Speak, OpenCV and ROS

ROS INTRODUCTION COURSE

KU

KASETSART
UNIVERSITY

KASETSART
UNIVERSITY

OpenCV

Image processing in Python?

Python provides lots of libraries for image processing, including —

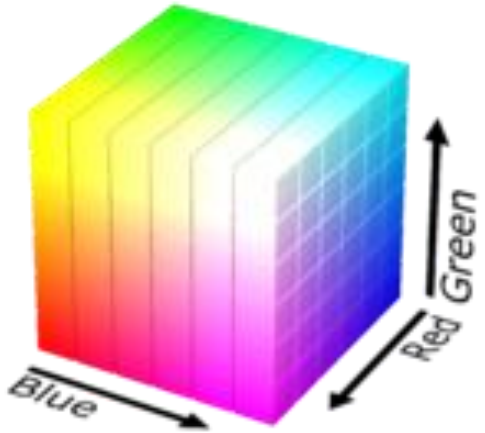
OpenCV — Image processing library mainly focused on real-time computer vision with application in wide-range of areas like 2D and 3D feature toolkits, facial & gesture recognition, Human-computer interaction, Mobile robotics, Object identification and others.

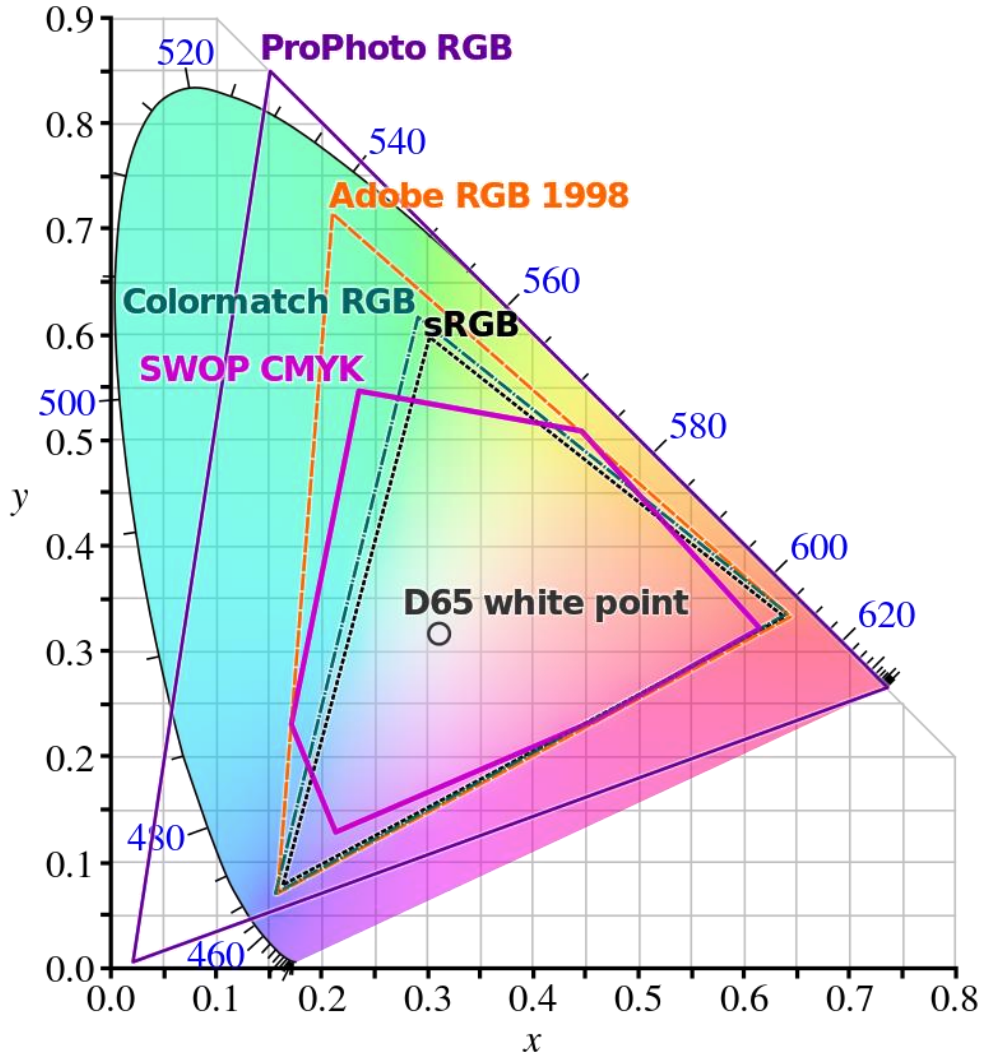
- **Numpy and Scipy libraries** — For image manipuation and processing.
- **Sckikit** — Provides lots of alogrithms for image processing.
- **Python Imaging Library (PIL)** — To perform basic operations on images like create thumnails, resize, rotation, convert between different file formats etc.

```
sudo apt-get install python-pip python3-pip python-numpy
pip install pillow    or    pip3 install pillow
sudo apt-get install python-opencv or pip3 install opencv-python
```

Image Color System

A **color space** is a specific organization of colors. In combination with physical device profiling, it allows for reproducible representations of color, in both analog and digital representations.





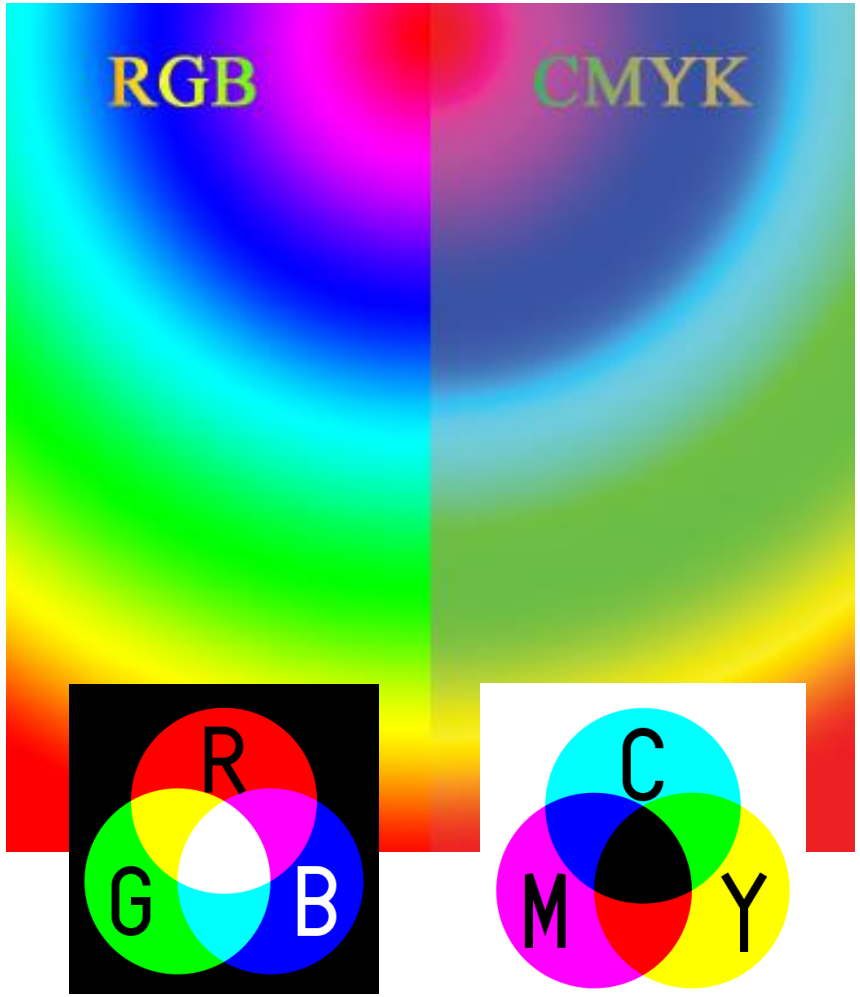
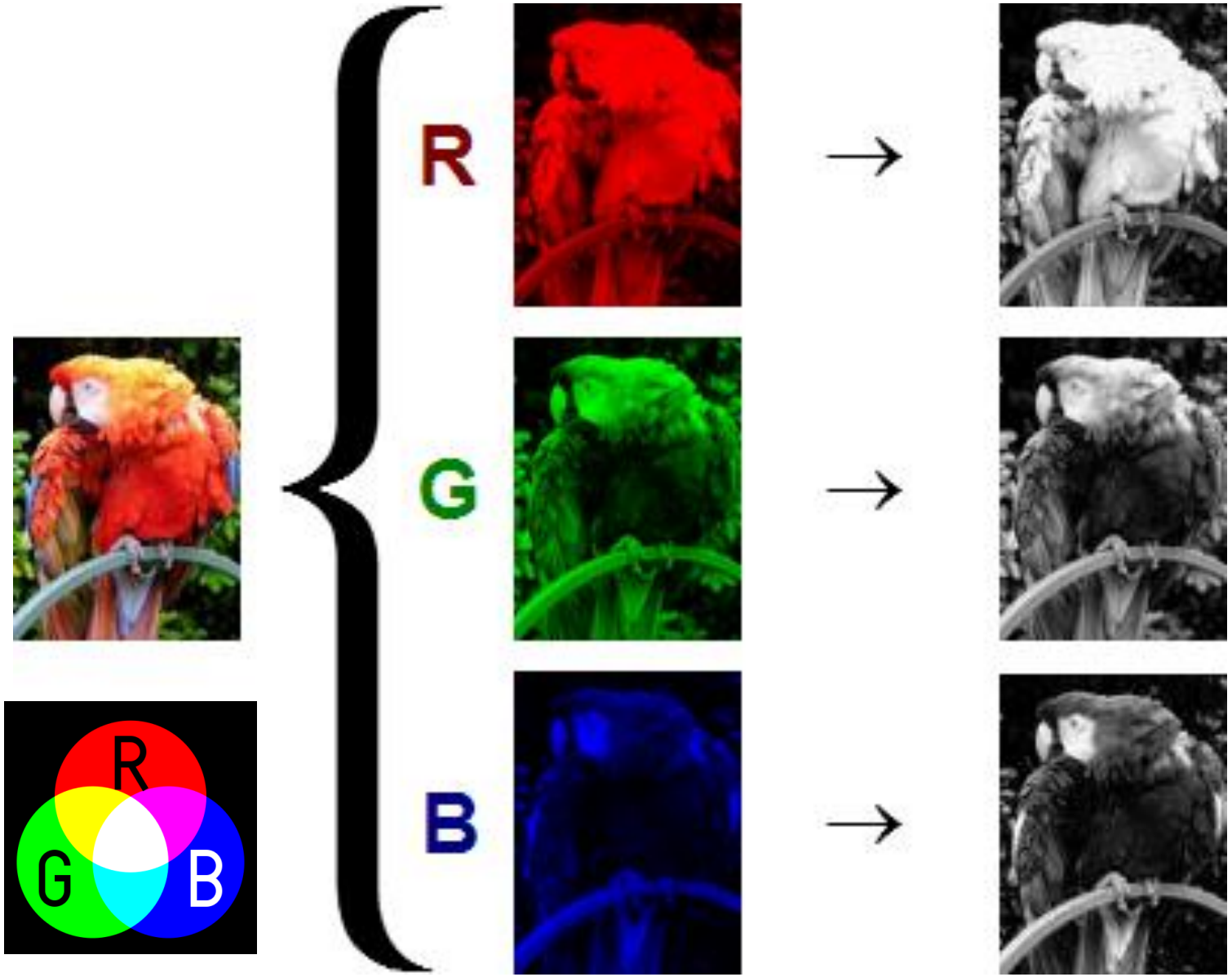


