Training Facial Recognition with Synthetic Faces

Patrick Vine, *MSc Electrical Engineering,University of Twente*

Abstract—The effective generation of synthetic faces may be useful for improving facial recognition training datasets. This work explores methods for generating synthetic faces and trained a generative network to synthesize front facing facial images of existing identities with different attributes as well as of completely new identities. The identities of the synthetic faces were evaluated using 3 pretrained facial recognition systems. Facial recognition networks were trained to compare the performance of training with the synthetic faces and real faces. The ability to use the synthetic faces for data augmentation was also evaluated. It was found that the mean equal error rate (EER) increased from 2.21% when using the real facial images to 5.27% when training with completely synthetic faces of new identities. When using the synthetic faces for data augmentation, the new identities could improve the mean EER. However, this improvement is not guaranteed with some training datasets leading to higher mean EER after training with more synthetic faces. There is clearly still a difference between the synthetic faces generated and real faces. Understanding what is still missing in the synthesized faces would be valuable research to more effectively enable training facial recognition with only synthetic faces.

Index Terms—synthetic faces; generative networks; data augmentation; face recognition;

1 Introduction

The current best performing facial recognition systems have been trained with a large amount of data. This leaves the best systems in the hands of those with the data, mostly large companies with an existing, large collection of facial images. Researchers have access to non-commercial, reasonably sized datasets such as VGGFace (around 2.6M images, 2.6K identities [26]) and Casia Webface (around 500K images, 10k identities [37]). In comparison, FaceNet was trained by Google on 100-200M images of around 8M identities [29]. Acquiring more data for researchers to compete with large private datasets, or new data for a commercial organisation that does not yet have any facial images, is time consuming and privacy issues may arise. Generating synthetic faces of new identities and more diverse facial images of existing identities may help to resolve this.

New facial images that are more diverse versions of existing identities in a dataset such as with new poses, new hairstyles, illumination or wearing glasses would be beneficial for increasing the diversity of identity images to learn from. New identities with this diversity may help with learning better discrimination between different identities. Synthesized faces may be suitable for data augmentation to help reduce bias in datasets and improve performance on certain attributes. Using an entirely synthetic dataset of new identities reduces potential privacy issues as the facial images are all of nonexisting people. It is desirable to be able to generate an entirely synthetic dataset of unique facial images in which the privacy and diversity can be controlled in order to train a facial recognition system to the level of the best networks.

1.1 Research Question

An initial research question for these ideas is: How well can you train facial recognition with synthetic faces?

To answer this research question the following sub-questions are investigated:

1) What are the ways of generating synthetic faces?

Fig. 1. Examples of synthetic faces. Left: A new facial image for an existing identity. Right: A completely new identity

2) How do the properties of the synthetic faces compare to properties of the original faces as seen by other facial recognition systems?
3) Does training with completely synthetic faces result in similar performance to training with the original non-synthetic faces?
4) Does training with the original non-synthetic faces supplemented with synthetic faces improve the performance of a facial recognition network?

In section 2, Related Work is discussed in order to address the first research sub-question. In section 3, the method used to train a face synthesis network is presented along with methods for analysing the properties of any synthesized face datasets. The method we used to train the facial recognition networks is also presented in section 3. In section 4, experiments and results are presented to answer the remaining 3 research sub-questions. Finally, in section 5, conclusions are drawn as to how well you can train facial recognition with the synthetic faces generated, and future research is proposed.

2 Related Work

In this section an overview of some of the ways of generating synthetic faces are presented. This is followed by an overview of some of the current work in deep facial recognition and the use of synthetic faces for training facial recognition networks.

2.1 Synthetic Faces

The idea of generating facial images to augment facial recognition datasets is not new. There are several surveys that focus on data augmentation specifically for facial recognition.

Li et al. [19] reviewed facial generation methods with a focus on mathematical models for creating new face samples. They looked at methods that used facial structure, the sample distribution, and differing viewpoints in order to generate new faces. They reviewed methods for using knowledge about facial structure to generate new images using the symmetry of the face, mirroring the face and other techniques. Sample distribution methods included combining perturbed data with the distribution function of the data to generate new samples. Viewpoint methods focused on 3D models that could produce different poses and illumination.

Wang et al. [36] broke techniques for data augmentation for facial recognition down into different transformation types and methods. They listed transformation types such as geometric, photometric, hairstyle, makeup, accessories, pose, expression, age, gender, skin colour. The transformation methods included basic image processing, model-based, and generative-based transformations. Model-based methods included 2D Active Appearance Models and 3D Morphable Models (3DMM). The generative-based transformations were further broken down into autoregressive generative models, variational autoencoders (VAEs), generative adversarial networks (GANs) and flow based generative models.

In our work, we have focused on using a generative technique to generate new versions of an existing face image as well as to generate new face images for new identities.

2.1.1 Techniques for modifying an existing face image

Masi et al. [24] applied pose, shape and expression transformations to CASIA WebFace to augment the number of per-identity images in the dataset. They used a 3D pose estimation technique to project the 2D image onto a 3D generic face and generated new poses from the 3D model. They varied the face shape by using 10 different generic 3D face shapes which introduced subtle facial changes in the 2D rendered images. They also manipulated facial expressions in 3D and rendered 2D images with differing expressions.

