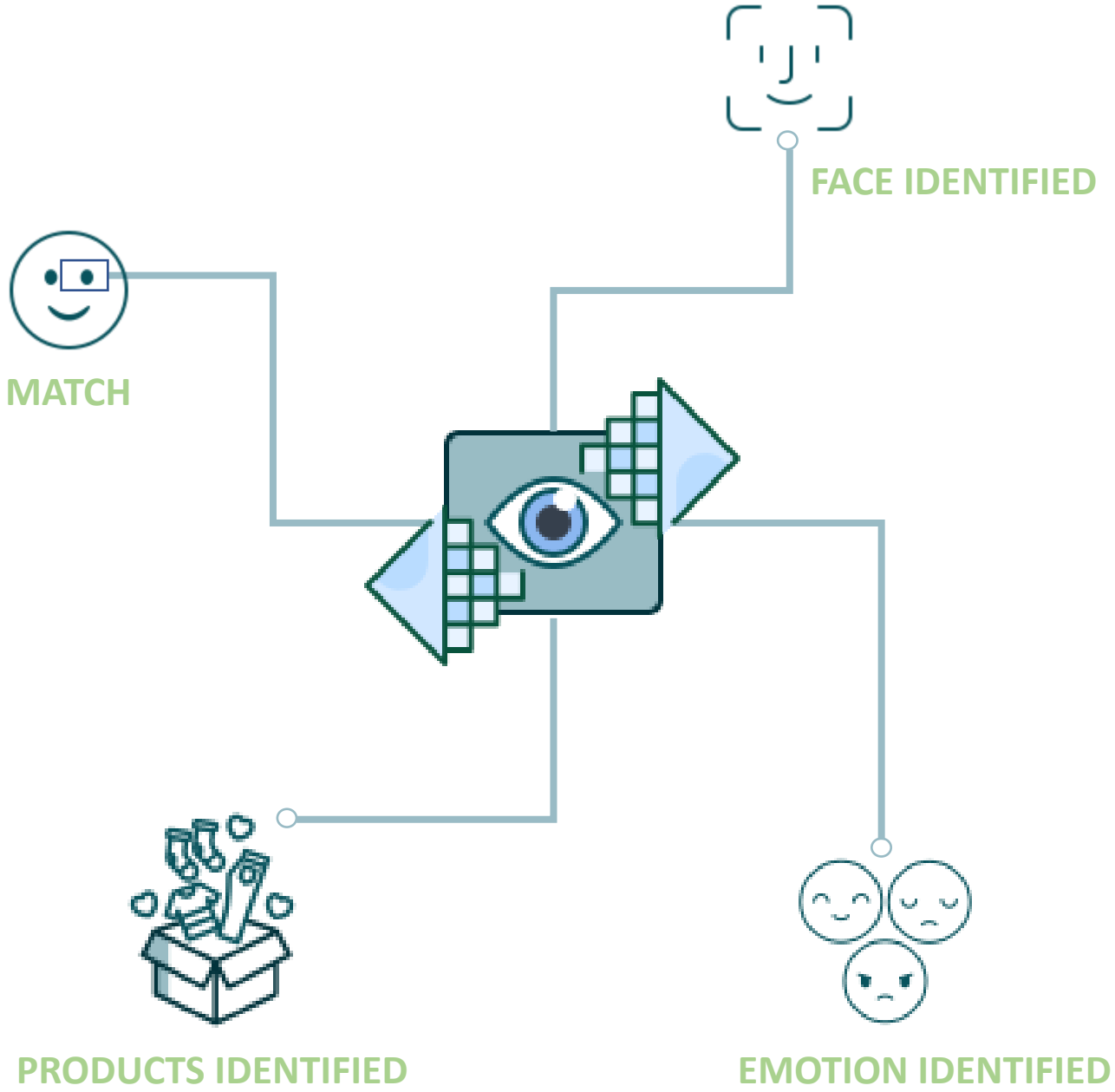
Computer Vision Lecture-one

Introduction



Eng: Noor Alhakim

Computer Vision

Make computers understand images and video (extract information from an image that is necessary to solve task).