Image Color System



RGB color format & calculation

RGB code has 24 bits format (bits 0..23):

RED[7:0]								GREEN[7:0]								BLUE[7:0]							
23							16	15							8	7							0

```
RGB = (R*65536)+(G*256)+B , (when R is RED, G is GREEN and B is BLUE)
```

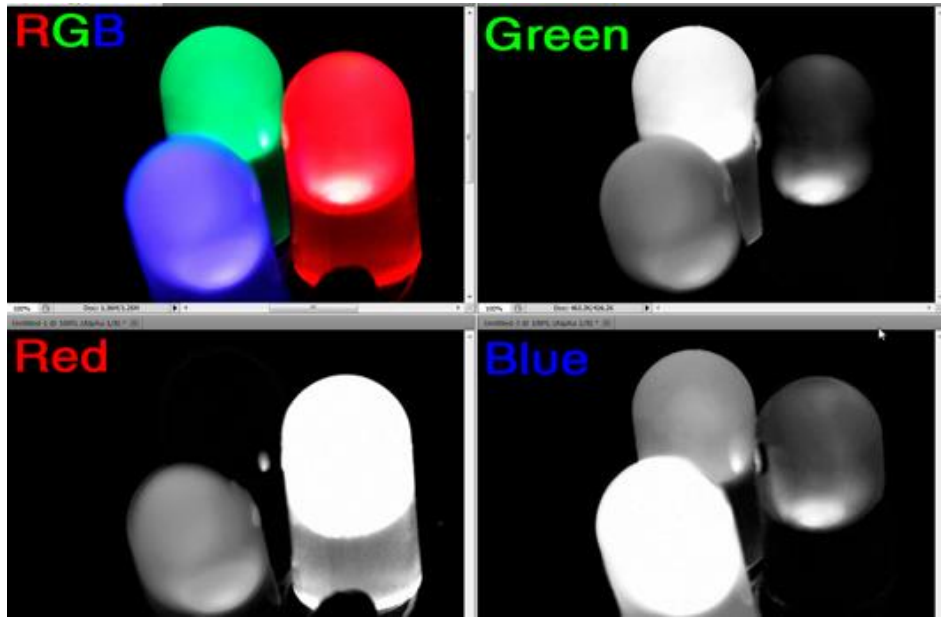
Calculation examples

White RGB Color

```
White RGB code = 255*65536+255*256+255 = #FFFFFF
```

Blue RGB Color

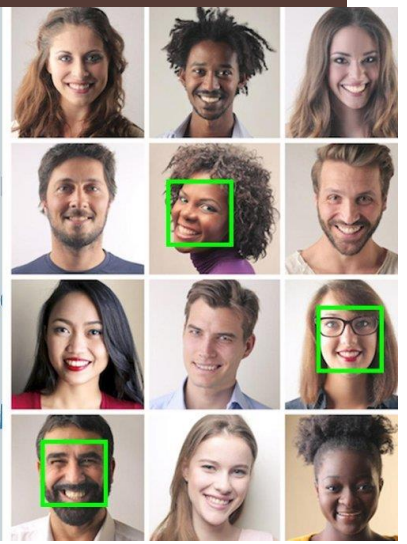
```
Blue RGB code = 0*65536+0*256+255 = #0000FF
```



What is OpenCV ?

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

```
img = cv2.imread("faces.j
gray = cv2.cvtColor(img,
path = "haarcascade_eye.x

eye_cascade = cv2.Cascade

eyes = eye_cascade.detect
print(len(eyes))
```



KASETSART UNIVERSITY

OpenCV Basic:

NumPy

NumPy.org

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

```
>>> A = np.array( [[1,1],
...               [0,1]] )
>>> B = np.array( [[2,0],
...               [3,4]] )
>>> A * B                          # elementwise product
array([[2, 0],
       [0, 4]])
>>> A @ B                          # matrix product
array([[5, 4],
       [3, 4]])
>>> A.dot(B)                       # another matrix product
array([[5, 4],
       [3, 4]])
```

OpenCV Basic:

opencv-python 4.1.1.26

Latest version

```
pip install opencv-python
```

Last released: Sep 2, 2019

Wrapper package for OpenCV python bindings.

OpenCV use BRG rather than RGB

Navigation

- Project description
- Release history
- Download files

Project description

downloads 47M

OpenCV on Wheels

Unofficial pre-built OpenCV packages for Python.

opencv-contrib-python==4.1.2.30

opencv-python==4.1.1.26

```
student@student-VirtualBox:~$ python
Python 2.7.12 (default, Aug 22 2019, 16:36:40)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'2.4.9.1'
>>> exit()
student@student-VirtualBox:~$ python3
Python 3.5.2 (default, Jul 10 2019, 11:58:48)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.1.1'
```

Cannot use pip to install opencv-python for python 2.7 due to error of numpy package

If need newer version of opencv in python 2.7 need to build from source process

OpenCV Basic:

cv2.VideoCapture(0): Means first camera or webcam.

cv2.VideoCapture(1): Means second camera or webcam.

cv2.VideoCapture("file name.mp4"): Means video file

cap = cv2.VideoCapture(0) --> ret, img =cap.read()

cv2.imread(image path)

cv2.imwrite(filename, img)

cv2.imshow("window name", image)

cv2.waitkey(0)

cv2.destroyAllWindows()

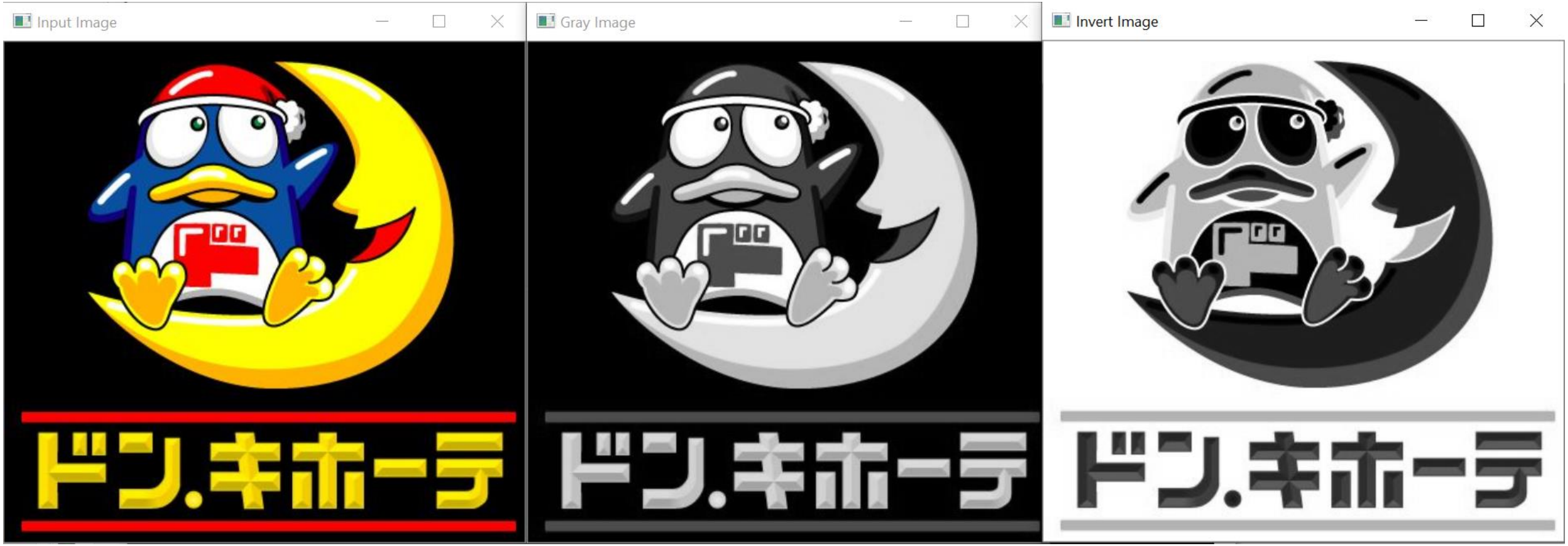
https://www.geeksforgeeks.org/opencv-python-tutorial/

