

Image Handling Using OpenCV

어플리케이션 만들기

정현석

OpenCV란?

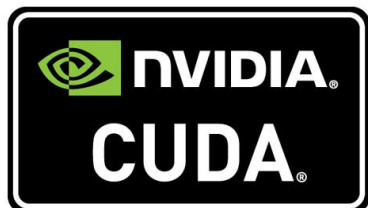
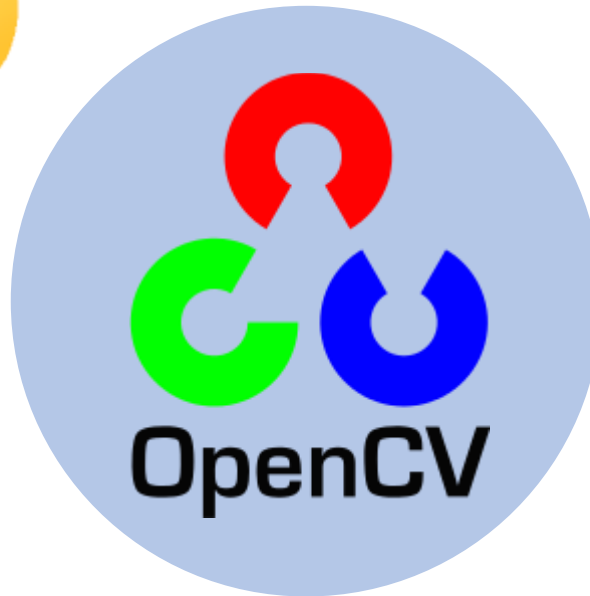
- OpenCV (Open Source Computer Vision)
- C++ 기반의 라이브러리
- 다양한 영상처리 알고리즘을 쉽게 사용가능
- 다양한 Platform과 Language wrappers를 지원하여 많은 Platform과 많은 Language에서 사용가능
- 현재 OpenCV 3.1버전 까지 배포됨
- 공식홈페이지 - [<http://opencv.org/>]
- Source - [<https://github.com/opencv/opencv>]



OpenCV



OpenCL



OpenCV 설치 - 1

■ 패키지를 통한 설치방법

- 기본 OpenCV 라이브러리 패키지 설치해주는 방법
- 현재 2.4.9.1 버전이 설치된다.
- 3rdParty 라이브러리를 연결하여 사용하는 라이브러리 사용할 때 설정 문제가 될 수 있다.
- Open CV 패키지 설치
 - `$ sudo apt-get install libopencv-dev`

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt-get install libopencv-dev  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  autopoint debhelper gettext gir1.2-gdkpixbuf-2.0 gir1.2-gtk-2.0  
  intltool-debian libasprintf-dev libatk1.0-dev libavcodec-dev libavcodec56  
  libavformat-dev libavformat56 libavresample-dev libavresample2 libavutil-dev  
  libavutil54 libcairo-script-interpreter2 libcairo2-dev libcbv-dev  
  libcvaux-dev libdc1394-22-dev libfontconfig1-dev libgdk-pixbuf2.0-0  
  libgdk-pixbuf2.0-common libgdk-pixbuf2.0-dev libgettextpo-dev libgettextpo0  
  libglib2.0-dev libgtk2.0-dev libharfbuzz-dev libharfbuzz-gobject0  
  libhighgui-dev libice-dev liblmbase-dev libjasper-dev libjbig-dev  
  libjpeg-dev libjpeg62-turbo-dev liblzma-dev libmail-sendmail-perl  
  libopencv-calib3d-dev libopencv-contrib-dev libopencv-core-dev  
  libopencv-features2d-dev libopencv-flann-dev libopencv-gpu-dev  
  libopencv-gpu2.4 libopencv-highgui-dev libopencv-imgproc-dev  
  libopencv-legacy-dev libopencv-ml-dev libopencv-objdetect-dev  
  libopencv-ocl-dev libopencv-ocl2.4 libopencv-photo-dev libopencv-photo2.4  
  libopencv-stitching-dev libopencv-stitching2.4 libopencv-superres-dev  
  libopencv-superres2.4 libopencv-ts-dev libopencv-ts2.4 libopencv-video-dev  
  libopencv-video2.4 libopencv-video2.4-jni libopencv-video2.4-jni  
  libopencv2.4-jni libopenexr-dev libpangol1.0-dev libpcre3-dev libpcrecpp0  
  libpixman-1-dev libpthread-stubs0-dev libraw1394-dev libraw1394-tools
```

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ pkg-config --modversion opencv  
2.4.9.1  
pi@raspberrypi:~ $
```

