Face Anti-Spoofing Architectures

A thesis submitted in partial fulfilment of

the requirements for the degree of

Bachelor of Technology

by

Shubham Lohiya (150102064)

Shubham (150102079)

Under the guidance of

Dr. Kannan Karthik

Associate Professor,

Department of EEE, IIT Guwahati

DEPARTMENT OF ELECTRONICS & ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

November 2018

Abstract

Facial anti-spoofing has been essential in recent times, with the natural integration of biometric-based access control systems, either based on fingerprint or face in smart phones. The same anti-spoofing frame (implemented at a slightly advanced level), is also required in unmanned surveillance stations to detect the presence of disguises or prosthetics deployed by illegal traffickers to avoid being snapped by hidden cameras. In this work, we have modified & integrated existing techniques for face recognition in order to achieve the task of real/spoof image classification. This work consists of two modules namely Image Quality Assessment (IQA) [1] exploiting the sharpness profile of the input image consisting of pixel-difference, similarity and edge based measures, second module exploits the textural features of an input images by extracting local binary pattern(LBP) and transition local binary pattern(t-LBP) [2] histograms(8 bins) of entire image and concatenating them to form 16 dimensional feature vectors. These features (LBP & t-LBP) complement each other and helps in order to achieve a better classification result. The experimental results, obtained on publicly available CASIA face anti-spoofing dataset showed classification accuracy of 96% for the first module and an accuracy of 89% for the second module. The results exhibited in this work proves that the examination of the quality of genuine face images uncovers profoundly significant data that might be proficiently used to separate them from spoofedimages. Due to the computational simplicity of these modules, makes it appropriate for real-time applications.

Problem Statement

There are two modes in which anti-spoofing can be performed:

1. Identity independent setting in which the anti-spoofing algorithm has no prior idea regarding the subject in the facial snapshot or video presented to the camera.

2. Reference based anti-spoofing wherein a person may claim to be someone else by presenting a prosthetic of that individual's face. Here, the anti-spoofing system has prior information (pre-stored genuine images available) regarding the subject who may have been impersonated.

The identity independent problem is less challenging from a technical viewpoint as compared to the reference based anti-spoofing problem. In the case of the latter, one has to deliberately ignore recognition based features and focus on "condition/acquisition-specific features" to detect some form of spoofing.

In our problem, we propose to develop a reference based anti-spoofing system for a closed unmanned authentication system, implemented for an organization. Natural full frontal poses under different lighting conditions will be stored in the database for different subjects. When a person presents his face (natural or disguised as someone else) to the camera, claiming to be subject-X, multiple snapshots will be taken and then a naturalness check will be done by comparing the test-snapshots with the pre-stored natural facial images of that specific subject-X.

Once the base-feature set for anti-spoofing is designed and calibrated, the problem becomes tantamount to an outlier detection algorithm, assuming that there is sufficient information in the database to learn the statistical model for subject-specific naturalness (from the point of view of the face). Any attempt to produce a spoofed version of the face should be detected by this anti-spoofing algorithm by treating this test-query set as an outlier. Ideas involving 1-class SVM and other anomaly detection algorithms will be explored.

Introduction

Spoofing attack is the act of outwitting a biometric sensor by presenting a counterfeit biometric evidence of a valid user. Face spoofing attacks are attempted in 3 major ways:

1. Print/Photo Attack: The attacker uses victim's photo and display it using digital device or in printed form to outwit the biometric sensor.
2. Replay/Video Attack: The attacker uses looped video of victim's face and display it using digital device to outwit the biometric sensor.
3. 3-D Mask/Prosthetic Attack: The attacker uses a mask of victim's face to outwit the biometric sensor.

These attacks have been mentioned in increasing order of their sophistication.

Attacks that involve showing of image on a 2D device or a planar surface can easily be learned by classifier as these type of spoofs boils down to image manipulation.

Methodology - 1 (Involving Image Quality Assessment)

In our solution, the original image is compared with a image processed with Gaussian filter having certain variance. Further, image quality measures are extracted from the given image that form the feature vector for the corresponding image. SVM classifier is trained using the computed feature vector and the label provided to the training example. Now given a query image, image quality measures are used to generate feature vector which is used against the trained SVM model. The trained SVM will classify the given query image as either a real face or a spoof. (Binary Classification)

The chosen image quality measures/features intent to estimate the appearance of image in a objective and reliable way.

