Project Tutorial 1

CS4185 Multimedia Technologies and Applications

Outline

- Introduction to OpenCV
- How to Install OpenCV
- Read & Write images & other related Functions
- The Assignment
- One Simple Comparison Example

Introduction of OpenCV

- ✓ The OpenCV (open computer vision) library is a library of programming functions mainly aimed for real time computer vision applications.
- ✓ It was developed by Intel and is now supported by Willow Garage.
- ✓ It's free for use under the open source BSD License.
- ✓ OS Support:

  Windows , Android ,Blackberry , Linux Distribution such as puppy, Ubuntu, …, Maemo , Free BSD, …
- ✓ Programming language

  OpenCV is written in C++ and its primary interface is in C++. There are full interfacing to Python, Java, MATLAB.
- ✓ OpenCV can be used in Embedded Systems.

Introduction of OpenCV

OpenCV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

- **core** - a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules.
- **imgproc** - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
- **video** - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.

Introduction to OpenCV

- **calib3d** - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
- **features2d** - salient feature detectors, descriptors, and descriptor matchers.
- **objdetect** - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
- **highgui** - an easy-to-use interface to video capturing, image and video codecs, as well as simple UI capabilities.
- **gpu** - GPU-accelerated algorithms from different OpenCV modules.
- ... some other helper modules, such as FLANN and Google test wrappers, Python bindings, and others.

Introduction to OpenCV

- OpenCV documentation:
  http://docs.opencv.org/index.html
- OpenCV tutorials:
  https://docs.opencv.org/4.6.0/d6/d00/tutorial_py_root.html
- OpenCV books: https://opencv.org/books/

Programming interfaces

- C++ (native)
  - Hard for beginners (programming, environments …)
  - Debugging C++ code is suffering
- Python
  - Easy
  - Widely used
- Java and MATLAB interfaces
  - Rarely used

- In this year's project tutorials, we will use Python as our teaching programming interfaces.
  - We also provide the C++ code for reference

Using OpenCV with Python3

1. Install python3 from Miniconda. Select the platform you used.
   https://docs.conda.io/en/latest/miniconda.html

```
(base) garry@JYs-MacBook-Air ~ $ python
Python 3.9.12 (main, Apr  5 2022, 01:52:34)
[Clang 12.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

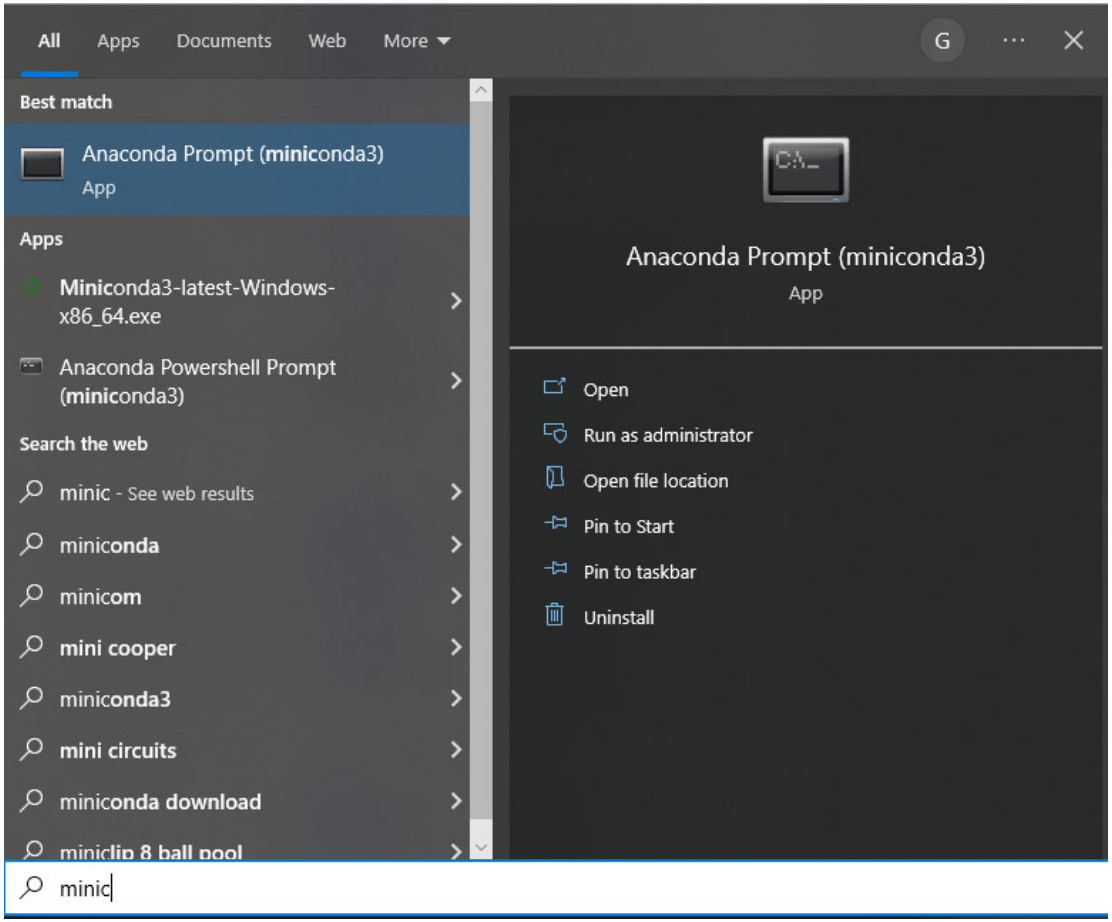
Latest Miniconda Installer Links

Latest - Conda

Platform	Name
Windows	Miniconda3 Windows 64-bit
	Miniconda3 Windows 32-bit
macOS	Miniconda3 macOS Intel x86 64-bit bash
	Miniconda3 macOS Intel x86 64-bit pkg
	Miniconda3 macOS Apple M1 64-bit bash
	Miniconda3 macOS Apple M1 64-bit pkg
Linux	Miniconda3 Linux 64-bit
	Miniconda3 Linux-aarch64 64-bit
	Miniconda3 Linux-ppc64le 64-bit
	Miniconda3 Linux-s390x 64-bit

Using OpenCV with Python3

The anaconda terminal (prompt) in windows



Using OpenCV with Python3

2. Install opencv from terminal by pip.

pip install opencv-python