OpenCV Basic:

```python
import cv2
import argparse
#Usage python main_img.py --file Scenic009smaller.bmp #
img_location = 'DonKi.jpg'
parser = argparse.ArgumentParser(description='input image name')
parser.add_argument('--file', help="location of the image file")
args = parser.parse_args()
if args.file:
    img_location = args.file
print('<Image File Location>')
print(img_location)
img = cv2.imread(img_location)
print('File loaded')
h,w,c = img.shape
print('----------------------------------------------------')
print("Image Information")
print("Dimension w x h = %s x %s" % (w,h))
print("Channels = %s" %(c))
print("Total Pixels of input image = %s pixels which %s channels" % (w*h,c))
print('----------------------------------------------------')
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
Invert_img = 255-gray_img
cv2.imshow('Input Image',img)
cv2.imshow('Gray Image',gray_img)
cv2.imshow('Invert Image',Invert_img)
cv2.imwrite('Invert_img.bmp',Invert_img)
print('File Saved')
cv2.waitKey(0)
```

```
<Image File Location>
DonKi.jpg
File loaded
----------------------------------------------------
Image Information
Dimension w x h = 421 x 404
Channels = 3
Total Pixels of input image = 170084 pixels which 3 channels
----------------------------------------------------
File Saved
```

OpenCV Basic:



OpenCV Basic:

```
import cv2
import argparse
#Usage python main_img.py --file Scenic009smaller.bmp #
img_location = 'DonKi.jpg'
parser = argparse.ArgumentParser(description='input image name')
parser.add_argument('--file', help="location of the image file")
args = parser.parse_args()
if args.file:
    img_location = args.file
print('<Image File Location>')
print(img_location)
img = cv2.imread(img_location)
print('File loaded')
h,w,c = img.shape
print('-----------------------------------------------')
print("Image Information")
print("Dimension w x h = %s x %s" % (w,h))
print("Channels = %s" %(c))
print("Total Pixels of input image = %s pixels which %s channels" % (w*h,c))
print('-----------------------------------------------')
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
Invert_img = 255-gray_img
cv2.imshow('Input Image',img)
cv2.imshow('Gray Image',gray_img)
cv2.imshow('Invert Image',Invert_img)
cv2.imwrite('Invert_img.bmp',Invert_img)
print('File Saved')
cv2.waitKey(0)
```

Pass file name

Import image file (BGR) to image (numpy) array

Get image information (height, Width, Channel)

Change BGR image to Gray Scale image

Invert Gray Scale image

Display image

Save image array to file

Wait for user press any key

Image Filter (2D convolution)

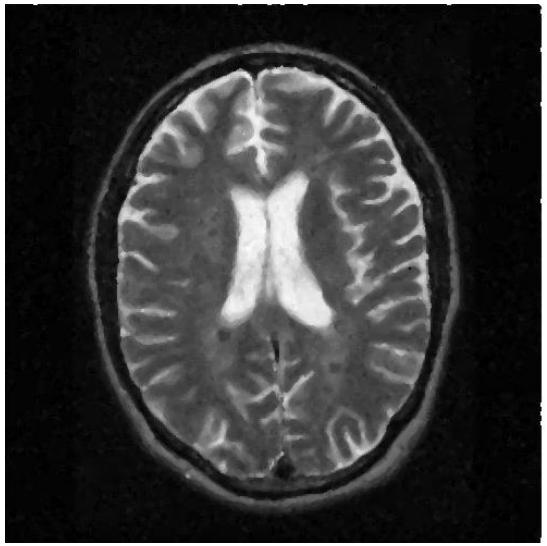
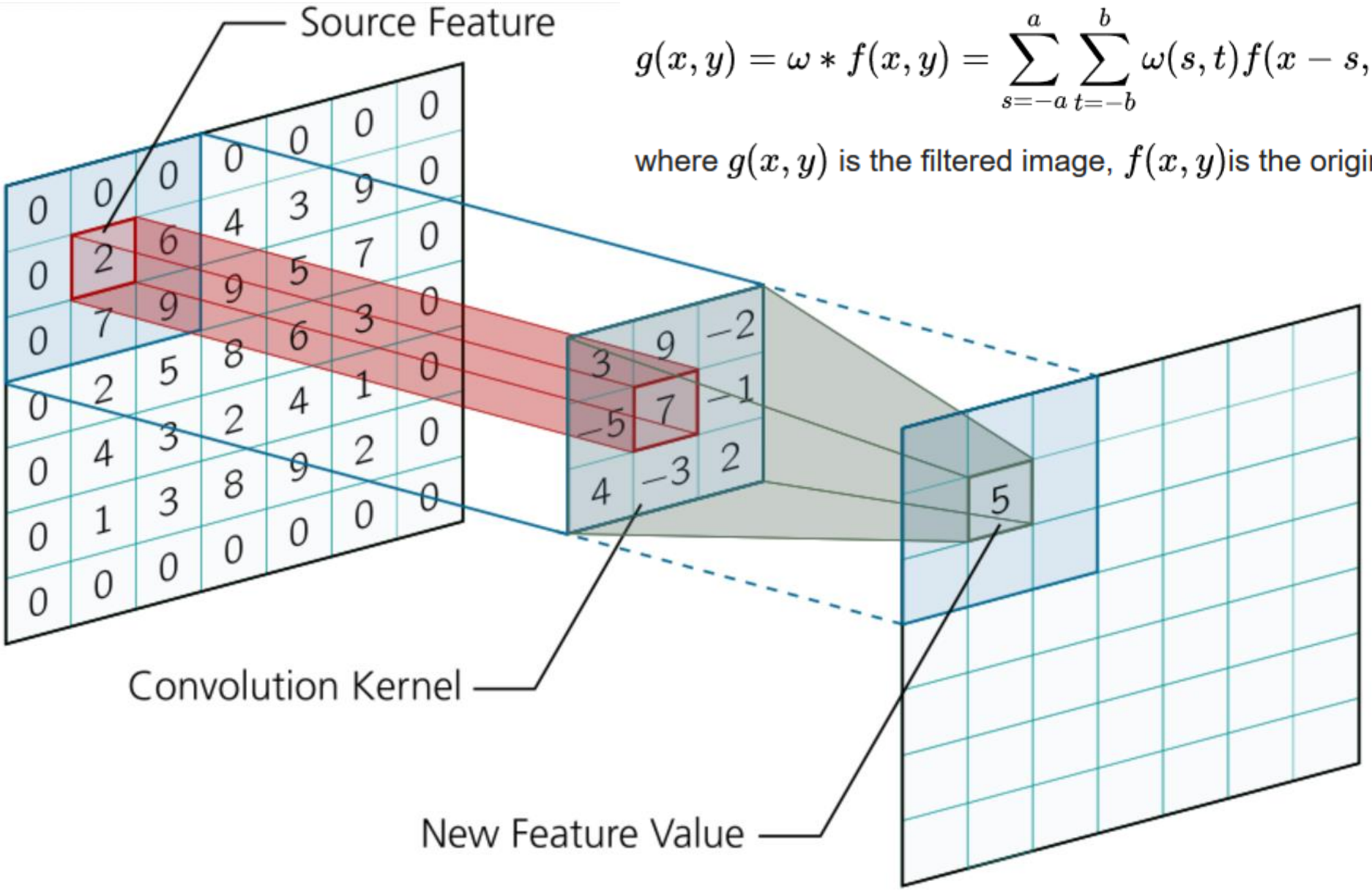


Image Filter (2D convolution)



$$g(x,y) = \omega * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t) f(x-s, y-t),$$

where $g(x,y)$ is the filtered image, $f(x,y)$is the original image, ω is the filter kernel.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