Lv et al. [23] combined landmark perturbation with synthesizing hairstyle, glasses, poses and varying illumination to augment the multi-PIE dataset. The landmark perturbation aimed to apply an affine transformation to the face to noisily align the landmarks and hence to learn recognition in the face of misaligned landmarks. Hairstyle and glasses synthesis was achieved through the use of templates and 2D image processing techniques that used the facial landmarks for alignment. Pose and illumination were achieved with a 3D face reconstruction to which different poses and light sources were applied and rendered.

Upchurch et al. [35] showed that many high-level semantic image transformations can be done via simple linear transforms in the deep feature space of a pretrained deep CNN such as VGG-19. A set of images with an attribute (e.g. wearing glasses) and a set of images with the opposite attribute (e.g. without glasses) were selected. Linear calculations were made in the deep feature space with the means of these two sets to find a vector to move along in the deep feature space. Images were generated by reversing the function transformation into the deep feature space back to the pixel space. Attributes like age, facial hair and glasses were successfully varied.

In recent years a focus has been on generative models to modify existing facial images. Generative networks have been shown to be able to apply makeup (e.g. BeautyGAN, Li et al. [21]), transform hair colour and age (e.g. StarGAN, Choi et al. [4]), add and remove accessories such as glasses and many more.

In this work we used a generative model to combine an identity image as the source of the identity to maintain in the synthesized face and an attribute image as the source of the attribute change to be seen to the identity's image.

2.1.2 Techniques for maintaining identity.

A key issue for generating faces for training facial recognition is identity. When modifying the attributes of a facial image, the identity needs to be maintained so that it is the same person. It is also useful to generate a new identity and then modify the attributes of that identity. A problem with generative networks is that if they are not trained to maintain the identity in the image, they may not maintain it.

Li et al. [20] sought to change an input facial image through controllable attributes by optimising a CNN combined with the input attributes. They combined an identity loss function with an attribute loss function to learn a transformation. The identity loss function aimed to keep the identity the same. The attribute loss function compared the output with the subset of all images with that attribute in the training set.

Shen et al. [31] proposed FaceID-GAN, a 3-player GAN with a classifier of face identity as the third player. They used a 3DMM representation as an additional input into the generator to supervise the learning. When trained on Casia WebFace, the generator generated faces of the same identity with diverse viewpoints and expressions and performed well on facial recognition benchmarks for maintaining identity information in the generated facial image. Cao et al. [3] implemented similar ideas but included higher resolution images.

"FaceFeat-GAN: a two-stage approach for identity-preserving face synthesis" (Shen et al. [32]) split the GAN learning into two stages. The first stage learnt features using adversarial loss. These features were input to the image generator in the second stage. The generator was also trained using adversarial loss. They used an externally trained facial recognition model to generate an identity feature which they also used as input to the image generator. They generated diverse identity-preserving facial images as viewed from an independently trained facial recognition system. They were able to change attributes of the image by manipulating these features.

Bao et al. [1] introduced the CVAE-GAN which combined a conditional variational autoencoder with a conditional GAN and found the system easier to train and to converge faster than other designs. Diverse samples could

be randomly generated from the distribution to generate facial images of size 128x128 pixels. In a later work, Bao et al. [2] trained a GAN to combine the attributes of one image with the identity of another image through the use of adversarial training. They used an identity network that was trained to create an identity feature and an attribute network that was trained to create an attribute feature. These features were fed into a generator network which creates an image combining the attributes of the one image with the identity of the other. They used discriminator and identity classifier networks for feedback to the training networks. They trained the initial network on MS-Celeb-1M. To expand the attribute capabilities, they sourced 1M unlabeled faces from Google and Flickr and trained the network with the identity feature network and the classifier network locked. They had success in varying pose, emotion and lighting in unseen identities of size 128x128 pixels.

The model we used to synthesize identity preserving faces is based on Bao et al's attribute and identity networks. However, we used a pretrained network for the identity network and focused on using a significantly smaller front facing dataset. In addition, we evaluated how the synthetic faces perform as source faces for training a facial recognition network.

2.2 Deep Facial Recognition

Facial Recognition using Deep Neural Networks (DNNs) has become prevalent since significant progress was made on image classification problems in competitions like the ImageNet challenges. Face verification aims to classify two unseen facial images as the same identity or not. This requires generic identity traits to be learnt from the training images. It is often solved by generating an identity representation that be used to compare 2 facial images. One method is for the network to output a vector of fixed size (see Figure 2). A similarity metric such as the cosine angle or euclidean distance is then used between the identity vectors generated from the 2 facial images to determine how close the vectors are. The smaller the value, the higher the probability of the 2 faces belonging to the same individual.

Validating all facial images as being matching or non-matching to every other image in a large database is computationally expensive. Labeled Faces in the Wild (LFW) [13] uses a method of face verification that compares matching and non-matching samples from the dataset. In their 13K face dataset, 3000 matching and 3000 non-matching pairs are defined for testing. Most DNN systems trained on sufficient, diverse data achieve over 99% accuracy on LFW for the task of face verification.

Different methods for training an identity vector have been used over the years. Initially, cross entropy loss was used to train the identities as categories and to use the layer before the final identity category layer as the identity vector. The ideal separation in an identity vector is for facial images belonging to the same identity to have small separation and images belonging to different identities have as large a separation as possible. Training with cross entropy loss does not optimise for this and may result in a vector representation that does not generalise well.

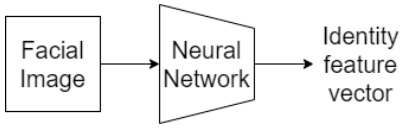

Fig. 2. Mapping a facial image (e.g. shaped 128x128x3) through a neural network to output an Identity feature vector (e.g. shaped 1x512).