Computer Vision Levels

High-Level Vision

Computer Vision vs Image Processing

The methods that are used in Image Processing can alter images in a variety of ways, including sharpening, smoothing, filtering, enhancing, restoring, and blurring amongst others. **Computer vision, on the other hand, is concerned with deciphering the meaning of what may be seen by computers**

Mid-Level Vision

Low-Level Vision

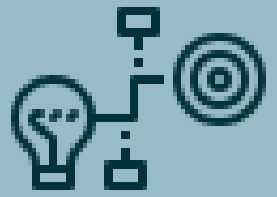
Tools Requirements

Pycharm

Python 3.6

VSCode

Opencv 3.4.1.15



Lecture Plan

Basic operation on images

- Read Images
- Gray and color Image in Read function
- Save Images

Image Types and Color spaces

- Binary Images
- Grayscale Images
- RGB Images
- Indexed Images
- Changing Color spaces

2D Transformations on images

- Scaling
- Translation
- Rotate
- Flip

Basic operation on images

- Read Images

```
image = cv.imread('Home.jpg')
cv.imshow("img",image)
cv.waitKey(0)
```

- Gray and color Image in Read function

```
#1 methode
image = cv.imread('Home.jpg’,0)
cv.imshow("gray",image)
cv.waitKey(0)
```

- Gray and color Image in Read function

```
#2 methode
image = cv.imread('Home.jpg')
gray = cv.cvtColor(image,
cv.COLOR_BGR2GRAY)
cv.imshow('gray',gray)
cv.waitKey(0)
```

- Image Size and Shape

```
image = cv.imread('Home.jpg')
cv.imshow(“img",image)
# size image
print( image.shape )
#(506, 760, 3)
# size image W*H*RGB
print( image.size )
# 1153680
```

Basic operation on images

- Image Type and Channels

```
# image type
print( image.dtype )
# uint8
#
b,g,r = cv.split(image)
# cv.imshow("blue",b)
# or
cv.imshow("blue",image[:,:,1])
# merge 3 color
img = cv.merge((b,g,r))
cv.imshow("image",img)
cv.waitKey(0)
```

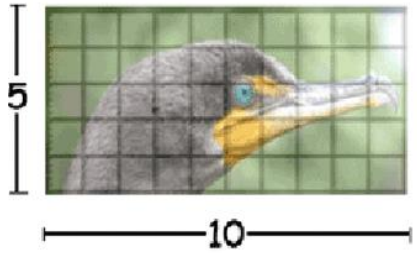
- Save Images

```
image = cv.imread('Home.jpg')
cv.imshow('image',image)
k = cv.waitKey(0)

if k == 27: # wait for ESC key to exit

cv.destroyAllWindows()

elif k == ord('s'): # wait for 's' key
to save and exit

cv.imwrite('messigray.png',image)
cv.destroyAllWindows()
```

