CSI 4133 – Lab 01

An introduction to OpenCV

Contents

Introduction to the following methods

- Display image
- Save image
- Down-sample image
  - Resize image
- Quantize Image
  - Scale image contents

Installation

- Windows

```
$ pip install opencv-python
```

- Mac

```
$ pip install opencv-python
```

Or

```
$ brew install opencv3 --with-contrib --with-python3
```

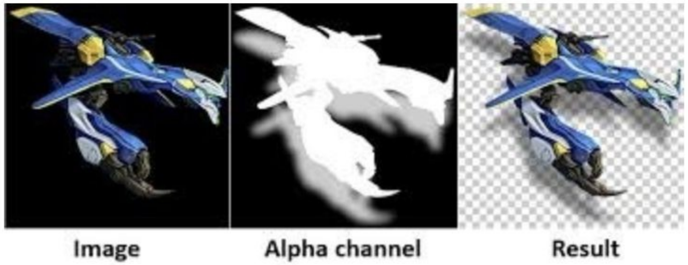
- Linux

```
$ sudo apt-get install libopencv-dev python-opencv
```

- Check installation

```
import cv2
```

PART A: Display an Image



Read an image

- cv2.imread(arg1, arg2)
  - arg1 – string representing the path of the image
  - arg2 – flag that specifies how the image should be read
    - cv2.IMREAD_COLOR / 1:
      - Loads a color image
      - Any transparency of the images is neglected
      - Default flag
    - cv2.IMREAD_GRAYSCALE / 0:
      - Loads image in grayscale mode
    - cv2.IMREAD_UNCHANGED / -1:
      - Loads image as such including alpha channel
      - Alpha channel controls the transparency of a color

PART A: Display an Image

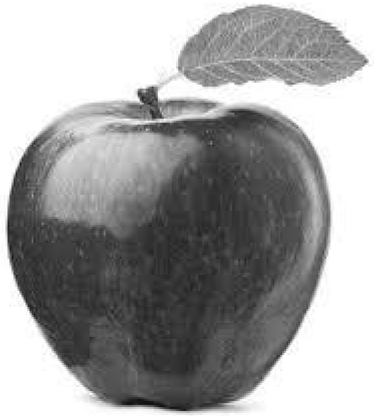
Display an image

- cv2.imshow(arg1, arg2)
    - arg1 – window name
    - arg2 – image
- cv2.waitKey(N)
    - Waits N milliseconds for a keyboard event
    - N=0 will make it wait indefinitely
- cv2.destroyAllWindows()
    - Destroys all the windows we created