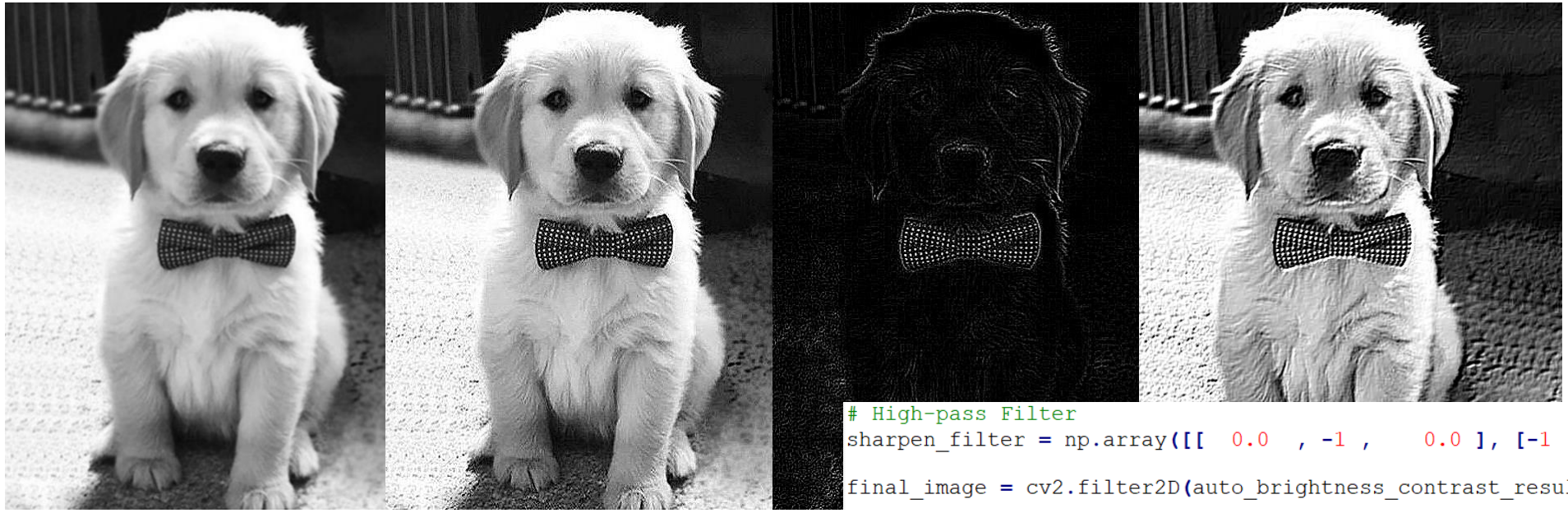
$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Image Filter (2D convolution)




cv2.filter2D( )

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \quad \begin{bmatrix}0 & -0.5 & 0\\ -0.5 & 3 & -0.5\\ 0 & -0.5 & 0\end{bmatrix} \quad \begin{bmatrix}-1 & -1 & -1\\ -1 & 8 & -1\\ -1 & -1 & -1\end{bmatrix} \quad \begin{bmatrix}-2 & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & 2\end{bmatrix}$$

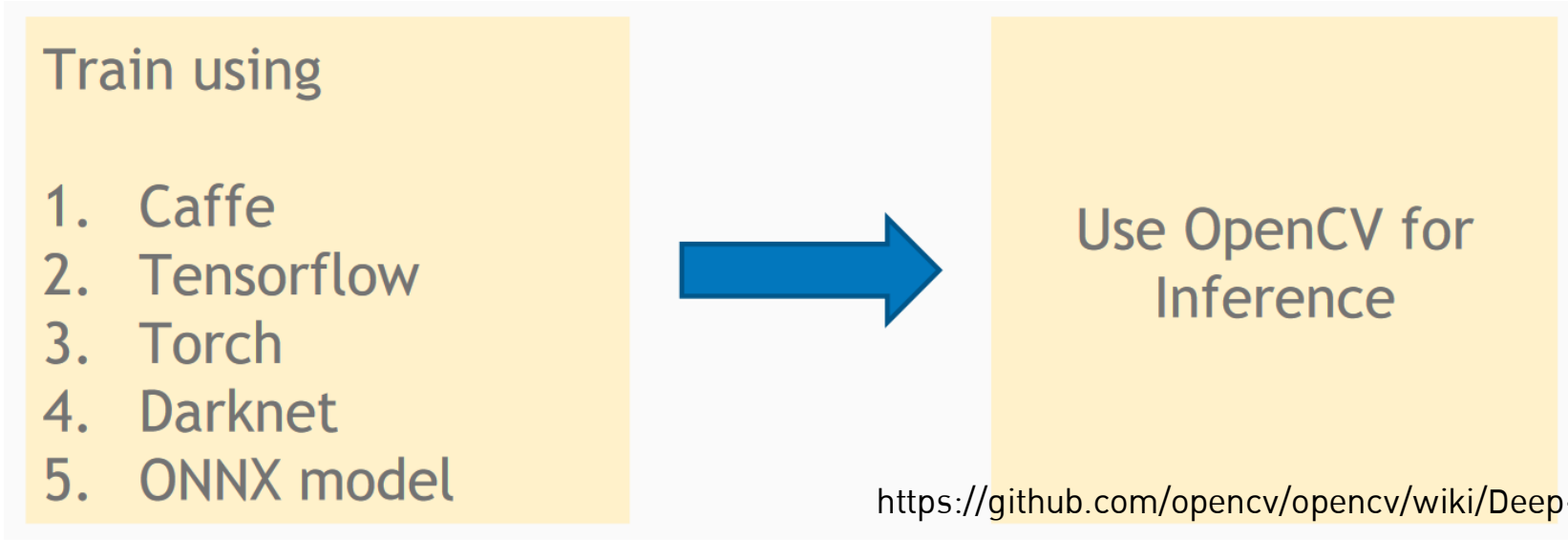


```
# High-pass Filter
sharpen_filter = np.array([[  0.0   , -1 ,    0.0 ], [-1 ,   5   , -1 ],[  0.0   , -1 ,    0.0 ]])

final_image = cv2.filter2D(auto_brightness_contrast_result, -1, sharpen_filter)
```

Cv2.dnn (Deep Neural Networks (dnn module))

Deep Neural Networks Module in OpenCV

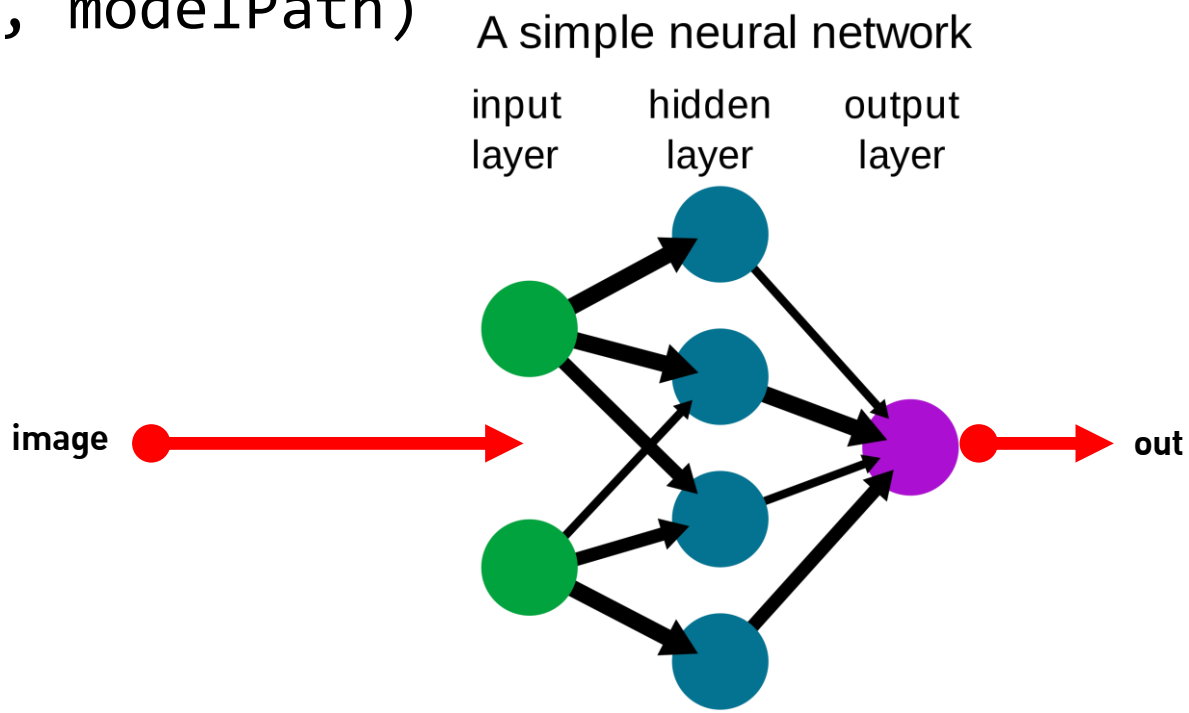
Deep Learning is the most popular and the fastest growing area in Computer Vision nowadays. Since OpenCV 3.1 there is DNN module in the library that implements forward pass (inferencing) with deep networks, pre-trained using some popular deep learning frameworks, such as Caffe. In **OpenCV 3.3** the module has been promoted from opencv contrib repository to the main repository (https://github.com/opencv/opencv/tree/master/modules/dnn) and has been accelerated significantly.



