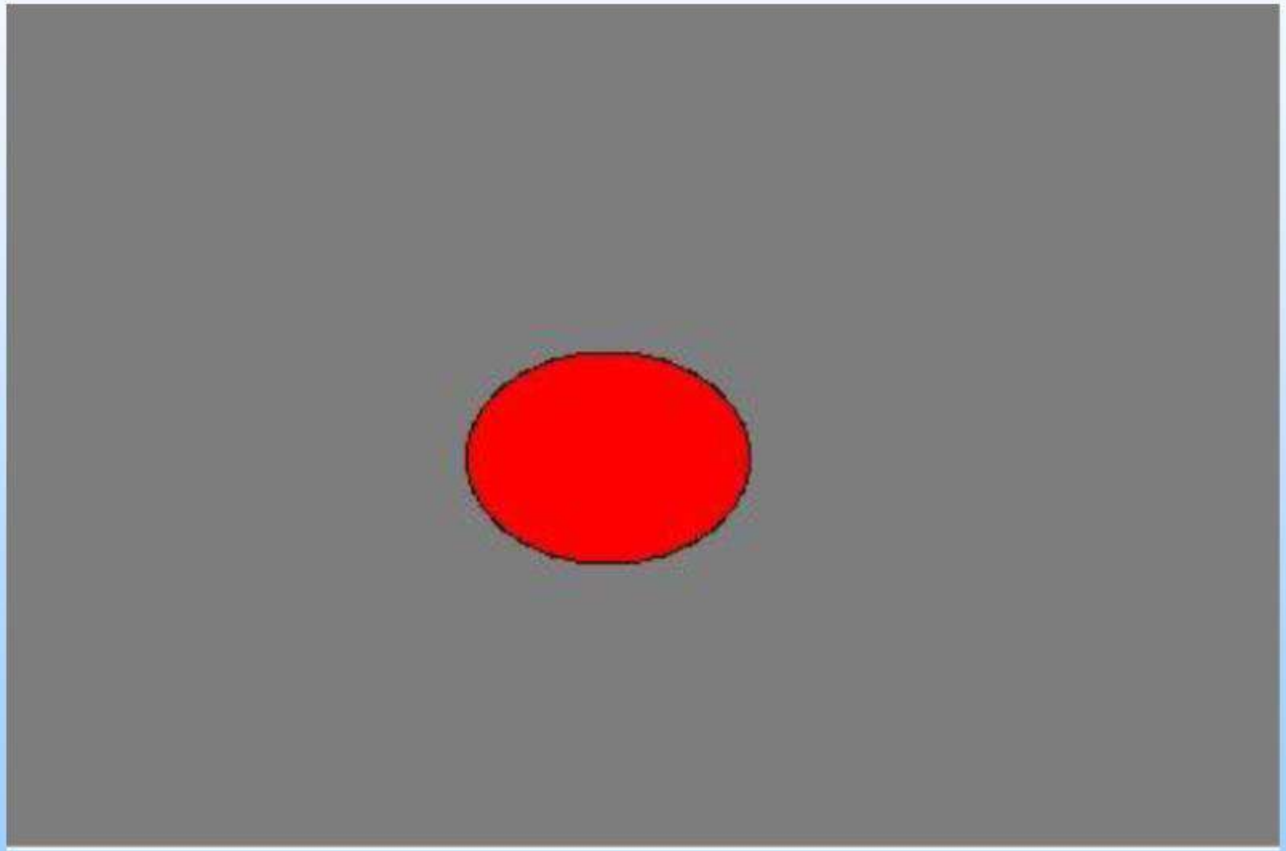
The ellipse() method draws the ellipse surrounded by bounding box xy on draw.