```
(CS4185) garry@JYs-MacBook-Air ~ $ pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-4.6.0.66-cp37-abi3-macosx_11_0_arm64.whl (30.0 MB)
Collecting numpy>=1.19.3
  Downloading numpy-1.23.2-cp310-cp310-macosx_11_0_arm64.whl (13.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.3/13.3 MB 42.2 MB/s eta 0:00:00
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.23.2 opencv-python-4.6.0.66
```

That is it!

- Check if it is installed successfully:

```
(CS4185) garry@JYs-MacBook-Air ~ $ python
Python 3.10.4 (main, Mar 31 2022, 03:37:37)
Type "help", "copyright", "credits" or "lice
>>> import cv2 as cv
>>>
```

Basic Structure

- Read from an image file

```
img = cv.imread(filename)
```

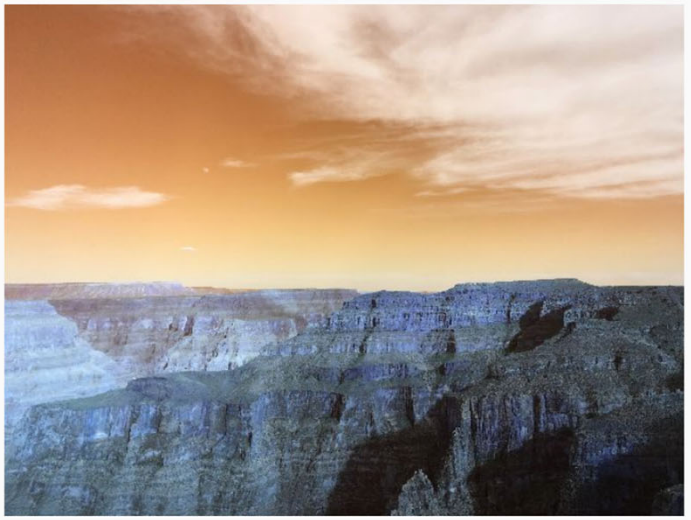
- If you read a jpg file, a 3-channel image is created by default. If you need a grayscale image, use:

```
img = cv.imread(filename, cv.IMREAD_GRAYSCALE)
```

OpenCV Color

- Figure (A) is the original RGB image, while Figure (B) is the same picture saved using BGR format

(A)



(B)

OpenCV Color

- We can also use the cvtColor function to convert the image in different color spaces

(A)

(B)

