Open Access Library Journal
2025, Volume 12, e12807
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Texture Analysis for Makeup-Free Biometrics: A Solution for Imposture Mitigation

R. Logeswari Saranya*, K. Umamaheswari

Department of Information Technology, PSG College of Technology, Coimbatore, India
Email: *sararajakave@gmail.com

How to cite this paper: Logeswari Saranya, R. and Umamaheswari, K. (2025) Texture Analysis for Makeup-Free Biometrics: A Solution for Imposture Mitigation. *Open Access Library Journal*, **12**: e12807
https://doi.org/10.4236/oalib.1112807

Received: December 12, 2024
Accepted: February 25, 2025
Published: February 28, 2025

Abstract

Face recognition is rapidly becoming one of the most popular biometric authentication methods. Most face recognition systems are focused on extracting features and enhancing their verification and identification capabilities. The detection of security vulnerabilities of different types of attacks has been given attention only in recent years. These attacks can include, but are not limited to: Obfuscation Spoofing and morphing; for example, a hacker can masquerade as a target to gain access to the biometric system. The application of cosmetics can alter the appearance of a face, leading to a decreased characteristic distinctiveness. Facial makeup includes variations in skin tones, the position of eyebrows and skin complexion. The cosmetic effect on an individual causes the face recognition system to falsely identify the person affecting the security of the biometric system. Adding a presentation attack detection module to the existing biometric system can be the solution to this problem. In this work, a CNN-based machine learning approach is adapted to classify the presentation attack using texture analysis. The proposed method is to extract the original face by removing makeup so that the FR system recognizes the person's real identity, resulting in decreased vulnerability. The false accept rate (FAR) is a measure of a biometric system's resistance to zero-effort attacks and is generally considered as the system's performance.

Subject Areas

Artificial Intelligence, Computer Vision

Keywords

Makeup Classification, Makeup Elimination, Convolutional Neural Network, GAN Models, Haar-Cascade Algorithm

1. Introduction

The biometric system is the security system that recognizes and identifies people based on their biological and behavioral characteristics. This biometric authentication system is used for various applications like the security of computers and mobile phones, airports, banks, military bases, biometric attendance, and tracking systems. Though biometric systems improve security, like any other system, they are vulnerable and prone to threats.

Face recognition is one of the popular biometric authentication methods. The system is said to be vulnerable when it falsely identifies an individual and gives them access rights which can lead to the exploitation of information. This vulnerability is due to a variety of attacks like spoofing, obfuscation, morphing, and makeup. The solution to the spoofing attacks has already been studied in lots of papers [1].

Presentation attack detection (PAD), also referred to as "anti-spoofing" or "Liveness detection," may be a critical capability to think about when deploying face recognition in automated authentication and identification scenarios. Whereas face recognition determines if a presented face matches a registered record, PAD determines whether the face itself may be authentic or is a copy of the face, from a photograph to a video sample on an LCD to a high-resolution 3D mask [1] [2].

Despite a good deal of progress in face recognition systems, vulnerabilities to face spoof attacks are mainly overlooked. The facial spoof attack may be a process during which a fraudulent user can subvert or attack a face recognition system by masquerading as a registered user and thereby gaining illegitimate access and advantages. Face spoofing attacks may be a major issue for companies selling face biometric-based identity management solutions. It is thus essential to develop robust, efficient, and compact face anti-spoofing (or liveness detection) methods, which are capable of generalizing well to discriminative, class-specific information and imaging conditions.

To achieve this goal, the main focus is on the attack caused by makeup, which is the makeup attack. The application of cosmetics on the face to look like other people like celebrities can sometimes confuse the biometric system. Makeup can cause variation in skin tone, skin complexion, lip color, eye shadows, the position of eyebrows, and the overall appearance of the person. This vulnerability due to the makeup attack must be addressed to improve the biometric system security. The overall flow of paper goes as first have gone through all the survey papers and got the limitations on the existing system. Second proposed structure is defined. Third, experimental results and evaluation are done. Finally, a comparison is done between models [3] [4].

The biggest limitations of such a method when applied to real-time biometric systems include data breaches, which are critical to ensure interception-free and theft-free biometric data. Light & Weather: Facial recognition might not perform very efficiently in low-light conditions as well as outdoors. Aging Biometric: Such

biometrics, like a photo of the face or fingerprint from a person, may change over time, thus affecting recognition accuracy [5] [6].

2. Literature Survey

Several papers and applications on makeup detection and facial makeup removal have been studied as a reference for this work. The details that have been inferred from those applications and papers are discussed below.

2.1. Classification of Makeup

The proposed method uses a feed-forward back-propagation neural network-based classifier for classification. The classification makes use of features extracted using a discrete wavelet transform approach from face samples [7] [8]. The most commonly used neural network architecture with the back propagation algorithm is the multilayer feed-forward network. This technique is not only computationally less extensive than other techniques but also provides the best results on various images. The robustness for image variation in rotations, illuminations, etc., must be improved. The evaluation of the robustness of the largest data sets is necessary for practical use.

The vulnerability of a widely used open-source face recognition system (*i.e.*) Arc face, to makeup presentation attacks using makeup-induced face spoofing datasets like MIFS and FRGCv2.The success rate of makeup attacks in the MIFS datasets has an impact on the security of the face recognition system. The warping technique is used to simulate improved makeup presentation attacks, which result in a higher success rate [9] [10]. The m-PAD technique is used to compare and classify the bonafide and makeup images. Provides better performance in all the datasets.

A convolutional neural network is used to distinguish between presentation with age-induced facial makeup and without makeup. Proposed presentation attack detection provides a 6.6% Average classification error rate in the AIM (Age-induced makeup) dataset and 4% in all the datasets. AIM dataset contains 200 + video presentations of old-age makeup and original faces each [1]. AIM dataset results in a 14% decrease in the median matching scores of recent CNN-based FR systems. Overall accuracy is 93% using the AIM dataset [11] [12].

