mathematics

MDPI

Article

Generating Synthetic Disguised Faces with Cycle-Consistency Loss and an Automated Filtering Algorithm

Mobeen Ahmad †, Usman Cheema †, Muhammad Abdullah, Seungbin Moon and Dongil Han *

Department of Computer Engineering, Sejong University, Seoul 05006, Korea; mobeen@sju.ac.kr (M.A.); usman@sju.ac.kr (U.C.); shehzi@sju.ac.kr (M.A.); sbmoon@sejong.ac.kr (S.M.)

* Correspondence: dihan@sejong.ac.kr

† These authors contributed equally to this work.

Abstract: Applications for facial recognition have eased the process of personal identification. However, there are increasing concerns about the performance of these systems against the challenges of presentation attacks, spoofing, and disguises. One of the reasons for the lack of a robustness of facial recognition algorithms in these challenges is the limited amount of suitable training data. This lack of training data can be addressed by creating a database with the subjects having several disguises, but this is an expensive process. Another approach is to use generative adversarial networks to synthesize facial images with the required disguise add-ons. In this paper, we present a synthetic disguised face database for the training and evaluation of robust facial recognition algorithms. Furthermore, we present a methodology for generating synthetic facial images for the desired disguise add-ons. Cycle-consistency loss is used to generate facial images with disguises, e.g., fake beards, makeup, and glasses, from normal face images. Additionally, an automated filtering scheme is presented for automated data filtering from the synthesized faces. Finally, facial recognition experiments are performed on the proposed synthetic data to show the efficacy of the proposed methodology and the presented database. Training on the proposed database achieves an improvement in the rank-1 recognition rate (68.3%), over a model trained on the original nondisguised face images.

Keywords: disguised face; synthetic database; synthetic faces; generative adversarial networks; CycleGAN; style transfer; data augmentation; Sejong Face Database; Synthetic Disguised Face Database

Citation: Ahmad, M.; Cheema, U.; Abdullah, M.; Moon, S.; Han, D. Generating Synthetic Disguised Faces with Cycle-Consistency Loss and an Automated Filtering Algorithm. *Mathematics* **2022**, *10*, 4. https://doi.org/10.3390/math10010004

Academic Editor: Bo-Hao Chen

Received: 19 November 2021
Accepted: 16 December 2021
Published: 21 December 2021

1. Introduction

Facial recognition (FR) has been a topic of interest for the last few decades. Among the computer vision domains, FR is the widely adopted solution in the industry and is considered a solved problem in controlled environments. A controlled environment is defined as one in which the images are captured in the frontal pose, with good illumination, and a neutral expression, and in which the subject is not trying to avoid recognition. However, in circumstances such as a spoofing scenario, the subject might try to hide his/her identity by using a disguise. Such scenarios pose a challenging problem for applications of FR. FR algorithms are trained on facial features, which are unique for each subject. If those features are hidden by the subject, the algorithm might fail to perform. There can be different outcomes in such a scenario, such as the failure to recognize, recognizing a different identity, or the failure to detect the face altogether.

Earlier FR research has been focused on the challenges of pose, illumination, and expression. However, recently the focus of research has diverted towards more complex issues, such as face alterations due to plastic surgery [1], twins [2], single-sample facial recognition [3], sketch-to-photo matching [4,5], multispectrum matching [6–8], facial expression recognition [9,10], and disguise [11–14]. As more and more systems are relying on automated FR algorithms, there is an increasing need to solve the challenges of facial disguise because of the emerging security threats.

Convolutional neural networks (CNN) have been used to solve the problem of disguised FR. However, current CNN models require a large amount of training data to achieve high performances on specialized tasks, such as person classification. While normal facial images can be acquired through the web, social media, and other online resources, the collection of disguised face datasets require a more methodological approach. The collection of face images with required disguises is a resource-intensive process. There are several ways in which individuals can disguise themselves by utilizing add-ons, such as masks, glasses, caps, scarfs, etc. Apart from these, fake but realistic-looking add-ons, such as fake mustaches, fake beards, and facial makeup, can be used to spoof an FR application. To create a comprehensive database of disguised faces, images need to be captured with the subjects wearing a wide array of disguise add-ons in various combinations. Additionally, the privacy concerns of subjects make the collection of a large-scale face database a difficult process. Because of these difficulties, there are limited publicly available databases (DBs) that feature face images with disguises capable of demonstrating real-world scenarios. The research in the domain of disguised facial recognition is faced with the issue of a lack of suitable training data. Currently, the available disguised face DBs are not sufficient for the training of FR algorithms. IIITD In, and the Beyond Visible Spectrum Disguise database, I^2BVSD [15], contain celebrity images with a limited set of disguises, but no labeling information for the disguises is provided. There are also some multispectral DBs, such as visible-depth [16], visible-thermal [17], and visible-infrared [18], which feature disguise images. However, as the focus of these databases is multispectral FR, there are insufficient variations of disguises. Even when a database with a focus on disguise add-ons is constructed, it is practically impossible to include all possible add-ons. Furthermore, from an applicability viewpoint, it is challenging to collect such a dataset. This constraint motivated us to design a system capable of generating disguised face images for scenarios where these are not available.

Generative adversarial networks (GANs) [19] have achieved breakthrough performances in the domain of synthetic image generation [20–22]. State-of-the-art GANs can generate realistic images that are identical to photographs. Among the several applications of GANs, style transfer has garnered significant interest from the computer vision community. Style transfer techniques enable the transfer of the style of one image to that of another while maintaining the context. This enables the reproduction of art pieces by great artists by synthesizing current-era pictures in a specific artist's style. CycleGAN [23] is one of the significant works in the recent literature, and it has been used in numerous applications. We aim to utilize this image style transferability to focus on specific facial regions and to generate images with the desired features while retaining the subject's identity features. We propose a methodology for disguised face synthesis and present a synthetic facial disguise database for the development of robust FR algorithms. A disguised face synthesis, based on generative adversarial networks (GAN) [19] and cycle-consistency loss [23], is performed in order to extend the available Sejong Face Database [13]. The proposed method for disguised face synthesis uses the SFD as its seed database. As the method requires disguised sample images, the synthesized database contains the same add-on variations and subject identities as the SFD. The number of face images for each disguise is extended from the original 5250 to 12,600 in the proposed database.

The data used in this study are categorized into three categories on the basis of their natures: The facial images of subjects without any face add-ons or accessories are referred to as "normal" images. Facial images with face add-ons, such as glasses, masks, caps, etc., are referred to as "disguise" images. Facial images that are captured through a camera are referred to as "real" images, whereas face images that are synthesized through the proposed method are referred to as "Gen." (generated) images. Similarly, the term "real normal" refers to the images without any face disguise add-ons captured by the camera. The term "real disguise" refers to the images where there is a disguise add-on, and the images are captured by the camera. The term "Gen. disguise" refers to the images digitally

generated using the proposed methodology. For simplicity, these terms are used in the rest of the text, as shown in Table 1.

Table 1. This table represents the data splits that were used in the facial recognition experiments.

Source	Add-On	Abbreviation
Photographed	No	Real normal
Photographed	Yes	Real Disguise
Synthetically generated	Yes	Gen. Disguise

Contributions

1. A synthetic disguised face database, namely, the "Synthetic Disguised Face Database", is presented as a training and evaluation resource for the robustness of FR algorithms to disguise. The presented DB features facial images with 13 synthetically generated disguise variations;
2. A methodology employing a GAN and cycle-consistency loss is proposed for synthetic disguised face generation, which will allow future research to extend the existing facial databases. The methodology can be applied to generate disguise add-ons not covered in this study;
3. The proposed method can be employed for runtime data augmentation during the training of facial recognition algorithms. Our experimental works prove the value of the proposed methodology over traditional methods of data augmentation;
4. A comprehensive analysis is presented by benchmarking the proposed "Synthetic Disguised Face Database" using the state-of-the-art FR method for different experimental configurations. Improved FR performance is achieved using the proposed data on real add-on images;
5. An automated filtering scheme is presented that filters out the low-quality image samples from the generated pool of synthetic images. The efficacy of the filtering is shown through the experimental results.

The rest of the manuscript is organized as follows: First, the proposed disguised face synthesis and the automated filtering methodologies are presented in Section 2. The details of the proposed database are discussed in Section 3. Section 4 presents the results of FR experiments using the proposed database and proves the efficacy of the methods and data. Finally, the manuscript is concluded in Section 5.

2. Related Work

We review the related works on the currently available disguised face databases, the currently proposed methods for facial synthesis, and, finally, the available methods for image synthesis. Our review analysis shows a need for disguised face databases, as well as the inability of the current methods to be adapted for disguised face synthesis.

2.1. Face Databases

In this study, various publicly available disguised face DBs are reviewed, along with their advantages and disadvantages. Only the databases pertaining to disguise or artificially synthesized face databases are discussed in this study.

2.1.1. Databases with Disguised Facial Images

A large variety of disguise add-ons and sufficient training data are required for the training of modern CNNs. **I^{2}BVSD** [15,17] contains frontal pose images with neutral expressions captured under constant illumination. Several disguise variations, such as fake facial hair, caps, wigs, masks, and glasses, are included in the database. A total of 75 subjects of South Asian ethnicity are included in the DB, of which 60 are male and 15 are female. There are five distinct variations of disguises available in the DB, namely, variations

in hairstyles, beards, and mustaches, glasses, caps, and masks. Another variation is a combination of two or more disguises. However, the DB does not provide disguise labels, which makes it difficult to use it for the development of disguised FR models. There are a total of 681 images for each modality, with at least one frontal face image, and from five to nine frontal disguised images per subject. The **BRSU Spoof Database** [24,25] is a multispectral DB, with images captured in visible and infrared modalities at frequencies of 935, 1060, 1200, and 1550 nm. There are several variations in the DB that render it challenging, such as expression, makeup, 3D masks, fake beards, glasses, fake noses, and presentation attacks. However, the DB features only five subjects, with nine to thirty disguise add-ons per subject. The emphasis of the BRSU Spoof DB is towards the multispectral domain. The **Spectral Disguise Face Database** [26] is a collection of 54 male subjects with normal and disguised face images. The DB features images captured in the visible and near-infrared spectrums ranging from 530 nm to 1000 nm. This DB is further divided into two categories; one is the bona fide set with natural images without disguise, and the second set has two disguise variations. However, the disguise set is limited, as it only features fake normal-length-beard and fake-long-beard variations. A total of 22 subjects in the bona fide set have natural beards, and the rest have natural mustaches. This DB lacks variation in disguises in order to properly train an inclusive disguise detection model. The **CASIA SURF Database** [18] is a large-scale multimodal facial presentation-attack database. It features 1000 Chinese subjects in three modalities: RGB, Depth, and Infrared. There are six attack categories in the database; however, the attacks used in the database are mainly the subjects holding a face photo, printed on a paper with six different configurations. As such attacks are irrelevant in public-domain facilities, such as in airport security, this data cannot be used for the training of a public FR application. The **Sejong Face Database** (SFD) [13] is a multimodal disguised face database, featuring 100 subjects, with 13 variations of facial disguises. The database contains caps, scarfs, wigs, and fake beards, etc., as common face add-ons. It consists of two subsets, namely, subset-A and subset-B. Subset-A contains the face images of 30 subjects, with 16 males and 14 females. Subset-B contains images of 70 subjects, with 5–10 images for each disguise. Sample images from the discussed databases are shown in Figure 1.

