# OpenCV Basics

Lecture #3

January 30, 2019

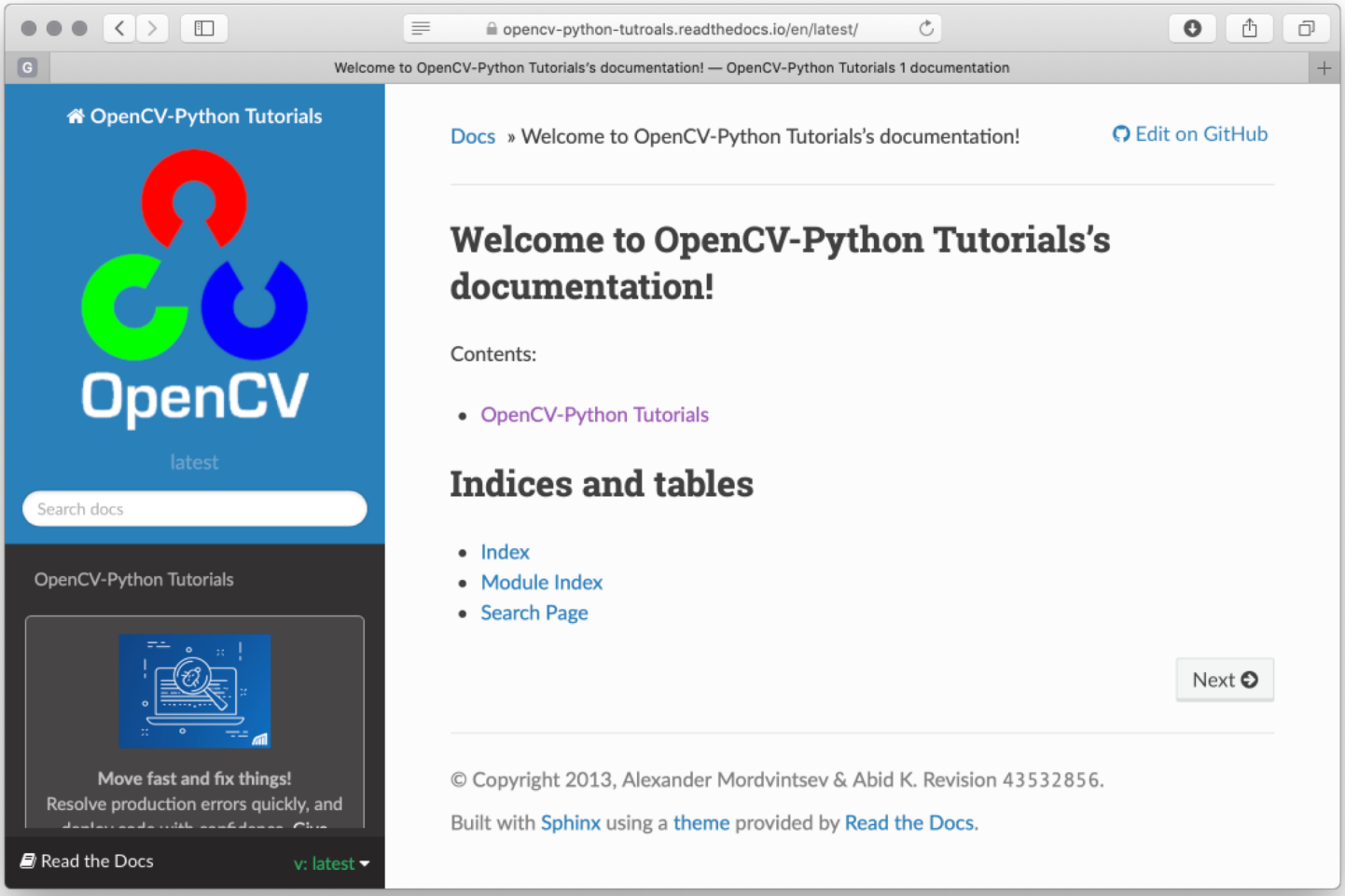
Colorado State University

# … and so it begins

## Intel's computer vision library: applications in calibration, stereo, segmentation, tracking, gesture, face and object recognition

| | |
|---|---|
| Authors | Gary R Bradski, Vadim Pisarevsky |
| Publication date | 2000/6/13 |
| Conference | 2013 IEEE Conference on Computer Vision and Pattern Recognition |
| Volume | 2 |
| Pages | 2796-2796 |
| Publisher | IEEE Computer Society |
| Description | Intel's Microcomputer Research Lab has been developing a highly optimized Computer Vision Library (CVLib) that automatically detects processor type and loads the appropriate MMX™ technology assembly tuned module for that processor. MMX optimized functions are from 2 to 8 times faster then optimized C functions. We will be demonstrating various algorithms supported by CVLib and handing out CDs containing the library. 1. Background ... Over the past year and a half, Intel1 has developed a computer vision library (CVLib) to support real time vision ... |
| Total citations | Cited by 101 |