The deployed method operates on the complete image without searching for any specific characteristics and hence it doesn't require any pre-processing steps such as eyes detection, face detection, etc. prior to the calculation of proposed features.

The image quality measures that have been considered in the work can be categorized into 3 disjoint groups based on the image information measured.

1. Pixel Difference Measures: These measures captures the distortion between the two images based on their pixel-wise differences.
2. Correlation Based Measures: These measures captures the similarity between the two images using different variants of the correlation function.
3. Edge Based Measures: These measures are used to capture the 2 most important visual features of the image namely corners and edges that play a key role for identification & characterization in human visual system.

The full-reference image quality measures used are:

1. Mean Squared Error: It is the average of the squared difference between the pixel intensities of the given image and the Gaussian filtered image.

$$\boxed{MSE(\mathbf{I},\hat{\mathbf{I}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j})^2}$$

2. Peak Signal to Noise Ratio: It is the ratio between the maximum possible power of signal to the power of corrupting noise that affects the exactness/correctness with which value of signal is represented.

$$\boxed{PSNR(\mathbf{I},\hat{\mathbf{I}}) = 10 \log\left(\frac{\max(\mathbf{I}^2)}{MSE(\mathbf{I},\hat{\mathbf{I}})} \right)}$$

3. Signal to Noise Ratio: It the ratio of signal power to the noise power.

$$\boxed{SNR(\mathbf{I},\hat{\mathbf{I}}) = 10 \log\left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j})^2}{N \cdot M \cdot MSE(\mathbf{I},\hat{\mathbf{I}})} \right)}$$

4. Structural Content: It is a perceptual metric/score. It quantifies image quality degradation caused by processing image. It is a reference based feature that requires two images - the original image and the processed image.

$$\boxed{SC(\mathbf{I},\hat{\mathbf{I}}) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j})^2}{\sum_{i=1}^{N} \sum_{j=1}^{M} (\hat{\mathbf{I}}_{i,j})^2}}$$

5. Maximum Difference: It is the maximum of absolute difference between the corresponding pixel intensity of the original image and the Gaussian filtered image.

$$\boxed{MD(\mathbf{I},\hat{\mathbf{I}}) = \max |\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j}|}$$

6. Average Difference: It is the average of difference between the corresponding pixel intensity of the original image and the Gaussian filtered image.

$$\boxed{AD(\mathbf{I},\hat{\mathbf{I}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j})}$$

7. Normalized Absolute Error: It is the sum of ratio of absolute difference between the corresponding pixel intensity of the original image and the Gaussian filtered image normalized(divided) by the pixel intensity of the original image.

$$\boxed{NAE(\mathbf{I}, \hat{\mathbf{I}}) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} |\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j}|}{\sum_{i=1}^{N} \sum_{j=1}^{M} |\mathbf{I}_{i,j}|}}$$

8. R-Averaged Maximum Difference: It is the average of R maximum absolute differences between the pixel intensity of the original image and the Gaussian filtered image.

$$\boxed{RAMD(\mathbf{I}, \hat{\mathbf{I}}, R) = \frac{1}{R} \sum_{r=1}^{R} \max_r |\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j}|}$$

9. Laplacian Mean Squared Error: It is based on importance of edge measurement. Large value of LMSE indicate that image is of poor quality.

$$\boxed{LMSE(\mathbf{I}, \hat{\mathbf{I}}) = \frac{\sum_{i=1}^{N-1} \sum_{j=2}^{M-1} (h(\mathbf{I}_{i,j}) - h(\hat{\mathbf{I}}_{i,j}))^2}{\sum_{i=1}^{N-1} \sum_{j=2}^{M-1} h(\mathbf{I}_{i,j})^2}}$$

10. Normalized Cross-Correlation: It is the accumulation of ratio of product of pixel intensity of the original image with the Gaussian filtered image normalized with the pixel intensity of the original image.

$$\boxed{NXC(\mathbf{I}, \hat{\mathbf{I}}) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j} \cdot \hat{\mathbf{I}}_{i,j})}{\sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j})^2}}$$

11. Mean Angle Similarity: It is an angle-based similarity measure which helps in achieving a sense of similarity in real and the modified image.

$$\boxed{MAS(\mathbf{I}, \hat{\mathbf{I}}) = 1 - \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (\frac{2}{\pi} \cos^{-1} \frac{\langle \mathbf{I}_{i,j}, \hat{\mathbf{I}}_{i,j} \rangle}{||\mathbf{I}_{i,j}|| \, ||\hat{\mathbf{I}}_{i,j}||})}$$