Python & Image Processing

```
from PIL import Image, ImageDraw
im1 = Image.new('RGB', (500, 300), (125, 125, 125))
draw = ImageDraw.Draw(im1)
draw.ellipse((200, 125, 300, 200), fill=(255, 0, 0),
outline=(0, 0, 0))
im1.show()
```

Python & Image Processing

Output



Python & Image Processing

- Rectangle

Following is, the syntax to draw a rectangle using python pillow:

```
draw.rectangle(xy, fill=None, outline=None)
```

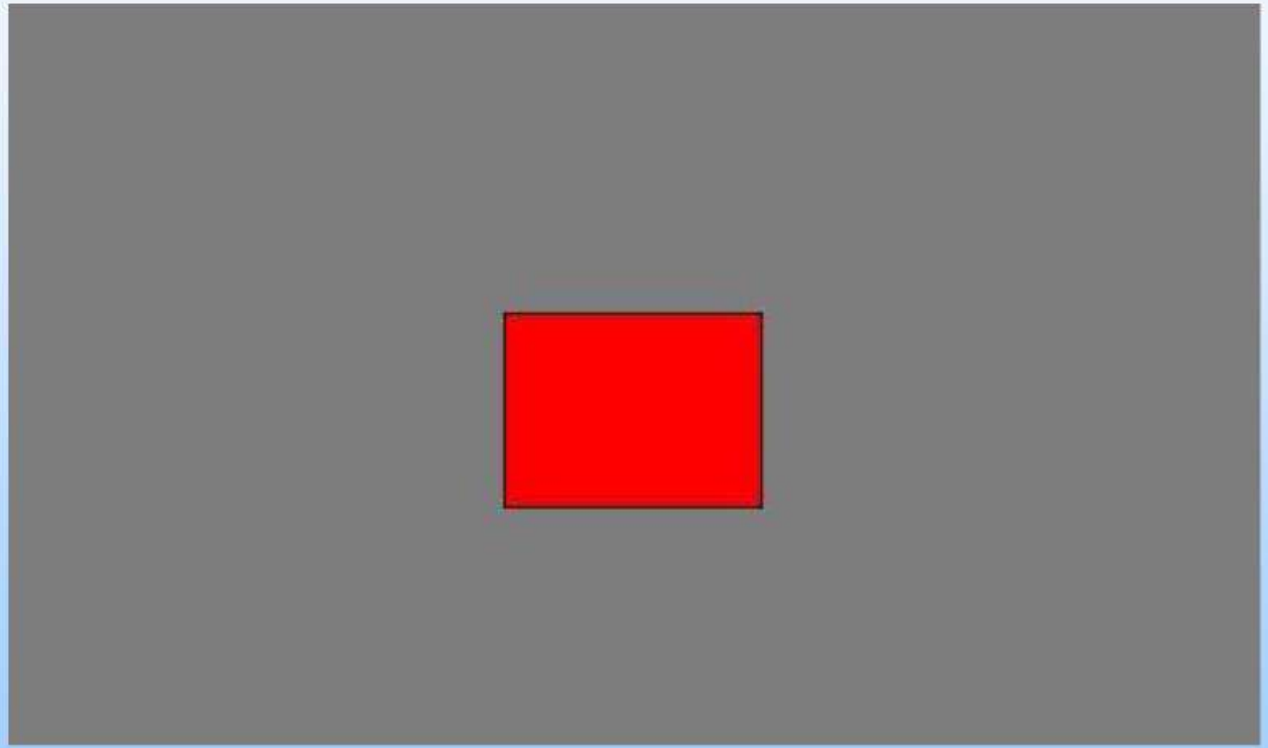
The rectangle() method draws the rectangle given bounding box xy on draw

Python & Image Processing

```
from PIL import Image, ImageDraw
im1 = Image.new('RGB', (500, 300), (125, 125, 125))
draw = ImageDraw.Draw(img)
draw.rectangle((200, 125, 300, 200), fill=(255, 0, 0), outline=(0, 0, 0))
im1.show()
```

Python & Image Processing

Output



Python & Image Processing

- Polygon (Triangle)

Following is, the syntax to draw a rectangle using python pillow:

```
draw.polygon(seq, fill=None, outline=None)
```