2.2. Makeup Exclusion

The WGAN-GP approach is used to remove the makeup. The dataset collected consists of five separate datasets (MIFS, FAM, YMU, VMU, MIW) of a total of 2600 images of 1300 different people. Each person has two images, one with makeup and the other without makeup. CNN model is developed to classify whether the person is with makeup or without makeup, and then a generative adversarial network (GAN) model is built to remove the makeup from the image [13] [14]. The best accuracy obtained for this model was 80% on training and 79% on testing.

The aim is to promote the existing verification system to accept or reject the

claimed identity of a person with makeup in an image. A makeup robust face verification framework is proposed based upon a generative adversarial network. The proposal synthesizes non-makeup face images from makeup images. Specifically, a patch wise contrastive loss is introduced in the generative model to constrict the distance between makeup and non-makeup images [10] [15].

A bidirectional tunable de-makeup network (BTD-Net) is proposed to remove makeup effects. For tractable learning of the makeup process, which is one-to-many mapping determined by the cosmetics that are applied, a latent variable is used which reflects the style of the makeup [16] [17]. This latent variable is extracted from the de-makeup process and used as a condition of the makeup process to constrain the one-to-many mapping to a specific result. The proposed BTD-Net surpassed the state-of-the-art techniques in estimating realistic no-makeup faces that correspond to the input makeup images.

3. Proposed Method for Imposture Identification

The proposed system has two major levels of processing: first classification and second removal. In the classification phase, the first image taken as an input may consist of noise which should be removed. A Gabor filter is used to remove noise. Second, pre-processing is done to extract the features of the image. Finally, based on feature, the different convolution neural network hidden layers are used to classify the categories of makeup and no makeup images. In the removal phase generative adversarial network is used to generate real images from fake images in a cyclic fashion.

3.1. Process Work Flow of Classification of Makeup

3.1.1. Data Preparation and Pre-Processing

Data Pre-processing is one of the important steps in machine learning. Each image in the dataset (both makeup and no makeup images) is processed before being given as input to the model. The face from the image is detected using the Haar-cascade algorithm. The detected face is then cropped and saved as an image. By doing this, the images that contain no face are eliminated and only the face is extracted leaving the unwanted information as shown in **Figure 7**. At first, the input images are resized into 256 × 256 and the bilateral filter is applied to remove any available noise. The Gabor filter is applied to the image to extract the features. This process is shown in **Figure 8**. The image is then divided into 9 × 9 = 81 blocks. Uniform LBP is calculated for each of these blocks. A histogram is computed for each block and is then concatenated to form a feature vector, as shown in **Figure 1**.

Select a filtering technique that is dynamically according to noise characteristics. For instance, bilateral filtering can be used for moderate noise, and more sophisticated methods such as deep learning-based de-noising, can be applied for high noise levels. While the bilateral filter works well under moderate noise circumstances coupled with edge-preserving noise filtering, it may not work in all

situations, especially in the case of high density noise. Combining it with adaptive-nose-specific techniques and training robust models significantly benefits the system's overall performance.

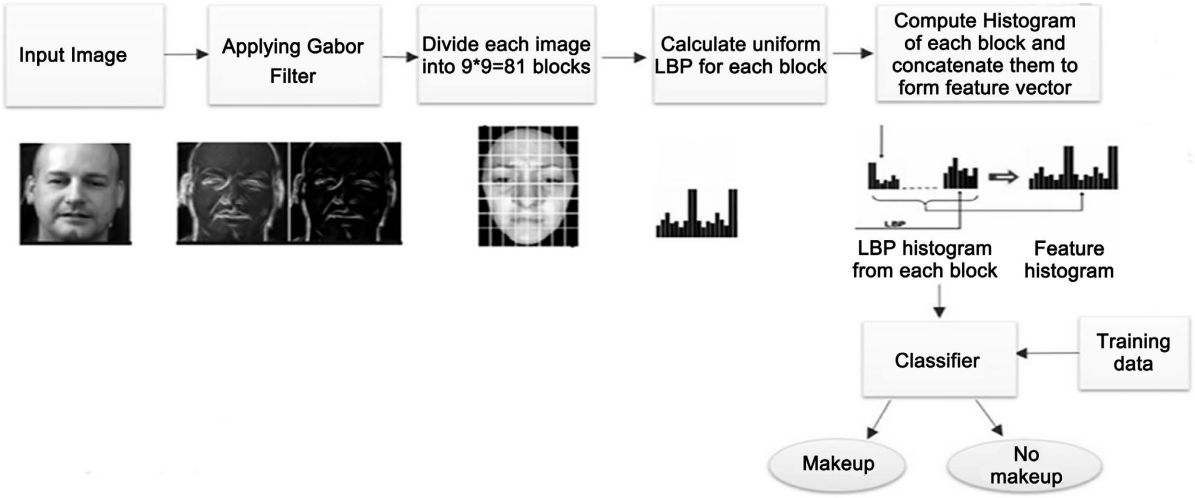

Figure 1. Workflow of classification model.

3.1.2. Step by Step Procedure of the Proposed VGG Architecture

The proposed deep convolution neural network VGG consists of several layers to classify the makeup and on makeup image.

Algorithm 1: Makeup Classification

```
Step 1: Initialize the network
        Initialize network with parameters:
            Input image size: 48x48 (grayscale)
            Kernel size: 3x3
            Number of convolutional layers: 5
            Max-pooling window size: 2x2
Step 2: Input the grayscale image
        Input_image = 48x48 grayscale image
Step 3: Define the convolutional layers with ReLU activation
        For layer in range (1, 6):  // Stack of 5 convolutional layers
            If layer == 1:
                Input = Input_image
            Else:
                Input = Output from the previous layer
//Perform 3x3 convolution
        Convolved_output = Convolution_2D (Input, kernel_size=3x3)
//Apply ReLU activation
        Activated_output = ReLU (Convolved_output)
//Down-sample using max-pooling (2x2)
        Pooled_output = MaxPooling_2D (Activated_output, window_size=2x2)
// Update output for the next layer
        Output = Pooled_output
Step 4: Flatten the final output
        Flattened_vector = Flatten (Output)
Step 5: Connect to the classification layer
        Classification_output = FullyConnectedLayer (Flattened_vector)
//Return the final classification
        Return Classification_output
```

The extracted feature is fed to the classifier model, which classifies the given input image as makeup or as no makeup. The classifier is built using the VGG architecture mentioned in Figure 2. This classifier returns a value that ranges from 0 to 1. Since the sigmoid activation layer is used, the value below 0.75 indicates that the image belongs to class 0 which is no makeup and the value above 0.75 indicates that the image belongs to class 1 which is makeup.