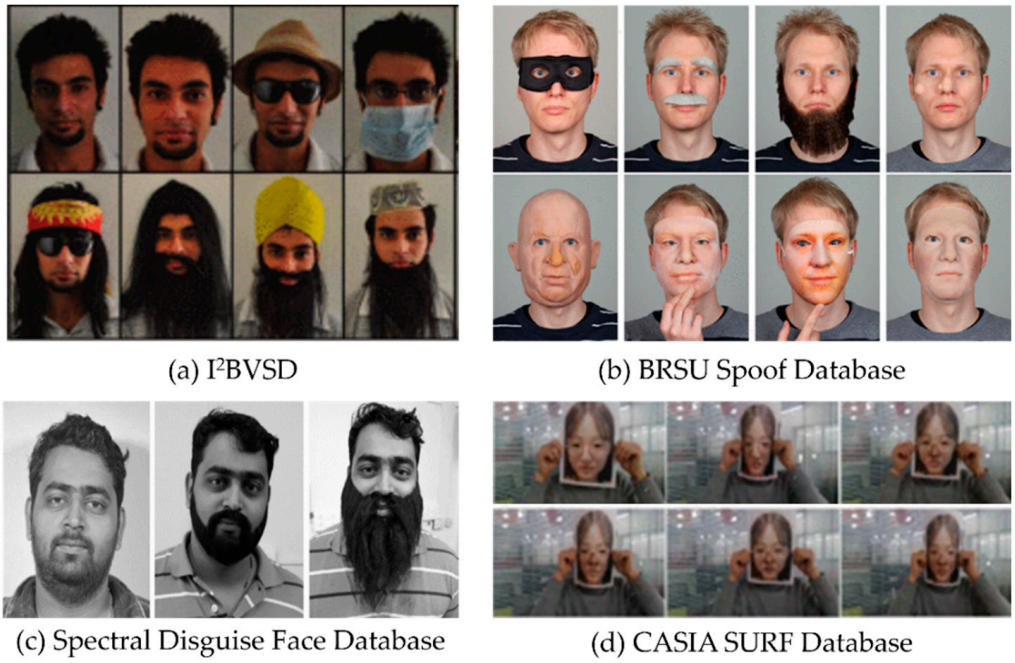

Figure 1. Sample images from the currently available disguised face databases: (**a**) I^2BVSD; (**b**) the BRSU Spoof Database; (**c**) the Spectral Disguise Face Database; and (**d**) the CASIA SURF Database.

A summary of disguised face databases, compared with the proposed Synthetic Disguised Face Database, is presented in Table 2.

Table 2. A summary of the currently available disguised face databases and the proposed Synthetic Disguised Face Database (Syn-DFD).

Database	Total No. of Subjects	Total No. of Images (Visible)	No. of Disguise Images/Subjects (Visible)	Disguise Labels	Gender Male:Female	No. of Add-Ons	Combination Add-Ons
I^2BVSD [15]	75	681	5–9	✘	60:15	5	✔
BRSU [24]	5	35	4–12	✔	4:1	4–12	✔
SDFD [26]	54	285	10	✔	54:0	3	✔
SFD-A [13]	30	390	13	✔	16:14	13	✔
SFD-B [13]	70	5250	75	✔	44:26	13	✔
Proposed Database (Syn-DFD)	**70**	**12,600**	**180**	✔	**44:26**	**13**	✔

2.1.2. Synthetic Face Databases

Of the previous works that have proposed methods for the construction of synthetic face databases, few methods propose 3D morphable models to replicate the frontal facial surface to generate facial images. Most prominent is the MIT-CBCL Face Recognition Database [27], which consists of frontal-, half-profile-, and profile-view synthetic images rendered from the 3D head models of 10 subjects. The head models are generated by fitting a morphable model to the high-resolution training images. The database features a total of 324 images per subject. Another database constructed from a 3D morphable model is the Basel Face Model (BFM) [28]. The BFM is a 3D morphable face model that is constructed from 100 male and 100 female subject images. It consists of a 3D shape model, covering the face surface from ear to ear, and a texture model. The database can also be considered a metadatabase, which allows for the creation of accurately labeled synthetic training and testing images. Apart from 3D morphable models, other synthetic databases are also available. The VMU (Virtual Makeup) Dataset [29] contains the face images of 51 Caucasian female subjects. The images are gathered from the FRGC (Face Recognition Grand Challenge) dataset [30], and makeup is applied synthetically. There are three types of makeup variations in the dataset: (1) The application of lipstick; (2) The application of eye makeup; (3) The application of full makeup. The dataset contains four images per subject, i.e., one before makeup, and one image for each makeup type. Another synthetic database is the Specs on Faces (SoF) Dataset [31], which is a collection of 42,592 images for 112 subjects (66 males and 46 females) who wear glasses, under different illumination conditions. In addition to glasses, the nose and mouth are occluded using a white block. A sample of these databases is shown in Figure 2.

As can be seen from Figure 2, the disguise variations contained in these databases are limited, which makes them insufficient for the realistic challenges of FR spoofing. Moreover, current methods utilize conventional computer vision techniques, which are difficult to generalize, in order to generate a range of face disguise variations, whereas the focus of this study is to present a method for disguised facial synthesis that can be generalized to a wide variety of applications.

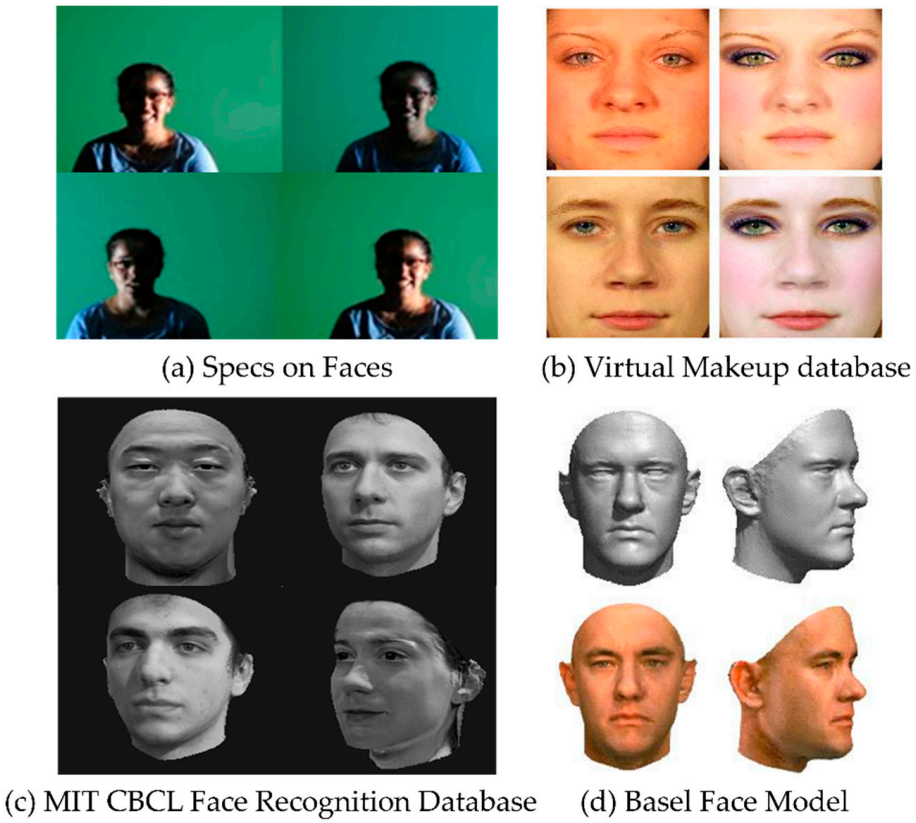

Figure 2. Sample images from various synthetic face databases: (**a**) Specs on Faces; (**b**) the Virtual Makeup Database; (**c**) the MIT CBCL Face Recognition Database; and (**d**) the Basel Face Model.

2.2. Image Synthesis Based on Generative Adversarial Networks

Earlier image-to-image translation methods are limited by the paired image requirement, i.e., the X and Y images need to be paired such that the positioning and orientation of the objects in both images should match, pixel-to-pixel. The most prominent method among paired image-to-image translation is pix2pix [32]. This method deals with synthesizing photos from label maps, reconstructing objects from edge maps, and colorizing images. This proposed work is motivated by the idea of generalizing tasks at a higher level. For instance, if a network is dictated to minimize a specific loss, the weights are trained such that the network learns to only minimize that specific loss function.

The critical problem is that the hand-engineered loss functions are not comprehensive; they are the mathematical approximations of what could be the best way to formulate (describe) a problem. Hence, problems, such as blurry output images, are commonplace in image-to-image translation methods, which use Euclidean distance as a loss. The authors of pix2pix suggest that, instead of designing task-specific losses for various image-to-image translation applications, the loss function should be left to be learned by the network itself. This is the most intuitive way of solving problems using neural networks, moving forward from hand-engineered features to feature learning. The next thing after feature learning is to let the network formulate its loss, or design networks, automatically [33]. The main idea behind GANs is their ability to learn the loss function. Before GANs, the CNNs were only good at minimizing the problem formulated in the shape of the employed loss function. However, the discriminator network in a GAN is responsible for predicting whether the image is real or fake. The network finds that images that are blurred, or otherwise inappropriate, cannot be classified as real.

Several methods have been explored in the domain of image style transfer. The key improvements in the domain of image-to-image translation arrived after the advent of GANs. The most prominent of such works are pix2pix [32] and CycleGAN [23]. Pix2pix is

limited by the condition that paired images are required. One-to-one paired images are challenging to obtain in many domains. Moreover, in some domains, they do not exist. For instance, the horse-zebra is an example of object transfiguration, which deals with the transformation of one object to a similar, but different, object. In such a scenario, the output is not even well-defined, so the availability of the paired input–output images is out of the question. The case is similar with the disguised faces, in the sense that it is not a texture or style transfer but, rather, a variation of object transfiguration. In object transfiguration, the whole object's texture is replaced by the target object's texture. However, while generating disguised facial images from normal faces, a specific (disguised) region needs to be transfigured while preserving the normal face regions (to preserve identity). Recently, there have been studies with regard to spoofing neural networks by introducing noise to the test images.

It is interesting to note that the noise artifacts invisible to the human eye can cause neural networks to misclassify with high confidence [34]. Such artifacts are also generated by GANs, which can play an adversarial role in the classification model. All these factors make synthetic disguised face generation a complex and challenging problem. One solution is to extract the facial regions manually, or by employing face-landmark-based methods to point the transfiguration algorithm to specific regions. However, such methodologies cannot be generalized for all disguises, and worse, most of the disguises obscure the key points deemed essential for landmark recognition.

3. Proposed Methodology

We propose to solve the problem of the lack of disguised face databases by employing image-to-image translation. The image-to-image translation is a domain of computer vision where the goal is to find the mapping between an input and output image. Various large-scale face databases are publicly available for the training of facial recognition systems. If a disguised face DB of the same scale is to be constructed, it will require significant resources and time. However, by virtue of the state-of-the-art image-to-image style transfer methods, the generation of synthetic disguised faces is possible by finding a mapping, $G : X \rightarrow Y$, between existing disguised and nondisguised facial images.