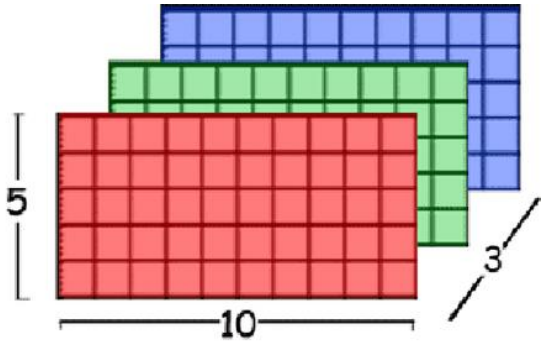
Image Types

Binary and Grayscale Image

RGB Image

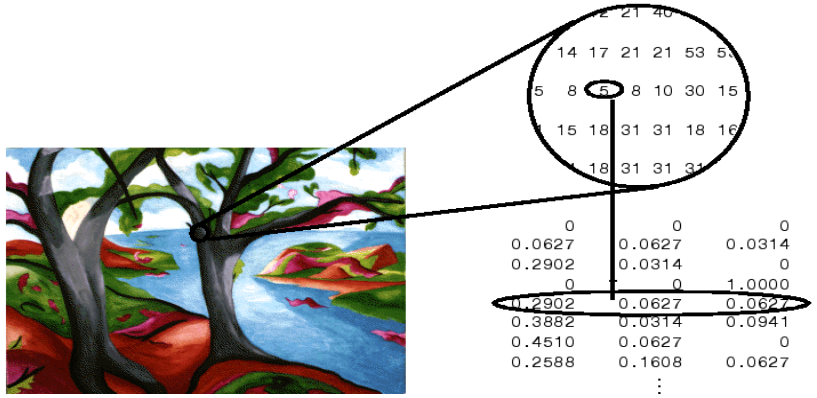


Original Color Image



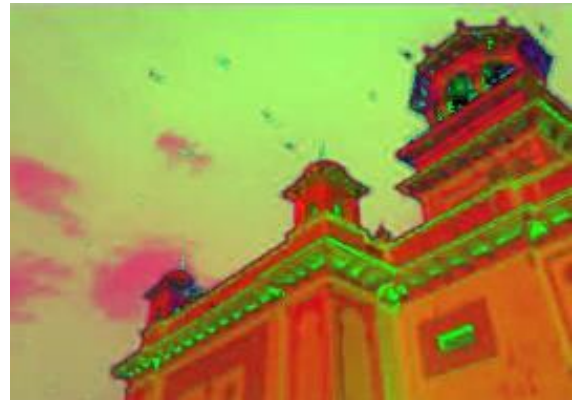
Matlab RGB Matrix

Indexed Image



Changing Color Spaces

```
image = cv.imread('Home.jpg')
cv.imshow("img",image)
# Hsv
HSV=cv.cvtColor(image,cv.COLOR_BGR2HSV)
cv.imshow("HSV",HSV)
#gray
gray=cv.cvtColor(image,cv.COLOR_BGR2GRAY)
cv.imshow("gray",gray)
# BGR
gbr=cv.cvtColor(gray,cv.COLOR_GRAY2RGB)
cv.imshow("gbr",gbr)

cv.waitKey(0)
```



2D Transformations on Images

- Scaling

```
img = cv.imread('Home.jpg',0)
cv.imshow("img",img)
print(img.shape)
cv.waitKey(0)
# resize
dimentions=img.shape
res =cv.resize(img,(int(dimentions[1]/2),int(dimentions[0]/2)))
cv.imshow("res",res)
cv.waitKey(0)
```

2D Transformations on Images

- Translation

```
img = cv2.imread('Images/Home.jpg')
rows,cols,depth = img.shape
cv2.imshow('img',img);

cv2.waitKey(0)

M = np.float32([[1,0,100],[0,1,50]])
dst = cv2.warpAffine(img,M,(cols,rows))
cv2.imshow('dst',dst);

cv2.waitKey(0)
```

2D Transformations on Images

- Rotate

```
img = cv2.imread('Images/Home.jpg',0)
rows,cols = img.shape

M=cv2.getRotationMatrix2D((cols/2,rows/2),
90,1)

dst = cv2.warpAffine(img,M,(cols,rows))
cv2.imshow('dst',dst);

cv2.waitKey(0)
```

2D Transformations on Images

- Rotate

```
img = cv2.imread('Images/Home.jpg')
img_rotate_90_clockwise =
cv2.rotate(img,cv2.ROTATE_90_CLOCKWISE)
img_rotate_90_counterclockwise
=cv2.rotate(img,cv2.ROTATE_90_COUNTERCLOCKWISE)
img_rotate_180 = cv2.rotate(img,cv2.ROTATE_180)
```

What is The Result?

2D Transformations on Images

- Flip

Original Image

```
img2 = cv2.flip(img, 1)
```

```
img3 = cv2.flip(img, 0)
```

```
img4 = cv2.flip(img, -1)
```

That’s All