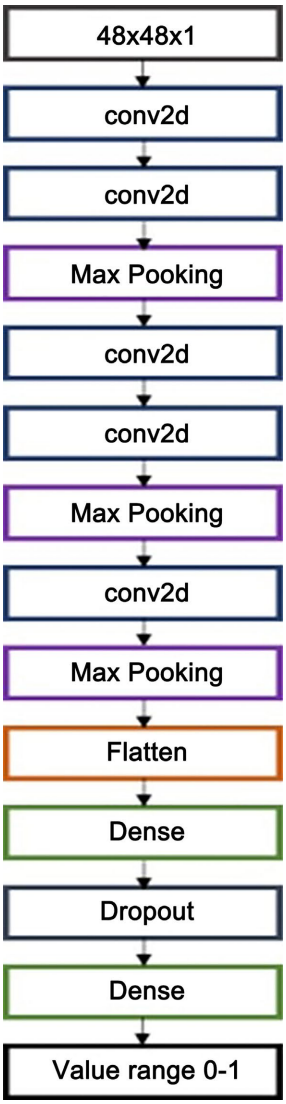

Figure 2. Proposed VGG architecture.

The cutoff of 0.75 for makeup classification is most probably the outcome of very thorough studies to reach an optimum between precision and recall, not static but to be taken again whenever applying to a dataset of different characteristics or when the application of the system changes. True Positive Rate (TPR) against False Positive Rate (FPR) for different thresholds can be drawn to find the most optimal point. The threshold chosen is high like 0.75 to minimize false positives (non-makeup is considered as makeup).

3.2. Process Work Flow for Makeup Elimination

The proposed makeup removal model uses generative adversarial network which works like a cyclic process. Back tracking can be done easier using the reverse mechanism. The working of generators and discriminators as follows.

- Generator A takes makeup images as input and generates the no-makeup images.
- Generator B takes the no-makeup images from generator A as input and generates makeup images.
- The generated images are then passed to the discriminator models which check the plausibility of the images and update the generator models accordingly as shown in Figure 3.

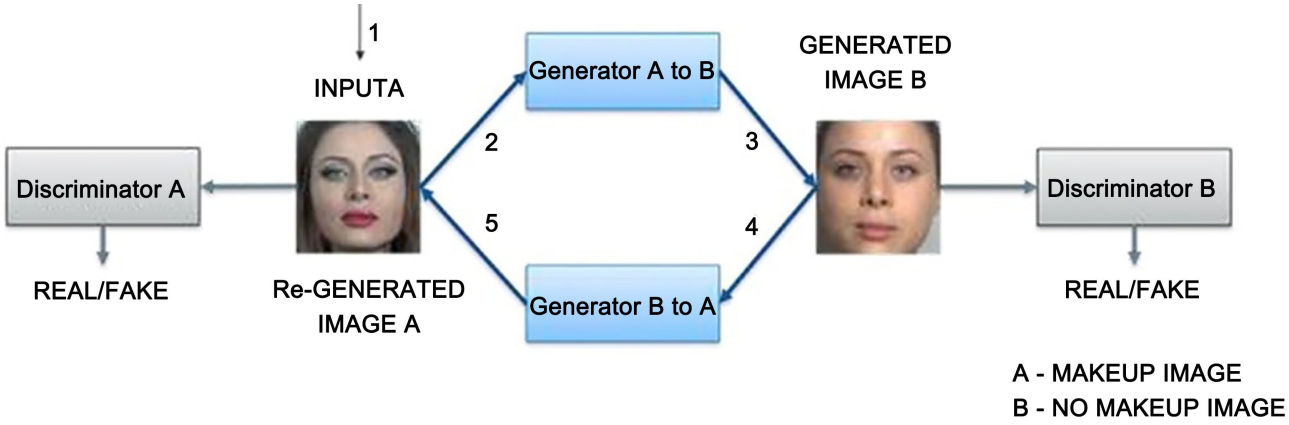

Figure 3. Workflow of makeup removal model.

Here, only the makeup to no makeup translation is required. Hence, we concentrate on generator A. The makeup image is given as an input to this model which generates the corresponding no-makeup image. After detecting whether the subject is wearing makeup or not using the presentation attack classification model, this makeup removal model is used to extract the original face. This can prevent the vulnerability caused by makeup attacks and improve the security of biometric systems.

3.2.1. Step by Step Procedure of the Proposed pix2pix GAN

The proposed Pix2pix GAN is based on the conditional generative adversarial network, where a target image is generated. The condition is placed on the given input image.

The encoder and decoder of the generator are comprised of standardized blocks of convolutional, batch normalization, dropout, and activation layers as shown in Figure 4. This standardization means that we can develop helper functions to create each block of layers and call it repeatedly to build up the encoder and decoder parts of the model. Major limitation of Pix2pix GAN is, it accepts only paired images as input for training the model. This problem is solved using Cycle GAN.