Training with Triplet Loss was used in FaceNet (Schroff et al. [29]). Triplet loss draws a training sample along with one matching sample of the same identity and one non-matching sample of another identity. Matching and non-matching distances are calculated. Triplet loss minimises the matching distance and maximises the non-matching distance. To make this training as effective as possible, hard samples are needed that are either misclassified or almost misclassified. The network learns to map the facial images for the same identities in clusters around the same point and as far away from all other identity clusters as possible. This improves the generalisation of the identity vector for unseen faces.

A problem with Triplet loss is that finding hard samples can be computationally expensive. Deng et al. [6] introduced ArcFace loss which attempts to directly optimise the distances within the data itself. The angle between the identity vector and the target weights to the identity categories with an angular margin is optimised. ArcFace loss optimises the angle between the identity vectors of facial images of the same identity to be as small as possible and between images of different identities to be as large as possible.

We used a similar method of face verification to LFW with 3000 matching and non-match pairs. LFW is not used as a Test dataset as the size and diversity of the dataset we used for training was insufficient for us to achieve an LFW accuracy of over 90%. We trained our facial recognition networks with ArcFace loss and used a cosine similarity metric to perform face verification between unseen identities in the Validation and Test datasets.

2.2.1 Using Synthetic Faces

When generating new faces, it is common to test the results on several face related tasks, including how facial recognition systems evaluate the synthetic faces and using the synthetic faces for data augmentation.

Masi et al. [24] augmented the number of per-identity images to generate a dataset that was approximately 5 times the size of the original. Yucer et al. [38] trained a facial recognition model on a racially augmented dataset to learn a model that was less prone to racial bias. Both works found some benefit in using the additional synthetic faces.

In CVAE-GAN Bao et al. [1] used their method to generate 200 additional faces per identity as well as 100 faces of 5000 new identities for augmentation. They trained a facial recognition model and compared the results for verification on LFW. They found using the additional synthetic data added 1% increase in accuracy over using the unaugmented dataset.

Tran et al. [34] compared the performance of their learnt identity representation in DR-GAN with training facial recognition using their synthetic faces. They found their identity representation performed better than training with synthetic faces.

Kortylewski et al. [18] generated a synthetic dataset through the use of a 3DMM learnt from 200 neutral face scans and 160 expression deformations. This model allowed them to sample from the learnt distribution through the use of the 3DMM parameters and generate new identities as well as control the pose, illumination and expression in 3D to generate images of faces. One drawback is that the image is not a realistic "in the wild" image as the 2D image only contains the face region, not the rest of the head and body. The backgrounds are also not realistic. They generated 20K identities with 100 images per identity and trained a facial recognition network with 80% accuracy on LFW. They then fine-tuned the network using different amounts of Casia WebFace data and found that they could get similar results to current state of the art with only using a quarter of the real facial data.

It is not common to compare how well synthetic faces are able to be used to train a facial recognition system on their own. A key contribution of our work is looking at the differences between the synthetic and original facial images as viewed from 3 pretrained facial recognition systems as well as comparing the facial recognition performance when training with only synthetic faces. We also used the synthetic faces for data augmentation.

3 METHODS

This section presents methods for generating synthetic faces, for analysing the resulting face dataset, and for training the deep facial recognition networks.

3.1 Synthetic Faces

The method used to synthesize identity preserving faces was based on the work of Bao et al. [2]. The framework overview can be seen in Figure 3. The framework consisted of an Attribute encoder, A, an Identity encoder, I, and a face Generator, G. The Generator, G, takes the identity and attributes vectors from I and A respectively and outputs a face image. During training a Discriminator, D, was used for adversarial feedback to the face generator. We differed from Boa et al. in that a pretrained FaceNet implementation, trained on VGGFace, was used for the Identity encoder network, I. The Identity encoder was not trained further.

Face synthesis involved selecting an identity face image x^s and an attribute face image x^a as inputs. The goal of the face synthesis network was to output a new face image, x', with the same identity as x^s and the attributes (such as hair and background) of x^a.

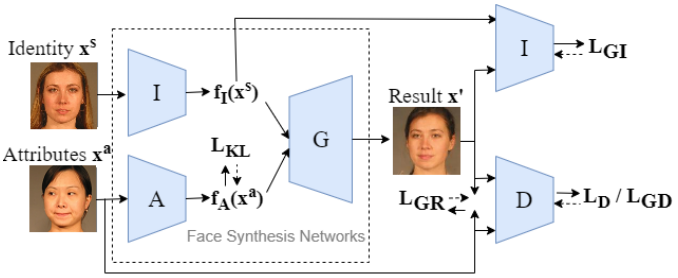

Fig. 3. High level face synthesis training framework with Identity Encoder (I), Attribute Encoder (A), Face Generator (G) and Discriminator (D) networks. Both I networks represent the same trained Identity Encoder.

Training of the face synthesis network involved two steps. First, we trained with $x^a = x^s$. The Attribute encoder and Generator were jointly trained to reconstruct the face and the Generator was trained to generate the same identity. Adversarial loss was also applied to the Generator via the Discriminator. The second training step selected another attribute image, x^a, so that $x^a \neq x^s$ and another image was synthesized by the Generator. The goal of this step was to train the Attribute encoder to transfer attributes to the synthesized face.