In this study, the use of cycle-consistency loss is proposed, which allows us to learn the mapping function from one domain to another, rather than from one sample image to another. This transferability of the cycle-consistency loss eliminates the necessity of paired input–output images for training. The goal is to transfer only the disguise features from the X domain to the Y domain, while preserving the identity features of the X domain, given samples x and y with the data distributions, $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$, respectively. The $p_{data}(x)$ can be defined as a set containing data distributions of the disguise features, $p_{disguise}(x)$, and the identity features, $p_{id}(x)$, as shown in Equation (1). The problem can be formulated to find a mapping, G, that preserves the $p_{id}(x)$ while transferring $p_{disguise}(y)$ from Domain X to Y. Therefore, the data distribution of a face image, denoted by $x \sim p_{data}(x)$, can be formulated as follows:

$$p_{data}(x) = [p_{id}(x),\ p_{disguise}(x)] \tag{1}$$

$$p_{data}(y) = [p_{id}(y),\ p_{disguise}(y)] \tag{2}$$

$$\text{Find } G : X \rightarrow Y \tag{3}$$

$$\text{such that } p_{data}(G(x)) = [p_{id}(x),\ p_{disguise}(y)] \tag{4}$$

where $p_{data}(\cdot)$, $p_{id}(\cdot)$, and $p_{disguise}(\cdot)$ are the feature data distributions for the image, subject identity, and disguise add-on, respectively. X is the input domain and Y is the output domain, and G is the learned mapping function. Simply, the data distribution of an output sample, $G(x)$, generated by applying a mapping, G, on sample x, shall be bounded by the data distribution of the disguise features of Domain Y and the data distribution of

the identity features of Domain *X*. The mapping can be defined by Equation (3), where it fulfills the criteria defined in Equation (4).

Finally, an automatic filtering method is proposed, which allows us to filter out the poor-quality samples from the pool of synthetically generated images. An overall pipeline of the proposed work is shown in Figure 3.

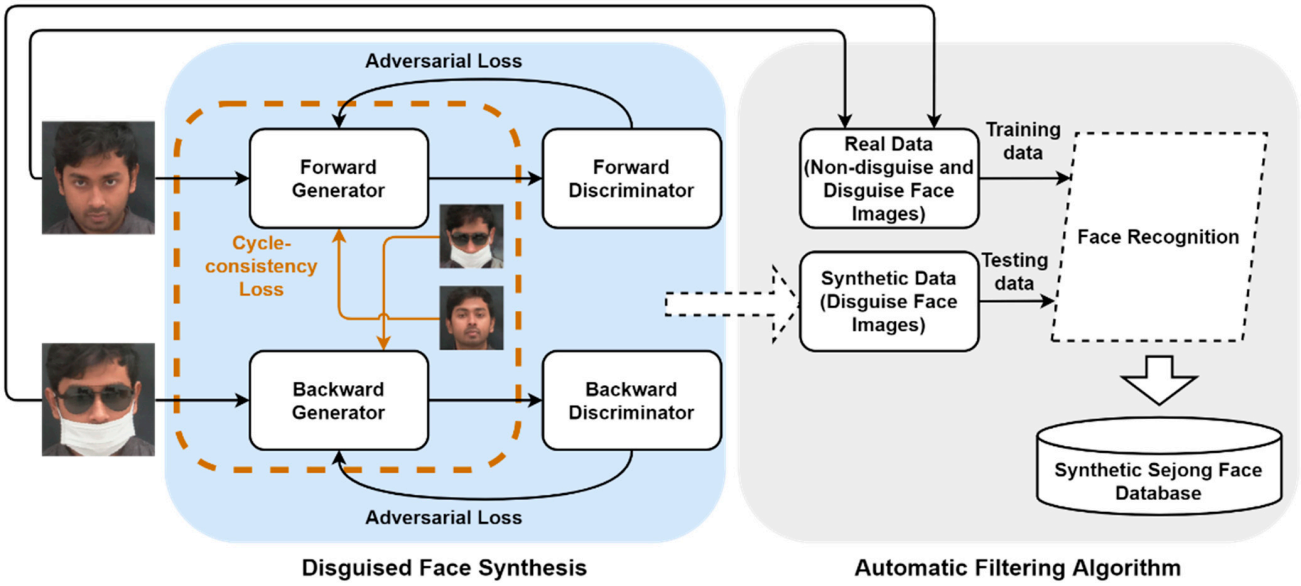

Figure 3. An overview of the proposed methodology. The right side of the figure depicts the disguised facial synthesis module, and the left portion presents the schematic of the proposed automated filtering algorithm.

3.1. Training the Disguised Face Database

In this study, the SFD [13] is used to demonstrate that the CycleGAN can be effectively used for generating disguised facial images from nondisguised facial images in an unpaired input–output setting. The SFD was chosen in light of an analysis of various publicly available disguised face databases. It can be observed, in Table 2, that the diversification provided by the SFD is comparatively significant. Moreover, this allows for performing experiments with various add-ons, a feature missing in other databases. Subset-B from the SFD was used in this study. Sample images from the SFD are shown in Figure 4.

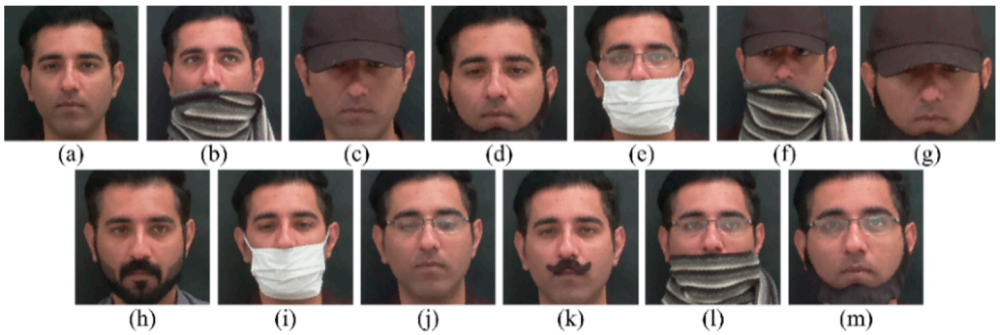

Figure 4. Sample real disguised face images from the Sejong Face Database that are used as seed images for training the synthetic disguised face generation model: (**a**) normal face; (**b**) scarf; (**c**) cap; (**d**) fake beard; (**e**) glasses and mask; (**f**) cap and scarf; (**g**) cap and fake beard; (**h**) real face with a beard; (**i**) mask; (**j**) glasses; (**k**) fake mustache; (**l**) glasses and scarf; and (**m**) glasses and fake beard.

In the SFD, the data are available such that paired input–output training samples are available and can be fed to the network. However, this will lead the model to fit on the

identity of the subjects, whereas our goal is to train a generalized model that can learn a disguise add-on while distilling the identity information. Therefore, the data are divided on the basis of the disguised add-ons, regardless of the subjects. All 13 add-ons from the SFD were used to generate the disguised facial images.

3.2. Disguised Face Synthesis

Among previous style transfer methods, CycleGAN seems promising because of the transitive nature of cycle-consistency loss. Instead of complicating the problem by defining several loss terms related to faces and nondisguised faces, a simple loss is employed, which verifies the generated images by generating nondisguised facial images from the generated disguised facial images. The complete loss formulation is described in this section.

The goal of the CycleGAN is to learn two mappings, $G : X \rightarrow Y$ and $F : Y \rightarrow X$, between two domains, X and Y, given the training samples, $\{x_i\}_{i=1}^{N}$, where $x_i \in X$ and $\{y_i\}_{j=1}^{M}$ where $y_i \in Y$. Additionally, two adversarial discriminators, namely, D_X and D_Y, are also part of the CycleGAN. The discriminator, D_X, is responsible for distinguishing the real samples, $\{x\}$, from the generated samples, $\{F(y)\}$, and, similarly, D_Y aims to differentiate between the real samples, $\{y\}$, and the generated samples, $\{G(x)\}$. Hence, the adversarial loss is computed as shown in Equation (5), such that the data distribution of the generated samples matches the data distribution of the real samples:

$$\mathcal{L}_{GAN}(G,\ D_Y, X, Y) = \mathrm{E}_y[\log(D_Y(y))] + \mathrm{E}_x[\log(1 - D_Y(G(x)))] \tag{5}$$

where $\mathcal{L}_{GAN}$ is the computed loss; G is the mapping function applied by the generator for the $X \rightarrow Y$ translation; $D.$ is the classification function applied by the discriminator; X is the input domain; and Y is the output domain. E_y and E_x are the expected values over all the instances of X and Y. Since there are two GANs integrated to generate samples in both directions, i.e., $X \rightarrow Y$ and $Y \rightarrow X$, the adversarial loss is also calculated for the generator, F (for $Y \rightarrow X$ translation), as shown in Equation (6):

$$\mathcal{L}_{GAN}(F,\ D_X, X, Y) = \mathrm{E}_x[\log(D_X(x))] + \mathrm{E}_y[\log(1 - D_X(F(y)))] \tag{6}$$

Both generators try to minimize their respective objectives against their respective adversaries, i.e., G tries to minimize the objective, $\mathcal{L}_{GAN}(G,\ D_Y, X, Y)$, against D_Y, and F tries to minimize the objective, $\mathcal{L}_{GAN}(F,\ D_X, X, Y)$, against D_X. Mathematically,

$$G^* = argmin_G max_{D_Y}(\mathcal{L}_{GAN}(G,\ D_Y, X, Y)) \tag{7}$$

$$F^* = argmin_F max_{D_X}(\mathcal{L}_{GAN}(F,\ D_X, X, Y)) \tag{8}$$

Theoretically, it is possible to learn the mappings, G and F, such that the generated samples have a distribution that is an identical distribution to that of the original samples. However, with sufficient network capacity, the generated samples are probably a random permutation of images in the target domain, due to the infinitesimally large solution space. Therefore, cycle-consistency loss is used to reduce the solution space such that the generated samples should be cycle-consistent. In other words, the generator, F, should be able to produce the original samples used by the generator, G, and vice versa. In other words:

$$x \rightarrow G(x) \rightarrow F(G(x)) \approx x \tag{9}$$

$$y \rightarrow F(y) \rightarrow G(F(y)) \approx y \tag{10}$$

The authors refer to Equation (7) as "forward" cycle-consistency loss, and to Equation (8) as "backward" cycle-consistency loss. This ensures that the generated sample is not a random permutation. The cycle-consistency loss is defined as follows:

$$\mathcal{L}_{Cycle}(G, F) = \mathrm{E}_x[||F(G(x)) - x||_1] + \mathrm{E}_y[||G(F(y)) - y||_1] \tag{11}$$

The complete loss function can be formulated by combining all four losses, as follows:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G,\ D_Y, X, Y) + \mathcal{L}_{GAN}(F,\ D_X, X, Y) + \mu\mathcal{L}_{Cycle}(G, F) \quad (12)$$

where μ controls the significance of the cycle-consistency loss over the adversarial losses. Essentially, it is a minimax objective optimization problem, where generators aim to minimize the distance between the probability distributions, whereas discriminators aim to maximize it. Formally, the minimax objective optimization problem can be defined as follows:

$$G^*, F^* = argmin_{G,F} max_{D_X, D_Y}(\mathcal{L}(G, F, D_X, D_Y)) \quad (13)$$

The generators, G and F, try to minimize the objective function, and the discriminators, D_X and D_Y, try to maximize it.

The proposed method is different from the traditional GANs in that traditional GANs take the input noise, z, to learn the distribution of $p_{data}(x)$ by iteratively updating the distribution of the generator, p_G. In the proposed methodology, the input, x, belongs to Domain X, where the goal of Generator G is to learn the data distribution, $p_{data}(y)$, and the goal for Generator F is to learn $p_{data}(x)$, given input y. It is important to note here that the mapping is not to find a relation between samples; rather, its goal is to find a relation between the two domains. The overall architecture of the synthetic disguised face generation method is shown in Figure 5. The property of cycle-consistency loss allows for performing image-to-image translation without needing paired input–output training samples to generate images, as shown in Figure 6.

Convergence Analysis

Here, we provide proof of convergence for one direction, $G : X \rightarrow Y$, which simplifies the problem to a single generator, G, and a discriminator, D, to demonstrate the optimality of the proposed method. To learn Generator G's distribution, p_G, over data Y, we define a prior distribution on the input data, $p_{data}(x)$, then represent a mapping to the data space as $G(X; \theta_G)$, where G is the learnable mapping function, with parameters, θ_G. Secondly, another function, $D(Y; \theta_D)$, is defined, which outputs a 0 or 1 depending upon its estimation of whether the input, y, came from p_G or $p_{data}(y)$.