OpenCV 설치 - 2

■ 컴파일을 통한 설치방법

- 원하는 버전 설치 가능
- 3rdParty 패키지 설정 및 Build설정 후 컴파일 가능

■ 의존성 패키지 설치 (필요한 것들만 설치)

- `$ sudo apt-get install build-essential` // 필수
- `$ sudo apt-get install libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev` // OpenCV 빌드와 실행시 필요
- `$ sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev` // 필요하면 설치

■ OpenCV Source Download

- `$ cd ~/<my_working_directory>`
- `$ git clone https://github.com/Itseez/opencv.git`

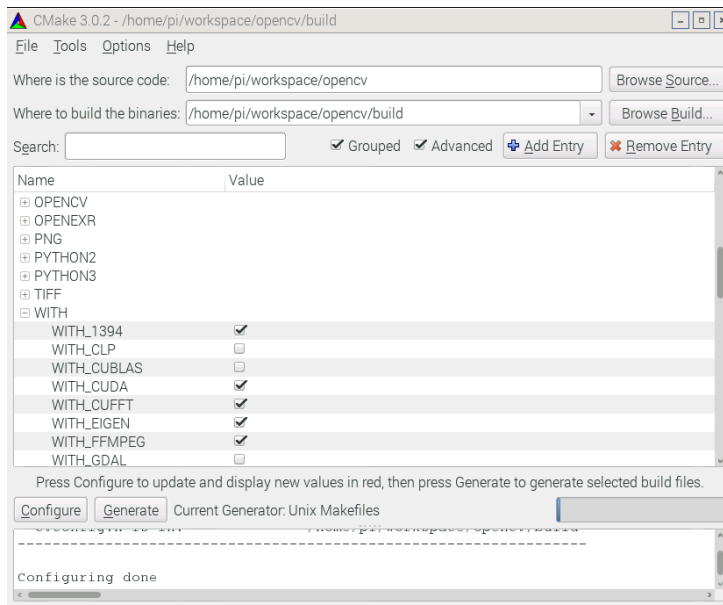
OpenCV 설치 - 2

■ 컴파일을 통한 설치방법

■ OpenCV Build 설정 (cmake)

- `$ cd ~/opencv`
- `$ mkdir release`
- `$ cd release`
- `$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..`

■ OpenCV Build 설정 (cmake-gui)