# Change (for me) – Python Interface




Colorado State University

# Versions and Choices

- CS Department Machines are setup

```
ross@denver:~ — ssh -Y denver.cs.colostate.edu — 64×8
[[ross@denver ~]$ python
[Python 2.7.15 (default, Oct 15 2018, 15:24:06)
[GCC 8.1.1 20180712 (Red Hat 8.1.1-5)] on linux2
Type "help", "copyright", "credits" or "license" for more inform
ation.
>>> import cv2
>>> cv2.__version__
'3.4.1'
```

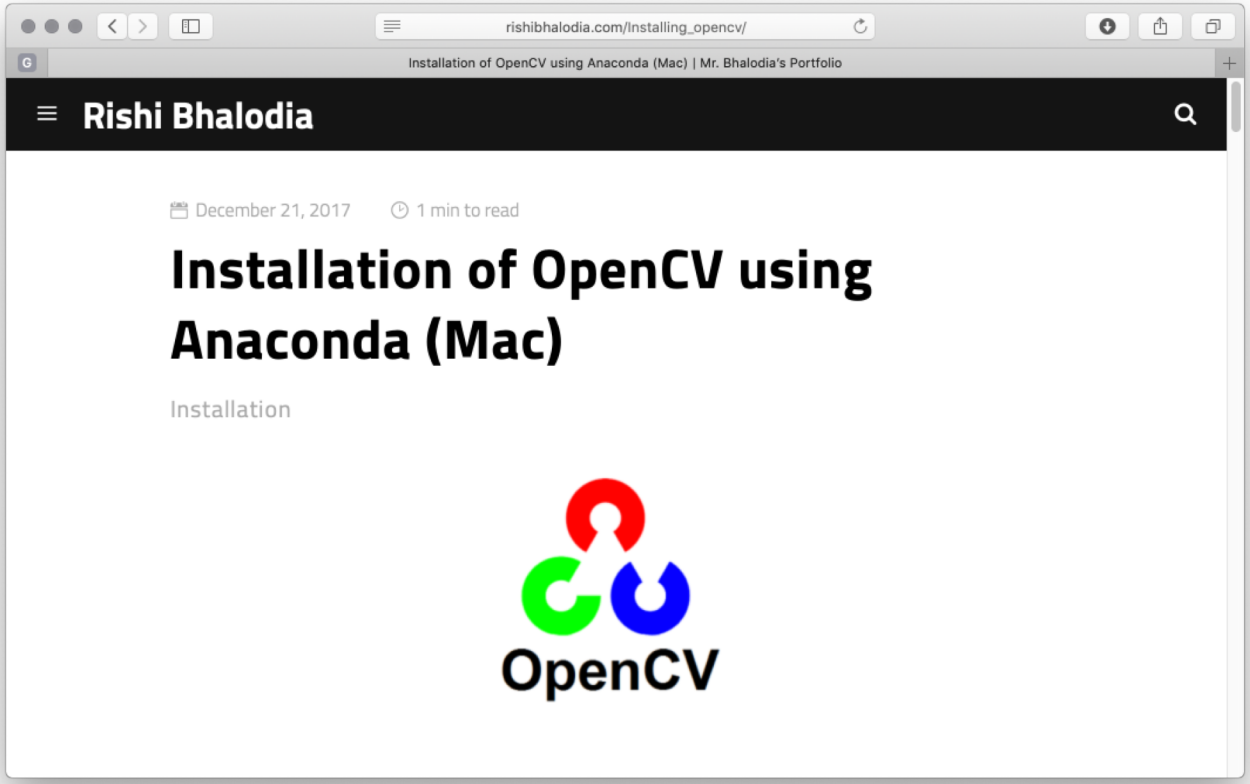
- Your machine is your domain
- But I will say a bit about how I am setup

Fair Disclosure: Some of you are more experienced with Python OpenCV environments. Advice and suggestions are for sharing!

