ECSE-415
Software for Computer Vision

Introduction to OpenCV-Python

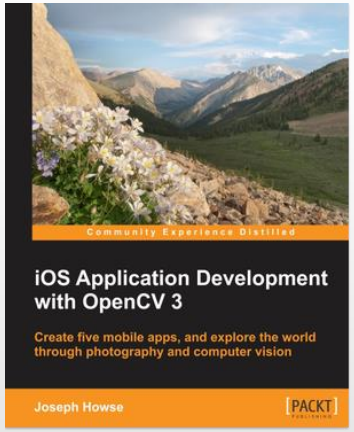
Most materials are from "**OpenCV-Python Tutorials Documentation**"

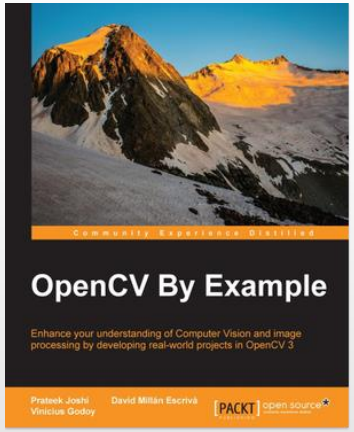
What is OpenCV?

- OpenCV is an open source library for developing computer vision applications
- Cross-platform:
  - C++, C, Python and Java interfaces
  - Supports Windows, Linux, Mac OS, iOS and Android
- Strong focus on real-time applications
  - Multi-core processing
  - Supports hardware acceleration: IPP, OpenCL, CUDA
- Under a BSD license, it can be freely used, distributed and adapted in both academic and commercial apps

Published books about OpenCV

- iOS Application Development with OpenCV 3 (2016)
- OpenCV By Example (2016)
- OpenCV Android Programming By Example (2015)
- OpenCV with Python Blueprints (2015)





- Check http://opencv.org/books.html for an up-to-date list of publications

OpenCV Interface so far

- OpenCV 1.x
  - First stable release
  - C-based API
- OpenCV 2.x
  - C++ API
- OpenCV 3.x
  - C++ API
  - Java API
  - Python API
- Lots of new methods/fixes are added on each minor release
- The programming interface noticeably changes on each major build

OpenCV Python Interface

- Python is a high level general purpose programming language
  - simplicity and code readability
  - compared to C/C++, Python is slower
- It is possible to create a Python wrapper for a C/C++ code
  - The C/C++ code is running without any performance penalty
  - We can benefit from easy coding in Python
- This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation
- We can also benefit from other highly optimized Python libraries
  - Numpy, SciPy, Matplotlib etc

Installing Python + Packages (Anaconda)

Installing Anaconda (Ubuntu)

- Anaconda is a ready-made Python distributions with more than 100 of popular Python packages
- Download and install Anaconda3
  - We assume the installation directory is **~/anaconda3**
  - Allow Anaconda to be added to the system PATH
  - Allow Anaconda to be registered as default Python

Installing Anaconda (Ubuntu)

- Download ***Anaconda3-4.4.0-Linux-x86_64.sh*** for Python 3.6
- On a Terminal
  - bash Anaconda3-4.4.0-Linux-x86_64.sh
- Follow the prompts on the installer screen
- Accept the defaults
- To make changes take effect, close and then re-open the terminal window

Test Python Installation

- On a Terminal
  - Enter `python` to run the Python interpreter
  - On the Python terminal, enter `import numpy` and make sure Numpy is working fine (you will get an error otherwise)

Test Python Installation(Ubuntu)