OpenCV 설치 - 2

■ 컴파일을 통한 설치방법

- OpenCV Build 설정 (cmake)
 - \$ make -j4
 - \$ sudo make install

```
pi@raspberrypi: ~/workspace/opencv/build
File Edit Tabs Help
pi@raspberrypi:~/workspace/opencv/build $ make -j4
[ 0%] [ 0%] [ 0%] Scanning dependencies of target libwebp
Generating opencv_core_pch_dephelp.cxx
Generating opencv_ts_pch_dephelp.cxx
Generating opencv_imgproc_pch_dephelp.cxx
Scanning dependencies of target opencv_ts_pch_dephelp
Scanning dependencies of target opencv_core_pch_dephelp
Scanning dependencies of target opencv_imgproc_pch_dephelp
[ 0%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/dec/alpha.c.o
[ 0%] [ 1%] [ 1%] Building CXX object modules/imgproc/CMakeFiles/opencv_imgproc_pch_dephelp.dir/opencv_imgproc_pch_dephelp.cxx.o
Building CXX object modules/core/CMakeFiles/opencv_core_pch_dephelp.dir/opencv_core_pch_dephelp.cxx.o
Building CXX object modules/ts/CMakeFiles/opencv_ts_pch_dephelp.dir/opencv_ts_pch_dephelp.cxx.o
[ 1%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/dec/frame.c.o
[ 1%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/dec/vp8.c.o
Linking CXX static library ../lib/libopencv_imgproc_pch_dephelp.a
[ 1%] Built target opencv_imgproc_pch_dephelp
[ 1%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/dec/tree.c.o
Linking CXX static library ../lib/libopencv_core_pch_dephelp.a
[ 1%] Built target opencv_core_pch_dephelp
[ 1%] Generating opencv_imgcodecs_pch_dephelp.cxx
Scanning dependencies of target opencv_imgcodecs_pch_dephelp
```

```
pi@raspberrypi: ~
File Edit Tabs Help
1 [|||||] [99.5%] Tasks: 76, 66 thr; 5 running
2 [|||||] [98.1%] Load average: 3.01 1.23 0.50
3 [|||||] [100.0%] Uptime: 01:58:41
4 [|||||] [97.7%]
Mem [|||||] [416/925MB]
Swp [|||||] [0/99MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
20418 pi 20 0 49960 39700 12028 R 98.5 4.2 0:03.14 /usr/lib/gcc/arm-
20441 pi 20 0 47912 37508 12056 R 98.5 4.0 0:02.48 /usr/lib/gcc/arm-
20434 pi 20 0 27032 21284 13332 R 95.7 2.2 0:02.66 /usr/lib/gcc/arm-
566 root 20 0 166M 73988 36096 S 6.0 7.8 1:14.25 /usr/bin/X :0 -se
19555 pi 20 0 5200 2852 2328 R 2.8 0.3 0:03.08 http
866 pi 20 0 47956 14484 10876 S 0.9 1.5 0:29.33 lxterminal
20381 pi 20 0 14380 6064 5404 S 0.9 0.6 0:00.06 import ./5.png
670 pi 20 0 95460 13116 8728 S 0.5 1.4 0:28.67 lxpanel --profile
426 nobody 20 0 2292 1436 1316 S 0.5 0.2 0:02.28 /usr/sbin/thd --d
19727 pi 20 0 4048 2156 1624 S 0.5 0.2 0:00.20 make -f CMakeFile
20402 pi 20 0 3756 1816 1620 S 0.0 0.2 0:00.02 make -f modules/p
20435 pi 20 0 3756 1884 1688 S 0.0 0.2 0:00.02 make -f modules/p
16330 pi 20 0 630M 224M 48652 S 0.0 24.2 3:25.48 /usr/bin/x-www-br
19785 pi 20 0 3884 2148 1708 S 0.0 0.2 0:00.15 make -f 3rdparty/
F1Help F2Setup F3Search F4Filter F5Free F6SortBy F7Nice F8Nice + F9Kill F10Quit
```