The global optimality can be defined as:

$$p_G = p_{data}(y) \quad (14)$$

where p_G is the distribution of Generator G, and $p_{data}(y)$ is the data distribution of Domain Y.

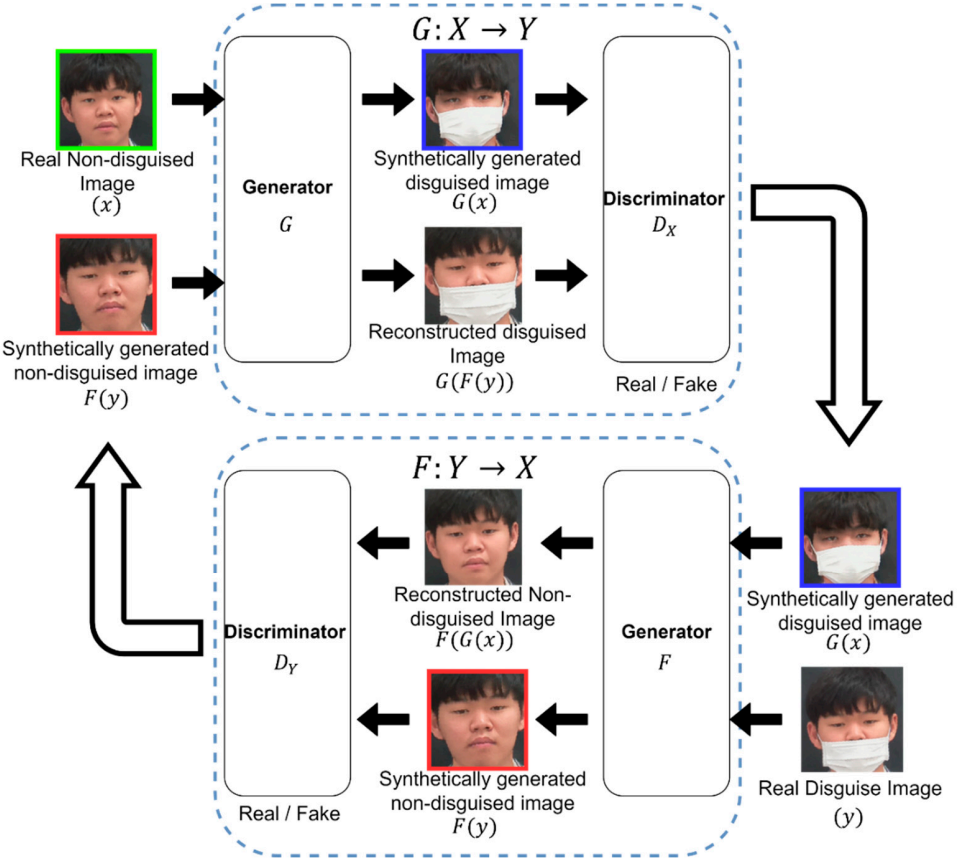

Figure 5. An overview of the proposed framework for disguised face synthesis. Two GANs are trained simultaneously, in forward and backward configurations. Here the generator, G, and the discriminator, D_X, are responsible for translating real facial images into disguised facial images. Generator F and Discriminator D_Y are working in the opposite direction.

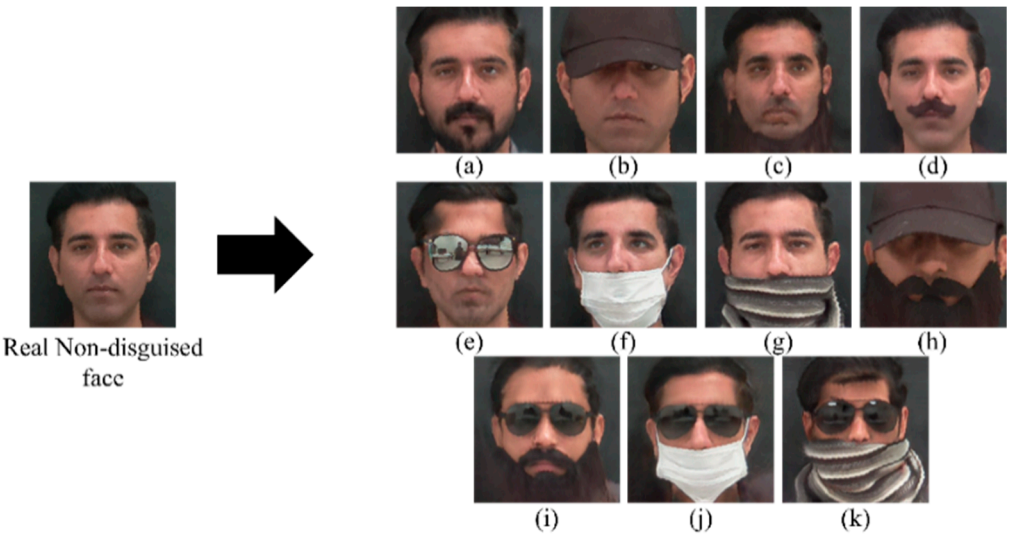

Figure 6. Sample generation results of synthetic disguised face add-ons. All images are generated from a real nondisguised facial image to: (**a**) beard; (**b**) cap; (**c**) fake beard; (**d**) fake mustache; (**e**) glasses; (**f**) mask; (**g**) scarf; and some combination add-ons, such as: (**h**) cap and fake beard; (**i**) glasses and fake beard; (**j**) glasses and mask; and (**k**) glasses and scarf.

Proposition 1. *For a given generator, G, the optimal discriminator, D_Y^*, can be defined as follows:*

$$D_Y^*(y) = \frac{p_{data}(y)}{p_{data}(y) + p_G(y)} \tag{15}$$

Proof. The training criterion for the discriminator, D_X, given Generator G, is to maximize the quantity, $\mathcal{L}(G, D_Y, X, Y)$:

$$\begin{aligned} \mathcal{L}(G, D_Y, X, Y) &= \int_y p_{data}(y) \log(D_Y(y)) dy + \int_x p_{data}(x) \log(1 - D_Y(G(x)) dx \\ &= \int_y p_{data}(y) \log(D_Y(y)) + p_F(x) \log(1 - D_Y(y)) dy \end{aligned} \tag{16}$$

Here, we can use the proof from [19] for $G : X \to Y$. For any $(a, b) \in \mathbb{R}^2\{0, 0\}$, the function $y \to a \log y + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator, D_Y, does not need to be defined outside of $Supp(p_{data}(y) \cup Supp(p_G)$, and the same holds for the discriminator, D_X, i.e., $Supp(p_{data}(x) \cup Supp(p_F)$, thus concluding the proof. □

The training objective for D_Y can be interpreted as maximizing the log-likelihood for estimating the conditional probability, $P(S = s|y)$, where S indicates whether y comes from $p_{data}(y)$, i.e., $y = 1$, or from p_G, i.e., $(y = 0)$. The minimax problem in Equation (6) can be reformulated as:

$$\begin{aligned} C(G) &= max_{D_Y} \mathcal{L}(G, D_Y, X, Y) \\ &= \mathrm{E}_{y \sim p_{data}(y)} \left[\log\left(D_Y^*(y)\right)\right] + \mathrm{E}_{x \sim p_{data}(x)} \left[\log\left(1 - D_Y^*(G(x))\right)\right] \\ &= \mathrm{E}_{y \sim p_{data}(y)} \left[\log\left(D_Y^*(y)\right)\right] + \mathrm{E}_{y \sim p_G} \left[\log\left(1 - D_Y^*(y)\right)\right] \\ &= \mathrm{E}_{y \sim p_{data}(y)} \left[\log \frac{p_{data}(y)}{p_{data}(y) + p_G}\right] + \mathrm{E}_{y \sim p_G} \left[\log \frac{p_G(y)}{p_{data}(y) + p_G}\right] \end{aligned} \tag{17}$$

Theorem 1. *The virtual training criterion defined as $C(G)$ in Equation (17) achieves the value of $-\log 4$at global minimum.*

Proof. For the global optimality proven in Proposition 1, i.e., $p_G = p_{data}(y)$, the optimal discriminator is $D_Y^*(y) = \frac{1}{2}$ (Equation (15)). Solving Equation (17) for $D_Y^*(y) = \frac{1}{2}$, we get $C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4$. It can be seen that this is the best possible value of $C(G)$, reached only for $p_G = p_{data}(y)$. Observe that:

$$\mathrm{E}_{y \sim p_{data}(y)}[-\log 2] + \mathrm{E}_{y \sim p_G}[-\log 2] = -\log 4$$

Moreover, by subtracting this expression from $C(G) = \mathcal{L}(G, D_Y, X, Y)$, we obtain:

$$C(G) = -\log 4 + KL\left(p_{data}(y) \parallel \frac{p_{data}(y) + p_G}{2}\right) + KL\left(p_G \parallel \frac{p_{data}(y) + p_G}{2}\right) \tag{18}$$

where KL is the Kullback–Leibler divergence. In the previous expression, the Jensen–Shannon divergence can be observed between the generator's distribution and the data-generating process:

$$C(G) = -\log 4 + 2 \cdot JSD(p_{data}(y) \parallel p_G) \tag{19}$$

Since the Jensen–Shannon divergence between two distributions is zero only when they are equal and always non-negative, it is shown that $C^* = -\log 4$ is the global minimum of $C(G)$, and that the only solution is $p_G = p_{data}(y)$. That is, the generator model perfectly replicates the generating process. □

Proposition 2. *If Generator G and Discriminator D have enough capacity, and the discriminator is allowed to reach its optimum at every training iteration given G, and p_G is updated according to the criterion:*

$$\mathrm{E}_x[\log(D_X(x))] + \mathrm{E}_y[\log(1 - D_X(F(y)))]$$

then p_G converges to $p_{data}(y)$.

Proof. Consider the function, $\mathcal{L}(G, D_Y, X, Y) = Q(p_G, D_Y, X, Y)$, as a function of p_G, as performed in the aforementioned criterion. It is to be noted that $Q(p_G, D_Y, X, Y)$ is convex in p_G. The derivative of the function is included in the subderivatives of a supremum of convex functions at the point where the maximum is attained [19]. In other words, if $f(x) = sup_{\alpha \epsilon A}\, f_\alpha(x)$ and $f_\alpha(x)$ is convex in x for every α, then $\partial f_\beta\ (x) \epsilon\ \partial f$ if $\beta = \arg sup_{\alpha \epsilon A}\, f_\alpha(x)$. This is equivalent to computing a gradient descent update for p_G at the optimal D_Y, given the corresponding generator, G. $sup_{D_Y} U(p_G, D_Y)$ is convex in p_G, with a unique global optimum, as given in Theorem 1. Hence, with sufficiently small gradient updates of p_G, it can converge to $p_{data}(y)$. Hence, the proof is concluded. □

The convergence analysis is provided for $G : X \to Y$. Similarly, the proof of convergence can be derived for the mapping, $F : Y \to X$. Furthermore, the cycle-consistency loss is calculated for the mappings, $G : F(Y) \to Y$ and $F :\ G(X) \to X$, by alternating the generator inputs between real samples, $[X, Y]$, and generations, $[F(Y), G(X)]$. Equation (16) can be written for the complete loss function mentioned in Equation (12) as follows:

$$\begin{aligned}
\mathcal{L}(G, F, D_X, D_Y) \quad &= \textstyle\int_y p_{data}(y) \log(D_Y(y))dy + \int_x p_{data}(x) \log(1 - D_Y(G(x))dx \\
&+ \textstyle\int_x p_{data}(x) \log(D_X(x))dx + \int_y p_{data}(y) \log(1 - D_X(F(y))dy \\
&+ \textstyle\int_y p_{data}(y) \log(D_Y(y))dy + \int_{F(y)} p_{data}(F(y)) \log(1 - D_Y(G(F(y))))d(F(y)) \\
&+ \textstyle\int_x p_{data}(x) \log(D_X(x))dx + \int_{G(x)} p_{data}(G(x) \log(1 - D_X(F(G(x))))d(G(y)) \\
&= \textstyle\int_y p_{data}(y) \log(D_Y(y)) + p_F(y) \log(1 - (D_Y(y))dy \\
&+ \textstyle\int_x p_{data}(x) \log(D_X(x)) + p_G(x) \log(1 - (D_X(x))dx \\
&+ \textstyle\int_{F(y)} p_{data}(F(y)) \log(D_X(F(y))) + p_G(G(x)) \log(1 - (D_X(F(y)))d(F(y)) \\
&+ \textstyle\int_{G(x)} p_{data}(G(x)) \log(D_Y(G(x))) + p_F(F(y)) \log(1 - (D_Y(G(x)))d(G(x))
\end{aligned} \tag{20}$$

Propositions 1 and 2 also hold for Equation (20), as the additional terms for cycle-consistency loss simply use the generated output for the same mapping. Given the infinite capacity for all the generators and discriminators, it can theoretically converge to the optimum.

