CURSO DE VISÃO COMPUTACIONAL COM PYTHON E OPENCV

Hello OpenCV =D

FILIPE ROLIM CORDEIRO (FILIPE.ROLIM@GMAIL.COM)

Quem sou eu?

- Filipe, muito prazer!
- Apaixonado por visão computacional e hearthstone!
- **Meta de vida**: ajudar os computadores a dominarem o mundo!

Quem são vocês?

Por que trocar isso...

NETFLIX

Por isso?

O Curso

- 9 dias de aulas
  - 36 horas
- Máximo de faltas permitidas: 3
- Abordagem baseada em problemas

O Curso

- Não haverá prova

Curso Light

Ok, mas o que vamos aprender no curso?

Habilidades

- Operações básicas em imagens;
- Detecção de bordas;
- Segmentação de imagens;
- Reconhecimento de pokemons;
- Reconhecimento de escrita;
- Reconhecimento de rostos;
- Reconhecimento de placas;
- Reconhecimento de Gestos;
- Reconhecimento de movimento
- E muito mais!!!!

O que danado é Visão Computacional, meu amigo?

Que parada é essa mano?

Motivação

LIFE
The Cameraphone Revolution
How This Little Gadget Will Change Your Life
28

SKYDIVE DUBAI

Reconhecimento de Placas

Reconhecimento de Caracteres

Recuperação de Imagens

Objetivo da Visão Computacional

- Permitir computadores **entenderem** o que está sendo visto na imagem

Sonho da Visão computacional

- Máquinas que vêem e compreendem o mundo ao seu redor

Só assim poderão dominar o mundo!! Vamos ajudá-las?

Biology
Psychology
Physics
Computer Science
Engineering
Mathematics
Computer Vision
Neuroscience
optics
Cognitive sciences
Image processing
graphics, algorithms, theory,…
Speech, NLP
systems, architecture, …
Robotics
Information retrieval
Machine learning

Tarefas Principais

(a) classification

(b) detection

(c) segmentation

2:53 PM
Google goggles labs
Landmark
Golden Gate Bridge
Golden Gate Bridge
Web Results
Golden Gate Bridge - Wikipedia, the free encyclopedia
The Golden Gate Bridge by night, with part of downtown San Francisco ... Golden Gate Bridge is the most popular place to commit suicide in the United States ...
http://en.wikipedia.org/wiki/Golden_Gate_Bridge
Seacliff Webcam - Weather Seacliff, Golden Gate Bridge (Seacliff ...

107-406 CALORIES
Salad And Chicken Nuggets

CHALK
CHALK

T-Mobile NL
16:06
Albums by Impala
Architecture
Time
Score

- Object detection
- Action classification
- Image captioning
- …

license plate
racket
hair
vest
shirt
child
shorts
picture
sneakers
car
hair
fence
grass
racket
rosette
shirt
boy
shorts
rosette
blue
on
shirt
striped
wearing
playing
dark
has
hair
short
wearing
shorts
camoflaged
boy
holding
racket
light
in front of
behind
behind
in front of
fence
in front of
grass
green
behind
behind
behind
car
behind
in front of
in front of
has
brown
white
holding
racket
light
parked
child
has
hair
brown
license plate
wearing
shirt
red
wearing
vest
blue
playing
blue
wearing
sneakers
denim
red
wearing
shorts
has
picture
yellow

sky
Building
signs
Building
Putnam
Crosswalk
Pedestrians
road

[redacted]

Aprendizagem

Visão Computacional ainda tem muitos desafios

Desafios

- Sensores de câmeras e limitações de lentes

Desafios

- Pontos de vista diferentes

Desafios

- Variação de Luz

Desafios

- Escala

Desafios

- Deformações não rígidas

Desafios

- Oclusão

Desafios

- Clutter

Desafios

- Variação de classes

Desafios

- Imagens Ambíguas/ Ilusão de ótica

Visão humana

- Funciona e processa de forma diferente

Você bebeu?

Você bebeu denovo?