The Identity network generated an identity vector, $f_I(x^s)$. The synthesized face, x' was fed back into the Identity network and L2 loss, L_{GI}, is applied to the Generator for the difference between the initial identity vector and the identity vector of the synthesized face.

$$L_{GI} = \frac{1}{2}||f_I(x^s) - f_I(x')||_2^2 \tag{1}$$

The Attribute network generated an attribute vector, $f_A(x^a)$. The Attribute network was similar to a variational autoencoder and output a mean vector μ, and log-variance vector, ϵ. As backpropogration can not flow through a random layer, the reparametrisation trick was used during training. This learnt these vectors and used them to transform a Gaussian distribution from which the output vector was sampled. The resulting output vector was calculated in the following way:

$$f_A(x^a) = \mu + r * exp(0.5 * \epsilon) \tag{2}$$

where $r \sim \mathcal{N}(0, 1)$. Kullback-Leibler (KL) divergence was used as part of the loss when training the Attribute encoder. Bao et al. [2] found that the KL loss, L_{KL}, reduced the entanglement of identity information in the attribute vector. The KL loss encouraged a Gaussian distribution in the generated attribute vector which may be able to learn shared attributes across identities, such as background colour, better than diverse identity information. The loss for each item in a batch was:

$$L_{KL} = \frac{1}{2}\sum_{i=0}^{N}[exp(\epsilon_i) + \mu_i^2 - 1 - \epsilon_i] \tag{3}$$

Where the sum was over the elements in the vector of size N. The mean of the KL loss for each batch was used when training.

The Generator and Attribute encoder were jointly trained with a L2 reconstruction loss between the original and synthesized faces, L_{GR}. Taking the idea to increase stability by training at multiple scales of the Generator from Karnewar and Wang [15], we trained with the average L2 reconstruction loss over each scale of the Generator as the Generator grows the image. To obtain an RGB image at each scale additional convolutional and activation layers were trained to map from the current number of channels to the 3 RGB channels for the image at that scale.

$$L_{GR} = \frac{1}{2 \cdot N}\sum_{\sigma=1}^{N}||x_\sigma^a - x_\sigma'||_2^2 \tag{4}$$

where N=6, σ = 1,2,...,N and x^a_σ was resize to be $2^{\sigma+1}$x$2^{\sigma+1}$ pixels, the same size as x'_σ. The 6 scales trained were 4x4, 8x8, 16x16, 32x32, 64x64, 128x128.

In addition, the Generator was trained with adversarial loss, L_{GD}, from the Discriminator, D. The adversarial loss applied to G was based on the L2 loss between the final layer of the discriminator, f_D, for the real image and the reconstructed image.

$$L_{GD} = \frac{1}{2}||f_D(x^a) - f_D(x')||_2^2 \quad (5)$$

The discriminator was trained by minimising the standard GAN loss [10],

$$L_D = -\mathbb{E}_{x\sim P_r}[\log D(x^a)] - \mathbb{E}_{z\sim P_z}[\log(1 - D(G(z))] \quad (6)$$

which attempted to determine the difference between the reconstructions and real faces. z concatenates $f_I(x^s)$ and $f_A(x^a)$.

If the network was not trained with the appropriate losses applied to the appropriate networks, the attribute encoder could learn an entangled representation of identity and facial attributes, even with a pretrained identity network. To avoid this, the losses were combined in the following way when $x^a = x^s$:

$$L_A = L_{GR} + \lambda_{KL}L_{KL} \quad (7)$$

$$L_G = L_{GR} + \lambda_{GD}L_{GD} + \lambda_{GI}L_{GI} \quad (8)$$

The loss for the Discriminator is made up of L_D as previously defined.

When $x^a \neq x^s$ only a weighted reconstruction loss was applied:

$$L_A = \lambda_A L_{GR} \quad (9)$$

$$L_G = \lambda_A L_{GR} \quad (10)$$

We used $\lambda_A = 0.1$. The reconstruction loss was reduced as the generated image was not intended to be the same as either of the original images, however the attributes are being encouraged to be transferred to the known identity, hence it should be somewhat similar, for example with the same background or hair style.

3.2 The Synthetic Face Datasets

3 datasets were generated for evaluation and were used in the facial recognition training experiments. These are referred to as the Reconstructed, New Faces and New Identities datasets. The FRGC training set is referred to as the Original dataset and was used as a control dataset.

Reconstructed: All 18143 images in this dataset were reconstructions of the Original dataset. This dataset gives a representative view of the closest that could be expected to the Original dataset from faces synthesized by this network as the identity training was optimised over the reconstructed faces.

New Faces: Each new facial image in the New Faces dataset was constructed from the mean identity vector of an identity from the Original dataset, combined with the attributes from each image from a different identity in the Original dataset. This created 18143 facial images with the same 482 identities as in the Original dataset and with the existing attributes from other identities applied to different identities.

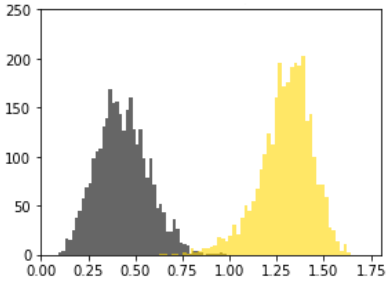

Fig. 4. Bimodal distribution of FaceNet euclidean distance scores for 3000 matching (grey) and 3000 non-matching (yellow) pairs of the Original dataset.

New Identities: Each new facial image in the New Identities dataset was constructed from 482 unique identities generated, combined with the attributes from each image from an identity in the Original dataset. This created 18143 faces with 482 new identities and with the same distribution of images to identities as the Original dataset.