Algorithm 2: Makeup Elimination for Paired Images

```
Step 1: Initialize the Generator (G) and Discriminator (D) models
        Initialize Generator G
        Initialize Discriminator D
Step 2: Define loss functions
        Adversarial_loss = BinaryCrossEntropyLoss () //For discriminator and generator
        L1_loss = MeanAbsoluteError ()          //For pixel-level similarity
        Additional_loss = CustomLoss ()         //For plausible translation
Step 3: Define optimizers
        Optimizer_G = Optimizer (Generator parameters)
        Optimizer_D = Optimizer (Discriminator parameters)
Step 4: Training loop
        For each epoch:
          For each batch of data:
            Step 1: Get source and target images
            Source_image, Target_image = LoadBatch ()
            Step 2: Train the discriminator
//Generate fake target image
            Generated_image = G (Source_image)
//Compute discriminator predictions
            Real_prediction = D (Source_image, Target_image)    //Real pair
            Fake_prediction = D (Source_image, Generated_image)  //Fake pair
//Compute discriminator loss
            D_loss = Adversarial_loss (Real_prediction, 1) + Adversarial_loss(Fake_prediction, 0)
//Update discriminator
            Optimizer_D.zero_grad ()
            Backpropagate (D_loss)
            Optimizer_D.step ()
            Step 3: Train the generator
//Generate fake target image
            Generated_image = G (Source_image)
//Compute adversarial loss for generator
            G_adversarial_loss = Adversarial_loss (D (Source_image, Generated_image), 1)
//Compute L1 loss
            G_L1_loss = L1_loss (Generated_image, Target_image)
//Compute additional loss
            G_additional_loss = Additional_loss (Generated_image, Source_image)
//Total generator loss
            G_loss = G_adversarial_loss + λ1 * G_L1_loss + λ2 * G_additional_loss
//Update generator
            Optimizer_G.zero_grad ()
            Backpropagate (G_loss)
            Optimizer_G.step ()
//End of training
```

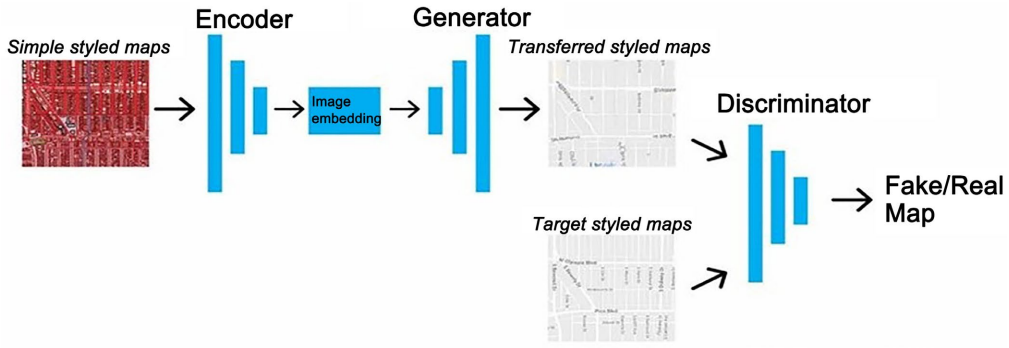

Figure 4. Encoder and Decoder of pix2pix GAN.

3.2.2. Step by Step Procedure of the Proposed Cycle GAN

This Cycle GAN is a model is for focusing unpaired images. In this proposed Cycle

GAN involves in training of two generator models and two discriminator models. The process is given as follows:

Algorithm 3: Makeup Elimination for Unpaired Images

```
Step 1: Define the PatchGAN Discriminator
                    Function BuildDiscriminator():
                    Input_image = Input(shape=(256, 256, 3))  //256x256 RGB input
//First Convolutional Layer
                    x = layers.Conv2D(64, kernel_size=4, strides=2, padding='same')(Input_image)
                    x = layers.LeakyReLU(alpha=0.2)(x)  //Instance normalization is not applied here
// Second Convolutional Layer
                    x = layers.Conv2D(128, kernel_size=4, strides=2, padding='same')(x)
                    x = InstanceNormalization()(x)
                    x = layers.LeakyReLU(alpha=0.2)(x)
//Third Convolutional Layer
                    x = layers.Conv2D(256, kernel_size=4, strides=2, padding='same')(x)
                    x = InstanceNormalization()(x)
                    x = layers.LeakyReLU(alpha=0.2)(x)
//Output Layer
                    x = layers.Conv2D(1, kernel_size=4, padding='same')(x)  # Output a single patch map
//Create and return the model
                    Discriminator = models.Model(inputs=Input_image, outputs=x)
                    Return Discriminator
Step 2: Define the Generator
                  Function BuildGenerator():
                    Input_image = Input(shape=(256, 256, 3))  # 256x256 RGB input
//First Convolutional Layer
                    x = layers.Conv2D(64, kernel_size=7, strides=1, padding='same')(Input_image)
                    x = InstanceNormalization()(x)
                    x = layers.ReLU()(x)
//Second Convolutional Layer
                    x = layers.Conv2D(128, kernel_size=3, strides=2, padding='same')(x)
                    x = InstanceNormalization()(x)
                    x = layers.ReLU()(x)
//Third Convolutional Layer
                    x = layers.Conv2D(256, kernel_size=3, strides=2, padding='same')(x)
                    x = InstanceNormalization()(x)
                    x = layers.ReLU()(x)
//Add Residual Blocks
                    For i in range(6):  //Example: 6 residual blocks
                      Residual_input = x
                      x = layers.Conv2D(256, kernel_size=3, padding='same')(x)
                      x = InstanceNormalization()(x)
                      x = layers.ReLU()(x)
                      x = layers.Conv2D(256, kernel_size=3, padding='same')(x)
                      x = InstanceNormalization()(x)
                      x = layers.Add()([x, Residual_input])  //Skip connection
//Create and return the model
                    Generator = models.Model(inputs=Input_image, outputs=x)
                    Return Generator
Step 3: Compile the Discriminator
                  Discriminator = BuildDiscriminator()
                  Discriminator.compile(optimizer=optimizers.Adam(learning_rate=0.0002, beta_1=0.5),
          loss='binary_crossentropy')
Step 4: Compile the Generator (optional pretraining setup)
                  Generator = BuildGenerator()
                  Generator.compile(optimizer=optimizers.Adam(learning_rate=0.0002,beta_1=0.5),loss='mean_absolut
          e_error')
Step 5 :Train for 4 epochs
                  For epoch in range(4):
                    For batch in TrainingData:
//Training logic for generator and discriminator goes here
                      Pass
```

3.3. Loss Functions to Improve Performance

Four loss functions have been implemented to improve the performance of the model. These loss functions are explained below.

