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

CS370: Artificial Intelligence

Class: BSCS-7AB

Lab 5: Offline Signature Verification

Date: 19-02-2020

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

Instructor: Dr. Imran Malik

Lab Engineer: Aftab Hussan

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 and 3. If you have not done them yet, now is a good time to start. On successful completion last labs, you should have the following:

1. For each of the 64 cells
   a. Coordinates of centroid, C = (cx, cy)
   b. Black to white transitions, T
   c. Aspect ratio, R = cell-width / cell-height of cell

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. For each of the 64 cells, calculate the following features:
   a. Centroid
   b. Black to white transitions
   c. Aspect ratio
   d. Number of black pixels
   e. Normalized size
   f. Angle of inclination of centroid
   g. Normalized sum of angles of inclinations of black pixels

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

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 extracted three features:

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

Now, we will extend our solution to extract four other features.

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: Number of black pixels

In this task, you have to count the number of black pixels in each cell of the signature. This can be done simply by iterating over all the pixels in the cell and counting pixels with color value 0.

Task 2: Normalized size

In this task, you have to compute size of each cell and normalize it with number of black pixels. This can be computed with the following formula:

$$s = \frac{cellWidth * cellHeight}{blackPixels}$$

Task 3: Inclination of centroid

In this task, you have to compute the angle the line from centroid to bottom-left corner of the cell makes with x-axis. This can be computed with the following formulae:

$$dx = cx - left$$

$$dy = bottom - cy$$

$$a = \arctan(\frac{dy}{dx})$$

Task 4: Normalized inclination of black pixels

In this task, you have to repeat task 3 for (x, y) coordinates of each of the black pixels, and sum all the angles. Then, normalize the sum of angles by dividing it with black pixels. This can be computed using the following formulae:

$$A = \frac{1}{N}\sum_{x=0}^{w}\sum_{y=0}^{h}\arctan(\frac{dy}{dx})$$

where N is the number of black pixels, and arctan is calculated for each of the black pixels only.

Recursive algorithm

Once you have successfully completed all the tasks above, your main algorithm would look something like this:

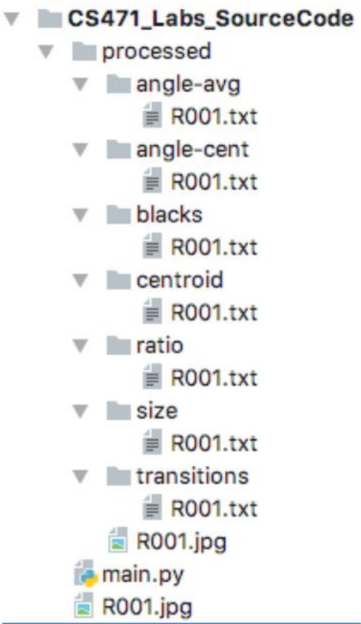
```
Algorithm
split(image, left, right, top, bottom, depth=0):
      cx, cy = findCentroid(image, left, right, top, bottom)
      if depth < 3:
            split(image, left, cx, top, cy, depth + 1)
            split(image, cx, right, top, cy, depth + 1)
            split(image, left, cx, cy, bottom,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

            b = findBlacks(image, left, right, top, bottom)
            s = findNormSize(left, right, top, bottom, b)
            a = findCentAngle(cx, cy, bottom, left)
            A = findNormAngles(image, left, right, top, bottom)

            # save features to file
```

Completion check

On successful completion of the lab tasks, you should have the following feature sets for each of the 64 cells in all the Ref signature images:

1. Centroid of cell
2. Black to white transitions in cell
3. Aspect ratio of cell
4. Number of black pixels in cell
5. Normalized size of cell
6. Inclination angle of centroid
7. Average inclination angles of black pixels

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



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 & **also word file having code and output screenshots**. Your submission should follow the following naming convention: YourName_RegNo_Section

Deadline: Before next lab.

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