```
img_gray = cv.cvtColor(img_rgb, cv.COLOR_BGR2GRAY)
```

Operations with images

- Save an image to a file:

  imwrite(filename, img);

- Display an image:

  imshow(windowname, img);

Operations with images

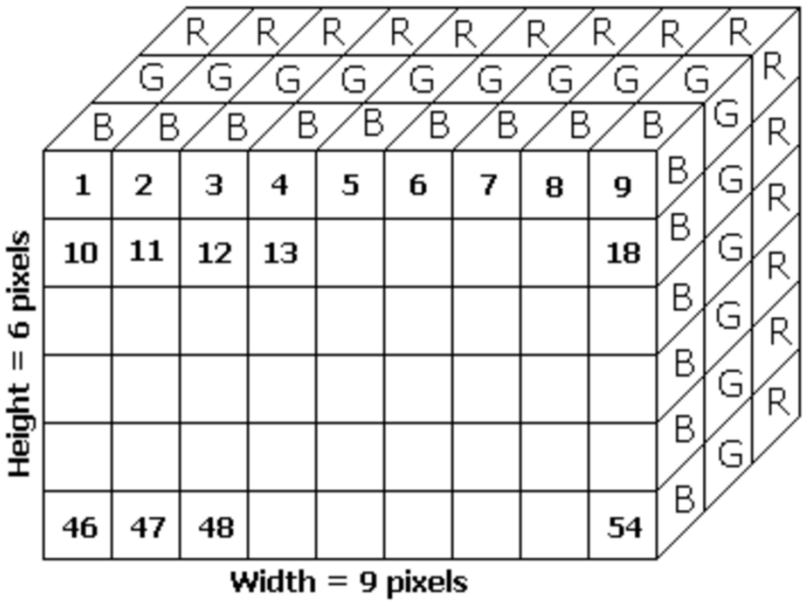
- Image Resize: resize(img, (weight, height))
  - resized_img = cv.resize(img, (320, 240))

(A) Original image

(B) Image resized at 320x240

Operation with images

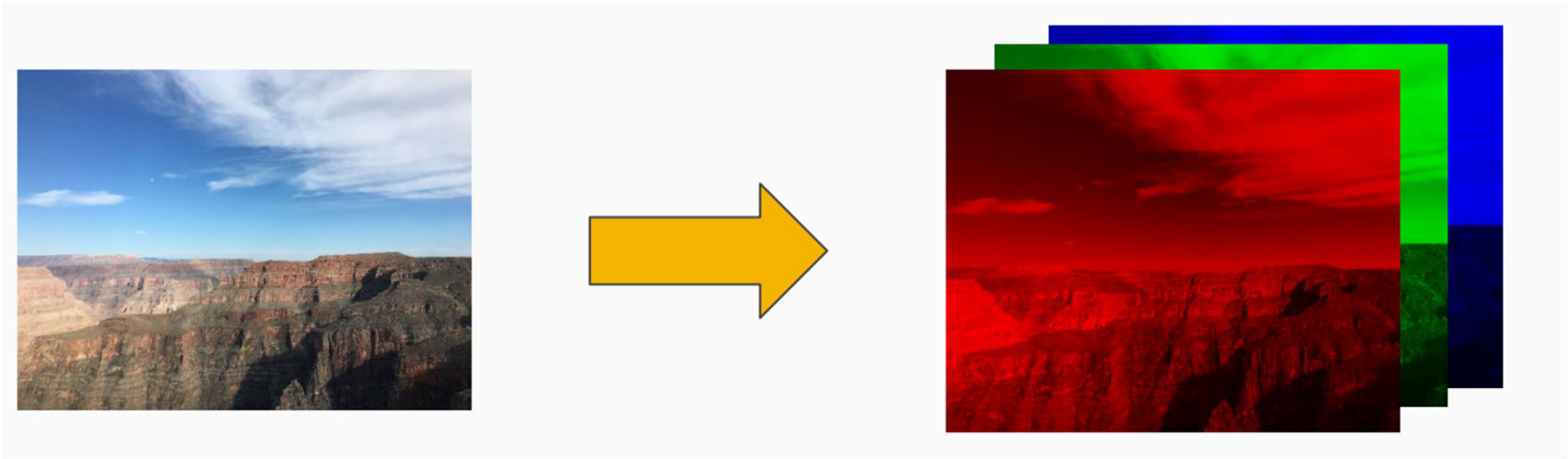
- OpenCV treats images as “arrays”
  - a numpy array that stores the image (each value of the array is a pixel)
  - img.shape returns
  - (height, width, channel)



Operation with images

- OpenCV default format: BGR
  - To obtain components from blue, green and red channel:

```
_blue = img[y,x,0]
_green = img[y,x,1]
_red = img[y,x,2]
```



Operations with images

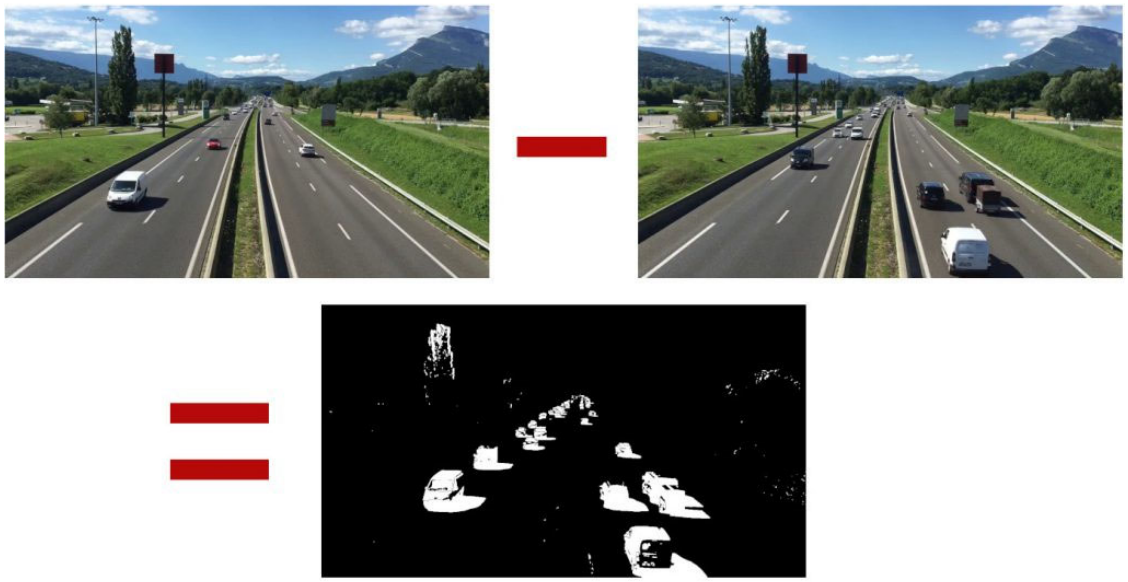
- Element access
  - You can access a pixel value by its row and column coordinates. For a BGR image, it returns an array of Blue, Green, Red values. For a grayscale image, just the corresponding intensity value is returned.

```
>>> px = img[100,100]
>>> print( px )
[157 166 200]

