Python Applied to Machine Learning and Statistics

Lecture 03: Computer Vision with Python

Kelwin Fernandes Ricardo Cruz Pedro Costa

September 16, 2016

OpenCV

OpenCV: Advantages

- BSD license... free for both academic and commercial use.
- Windows, Linux, Mac OS, iOS and Android.

OpenCV: Advantages

- BSD license... free for both academic and commercial use.
- Windows, Linux, Mac OS, iOS and Android.

OpenCV: Advantages

- BSD license... free for both academic and commercial use.
- Windows, Linux, Mac OS, iOS and Android.
- A lot of modules:

- Image processing
- I/O
- High-level GUI
- Video Analysis
- Camera Calibration and 3D Reconstruction
- Feature Detection and Description
- Object Detection
- Machine Learning
- Clustering and Search in Multi-Dimensional Spaces
- Computational Photography
- Images stitching
- Operations on Matrices
- Background Segmentation
- Optical Flow
- Stereo Correspondence
- Super Resolution
- Video Stabilization
- Image Registration
- RGB-Depth Processing
- Surface Matching
- Structure From Motion
- Deformable Part-based Models
- . . .

Input and Output

- Reading images.

```python
import cv2

img = cv2.imread(path)
img = cv2.imread(path, 0)       # Grayscale
```

- Show images.

```python
cv2.namedWindow('winname')     # Optional
cv2.imshow('winname', img)
cv2.waitKey(0)
cv2.destroyWindow('winname')   # Optional
```

- Store images (allows parameters like quality, compression, binarization).

```python
cv2.imwrite(path, img)
```

Input and Output: Video

- Opening a video stream.

```
cap = cv2.VideoCapture(0)                 # Default camera

while(True):
    ret, frame = cap.read()
    if ret:
        cv2.imshow(’img’, frame)
        if cv2.waitKey(1) & 0xFF == ord(’q’):
            break
    else:
        break
cap.release()
```

- Video from file.

```
cap = cv2.VideoCapture(’video.avi’)
```

Image representation

- In C++ OpenCV has its own data type... `Mat`
- In python, images are represented as `numpy` matrices.

- Grayscale: `img.shape` → `(rows, cols)`.
- Color: `img.shape` → `(rows, cols, channels)`.

- Default type: `np.uint8`... [0..255]
- Possible conversion to other types using `numpy` operators: `img.astype(np.float64)`.

Colorspaces

- Default colorspace: `BGR`.
- Function to perform colorspace conversion: `cvtColor`

```
hsv = cvtColor(img, cv2.COLOR_BGR2HSV)
gray = cvtColor(img, cv2.COLOR_BGR2GRAY)
bgr = cvtColor(hsv, cv2.COLOR_HSV2BGR)
```

- Splitting and merging channels.

```
b, g, r = cv2.split(img)
img = cv2.merge([b, g, r])
```

- Splitting using `numpy` slices:

```
b = img[:, :, 0]
g = img[:, :, 1]
r = img[:, :, 2]
```

Direct manipulation, ROIs, masks

- Random access

```
px = img[42, 21]
img[42, 21] = [0, 0, 253]
```

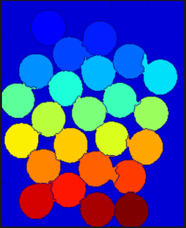
- Region of Interest

```
roi = img[100:200, 150:200]
img[200:300, 220:270] = roi
```

- Apply a mask

```
img = cv2.bitwise_and(img, img, mask=mask)  # Using OpenCV
img[mask == 0] = 0                          # Using numpy
```

Thresholding, morphologic op., watershed



- Thresholding

```
cv2.threshold(gray, 0, 255,
                   cv2.THRESH_BINARY_INV +
                   cv2.THRESH_OTSU)
```

- Morphologic operations

```
kernel = np.ones((3,3), np.uint8)
dst = cv2.dilate(img, kernel, iterations=3)
```

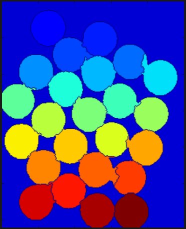
- Distance transform

```
dst=cv2.distanceTransform(img, cv2.DIST_L2,
    5)
```

- Watershed

```
markers = cv2.connectedComponents(fground)
dst = cv2.watershed(img, markers)
```