https://github.com/opencv/opencv/wiki/Deep-Learning-in-OpenCV

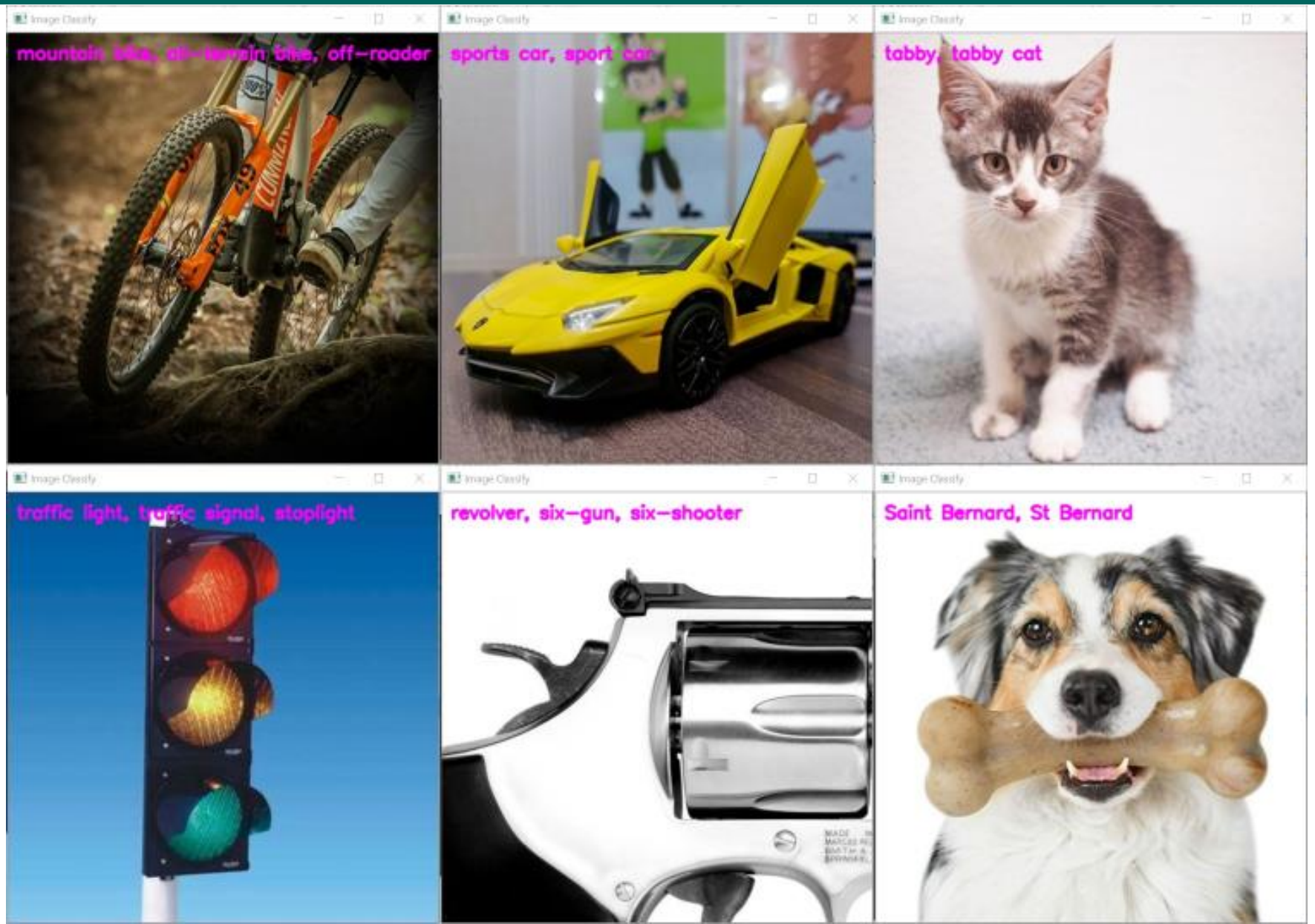
Cv2.dnn for object detection

```
protoPath = "./face_detection_model/deploy.prototxt"
modelPath =
"./face_detection_model/res10_300x300_ssd_iter_140000.caffemodel"

detector = cv2.dnn.readNetFromCaffe(protoPath, modelPath)
detector = cv2.dnn.readNet(protoPath, modelPath)

imageBlob = cv2.dnn.blobFromImage( )
detector.setInput(imageBlob)
detections = detector.forward()
```

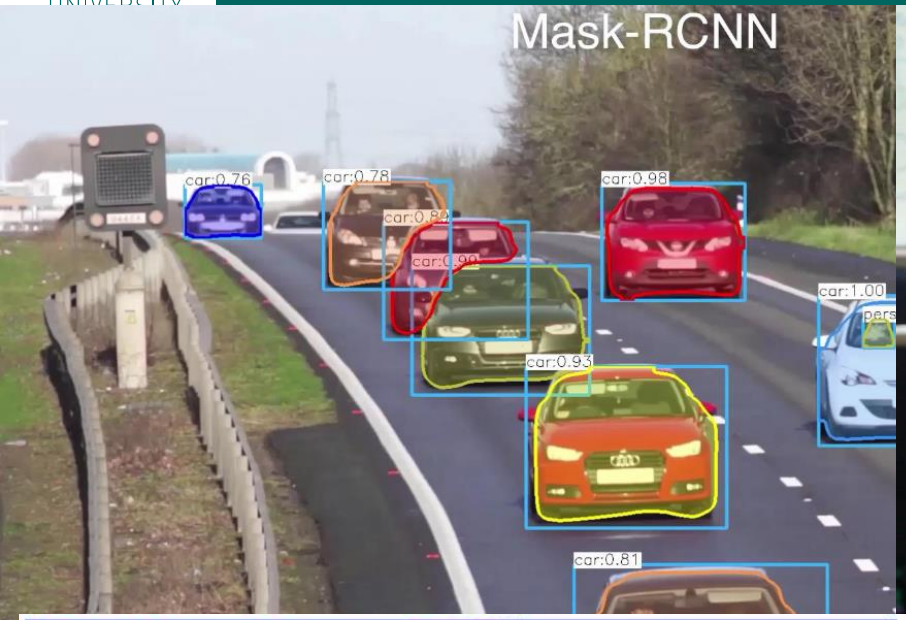


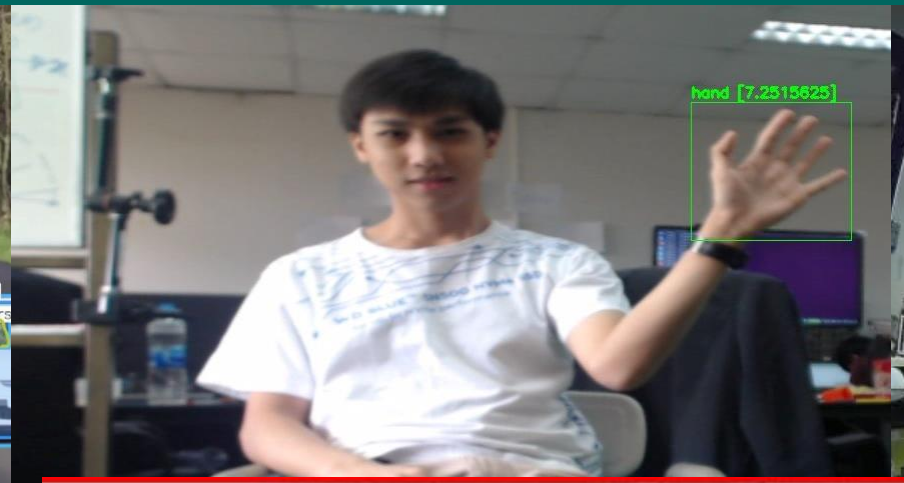
https://github.com/opencv/opencv/wiki/Deep-Learning-in-OpenCV

Cv2.dnn

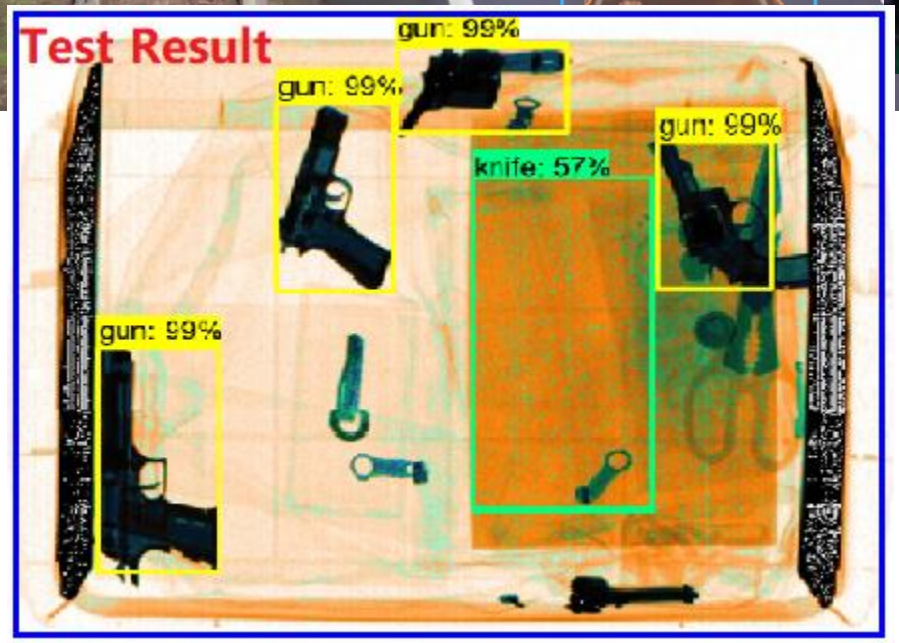

mountain bike, all-terrain bike, off-roader
sports car, sport car
tabby, tabby cat
traffic light, traffic signal, stoplight
revolver, six-gun, six-shooter
Saint Bernard, St Bernard

Cv2.dnn


Mask-RCNN


hand [7.2515625]


person 0.90533316
person 0.91750956
drinking
reading


Test Result
gun: 99%
gun: 99%
gun: 99%
knife: 57%
gun: 99%


Male, (17–25), Happy
bottle


fire

KASETSART
UNIVERSITY

https://tinyurl.com/y5mv6fo9

https://drive.google.com/file/d/1oUCPIOM0_8-Xkyvx668WpRrYNsDeIRAZ/view?usp=sharing

OpenCV and ROS

OpenCV and ROS

```
ros@ros-VirtualBox: ~
ros@ros-VirtualBox:~$ python
Python 2.7.12 (default, Apr 15 2020, 17:07:12)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.3.1-dev'
>>>
```