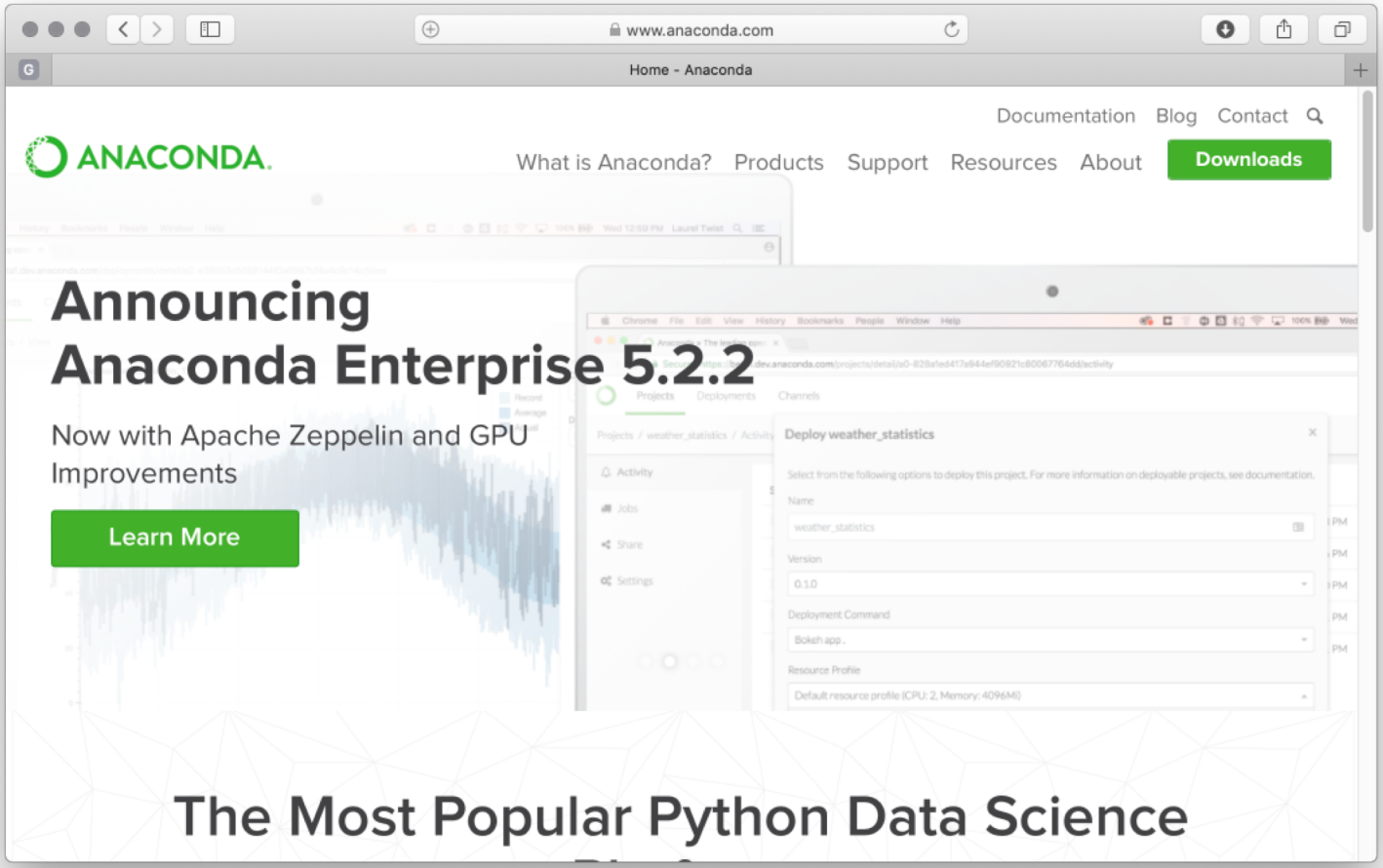
Colorado State University

# If you have a Mac

I did not make an extensive survey out of this topic. I found Bhalodia's guideline worked (with one fix).



# About Anaconda



# My Environment

- I am trying to match versions, but not always easy to accomplish.

