**Project Report**

**Biometric Template Protection**

**Based on Binarized Statistical Features and Bloom Filters**

**Abstract:**

The use of Biometric Systems in smartphones is spreading widely. This raises the need to protect biometric templates in such devices to preserve the privacy of biometrics. We propose a method to secure biometric templates on smartphones using bloom filters and binary statistical image features. This method is evaluated on a dataset of 25 subjects. The dataset contains images of celebrities. Biometrics used for the evaluation are the face and the two eye periocular regions. The result obtained is that this system has an 85% accuracy in authenticating a given image. It proves that we can protect and preserve privacy without compromising accuracy. With further modifications, we can increase the accuracy even more. This method is very reliable in protecting the biometric templates and authenticating them at the same time on a smartphone.

**Introduction:**

Biometric systems are methods of verifying or recognizing the identity of a living person based on physiological characteristics, such as a fingerprint or face pattern, or behavioural characteristics, such as handwriting or keystroke patterns. Biometrics is becoming more common in our everyday lives. Biometric-based systems have proven to be trustworthy and precise. The accuracy of biometric systems is improving all the time, but it remains a challenge. More than accuracy, however, the privacy concerns associated with storing and protecting biometric templates pose a greater challenge. Because biometric systems are vulnerable to hacking, there is social apprehension about using biometrics in everyday life.

Biometric template protection is a widely studied area in biometrics. It's required to keep the templates safe from being hacked or attacked. In this study, we use a most promising biometric template protection approach using Bloom filters, and we use the face and periocular regions as biometrics. Studies have shown that it is possible to protect and preserve privacy while not affecting verification accuracy. We aim to maintain the balance between accuracy and privacy.

The proposed approach involves extracting statistical image features from captured face and periocular images, binarization of the extracted feature vectors, and Bloom filter transformation to protected templates. We chose bloom filters because they provide non-degraded biometric performance while also securing the biometric template.

**Proposed Template Protection Approach**

**Face and Periocular Segmentation:**

As mentioned above our Region of Interest in this study is the Face and Periocular regions. So, to identify whether a person is the person he claims, we need to capture his face and process the biometric system and identify if it is him or not.

To segment a face from a captured image, we use an old classifier called Haar Cascade classifier for the face. It segments out the face from a given image with great accuracy. After segmenting the face, we use the eye Haar Cascade classifier on the segmented face to get good results. After getting the eyes, we extract the periocular region by segmenting the region surrounding the eyes up to a limit. We can get the periocular region by segmenting a part of the face around the eyes. The amount of part we need to extract can be calculated by the size of the eye image we get.

The segmented face images are resized to 64 × 80 pixels and the periocular image to a size of 120 × 88 pixels that will be used for feature extraction.

a.) Captured Image b.) Face Segmentation

c.) Right Periocular d.) Left Periocular

# Feature Extraction and Binarization

Using a good feature extraction technique is important. The Bloom filter, like many other template protection approaches, requires binary feature vectors as input. In this study, we used the binary statistical image features (BSIF) that can provide binary features by performing convolution with several filters that are learned on natural images. We can create special domain-based BSIF filters. But the filters learned on natural images work fine on the face, iris, and periocular regions. We use a feature extraction scheme based on eight different filter sizes such as 3 × 3, 5 × 5, 7 × 7, 9 × 9, 11 × 11, 13 × 13, 15×15, and 17×17 and each filter is 8 bit in length.

After segmenting out the face and periocular regions, we divide the images into 32 blocks. Each block is of size 8 x 20 pixels in the case of the face and 15 x 22 pixels in the case of Periocular regions. We extract the BSIF features by doing convolution operations of these filters onto these blocks. We get a response image for that filter on that block. We encode that response to a histogram of dimension 1 x 256. Here each point on the histogram represents the no of pixels presents having that grey level value in that response image. We have eight filters, so we get a feature size of 8 x 256 for each block. We repeat this process for each block and concatenate the result to a final BSIF feature vector set of dimensions, 65536.

The extracted histogram features h is binarized as follows:

**h(i , j) = 1 if h(i , j) > 0 and 0 if h(i , j) = 0.**

We finally reorganize the binarized features to have a matrix of dimension 4×8 rows and 8×256 columns that can be used to generate a protected template using Bloom filter.



a.) Original Images b.) 3x3 Filter Response c.) 15x15 Filter response

We can see the difference between 3x3 filter and 15x15 filter responses. As the filter size increases the detail in the image decreases, but it only shows the important features present.

Like this, we get responses of the images, and then we encode them into 1x256 histogram and concatenate. In this way, we can get binary feature vectors that are useful for bloom filters.

