Final Project: Fingerprint Spoof Detection

Abstract: A fingerprint spoof detector is a pattern classifier that is used to distinguish a live finger from a fake (spoof) one in the context of an automated fingerprint recognition system. The aim of this project is to develop fingerprint spoof detector.

Database: LivDet 2011 database consisting of live and spoof images fabricated using five materials such as Silgum, Latex, Gelatin and Ecoflex separated into training and test sets (non-overlapping set). The original images from these databases and features such as LBP, BSIF and BGP are extracted from live and fake images using four different fingerprint sensors. The database can be downloaded from the following link: https://umkc.box.com/s/ncuocvn24jrvw51p7xplhmuapu1z396t

The aim is to develop fingerprint spoof detector based on CNN (Project 1). Students who do not have computing capabilities for running deep learning can choose to work on Project 2. Both the projects 1 and 2 are described as follows:

Project #1: Development of fingerprint spoof detector based on CNN using the training set consisting of live and spoof samples from various materials. Use the testing set to evaluate the performance of the spoof detector and generate training, validation and testing ROC Curves. ROC Curve is plotted for Spoof Accept Rate vs Live Reject Rate.

Task 1: Implement the fingerprint spoof detector using pretrained CNN using AlexNet. The pretrained CNN model is going to be re-purposed to implement spoof detector. Use the training live and spoof images from all materials from LivDet2011 database to implement the classifier. The trained classifier will be evaluated using testing part of the database. The images from all the sensors will be combined together.

- As the AlexNet is trained on RGB images of size 227 x 227. Images from LivDet dataset will be resized (imresize (I, [227 227])). For grayscale images, replicate the image 3 times to create an RGB image (I = cat (3, I, I, I))
- Use CNN for feature extraction from training and test images. The extracted features will be used with regular NN for classification.

Please refer the following example that use AlexNet for multi-class classification:
https://www.mathworks.com/help/vision/examples/image-category-classification-using-deep-learning.html

Task 2: Train the spoof detector based on CNN (as mentioned in Task 1) using live and spoof samples from four fabrication materials only from training set and test the spoof detector using the live samples and spoof samples from the fifth material in the testing set.

Bonus points: For Task 2, reporting which four combination of spoof materials when used for training, resulted in highest accuracy on the fifth spoof material when used for testing.

Project #2: Development of fingerprint spoof detector using Deep network construction based on Autoencoders. Use features such as LBP, BSIF and BGP extracted from training and testing live and spoof images. The performance of the classifier will be evaluated by reporting training, validation and testing ROC Curves. ROC Curve is plotted for Spoof Accept Rate vs Live Reject Rate.

Task 1: Implement the fingerprint spoof detector using Deep network construction using Autoencoders. Features such as LBP, BSIF and BGP from Digital and Sagem fingerprint sensors will be used for training and testing the deep network. These extracted features along with the class labels are provided to you. For example, Train_All_Data_DigiBSIF.mat consist of BSIF features extracted from training live and spoof images (all the materials) from Digital sensor. Train_All_Label_Digi_BSIF.mat consist of corresponding labels (live or spoof). Similarly, Test_All_Data_DigiBSIF.mat consist of BSIF features extracted from testing live and spoof images (all the materials) from Digital sensor. Further, Train_All_Data_SageLBP.mat consist of LBP features extracted from testing live and spoof images (all the materials) from Sagem sensor.

-Use these features individually to train and test the deep network. Data from Digital and Sagem sensor will be combined together. N number of autoencoders will be stacked together for optimum performance. Report the performance individually for each of these features.

Please refer the following example as a reference: https://www.mathworks.com/help/nnet/ug/construct-deep-network-using-autoencoders.html?searchHighlight=deep%20network%20using%20stack%20of%20autoencoders

Task 2: Implement the spoof detector using Deep network construction using Autoencoders as mentioned in Task 1 individually for each feature and each sensor. Specifically, the Deep Network will be trained on only one kind of feature from one sensor only and tested on same features from Different Sensor. For example: using

Train_All_Data_DigiBSIF.mat for training and Test_All_Data_Sage_BSIF.mat for testing and reporting the performance degradation.

Bonus points: Evaluating fusion of features (such as LBP and BSIF) and reporting the combination resulting in highest accuracy for the Task 1.

Deliverables: Please submit your original individual work in terms of a PPT plus all your MATLAB code electronically here (via blackboard ONLY). Specifically, the following should be submitted.

- An original and well documented matlab code (40 points)
- Power point presentation illustrating (60 points)
    - The steps adopted in execution of the project (20 points)
    - ROC curves for training, validation and test set. Also, report Spoof Accept Rate at 0.1%, 0.5% and 1% Live Reject Rate (30 points)
    - Conclusions drawn (10 points)
- Bonus points (5 points)