IPython: opencvPython/tutorial00 — ipython — 71×16

```
[Hinton:code/opencvPython/tutorial00] ross% ipython
Python 2.7.15 |Anaconda, Inc.| (default, Dec 14 2018, 13:10:39)
Type "copyright", "credits" or "license" for more information.

IPython 5.8.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: import cv2

In [2]: cv2.__version__
Out[2]: '4.0.0'

In [3]:
```

# Now, OpenCV 'Hello World'

opencv-python-tutroals.readthedocs.io/en/latest/p

Getting Started with Images — OpenCV-Python Tutorials 1 documentation

OpenCV

latest

Search docs

- OpenCV-Python Tutorials
  - Introduction to OpenCV
  - Gui Features in OpenCV
    - Getting Started with Images
      - Goals
      - Using OpenCV
      - Using Matplotlib
      - Additional Resources
      - Exercises
    - Getting Started with Videos
    - Drawing Functions in OpenCV
    - Mouse as a Paint-Brush

Read the Docs v: latest

## Getting Started with Images

### Goals

- Here, you will learn how to read an image, how to display it and how to save it back
- You will learn these functions : **cv2.imread(), cv2.imshow() , cv2.imwrite()**
- Optionally, you will learn how to display images with Matplotlib

### Using OpenCV

#### Read an image

Use the function **cv2.imread()** to read an image. The image should be in the working directory or a full path of image should be given.

Second argument is a flag which specifies the way image should be read.

- cv2.IMREAD_COLOR : Loads a color image. Any transparency of image will be neglected. It is the default flag.
- cv2.IMREAD_GRAYSCALE : Loads image in grayscale mode
- cv2.IMREAD_UNCHANGED : Loads image as such including alpha channel

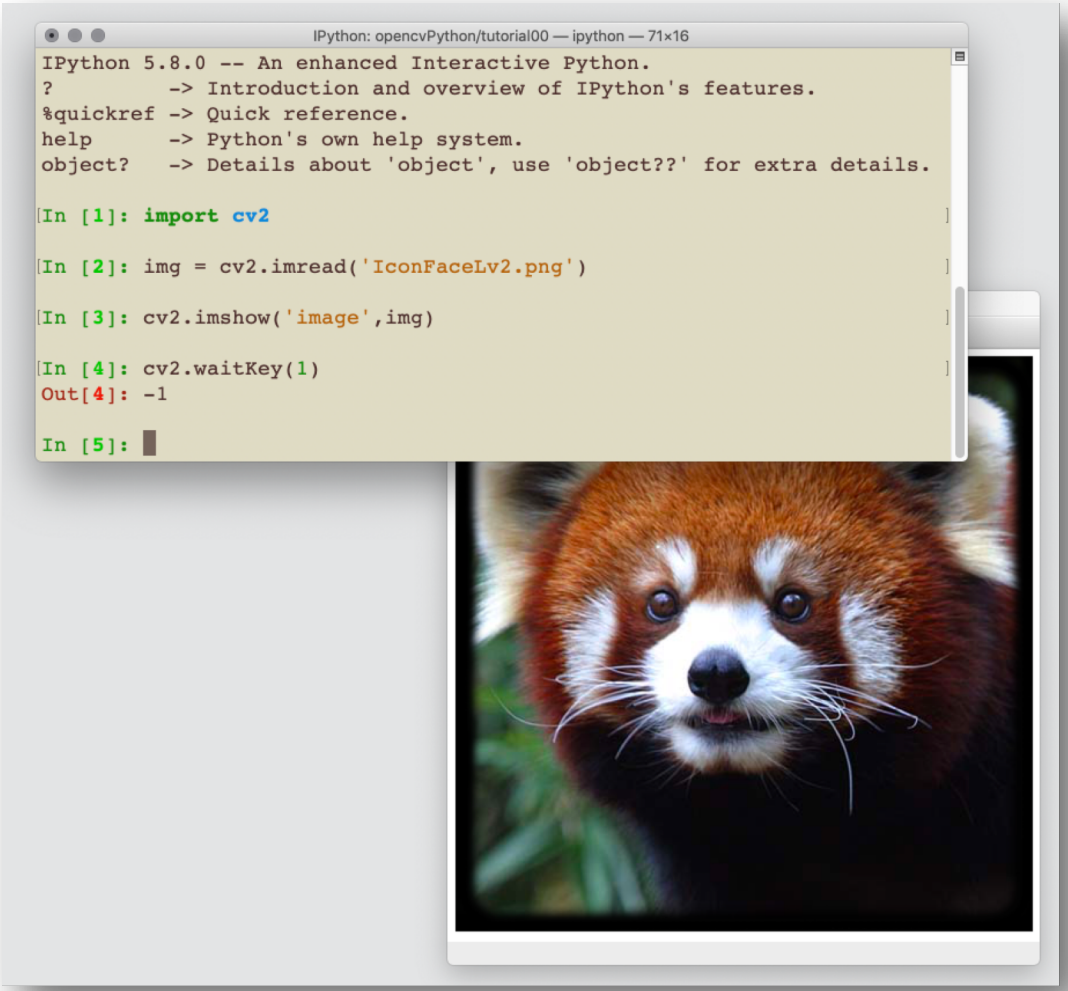
**Note**

Instead of these three flags, you can simply pass integers 1, 0 or -1 respectively.

