SIC Batch 5

Week 6 - Introduction to Computer Vision

What Can Computer Vision do?

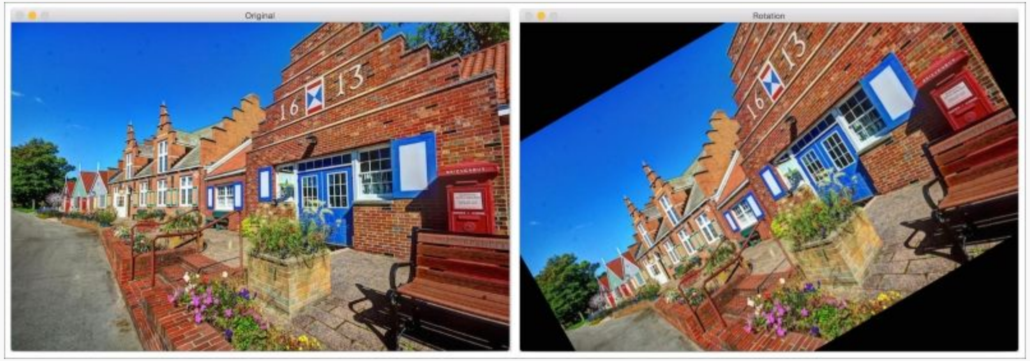
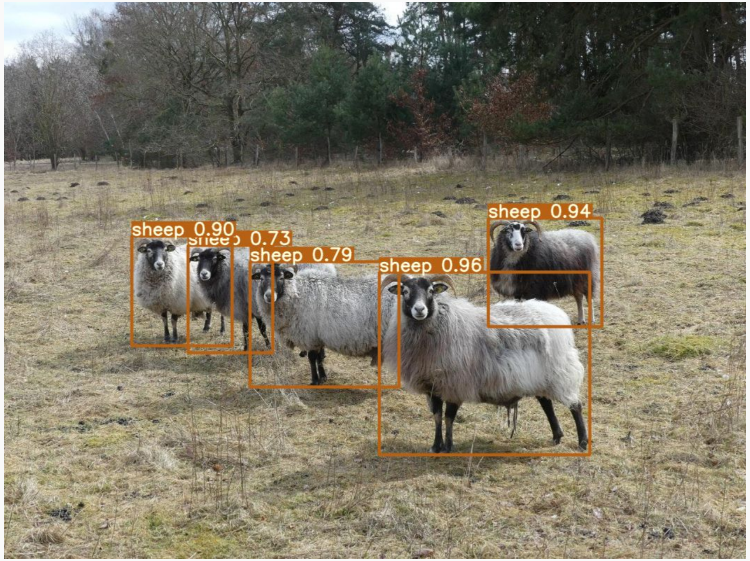


Image Processing



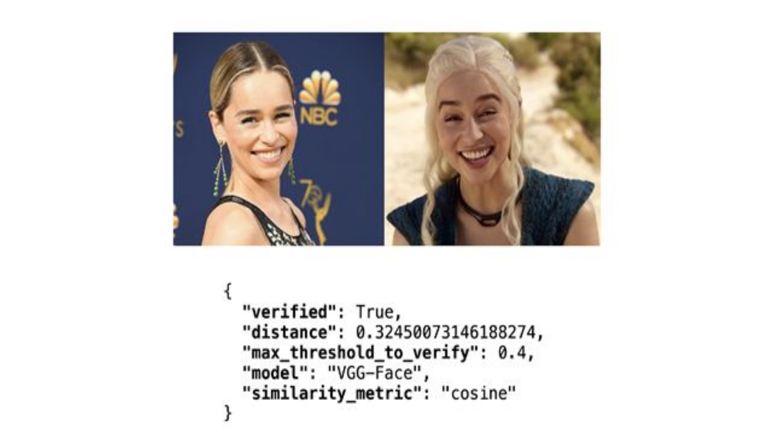
Object Detection

What Can Computer Vision do?

