

A brief introduction to OpenCV

SAHITI KORRAPATI^{1,*}

¹School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

*Corresponding authors: sakorrap@iu.edu, S17-IR-2013

techpaper-2, April 30, 2017

This paper provides a brief introduction to OpenCV. OpenCV is an open source computer vision and machine learning software library, which was originally introduced more than a decade ago by Intel. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms [1].

© 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

Keywords: OpenCV, Computer Vision, Machine Learning

<https://github.com/cloudmesh/sp17-i524/raw/master/paper2/S17-IR-2013/report.pdf>

1. INTRODUCTION

Computer Vision is the science of programming for a computer to process and understand images and videos so that the machines can detect and recognize faces, identify objects, classify human actions in videos, track moving objects, etc [2]. In order to advance vision research and disseminate vision knowledge, it is highly critical to have a library of programming functions with the optimized and portable code [3]. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception [2].

In the early days of OpenCV, the goals of the project were described as [4]:

1. Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
2. Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
3. Advance vision-based commercial applications by making portable, performance-optimized code available for free with a license that did not require code to be open or free itself.

It is an open-source BSD-licensed library that now includes several hundreds of computer vision and machine learning algorithms. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. Since the official launch of OpenCV in 1999, a number of programmers have contributed to the most recent library developments. It has C++, C, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. Though machine learning algorithms are added to OpenCV to support computer vision development, they can be used in other applications like speech

recognition and anomaly detection. CUDA and OpenCL interfaces are being actively developed right now [2].

2. PLATFORMS FOR OPENCV

OpenCV was designed to be a cross-platform tool, due to which it is completely written in C. This makes it portable to any commercial system, right from PCs, Macs, to robotic on board computers as the C compilers were stable during the initial days of development. Moreover, low-level optimization is possible through C. When it comes to Computer Vision, such optimizations may easily lead to significant speedup [5]. Since OpenCV 2.0 version, there is a C and C++ interface also, and all new packages are written in C++. However, to encourage widespread use, wrappers for popular programming languages like Python and Java have been developed and OpenCV can be used both on mobile and desktop. We look at the latest additions to OpenCV platforms [5]:

2.1. CUDA

CUDA is a parallel computing platform and application programming interface (API) model created by Nvidia. It allows software developers and software engineers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing [6].

In 2010 a new module that provides GPU acceleration was added to OpenCV. The 'gpu' module covers a significant part of the library's functionality and is still in active development. It is implemented using CUDA and therefore benefits from the CUDA ecosystem, including libraries such as NPP (NVIDIA Performance Primitives). With the addition of CUDA acceleration to OpenCV, developers can run more accurate and sophisticated OpenCV algorithms in real-time on higher-resolution images while consuming less power.

2.2. Android

Since 2010 OpenCV was ported to the Android environment, it allows to use the library in mobile applications development.

2.3. iOS

In 2012 OpenCV development team actively worked on adding extended support for iOS. Full integration is available since version 2.4.2 (2012).

2.4. OpenCL

In 2011 a new module providing OpenCL accelerations of OpenCV algorithms was added to the library. This enabled OpenCV-based code taking advantage of heterogeneous hardware, in particular utilize potential of discrete and integrated GPUs. Since version 2.4.6 (2013) the official OpenCV WinMega-Pack includes the OpenCL module.

In the 2.4 branch OpenCL-accelerated versions of functions and classes were located in a separate ocl module and in a separate namespace (*cv::ocl*), and often had different names (e.g. *cv::resize()* vs *cv::ocl::resize()* and *cv::CascadeClassifier* vs *cv::ocl::OclCascadeClassifier*) that required a separate code branch in user application code. Since OpenCV 3.0 (master branch as of 2013) the OpenCL accelerated branches transparently added to the original API functions and are used automatically when possible/sensible.

3. MODULES IN OPENCV

OpenCV was built as a modular program, which means there are several shared or static libraries to pick from. An overview of the modules present in the latest version of OpenCV (3.2.0) [7]:

1. **Core functionality** a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules.
2. **Image processing** an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
3. **video** a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
4. **calib3d** basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
5. **features2d** salient feature detectors, descriptors, and descriptor matchers.
6. **objdetect** detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
7. **highgui** an easy-to-use interface to simple UI capabilities.
8. **Video I/O** an easy-to-use interface to video capturing and video codecs.
9. **gpu** GPU-accelerated algorithms from different OpenCV modules.

10. There are other helper modules, such as FLANN and Google test wrappers, Python bindings, and others.

4. SETUP AND CONFIGURATION

Since, Python is the most popular language for Machine Learning and Computer Vision, set up and configuration of OpenCV for Python on Windows may be done as shown below.

