Pattern Recognition Project Presentation

The Blue Guys

Group Members

Ioannis Glampedakis

Antonios Chaidaris

Finsik Mengjiqi

Hamed Hemati

KWS - Implementation Details

Implemented in Python

Used external libraries

KWS

Things we did:

1 - Find a way to cut the segmentations (used PIL)

2 - Find appropriate features for describing images

3 - Using a library for doing DTW

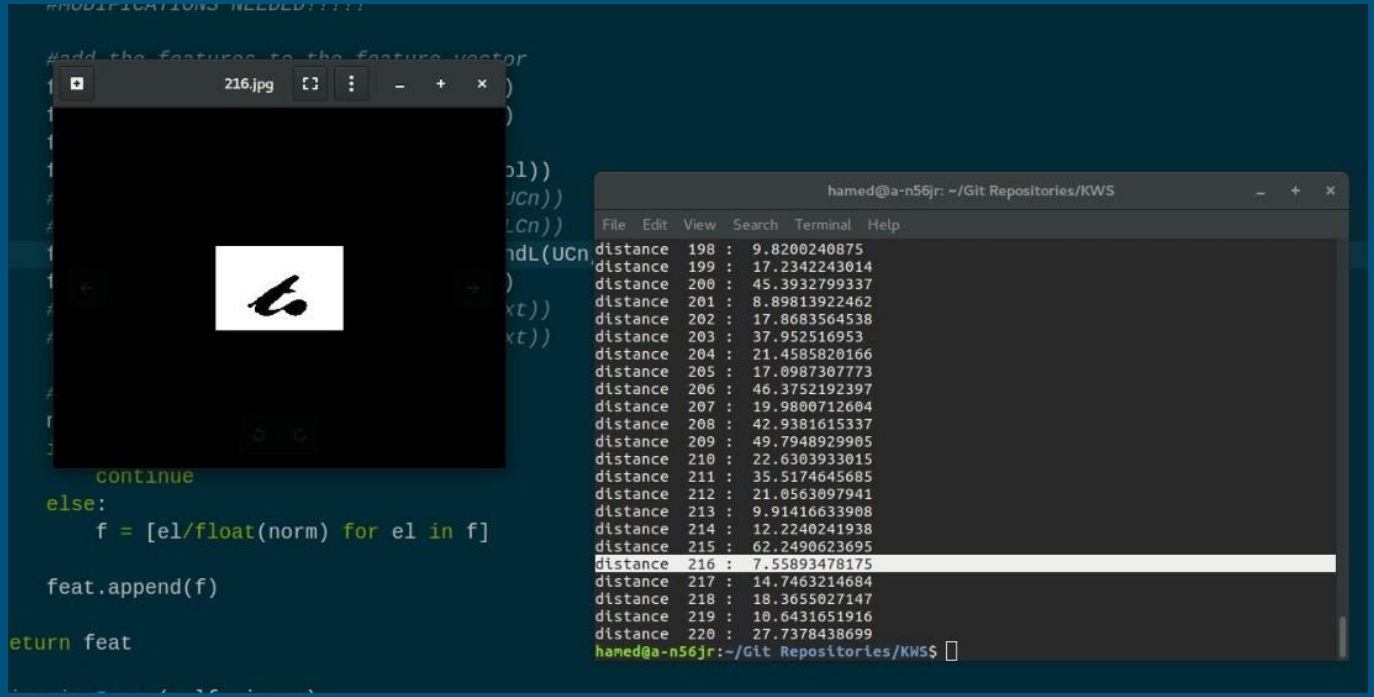
4 - Getting it all together

KWS - Feature Vector

Used Features:

- Fraction of black pixels
- Upper contour
- Lower contour
- Number of transitions
- Fraction of black pixels between lower half and upper half

KWS - Features



KWS - DTW Library

- Learnt the theories behind DTW first
- Used fastdtw 0.3.0 for the DTW part

Signature Verification

Collect biometric data as the users are signing at certain time intervals

Data: coordinates, pressure, pen-up/down, angles

Signature Verification

Problem: Even the same person can take variable amounts of time to make a signature.

Difficult to compare uneven sets of data.

Dynamic Time Warping

Solution: Dynamic Time Warping is a technique to measure the distance between two uneven temporal sequences.

Data Preparation

- Read data from file save them in an nx7 matrix.
- N is variable depending how much time the signatory need to sign
- 7 is the number of the original features.
- Data
  - Time
  - X, Y coordinates
  - Pressure
  - Pen-up
  - Azimuth
  - Inclination

Feature extraction

- Calculate speeds for each coordinate on every time-stamp
- Features used:
  - X, Y
  - Speed X and Y
  - Pressure
- Final matrix n x 5

Preprocessing

Rescaling:

$$RescaledX = \frac{x - min(x)}{max(x) - min(x)}$$

Distances

- For each signature to be verified we calculate its distance from each of the original signatures of the particular signatory.
- Calculate the mean.