OpenCV 설치 - Tegra

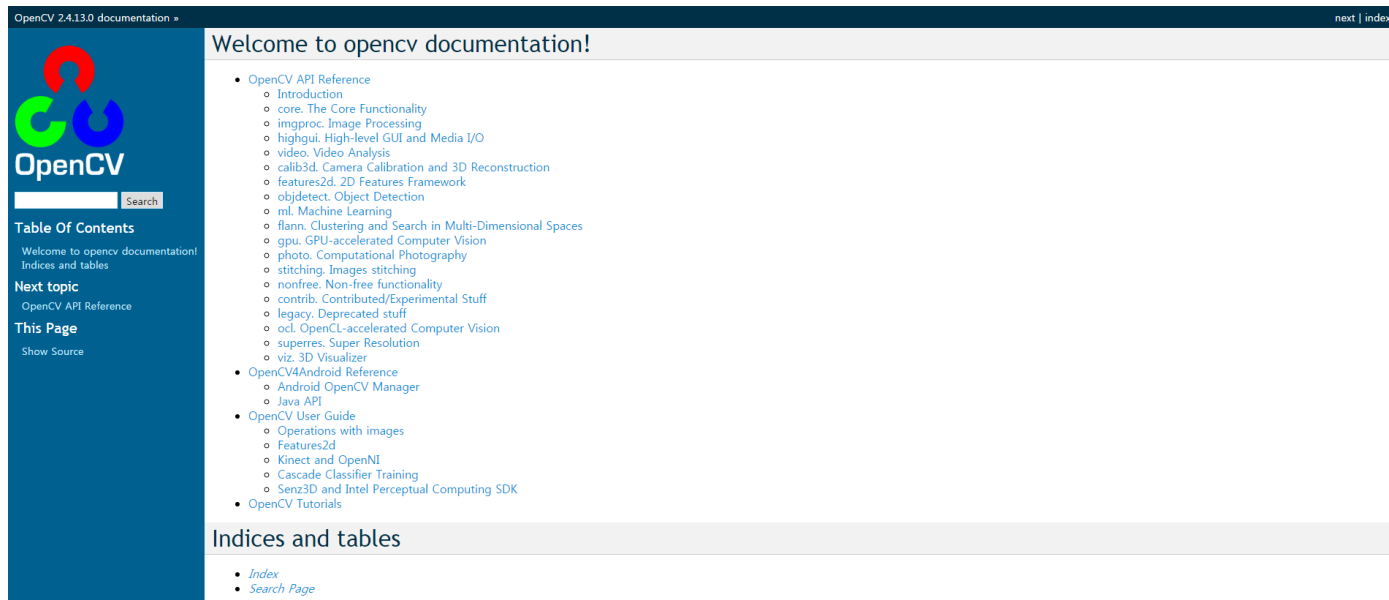
- **Nvidia 에서 Tegra에 대한 OpenCV 지원**

- GPU와 Tegra CPU (NEON SIMD + Multi-threading) 최적화를 통한 OpenCV
- <https://developer.nvidia.com/embedded/downloads> 에서 상황에 맞는 OpenCV4Tegra 설치 파일 다운로드
 - `sudo dpkg -i libopencv4tegra_2.4.x.x_armhf.deb`
 - `sudo dpkg -i libopencv4tegra-dev_2.4.x.x_armhf.deb`

OpenCV 사용법

■ 버전별 자세한 Documentation 제공

- <http://docs.opencv.org/>



OpenCV 2.4.13.0 documentation » next | index

Welcome to opencv documentation!

- OpenCV API Reference
 - Introduction
 - core, The Core Functionality
 - imgproc, Image Processing
 - highgui, High-level GUI and Media I/O
 - video, Video Analysis
 - calib3d, Camera Calibration and 3D Reconstruction
 - features2d, 2D Features Framework
 - objdetect, Object Detection
 - ml, Machine Learning
 - flann, Clustering and Search in Multi-Dimensional Spaces
 - gpu, GPU-accelerated Computer Vision
 - photo, Computational Photography
 - stitching, Images stitching
 - nonfree, Non-free functionality
 - contrib, Contributed/Experimental Stuff
 - legacy, Deprecated stuff
 - ocl, OpenCL-accelerated Computer Vision
 - superres, Super Resolution
 - viz, 3D Visualizer
- OpenCV4Android Reference
 - Android OpenCV Manager
 - Java API
- OpenCV User Guide
 - Operations with images
 - Features2d
 - Kinect and OpenNI
 - Cascade Classifier Training
 - Senz3D and Intel Perceptual Computing SDK
- OpenCV Tutorials

Indices and tables

- [Index](#)
- [Search Page](#)

OpenCV - 기본 자료형

■ 이미지 자료형

■ Mat

- C++ 기반의 API 사용시 사용되는 이미지의 정보가 저장되는 클래스

■ IplImage

- C기반의 API 사용시 사용되는 이미지의 정보가 저장되는 구조체

IplImage	Mat	description
nChannels	Mat::channels()	이미지 채널의 개수
width	Mat::cols	가로 픽셀의 개수
height	Mat::rows	세로 픽셀의 개수
imageData	Mat::data	실제 이미지의 첫 픽셀에 대한 포인터
widthStep	Mat::step1()	가로로 된 한 라인의 바이트
imageSize		width * height * nChannels
	Mat::total()	cols * rows