Problem with ROS **When use dnn**

```
OpenCV Error: Assertion failed (input.dims == 4 && (input.type() == 5 || input.type() == 6)) in finalize, file /tmp/binary
deb/ros-kinetic-opencv3-3.3.1/modules/dnn/src/layers/convolution_layer.cpp, line 78
[ERROR] [1590997247.972806]: bad callback: <function callback_facedetection at 0x7fa15f011848>
Traceback (most recent call last):
  File "/opt/ros/kinetic/lib/python2.7/dist-packages/rospy/topics.py", line 750, in _invoke_callback
    cb(msg)
  File "dectect_face_ros.py", line 33, in callback_facedetection
    detections = detector.forward()
error: /tmp/binarydeb/ros-kinetic-opencv3-3.3.1/modules/dnn/src/layers/convolution_layer.cpp:78: error: (-215) input.dims
== 4 && (input.type() == 5 || input.type() == 6) in function finalize
```

Problem cause by CV2.SO (binary kernel object)

Cv2.dnn

ROS-kinetic CV2.so location

OpenCV3.3.1- dev(path : /opt/ros/kinetic/lib/python2.7/dist-packages/cv2.so)

$cd

$cd /opt/ros/kinetic/lib/python2.7/dist-packages

$ls cv*

$sudo apt-get install python-pip

```
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$ ls cv*
cv2.so  cv_bridge-1.12.8.egg-info

cv_bridge:
boost  core.py  core.pyc  __init__.py  __init__.pyc
```

$sudo mv cv2.so cv4.so

```
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$ sudo mv cv2.so cv4.so
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$ ls cv2*
ls: cannot access 'cv2*': No such file or directory
```

Cv2.dnn

```
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$ python
Python 2.7.12 (default, Apr 15 2020, 17:07:12)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named cv2
>>> exit()
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$
```

$sudo apt-get install python-pip

$pip install opencv-contrib-python==3.4.2.16 opencv-python==3.4.2.16

Cv2.dnn

$pip install opencv-contrib-python==3.4.2.16 opencv-python==3.4.2.16

```
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$ pip install opencv-contrib-python==3.4.2.16
opencv-python==3.4.2.16
Collecting opencv-contrib-python==3.4.2.16
  Downloading https://files.pythonhosted.org/packages/43/1d/e5e7c01fba5ae64abbf76cb3d38ffb3958c38b46ec6292166
e549dde75a1/opencv_contrib_python-3.4.2.16-cp27-cp27mu-manylinux1_x86_64.whl (30.6MB)
    100% |                                | 30.6MB 40kB/s
Collecting opencv-python==3.4.2.16
  Downloading https://files.pythonhosted.org/packages/b0/37/5baf002774374f0694330cc4d11cbd8af38890e928246ee58
877111a70e3/opencv_python-3.4.2.16-cp27-cp27mu-manylinux1_x86_64.whl (25.0MB)
    100% |                                | 25.0MB 42kB/s
Collecting numpy>=1.11.1 (from opencv-contrib-python==3.4.2.16)
  Downloading https://files.pythonhosted.org/packages/2d/f3/795e50e3ea2dc7bc9d1a2eeea9997d5dce63b801e08dfc37c
2efce341977/numpy-1.18.4.zip (5.4MB)
    100% |                                | 5.4MB 178kB/s
    Complete output from command python setup.py egg_info:
    Traceback (most recent call last):
      File "<string>", line 1, in <module>
      File "/tmp/pip-build-TrjkuE/numpy/setup.py", line 32, in <module>
        raise RuntimeError("Python version >= 3.5 required.")
    RuntimeError: Python version >= 3.5 required.

    ----------------------------------------
Command "python setup.py egg_info" failed with error code 1 in /tmp/pip-build-TrjkuE/numpy/
You are using pip version 8.1.1, however version 20.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$
```

What ! Python 2.7 need 3.5

Cv2.dnn

Problem with OpenCV in C/C++

$pip install --upgrade pip

$pip install --upgrade buildtools

$pip install opencv-contrib-python==3.4.2.16 opencv-python==3.4.2.16

```
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$ pip install opencv-contrib-python==3.4.2.16
opencv-python==3.4.2.16
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
/home/ros/.local/lib/python2.7/site-packages/pip/_vendor/requests/__init__.py:83: RequestsDependencyWarning:
Old version of cryptography ([1, 2, 3]) may cause slowdown.
  warnings.warn(warning, RequestsDependencyWarning)
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Pytho
n 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about
Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-
support
Defaulting to user installation because normal site-packages is not writeable
Collecting opencv-contrib-python==3.4.2.16
  Using cached opencv_contrib_python-3.4.2.16-cp27-cp27mu-manylinux1_x86_64.whl (30.6 MB)
Collecting opencv-python==3.4.2.16
  Using cached opencv_python-3.4.2.16-cp27-cp27mu-manylinux1_x86_64.whl (25.0 MB)
Collecting numpy>=1.11.1
  Downloading numpy-1.16.6-cp27-cp27mu-manylinux1_x86_64.whl (17.0 MB)
     |                                | 17.0 MB 10.7 MB/s
Installing collected packages: numpy, opencv-contrib-python, opencv-python
Successfully installed numpy-1.16.6 opencv-contrib-python-3.4.2.16 opencv-python-3.4.2.16
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$
```

KASETSART UNIVERSITY

```
sudo apt -y remove x264 libx264-dev

## Install dependencies
sudo apt -y install build-essential checkinstall cmake pkg-config yasm
sudo apt -y install git gfortran
sudo apt -y install libjpeg8-dev libjasper-dev libpng12-dev

sudo apt -y install libtiff5-dev

sudo apt -y install libtiff-dev

sudo apt -y install libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev
sudo apt -y install libxine2-dev libv4l-dev
cd /usr/include/linux
sudo ln -s -f ../libv4l1-videodev.h videodev.h
cd $cwd

sudo apt -y install libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
sudo apt -y install libgtk2.0-dev libtbb-dev qt5-default
sudo apt -y install libatlas-base-dev
sudo apt -y install libfaac-dev libmp3lame-dev libtheora-dev
sudo apt -y install libvorbis-dev libxvidcore-dev
sudo apt -y install libopencore-amrnb-dev libopencore-amrwb-dev
sudo apt -y install libavresample-dev
sudo apt -y install x264 v4l-utils

```

```
# Optional dependencies
sudo apt -y install libprotobuf-dev protobuf-compiler
sudo apt -y install libgoogle-glog-dev libgflags-dev
sudo apt -y install libgphoto2-dev libeigen3-dev libhdf5-dev doxygen
```

```
sudo apt -y install python3-dev python3-pip python3-venv
sudo -H pip3 install -U pip numpy
sudo apt -y install python3-testresources
```

```
sudo apt -y install python-dev  python-pip  python-venv
sudo -H pip  install -U pip numpy
```

```
pip3 install wheel numpy scipy matplotlib scikit-image scikit-learn ipython dlib
```

```
pip install wheel numpy scipy matplotlib scikit-image scikit-learn ipython dlib
```

```
#Specify OpenCV version
cvVersion="3.4.4"
```

```
git clone https://github.com/opencv/opencv.git
cd opencv
git checkout $cvVersion
cd ..

git clone https://github.com/opencv/opencv_contrib.git
cd opencv_contrib
git checkout $cvVersion
cd ..
```

```
cd opencv
mkdir build
cd build
```

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
      -D CMAKE_INSTALL_PREFIX=/usr/local \
      -D INSTALL_C_EXAMPLES=OFF \
      -D INSTALL_PYTHON_EXAMPLES=OFF \
      -D WITH_TBB=ON \
      -D WITH_V4L=ON \
      -D WITH_QT=ON \
      -D WITH_OPENGL=ON \
      -D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules \
      -D BUILD_EXAMPLES=OFF ..
```

```
nproc
make -j4
sudo make install
sudo sh -c 'echo "/usr/local/lib" >> /etc/ld.so.conf.d/opencv.conf'
sudo ldconfig
```

Cv2.dnn