Colorado State University

# Tutorial 00 – Load and Show

- Sources of Complexity
- Windows are named
- waitKey command matters
- …




Colorado State University

# Next up … Image Container



OpenCV

Search

## Table Of Contents



## Previous topic

*core* module. The Core Functionality

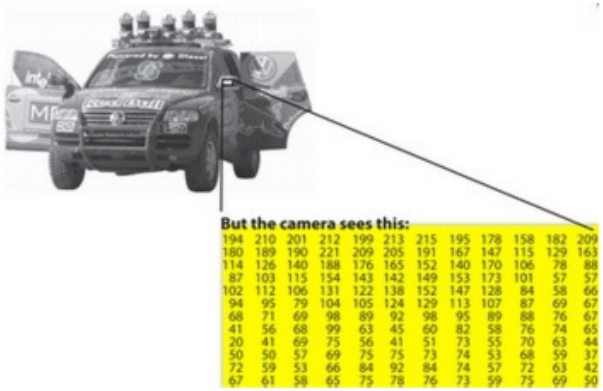
## Next topic

How to scan images, lookup tables

## Mat - The Basic Image Container

### Goal

We have multiple ways to acquire digital images from the real world: digital cameras, scanners, computed tomography, and magnetic resonance imaging to name a few. In every case what we (humans) see are images. However, when transforming this to our digital devices what we record are numerical values for each of the points of the image.




For example in the above image you can see that the mirror of the car is nothing more than a matrix containing all the intensity values of the pixel points. How we get and store the pixels values may vary according to our needs, but in the end all images inside a computer world may be

# MAT as of OpenCV 2.0 (and up)

- Memory management handled for us.
- A tremendous amount of information hiding.
- All and all – a good thing.
- All common types supported
  - Gray scale
  - Color (RGB, HSV, ..)
  - In case you wondered:
    - "There are more than 150 color-space conversion methods available in OpenCV."

# Useful Online Resource

opencv-python-tutroals.readthedocs.io/en/

Basic Operations on Images — OpenCV-Python Tutorials 1 documentation

OpenCV-Python Tutorials

OpenCV

latest

Search docs

OpenCV-Python Tutorials

Introduction to OpenCV

Read the Docs v: latest

Docs » OpenCV-Python Tutorials » Core Operations » Basic Operations on Images

Edit on GitHub

## Basic Operations on Images

### Goal

Learn to:

- Access pixel values and modify them
- Access image properties
- Setting Region of Image (ROI)
- Splitting and Merging images

Almost all the operations in this section is mainly related to Numpy rather than OpenCV. A good knowledge of Numpy is required to write better optimized code with OpenCV.

# Tutorial01 – Print Part of an Image

tutorial01.py

```python
import cv2
import numpy as np

img = cv2.imread('IconFaceLv2.png')

def showPanda() :
    cv2.namedWindow('panda',cv2.WINDOW_NORMAL)
    cv2.imshow('panda',img)
    cv2.waitKey(0)


def printEye () :
    eye = img[180:190, 210:220]
    print eye
```