3.3 Properties of a Facial Recognition Dataset

A common method for training a facial recognition system is to optimise the output to be a bimodal distribution where matching pairs map to one mode and non-matching pairs map to a second mode. These two modes can then be split by a threshold chosen for a specific anticipated error rate. Ideally these two modes are completely separable but in practice they are often not. The bimodal distribution for the Original dataset can be seeing in Figure 4. When evaluating a synthesized dataset we would anticipate seeing the same bimodal distribution.

The Original dataset contained 18143 faces. There were over 500K possible matching pairs and 150M possible non-matching pairs in this dataset. Evaluating this number of pairs over multiple datasets over multiple facial recognition systems could take a significant amount of time. Randomly sampling a subset of the pairs gave an approximation of the full distribution. We used 3000 matching and 3000 non-matching pairs randomly sampled from each dataset to provide a view of the distribution.

The score distributions produced were evaluated by 3 different pretrained facial recognition systems. A FaceNet implementation [8], an ArcFace implementation [14] and a DLib implementation [9] were used. The FaceNet implementation may have performed well due to being in the training loop. The ArcFace and DLib implementations were only used for evaluation and hence provided a useful comparison.

An additional property that was evaluated was the uniqueness of the faces synthesized for a new identity. The faces in the New Identities dataset were intended to be unique identities from the Original dataset. These new faces were evaluated through Face Identification by exhaustively searching for the lowest score between each facial image in the New Identities dataset and all images in the Original dataset. In addition, the lowest non-match score between each facial image in each identity of the Original dataset to

all other identities in the Original dataset was determined. The Original dataset provides a control that was considered as a normal lowest non-matching score distribution and hence was compared to the distribution of the New Identities to Original scores to determine if these distributions were similar. The New Identities to Original distances were controlled by a threshold, α_U, when finding new identities which could be tuned further. The New Identities would not be unique if the New Identities to Original score distributions have smaller distances than within the Original dataset.

3.4 Deep Facial Recognition

Our facial recognition networks were trained with ArcFace loss (Deng et al. [6]). ArcFace loss is optimised to use the cosine similarity metric, therefore cosine angle was used for all results to find the distance between two identity vectors to determine how similar they were. The smaller the cosine distance between two identity vectors, the more likely that the two faces are of the same identity.

A ResNet-50 network [12] pretrained on ImageNet[1] was used for the training of all facial recognition networks. Following Deng et al. [6], the layers following the last convolution layer of the network followed the format of BN-Dropout-FC-BN to output a 512-d identity vector. The network was trained with ArcFace loss with the 482 identities in the training dataset.

All networks to be compared were trained in exactly the same way with stochastic gradient descent. Unless specified, only the training dataset used changed per experiment. Five networks with the final layers randomly initialised were created and stored. Each experiment trained a network with each of these five configurations and the mean and standard deviations of the equal error rates (EER) per trained network combination were compared. The EER is when the false match rate is equal to the false non-match rate.

4 Experiments & Results

4.1 Synthetic Faces

4.1.1 Initial Experiments

GANs have been shown to produce state of the art, high resolution faces such as those generated by StyleGAN (Karras et al. [17]). Zhang et al. [39] found minor discrepancies on biometric image quality metrics when comparing StyleGAN to real images. Therefore a reasonable starting point for generating synthetic faces in current research was to use a GAN. Experiments were initially made to generate 64x64 images with a DCGAN architecture (Radford et al. [28]). This was relatively stable, however, the faces synthesized were often slightly deformed and exhibited artifacts. Moving to train 128x128 images was less stable, even when using the Progressive GAN method from Karras et al. [16] to grow the network output size. The issue of checkerboarding due to transposed convolutions growing the network size led to changing the DCGAN to use UpSampling for growing the network [25].

A goal for this research was to control the identity of the synthesized face. An early experiment trained an autoencoder to generate the faces. However the latent space of the encoder was highly entangled and the generator was unable to synthesize clear or crisp faces. Another experiment trained a face generator to convert an identity vector provided from a pretrained Identity network into a face. This was found to have low image quality, particularly outside of the central face. The background was particularly blocky. Interestingly, due to a single identity vector representing a pose invariant face, the network tended to generate only front facing faces despite not all images in the training set being front facing.

Another experiment combined an encoder for attributes and an encoder for identity to enforce the disentanglement of the identity from the attributes. This was similar to the work of Bao et al. [2] but without the variational or adversarial feedback components. This had some success with being able to represent an identity however the faces rendered were not clear or crisp. Experimenting with L1 instead of L2 reconstruction loss did not make a significant difference to the synthesized face quality.

The next set of experiments incorporated the variational encoder and adversarial feedback aspects from Bao et al. These experiments are described in detail in the following sections.

It should be noted that the above initial experiments were all trained with the CelebA dataset [22]. CelebA is a larger and more complex dataset than the final dataset used. The experiments were run for 50 epochs which is a smaller number of epochs than the final training run of the final networks used. Some of these experiments may work more effectively if trained with the same FRGC dataset used for training the final networks.

4.1.2 The Face Synthesis Network

The controlled environment portraits of the Facial Recognition Grand Challenge dataset [27] were used for training the face synthesis network. These facial images were cropped to include the whole face, head and hair[2]. All facial images were front facing with limited difference in pose. There was some variation in illumination. The training dataset contained 18143 facial images of 482 individuals. The number of images per identity ranged from 4 to 130 with a mean of 38.