Gesture Recognition

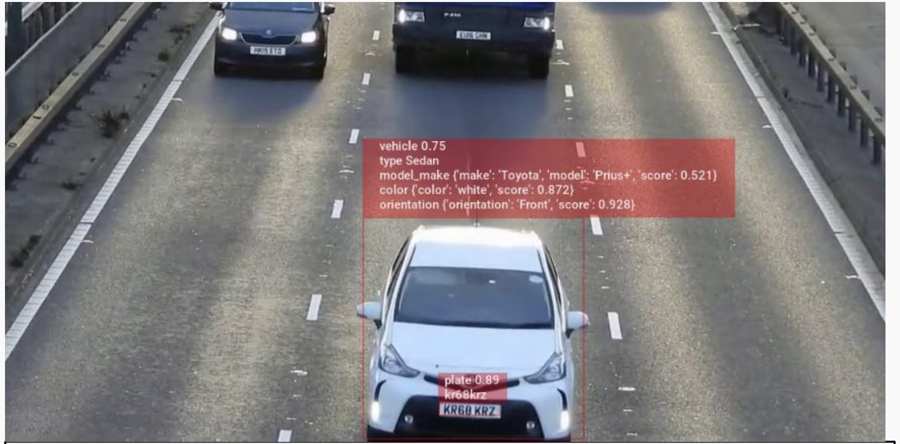
People Counting

What Can Computer Vision do?



```
{
  "verified": True,
  "distance": 0.32450073146188274,
  "max_threshold_to_verify": 0.4,
  "model": "VGG-Face",
  "similarity_metric": "cosine"
}
```

Face Recognition



```
vehicle 0.75
type Sedan
model_make {'make': 'Toyota', 'model': 'Prius+', 'score': 0.521}
color {'color': 'white', 'score': 0.872}
orientation {'orientation': 'Front', 'score': 0.928}
```

```
plate 0.89
kr68krz
```

Vehicle Analytics

What is OpenCV

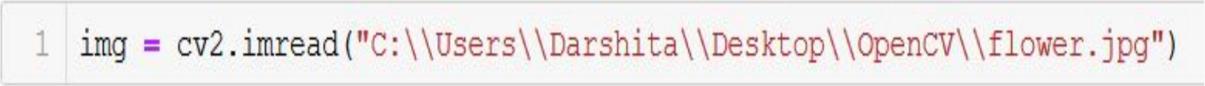
- OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.
- The Library has more than 2500 optimized algorithms for tasks like Object detection, Image Stitching, Classify Human actions in videos etc.
- The library is used extensively in companies (like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota), research groups and by governmental bodies.
- It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

Representing an Image

- Images are a matrix of pixels, where each pixel contains a value in the range 0-255 to denote the color represented by it.
- Every color that can be rendered on a computer screen can be represented as a combination of Red, Green and Blue.
- Eg - Red (255,0,0), Green(0, 255, 0), Blue (0, 0 , 255), Orange(255,165,0)

OpenCV Basics

- Reading an Image



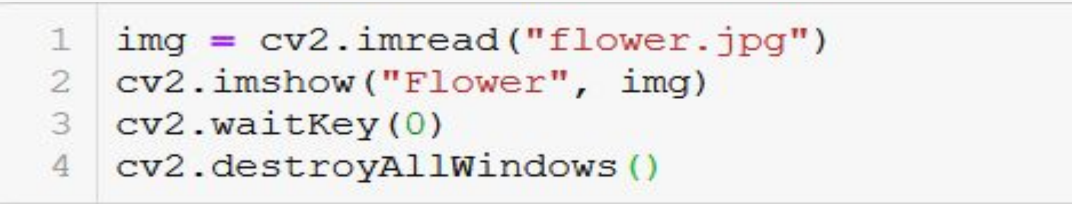
```
img = cv2.imread("C:\\Users\\Darshita\\Desktop\\OpenCV\\flower.jpg")
```

cv2.imread command reads an Image from the given path and stores it in a form of a matrix.

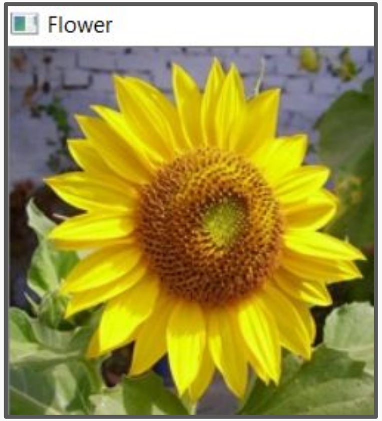
Note - When the image is read with the OpenCV function imread(), the order of colors is **BGR (blue, green, red)**.