PART A: Display an Image

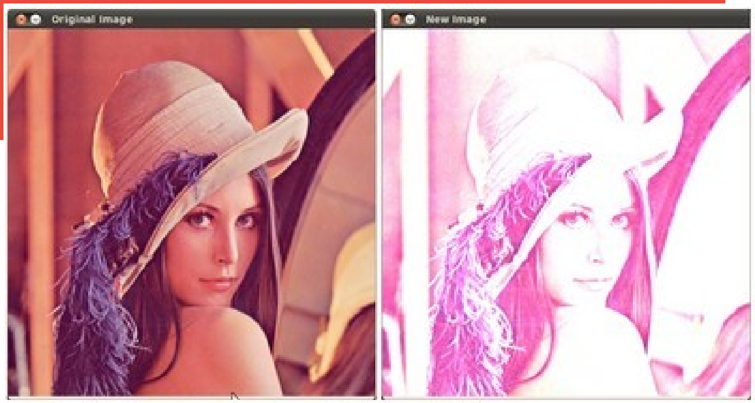
cv2.IMREAD_COLOR

cv2.IMREAD_GRAYSCALE



cv2.IMREAD_UNCHANGED

PART B: Save an Image



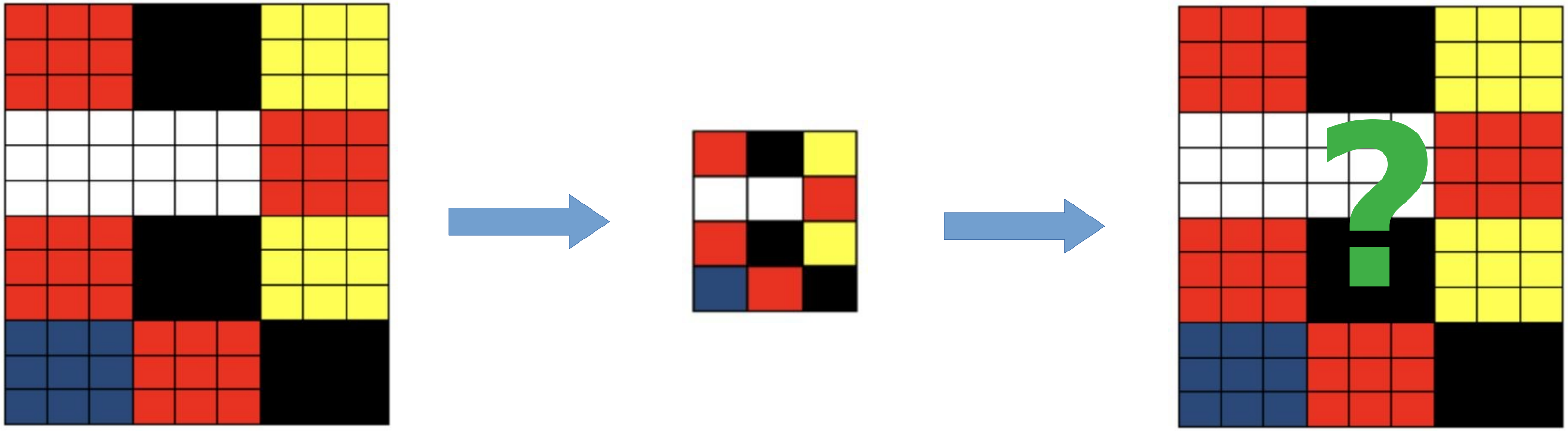
Save an image

- cv2.imwrite(arg1, arg2)
  - **arg1** – string representing the file name. Include image format (jpg, png, …)
  - **arg2** – image to be saved
- Creating an image from scratch
  - import **numpy** as np
  - np.zeros((height,width,3), np.uint8)
  - Use **.shape** to find height and width
    - What is an image??
- For more information
  - https://docs.opencv.org/4.6.0/d9/df8/tutorial_root.html
  - Adding
  - Blending
  - Changing brightness

PART C: Down-sample an Image

Selecting one single value to represent several values

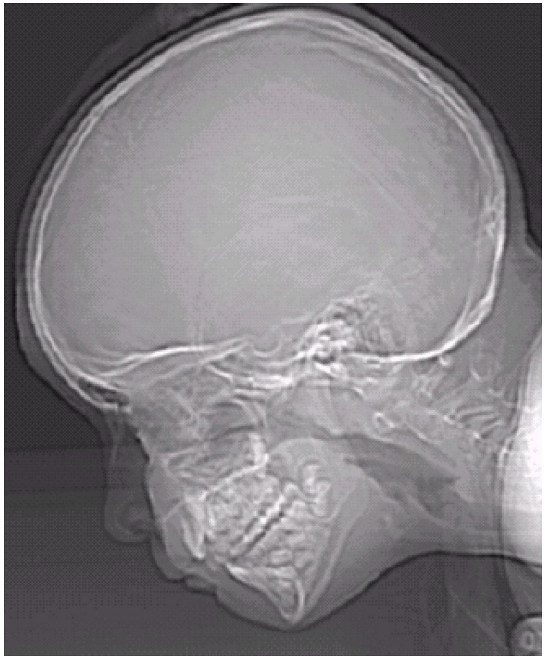
- Makes the data more manageable
- Reduces the dimensionality of the data, enabling faster processing of data
- Reduces storage size of the data