Imagem é uma matriz de números

62	62	63	64	65	66	67	67	69	70	71	72	72	73	73	73	73	72	72	71	70	69	67	66	66	66	65	63	62	61	60	60
61	62	63	64	66	66	67	68	68	69	70	71	71	72	72	73	72	72	71	71	70	69	68	66	66	65	65	63	62	61	60	60
61	62	63	64	66	66	68	68	69	70	70	71	72	73	73	73	72	72	71	71	69	68	67	66	66	65	65	64	63	62	61	61
61	63	64	64	66	67	68	68	68	69	70	71	71	73	73	74	73	73	73	71	70	69	68	66	66	65	64	63	62	61	61	60
61	63	64	65	67	68	69	69	70	70	71	71	72	55	53	69	72	72	71	71	70	69	68	67	66	65	64	63	62	60	60	60
63	64	65	66	67	68	69	69	70	70	71	72	42	4	5	11	48	72	71	71	69	69	68	67	66	65	64	62	62	60	59	59
63	65	66	66	68	68	69	70	71	71	71	72	18	4	4	7	8	66	71	70	69	68	68	67	66	65	64	63	61	59	59	58
63	65	67	67	68	69	69	70	71	71	72	64	4	27	24	54	33	29	52	64	68	68	67	66	65	64	63	62	61	59	58	58
64	65	66	66	68	69	70	71	41	24	24	12	17	24	48	60	37	43	36	52	66	68	67	66	65	64	63	61	60	59	58	57
65	66	67	67	68	69	71	49	6	6	6	5	34	36	12	47	34	17	29	54	43	63	67	66	65	64	63	62	60	59	58	57
64	65	66	66	68	69	38	6	6	5	5	7	16	19	4	47	44	27	24	40	67	66	66	65	65	64	63	61	60	59	58	57
63	64	65	65	67	30	6	6	5	5	5	6	8	9	20	27	51	78	41	44	66	65	65	65	65	64	63	62	60	59	58	57
63	64	65	65	34	5	5	5	5	5	5	5	4	19	6	7	54	64	20	59	65	65	64	64	64	63	62	61	60	59	57	56
63	64	64	65	14	5	6	5	5	4	5	4	18	7	5	4	19	10	11	65	64	64	64	63	61	66	62	61	60	59	58	56
63	64	64	65	53	7	4	5	6	6	7	10	6	5	5	4	21	24	18	64	64	64	63	62	64	65	62	62	60	59	58	57
64	64	64	64	65	50	4	4	4	5	11	16	6	6	4	6	35	16	26	66	64	64	63	61	72	67	63	62	61	59	58	57
64	64	64	64	65	46	4	4	4	5	6	9	8	5	29	10	43	56	29	57	64	64	63	61	70	67	62	64	65	59	59	57
64	64	64	65	66	27	5	4	4	5	6	6	6	18	66	20	57	60	46	36	75	70	62	61	70	67	62	61	60	59	58	58
49	50	62	65	57	5	5	6	5	6	6	6	6	41	59	28	60	58	44	22	63	71	72	60	69	68	61	60	58	59	59	58
42	52	57	52	26	5	5	5	5	5	5	5	5	70	50	43	61	62	64	39	42	64	60	62	56	63	65	65	67	61	53	53
32	32	32	33	6	5	5	5	5	5	6	6	11	39	21	33	51	50	45	46	18	32	36	33	23	44	70	71	51	42	27	31
50	50	51	39	5	5	5	5	6	5	6	6	42	69	28	34	42	39	43	37	26	29	40	26	29	26	35	42	35	33	18	19
52	53	51	22	5	5	5	5	6	5	6	5	44	56	17	51	54	53	54	56	51	22	54	54	55	55	54	53	53	53	52	52
54	54	53	8	5	5	5	5	6	5	6	13	52	42	21	51	54	51	49	49	50	22	41	45	42	42	41	40	41	44	43	42
52	52	54	36	8	5	5	6	6	5	6	28	55	32	32	54	53	51	51	51	51	44	25	51	51	49	49	50	49	48	46	46
54	54	52	53	30	7	5	6	6	5	6	40	54	29	52	51	53	56	55	52	52	51	38	52	52	50	49	46	46	45	46	47
51	52	51	53	27	14	5	4	5	4	7	47	51	21	39	49	47	49	52	52	52	49	35	31	48	46	47	47	47	46	46	43
48	50	51	53	25	14	17	8	4	4	17	46	40	18	43	47	46	49	52	54	53	53	54	18	50	49	46	47	47	47	47	45
49	49	49	49	22	12	20	24	6	14	35	51	39	48	48	50	51	51	49	51	51	52	50	41	58	48	47	47	47	45	45	46
51	49	50	50	22	13	19	36	13	12	42	50	40	73	50	50	50	49	48	49	49	48	49	45	51	46	44	44	44	42	45	47
47	49	49	47	20	16	26	39	21	15	36	48	42	61	47	48	51	47	50	51	51	51	49	47	47	52	47	47	44	43	45	46
48	50	48	52	19	13	33	36	18	18	36	49	51	54	47	47	49	46	46	49	49	49	49	47	44	53	44	48	44	46	46	45