A validation set of FRGC identities that did not appear in the training set was available. This dataset consisted of 3629 facial images of 86 identities. This was used to evaluate the performance on unseen images.

Using front facing faces reduced the complexity of the problem. The FRGC dataset allowed evaluation of this simpler case. While the training dataset size was not large for deep learning, the face distribution to learn was also limited in that all of the faces were front facing with limited pose differences and similar crop.

The Generator and Discriminator networks were based on the original DCGAN configuration (Radford et al. [28]). The Attributes network was based on the same network architecture as the Discriminator with the final layers splitting

1. Implementation: https://pytorch.org/vision/stable/models.html

2. Using a combination of the DLib frontal face detector and the imutils FaceAligner class

to generate the mean and log variance vectors. Exact details of the DNN layers are provided in Appendix A.

The values of $\lambda_{KL} = 0.0001$, $\lambda_{GD} = 0.0005$ and $\lambda_{GI} = 1$ were used when training our final face synthesis network.

Increasing λ_{KL} resulted in stronger identity results with the trade-off of less transfer of the attributes in the synthesized face. With $\lambda_{KL} = 0.0005$ the facial image was found to vary little. $\lambda_{KL} = 0.0001$ was found to be a reasonable trade-off.

Bao et al. used a value of $\lambda_{GD} = 0.001$ in their work. Experimentally it was found that $\lambda_{GD} = 0.0005$ performed better for our setup.

Attempts at varying λ_{GI} did not result in noticeable differences for $\lambda_{GI} = 10$ and $\lambda_{GI} = 100$. FaceNet normalised the identity vector output to place it on a unit hypersphere. An experiment using the non-normalised FaceNet identity vector in the L2 identity loss was shown to work well at early epochs and less well later. This may be because the L2 distance between the two non-normalised identity vectors were initially large which the normalisation damps while in the later epochs the L2 distances were smaller between the non-normalised vectors than the normalised vectors, therefore, over time the normalisation helps learning more. The identity was found to be impacted most significantly over the first 50 epochs by reducing the batch size. As batch size decreased, the final L2 identity loss after 50 epochs decreased more. This may be due to less averaging in the learning which allows different parts of the network to be optimised to generate more diverse identity features. It may be that the batch normalisation was more effective for this problem with smaller batch sizes. A batch size of 8 was used in the final network training.

Minimising the identity loss during the attribute training step ($x^a \neq x^s$) did not appear to work in the tests that were run. It may be interesting to revisit that in future work with the current knowledge of the training times and identity performance or, alternatively, to start training the attributes with identity loss from the network we have trained already.

All networks were trained with the Adam optimiser. The learning rate for the Attribute encoder and Generator networks was set to 0.0001. The learning rate for the Discriminator network was set to 0.0004. Setting the Discriminator learning rate to 0.0001 was found to result in lower quality faces and the training did not work as well.

Experiments were done with removing the multi-scale reconstruction loss. It was found that the identity loss reduced over the first 50 epochs and then consistently increased for the next 200 epochs while the reconstruction loss decreased and remained low.

The network was trained with the identity vectors of the faces in the dataset. Including training with $x^a \neq x^s$ improved the ability to generate a face-like image when using unseen identity vectors. To improved the ability to generate a face-like image further, an identity Generative Adversarial Network was trained to generate identity vectors with a similar distribution to the available identity vectors. The identity GAN was a learnt function that mapped a Gaussian vector to an identity vector. This was used during training to provide unseen identity vectors to the network. Every second Attribute training step ($x^a \neq x^s$) drew the identity vectors for attribute training from this distribution instead of using and identity from x^s. This improved the balance between the identity loss when $x^s = x^a$ and when $x^s \neq x^a$ when compared to training without these additional identity vectors.

The network used in these results trained for 300 epochs and balanced both the results from the differing losses on seen, unseen and random identities drawn from the GAN function while visually comparing the ability of the network to generate recognisable faces for unseen identity vectors.

4.1.3 Finding Valid New Identities

Finding a new identity consisted of several steps. A new identity needed to be found that was sufficiently different from all other identities. The generator needed to be able to use the new identity to synthesize faces that still belong to the same new identity. All new faces for the identity needed to be different enough to all other faces in the Original dataset so that these were indeed new and not different versions of the Original dataset identities.

Experiments were done with the identity GAN to generate new identities, however it was difficult to find enough unique identities using this method.

Schroff et al. [29] used a euclidean distance measure with FaceNet, therefore we used the euclidean distance between FaceNet Identity vectors and defined several thresholds.

The threshold for uniqueness while initially searching, α_S, was set to 1.3. This matches the mean FaceNet euclidean distance of the non-matching pairs in the Original dataset (see Table 1). We searched for new identities that were α_S euclidean distance from the other identities in the search parameters.

The FaceNet implementation outputs identity vectors that were normalised to lie on a unit 512 dimensional hypersphere. In theory any point on the hypersphere was a valid identity vector. In practice the face synthesis network was trained on identity vectors from the Original dataset so the identities that might synthesize a face with the same identity vector as the input identity vector may be biased towards the identities of the training dataset. New identities near an existing identity may be more likely to be synthesized by the generator and retain the same new identity. It may also be that the FaceNet implementation has not filled the entire available space on the hypersphere. As a result, the method for finding a new identity searched near an existing identity.

The method used to find a new identity followed these steps:

In: P_I, dataset of identity vectors. P_A, dataset of attribute vectors. G, the face generator network. f_I, the identity network.