3.3. Automated Filtering Algorithm

GANs have shown the capability to produce realistic photos; however, in certain circumstances, they fail to generate lifelike images. In this study, it is observed that when the features that are desired to be transferred over a major proportion of the image, the algorithm renders inferior images and, in some cases, fails to preserve the identity of the subject. For instance, notice the "scarf and glasses" add-on; the complete face is hidden under disguise with a small portion of the nose visible, which obfuscates the model, thus resulting in subpar image-to-image translations.

Therefore, we present an automated filtration scheme that allows us to remove low-quality images from the pool of generated images to avoid any degradation during facial recognition model training. An overall pipeline of the algorithm is shown in Figure 7.

The approach is straightforward to implement, as it utilizes common modules used for FR. First, a SqueezeNet-based facial recognition model is trained on a combination of real nondisguised facial images (real normal), and real disguised facial images (real add-ons). This model is then used to test all the synthetically generated disguise images (Gen. add-ons). The trained model has not seen the generated images beforehand; however, the data

distribution is similar between the generated disguise images and the real disguise images because of the cycle-consistency loss. Therefore, it is hypothesized that the images that share the same distribution as the real images will be correctly recognized by the trained FR model. Hence, this FR model can be used to differentiate between good generations and bad generations of the synthesized facial images. In the first iteration, the generated images that are correctly predicted are saved and included in the training data for the second iteration. For the second iteration, the model is trained on a combination of real nondisguised facial images (real normal), real disguised facial images (real add-ons), and synthetic disguise face images (generated add-ons). This trained model is tested on the images that were falsely recognized in the first iteration. The images that are correctly recognized in the current iteration are saved, and the rest of the images are discarded. This way, the generated images can be filtered efficiently without human interference. The process is only repeated once because CNN models are vulnerable to overfitting and, if further iterations are performed, the model may start to learn on subpar generations, causing overfitting and thereby degrading the overall performance of the system.

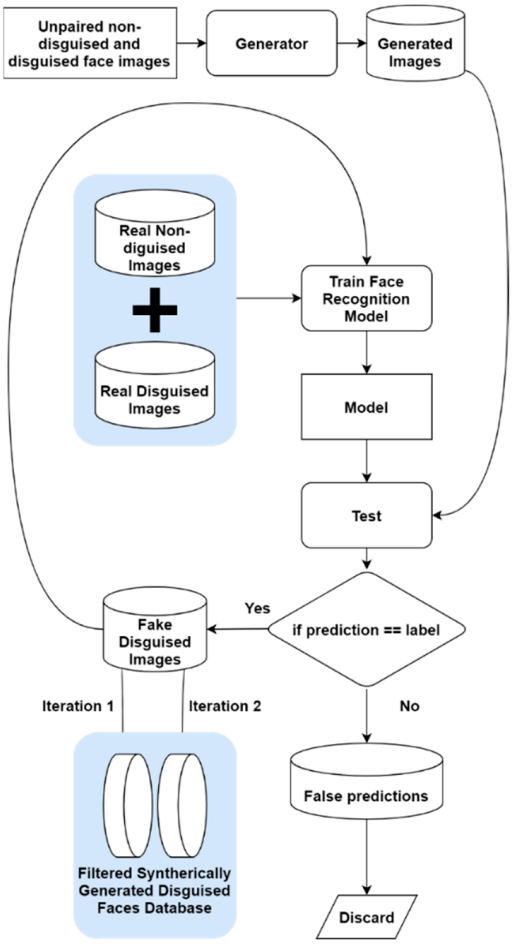

Figure 7. A schematic of the automatic filtering algorithm, which utilizes the concept of incremental learning. First, an FR model is trained on a combination of nondisguised real facial images and disguised real facial images. This model is then tested on the synthetically generated data, where the correctly recognized images are added to the training set. Then, the model is trained on the new training set. This newly trained model is again tested on the remaining synthetic images. The images correctly recognized in both iterations are retained, while the ones that are falsely recognized are discarded.

4. Experimental Work and Results

The details of the proposed method and the database are presented in this section. Additionally, the effects of manually filtering the generated images, and the merits of the automated filtering scheme and the synthetic data, are shown through our FR experimental results.

4.1. Disguised Face Synthesis

A separate image generator is trained for each disguise add-on. To ensure that the model learns only the disguise features, all the images from a single add-on belonging to all of the subjects are used as one domain, and all of the nondisguised images from all of the subjects are used as the other domain. For instance, if it is desired to generate a cap-disguise generator, all the images with cap disguises are used as class A, and all the images without cap disguises are used as class B. Eventually, two generators are trained as a result, i.e., a generator responsible for the $A \rightarrow B$ transformation, and a generator for the $B \rightarrow A$ transformation. However, the results of removing the disguise add-on, i.e., $B \rightarrow A$, are not useful. It is improbable to generate the identity of a person without first training on the same subject because of overfitting and confusion between identities.

To generate images for the presented Synthetic Disguised Face Database, each nondisguised image from the Sejong Face Database was used as an input for the generator, $A \rightarrow B$. This resulted in disguised facial images with the specific add-on that the generator was trained on. Each real image was used to generate 15 disguise images. Sample generator images are shown in Figure 8. It is to be noted that some disguise add-ons, such as fake beards and fake mustaches, are only translated for male subjects. Similarly, makeup and wig add-ons are only used for female subjects.

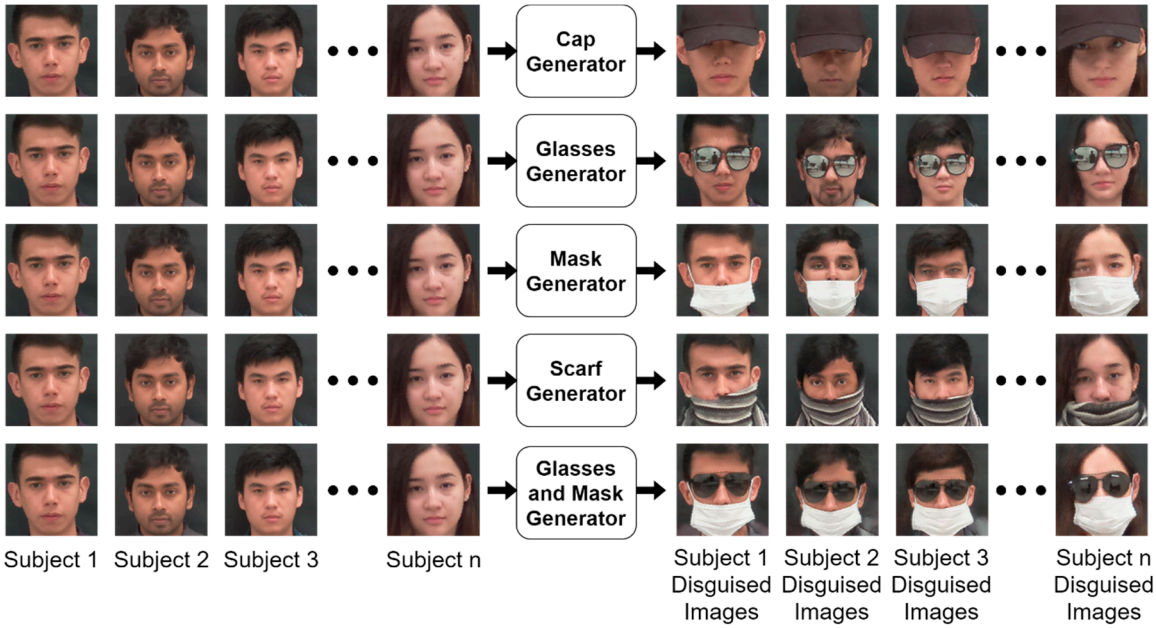

Figure 8. The proposed methodology for disguised facial generation at the inference phase. As shown in the figure, a separate generator is trained for each add-on. Each generator can generate disguised facial images while preserving subject identity.

The experiments for the facial synthesis were performed using PyTorch, an open-source deep learning framework, in a Linux environment. The training was performed using one Nvidia 3080 GPU. A simple resize and crop preprocessing was used, where the training data was first resized to 286 × 286, and was then cropped to 256 × 256 pixels. The images were categorized on the basis of the disguise add-on. One class contained images from all subjects with the specific disguise add-on. The other class contained normal facial

images of all the subjects. The CycleGAN network was trained using two adversarial losses and a cycle-consistency loss, as described in Section 3 The number of filters in the last convolutional layer of both generators was set to 64. For both discriminators, the best results were achieved by setting the number of filters of the first convolutional layer to 64. Because of memory constraints, a single image batch was used for training for 200 epochs because four networks are being trained at a time. The Adam optimizer was used for training all the networks, with a learning rate of 2×10^{-3}. For the first 100 epochs, the learning rate was kept constant, and was then linearly decayed for every 50 epochs. As each epoch took an average of 94 s to complete, the model was trained for one disguise add-on in approximately five and half hours. The Fréchet inception distance (FID) [35] was used for evaluating the quality of the generated images, which was an improvement of the inception score (IS) [36]. The FID calculates the Fréchet distance, with the mean and covariance between the real and the fake image distributions. To further analyze the convergence of the proposed method, the Kernel inception distance (KID) [37] was calculated between the real and synthetically generated images. The KID is the squared maximum mean discrepancy between inception representations and is an improved metric for measuring the GAN convergence and quality.

It is observed that the proposed image generation method learns the within-add-on variations that exist in the base database and applies them while generating new samples. Examples can be seen in Figure 9a, where the synthesized images contain different types of glasses, whereas the original subject is captured with a single type of glasses. Therefore, it is shown that the proposed method not only learns subject-specific variations, but also learns to translate the disguise variations from other subjects as well. In Figure 9b, the images are shown from the category, "fake beard"; here, it is shown that the model generates different types of fake beards for different seed images. Figure 9c is taken from the category of "wig and glasses". Here, it is seen that the seed database images of that subject contain a single type of "wig and glasses", whereas we are able to generate images of the subject with two different types of wigs. Finally, in Figure 9d, the subject is seen to be wearing a golden-colored "fake beard" in the seed database, whereas the proposed method adds two more variations of the "fake beard", while preserving the subject's identity.