```
ros@ros-VirtualBox:/opt/ros/kinetic/lib/python2.7/dist-packages$ python
Python 2.7.12 (default, Apr 15 2020, 17:07:12)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.4.2'
>>>
```

READY for cv2.dnn

$tar –xzvf ROS_OpenCV.tar

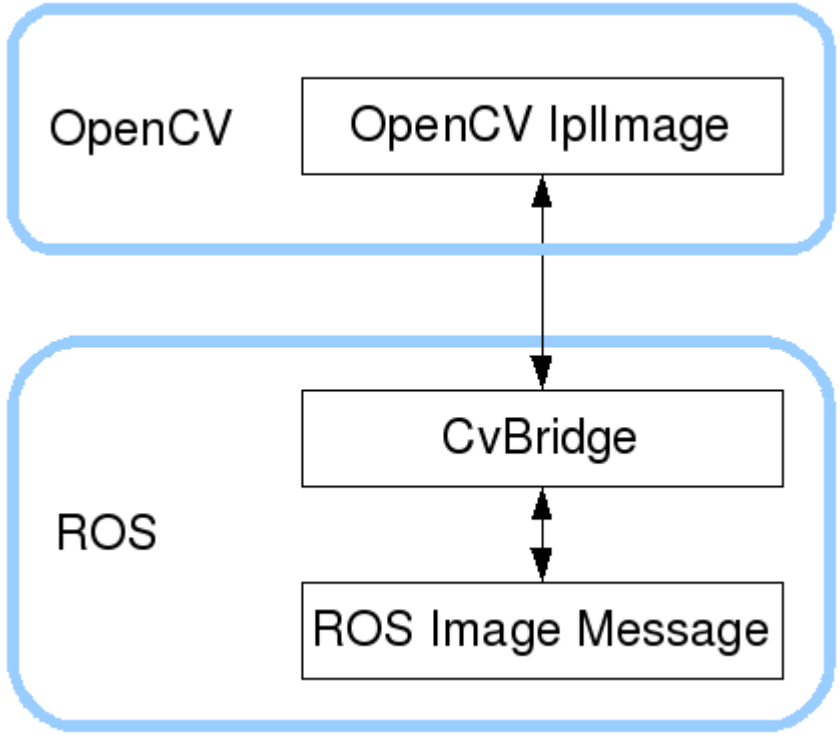
```
ros@ros-VirtualBox:~/ROS_OpenCV$ ls
dectect_face_ros.py   object_detection_ros.py          vokoscreen-2020-05-19_14-24-16.mkv
face_detection_model  publish_video.py
imutils               vokoscreen-2020-05-19_12-57-55.mkv
```

$roscore

$python publish_video.py

$rosrun image_view image_view image:=/camera0/image_raw

CvBridge



ROS passes around images in its own sensor_msgs/Image message format, but many users will want to use images in conjunction with OpenCV. CvBridge is a ROS library that provides an interface between ROS and OpenCV. CvBridge can be found in the cv_bridge package in the vision_opencv stack.

```
from cv_bridge import CvBridge
bridge = CvBridge()
cv_image = bridge.imgmsg_to_cv2(image_message, desired_encoding='passthrough')
```

KASETSART UNIVERSITY

sensor_msgs/Image Message

File: sensor_msgs/Image.msg

Raw Message Definition

```
# This message contains an uncompressed image
# (0, 0) is at top-left corner of image
#

Header header        # Header timestamp should be acquisition time of image
                     # Header frame_id should be optical frame of camera
                     # origin of frame should be optical center of camera
                     # +x should point to the right in the image
                     # +y should point down in the image
                     # +z should point into to plane of the image
                     # If the frame_id here and the frame_id of the CameraInfo
                     # message associated with the image conflict
                     # the behavior is undefined

uint32 height         # image height, that is, number of rows
uint32 width          # image width, that is, number of columns

# The legal values for encoding are in file src/image_encodings.cpp
# If you want to standardize a new string format, join
# ros-users@lists.sourceforge.net and send an email proposing a new encoding.

string encoding       # Encoding of pixels -- channel meaning, ordering, size
                      # taken from the list of strings in include/sensor_msgs/image_encodings.h

uint8 is_bigendian    # is this data bigendian?
uint32 step           # Full row length in bytes
uint8[] data          # actual matrix data, size is (step * rows)
```

```python
import numpy as np
import cv2
import rospy
from sensor_msgs.msg import Image
from cv_bridge import CvBridge, CvBridgeError
bridge = CvBridge()
def main():
    # Set up node.
    rospy.init_node("usb_camera0_publisher", anonymous=True)
    img_pub = rospy.Publisher("/camera0/image_raw", Image, queue_size=10)
    rate = rospy.Rate(5)
    # Open video.
    video = cv2.VideoCapture(0)

    while not rospy.is_shutdown():
        message = "Capture image at %s" % rospy.get_time()
        rospy.loginfo(message)
        ret, img = video.read()
        #cv2.imshow('Frame',img)
        #cv2.waitKey(1)
        img_msg = bridge.cv2_to_imgmsg(img,"bgr8")
        img_msg.header.stamp = rospy.Time.now()
        img_pub.publish(img_msg)
        rate.sleep()

    print(" Cv2 Capture is done")
    video.release()
    cv2.destroyAllWindows()
```

https://tinyurl.com/y5mv6fo9

img = bridge.cv2_to_imgmsg(Image, "bgr8")

- mono8: CV_8UC1, grayscale image
- mono16: CV_16UC1, 16-bit grayscale image
- bgr8: CV_8UC3, color image with blue-green-red color order
- rgb8: CV_8UC3, color image with red-green-blue color order
- bgra8: CV_8UC4, BGR color image with an alpha channel
- rgba8: CV_8UC4, RGB color image with an alpha channel

Cv2.dnn

KASETSART UNIVERSITY

ros@ros-VirtualBox: ~/ROS_OpenCV

```
[INFO] [1591107167.665112]: Capture image at 1591107167.66
[INFO] [1591107167.865674]: Capture image at 1591107167.87
[INFO] [1591107168.065611]: Capture image at 1591107168.07
[INFO] [1591107168.265022]: Capture image at 1591107168.26
[INFO] [1591107168.466501]: Capture image at 1591107168.47
[INFO] [1591107168.666270]: Capture image at 1591107168.67
[INFO] [1591107168.865820]: Capture image at 1591107168.87
[INFO] [1591107169.066785]: Capture image at 1591107169.07
[INFO] [1591107169.265991]: Capture image at 1591107169.27
[INFO] [1591107169.466013]: Capture image at 1591107169.47
[INFO] [1591107169.665425]: Capture image at 1591107169.67
[INFO] [1591107169.865194]: Capture image at 1591107169.87
[INFO] [1591107170.065215]: Capture image at 1591107170.07
[INFO] [1591107170.265516]: Capture image at 1591107170.27
[INFO] [1591107170.465677]: Capture image at 1591107170.47
[INFO] [1591107170.665032]: Capture image at 1591107170.66
[INFO] [1591107170.865927]: Capture
[INFO] [1591107171.070089]: Capture
[INFO] [1591107171.265679]: Capture
[INFO] [1591107171.465585]: Capture
[INFO] [1591107171.665326]: Capture
[INFO] [1591107171.865361]: Capture
```

/camera0/image_raw

(x=118, y=262) ~ R:160 G:175 B:172

roscore http://ros-VirtualBox:11311/

```
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ros-VirtualBox:46661/
ros_comm version 1.12.14
```

ros@ros-VirtualBox: ~

```
ros@ros-VirtualBox:~$ rosrun image_view image_view image:=/camera0/image_raw
[ INFO] [1591107147.375128180]: Using transport "raw"
```

Image Captured by opencv

Publish to ROS by cvBridge

Display by package image view

Cv2.dnn

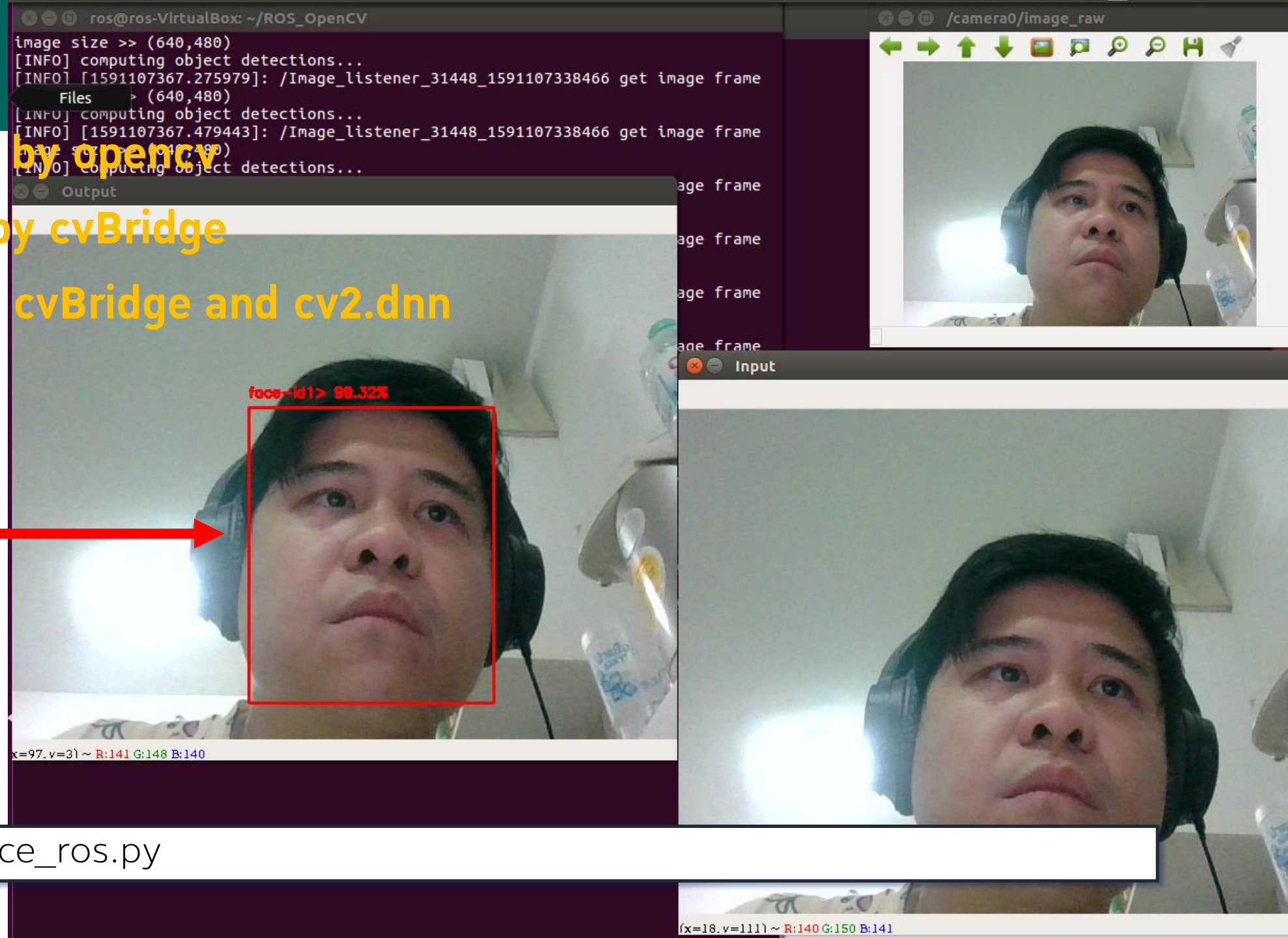
Image Captured by opencv

Publish to ROS by cvBridge

Convert back by cvBridge and cv2.dnn

Face detection !

By dnn