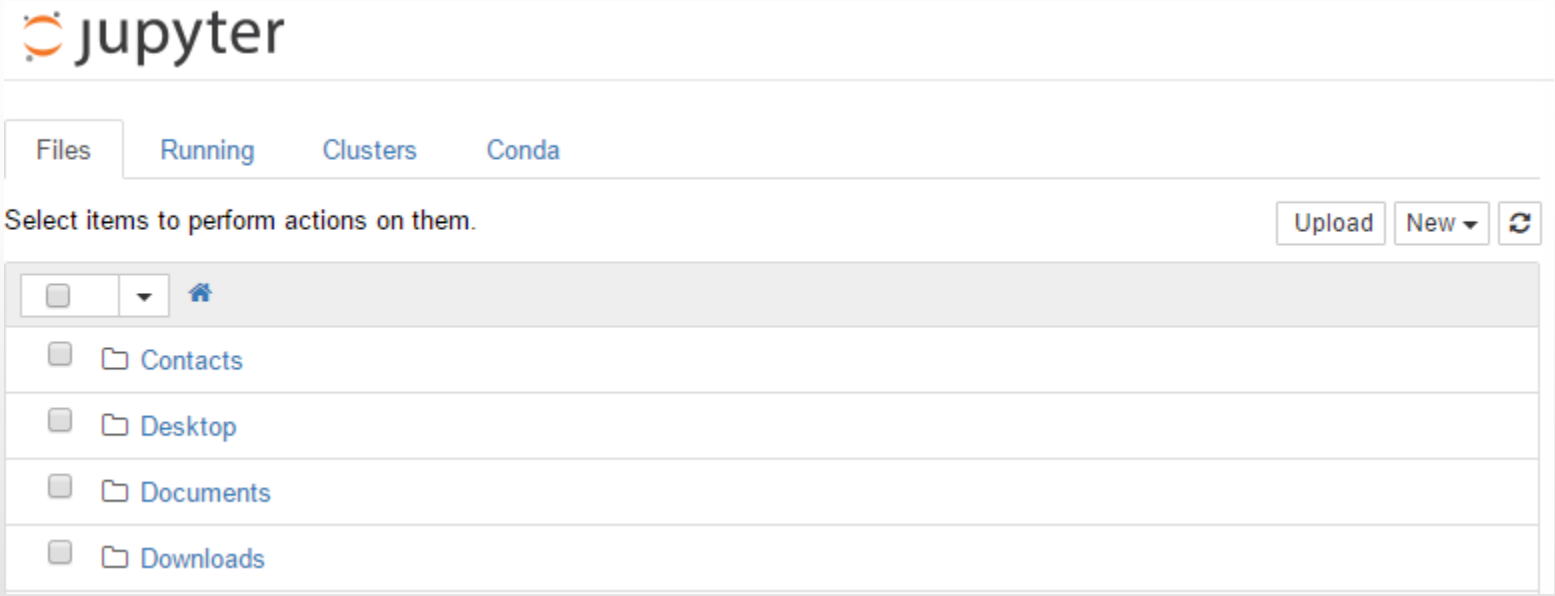
```
gary@gary-MacBookPro: ~
gary@gary-MacBookPro:~$ python
Python 3.6.1 |Anaconda custom (64-bit)| (default, May 11 2017, 13:09:58)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>>
```

Jupyter notebook: An Alternative to Python Terminal

- IPython kernel (already installed with Anaconda) provides a rich architecture for interactive Python programming and data visualization

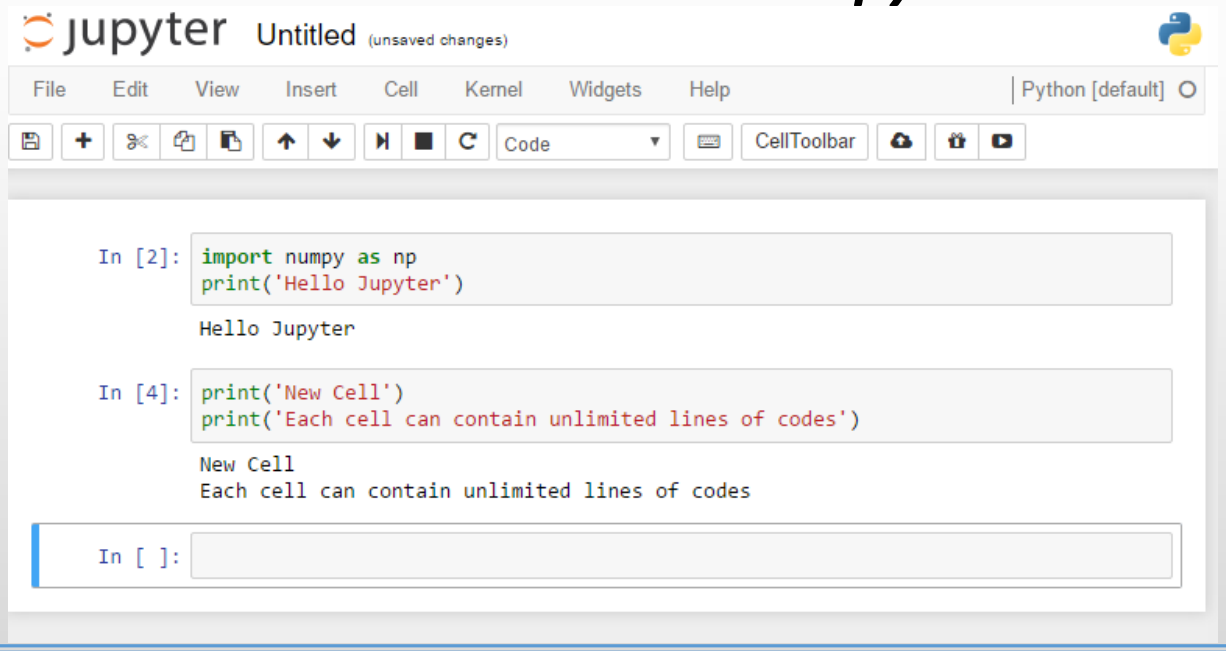
- On a Command Prompt (Windows) or a Terminal (Ubuntu)
  - Enter jupyter notebook
- The Jupyter server will be automatically displayed in a browser tab