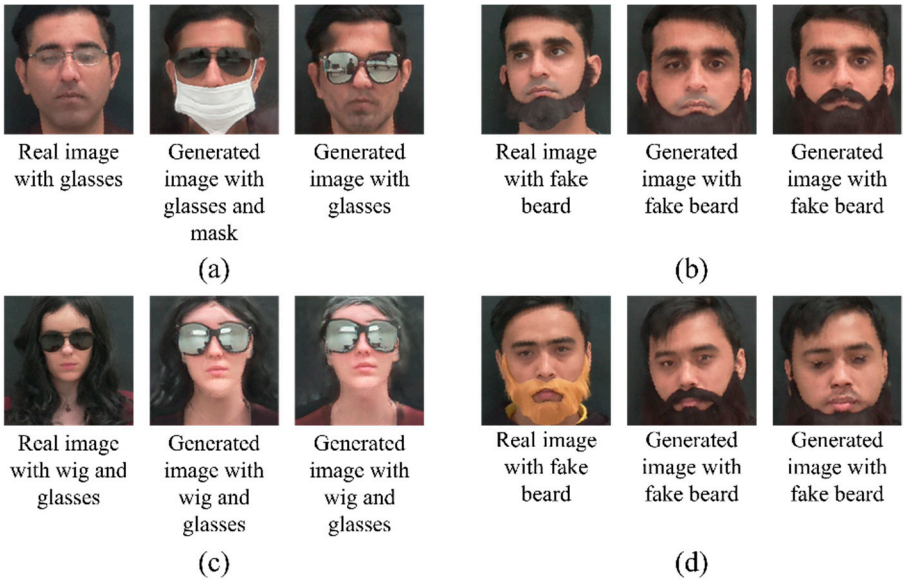

Figure 9. Sample result images from the Synthetic Disguised Face Database. An example of generated samples with variations in (**a**) glasses, (**b**) fake beard, (**c**) wig and glasses, and (**d**) fake beard. Interestingly, the generator learns the variations from other subjects and translates them to different subjects while preserving identity. This means that the proposed method generates variations within a disguise add-on class.

The FID and KID metrics were computed between the real and synthetically generated images, as shown in Table 3. For the sake of comparison, the same metrics were computed between two sets of original data, which served the purpose of a baseline quality measure. It is to be noted that the distribution within the seed database is high, i.e., an FID of 30.967. Therefore, the FID value of 38.177, which was calculated between the generated synthetic database and the seed database, is acceptable. This is also evident from the qualitative results presented in Section 4.3.

Table 3. Quantitative results of the presented study on normal-to-disguised-face generation. The KID means and standard deviations are provided.

Data	FID	KID (Mean ± Standard Deviation)
Real data splits (baseline)	30.967	0.01128 ± 0.00059
Synthetic Images	38.177	0.01684 ± 0.00037

The presented database contains a total of 12,600 images. The seed database contains a total of 75 disguise images per subject, while the proposed database contains 180 disguise images per subject, which is 2.4 times higher than the seed database. Complete details are provided in Table 4. The sample-generated results are shown in Section 4.3. For the sake of comparison, the same subject images are used to present the sample images from the seed database and the proposed Synthetic Disguised Face Database.

Table 4. The types of disguised add-ons available in the SFD [13] and Syn-DFD, along with the total number of images, and information about the gender of subjects. It is to be noted that some add-ons are gender-specific.

Add-On	Add-On Name	Number of Images		Gender	
		Sejong Face Database	Proposed Database	Male	Female
No Add-on	Natural Face	15	-	✔	✔
	Real Beard	10	15	✔	✘
Accessory Add-on	Cap	5	15	✔	✔
	Scarf	5	15	✔	✔
	Glasses	5	15	✔	✔
	Mask	5	15	✔	✔
	Makeup	5	15	✘	✔
Fake Add-on	Wig	10	15	✘	✔
	Fake Beard	5	15	✔	✘
	Fake Mustache	5	15	✔	✘
Combination Add-on	Wig + Glasses	5	15	✘	✔
	Wig + Scarf	5	15	✘	✔
	Cap + Scarf	5	15	✔	✔
	Glasses + Scarf	5	15	✔	✔
	Glasses + Mask	5	15	✔	✔
	Fake Beard + Cap	5	15	✔	✘
	Fake Beard + Glasses	5	15	✔	✘

However, GANs are prone to outputting poor-quality results, as shown in Figure 10. Such poor generations can lead to the degradation of the FR system if they are not removed from the FR training data before the training phase. Such problems arise when there is insufficient information for the generator, as seen in Figure 10. When the subject identity is hidden to a greater extent, it is challenging for the human evaluator to correctly identify subjects. This problem is common in combination add-ons, such as “scarf and glasses”, “scarf and wig”, “mask and glasses”, and “fake beard and cap”.

Figure 10. An example of the poor-quality samples generated. Such sample generation is inevitable because transfiguration is a complex problem because of the varying subject identities and the challenging nature of the disguise add-ons, e.g., "wig and scarf", and "glasses and scarf".

4.2. Manual Filtering

To filter out the bad samples from the pool of generated images, a human evaluation was performed. An interface was designed that shows two images at once. One is the "real normal" image, while the second is the "Gen. add-on" image. The observer is asked to accept or reject the sample on the basis of the following criteria:

- Both images must be recognizable as the same person;
- The image quality must be lifelike;
- There should be no discrepancy between the original and generated samples, such as in the normalcy of the facial features;
- Ensuring the face is not completely hidden by the generated disguise add-on.

The interface used for the manual filtering is shown in Figure 11, where (a) is an example of a scenario where the generated sample accorded with the criteria set for acceptance, and Figure 11b represents a rejected sample, which accorded with the aforementioned criteria.

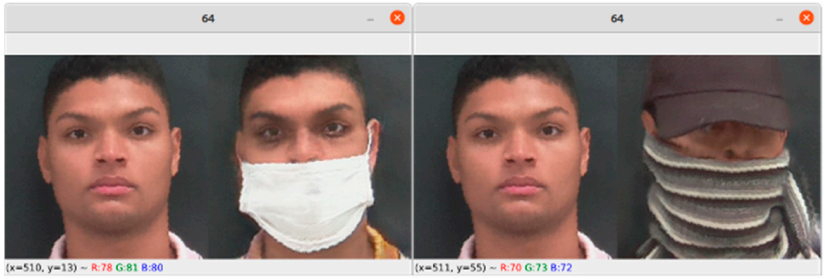

Figure 11. An example of (**a**) an acceptable sample, and (**b**) a rejected sample.

4.3. Automated Filtering Algorithm

As mentioned in Section 3.3, the proposed image-generation method is prone to outputting subpar images because the generator finds a mapping between two domains. Noise, blur, identity transfer, and pixilation are some of the artifacts observed in the synthesized data. Therefore, there is a chance that the model finds a mapping (random permutation) that is translatable between domains, but that does not match the visual context. This can result in rendering an imperceptible face or image, where the identity is not preserved. Such failed generations can significantly degrade the performance of the FR system. To mitigate the effects of bad generations, two filtered subsets of the dataset are also provided. The pseudocode of the filtration process is provided in Algorithm 1.

Algorithm 1: Pseudocode of the proposed automated filtering algorithm.

```
1. Initialization:
2.       Real_Normal[]
3.       Real_Addons[]
4.       Normal_to_Addon_Generator()
5.       filtered_Gen_Addons[]
6.       false_predictions[]
7.    Gen_Addons = Normal_to_Addon_Generator(Real_Normal, Real_Addons)
8.    FR_model = SqueezeNet.train(Real_Normal, Real_Addons)
9.    predictions[] = FR_model.predict(Gen_Addons)
10. if predictions[x] == ground_truth[x]:
11.    filtered_Gen_Addons.append(predictions[x])
12. else:
13.    false_predictions.append(predictions[x])
14.    FR_model_2 = SqueezeNet.train(Real_Normal, Real_Addons+fi ltered_Gen_Addons)
15.    predictions[] = FR_model.predict(false_predictions)
16. if predictions[x] == ground_truth[x]:
17.    filtered_Gen_Addons.append(predictions[x])
18.    Gen_DB.save(filtered_Gen_Addons)
19. else:
20.    false_predictions.append(predictions[x])
21.    false_predictions = Null
```

Figure 12 presents some sample images from the final generated samples after the automated filtering.

The first set was manually filtered, where each image was observed by a human and a decision was made to retain or discard the image. The second filtered set was filtered through the automated process, explained in Section 3.3. Table 5 presents the number of images, before and after filtration.

Table 5. The number of synthetically generated images, before and after applying the automatic filtering algorithm.

Method	Total Number Images	Images/Subjects
No Filtering	12,600	180
Manual Filtering	6780	88
Automatic filtering	4158	60

Figure 12. Image generation results. Fake beard and fake mustache experiments were not conducted for the female subjects.

4.4. Facial Recognition Experiments and Results

The experiments for the face classification were performed using PyTorch, an open-source deep learning framework, in a Linux environment. The training was performed using one Nvidia 3090Ti GPU. The training data, as described in Section 4.4.1, was first resized to a size of 224 × 224 pixels, and the labels contained the subject identity only. SqueezeNet, a recently popular network for FR and other image classification tasks, was used, along with the softmax layer and binary cross-entropy loss. The network was trained using an Adam optimizer [38], with an initial learning rate of $3e^{-4}$, and the learning rate was reduced by a factor of 0.8 when the error plateaued. The network was trained using a mini-batch size of 64 until the loss plateaus. The training process took an average of 4 h for each setting. Data augmentation was performed to mitigate overtraining and to increase the training size such that all configurations ended up being trained at the same number of images. Geometric transformations for rotation, shift in both axes, brightness shift, shear, and horizontal flip were applied to the training data as standard augmentation techniques. Image normalization was performed using feature-wise centering and feature-wise standard normalization, calculated on the entire training dataset.

4.4.1. Experiment Configurations

To validate the utility of the presented synthetic database, FR experiments were performed using five different training configurations. The configurations were categorized on the basis of the type of data used for training, as shown in Table 6. Configuration 0 in Table 6 served as a baseline, where the network was trained on "real normal" images, which have up to 15° variations in the pose. This model mimics the scenario where the training data only contains images taken in a controlled real-world scenario, such as for identification documents. Configuration 1 is the network trained on a combination of "real normal" plus "real add-on" images. This configuration shows the baseline results for the scenario, where the training data contains add-on facial images. Realistically, capturing subject faces with a large variety of add-ons is unfeasible, unless the data is acquired for specific purposes, such as collecting a specialized database. Configuration 2 utilized "real normal" images and "Gen. add-on" facial images from the proposed database. The model trained on this data shows the utility of the presented synthetic database and the image-generation method. It is hypothesized that this model has a better chance of outperforming Configuration 0, as this model was trained on synthetically generated disguised images. Configuration 3 and Configuration 4 serve the purpose of comparison, so that further analysis can be drawn to understand the synthetic data and how it can be improved. In Configuration 3, the model was trained on a combination of all the available training data, i.e., real normal, real add-on, and Gen. add-ons. This will help to realize the importance of synthetic data in circumstances where real disguised face images are also available. The results of this configuration helped test the hypothesis that asserts that, even when real disguised face images are available, the proposed methodology can improve the performance of an FR system. Configuration 4 is the model trained on the proposed synthetic data only; it shows the effectiveness of the proposed methodology for data generation and the images as standalone resources for training FR algorithms. Configuration 5 uses the "real normal" facial images from the original SFD dataset and manually filtered (MF) "Gen. Add-on" images from the proposed dataset. Configuration 6 uses the "real normal" facial images from the original SFD dataset and autofiltered (AF) "Gen. Add-on" images from the proposed dataset.

4.4.2. Facial Recognition Results

Using the proposed dataset shown in this section, the FR results prove the efficacy of the proposed synthetic face database. The models trained using different configurations were tested on two test sets, i.e., "real normal" and "real add-on". The information regarding the test sets is shown in Table 7. It is to be noted that, for the training of Configuration 0, the testing was performed on a limited set of images because a larger portion of "real normal" images were used as the training data.