Set: IDS = [], list of new identity vectors. $\alpha_S = 1.3$, search distance. $\alpha_I = 0.95$, non-matching distance to all Ids. $\alpha_M = 0.7$, matching distance. $\alpha_U = 0.6$, closest distance to P_I allowed.

1: sample $I_{original} \sim P_I$

2: I_{last} = IDS[-1] IF length(IDS)> 1 # last Id created

3: sample $I_{recent} \sim$ IDS[-10:-1] IF length(IDS)> 2

4: $I_{start} = 0 - I_{original}$ # start on the other side of the hypersphere.

5: $I_{initial}$ = minimise($I_{original}$, I_{start}, [I_{last}, I_{recent}], α_S) # Starting at I_{start} find a point on the hypersphere that is α_S distant from $I_{original}$, I_{last} and

I_{recent} using the Sequential Least Squares Programming implementation from SciPy.
6: sample $A \sim P_A$, $x'_{initial} = G(I_{initial}, A)$ # synthesize a face
7: $I_{new} = f_I(x'_{initial})$ # The new identity vector
8: IF I_{new} distance to ANY $IDS < \alpha_I$ GOTO 1:
9: REPEAT 200 times
10: sample $A \sim P_A, x'_{synth} = G(I_{new}, A)$
11: $I_{synth} = f_I(x'_{synth})$
12: IF I_{synth} distance to $I_{new} > \alpha_M$ GOTO 1:
13: IF I_{synth} distance to ANY $P_I < \alpha_U$ GOTO 1:
14: IDS $\leftarrow I_{new}$

In practice, when searching for 200 passing facial images, a retry scheme was used to find another A and synthesize another x'_{synth} instead of failing immediately. That is left out for simplicity. 500 identities were created for the New Identities dataset using this algorithm.

I_{recent} was randomly sampled from the last 10 identities to improve the diversity of the new identities found.

We used I_{new} as the new identity as I_{new} was an identity that the generator can synthesize and not the same as $I_{initial}$. All new identities were tested to be at least $\alpha_I = 0.95$ from all other identities generated. This distance is less that 1.3 as this was the distance that was practically attainable by the face synthesis network to generate more than 482 new identities. Any new identity that could generate 200 images within a distance of $\alpha_M = 0.7$ from the found I_{new} and $\alpha_U = 0.6$ from all other faces in the Original dataset was kept. α_M ensured clustering around I_{new}. α_U was the threshold for a face identification match. Anything closer would be considered to overlap with the Original dataset identities. Creating 200 faces showed that the face synthesis network could generate faces of this identity. All of these thresholds could be experimented with further.

To identify low quality faces, identity vectors for each image were generated using DLib [9]. The DLib implementation detected the face before generating a vector and returned no vector if there was no face detected. Any images that did not return an identity vector were considered low quality. 7 of the 500 identities included faces that were unable to generate Dlib identity vectors and were excluded. The first 482 identities were used from this set of 493 new identities. These identities were used as the identity vectors for generating future datasets.

4.1.4 Evaluating Face Dataset Properties

In this section the ability to generate faces is considered along with how the properties of the synthetic faces compare to the properties of the original faces as seen by other facial recognition systems.

A face synthesis network requires an ability to generate faces even when presented with unexpected input. To experiment with this a reconstruction was synthesized with an image of a teapot as input to the identity and attribute networks. The result of the face synthesis is seen in Figure 5. This shows the ability for the face synthesis network to output faces.

Some low quality faces were synthesized when searching for new identities. Examples of these faces can be seen in Figure 6. These images are face-like, though they are not acceptable as actual faces. Some examples of reconstructed faces, new faces and new identities are shown in Figure 7.

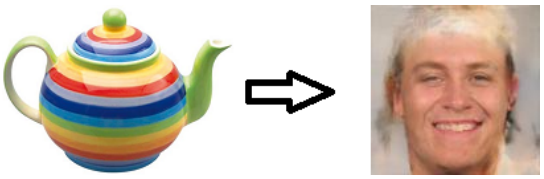

Fig. 5. Ability to generate faces: Using an image of a teapot as the image for both the Attribute and Identity networks results in a face.

Fig. 6. Ability to generate faces: Low quality faces, but still in the general face-like image space.

Initial experiments were conducted with 20 identities and 20 faces from the Original, Reconstructed, New Faces and New Identities datasets. The score distributions and statistics of the scores for all matching and non-matching pairs were compared. The outcome showed the scores to be distributed as expected.

The score distributions and statistics for each of the 4 datasets were evaluated by randomly drawing 3000 matching and non-matching samples 5 times. Table 1 shows the minimum and maximum scores with standard deviation as well as the mean score and standard deviation of the scores as viewed by each facial recognition system. The histograms of one sample of these distributions can be seen in Appendix B. Table 2 shows the mean EER and standard deviation of each dataset as viewed from each facial recognition system. The New Identities dataset had the most stable mean EER.

