ICFHR 2012 - Arabic Writer Identification

Identify which writer wrote which documents.

$1,000 · 42 teams · 6 years ago

Competition Data

Edit

- sample_entry.csv
- features.7z
- features.zip
- images.7z
- images.zip
- matlab_benchmarks.7z
- matlab_benchmarks.zi...

images.7z 5.62 MB

Download

Data Description

Additional 1-Nearest-Neighbor benchmark in Python

In this contest, more than 200 writers were asked to write three different paragraphs in Arabic language. The first two paragraphs are used for training and the third one for testing. For some writers, the first two paragraphs have been removed from the training set to test the ability of systems to detect unknown writers. Also, some writers have written the third paragraph more than once and some other participants did not writer the third paragraph at all.

Images are provided in PNG binary format. The binarization has been performed using Otsu's method. The images are provided in two subfolders “train” and “test”. The folder “train” contains images having the following format XXX_Y.png where XXX represents the ID of the writer and Y represents the number of the paragraph. The folder “test” contains images of the third paragraph, they all have the following format ZZ.png, where ZZ is the ID of the image in the test set.

Participants are asked to provide for each ZZ image, the ID of the most probable writer (among those which are in the training set). Like mentioned before, some ZZ images do not actually correspond to any writer in the training set. Participants are supposed to produce an ID eqal to 0 for those images.

The competition is judged using the caterogization accuracy, which corresponds to the percentage of correctly identified writers.
In the event there is a tie on the private leaderboard, the tie will be broken by the submission time (the prize money will go to the first submission).
For participants who are not familiar with image processing, some geometric features are provided. These features are given in the form of histograms. The list below shows the name and the number of values in each of these features:

- LengthsOfBranchesHist_10 (10 values)
- ThicknessLengthsCircleHist30 (30 values)
- tortuosityHist10 (10 values)
- tortuosityDirectionHist10 (10 values)
- tortuosityDerivateHist10 (10 values)
- tortuosityDerivateDirectionHist10 (10 values)
- DirectionPerpendicular5Hist10 (10 values)
- CurvaturePerpendicular5Hist100 (100 values)
- CurvatureAli5Hist100 (100 values)
- CurvaturesDerivateAli5Hist100 (100 values)
- CurvatureAli10Hist100 (100 values)
- CurvaturesDerivateAli10Hist100 (100 values)
- CurvatureAli15Hist100 (100 values)
- CurvaturesDerivateAli15Hist100 (100 values)
- CurvatureAli20Hist100 (100 values)
- CurvaturesDerivateAli20Hist100 (100 values)
- chaincodeHist_4 (4 values)
- chaincodeHist_8 (8 values)
- chaincode8order2_64 (64 values)
- chaincode4order2_16 (16 values)
- chaincode4order3_64 (64 values)
- chaincode8order3_512 (512 values)
- chaincode4order4_256 (256 values)
- chaincode8order4_4096 (4096 values)
- distribution_types_73 (73 values)
- distribution_types_differences_73 (73 values)
- directions_hist1_4 (4 values)
- directions_hist2_8 (8 values)
- directions_hist3_12 (12 values)
- directions_hist4_16 (16 values)
- directions_hist1a2_12 (12 values)
- directions_hist1a2a3_24 (24 values)
- directions_hist1a2a3a4_40 (40 values)

Participants are free to use these features or their own features or even a combination of both.
The submissions must be in a 2 columns format: The first one contains in ID of the questioned document and the second one, the ID of the most probable writer (or 0 in case of unknown writer). Please see sample_entry.csv for an example.
Finally, a sample code implementing the Edge-Based Directional Features in matlab is provided along with the corresponding benchmarks (cf. Bulacu et al. 2003' article for further details about the method).

© 2017 Kaggle Inc