Jupyter notebook: An Alternative to Python Terminal

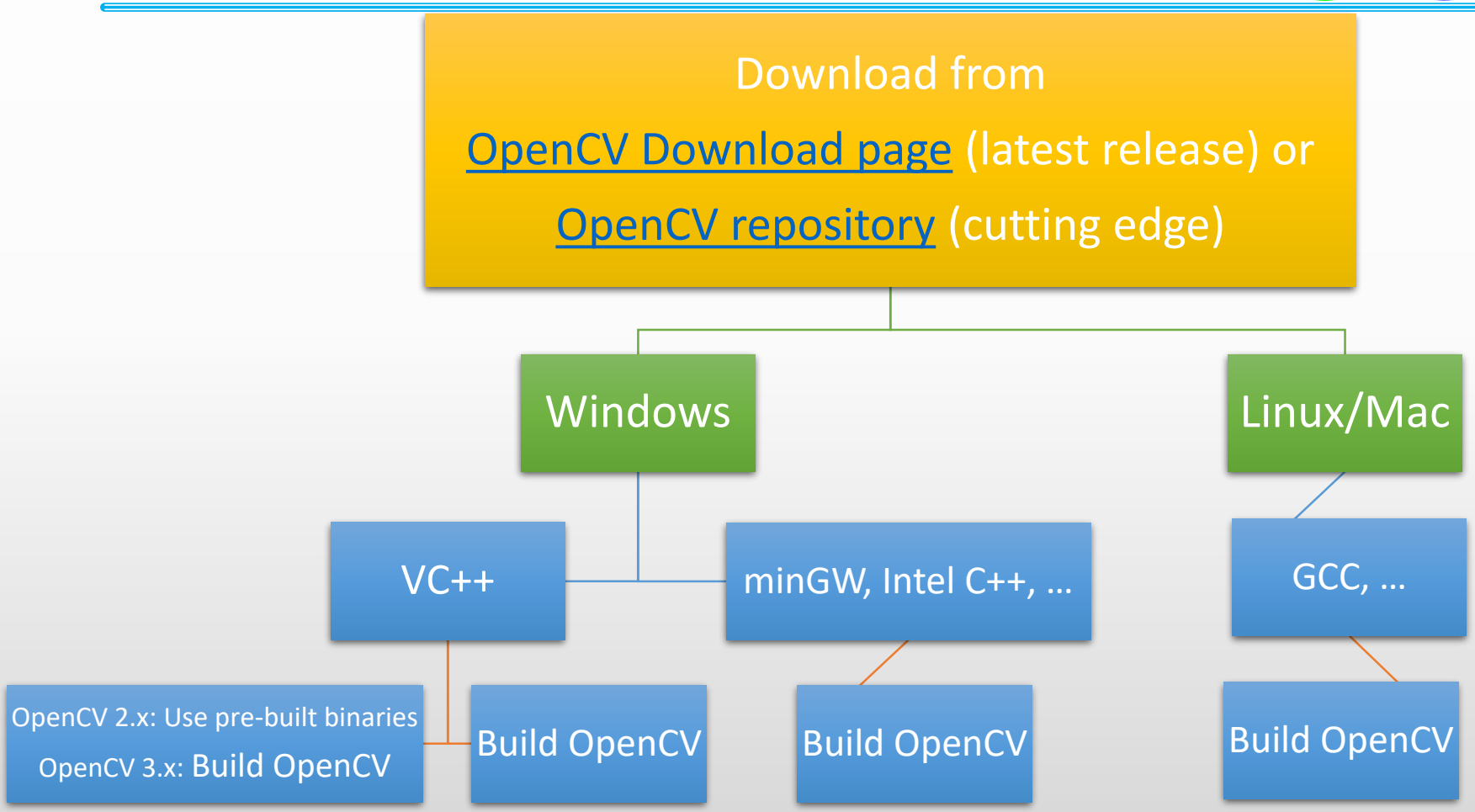


Jupyter notebook: An Alternative to Python Terminal

- Using the New menu, we can create a new Python file
- We can create multiple cells and run each separately
- We can save the notebooks as a ***.ipynb*** file



How to Use OpenCV?



Setting up OpenCV-Python

Ubuntu

Install Dependencies

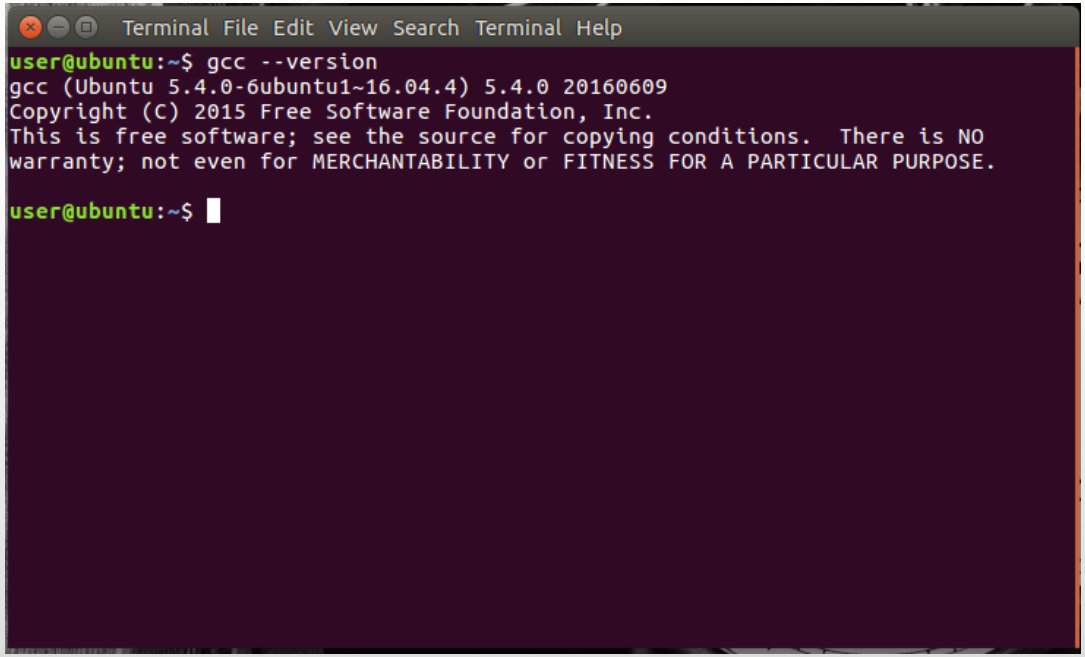
- Make sure to upgrade and update the pre-installed packages first
  - sudo apt-get upgrade
  - sudo apt-get update
- Install developing tools
  - sudo apt-get install build-essential cmake-gui pkg-config
- Install the following packages for accessing various image formats, video frames and cameras
  - sudo apt-get install libjpeg8-dev libtiff5-dev libjasper-dev libpng12-dev
  - sudo apt-get install libdc1394-22-dev
  - sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev
  - sudo apt-get install libv4l-dev libxvidcore-dev libx264-dev

Install Dependencies

- Install the GTK library, required by the OpenCV *highgui* module
  - sudo apt-get install libgtk2.0-dev libgtk-3-dev
- Other required libraries
  - sudo apt-get install libtbb2 libtbb-dev libatlas-base-dev gfortran