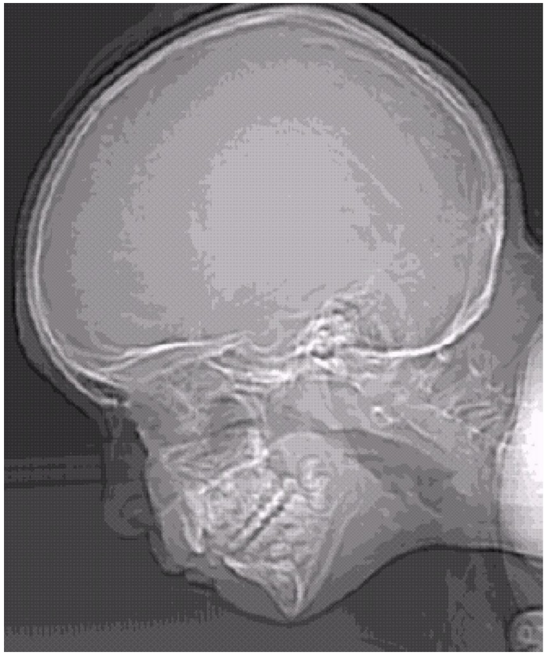
PART D: Quantize an Image

Mapping of a large range of possible sample values into a smaller range of values or codes.