12. Mutual Information:It is the measure of the amount of the information contained by modified image about the real input image.

$$\boxed{I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\, p(y)} \right)}$$

13. Total Edge Difference: It is the total number of locations in both the images where either the original image or the Gaussian filtered image has a pixel that is part of edge in the corresponding image normalized by the total number of pixels in the image.

$$\boxed{TED(\mathbf{I}_{\mathbf{E}}, \hat{\mathbf{I}}_{\mathbf{E}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |\mathbf{I}_{\mathbf{E}i,j} - \hat{\mathbf{I}}_{\mathbf{E}i,j}|}$$

14. Total Corner Difference: It is the absolute difference between the total number of corners between the original image and the Gaussian filtered image normalized by the maximum of number of corners in the original image and the Gaussian filtered image.

$$\boxed{TCD(N_{cr}, \hat{N}_{cr}) = \frac{|N_{cr} - \hat{N}_{cr}|}{\max(N_{cr}, \hat{N}_{cr})}}$$

Methodology - 2(Involving Linear Binary Patterns)

Visual inspection of image of a real user and spoofed image of the user look very similar but if we translate the given image to proper feature space then some disparities many become evident. We have tried to capture the texture properties of the image with features based on Local Binary Patterns(LBP).The simplest Local Binary Pattern for a particular pixel is usually denoted as LBP^{3X3}and is formed by comparing the intensity values of that pixel with the intensity values of the pixels in its 3x3 neighbourhood. In this way, each pixel is assigned a label with value from 0 to 2^8-1. In the case of uniform Local Binary Pattern(LBP^{u2}), only the labels which contain at most two 0-1 or 1-0 transitions are considered. The feature vector of an image or a region/section of the image is formed by calculating a histogram of the pixel labels. One other strong motivation for using LBP is it’s illumination invariant property i.e. robustness of LBP to monotonic grey-scale changes.

There are two ways in which we can capture the texture feature for an image. These are as follows:

1. First option would be to compute the LBP feature for all the pixels in the image and put them in one histogram where bins in histograms would be equal value ranges between 0 to 2^8-1. The votes for different ranges in the histogram formed will be denoting our final feature vector.
2. Second option would be to divide the image into K*K blocks and computing the LBP feature for each of the blocks separately. Finally, concatenating the per block computed features to obtain our final feature vector.

In this work, we have considered the first method to generate the LBP histogram for the entire input images. In addition to LBP, we have also adopted another modification to LBP i.e. transition LBP(t-LBP), which acts as a complementary feature to the computed LBP feature.

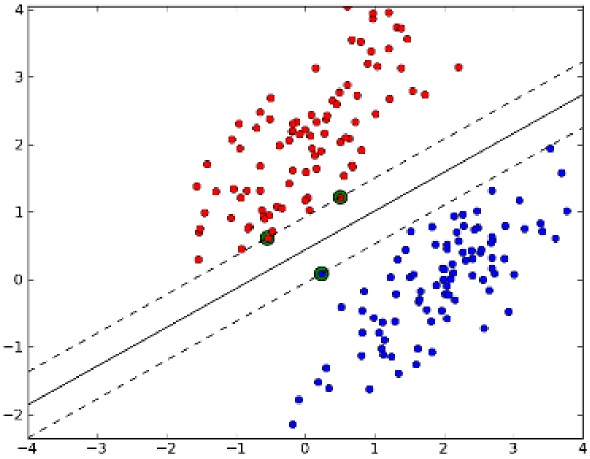
The formulation of t-LBP is as follows:

$$\boxed{tLBP_{P,R} = s(g_0 - g_{P-1}) + \sum_{p=1}^{P-1} s(g_p - g_{p-1})2^p.}$$

It can be viewed as andata about partial ordering of border pixels. Additionally, t-LBP too enjoys the benefit of being illumination invariant.

Support Vector Machine

Support Vector Machine(SVM) is a discriminative classifier. The hyperplane is drawn such that the distance between the support vectors and the hyperplane is as maximum as possible. This means the drawn hyperplane best splits the data. It draws a optimal hyperplane i.e. a hyperplane with maximum margin. This built hyperplane is used to classify new points that are queried against the learned SVM model.