Thresholding, morphologic op., watershed



- Thresholding

```
cv2.threshold(gray, 0, 255,
                  cv2.THRESH_BINARY_INV +
                  cv2.THRESH_OTSU)
```

- Morphologic operations

```
kernel = np.ones((3,3), np.uint8)
dst = cv2.dilate(img, kernel, iterations=3)
```

- Distance transform

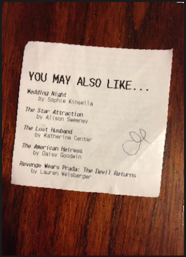
```
dst=cv2.distanceTransform(img, cv2.DIST_L2,
     5)
```

- Watershed

```
markers = cv2.connectedComponents(fground)
dst = cv2.watershed(img, markers)
```

Segment the coins using these or other functions.

Blur, Edge detection, Contours and Perspective



YOU MAY ALSO LIKE...

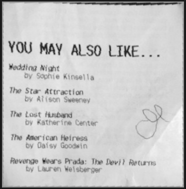
Wedding Night
by Sophie Kinsella

The Star Attraction
by Alison Sweeney

The Lost Husband
by Katherine Center

The American Heiress
by Daisy Goodwin

Revenge Wears Prada: The Devil Returns
by Lauren Weisberger



YOU MAY ALSO LIKE...

Wedding Night
by Sophie Kinsella

The Star Attraction
by Alison Sweeney

The Lost Husband
by Katherine Center

The American Heiress
by Daisy Goodwin

Revenge Wears Prada: The Devil Returns
by Lauren Weisberger

- Gaussian Blur

```
gray = cv2.GaussianBlur(gray, (k, k), 0)
```

- Edge detection

```
edged = cv2.Canny(gray, 75, 200)
```

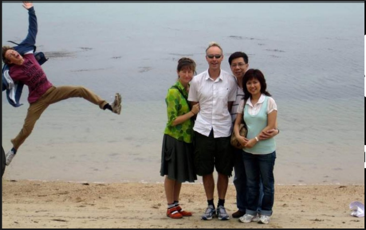
- Contours

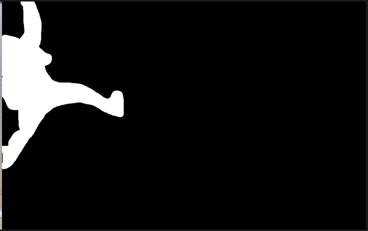
```
cnts, _ = cv2.findContours(edged.copy(),
                           cv2.RETR_LIST,
                           cv2.CHAIN_APPROX_SIMPLE)

cv2.contourArea(c)
peri = cv2.arcLength(c, True)
approx = cv2.approxPolyDP(c, k * peri, True)
```

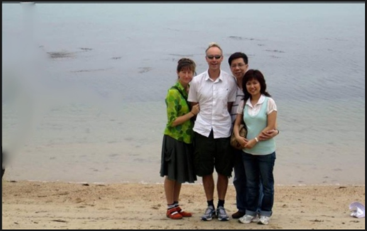
- Perspective transform

```
M = cv2.getPerspectiveTransform(rect, dst)
warped = cv2.warpPerspective(img, M,
                             (mxwidth, mxheight))
```

Image inpainting





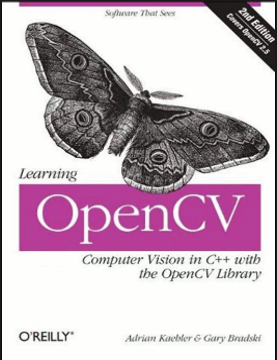


```
dst = cv2.inpaint(img, mask, 10, cv2.INPAINT_TELEA)
```

Resourcers

pyimagesearch

be awesome at building image search engines



Software That Sees

2nd Edition

Covers OpenCV 2.x

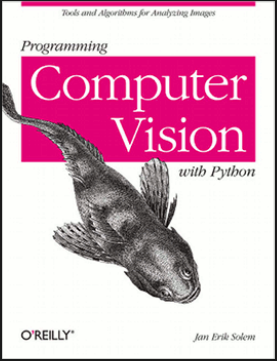
Learning

OpenCV

Computer Vision in C++ with the OpenCV Library

O'REILLY®

Adrian Kaehler & Gary Bradski



Tools and Algorithms for Analyzing Images

Programming

Computer Vision

with Python

O'REILLY®

Jan Erik Solem