# Biometric Template Protection: Bloom Filters

We use the Bloom filter because of its irreversible property in protecting biometric templates. We generate 16bit length bloom filters.

We now have 32 blocks of 8 x 256 binary matrices for an image. We further divide these 8 x 256 blocks into 32 sub-blocks of size 4 x 16. Each column in this sub-block is used as an index in the bloom filter. Initially, we take a 16bit length vector which contains full of zeros. Then we take a sub-block, then each column of the sub-block is encoded to a decimal number which is then used as an index in the bloom filter. The value of the bit indicated by that index is then set to 1. If it is already set to 1, we leave it to be 1.

In this way, we create a bloom filter of 16bit size. We get 1024 bloom filters of size 16bit for an image. This makes the size of a template to be 2kB.

We compare any two templates using a dissimilarity score using the formula:

**DS(bi ,bj) = HD(bi ,bj )/ |bi| + |bj|**

Here |b| is the no of bits set to 1, HD is the no of disagreeing pairs of bits between those two templates.

We store the scores obtained for the three biometric regions separately. Then we use them to get the final decision. We make the final decision by taking the majority decision obtained by these three scores for a particular image.

**Problems Identified and their Solutions:**

The problems identified are as follows:

1. It is difficult to know what is called a periocular region. Whether we should include or exclude the eyebrows. How to segment it out from the face. It is also difficult to segment out the face region. The face may be tilted in the image or, if the person rotated his head slightly we get some unwanted portion in our segmented face.

This is an important part of the model pipeline. Without getting the face and periocular regions properly we can’t move further ahead, because the results we get will not be correct and we may get fewer accuracies.

1. The time complexity for the whole model was high. It took almost 1.1s - 1.3s for the model to run and create a biometric template for a given image. Due to this, it was difficult to run the model on bigger datasets. Using big datasets may give even better results as we use different sets of faces and we know more about different features.

1. There was no proper dataset. The dataset I have used was of the celebrities. The images have different lighting on them, and they have different fashion senses in different images. As we know, if the pixel greyscale value changes the template may change so much and it could give us wrong results.

**Solutions and their outcomes**

**The first problem** of the periocular region was solved by making some changes in the outcome haar cascade eye classifiers. I have used two different classifiers for each eye instead of a single classifier to get better results. The outputs of each classifier are then extended to some more regions depending on the size of the face image extracted. I have calculated the ratio of how much we should extend the region depending on the size of the face image to get a good periocular region. I have got good results for the periocular region segmentation.

The face tilting problem is also solved by tilting the image such that the line connecting the centers of the eye regions is made horizontal. But this change is not improving accuracy very much and it increases the time complexity slightly. So this change is not implemented in this model.

The removal of the unwanted portion of the image is very difficult. I was not able to remove the extra background coming in the face image. The classifier used is also not able to remove it.

 The part that is on the right side of the red line is the

unwanted portion. It may not be required to implement this because we only use correct frontal face images, but if we make it happen, it will increase the accuracy by a small amount.

**The Second problem**, i.e the time complexity problem is solved. I have managed to reduce the time complexity by a small factor. In the segmentation part, I have used the LBP cascade classifier for the face instead of the haar cascade classifier. It reduced the time taken for the model to run on an image to 0.75s - 0.8s. This may not look like a great reduction, but when we run on larger datasets, we find it is a significant change. I couldn’t change the time taken by the feature extractor. It takes most of the time to run. The for loops and all other things involved are required for the feature extraction part, so I couldn’t remove any component from that part.

**The Third problem** is there is no proper dataset. I have tried to find the dataset they have used in the paper, but I couldn’t find that anywhere. I have tried to gather real images dataset i.e the images taken by the front camera of the phone for evaluation. But I couldn’t gather many. So, I have not changed the dataset used. We may get different results or greater accuracies if we use real images because the lighting and the makeup on the face may not vary much.

**Results:**

**Dataset:** The dataset contains 200 images i.e 8 images from 25 subjects. These images are gathered from an online dataset of images called CASIA which contains images of celebrities.

**Scores:** As mentioned above we have used a dissimilarity score to compare two templates. These scores are calculated individually for each biometric and stored for finding the decision threshold. The decision threshold is the threshold to decide if the two images are a match or not depending on the score between them. As we have used a dissimilarity score, if the score is less than the threshold it is a match, if it is greater, then it is not a match.

If we find a match we check whether it is truly a match or not by comparing the names of both images. Matched images should have the same person number in their name.

**Plotting the ROC curve:**

A ROC curve is obtained by plotting **False Rejection Rate (FRR)** vs **False Acceptance Rate (FAR)**, by varying the decision threshold.