Building OpenCV on Ubuntu

- Check if you are using an up-to-date GCC version (5 or 6)



Building OpenCV on Ubuntu

- Finally, the easiest way to install OpenCV is using the Anaconda package environment
- In terminal
  - conda update conda
  - conda install opencv
- Accept the installation of prerequisite packages
- Wait for installation to complete
- Many other anaconda command can be found at this tutorial

Installing OpenCV on Ubuntu

- The include folder ***~/anaconda3/include/opencv & opencv2***
  - Contains all the header files required for building application using OpenCV C++
- The lib folder ~***/anaconda3/lib***
  - Contains all the built library files (*.so* files)
- The python folder ~***/anaconda3/lib/python3.6***
  - Contains the built *opencv-python* library

Setting PYTHON_PATH

- We need to inform Python of the location of OpenCV-Python library (***cv2.*.so***), installed at ***~/anaconda3/lib/python3.6/site-packages***
- Open ***.bashrc*** (gedit ~/.bashrc) and add the following
  - export PYTHONPATH=$PYTHONPATH:~/anaconda3/lib/python3.6/site-packages
  - Export PATH="~/anaconda3/bin:$PATH"

Test OpenCV-Python installation

- Open a Terminal
  - Enter python to run the Python interpreter
    - On the Python terminal, enter
      - import cv2
      - print(cv2.__version__)

```
gary@gary-MacBookPro: ~
gary@gary-MacBookPro:~$ python
Python 3.6.1 |Anaconda custom (64-bit)| (default, May 11 2017, 13:09:58)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print(cv2.__version__)
3.1.0
>>>
```

OpenCV Python Interface

Getting Started with Images: Reading images

- Use the function ***cv2.imread()*** to read an image
- The function accepts two arguments
  - The first argument is a string containing the path of the image file
  - The optional second argument is a flag
    - ***cv2.IMREAD_COLOR*** : Loads a color image (default)
    - ***cv2.IMREAD_GRAYSCALE*** : Loads image in grayscale mode
    - You can check the values of these flags by printing them (e.g. print(cv2.IMREAD_COLOR))
- The function returns a Numpy array, containing the image data

Getting Started with Images: Reading images

```
import numpy as np
import cv2

# Load a color image
img = cv2.imread('/home/user/opencv/samples/data/lena.jpg')
```

Getting Started with Images: Displaying images

- Use the function ***cv2.imshow()*** to display an image
- The function accepts two arguments
  - The first argument is a string indicating the window name
  - The second argument is the image to display

```
import numpy as np
import cv2

# Load a color image
img = cv2.imread('/home/user/opencv/samples/data/lena.jpg')

# Display the image
cv2.imshow('image',img)
# infinitely wait for a user keypress
cv2.waitKey(0)
# Close all windows
cv2.destroyAllWindows()
```

Getting Started with Images: Displaying images

- ***cv2.waitKey()*** is a keyboard binding function
- Its argument is the time in milliseconds
- If 0 is passed, it waits indefinitely for a key stroke

```
import numpy as np
import cv2

# Load a color image
img = cv2.imread('C:\opencv\sources\samples\data\lena.jpg')

# Display the image
cv2.imshow('image',img)
# infinitely wait for a user keypress
cv2.waitKey(0)
# Close all windows
cv2.destroyAllWindows()
```

Getting Started with Images: Displaying images

- ***cv2.destroyAllWindows()*** simply destroys all the windows we created
- If you want to destroy any specific window, use the function ***cv2.destroyWindow()*** where you pass the exact window name as the argument

```
import numpy as np
import cv2

# Load a color image
img = cv2.imread('C:\opencv\sources\samples\data\lena.jpg')

# Display the image
cv2.imshow('image',img)
# infinitely wait for a user keypress
cv2.waitKey(0)
# Close all windows
cv2.destroyAllWindows()
```

Getting S g images

jupyter Untitled (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Python [default]

Code CellToolbar

In [*]:
```
import numpy as np
import cv2

# Load a color image
img = cv2.imread('C:/opencv/sources/samples/data/lena.jpg')

# Display the image
cv2.imshow('image',img)
# Infinitely wait for a user keypress
cv2.waitKey(0)
# Close all windows
cv2.destroyAllWindows()
```

In [ ]:

image