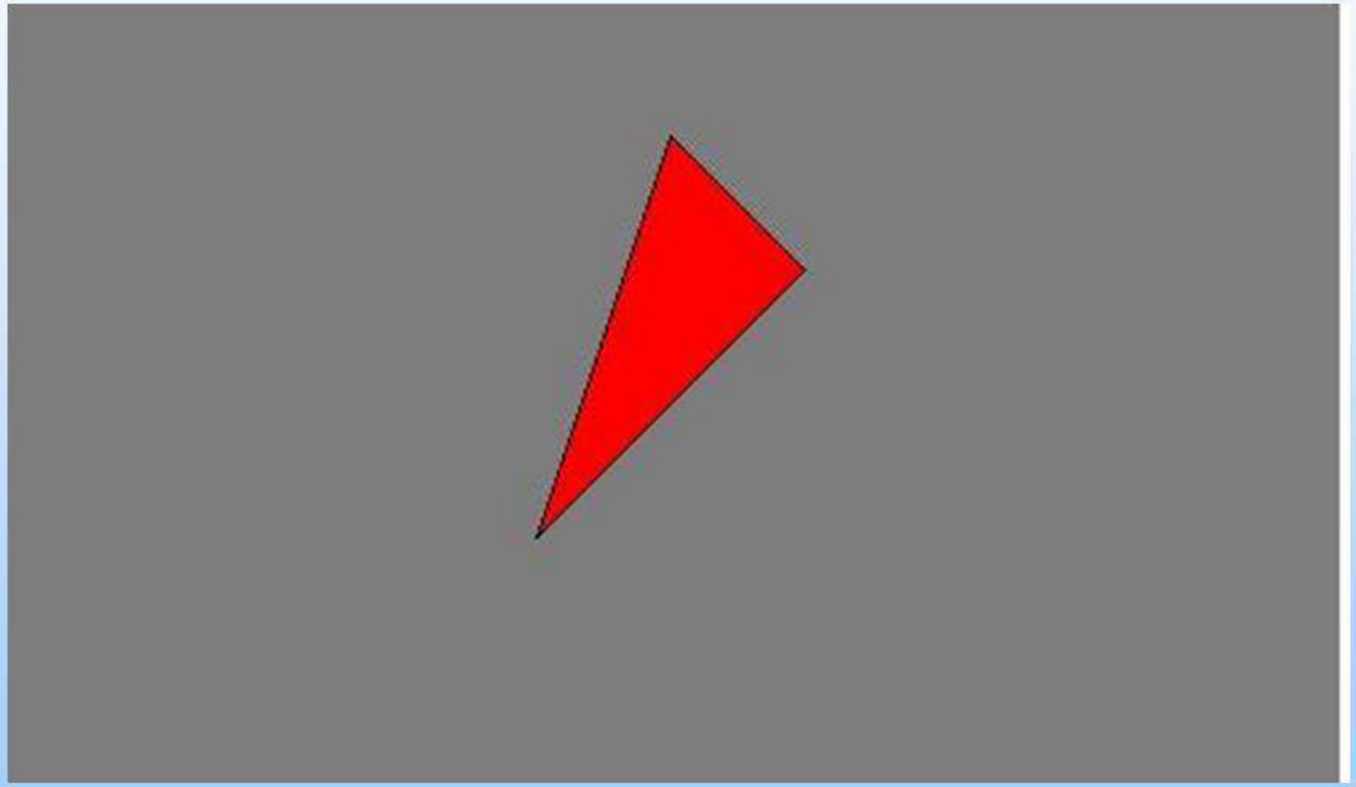
The polygon() method draws a polygon connecting with straight lines the co-ordinate sequence locations seq on draw.

