

FACIAL RECOGNITION: TEACHING A COMPUTER TO FIND SIMILAR FACES

Signal processing is an enabling technology that incorporates knowledge from a broad range of disciplines including electrical engineering, computer science, mathematics and statistics. Signals can be audio, images, movies, or even data streams such as stock market prices. The goal of most signal processing systems is to convert an electrical signal or data stream into a form that can be easily manipulated. For example, when you use the speech input and output modes on your cell phones, your voice signal is converted to text so that the system can understand and respond to your commands. Similarly, the responses to your commands are converted from text to an audio signal played over the speaker. Signal processing systems use many mathematical, statistical, computational, and linguistic representations to model, analyze, synthesize and discover information.

In this laboratory, we will introduce some basic concepts in a field of signal processing known as image processing. We will specifically show you how to measure the similarity between two images. This lab will introduce you to two important engineering skills used by many engineering disciplines: computer programming and image processing.

The broader goal for this laboratory is to introduce you to two rapidly growing fields: machine learning and big data. *Machine learning*, a branch of artificial intelligence (computer science) that incorporates closely-related fields of statistics (mathematics) and signal processing (electrical and computer engineering), involves the study of systems that can learn from data. For example, a machine learning system could be trained to recognize your facial mood when you are happy or sad by measuring and processing your picture and modeling their typical gesture for each emotion. Machine learning approaches, however, require large amounts of data to learn how such picture vary many dimensions of variability including variations in facial mood and subject population.

Big Data is a blanket term for any collection of data sets so large and complex that it becomes difficult to process using simple database management tools or traditional data processing applications. For example, every day, we create [2.5 quintillion bytes of data](#). To put this in perspective, 90% of the data in the world today has been created in the last two years alone. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. Google attempts to process all data found on the Internet each day, making it easily searchable. This type of problem is often referred to as a big data problem.

The fields of machine learning and big data together allow us to discover trends and to predict behavior. For example, most sports teams today employ statisticians that analyze vast amounts of data from sporting events to predict an athlete's success. This concept was popularized in the movie [Moneyball](#) and the book [The Signal and the Noise](#). How would you predict who will win the World Cup? who will win American Idol? who the next president will be? There are a number of web sites devoted to such topics, including [fivethirtyeight.com](#).

During this laboratory, we will assist you in working through each step of the process. We don't expect you to become experts in any of these technologies after one short lab. Our goal is to help you become acquainted with many engineering disciplines that contribute to our understanding and manipulation of human vision. Remember that you should not be afraid to ask for help if you get stuck on something – we all have been there and have experienced similar problems. No question is too basic.

Introduction to Image Processing

What is an image? An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows. In a typical grayscale image, each picture element has an assigned intensity that ranges from 0 to 255 and is represented using 8 bits of information. An example of such an image is shown to the right in Figure 1. A color image has 24 bits of information associated with it. Each pixel is often represented by a red, green and blue value, or RGB coordinates. Each of these values are represented by 8 bits and can assume values in the range [0, 255]. Therefore, such an image uses 24 bits of information per pixel.

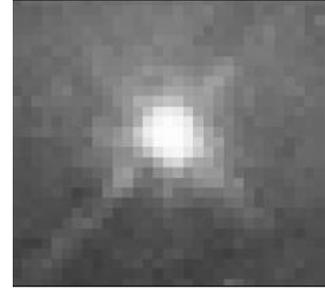


Figure 1. An array or matrix of pixels arranged in columns and rows.

Image data can be quite large. The 12 Megapixel camera in your phone, for example, takes pictures using a resolution of 4000 x 3000 pixels. This requires $12,000,000 \times 24 \text{ bits} = 288,000,000$ million bits of information. Advanced signal processing techniques, such as JPEG compression, are used to reduce this information to a manageable size so your phone does not run out of memory.

Introduction to the Python Programming Language

[Python](#) is a widely used general-purpose, high-level programming language. Its main feature is that it is a scripting language. This means you can type instructions as plain text, store them in a file, and run them directly using a Python interpreter. Though you sacrifice some efficiency, you gain much in terms of flexibility and reduced software development time. There is a wealth of information available on [how to learn Python](#), including the O'Reilly book shown in Figure 2. It is easy to use if you are familiar with basic programming concepts.

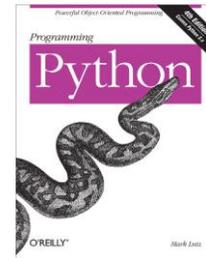


Figure 2. O'Reilly books on are always good places to begin when you want to study a programming language in detail.

The most important thing to remember is that any time you need to figure out how to do something in Python, just do a Google search. You will undoubtedly find that someone has written an entire web page devoted to solving your problem.