Aqui são 256 tons de cinza!! (e não apenas 50)

O que nós enxergamos

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

O que o computador enxerga

Largura
520
j=1
i=1
Itensidade: [0,255]
500
Altura
im[176][201] tem valor
164
im[194][203] temo valor 37

Tons de Cinza

Que danado é OpenCV?

Por que é que a gente vai usar isso mano?

O que é OpenCV

- Open Source Vison Library
- Biblioteca
- Código aberto
- Visão Computacional
- Alto desempenho
- Uma das ferramentas mais populares da indústria de visão computacional

Usuários

Google

intel

HONDA

YAHOO!

IBM

TOYOTA

Microsoft

SONY

Plataformas

Linguagens

- C++
- C
- Python
- Java

Calvin Klein

Código em Python

```python
# import the necessary packages
import cv2

# load our image and convert it to grayscale
image = cv2.imread("../turma_md2016_1.jpg")
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# load the face detector and detect faces in the image
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
rects = detector.detectMultiScale(gray, scaleFactor=1.05, minNeighbors=7,
    minSize=(30, 30), flags=cv2.cv.CV_HAAR_SCALE_IMAGE)

# loop over the faces and draw a rectangle surrounding each
for (x, y, w, h) in rects:
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

# show the detected faces
cv2.imshow("Final", image)
cv2.waitKey(0)

#Save detection
cv2.imwrite('faces.jpg', image)
```

Quick Python Tutorial

python™

IS LOVE

Python é fácil

- Em python...

```
print “Hello World”
```

Exercício

- Faça um Hello World

Características

- Uso de "; " no fim de comandos não é obrigatório
- Dinamicamente tipada
- Exemplo
  - >>>a = 10
  - >>>a = 'teste'

Desvio Condicional

- Sintaxe:

```
if <condição>:
<bloco de comandos>
elif <condição>:
<bloco de comandos>
elif <condição>:
<bloco de comandos>
      .....
else:
<bloco de comandos>
```

Desvio Condicional

▷Exemplo: imprimir diferença ou soma de 2 números

```
num = input('Primeiro número')
num2 = input('Segundo número')
if (num>=num2):
    r = num - num2
else:
    r = num + num2
print 'O Resultado é:',r
```

Não precisa do (parêntesis), mas pode ser usado

Desvio Condicional

▷Exemplo:

```
num1 = input("Digite a primeira nota...")
num2 = input("Digite a segunda nota...")
media = (num1 + num2)/2.0
if media == 10:
    print "Aprovado nota máxima"
elif media>=7:
    print "Aprovado com média = ", media
else:
    print "Reprovado com média = ", media
```

For

- O **for** de python é parecido com o **foreach** de outras linguagens
- Formato Geral:

```
for var in lista:
    <bloco de
comandos1>
else:
     <bloco de
comandos2>
```

For

- O [...] representa uma lista

```
>>> for elemento in [1,3,3,6,4,2,7]:
            print elemento*2

2
6
6
12
8
4
14
```