3.3.1. Adversarial Loss

The adversarial loss is calculated based on the probabilities returned by the discriminator network. In the adversarial model, the discriminator network is fed with generated images generated by the generated network [18].

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}\left(D_{\theta_G}\left(I^{LR}\right)\right)$$

Here, $D_{\theta_G}\left(I^{LR}\right)$ is the generated image and $D_{\theta_D}\left(D_{\theta_G}\left(I^{LR}\right)\right)$ represents the probability that the generated image is a real image.

- Adversarial loss—In Adversarial loss, the makeup image is given as an input to Generator B which generates the no makeup image. The no makeup image is given to discriminator B which should discriminate the image as real or fake one.

3.3.2. Identity LOSS

Loss of identity is added to maintain the tone. It says that if the generator receives an image of the target class, it should return the same image.

$$F(x) \approx x \text{ and } G(y) \approx y.$$

λ is a term added to define the relative importance of cycle and identity losses, compared to the GAN losses.

- Identity loss—In Identity loss, the no-makeup image is given as an image to Generator B which should give the same no-makeup image as the output.

3.3.3. Cycle Loss

From left to right: input, cycle consistency loss alone, adversarial loss alone, GAN + forward cycle loss ($F(G(x)) \approx x$, labels → photos) GAN + backward cycle loss ($G(F(y)) \approx y$, photos → labels, Cycle GAN (ours), and ground truth. Both Cycle alone and GAN + backward fail to produce images similar to the target domain [19].

- Forward cycle loss: In forward cycle loss, the makeup image is given as an input to Generator B, which generates the no makeup image. When this no-makeup image is given to Generator A, it should generate the same makeup image. The Forward cycle loss follows the order 2 -> 3 -> 4 -> 5 as shown in **Figure 3**.
- Backward cycle loss: The backward cycle loss is the reverse process of the forward cycle loss. It follows the order 4 -> 5 -> 2 -> 3 as shown in **Figure 3**.

3.4. Data Collection

For imposture identification, the dataset should contain makeup and no makeup images. Due to limited dataset available, dataset such as MIFS (Makeup Induced Face Spoofing), YMU (YouTube Makeup dataset), VMU (Video Makeup dataset), FAM (FAce Makeup), MIW (Makeup In Wild) are combined together. This combined dataset is split into two categories—(i) Makeup and (ii) No makeup images. The dataset contains 949 makeup images and 1085 no makeup contains images as shown in **Figure 5** and **Figure 6**.

Biometric features might be related to ethnic groups (facial structures, skin textures, and iris patterns). The lack of such representation can lead to biased feature

extraction. Systems designed for cross-generational applications (healthcare or education) tend to have a reduction in reliability if different ages are not considered. Gender diversity should also be analyzed in conjunction with ethnicity and age to alleviate the compounding bias and ensure fair performance for all. Achieving dataset diversity in terms of ethnicity, age, and gender would be an ethical and technical requirement for robustness, fairness, and generalizability in biometric models. Well-distributed datasets lead to inclusive, reliable systems deployable in global settings.

3.5. Model Training Step

1) The batch size is fixed at one image. Since the dataset has 949 makeup images, the batches per epoch will be 949, *i.e.*, for one epoch 949 training iterations will be done.

2) A batch of real images and fake images from both domains (makeup and no makeup) is generated, and the fake images are updated in the discriminator's fake image pool.

3) Then, for each iteration, both the generator and discriminator will be trained over one batch of samples, and the model will be saved.

3.6. Model Testing Step

1) The model from makeup to no makeup is loaded.

2) The input image is resized and normalized and it will be passed to the loaded model.

3) The model will generate the no-makeup image and plot it.

4. Experimental Result and Performance Analysis

4.1. Sample Images from Dataset

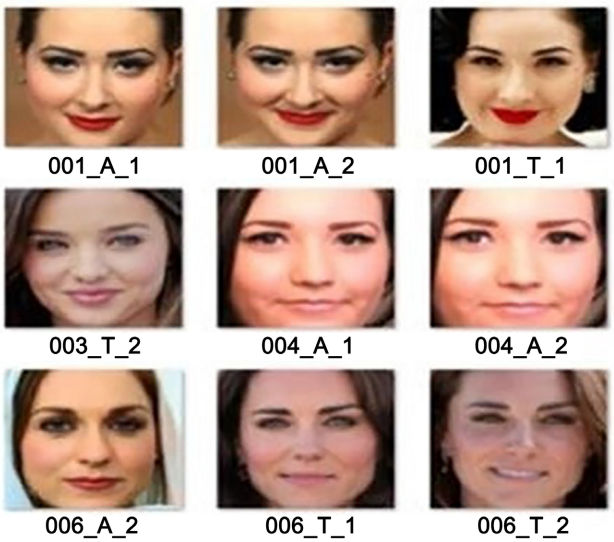

Figure 5. Sample makeup images.

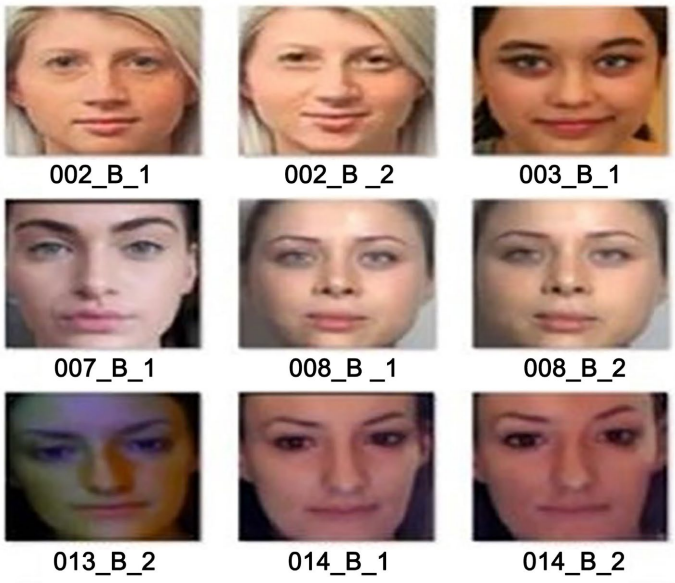

Figure 6. Sample no makeup images.