As part of your preparation for the lab, we encourage you to review these excellent resources:

- *The Python Tutorial*: walks you through the basics of Python programming;
- *Python Tutorial for Beginners*: a short YouTube video demonstrating how to write a short program;
- *Learn Python*: a nice interactive web site that teaches you the basics.

Python is a popular choice for a programming language today because so many people are using it. There are millions of web pages devoted to its use and you can generally find a package that was developed for about any task. You can write many different types of code using Python. In this image processing laboratory, we will focus on doing some simple things like reading images from a file and doing some simple computations on these images.

Python Installation

Before we can begin, we need to setup the Python environment required to run the scripts we will use in this workshop. We have already setup the required modules needed to run the scripts provided and have hosted the environment on the web. First, open the following URL in your browser: http://www.isip.piconepress.com/projects/we2/2016_summer/. Download the file we2_v00.zip file hosted

on the website. A zip file is an archival file format that allows users to compress multiple files into one. To get the original files we need to unzip the file. An open-source program we will use to extract the files is 7zip.

Go to your Downloads folder, and right click the we2_v00.zip file. In the context menu that pops-up click "7zip" -> "Extract to we2_v00". A new folder named we2_v00 will be created and contain the contents of the zip file you downloaded.

Open the we2_v00 folder. We need to open a cmd window in this location. A quick way to open a cmd window in a specific location is to hold shift and right click in the white-space within the file explorer window. In the context menu that pops-up, click on "Open command window here". You are now ready to begin your first task.

Task 1: Hello World

Write a simple program to display the text "Hello World" on your computer screen. Open a cmd window and change directories to the location of the tools using the cd command. Open up a plain text editor and follow [these instructions](#), saving the file to we2_hello.py. To run this program, type:

```
..\python\python we2_hello.py
```

This is how we run python from the command line.

Task 2: Opening and Reading Images

The next step is to open your image files and read the data. Again, using your text editor, open the file named "we2_reading_image.py". Execute this code using the following command:

```
..\Python\python.exe we2_read_image.py data/jp_gates_original.png
```

You should see the image displayed on your screen as the output. Read the code and study each step, particularly the line containing the function call "cv2.imread()".

Task 3: Finding the Similarity Between Two Images

Take two arbitrary images and compare them to determine if they are identical, or near identical in some way. We can accomplish that using the Mean Square Error (MSE) algorithm, which is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Open up we2_mse.py in text editor. The example program contains a function that takes name of two images ("imageA" and "imageB") as an input. The output is value of the mean square error between these two images. In Python we calculate this as follows:

```
# sum of the squared difference between the two images;
#
Sum_diff = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)
err = ...
```

Save your changes and run this program as follows:

```
..\python\python we2_mse.py data/jp_gates_contrast.png data/jp_gates_original.png
```

Modify the code to compute the mean square error between two images.

Task 4: Compare Two Images

The `we2_compare.py` is a script that displays the MSE between different images. The previous modules you studied are used to implement the comparison function. Read the code and study each step to figure out how these modules combined together.

Task 4: Find Your Doppelganger

Next, we are going to integrate what you have learned to run an image processing experiment. Let's find out who is the most similar person to you in the class. Open up "`we2_most_similar_images.py`" in a text editor. The example program contains some lines that save pictures in an array. The code is shown below:

```
# loop through all of the images in the specified directory and save them in
# a variable
#
for filename in os.listdir(image_dir_a):
    data.append(filename)

# read the reference image and transform into the gray scale
#
original = cv2.imread(image_ref_a)
original = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)

# iterate through all the images in the directory and compare them to the
# reference image
#
for i in xrange(0,len(data)):
    ...
```

Modify this code to find the picture that has the minimum MSE with your picture.

Download the pictures of all your classmates from the following link:

<https://www.dropbox.com/sh/ydetabxp6foksmw/AAA-YuXiawoxeZocRQlaHz0Oa?dl=0>

Save your changes and run your program this way:

```
..\Python\python we2_most_similar_image.py data/ data/jp_gates_contrast.png
```

Summary

In this laboratory, we have shown you some basic computer programming using the Python programming language. Software, particularly computer simulation, plays a very important role in modern engineering. Interpretive languages such as Python are fairly easy to learn once you understand some basic concepts of how computer programs work and how we manage computer data. In our undergraduate studies you will have an opportunity to learn more about computer programming, and more about the mathematics demonstrated here to compare the information content in signals such as images.

To learn more about these fields, we encourage you to visit our research laboratories web pages: www.isip.piconepress.com and <http://www.nedcdata.org/>.