IPython: opencvPython/tutorial01 — ipython — 48×16

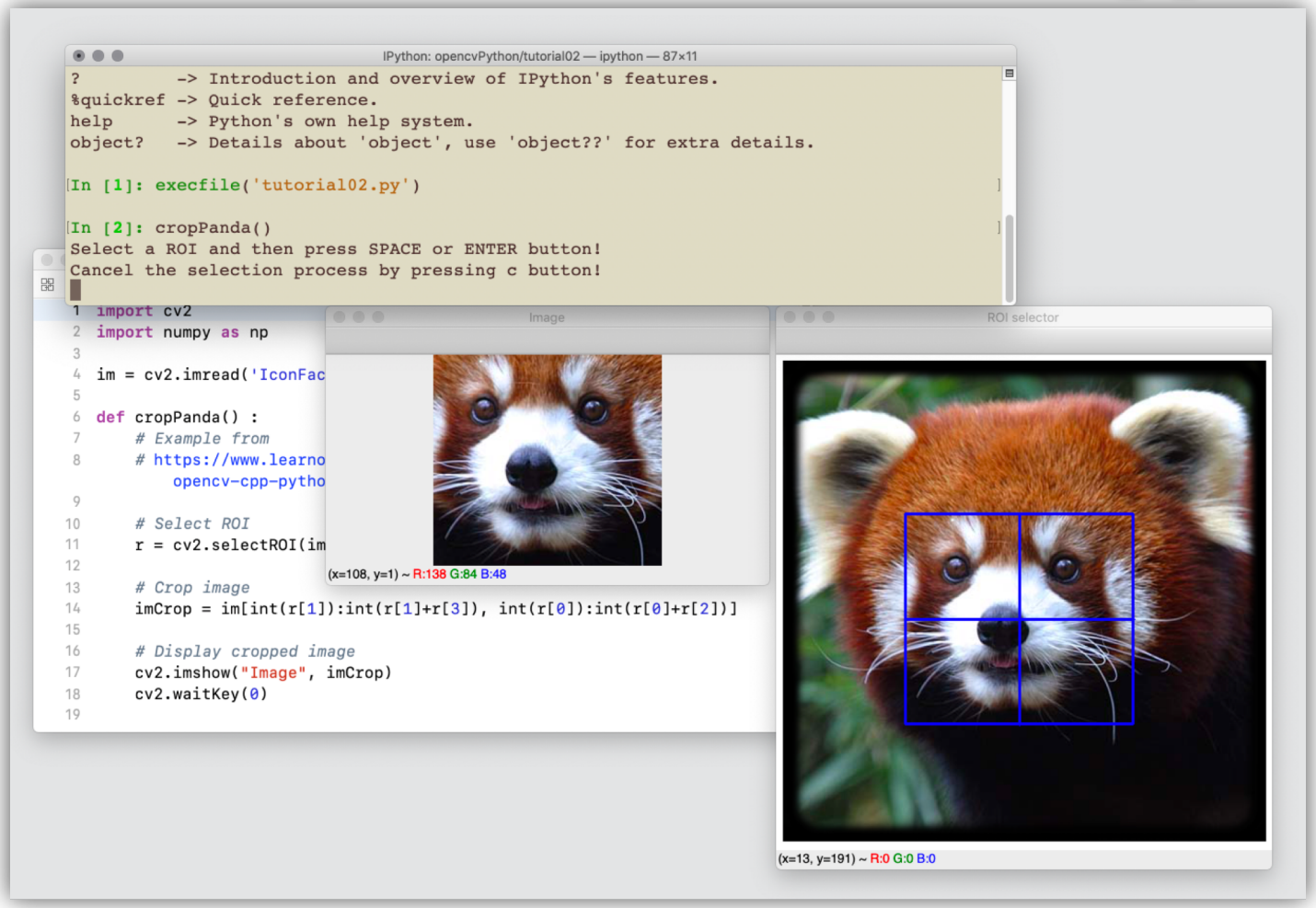
```
 [ 39  58 101]
 [ 44  67 113]
 [ 36  60 106]]

[[246 230 231]
 [236 224 224]
 [217 209 209]
 [188 185 194]
 [165 166 186]
 [129 140 167]
 [ 76  92 129]
 [ 44  63 106]
 [ 47  70 116]
 [ 37  61 107]]]

In [4]:
```

panda

(x=212, y=150) ~ R:147 G:74 B:29

Colorado State University

# Tutorial02 – ROIs and Cropping




Colorado State University

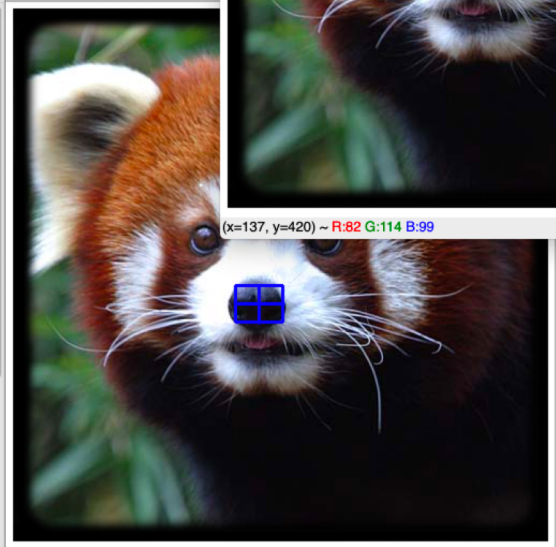
# Tutorial03 – Altering Pixels

IPython: opencvPython/tutorial03 — ipython — 87×11

```
?          -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: execfile('tutorial03.py')

In [2]: colorPanda()
Select a ROI and then press SPACE or ENTER button!
Cancel the selection process by pressing c button!
```

