

Computer Vision

Introduction to OpenCV

Cristiano Nunes – CEFET-MG
Prof. Flávio Cardeal – CEFET-MG



cfgnunes@cefetmg.br



cardeal@cefetmg.br

Abstract

- This lecture introduces a practical overview of OpenCV library by using Python language.

What is OpenCV?

- Open source Computer Vision library;
- Has more than 2500 optimized algorithms;
- Support a lot of different languages:
 - C++, Python, Java, MATLAB;
 - But is written natively in C++;
- Cross form platform:
 - Supports Windows, Linux, Android and Mac OS.

OpenCV applications

- Various applications:
 - Image Processing;
 - Human-Computer Interaction (HCI);
 - Object Identification;
 - Object Recognition;
 - Face Recognition;
 - Motion Tracking;
 - ...and so on.

OpenCV modules

- OpenCV has a modular structure:
 - **core** – contains basic structures and algorithms;
 - **imgproc** – contains the main image processing functions;
 - **video** – motion estimation, feature tracking and foreground extraction;
 - **highgui** – image and video reading and writing functions, and user interface functions;
 - **objdetect** – object detection functions, face and peoples detectors for example.

OpenCV documentation

- All documentation are n the following link:

<https://docs.opencv.org/master/>



The screenshot shows the OpenCV documentation website. At the top left is the OpenCV logo, consisting of three interlocking circles in red, green, and blue. To the right of the logo is the text "OpenCV" in a large, bold, black font. Below "OpenCV" is the text "Open Source Computer Vision" in a smaller, black font. To the right of the text "OpenCV" is a dropdown menu showing "4.0.1-dev" with a downward arrow. Below the logo and text is a horizontal navigation bar with several tabs: "Main Page", "Related Pages", "Modules", "Namespaces", "Classes", "Files", "Examples", and "Java documentation". Below the navigation bar is a section titled "OpenCV modules" in a bold, black font. To the right of this title is a list of links:

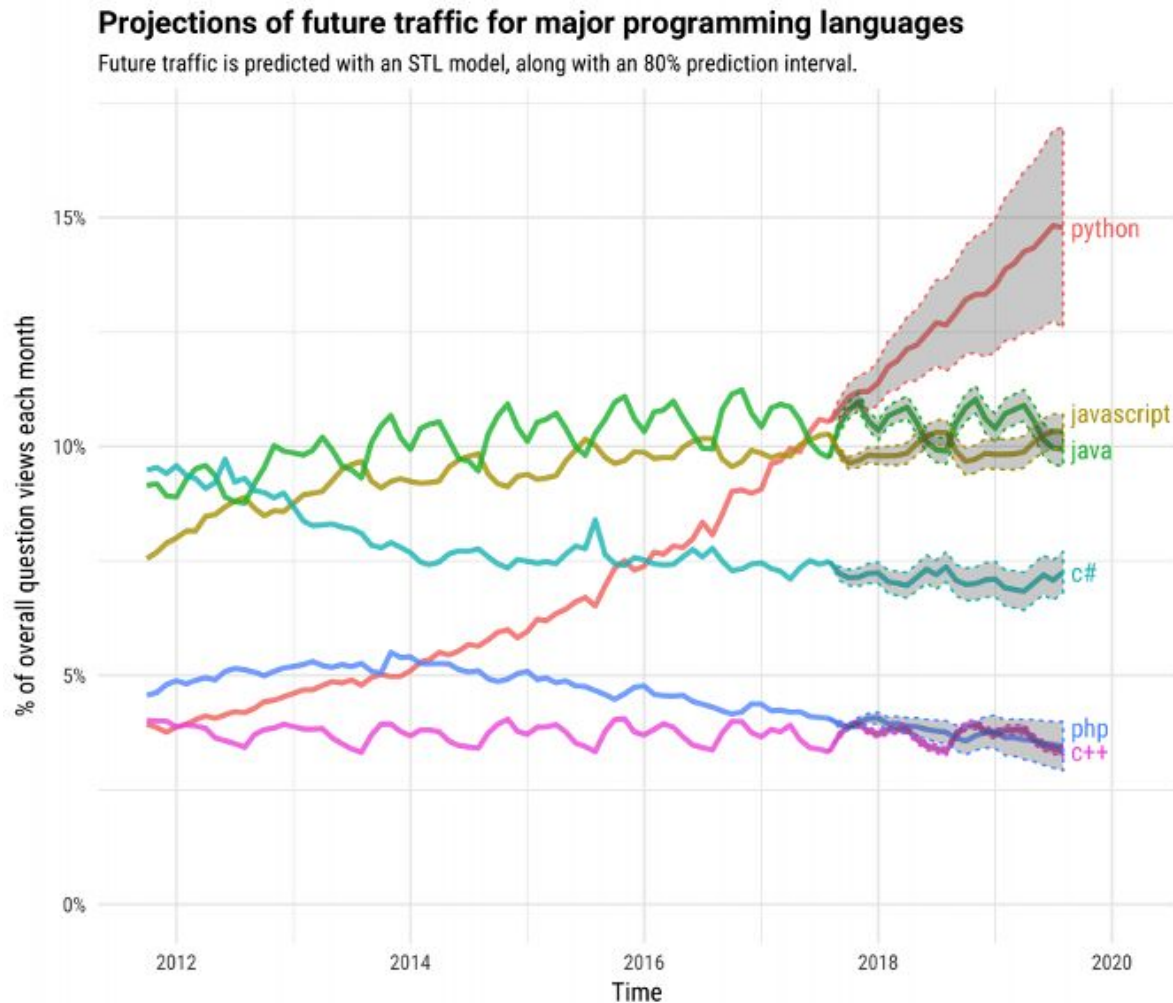
- Introduction
- OpenCV Tutorials
- OpenCV-Python Tutorials
- OpenCV.js Tutorials
- Tutorials for contrib modules
- Frequently Asked Questions
- Bibliography
- Main modules:
 - core. Core functionality
 - imgproc. Image Processing
 - imgcodecs. Image file reading and writing