Função **`range`**

- Retorna uma progressão aritmética de inteiros numa lista
- range (**início**, **parada**, **incremento**)
  - **início** (opcional) é o primeiro valor a ser gerado na lista (default: 0)
  - **parada** é o limite da progressão: a progressão termina no último valor antes de parada
  - **incremento(opcional**) é o incremento da progressão (default:1)

Função range

- A função **range** retorna uma lista de números

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
>>> range(5,10)
[5, 6, 7, 8, 9]
```

```
>>> range(0,10,2)
[0, 2, 4, 6, 8]
```

```
>>> for i in range(10):
        print i,

0 1 2 3 4 5 6 7 8 9
>>> for i in range(5,10):
        print i,

5 6 7 8 9
>>> for i in range(0,10,2):
        print i,

0 2 4 6 8
```

For

- Mais exemplos:

```
>>> for x in ["spam", "eggs","ham"]:
        print x


spam
eggs
ham
```

```
>>> for m in ["teste","de","for"]:
        print (m, len(m))

('teste', 5)
('de', 2)
('for', 3)
```

Exercício

- Faça um código para mostrar a tabuada dos números ímpares entre 10 e 20

Numpy

- Biblioteca de python para trabalhar com arrays multidimensionais
  - Bastante utilizado pelo opencv
  - Imagens são matrizes em numpy

Numpy

- ndim
- shape
- size
- dtype

Numpy

```
>>> import numpy as np
>>> a = np.arange(15).reshape(3, 5)
>>> a
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
>>> a.shape
(3, 5)
>>> a.ndim
2
>>> a.dtype.name
'int64'
```

Numpy – Criação de Matrizes

- zeros
- ones

```
>>> np.zeros( (3,4) )
array([[ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
>>> np.ones( (2,3,4), dtype=np.int16 )
array([[[ 1, 1, 1, 1],
        [ 1, 1, 1, 1],
        [ 1, 1, 1, 1]],
       [[ 1, 1, 1, 1],
        [ 1, 1, 1, 1],
        [ 1, 1, 1, 1]]], dtype=int16)
```

Numpy

- Como gerar imagens brancas/pretas?

Hello OpenCV World!

```
aula1 — python2.7 — 62×5
(cv_24) MacBook-Pro-de-Filipe:aula1 filipe$ python hello.py
largura: 480 pixels
altura: 360 pixels
channels: 3
```

Code

```
import cv2

#le a imagem e mostra informacoes basicas
image = cv2.imread('hello.jpg')
print "largura: %d pixels" % (image.shape[1])
print "altura: %d pixels" % (image.shape[0])
print "channels: %d" % (image.shape[2])

#mostra a imagem e esperar pressionar a tecla
cv2.imshow("Imagem", image)
cv2.waitKey(0)
```

Exercício

- Leia e mostre uma imagem usando OpenCV

Salvar Imagem

- **imwrite**

```
cv2.imwrite("novaImagem.png",image)
```

- **Exercício**
  - Salve a imagem lida com um novo nome

Sistema de Coordenadas

Como faz isso?

Numpy Slice

- Array[inicio:fim-1]
- Matriz[inicio:fim-1, inicio:fim-1]

Exercício

- Faça o slicing da imagem abaixo

Mudando a cor

```
image[0:metade_altura, 0:metade_largura] = (0, 255, 0)

cv2.imshow("Atualizado", image)
cv2.waitKey(0)
```

O que acontece??

Exercício

- Gere a seguinte imagem

Desenho com OpenCV

Coordenadas para Desenho

Desenhando

- Básico:
  - cv2.line
  - cv2.rectangle
  - cv2.circle

- Onde vai ser desenhado?
  - Matriz nova
    - zeros, ones
  - Imagem

Desenhando linhas

- cv.line
- Parametros:
  - Img
  - Pt1
  - Pt2
  - Cor
  - Thickness (opcional)
- Exemplo

```
cv2.line(imagem,(100,0),(300,300),(0,255,255))
```

Desenhando linhas

```
#inicializa o canvas de tamanho 300x300
#com 3 canais: R,G, B, com fundo preto
canvas = np.zeros((300,300,3), dtype="uint8")

green = (0,255,0)
cv2.line(canvas,(100,0),(300,300),green)
cv2.imshow("Canvas", canvas)
cv2.waitKey(0)
```

Desenhando linhas