Python & Image Processing

```
from PIL import Image, ImageDraw
img = Image.new('RGB', (500, 300), (125, 125, 125))
draw = ImageDraw.Draw(img)
draw.polygon( ((200, 200), (300, 100), (250, 50)),
fill=(255, 0, 0),
outline=(0, 0, 0))
img.show()
```

Python & Image Processing

Output



Python & Image Processing

- Above example is from the PIL library of python. We can use other library like open-cv, matplotlib & numpy for image processing.
- The following example program to demonstrate the use of much powerful library for image processing

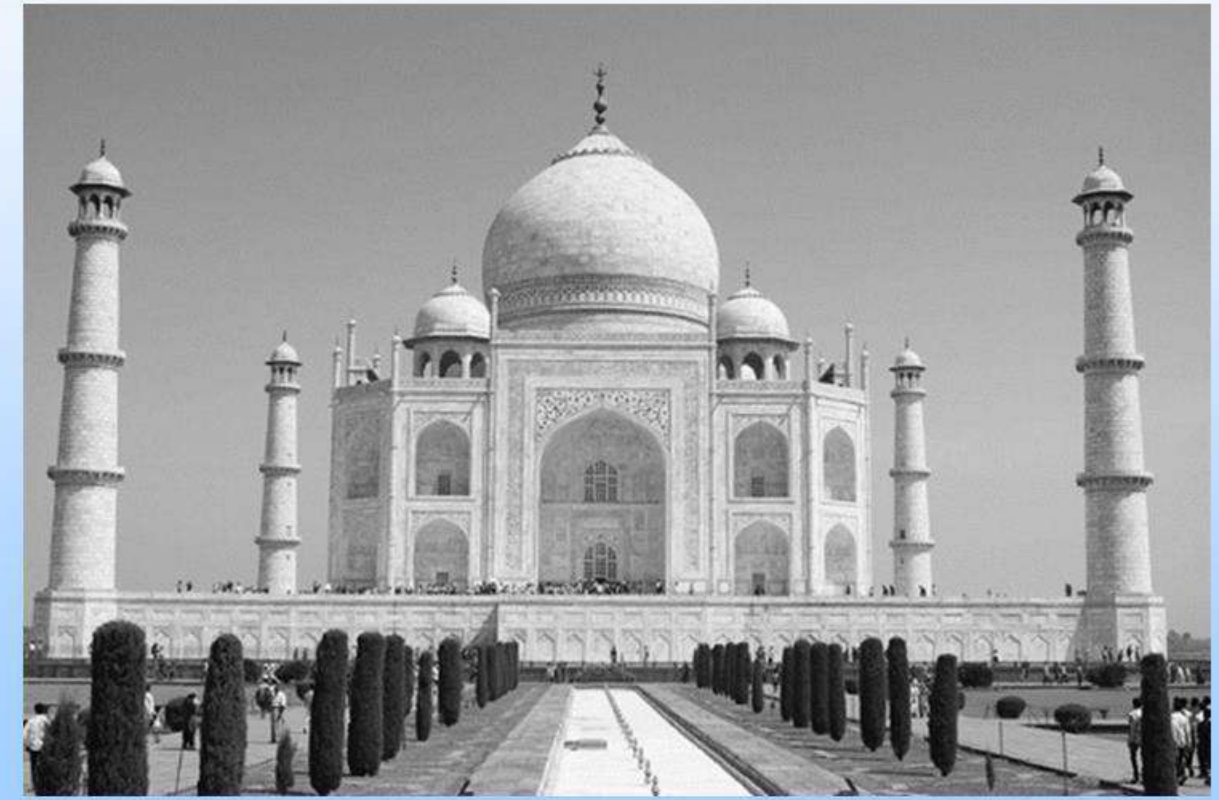
Python & Image Processing

- Showing image in grayscale:

```
#Import required library
import cv2
import numpy as np
