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

CS370: Artificial Intelligence

Class: BSCS-7AB

Lab 4: Offline Signature Verification

Date: 12-02-2020

Time: 10am-01pm & 2pm-05pm

Instructor: Dr. Imran Malik

Lab Engineer: Aftab Hussain

Introduction

The purpose of this lab is to get familiar with offline signature verification and perform initial verification routines by extracting basic features. This is an extension of previous lab, which you should have completed before beginning this lab.

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 for image processing
- numpy

Prerequisites

Before you begin working on this lab, you should have completed lab 2. If you have not done it yet, now is a good time to start. On successful completion of previous lab, 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)

Lab Task

In this lab, you will modify your own implementations of last lab to perform the following tasks. Additions to the previous lab are highlighted.

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. **Repeat steps 2-3 until you have segmented the image into 64 cells.**
5. Calculate black to white transitions for each of the 64 cells.
6. **Calculate aspect ratio of each cell.**

You should perform the aforementioned tasks for each of the images in TestSet/Reference folder in the dataset you downloaded in last lab. Save extracted features in text files using numpy.

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.

At the end of last lab, we had the following signature:

Now, we will extend our solution to divide the signature into 64 cells instead of 4, for each cell, extract the following features:

1. Centroid
2. Black to white transitions
3. Aspect ratio (= width/height)

After extracting these features from each cell, dump them to text files. You may use numpy's dump function for this, or any method of your choice. Repeat this process for each of the reference signatures. This means that, for each reference signature, there would be three files, each with 64 entries, one for each cell.

Task 1: Dividing the signature into 64 cells

Extend your solution from previous lab where you divided the image into four segments after calculating bounding box and centroid, to divide it into 64 cells. This can be done either iteratively or recursively.

```
Algorithm
split(image, left, right, top, bottom, depth=0):
      cx, cy = findCentroid(image, left, right, top, bottom)
      if depth < 3:
            split(image, left, cx, top, cy, depth + 1)
            split(image, cx, right, top, cy, depth + 1)
            split(image, left, cx, cy, bottom, depth + 1)
            split(image, cx, right, cy, bottom, depth + 1)
      else:
            t = findTransitions(image, left, right, top, bottom)
            r = findRatio(left, right, top, bottom)

            # save centroid, transitions and ratio to file
```

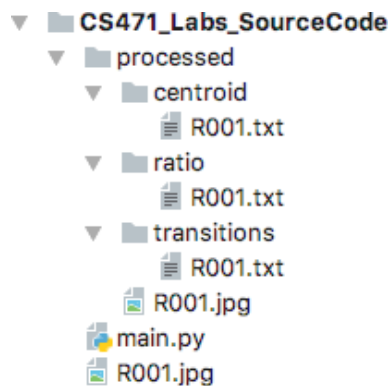After completing this task, your signature image should look like this:
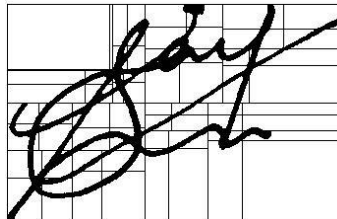
Completion check

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

1. For each of the Ref signature images
    a. For each of the 64 cells
        i. Coordinates of centroid, C = (cx, cy)
        ii. Black to white transitions, T
        iii. Aspect ratio, R = width of cell / height of cell

The feature sets calculated above should be dumped in text files, using numpy or otherwise. Your project files would be something like:

```
▼ 📁 CS471_Labs_SourceCode
    ▼ 📁 processed
        ▼ 📁 centroid
            📄 R001.txt
        ▼ 📁 ratio
            📄 R001.txt
        ▼ 📁 transitions
            📄 R001.txt
        🖼 R001.jpg
    🐍 main.py
    🖼 R001.jpg
```

Each of the .txt files contains values of a particular feature for each of the 64 cells. The screenshot shows the output after processing one signature, R001.jpg. You have to repeat the tasks for all Ref signatures in the dataset.

Deliverables

Submit your source code on LMS in a compressed file. Your submission should follow the following naming convention: YourName_RegNo_Section

Deadline: Before the next lab.

Do you have any problem in understanding? You are allowed (in fact encouraged) to discuss in the lecture.