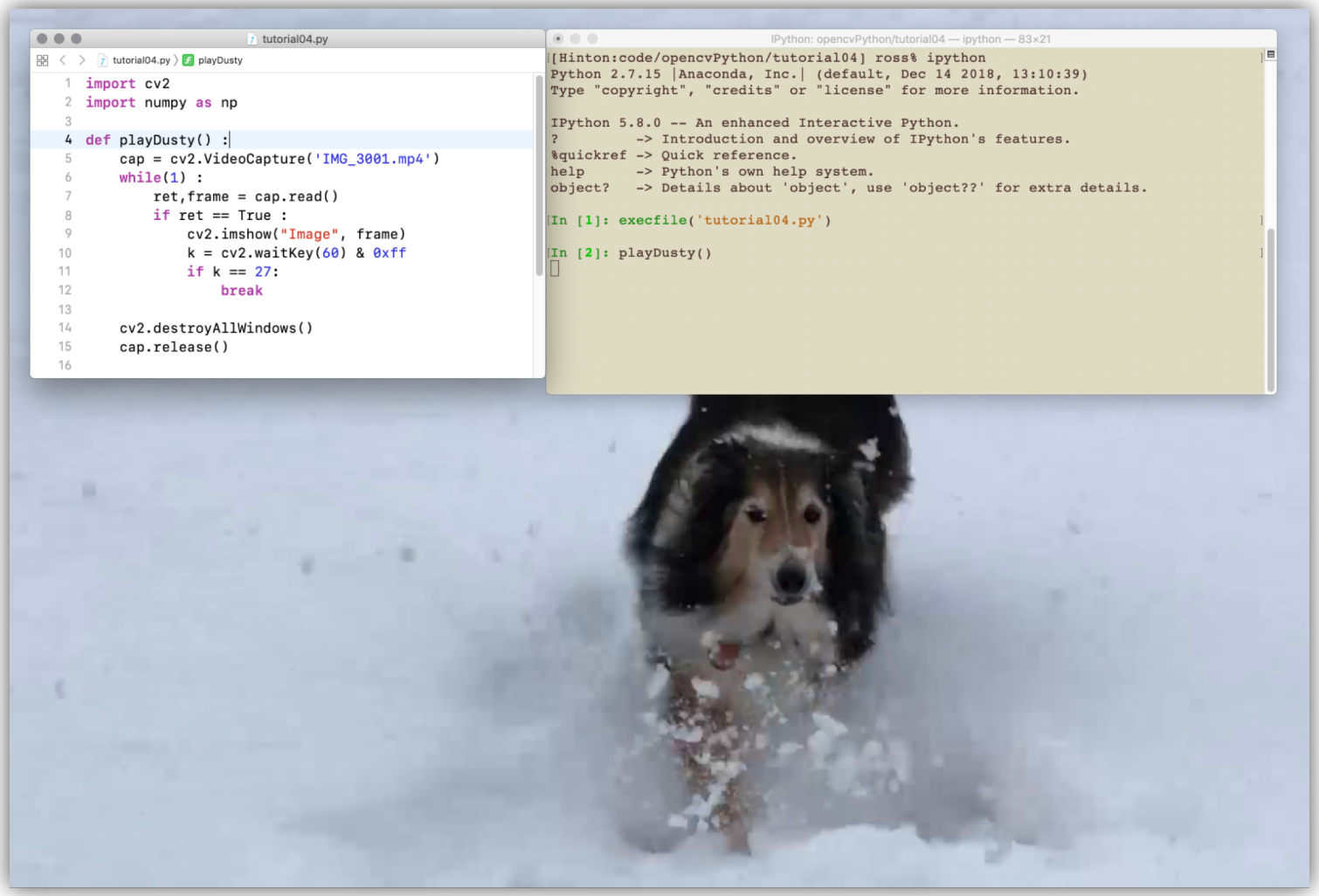
tutorial03.py

```python
import cv2
import numpy as np

im = cv2.imread('IconFaceLv2.png')

def colorPanda() :
    # Example extended from
    # https://www.learnopencv.com/how-to-select-a-bounding-box-roi-in-opencv-cpp-python/

    # Select ROI
    r = cv2.selectROI(im)

    # Turn pixels in the ROI Red !