4.2. Data Pre-Processing

Images are converted into pixel values and stored in an array. Then, normalization is applied to speed up the convergence as shown in Figure 7.

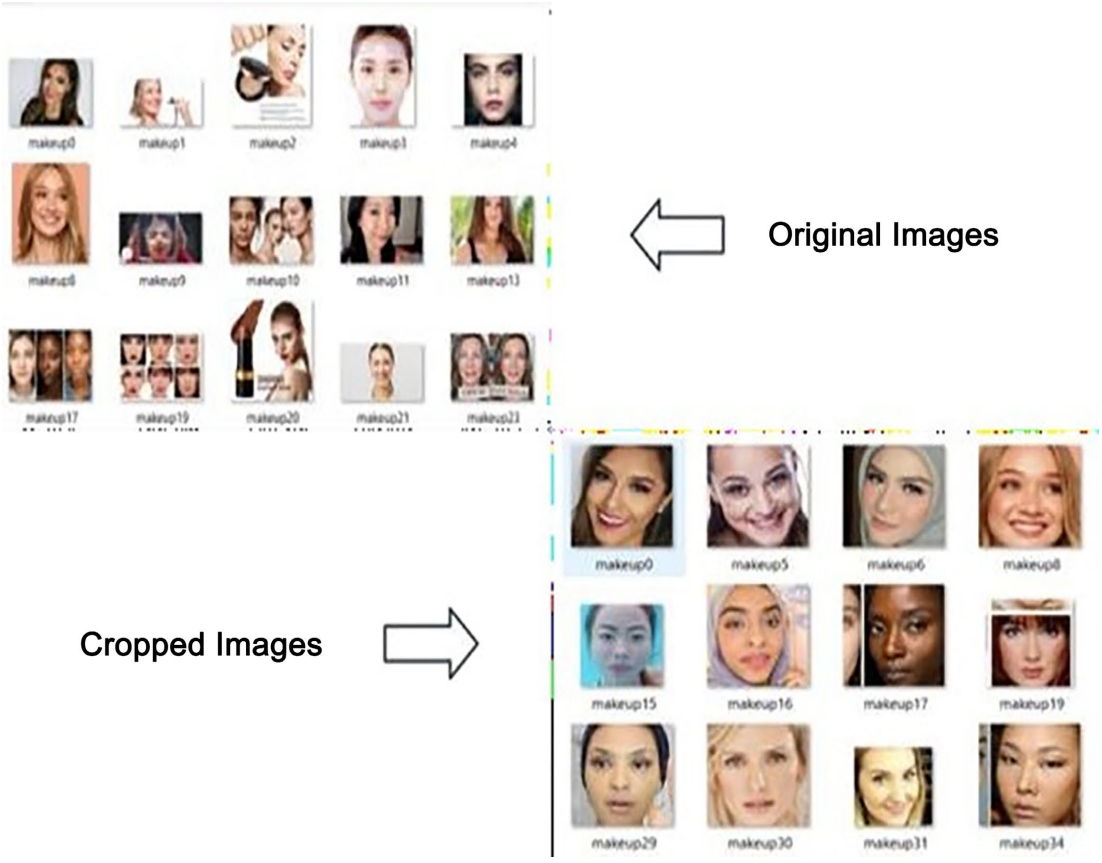

Figure 7. Pre-processed cropped images from original images.

4.3. Feature Extraction in Makeup Classification Model (Figure 8)

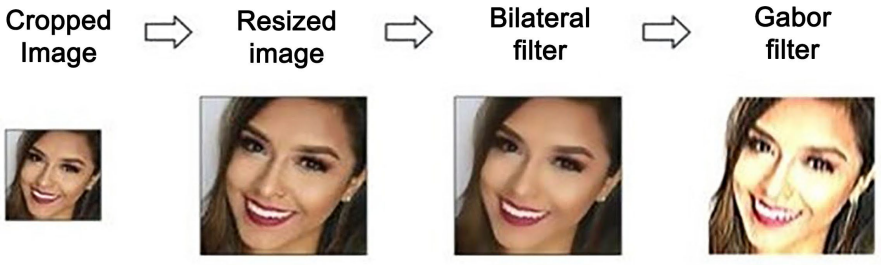

Figure 8. Feature extraction using gabor filter.

4.4. Sample Result of Makeup Elimination from Dataset

The detected makeup image from the classification model is passed into the makeup removal model to get the no makeup image as the output shown in Figure 9.