Points marked from red ink belong to class A and the points marked with blue ink belong to class B. The green points shown in the figure are the support vectors for the drawn optimal hyperplane separating class A from class B. Perpendicular distance of the dotted line from the bold black line i.e. the hyperplane is called margin.

SVM is constrained optimization problem which in one way can be solved by Lagrange Multipliers technique.

minimize:

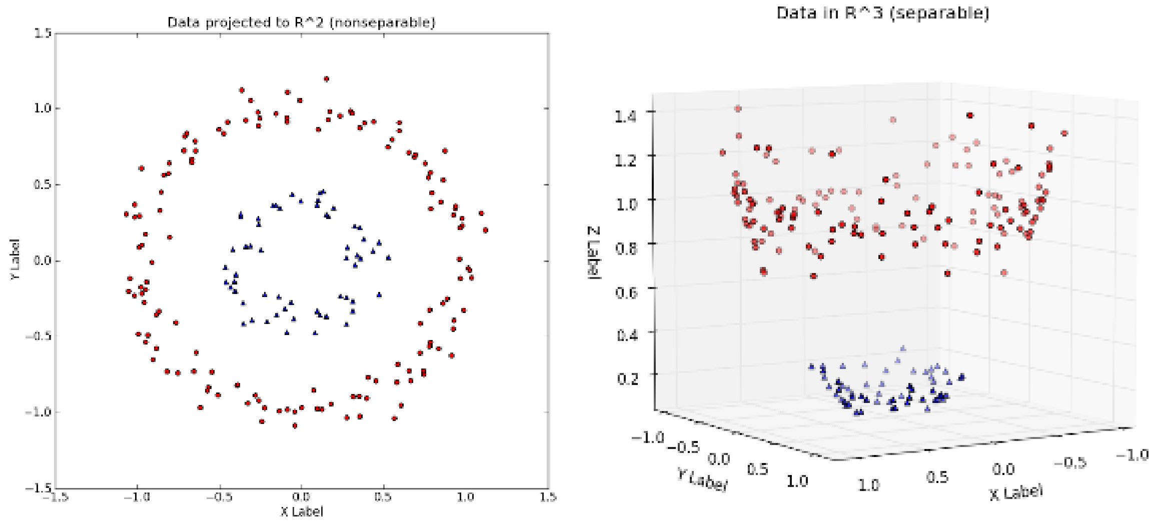
$$W(\alpha) = -\sum_{i=1}^{\ell} \alpha_i + \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \mathbf{x}_j$$

subject to:

$$\sum_{\substack{i=1 \\ 0 \leq \alpha_i \leq C}}^{\ell} y_i \alpha_i = 0 \qquad (4)$$

SVM can easily be used for 2 class classification. Also, it can be extended for multi-class classification by using one vs rest classification scheme.

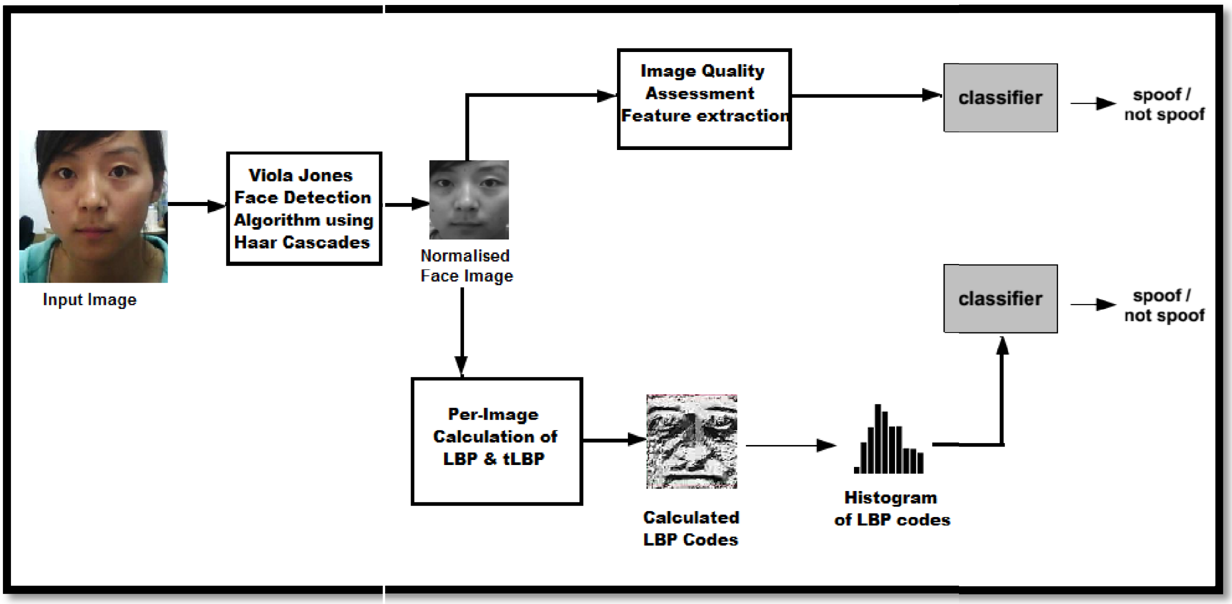
If we can transform our given set of data points to higher dimension the there is a chance that we can separate different classes in our data in higher dimension which in lower dimension was non-separable. For SVM to be trained for that higher dimension, we don’t need the exact transformation of our data but we just need the inner product of our data in that higher dimensional space. It is quite tough to get the exact transformation of our data as compared to just the inner product of our data in higher dimension. This process of using the inner products of our data points and surpassing the process of exact transformation of data in higher dimension is termed as Kernel trick.