    im[int(r[1]):int(r[1]+r[3]), int(r[0]):int(r[0]+r[2])] = [0,0,255]

    # Display altered image
    cv2.imshow("Image", im)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

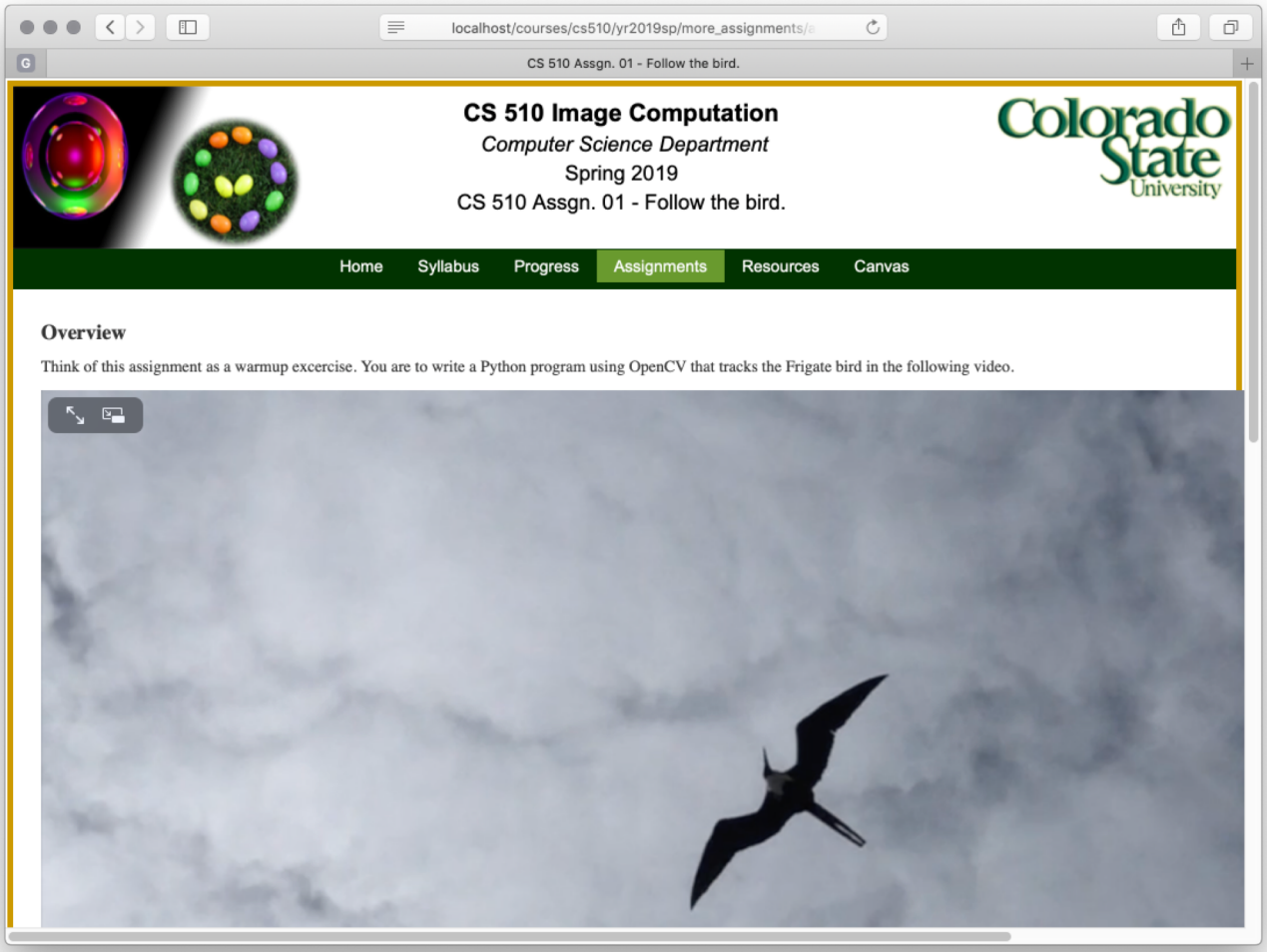

Colorado State University

# Tutorial04 - Video




Colorado State University

# Assignment 1: Frigate Bird




Colorado State University