FAR refers to the likelihood of the biometric system incorrectly accept an unauthorized user as an authorized one i.e the rate of false acceptance over the number of imposter attempts. It is defined as:

*FAR = Number of incorrect matches recognized*

*Total number of matches*

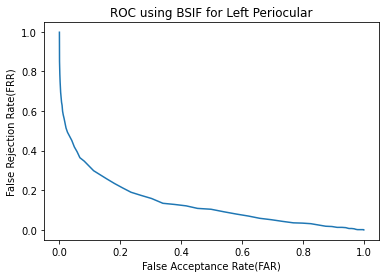
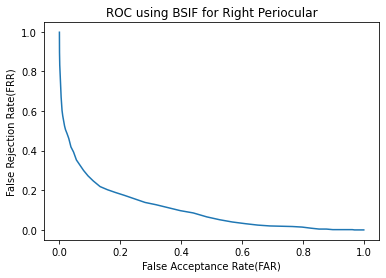
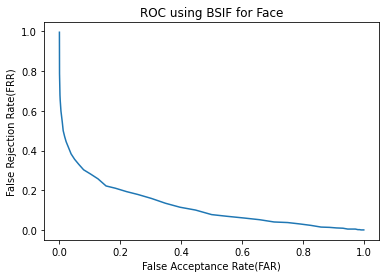
FRR on the other hand is the likelihood of the biometric system incorrectly reject an authorized user by considering him to be an unauthorized user, i.e the rate of false rejection over the number of genuine attempts by the user. It is defined as:

*FRR = Number of correct matches not recognized*

*Total number of matches*

So, we calculate the FRR and FAR for different threshold values between 0.25 and 0.55 and store both of them in two separate lists.

Then we plot the graph between these lists and we get the ROC curve. The three ROC curves for three biometrics on this data set are shown below.



If the decision threshold is increased then FAR decreases and FRR increases, if the decision threshold decreases FAR increases, and FRR decreases. To get an equilibrium point between these two values we take the point where both values get equal. So EER (Equal Error Rate) is the point at which FAR and FRR are equal. It gives a threshold to evaluate the recognition performance of a system.

|  |  |  |
| --- | --- | --- |
| **Characteristic** | **Accuracy at EER point** | **EER Threshold** |
| Face | 80.307% | 0.337 |
| Left Periocular | 79.017% | 0.363 |
| Right Periocular | 81.347% | 0.363 |
| Fused | 85.778% | ------------------ |

So, we get a final accuracy of 85.778% if we use this model. This accuracy is good and this model also protects the biometric template.

This model has better accuracies for both periocular regions than what is given in the paper. But the face biometric accuracy is less compared to that of the value present in the paper.

The reason behind this may be because in the paper they have used real face images and perfectly frontal face images. Whereas in our model, the face images are not real-time face images and they contain some irregularities between images of the same person. And they are not all perfectly frontal images, so there may be some unwanted background present that we cannot ignore.

**Future work**

The future work to do is we need to solve the problems of face tilting and removal of unwanted face background. We also need a good dataset for better results. We need to reduce the time complexity also if possible, to run the model on larger datasets.

We need to find another binarized feature extractor that gives binary features for images because if we are going to use bloom filters, we need binary features. If we can get another binary feature extractor better than BSIF we can get even better results and we need not have to change the bloom filter part.

We also need to find other ways to protect the template. If we want to use other feature extractors we need something better than bloom filters.

Working on these things may lead to a better model than this that can simultaneously verify and protect the biometric template.

**Conclusion**

We can change the password if we want but we cannot change the biometric characteristics of a person. If this biometric data is hacked or has been compromised there is nothing we can do.

Usage of these biometric systems is increasing especially in the case of smartphones. This further justifies the need for protecting the templates. In this study, we have followed a simple method that uses Bloom filters and BSIF on the face and periocular regions. The accuracy of 85.778% shows us that it is possible to verify a person with great accuracy while still protecting the biometric template. This shows that this model can be used in real life on a smartphone without any problem.

**References**

1. Kannala, J., Rahtu, E.: BSIF: Binarized statistical image features. In: 21st

International Conference on Pattern Recognition (ICPR), pp. 1363–1366

(2012)

1. Raja, K.B., Raghavendra, R., Stokkenes, M., Busch, C.: Smartphone authentication system using periocular biometrics. In: 2014 International

Conference of the Biometrics Special Interest Group (BIOSIG), pp. 1–8,

September 2014

1. Raja, K.B., Raghavendra, R., Stokkenes, M., Busch, C.: Multi-modal authentication system for smartphones using face, iris and periocular. In: Proceedings of 2015 International Conference on Biometrics, ICB 2015, pp. 143–150 (2015)
2. Viola, P., Jones, M.: Robust real-time face detection. Int. J. Comput. Vis. 57(2), 137–154 (2004)