Various types of kernels are available for SVM. For example:

1. Polynomial Kernel
2. Gaussian Kernel

Experimental Set-up


Input Image
Viola Jones Face Detection Algorithm using Haar Cascades
Normalised Face Image
Image Quality Assessment Feature extraction
classifier
spoof / not spoof
Per-Image Calculation of LBP & tLBP
Calculated LBP Codes
Histogram of LBP codes
classifier
spoof / not spoof

In this work, we have used CASIA face anti-spoofing dataset as base database. The dataset is divided into 80:20 train-test split. For the training data, statistical features are calculated and concatenated to form a 14-dimensional feature vector which is inputted to train the Support Vector Machine (SVM). Subsequent to the offline training, same feature vectors are created for the test images and is tested using the trained SVM. For the second module, similar approach is adopted. We compute a 16-dimensional feature for each training image. We compute LBP histogram (8-bins) and t-LBP (8-bins), further normalized and concatenates them to the form the feature vector. Offline training was done, post which testing was done using test dataset.

Results

1. Image Quality Assessment Module

- Train dataset: Around 10k images

- Test dataset: Around 2.5k images, having 1k real face images and 1.5k spoofed images.

Confusion Matrix		
Classified as:	**Real Image**	**Spoof Image**
Real Images	964	52
Spoof Images	47	1460

Classification Report				
	Precision	**Recall**	**F1-Score**	**Support**
Real Images	0.95	0.95	0.95	1016
Spoof Images	0.97	0.97	0.97	1507
Avg/Total	0.96	0.96	0.96	2523

On inspection of SVM classifier summary, the three features which turned out to be most important are as follows:

- Structural Content
- Normalised Cross Correlation
- Mutual Information

2. Textural feature based classification

- Train dataset: Around 10k images

- Test dataset: Around 2.5k images, having 1k real face images and 1.5k spoofed images.

Confusion Matrix		
Classified as:	**Real Image**	**Spoof Image**
Real Images	847	175
Spoof Images	92	1409

Classification Report				
	Precision	**Recall**	**F1-Score**	**Support**
Real Images	0.90	0.83	0.86	1022
Spoof Images	0.89	0.94	0.91	1501
Avg/Total	0.89	0.89	0.89	2523

Future Works

In the work presented, we have mainly focussed on sharpness, locality and textural based features. We have planned to work on image distortion analysis which consists of designing specularity, color and contrast based features for the purpose of classification of real and spoofed faces.

Our work till now focuses on classifying any given input image as real or spoofed face image, hence a non-referential model. But as stated in the problem statement, our final goal is to build an referential model for an organization. We have planned to achieve this through building feature set for each individual of the organization, and do a cluster analysis to validate the person as an employee of the organization(which he/she has claimed to be).

References

[1] J. Galbally and S. Marcel, "Face Anti-spoofing Based on General Image Quality Assessment," *2014 22nd International Conference on Pattern Recognition*, Stockholm, 2014, pp. 1173-1178.

[2] Trefný, Jiří&Matas, Jiri. (2010). Extended Set of Local Binary Patterns for Rapid Object Detection. 15th Computer Vision Winter Workshop, Volume 2010.