The results for different training configurations, tested on normal faces and disguised faces, are presented in Table 8. Configuration 0 serves as an example of a system trained only on nondisguised faces. When tested with similar facial images, the system performs well, achieving an overall FR accuracy of 99.6%. However, when presented with disguised facial images, the system's performance drops down to a 26% recognition rate, as shown in Table 8. This shows that an FR system is not robust against the unseen challenges of FR. Configuration 1, a system trained on "real normal" plus "real add-on" images, shows a smaller gap between its FR performance on the normal and add-on facial images: 69% and 88%, respectively. Configuration 1 achieves a 30% lower accuracy than Configuration 0; this could be because the training data for Configuration 1 contains only four normal facial images per subject, which are not sufficient for the network to learn the identity-related features of a nondisguised face. Configuration 2 is the network trained on the "real normal" images, available in the original SFD database, and the "Gen. add-on" images from the proposed synthetic face database. This configuration shows the efficacy of the proposed method for synthetic disguised face generation. Compared with Configuration 1, training on synthetic data improves the FR accuracy of normal facial images, from 69% to 86%.

This shows that the presented data increases the amount of training data, and also that the synthetic images are able to retain the identity features. Similarly, Configuration 2 achieves 72% recognition accuracy for "real add-on" images, compared with the 36% accuracy of Configuration 0. This shows that the presented synthetic data are an effective methodology for improving the robustness of FR algorithms against the challenges of disguise.

The filtration process removes the noisy and nonclassified synthetic images from the generations, as described in Sections 3.2 and 3.3. An improved recognition rate on the real disguised facial images is expected with the improvement in the quality of the training data. The hypothesis is validated by the increased recognition rate of Configuration 5, where manually filtered training data achieves a higher recognition rate: 77.8% over Configuration 2.

The automatically filtered data was used for training Configuration 6. Configuration 6 achieves the highest recognition rates among all the training experiments. It can be concluded that, as the automation is performed using a CNN, the images that create issues, or that have nonidentity features, are filtered out during the filtration process. As a result, ideal training data is retained, which makes Configuration 6 surpass even the performance of Configuration 1, where real disguised facial images are used for training.

Table 6. This table presents the configurations of the facial recognition experiments conducted to demonstrate the efficacy of the synthetically generated disguised faces. Different data configurations were used in experiments for the sake of comparison and were tested on the set of real add-ons.

Training Configuration	Training Data Type	Total Images	Images Per Subject	Subjects	Disguise Add-Ons
Configuration 0	Real normal	685	10	All	All
Configuration 1	Real normal + Real Add-on	986	15 (4 + 11)		
Configuration 2	Real normal + Gen. Add-on	1240	19 (4 + 15)		
Configuration 3	Real normal + Real Add-on + Gen. Add-on	1276	20 (4 + 8 + 8)		
Configuration 4	Gen. Add-on	1486	23		
Configuration 5	Real normal + Gen. Add-on (Manual Filtering)	1240	19 (4 + 15)		
Configuration 6	Real normal + Gen. Add-on (Automatic Filtering)	1240	19 (4 + 15)		

Table 7. The test data used for evaluation of facial recognition models trained with the presented training configurations.

Test Set	Data Description	Number of Total Images	Number of Images Per Subject
Real Normal *	Photographed images of nondisguised faces (reduced set used for Configuration 0)	251	4
Real Normal	Photographed images of nondisguised faces	564	9
Real Add-on	Photographed images of disguised faces	2272	36

* Real Normal used for testing Configuration 0 has less test images.

Table 8. This table presents the facial recognition results achieved on the real nondisguised (real normal) and the disguised (real add-on) faces by the models trained using the training configurations mentioned in Table 6.

Training Configuration	Training Data Type	Accuracy (%)	
		Real Normal	Real Add-On
Configuration 0	Real normal	99.6 *	26
Configuration 1	Real normal + Real Add-ons	69	88
Configuration 2	Real normal + Gen. Add-ons	86	72
Configuration 3	Real normal + Real Add-ons + Gen. Add-ons	62	89
Configuration 4	Gen. Add-ons	74	71
Configuration 5	Real normal + Gen. Add-ons w/Manual Filtering	-	77.8
Configuration 6	Real normal + Gen. Add-ons w/Automatic Filtering	-	94.3

* Real Normal used for testing Configuration 0 has less test images, as shown in Table 7.

Figure 13a presents the results from three models having the same architecture, but that were trained on different datasets. The model trained on the "real normal" set (Configuration 0) performs well for nondisguised images, whereas the accuracy drops significantly when the same model is tested on "disguised" images. The model trained on a combination of "real normal" and "real add-on" (Configuration 1) performs worse than the first model; however, it achieves an improved accuracy as compared to the model trained on "real normal" data only. This result provides evidence that the use of disguised facial images during the training phase in order to achieve satisfactory results is beneficial. On the other hand, the model trained on a combination of "real normal" and synthetic data (Configuration 2), generated by using the proposed scheme, achieved the best recognition rate for the test set of the "disguised" images.

Figure 13b presents an ablation analysis of the proposed generation methodology. The accuracy of the three models, having the same architecture but different training data, is compared on "real disguised" facial images. First, accuracy is reported where the model is trained on all the generated samples (Configuration 2), including bad generations for the sake of comparison. The model achieved an approximately 72% rank-1 recognition rate. The next model was trained on a set of generated samples that were manually cleaned (Configuration 5) by following a set of rules, as described in Section 4.3. This model achieved a 6.8% improvement in the recognition rate, as compared to the model trained on raw generated data. Finally, the model was trained on a set of autofiltered images (Configuration 6). This model achieved the highest accuracy, 94.3%, for the disguised facial images.

Figure 13c presents a comparison that was performed to signify the importance of the generated disguised images: two models having the same architecture trained on two different data. First, the model was trained on a combination of "real normal" and "Gen. add-on" images (Configuration 2). Second, the model was trained on only the "Gen. add-on" images (Configuration 4). It can be observed that the models achieved comparable accuracy. The similar inference accuracy of these models shows that the inclusion of "real normal" facial images does not significantly impact the performance of the trained models on disguised faces.

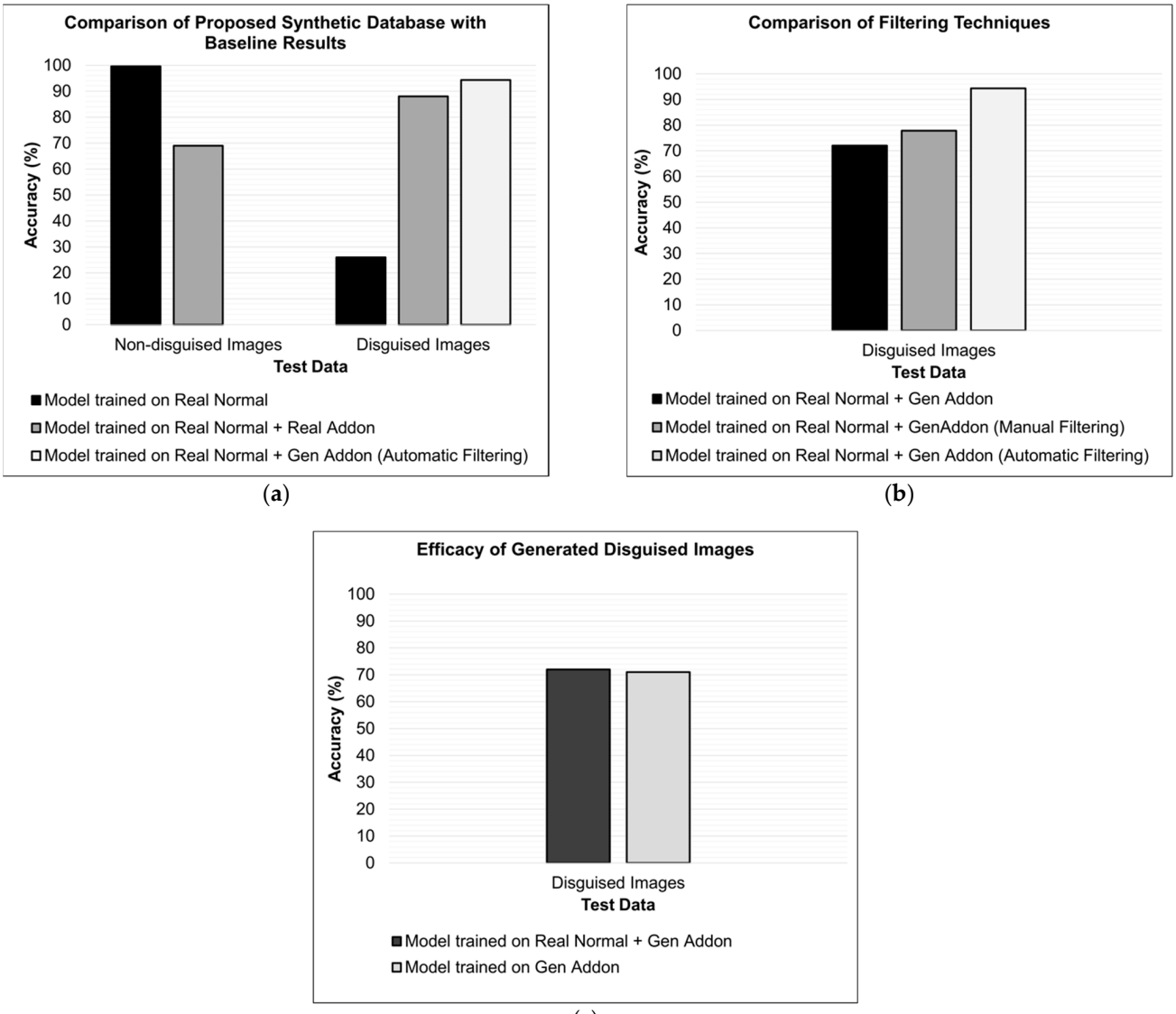

Figure 13. (**a**) Facial recognition results of three models trained on different training data on "real normal" and "real disguised" facial images. (**b**) A comparative analysis was performed to realize the importance of generating synthetic images and the proposed filtration scheme when tested on disguised facial images. (**c**) The efficacy of generated images compared with nondisguised images, and how they can be utilized for improved facial recognition, results in specialized scenarios.

5. Discussion

The qualitative analysis of the proposed framework for disguised facial synthesis shows the usefulness of the method to the synthesis of disguised faces where the original data is not available. Synthetic images can also be useful for human facial recognition. Our quantitative FR experiments show that the usage of the proposed synthetic data improves the robustness of FR models for disguised face scenarios, proving the merits of the proposed method and database.

The proposed facial synthesis model preserves identity in the majority of the scenarios; however, when the majority of facial features are hidden by disguise, it becomes a challenging problem. Well-suited examples for this scenario are combination add-ons, such as "scarf and cap", which completely hide the subject's face. In such a scenario, the model fails to learn the identity features and generates samples with the identities of source

domain subjects. However, integration with the proposed automated filtration scheme can eliminate such cases.

The filtering of the synthetic data was performed to remove the low-quality samples. Consequently, the FR experiments show that the filtered training data improves the quality of the model and that improved recognition rates are achieved using the manual- and automatic-filtered synthesized data. Manual filtering is subjective and is prone to human error compared to automatic filtering. Two major factors yield a lower performance when manual filtering is applied:

1. Human error

During the manual filtration method, described in Section 4.2., a human evaluator accepts or rejects on the basis of defined criteria. However, human bias can result in inconsistencies in the filtered data. For instance, the operator is prone to reject a bad, but acceptable generation after viewing 10 to 15 good samples. Similarly, in the opposite scenario, the operator is prone to accept a bad generation after having observed worse samples immediately before. After viewing multiple bad samples, even a slight improvement in the generated image has an increased chance of being accepted because of the bias set by the previous samples. However, the automatic filtration system is not affected by such biases.