OpenCV – 이미지 로드

- C : `IplImage* cvLoadImage(const char* filename, int iscolor=CV_LOAD_IMAGE_COLOR)`
- C++ : `Mat imread(const string& filename, int flags=CV_LOAD_IMAGE_COLOR)`

CV_LOAD_IMAGE_ANYCOLOR	원본 이미지의 속성 그대로 Load
CV_LOAD_IMAGE_COLOR	원본 이미지의 속성을 3채널 이미지로 변환하여 Load
CV_LOAD_IMAGE_GRAYSCALE	원본 이미지의 속성을 1채널 이미지로 변환하여 Load

OpenCV – 이미지 캡처

- 캡처를 사용하기 위해 디바이스로부터 정보를 저장하는 클래스 또는 구조체
 - C : `CvCapture* cvCaptureFromCAM(int device)`
 - C++ : `VideoCapture::VideoCapture(int device)`
- 캡처를 하기 위해 디바이스로부터 프레임을 잡는다.
 - C : `int cvGrabFrame(CvCapture* capture)`
 - C++ : `bool VideoCapture::grab()`
- 해당 프레임을 이미지 형태로 반환한다.
 - C : `IplImage* cvRetrieveFrame(CvCapture* capture, int streamIdx=0)`
 - C++ : `bool VideoCapture::retrieve(Mat& image, int channel=0)`

OpenCV – 이미지 저장

- 캡처를 사용하기 위해 디바이스로부터 정보를 저장하는 클래스 또는 구조체
 - C : `int cvSaveImage(const char* filename, const CvArr* image, const int* params=0)`
 - C++ : `bool imwrite(const string& filename, InputArray img, const vector<int>& params=vector<int>())`

filename	저장되는 파일의 경로와 이름
image or img	저장되는 이미지
params	저장되는 이미지 포맷에 따른 품질 또는 압축에 대한 정보 Default로 0값이 된다.

OpenCV - 예제

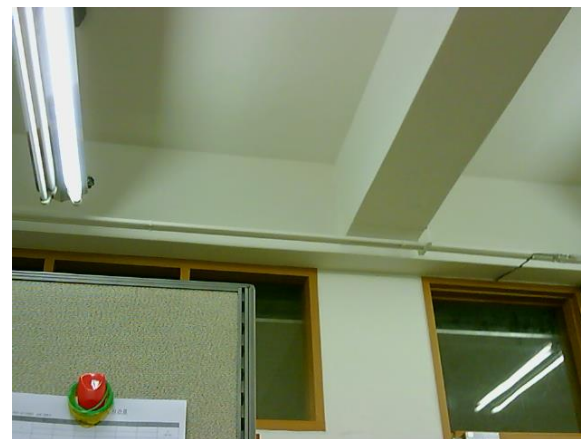
- 카메라로부터 캡처 후 png 파일로 저장

```
#include <opencv2/opencv.hpp>

using namespace cv;

int main()
{
    VideoCapture cap(0); // Video0 카메라
    Mat img;
    if(!cap.isOpened()) // 정상적으로 열렸는지 확인
        return -1;

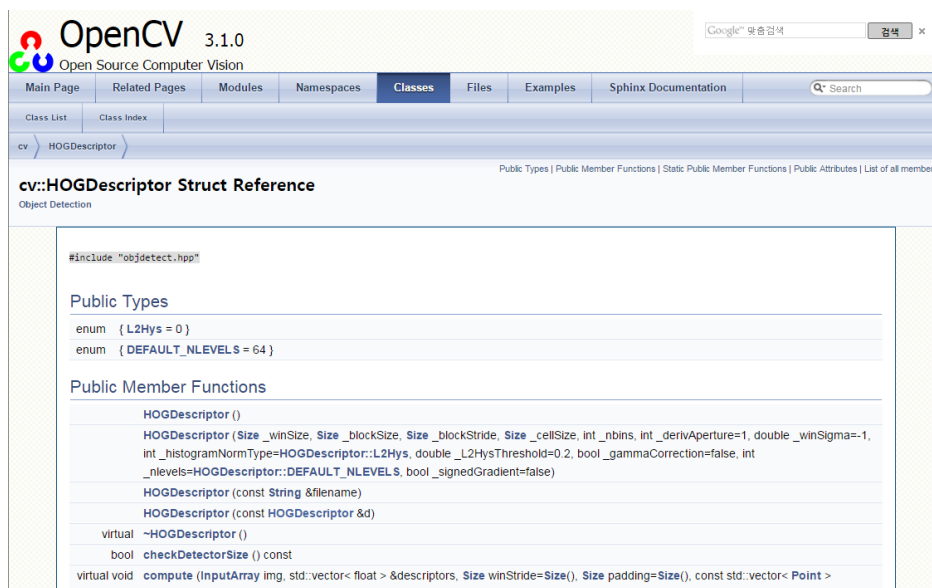
    cap.grab(); // video frame 선택
    cap.retrieve(img); // frame에서 이미지로 반환
    imwrite("./capture.png", img); // 이미지 저장
    return 0;
}
```