```
$python dectect_face_ros.py
```

KASETSART UNIVERSITY

```python
import rospy
import numpy as np
import cv2
import datetime
import os
import time
from std_msgs.msg import String

from sensor_msgs.msg import Image

from cv_bridge import CvBridge, CvBridgeError

bridge = CvBridge()

# for more model ref to https://github.com/opencv/opencv_extra/blob/master/testdata/dnn/download_models.py

print("[INFO] loading face detector...")
protoPath = "./face_detection_model/deploy.prototxt"
modelPath = "./face_detection_model/res10_300x300_ssd_iter_140000.caffemodel"
detector = cv2.dnn.readNetFromCaffe(protoPath, modelPath)
detector = cv2.dnn.readNet(protoPath, modelPath)
```

Load Neural Network model + weight to System

```python
def callback_facedetection(Image):

    rospy.loginfo(rospy.get_caller_id()+" get image frame")
    img = bridge.imgmsg_to_cv2(Image, "bgr8")
    (h, w) = img.shape[:2]
    print("image size >> (%s,%s)"%(w,h))
    cv2.imshow("Input", img)
    cv2.waitKey(1)
    print("[INFO] computing object detections...")
    imageBlob = cv2.dnn.blobFromImage(cv2.resize(img, (300, 300)), 1.0, (300, 300),(104.0, 177.0, 123.0), swapRB=False, crop=False)
    detector.setInput(imageBlob)
    detections = detector.forward()

    face_count = 0
    for i in range(0, detections.shape[2]):

        # extract the confidence (i.e., probability) associated with the prediction
        confidence = detections[0, 0, i, 2]

        # filter out weak detections
        if confidence > 0.88:

            face_count = face_count+1

            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
```

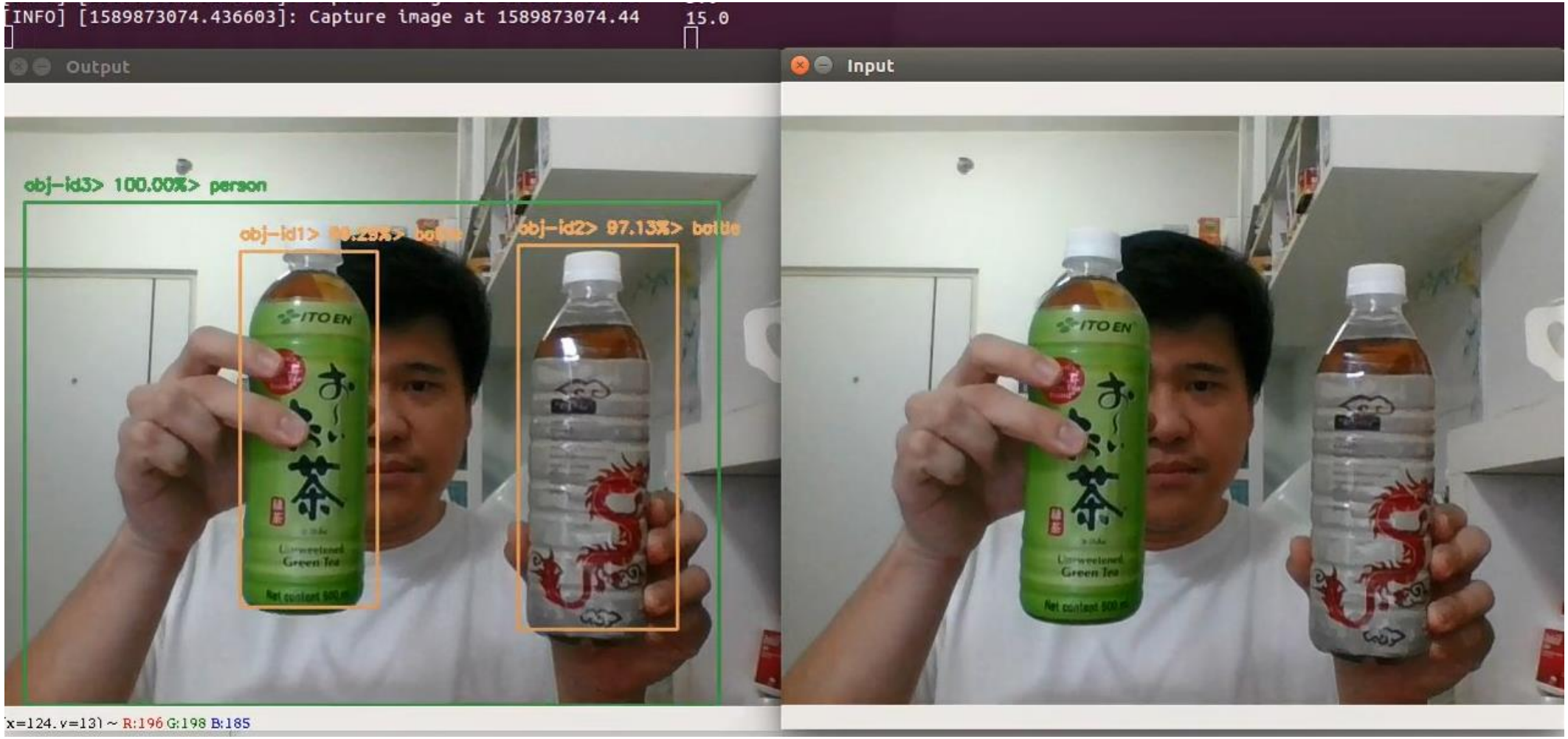
img = bridge.imgmsg_to_cv2(Image, "bgr8")

```python
def listener():

    rospy.init_node('Image_listener', anonymous=True)
    rospy.Subscriber("/camera0/image_raw", Image, callback_facedetection)
     # spin() simply keeps python from exiting until this node is stopped
    rospy.spin()



if __name__ == '__main__':
    print('-----------------------------------------------------------------------')
    print(cv2.__version__)
    print('-----------------------------------------------------------------------')
    try:
        listener()
    except rospy.ROSInterruptException:
        pass
```

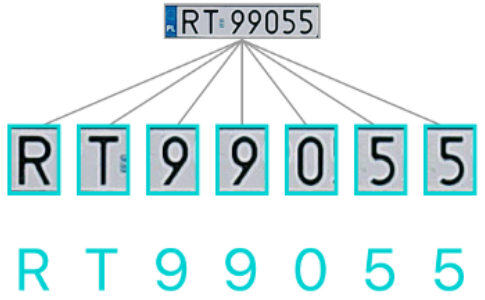
Rosrun image view image view image:=/camera0/raw

Cv2.dnn

$python object_detection_ros.py



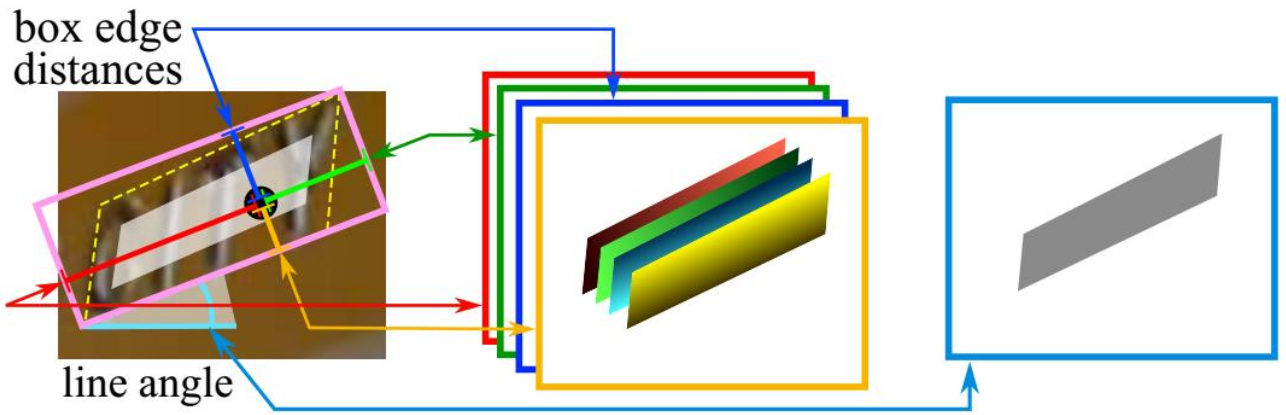
STOP



Car detection → License plate detection → Symbol detection

KU
KASETSART
UNIVERSITY

EAST: An Efficient and Accurate Scene Text Detector



Cv2.dnn

EAST (cv2.dnn) + Tesseract thai

↓

Character detect + OCR thai


Problem OCR
\|7กข 7000
7กข 7000
กรุงเทพมหานคร
กรุงเทพมหานคร
Problem OCR

If deep learning is trained, Cv2.dnn can implemented