# accessing only blue pixel
>>> blue = img[100,100,0]
>>> print( blue )
157
```

Operations with images

- Find the per-element absolute difference between two images
  - cv2.absdiff(img1, img2)
  - Useful for comparing two images

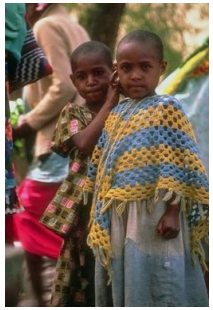


Operations with images

- Lots of operations are available in OpenCV, such as Image Filtering (smooth, blur etc.), Geometric Image Transformations (resize etc.).
- Detailed information can be found in https://docs.opencv.org/master/d7/da8/tutorial_table_of_content_imgproc.html

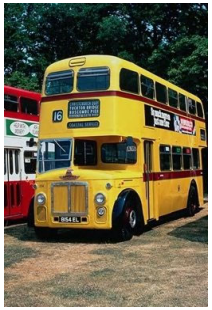
CS4185 Multimedia Technologies and Applications: Course Assignment

- A database of 1000 images, divided into 10 types, is given. Each type contains 100 images.


African


Beach


Building

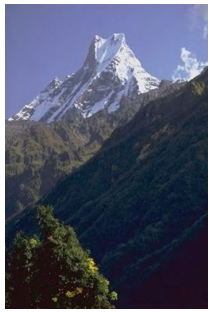

Bus


Dinosaur


Elephant


Flower


Horse


Mountain


Dish

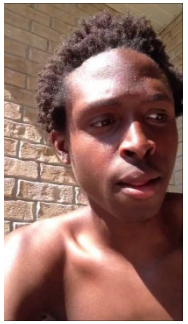
CS4185 Multimedia Technologies and Applications: Course Assignment

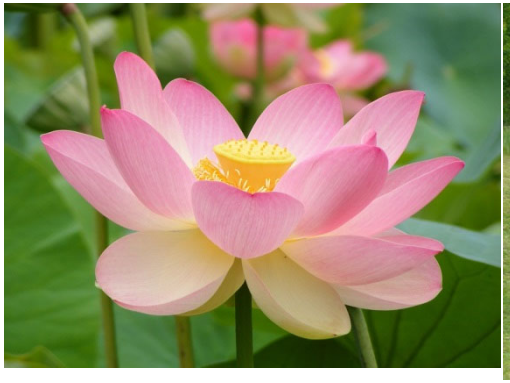
- A database of 1000 images, divided into 10 types, is given. Each type contains 100 images.
  - African: 0 – 99.jpg
  - Beach: 100 – 199.jpg
  - Building: 200 – 299.jpg
  - Bus: 300 – 399.jpg
  - Dinosaur: 400 – 499.jpg
  - Elephant: 500 – 599.jpg
  - Flower: 600 – 699.jpg
  - Horse: 700 – 799.jpg
  - Mountain: 800 – 899.jpg
  - Dish: 900 – 999.jpg

CS4185 Multimedia Technologies and Applications: Course Assignment

- Given 7 example images, students are asked to retrieve relevant images from the database, i.e., retrieving images that belong to the same type as the example image.

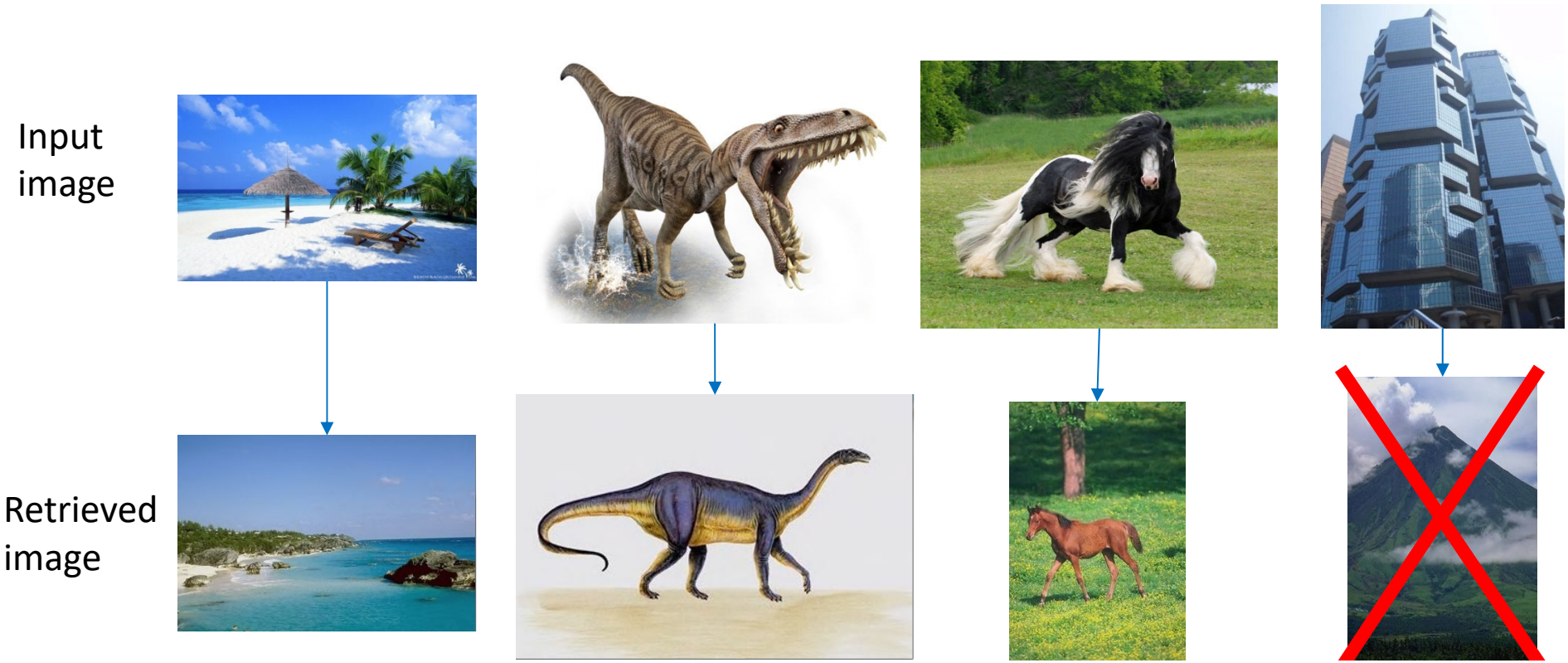







CS4185 Multimedia Technologies and Applications: Course Assignment

- With the given program, only 3 of the 7 example images can find correct best matching images.



CS4185 Multimedia Technologies and Applications: Course Assignment

- Students are asked to extend the program and improve retrieval algorithms.
- This assignment can be carried out as individual or group projects. The maximum number of members in each group is 3.
- There are two levels of requirements for the project, basic and advanced, to cater for students of different backgrounds and interests.

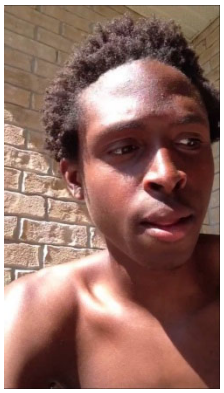
CS4185 Multimedia Technologies and Applications: Course Assignment

- **Basic Requirements (80%)**

  Students are required to finish the following four tasks in the basic requirements:

  1. ***Improve the number of correctly matched images (20%)***
  2. ***Modify the above program to retrieve similar images (20%)***
  3. ***Improve on the Precision (20%)***
  4. ***Improve on the Recall (20%)***

Precision and Recall



- **Precision**: The percentage of retrieved images that are matched.
- **Recall**: the percentage of matched images that are retrieved.

$$precision = \frac{\text{No. relevant documents retrieved}}{\text{Total No. documents retrieved}},$$

$$recall = \frac{\text{No. relevant documents retrieved}}{\text{Total No. relevant documents in the collection}}$$

49 images retrieved, all belong to the Africa category.
So the precision is 100%, and the recall is 49%.

Precision and Recall

- Note: you should use the same set of configurations for all 7 test images, instead of setting different thresholds for different test images like the following:
  - e.g., if input_name == flower.jpg: do something.
- Automatically changing some setting according to the extracted features is allowed.

CS4185 Multimedia Technologies and Applications: Course Assignment

➢ Advanced Requirements (25%)

The extension includes two parts, technical improvement and UI design. The technical improvement may include new retrieval algorithms (e.g., 80+% of precision and 55+% of recall), high dimensional data indexing (efficiently storing and managing the features extracted from the database, modifying the program so that it does not need to compute the features every time), retrieval algorithms for particular types of images (e.g., sunset images, images containing human faces), a crawler to obtain images from the internet, or adding semantic information to help improve the retrieval performance. Here, 15% of marks will be given based on the technical difficulties and another 10% will be given based on the UI design.

CS4185 Multimedia Technologies and Applications: Course Assignment

- Submission Details
  - Due date: ***Nov. 17, 2024***
  - Program
  - Demo
  - Report
- Refer to the document for details

Image Comparison Tutorial

What does this program do?

- Loads an *input image* and 1000 *database images* to be compared with it.
  - Converts the images to grayscale
  - Compares the *base image* with the *database image* using pixel-by-pixel difference.
  - Displays the numerical matching parameters obtained.
  - Displays the input image and the best match result.

Image Comparison Tutorial

- Load and show the input example image, and then convert it to the grayscale.