```
#inicializa o canvas de tamanho 300x300
#com 3 canais: R,G, B, com fundo preto
canvas = np.zeros((300,300,3), dtype="uint8")

green = (0,255,0)
cv2.line(canvas,(0,0),(300,300),green)
cv2.imshow("Canvas", canvas)
cv2.waitKey(0)
```

```
red = (0,0,255)
cv2.line(canvas,(200,0),(0,300),red,3)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

Desenhando

```
#inicializa o canvas de tamanho 300x300
#com 3 canais: R,G, B, com fundo preto
canvas = np.zeros((300,300,3), dtype="uint8")
```

```
green = (0,255,0)
cv2.line(canvas,(0,0),(300,300),green)
cv2.imshow("Canvas", canvas)
cv2.waitKey(0)
```

```
red = (0,0,255)
cv2.line(canvas,(300,0),(0,300),red,3)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

```
cv2.rectangle(canvas,(10,10),(60,60),green)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

Desenhando

```
#inicializa o canvas de tamanho 300x300
#com 3 canais: R,G, B, com fundo preto
canvas = np.zeros((300,300,3), dtype="uint8")
```

```
green = (0,255,0)
cv2.line(canvas,(0,0),(300,300),green)
cv2.imshow("Canvas", canvas)
cv2.waitKey(0)
```

```
red = (0,0,255)
cv2.line(canvas,(300,0),(0,300),red,3)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

```
cv2.rectangle(canvas,(10,10),(60,60),green)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

```
cv2.rectangle(canvas, (50,200),(200,225),red,5)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

Desenhando

```
#inicializa o canvas de tamanho 300x300
#com 3 canais: R,G, B, com fundo preto
canvas = np.zeros((300,300,3), dtype="uint8")
```

```
green = (0,255,0)
cv2.line(canvas,(0,0),(300,300),green)
cv2.imshow("Canvas", canvas)
cv2.waitKey(0)
```

```
red = (0,0,255)
cv2.line(canvas,(300,0),(0,300),red,3)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

```
cv2.rectangle(canvas,(10,10),(60,60),green)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

```
cv2.rectangle(canvas, (50,200),(200,225),red,5)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

```
blue = (255,0,0)
cv2.rectangle(canvas,(200,50),(225,125),blue, -1)
cv2.imshow("Canvas",canvas)
cv2.waitKey(0)
```

Desenhando círculos

- cv2.circle
- Parametros
  - Img
  - Centro
  - Raio
  - Cor
  - Thickness (opcional)
- Exempo

```
cv2.circle(canvas, (30, 30), 10, (255,255,255))
```

Círculos

```
canvas = np.zeros((300,300,3),dtype="uint8")
(centerX,centerY)=(canvas.shape[1]/2,canvas.shape[0]/2)
white = (255, 255, 255)

#desenha um circulo, aumentando o raio em 25 unidades
for r in xrange(0, 175, 25):
    cv2.circle(canvas, (centerX, centerY), r, white)

cv2.imshow("Canvas", canvas)
cv2.waitKey(0)
```

Plot através do matlplotlib

```
import cv2
import imutils
import matplotlib.pyplot as plt

image = cv2.imread('titanic.jpg')

plt.figure("Imagem")
plt.imshow(imutils.opencv2matplotlib(image))
plt.show()
```

Code: matplot.py

Exercício

- Faça o desenho abaixo

Onde aprender mais

- *Processamento Digital de Imagens*, R. Gonzalez, R. Woods, 3ª edição, Pearson, 2011

Onde aprender mais

- *Computer Vision: Algorithms and Applications*, R. Szeliski, 1ª edição, Springer, 2010

Onde aprender mais

- *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*, A. Kaehler, G. Bradski, 2ª edição, O'Reilly Media, 2013

▷ Tem pra python também

IDEAS
click
Think?
good
Inspiration
CHECK LIST:
- SEO
- 3G / 4G
cloud
SHARE