Why learn Python?

Some reasons which makes Python so popular:

- Easy to code and read;
- It is Free and Open-Source;
- It is a High-Level Language;
- It is Portable;
- It has large standard library.

Why learn Python?



Source: Stack Overflow
<http://stackoverflow.com>

Why learn Python?

TIOBE Index

Mar 2019	Mar 2018	Change	Programming Language	Ratings
1	1		Java	14.880%
2	2		C	13.305%
3	4	▲	Python	8.262%
4	3	▼	C++	8.126%
5	6	▲	Visual Basic .NET	6.429%
6	5	▼	C#	3.267%
7	8	▲	JavaScript	2.426%

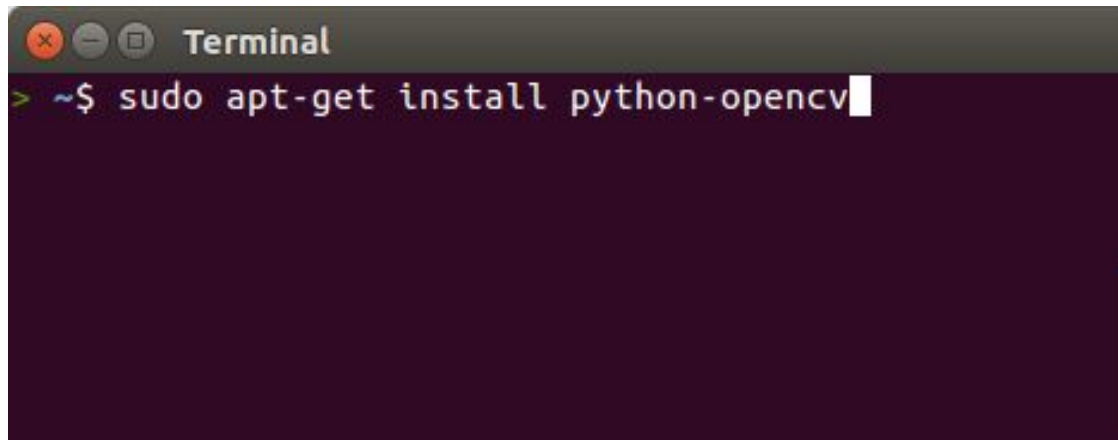
Source:

<https://www.tiobe.com/tiobe-index/>

Installing the packages

- To install the needed packages, use the following command (in Ubuntu):

```
$ sudo apt-get install python-opencv
```