4.1. Setting up OpenCV Python on Windows

OpenCV requires numpy and matplotlib (optional but recommended) packages to be installed in a Python 2.7 environment to be able to run successfully. After making sure that the dependencies are installed, the following steps will help to setup OpenCV for Python [8]:

1. Download latest OpenCV release from sourceforge site [9] and double-click to extract it.
2. Go to *opencv/build/python/2.7* folder.
3. Copy *cv2.pyd* to *C:/Python27/lib/site-packages*.

Now OpenCV will work as a Python library named cv2. For checking if it works, type the following code in Python environment.

```
import cv2
```

5. LICENSING

OpenCV is an open source software, and OpenCV allows redistribution in terms of source or binary forms, with or without modifications. All re distributions should bear the copyright information provided at the OpenCV licensing page [10]. The source code is available on Github [11].

6. USE CASE

OpenCV library can be used for image recognition technology. There are numerous applications of image recognition, like facial recognition for security purposes, facial tagging in cameras, image and video processing in self driving cars. For instance, take a data set containing pictures taken from a forest surveillance cameras, we can process the images to find out the number of animals at any given point of time. This will be helpful in keeping track of endangered animals and to keep a check on animal poaching.

A sample code of loading an image of a watch for the purpose of image recognition using OpenCV in Python is demonstrated below:

Here, we use numpy and matplotlib libraries which we installed earlier [12].

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('watch.jpg', cv2.IMREAD_GRAYSCALE)
cv2.imshow('image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Images can also be displayed for which sample code is shown below [12]:

```
plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
plt.xticks([], plt.yticks([])) # to hide tick values on X and Y
plt.plot([200,300,400],[100,200,300], 'c', linewidth=5)
plt.show()
```

7. USEFUL RESOURCES

The OpenCV website has detailed and structured documentation for its modules, for all operating systems and platforms. Further reading is suggested based on the requirements of OS, platform and language by clicking on the version of OpenCV that is currently in use [13].

8. CONCLUSION

OpenCV is one of the leading Computer Vision software which offers various modules for image and video processing and a library of various Machine learning algorithms. OpenCV is an ever growing platform owing to its open source nature. OpenCV is versatile in terms of its operating systems, platforms and programming languages, which makes it an even more popular tool. The open source Robot Operating System (ROS) uses OpenCV as its primary image processing software. This expands the usage of OpenCv even further.

ACKNOWLEDGEMENTS

The author thanks Professor Gregor Von Laszewski and all the AIs of big data class for the guidance and technical support.

REFERENCES

- [1] OpenCV team, "Opencv library," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <http://opencv.org/>
- [2] OpenCV team, "About - opencv library," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <http://opencv.org/about.html>
- [3] I. Culjak, D. Abram, T. Pribanic, H. Dzap, and M. Cifrek, "A brief introduction to opencv," in *2012 Proceedings of the 35th International Convention MIPRO*, May 2012, pp. 1725–1730.
- [4] Wikipedia, "History," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <https://en.wikipedia.org/wiki/OpenCV>
- [5] OpenCV team, "Platforms - opencv library," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <http://opencv.org/platforms/>
- [6] Wikipedia, "Cuda," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <https://en.wikipedia.org/wiki/CUDA>
- [7] OpenCV team, "Opencv: Introduction," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <http://docs.opencv.org/3.2.0/d1/dfb/intro.html>
- [8] OpenCV team, "Install opencv-python in windows," Web Page, 2017, accessed mar-30-2017. [Online]. Available: http://docs.opencv.org/3.2.0/d5/de5/tutorial_py_setup_in_windows.html
- [9] Slashdot Media, "Opencv," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <https://sourceforge.net/projects/opencvlibrary/files/>
- [10] OpenCV team, "License - opencv," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <http://opencv.org/license.html>
- [11] A. Pavlenko, "Opencv," Github Repository, 2017, accessed mar-30-2017. [Online]. Available: <https://github.com/opencv>
- [12] PythonProgramming, "Opencv with python intro and loading images tutorial," Web Page, 2017, accessed apr-16-2017. [Online]. Available: <https://pythonprogramming.net/loading-images-python-opencv-tutorial/>
- [13] OpenCV team, "Opencv documentation index," Web Page, 2017, accessed mar-30-2017. [Online]. Available: <http://docs.opencv.org/>

AUTHOR BIOGRAPHIES

Sahiti Korrapati is pursuing her MSc in Data Science from Indiana University Bloomington