[capture.png]

OpenCV – Object detect

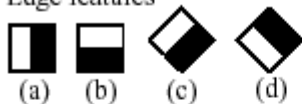
- OpenCV 특정 Feature에 대해 검출 할 수 있는 다양한 알고리즘이 존재한다.
- Haar, LBP, HOG, CT등의 알고리즘에 대한 라이브러리 존재
- e.g.) 번호판 검출, 얼굴 검출, 손 검출, 보행자 검출



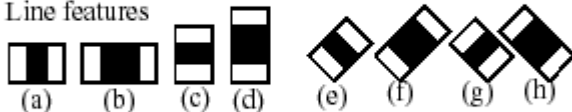
OpenCV - Haar 알고리즘

- 특정 Feature는 명암의 차이가 있다는 이론
 - 이미지에서 영역과 영역의 밝기차를 이용하여 Feature로서의 특징을 찾아내는 방법
 - 사람의 얼굴, 눈, 코, 입 등은 특징적인 밝기 차이가 있기에 이를 통해 Feature를 찾아 낼 수 있다.

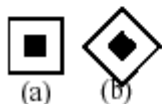
1. Edge features



2. Line features



3. Center-surround features



OpenCV - Haar data

- OpenCV에서 Machine Learning을 통해 Training된 Feature별 Haar data를 xml 형태로 제공한다.
- 경로 - (opencv_source_directory)/data/haarcascades/
- [<https://github.com/opencv/opencv/tree/master/data/haarcascades>]

haarcascade_eye.xml

haarcascade_eye_tree_eyeglasses.xml

haarcascade_frontalcatface.xml

haarcascade_frontalcatface_extended....

haarcascade_frontalface_alt.xml

haarcascade_frontalface_alt2.xml

haarcascade_frontalface_alt_tree.xml

haarcascade_frontalface_default.xml

haarcascade_fullbody.xml

OpenCV - Haar training

- 추가적으로 Training 할 수 있게 제공
- 하지만 Training시 Sample 이미지가 많아야 정확해지고 Positive Sample과 Negative Sample 모두 필요
- 경로 - (opencv_source_directory)/app/haartraining/
training 소스코드 존재하므로 이를 컴파일하여 사용가능

OpenCV - face detect 예제 흐름

Haar Data Load

분류할 이미지 Read

이미지 Gray Scale

Haar Data를 통해
Haar 알고리즘으로 이미지 탐색

탐색된 결과 확인

OpenCV - face detect 예제

// haar data load

```
const String haar_face_cascade = "./haarcascade_frontalface_alt.xml";  
CascadeClassifier face;  
face.load(haar_face_cascade);
```

// Image Load & Gray Scale

```
Mat img = imread(resource_dir + filename);  
Mat gray;  
cvtColor(img, gray, CV_RGB2GRAY);
```

// haar detecting

```
vector<Rect> face_pos;  
face.detectMultiScale(gray, face_pos, 1.1, 3, 0 | CV_HAAR_SCALE_IMAGE,  
Size(10,10));
```

OpenCV - face detect full source

감사합니다.

Q&A