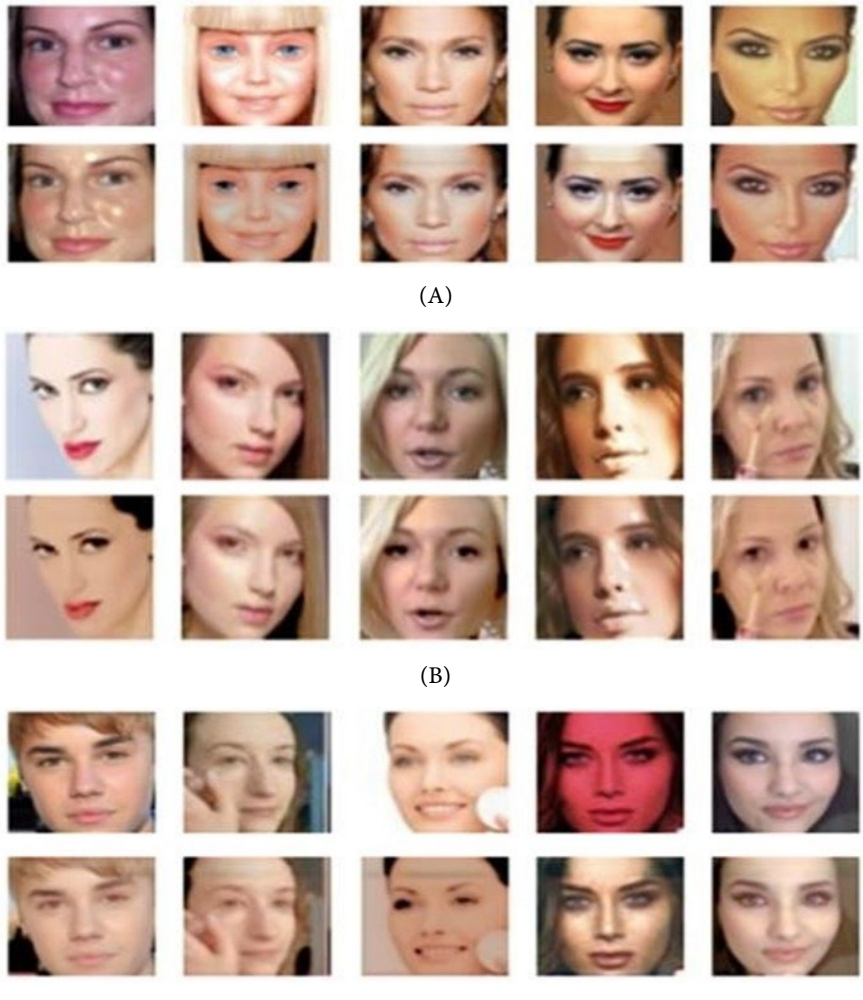

Figure 9. Makeup to no makeup generation (of each image A, B, and c the first row contains makeup images and the corresponding image after makeup removal is in the second row).

4.5. Analysis of Evaluation Metric in Makeup Classification Model

The classification model is used to identify whether the person is wearing makeup or not. When an image is passed to this model it outputs a value ranging from 0 to 1. The performance of the classification model is evaluated in terms of accuracy, loss, precision, recall, and f1 score (See Table 1).

Table 1. Evaluation of makeup classification model.

Evaluation metrics	Training size = 90% Testing size = 10%
Accuracy (%)	97.48
Precision (%)	97.44
Recall (%)	96.59
F1 score (%)	95.39

4.6. Comparison of Proposed GAN Models Based on Image Quality

By training both pix2pix GAN and Cycle GAN and comparing their image quality measure them as shown in Table 2 [20] [21]. It has been proved that Cycle GAN is more efficient than pix2pix GAN which is visually shown in Figure 10.

Table 2. Comparing the quality measure of gan models.

GAN models	SSIM (Structural similarity index measure)	FSIM (Feature similarity index measure)	FID (Frechet inception distance)
Pix2pix	42.86	69.34	19.50
Cycle	48.70	75.48	39.48

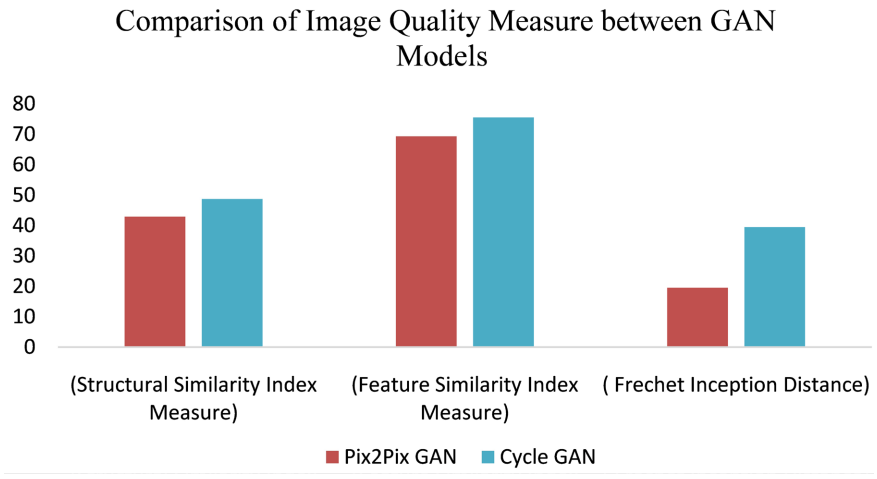

Figure 10. Image quality measure difference.

4.7. Error Analysis for Identifying Common Failure Modes

Error analysis helps understand the biometric system's weakness, mainly the

failure modes and their consequences in security consideration. The failure modes are identified, and targeted improvements are presented to increase overall performance.

4.7.1. False Acceptances (False Positives)

Poor threshold: A low threshold increases the chances of a false positive. Similar biometric characteristics exist between individuals (e.g., family members in facial recognition). Noise and artifacts simulating genuine features. Highly serious security risk because of unauthorized users. Primarily critical in access control and financial applications.

4.7.2. False Rejects (False Negatives)

Biometric data variability (e.g., changes in makeup, lighting, or expressions in facial recognition). Partial or occluded input (e.g., fingerprint smudges or incomplete face scans). Model over fitting to a specific pattern in data. Less direct security risk but damages user experience and trust. Might drive users to system circumvention or insecure alternatives.

4.7.3. Spoof Type Attacks

Lack of enough anti-spoofing mechanisms in the system. Reliance on static features that are very easily replicated by attackers. A significant risk for security because unauthorized access would be granted by attackers.

Performing a detailed error analysis is essential for identifying common failure modes in biometric systems and their impact on security. By addressing these issues through targeted mitigations, the system becomes more robust, reliable, and secure, ensuring user confidence and operational success.

5. Conclusions

Thus, by implementing this method in the existing biometric system, the real identity of the person can be identified, even though the person intends to fraudulently access the system by wearing makeup. Thus, the vulnerability caused by makeup attacks can be prevented which improves the security of the biometric system. The captured image from a live video camera is passed to the makeup classification model to identify whether the person is wearing makeup or not. After this, the identified makeup image is passed to the makeup removal model to extract the bare face.