from matplotlib import pyplot as plt

im = cv2.imread('TajMahal.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imshow('image',im)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

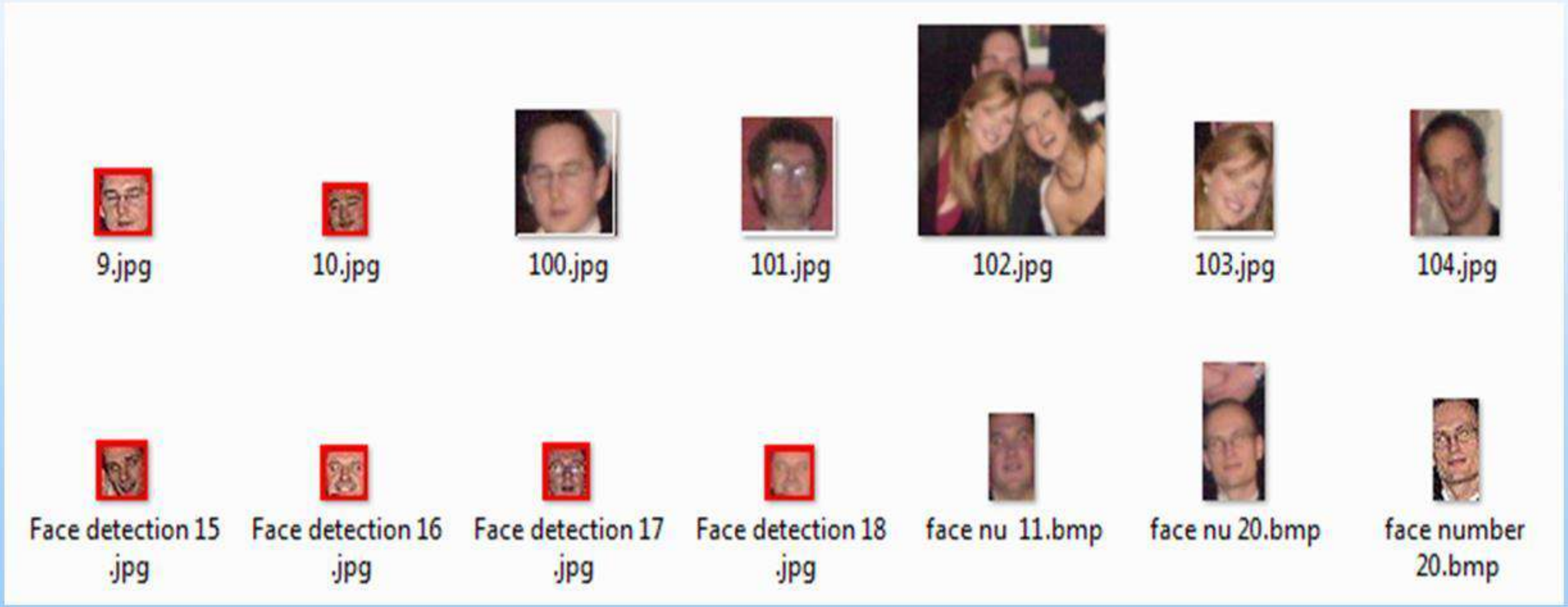
Python & Image Processing

Output

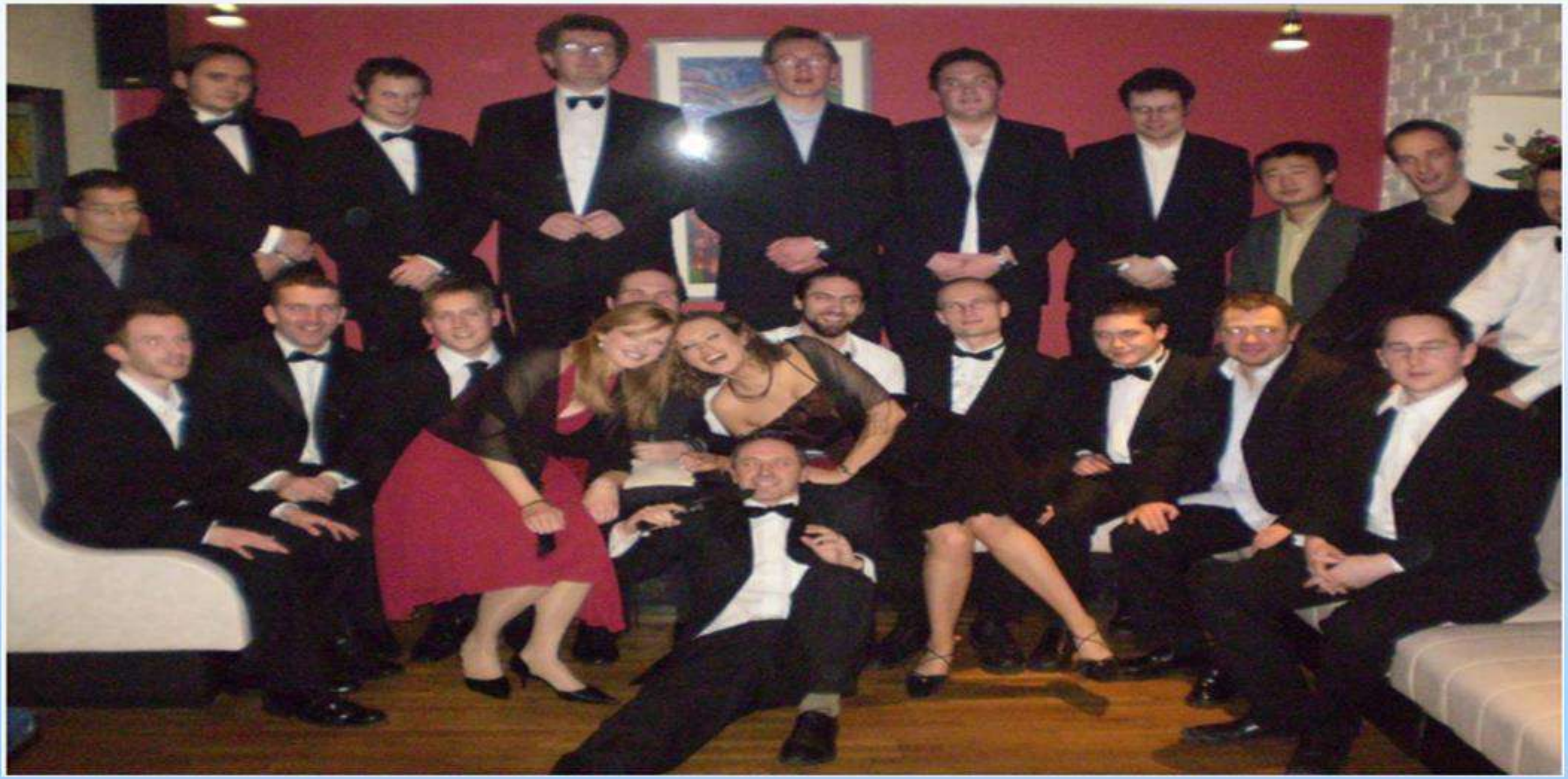


Exercuse:1

- **Consider the following Objects and image:**