OpenCV Basics

- Displaying an Image



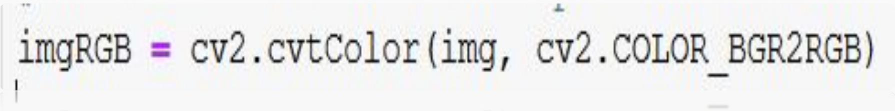
```
img = cv2.imread("flower.jpg")
cv2.imshow("Flower", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



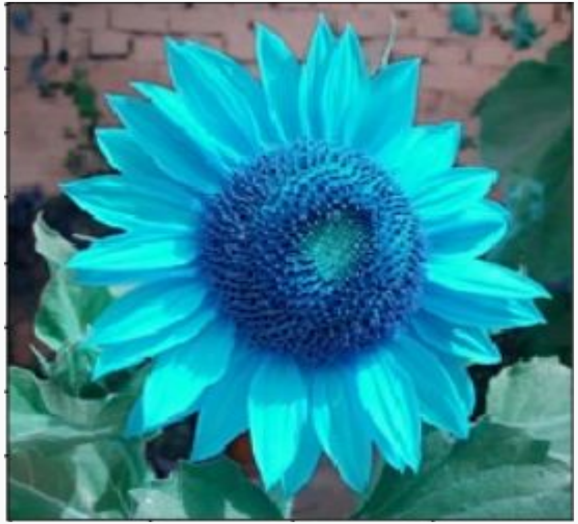
cv2.imshow() method is used to display an image in a window. The window automatically fits the image size.

OpenCV Basics

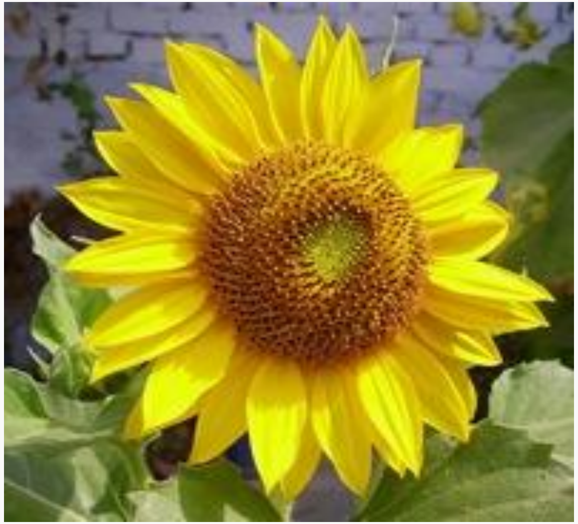
- Converting RGB to BGR



```
imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```



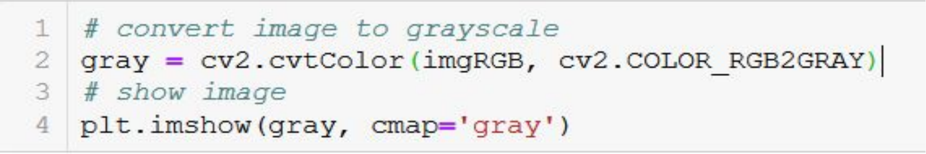
Input BGR image



Output RGB image

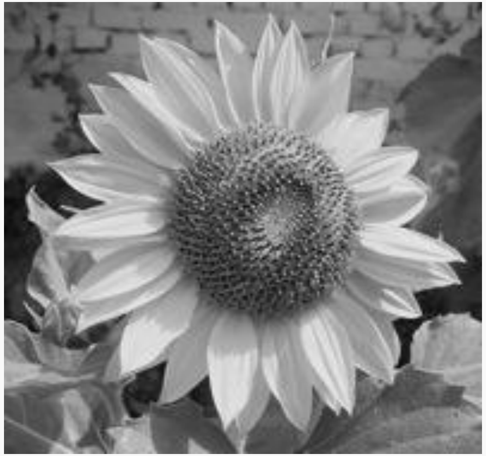
OpenCV Basics

- Converting RGB to Grayscale



```
# convert image to grayscale
gray = cv2.cvtColor(imgRGB, cv2.COLOR_RGB2GRAY)
# show image
plt.imshow(gray, cmap='gray')
```

