Automatic Signature Analysis and Verification using Local Features

●●●

Team:

1. Mohit pavan kumar - 2018102016
2. Tejaswini suma - 2018102018
3. Raja mavullu - 2010102032
4. Gowthami gongati- 2018102048

Dataset-4NSigComp2010(1/3)

- The collection contains offline signature samples. The signatures were collected under the supervision of Bryan Found and Doug Rogers in the years 2002 and 2006, respectively.
- The images were scanned at 600 dots per inch resolution and cropped at the Netherlands Forensic Institute for the purpose of 4NSigComp2010 signature verification competition.
- The signature collection done for training contains 209 images.
- The signatures comprise 9 reference signatures by the same person and 200 signatures for training .
- The 200 train signatures comprise
  - 76 genuine signatures written by person in his/her normal signature style.
  - 104 simulated/forged signatures
  - 20 disguised signatures written by the reference writer.

Dataset-4NSigComp2010(2/3)

- The signature collection for testing contains 125 signatures.
- The signatures comprise 25 reference signatures by the same person(but different from that of person from training set)” and 100 questioned signatures.
- The 100 questioned signatures comprise
  - 3 genuine signatures written by the person in his/her normal signature style
  - 90 simulated signatures
  - 7 disguised signatures written by the reference writer.
- The signatures of the training set are arranged according to the following folder structure:
  - Disguised: Contains all the disguised signatures of specimen
  - Genuine: Contains all the genuine signatures of specimen
  - Reference: Contains the reference signatures of specimen which can be used for training purposes
  - Simulated: Contains all the skilled forgeries for specimen

Dataset-4NSigComp2010(3/3)

- The signatures of the test set are arranged according to the following folder structure:
  - Reference: Contains the reference signatures of another specimen(different from that of training specimen)(used for testing classifier)
  - Questioned: Contains all the signatures for which the task is to verify signature of another specimen.
  - Correct Answer Key: The key containing details about the type of signatures for specimen author ‘B’, i.e., disguised, forged, or genuine.
- Note:-The naming convention also reveals the type of signature for the training set.
  - D023: a disguised signatures
  - G003: a genuine signature
  - S001: a forged/simulated signature
  - R001: a reference signature.

Local Stability analysis

- There are 2 types of analysis,, global(looking at whole signature and analysing it together) and local(which focuses on finer details in the signature to analyse)
- This paper uses Local stability analysis because local analysis helps better in analysis disguised signature(In this type of signature, majority of the signature is kept same and only some of the finer details are changed(like strokes etc))
- We here analyse local stability using SURF(Speeded Up Robust Features) which detects blobs in the image using the approach of box filer(hessian matrix) instead of gaussian used in SIFT(Scale Invariant Feature Transform)(comparison is given in the next slide)
- There are 2 hypotheses assumed to be true for analysis but also verified using SURF. They are
  - Some key points give better stability than others i.e, the image isn’t homogeneous(not all keypoints are same)
  - The stability behaviour observed is generalized for other persons

SIFT vs SURF

- Implementation process of SIFT consists stages as follows-
  - Scale space extreme detection where the key points of images are detected and it Eliminates Unreliable key points
  - Direction of keypoints where finding the gradient of magnitude to those selected keypoints and results of calculated feature of keypoints is called feature descriptors.
  - Matching of features of two different images. At first features are extracted from test image using SIFT Algorithm. These features are matched with SIFT features which are obtained from the training image. Matching of these two features is done by a Euclidean-distance based nearest neighbor approach
- SURF is similar to SIFT in performance and reduces the computational complexity.
- SURF detects image keypoints and generates descriptors. Based on Hessian matrix, SURF obtains the keypoints. Simplify the operation and helps to reduce in computational cost by applying appropriate filter to the integral image. Haar wavelet responses in x and y direction are computed to determine the orientation. Based on integral image and Hessian matrix, robust keypoint descriptors are detected by using SURF algorithm. Finally key points of two face images are matched for the purpose of recognition.

Local Stability analysis algorithm

- we took one genuine signature(let it be x), from the set and extracted key points from the remaining genuine signatures of that person. This made the reference signature database for comparing the distances of key points of signature 'gen_sig' with that of its database.
- SURF keypoints are extracted from signature 'x' and these distances are compared with the concerned key-points database. The Euclidean distance of every key point is calculated from the database and is assigned to a matrix.
- We repeated this in an 'n-1 cross validation' manner and get the distances of each keypoint from the concerned reference key points dataset.
- After normalizing the values, we took the sum of these values as the actual color of a bin in the histogram where each bin contains the sum of individual keypoint distances from the reference authors dataset for that author.
- White color-no keypoints, green color-stable region(min distance), red color- maximum distance to reference database keypoints.

Signature Verification :

1. Compute kepoints from the genuine set leaving one signature and find temporary key points database
2. Find the average distance from the keypoints and mark the distances less than average as green and remaining red where green represents stable areas
3. Repeat the same for n specimen in n-1 cross validation manner
4. Once a query occurs then find the key points using surf for that and compare with the database found early one by one, and calculate the distance and if it is less than threshold keep it .
5. Finally calculate the probability for the signature to be genuine considering total keypoints in the query and number of key points matched

Train and Test:

1. Based on the probabilities we got we will 1st classify whether it is genuine or not and if it is not then we will classify it as disguised or forged
2. Here if it is not genuine then we will classify it as disguised or forged :
   a. In forged the distances will be lesser than disguised because forged will be kept perfectly fine
   b. But in disguised the distances and the keypoints are intentionally kept wrong so less probability based on this forged and disguised will be calculated
3. Here in train we will never use forged because the key points will not be correct in the set as it is not done by the same person we will use points which are kept by the same person ie. disguised and genuine
4. The testing error rate expected is 15% and 5% more or less can be achieved by this technique

We have done till the keypoint extraction using inbuilt surf functions and tried to create the database the errors are yet to be fixed
The train and test using database is remained , we will try to complete it within the time but the lag is definitely there according to our schedule we will fulfill that soon

Problem facing :

1. Sift and surf are patented to use now (from february) so we found the following alternatives :
   a. Using virtual environment (which uses python 3.5 and corresponding cv version)
   b. Add dependencies used for making sift and surf (not sure if it is possible)
   c. Use the other softwares like brisk , akaze or kaze (not as efficient as sift and surf so didn't try waiting to ask permission)
   d. Till today we tried rough implementation and we are facing few errors, We will be happy to take any valuable suggestions.

THANK YOU