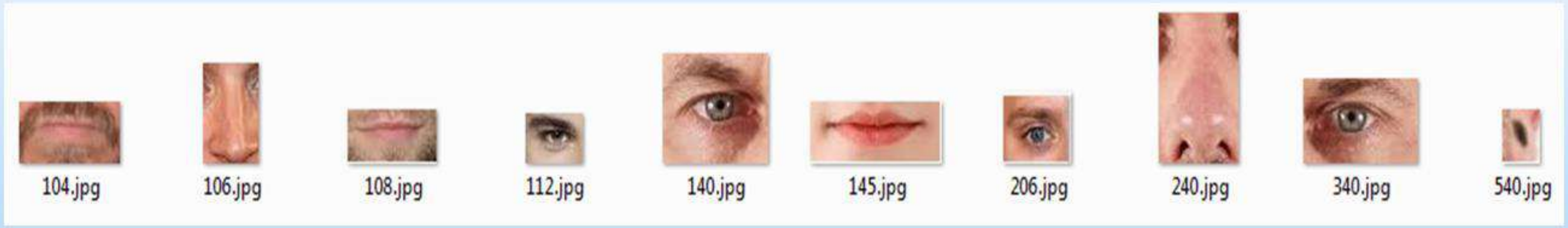
Exercuse:1

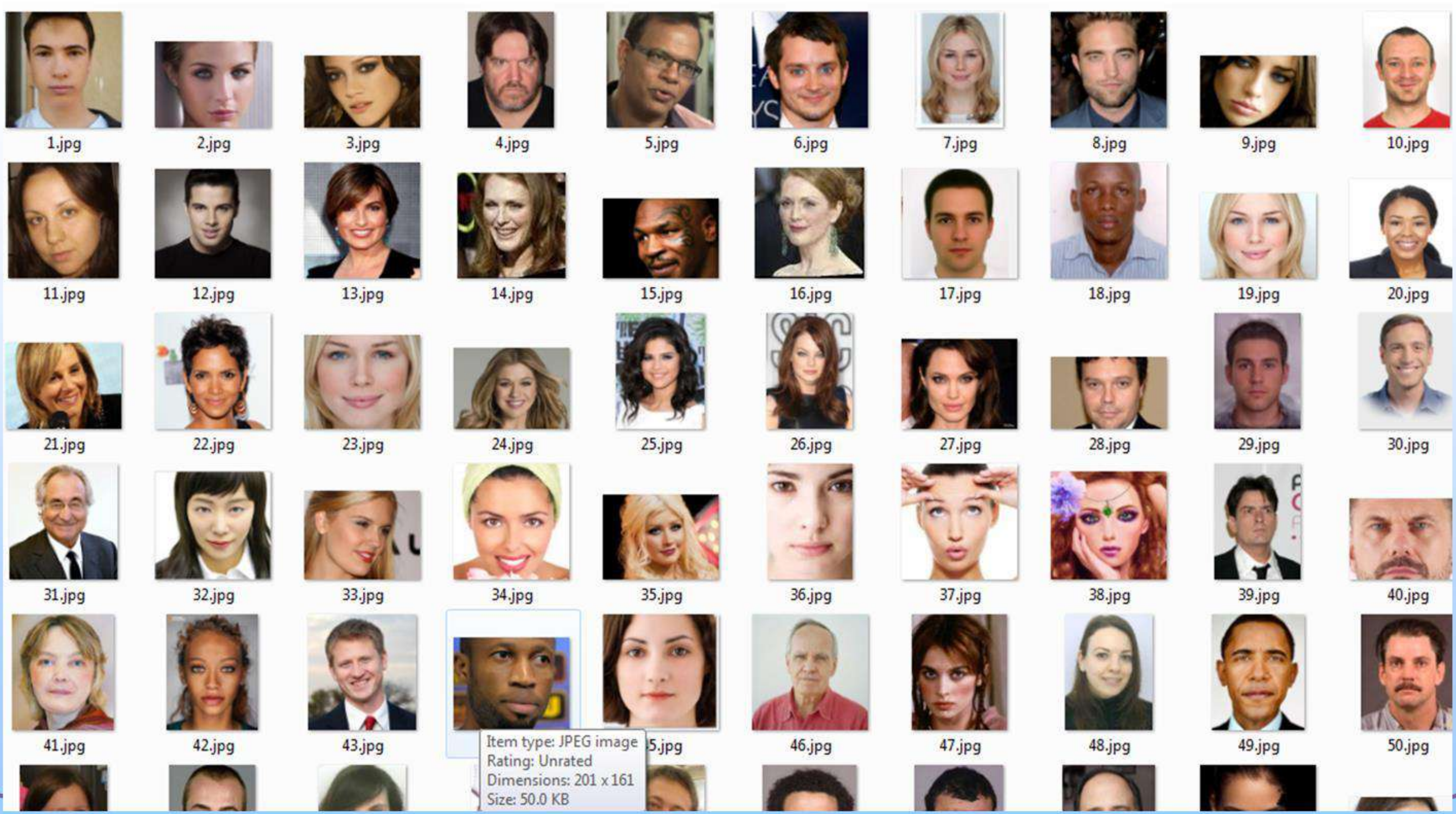


And write a program by Python to object recognition

Exercise:2

- Consider the following patterns and images:





Item type: JPEG image
Rating: Unrated
Dimensions: 201 x 161
Size: 50.0 KB

Exercise:2

- And write a program by Python to pattern recognition