To improve this model further, instead of an image dataset, video samples can be collected. It will be more appropriate for the biometric system since it captures real-time instances. The security of the biometric system can be increased by preventing not only the makeup attack but also other presentation attacks like spoofing and morphing which are more likely. So, a combination of all the attack prevention models can be developed.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Yu, Z., Li, X., Niu, X., Shi, J.G. and Zhao, G. (2020) Face Anti-Spoofing with Human Material Perception. *Computer Vision-ECCV* 2020, Glasgow, 23-28 August 2020, 557-575. https://doi.org/10.1007/978-3-030-58571-6_33

[2] Ming, Z., Visani, M., Luqman, M.M. and Burie, J.C. (2020) A Survey on Anti-Spoofing Methods for Facial Recognition with RGB Cameras of Generic Consumer Devices. *Journal of Imaging*, **6**, Article 139. https://doi.org/10.3390/jimaging6120139

[3] Kotwal, K., Mostaani, Z. and Marcel, S. (2020) Detection of Age-Induced Makeup Attacks on Face Recognition Systems Using Multi-Layer Deep Features. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, **2**, 15-25. https://doi.org/10.1109/TBIOM.2019.2946175

[4] Sun, Z.Y., Liu, F., Liu, W., Xiong, S.W. and Liu, W.X. (2020) Local Facial Makeup Transfer via Disentangled Representation. *Computer Vision-ACCV* 2020, 21 June 2020.

[5] Majumdar, P., Agarwal, A., Vatsa, M. and Singh, R. (2021) Facial Retouching and Alteration Detection. Springer. https://doi.org/10.1007/978-3-030-87664-7_17

[6] Kips, R., Gori, P., Perrot, M. and Bloch, I. (2020) CA-GAN: Weakly Supervised Color Aware GAN for Controllable Makeup Transfer. *Computer Vision-ECCV 2020 Workshops*, Glasgow, 23-28 August 2020, 280-296. https://doi.org/10.1007/978-3-030-67070-2_17

[7] Gorabal, J.V. and Manjaiah, D.H. (2021) Texture Analysis for Face Recognition. *International Journal of Graphics and Multimedia* (*IJGM*), **4**, 20-30.

[8] Ramachandra, R. and Busch, C. (2017) Presentation Attack Detection Methods for Face Recognition Systems. Norwegian University of Science and Technology (NTNU).

[9] Rathgeb, C., Drozdowski, P., Fisher, D. and Busch, C. (2020) Vulnerability Assessment and Detection of Makeup Presentation Attacks. *IEEE*: 8*th International Workshop on Biometrics and Forensics* (*IWBF*), Porto, 29-30 April 2020, 1-6. https://doi.org/10.1109/IWBF49977.2020.9107961

[10] Du, Y.T., Qiao, T., Xu, M. and Zheng, N. (2021) Towards Face Presentation Attack Detection Based on Residual Color Texture Representation. *Journal of Security and Communication Networks*, **2021**, Article ID: 6652727. https://doi.org/10.1155/2021/6652727

[11] Moon, Y., Ryoo, I., and Kim, S. (2021) Face Antispoofing Method Using Color Texture Segmentation on FPGA. *Journal of Security and Communication Networks*, **2021**, Article ID: 9939232. https://doi.org/10.1155/2021/9939232

[12] Huang, Y., Zhang, W. and Wang, J. (2020) Deep Frequent Spatial-Temporal Learning for Face Anti-Spoofing. arXiv Preprint arXiv:2002.03723.

[13] Rathgeb. C, Drozdowski. P. and Busch, C. (2021) Detection of Makeup Presentation Attacks Based on Deep Face Representations. 2020 25*th International Conference on Pattern Recognition* (*ICPR*), Milan, 10-15 January 2021, 3443-3450. https://doi.org/10.1109/ICPR48806.2021.9413347

[14] Shao, R., Lan, X.Y., Li, J.W. and Yuen, P.C. (2019) Multi-Adversarial Discriminative Deep Domain Generalization for Face Presentation Attack Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 15-20 June 2019, 10015-10023. https://doi.org/10.1109/CVPR.2019.01026

[15] Karmakar, D., Mukherjee, P. and Datta, M. (2021) Spoofed Facial Presentation Attack Detection by Multivariate Gradient Descriptor in Micro-Expression Region. *Pattern*

Recognition and Image Analysis, **31**, 285-294, https://doi.org/10.1134/S1054661821020097

[16] Anchieta, N.M., Mafra, A.L., Hokama, R.T., *et al.* (2021) Makeup and Its Application Simulation Affect Women's Self-Perceptions. *Archives of Sexual Behavior*, **50**, 3777-3784. https://doi.org/10.1007/s10508-021-02127-0

[17] Yu, Z., Li, X., Shi, J., Xia, Z. and Zhao, G. (2021) Revisiting Pixel-Wise Supervision for Face Anti-Spoofing. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, **3**, 285-295. https://doi.org/10.1109/TBIOM.2021.3065526

[18] Liu, M., Mu, J., Yu, Z., Ruan, K., Shu, B. and Yang, J. (2022) Adversarial Learning and Decomposition-Based Domain Generalization for Face Anti-Spoofing. *Pattern Recognition Letters*, **155**, 171-177. https://doi.org/10.1016/j.patrec.2021.10.014

[19] Cai, T., Chen, F., Liu, W., Xie, X. and Liu, Z. (2022) Face Anti-Spoofing via Conditional Adversarial Domain Generalization. *Journal of Ambient Intelligence and Humanized Computing*, **14**, 1-14. https://doi.org/10.1007/s12652-022-03884-z

[20] Wen, D., Han, H. and Jain, A.K. (2015) Face Spoof Detection with Image Distortion Analysis. *IEEE Transactions on Information Forensics and Security*, **10**, 746-761. https://doi.org/10.1109/TIFS.2015.2400395

[21] Chang, H.H. and Yeh, C.H. (2022) Face Anti-Spoofing Detection Based on Multi-scale Image Quality Assessment. *Image and Vision Computing*, **121**, Article 104428. https://doi.org/10.1016/j.imavis.2022.104428