```
src_input = cv.imread("man.jpg")

cv.imshow("Input", src_input)

# change the image to gray scale
src_gray = cv.cvtColor(src_input, cv.COLOR_BGR2GRAY)
```

Image Comparison Tutorial

- Load the database image, and convert it to grayscale.

```
for img in database:
    # read image
    img_rgb = cv.imread(img)
    # convert to gray scale
    img_gray = cv.cvtColor(img_rgb, cv.COLOR_BGR2GRAY)
```

Image Comparison Tutorial

- Compare these two images, get the pixel difference score, and check if it is better than all checked ones.

```
# find the minimum difference
if diff <= min_diff:
    # update the minimum difference
    min_diff = diff
    # update the most similar image
    closest_img = img_rgb
    result = img
```

Image Comparison Tutorial

- Pixel-by-pixel difference function.

```python
# Compute pixel-by-pixel difference and return the sum
def compareImgs(img1, img2):
    # resize img2 to img1
    img2 = cv.resize(img2, (img1.shape[1], img1.shape[0]))
    diff = cv.absdiff(img1, img2)
    return diff.sum()
```

Image Comparison Tutorial

- Display the best match image, and wait for “ESC” to close the program.

```
print("the most similar image is %s, the pixel-by-pixel difference is %f " % (result, min_diff))
print("\n")

cv.imshow("Result", max_img)
cv.waitKey(0)
cv.destroyAllWindows()
```