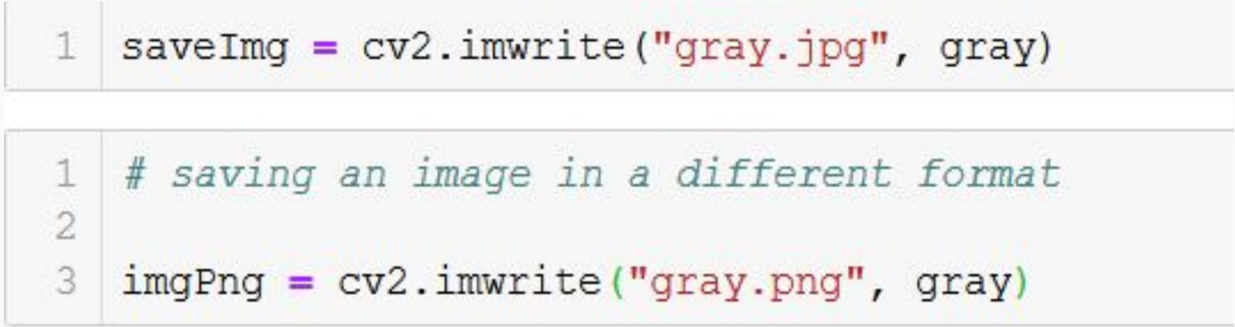
Input RGB Image



Output Grayscale Image

OpenCV Basics

- Saving an Image



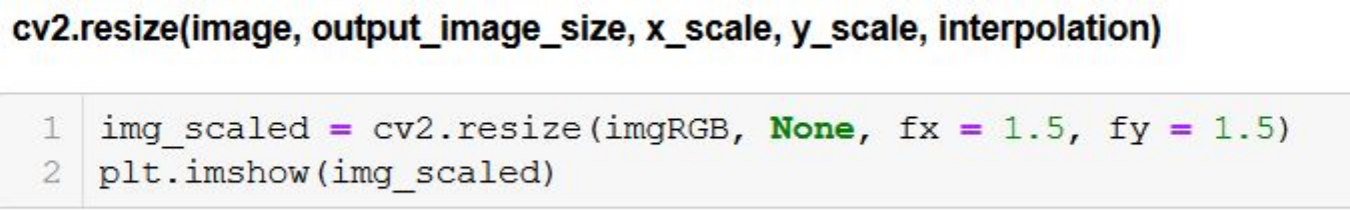
```
saveImg = cv2.imwrite("gray.jpg", gray)
```

```
# saving an image in a different format

imgPng = cv2.imwrite("gray.png", gray)
```

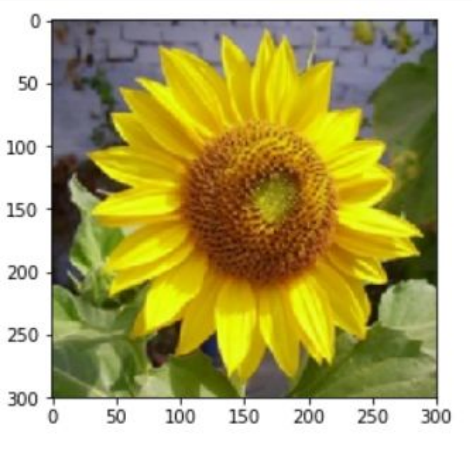
OpenCV Basics

- Image Scaling



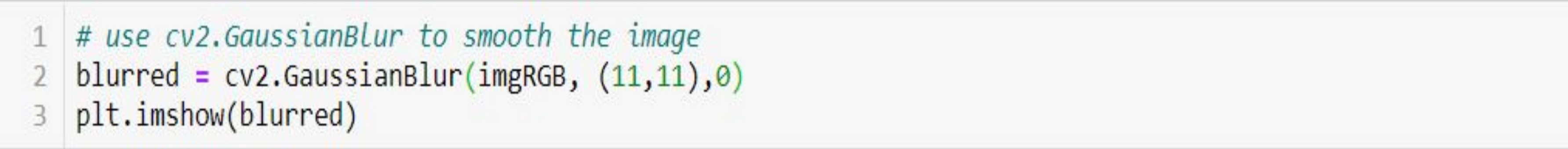
cv2.resize(image, output_image_size, x_scale, y_scale, interpolation)

```python
img_scaled = cv2.resize(imgRGB, None, fx = 1.5, fy = 1.5)
plt.imshow(img_scaled)
```



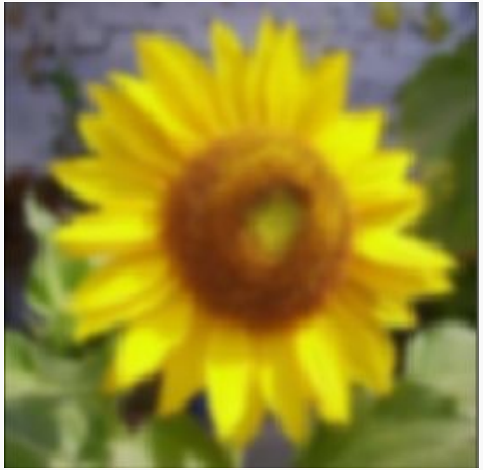
OpenCV Basics

- Image Smoothing



```python
# use cv2.GaussianBlur to smooth the image
blurred = cv2.GaussianBlur(imgRGB, (11,11),0)
plt.imshow(blurred)
```