2. The way a model perceives an image is very different than humans

The automatic filtering algorithm is based on the FR algorithm, which is trained on "real normal" and "disguised" facial images. Therefore, the images filtered by the automatic algorithm are more likely to be appropriate for the further training of FR models in the next step, as compared to the ones filtered by the human evaluator. Essentially, the perceptions of humans and machines rely on different components of an image.

Furthermore, the computational complexity of the proposed methods is analyzed by calculating the theoretical number of multiply-add operations in the four models that are part of the CycleGAN method, whereas the model sizes are presented on the basis of the number of trainable parameters in the model. A unit of Mac states one multiplication operation and one addition operation, also knowns as the multiply-accumulation operation. The computational complexity and the number of parameters are presented in Table 9.

However, during inference time, our method only utilized a single generator, i.e., Generator G, and, therefore, the inference computational complexity was significantly lower, which made it suitable for runtime data augmentation as well.

Table 9. Computational complexity and model sizes of the proposed methodology are presented in the form of the number of multiple-add operations and the number of trainable parameters. It is to be noted that, during the inference phase, only one generator is used; therefore, the computational complexity during inference time is only based on Generator *G*.

Models	Computational Complexity (GMac)	Number of Parameters (Million)	Training	Inference
Generator G	56.89	11.38	✔	✔
Generator F	56.89	11.38	✔	✘
Discriminator D_X	3.15	2.76	✔	✘
Discriminator D_Y	3.15	2.76	✔	✘
Total (Training)	120.08	28.28	-	-
Total (Inference)	56.89	11.38	-	-

6. Conclusions

This work presents a GAN-based methodology for unsupervised disguised facial synthesis. A disguised facial image database is presented using the proposed algorithm. Additionally, an automated data filtration scheme is proposed for valid image selection. The proposed methodology synthesizes disguised facial images using nondisguised facial

images, while preserving the identity and the representative features in the synthesized image. The methodology is used to present an extended synthesized face database for the significant increase in the number of disguised facial images. The automated filtration scheme removes the poor GAN generations, improving the synthesized data quality for the training of FR algorithms. The merits of the proposed algorithm and database are shown through the results of our FR experiments.

The FR experiments on the synthesized data show that the proposed method is effective at transferring facial disguises while preserving the identity information of the subject. Additionally, comparative results with networks trained on nondisguised facial images prove the utility of the synthesized disguised data for making FR algorithms more robust to spoofing and disguise attacks. The experiments on the autofiltered data improve the FR rates on "real disguised" images, further reinforcing the merits of the proposed work. The proposed method can be employed to synthesize the disguise variations of facial images where only nondisguised facial images are available. Additionally, the synthesized disguised face database provides a larger number of training images, which can be used to train more robust FR algorithms for practical applications. In the future, we plan to extend the proposed method's additional conditional attributes to generate disguised add-ons with desired variations, such as color, size, texture, etc. The proposed method can also be optimized for other imaging modalities to synthesize, for example, infrared thermal images.

Author Contributions: Conceptualization, M.A. (Mobeen Ahmad) and U.C.; coding, M.A. (Mobeen Ahmad), U.C., and M.A. (Muhammad Abdullah); supervision, D.H. and S.M.; writing-original draft, M.A. (Mobeen Ahmad) and U.C.; review and editing, S.M. and D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the "Cooperative Research Program for Agriculture Science and Technology Development (Project No. PJ015686)" Rural Development Administration, Republic of Korea, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2021R1F1A106168711).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The Sejong Face Database used in the study is available at github.com/usmancheema89/SejongFaceDatabase (accessed on 18 November 2021). The presented synthetic database and image generation code will be made available at github.com/ahmadmobeen/SyntheticDisguiseFaceDatabase (accessed on 18 November 2021) and github.com/ahmadmobeen/FaceSynthesis (accessed on 18 November 2021), respectively.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Singh, R.; Vatsa, M.; Bhatt, H.S.; Bharadwaj, S.; Noore, A.; Nooreyezdan, S.S. Plastic Surgery: A New Dimension to Face Recognition. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 441–448. [CrossRef]
2. Phillips, P.J.; Flynn, P.J.; Bowyer, K.W.; Bruegge, R.W.V.; Grother, P.J.; Quinn, G.W.; Pruitt, M. Distinguishing Identical Twins by Face Recognition. In Proceedings of the 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), Santa Barbara, CA, USA, 21–25 March 2011; pp. 185–192.
3. Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Jacques, S. Multi-block Color-binarized Statistical Images for Single-sam-Ple Face Recognition. *Sensors* **2021**, *21*, 728. [CrossRef] [PubMed]
4. Bhatt, H.S.; Bharadwaj, S.; Singh, R.; Vatsa, M. On Matching Sketches with Digital Face Images. In Proceedings of the IEEE 4th International Conference on Biometrics: Theory, Applications and Systems (BTAS), Washington, DC, USA, 27–29 September 2010.
5. Klare, B.; Li, Z.; Jain, A.K. Matching Forensic Sketches to Mug Shot Photos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 639–646. [CrossRef] [PubMed]
6. Chen, X.; Flynn, P.J.; Bowyer, K.W. IR and Visible Light Face Recognition. *Comput. Vis. Image Underst.* **2005**, *99*, 332–358. [CrossRef]

7. Klare, B.; Jain, A.K. HeTerogeneous Face Recognition: Matching NIR to Visible Light Images. In Proceedings of the International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010.
8. Singh, R.; Vatsa, M.; Noore, A. Hierarchical Fusion of Multi-Spectral Face Images for Improved Recognition Performance. *Inf. Fusion* **2008**, *9*, 200–210. [CrossRef]
9. Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Taleb-Ahmed, A. Past, Present, and Future of Face Recognition: A Review. *Electronics* **2020**, *9*, 1188. [CrossRef]
10. Taskiran, M.; Kahraman, N.; Erdem, C.E. Face Recognition: Past, Present and Future (a Review). *Digit. Signal Process.* **2020**, *106*, 102809. [CrossRef]
11. Ramanathan, N.; Chellappa, R.; Roy Chowdhury, A.K. Facial Similarity across Age, Disguise, Illumination and Pose. In Proceedings of the International Conference on Image Processing (ICIP), Singapore, 24–27 October 2004; Volume 3.
12. Singh, R.; Vatsa, M.; Noore, A. Face Recognition with Disguise and Single Gallery Images. *Image Vis. Comput.* **2009**, *27*, 245–257. [CrossRef]
13. Cheema, U.; Moon, S. Sejong Face Database: A Multi-Modal Disguise Face Database. *Comput. Vis. Image Underst.* **2021**, *208–209*, 103218. [CrossRef]
14. Noyes, E.; Parde, C.J.; Colón, Y.I.; Hill, M.Q.; Castillo, C.D.; Jenkins, R.; O'Toole, A.J. Seeing through Disguise: Getting to Know You with a Deep Convolutional Neural Network. *Cognition* **2021**, *211*, 104611. [CrossRef] [PubMed]
15. Dhamecha, T.I.; Nigam, A.; Singh, R.; Vatsa, M. Disguise Detection and Face Recognition in Visible and Thermal Spectrums. In Proceedings of the 2013 International Conference on Biometrics (ICB), Madrid, Spain, 4–7 June 2013.
16. Min, R.; Kose, N.; Dugelay, J.L. KinectfaceDB: A Kinect Database for Face Recognition. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *44*, 1534–1548. [CrossRef]
17. Dhamecha, T.I.; Singh, R.; Vatsa, M.; Kumar, A. Recognizing Disguised Faces: Human and Machine Evaluation. *PLoS ONE* **2014**, *9*, e99212. [CrossRef] [PubMed]
18. Zhang, S.; Liu, A.; Wan, J.; Liang, Y.; Guo, G.; Escalera, S.; Escalante, H.J.; Li, S.Z. CASIA-SURF: A Large-Scale Multi-Modal Benchmark for Face Anti-Spoofing. *IEEE Trans. Biom. Behav. Identity Sci.* **2020**, *2*, 182–193. [CrossRef]
19. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
20. Khaldi, Y.; Benzaoui, A. Region of Interest Synthesis Using Image-to-Image Translation for Ear Recognition. In Proceedings of the 2020 4th International Conference on Advanced Aspects of Software Engineering (ICAASE), Constantine, Algeria, 28–30 November 2020.
21. Khan, A.; Jin, W.; Haider, A.; Rahman, M.; Wang, D. Adversarial Gaussian Denoiser for Multiple-Level Image Denoising. *Sensors* **2021**, *21*, 2998. [CrossRef] [PubMed]
22. Khaldi, Y.; Benzaoui, A. A New Framework for Grayscale Ear Images Recognition Using Generative Adversarial Networks under Unconstrained Conditions. *Evol. Syst.* **2020**, *12*, 923–934. [CrossRef]
23. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2242–2251.
24. Steiner, H.; Kolb, A.; Jung, N. Reliable Face Anti-Spoofing Using Multispectral SWIR Imaging. In Proceedings of the 2016 International Conference on Biometrics (ICB), Halmstad, Sweden, 13–16 June 2016. [CrossRef]
25. Steiner, H.; Sporrer, S.; Kolb, A.; Jung, N. Design of an Active Multispectral SWIR Camera System for Skin Detection and Face Verification. *J. Sens.* **2016**, *2016*, 9682453. [CrossRef]
26. Raghavendra, R.; Vetrekar, N.; Raja, K.B.; Gad, R.S.; Busch, C. Detecting Disguise Attacks on Multi-Spectral Face Recognition Through Spectral Signatures. In Proceedings of the International Conference on Pattern Recognition, Beijing, China, 20–24 August 2018; pp. 3371–3377. [CrossRef]
27. Weyrauch, B.; Heisele, B.; Huang, J.; Blanz, V. Component-Based Face Recognition with 3D Morphable Models. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Washington, DC, USA, 27 June–2 July 2004. [CrossRef]
28. Paysan, P.; Knothe, R.; Amberg, B.; Romdhani, S.; Vetter, T. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Genova, Italy, 2–4 September 2009; pp. 296–301. [CrossRef]
29. Dantcheva, A.; Chen, C.; Ross, A. Can Facial Cosmetics Affect the Matching Accuracy of Face Recognition Systems? In Proceedings of the 2012 IEEE 5th International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, USA, 23–27 September 2012; pp. 391–398. [CrossRef]
30. Phillips, P.J.; Flynn, P.J.; Scruggs, T.; Bowyer, K.W.; Chang, J.; Hoffman, K.; Marques, J.; Min, J.; Worek, W. Overview of the Face Recognition Grand Challenge. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; pp. 947–954. [CrossRef]
31. Afifi, M.; Abdelhamed, A. AFIF4: Deep Gender Classification Based on AdaBoost-Based Fusion of Isolated Facial Features and Foggy Faces. *J. Vis. Commun. Image Represent.* **2019**, *62*, 77–86. [CrossRef]
32. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

33. Ahmad, M.; Abdullah, M.; Moon, H.; Yoo, S.J.; Han, D. Image Classification Based on Automatic Neural Architecture Search Using Binary Crow Search Algorithm. *IEEE Access* **2020**, *8*, 189891–189912. [CrossRef]
34. Subramanya, A.; Pillai, V.; Pirsiavash, H. Fooling Network Interpretation in Image Classification. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.
35. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
36. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
37. Binkowski, M.; Sutherland, D.J.; Arbel, M.; Gretton, A. Demystifying MMD GANs. In Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
38. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.