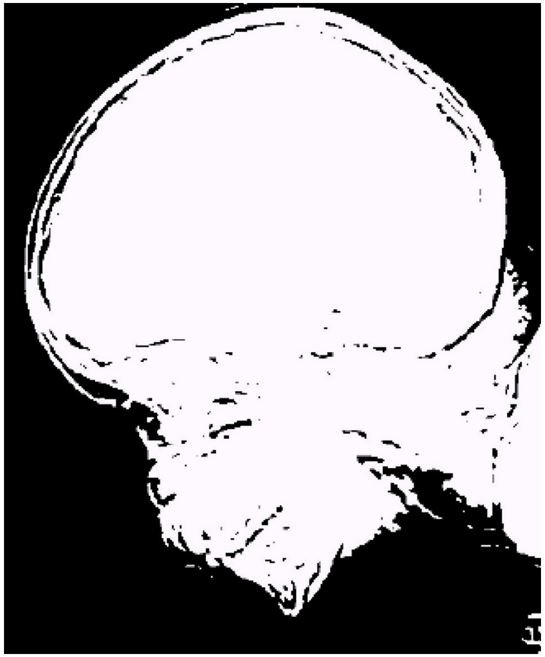
A gray image of 256 levels



A gray image of 16 levels



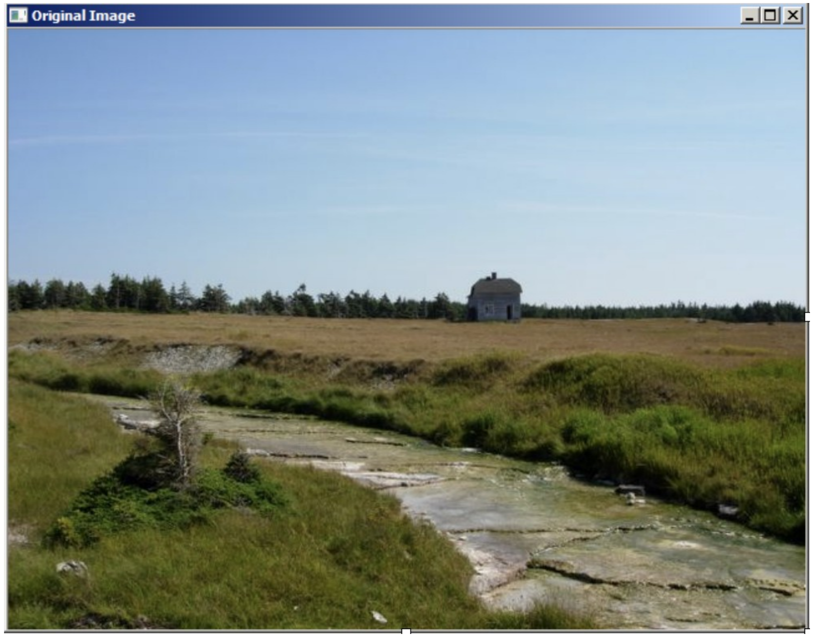
A gray image of 2 levels



PRACTICE

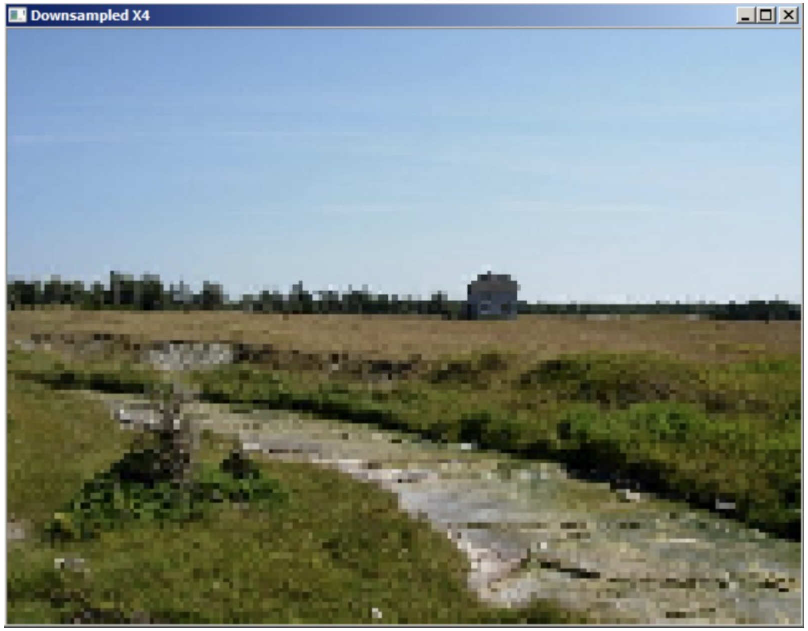
Goal: Introduce students to methods that create new images, resize images & scale image contents.

- Task 1: Display image
  - Load image ‘field.jpg’ and show the original image.



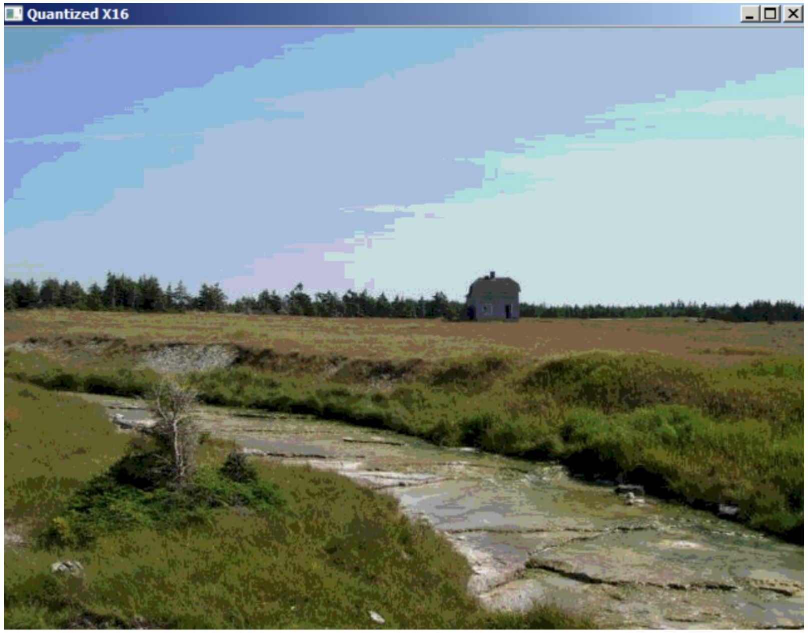
PRACTICE

- Task 2: Down-sample Image
  - Mini-Goal: Down-sample the loaded image (variable id='imgOriginal') by a factor of 4
  - *Hint: Try shrinking the original image by a factor of four and then expanding the resulting image back to the original dimensions*
  - An example – Down-sampled image (X4)



PRACTICE

- Task 3: Quantize Image
  - Mini-Goal: Quantize a loaded image (variable id='imgOriginal') by a factor of 32.
  - An example – Quantized image (X16)



END

THANK YOU