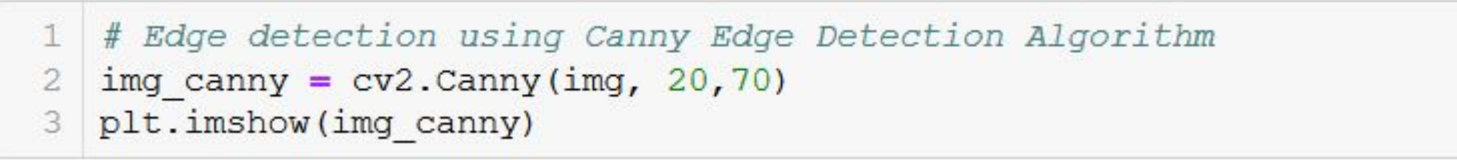
Input Image



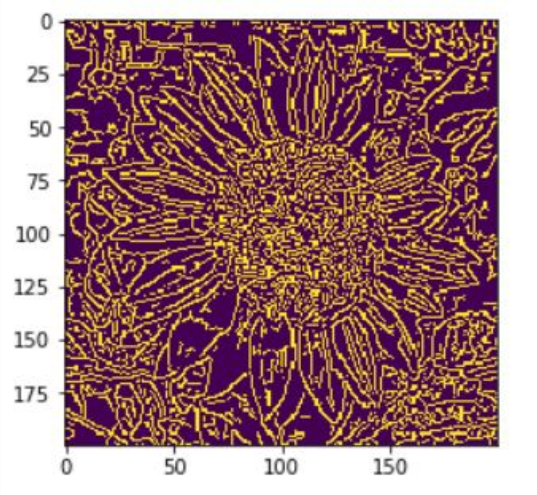
Output Image

OpenCV Basics

- Edge Detection

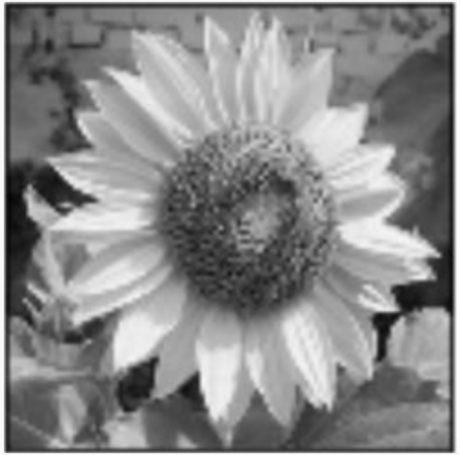


```
# Edge detection using Canny Edge Detection Algorithm
img_canny = cv2.Canny(img, 20,70)
plt.imshow(img_canny)
```

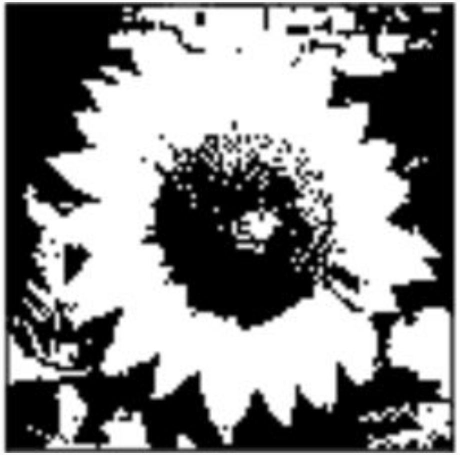


OpenCV Basics

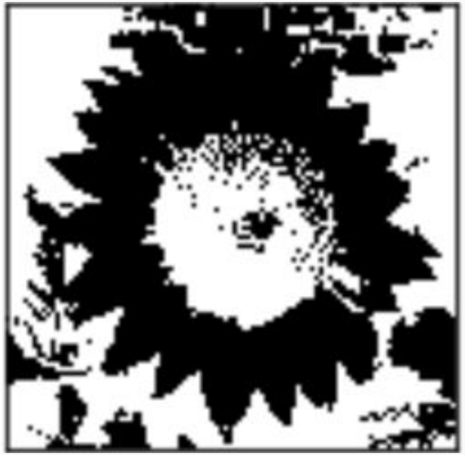
- Image Thresholding



Input Image



Binary



Binary-INV

Hands-On Lab

Challenge!

Image processing:

- Try to create and enhance your own image with multiple openCV techniques
- Please combine multiple techniques such as blurring, masking, resizing etc. Also play around with the numbers :)

Face and eye recognition:

- Try to use your own image to perform face and eye detection
- Then perform face and eye detection on the real time camera via webcam (if possible) or from recorded video! - Maybe required local installation of cv2
- Share your result with your friends!