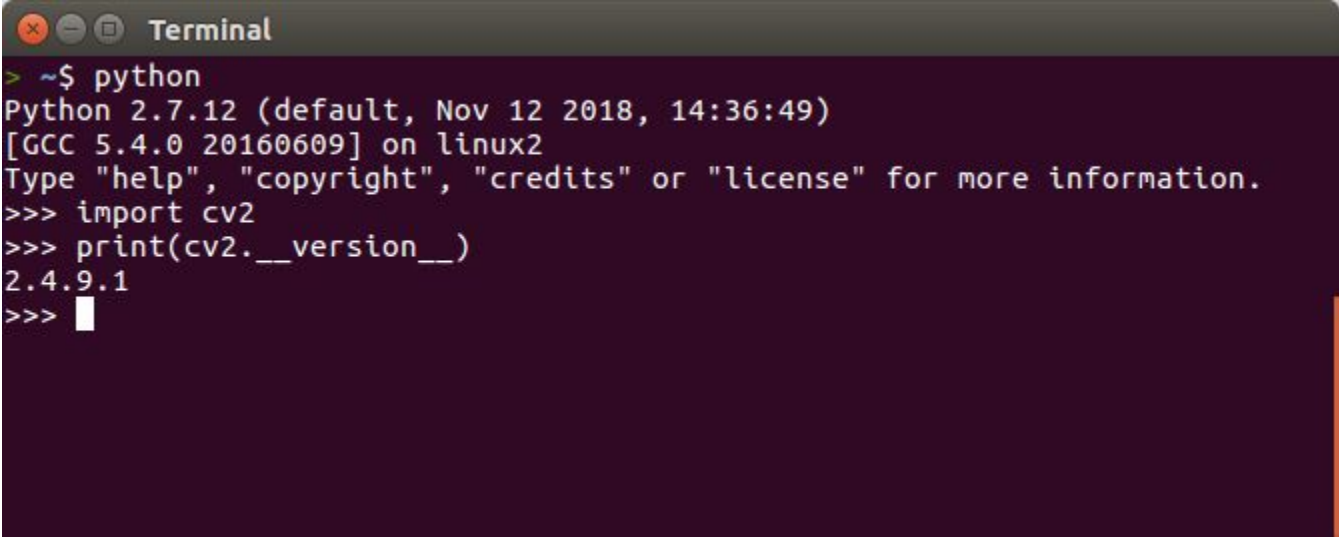
A screenshot of a Linux terminal window. The window has a title bar with the word "Terminal" and standard window control buttons (close, minimize, maximize). The terminal background is dark purple. The prompt character is a green greater-than sign (>). The command entered is "sudo apt-get install python-opencv" in white text, followed by a white cursor block.

```
Terminal  
> ~$ sudo apt-get install python-opencv
```

Checking the Installation

- To check the installation of Python and OpenCV, use the following commands in terminal:

```
$ python
import cv2
print(cv2.__version__)
```

A screenshot of a terminal window titled "Terminal". The window has a dark background with light-colored text. The terminal shows the following sequence of commands and output:

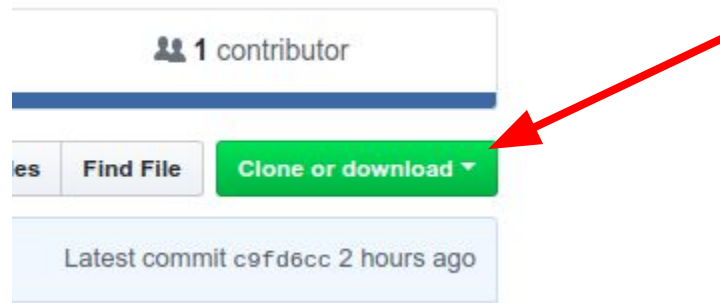
```
> ~$ python
Python 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print(cv2.__version__)
2.4.9.1
>>> █
```

Getting the examples

- All examples used in this lecture are available in the following link:

<https://github.com/cfgnunes/cv-lab>

- Access the link above and download the samples by clicking in the "download" button:



Example 1:

01_reading_image.py

```
1 import cv2
2
3 img = cv2.imread('./images/input.jpg')
4
5 cv2.imshow('Input image', img)
6
7 cv2.waitKey()
8 |
```

Example 2:

02_saving_image.py

```
1  import cv2
2
3  gray_img = cv2.imread('images/input.jpg', cv2.IMREAD_GRAYSCALE)
4
5  cv2.imshow('Grayscale', gray_img)
6
7  cv2.imwrite('images/output.jpg', gray_img)
8
9  cv2.waitKey()
10 |
```

Example 3:

03_convert_image_format.py

```
1 import cv2
2
3 img = cv2.imread('images/input.jpg')
4
5 cv2.imwrite('images/output.png', img, [cv2.IMWRITE_PNG_COMPRESSION])
6 |
```

Example 4:

04_image_scaling.py

```
1  import cv2
2
3  img = cv2.imread('images/input.jpg')
4
5  img_scaled = cv2.resize(img, (200, 200))
6
7  cv2.imshow('Scaling - Skewed Size', img_scaled)
8
9  cv2.waitKey()
10 |
```


Example 5:

05_gaussian_filter.py

```
1 import cv2
2
3 img = cv2.imread('images/input.jpg')
4 cv2.imshow('Input', img)
5
6 img_gaussian = cv2.GaussianBlur(img, (17, 17), 0)
7 cv2.imshow('Gaussian filter', img_gaussian)
8
9 cv2.waitKey()
10 |
```

Example 6:

```
1 import cv2
2
3 img = cv2.imread('images/blox.jpg')
4 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
5
6 cv2.imshow("Input image", img)
7
8 corners = cv2.goodFeaturesToTrack(
9     gray, maxCorners=30, qualityLevel=0.05, minDistance=25)
10
11 red_color = (0, 0, 255)
12
13 for item in corners:
14     x, y = item[0]
15     cv2.circle(img, (x, y), 3, red_color, -1)
16
17 cv2.imshow("Top 'k' features", img)
18
19 cv2.waitKey()
20
```

References

OpenCV-Python Tutorials.

<https://docs.opencv.org/master/>

OpenCV 3.x with Python By Example - Second Edition

<https://github.com/PacktPublishing/OpenCV-3-x-with-Python-By-Example>