



National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS370: Artificial Intelligence
Class: BSCS-7AB

Lab 2: Offline Signature Verification

Date: January 30th, 2019

Time: 10am-01pm & 2pm-5pm

Instructor: Dr. Dr Imran Malik



Introduction

The purpose of this lab is to get familiar with offline signature verification and perform initial verification routines by extracting basic features.

Objectives

After completing this lab, students will be able to understand how to:

- Process images in Python
- Extract features from signatures

Software Tools/Requirements

- Solutions should be made in Python
- Use PIL or OpenCV

Prerequisites

Before you begin working on this lab, make sure that you have:

- Set up Python on your machine
- Installed either Python Imaging Library (PIL) or OpenCV.
- Downloaded the 4NSigComp2010 dataset from [http://www.iaprtc11.org/mediawiki/index.php/ICFHR_2010_Signature_Verification_Competition_\(4NSigComp2010\)](http://www.iaprtc11.org/mediawiki/index.php/ICFHR_2010_Signature_Verification_Competition_(4NSigComp2010))

Lab Task

Take one Ref signature from downloaded dataset¹ and perform the following tasks:

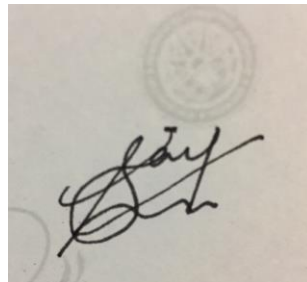
1. Develop a bounding box around the signature content.
2. Find out the centroid of the signature.
3. Segment signature from centroid vertically and horizontally (the signature will be divided into four pieces)
4. Calculate black to white transitions for each of the four segments.

¹ In case you are unable to download the dataset, you can use any image of a signature to perform this lab.



Description

In this section, we will walk through the lab tasks and see how to complete them. This walkthrough is to provide you a starting point for your own implementations, and hence is abstract and leaves out many implementation details. You are supposed to figure them out yourself. We will use the following signature as a reference.



Preprocessing

Convert the signature to a binary (black-and-white) image before extracting features. With PIL, this can be done by using the `Image.convert()` method. For further information on how to use PIL, refer to PIL handbook at <http://www.pythonware.com/media/data/pil-handbook.pdf>. At this point, your signature image should look like this:



Task 1: Developing a bounding box

In this task, you have to locate the signature in the image and develop a bounding box around the signature content only, ignoring the white space around it.

Algorithm

```
left := width
right := 0
top := height
bottom := 0
for x in (0, width) and y in (0, height)
    color := image.getpixel(x, y)
    if color is 0
        if x > right
            right := x
        if x < left
            left := x
        if y > bottom
```



```
        bottom := y  
    if y < top  
        top := y
```

After completing this task, your signature image should look like this:



Task 2: Locating the centroid

Centroid is the point where center of mass of the signature image is located. It can be computed using the algorithm given below.

Algorithm

```
cx := 0  
cy := 0  
n := 0  
for x in (0, width) and y in (0, height):  
    if image.color(x, y) is 0:  
        cx := cx + x  
        cy := cy + y  
        n := n + 1  
cx := cx / n  
cy := cy / n
```

Task 3: Dividing the image at centroid to create four segments

You should have the following at this point:

1. Bounding box = (left, right, top, bottom)
2. Centroid = (cx, cy)

You now have to divide the image into four segments. The four segments can be computed by locating the boundaries of each of the four segments:

1. (left, cx, top, cy) are the boundaries of top-left segment
2. (cx, right, top, cy) are the boundaries of top-right segment
3. (left, cx, cy, bottom) are the boundaries of bottom-left segment
4. (cx, right, cy, bottom) are the boundaries of bottom-right

segment After completing this task, your signature image should look like this:



Task 4: Finding black to white transitions

In this lab task, you have to find black to white transitions for each of the four segments/ blocks of the signature you made in Task3. That is, you have to calculate the number of white pixels in the neighborhood (use 8-connectivity) of each black pixel in the image. For details about pixel connectivity, refer to your famous friend [Google](#)

Algorithm

```
prev := image.color(0,0)
n := 0
for x in (1, width) and y in (1, height):
    curr = image.color(x,y)
    if curr is 255 and prev is 0
        n := n + 1

    prev := curr
```

Completion check

On successful completion of the lab tasks, you should have the following:

1. Bounding box of the signature in image
 - a. B = (left, right, top, bottom)
2. Coordinates of centroid of the signature
 - a. C = (cx, cy)
3. Four values of black to white transitions, for each of the four segments
 - a. T = (TL, TR, BL, BR)

Deliverables

Submit single word file having source code and output on LMS. Your submission should follow the following naming convention: YourName_RegNo_Section_lab02



National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science