The mean threshold for the FaceNet mean EER was 0.831

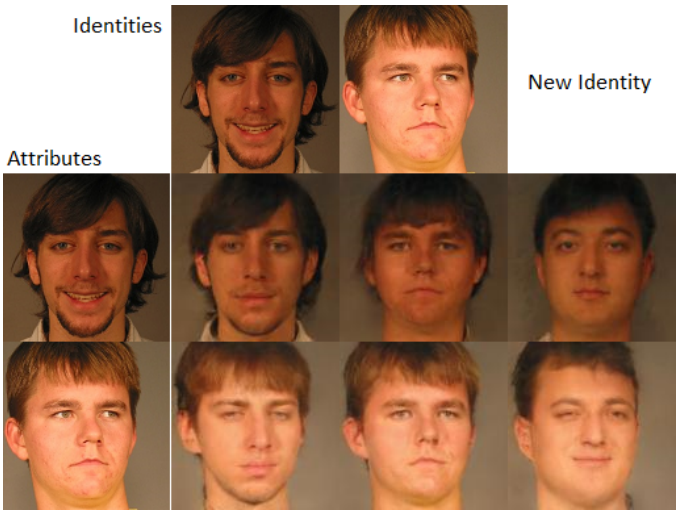

Fig. 7. Examples of reconstruction, new faces and new identities. Top Row: Identities to apply change to. First column - Attribute Image applied to all identities from the top row to generate the image at the intersection.

	Matching				Non-Matching			
FaceNet	Original	Reconstructed	New Faces	New Identities	Original	Reconstructed	New Faces	New Identities
Minimum	0.09 (0.01)	0.09 (0.02)	0.06 (0.00)	0.08 (0.00)	0.64 (0.04)	0.63 (0.07)	0.63 (0.07)	0.81 (0.05)
Maximum	0.95 (0.02)	1.06 (0.03)	1.22 (0.08)	1.17 (0.03)	1.66 (0.02)	1.67 (0.01)	1.64 (0.02)	1.68 (0.02)
Mean	0.43 (0.14)	0.47 (0.15)	0.38 (0.16)	0.43 (0.16)	1.30 (0.14)	1.30 (0.14)	1.26 (0.14)	1.29 (0.13)
DLib	Original	Reconstructed	New Faces	New Identities	Original	Reconstructed	New Faces	New Identities
Minimum	0.07 (0.01)	0.08 (0.01)	0.06 (0.01)	0.06 (0.01)	0.36 (0.01)	0.33 (0.02)	0.33 (0.02)	0.44 (0.02)
Maximum	0.54 (0.02)	0.61 (0.02)	0.68 (0.04)	0.66 (0.03)	1.06 (0.01)	1.06 (0.02)	1.05 (0.04)	1.01 (0.02)
Mean	0.27 (0.07)	0.30 (0.09)	0.26 (0.09)	0.29 (0.09)	0.79 (0.10)	0.78 (0.10)	0.75 (0.10)	0.74 (0.08)
ArcFace	Original	Reconstructed	New Faces	New Identities	Original	Reconstructed	New Faces	New Identities
Minimum	0.16 (0.01)	0.16 (0.01)	0.13 (0.02)	0.13 (0.01)	0.95 (0.07)	0.93 (0.05)	0.93 (0.07)	0.92 (0.04)
Maximum	1.05 (0.03)	1.19 (0.04)	1.19 (0.03)	1.24 (0.03)	1.61 (0.02)	1.58 (0.01)	1.56 (0.01)	1.57 (0.02)
Mean	0.56 (0.14)	0.61 (0.16)	0.55 (0.18)	0.59 (0.18)	1.37 (0.07)	1.36 (0.08)	1.32 (0.08)	1.31 (0.09)

TABLE 1
Score statistics per facial recognition system per Face dataset
Mean minimum score (standard deviation), mean maximum score (standard deviation), mean score (mean standard deviation of the scores) from 3000 matching and 3000 non-matching pairs randomly sampled 5 times.

	Original	Recon.	New Faces	New Id.
FaceNet	0.45% (0.07)	0.76% (0.06)	0.97% (0.08)	0.55% (0.07)
DLib	0.66% (0.05)	1.52% (0.15)	1.05% (0.05)	0.51% (0.06)
ArcFace	0.05% (0.03)	0.33% (0.05)	0.35% (0.03)	0.53% (0.09)

TABLE 2
mean EER (standard deviation) for each Face dataset per facial recognition system from 3000 matching and 3000 non-matching pairs randomly sampled 5 times.

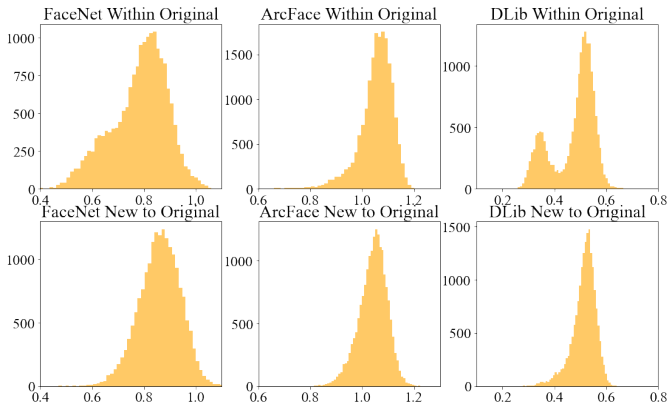

Fig. 8. Histograms of lowest non-matching score between all Identities for each facial image. Top Row - between all identities in Original dataset only. Bottom Row - between each image in the New Identities dataset to all images in the Original dataset. Lowest score was found by an exhaustive search between each facial image to all other images of other identities to find the lowest non-match score and hence the closest facial image in the dataset. The first histogram bucket on the left represents the lowest non-match score in the distribution. The New Identities dataset increased the lowest non-matching score slightly due to the $\alpha_U = 0.6$ threshold used when creating new identities.

on the Original dataset. In contrast, Schroff et al. [29] found a threshold of 1.242 when testing on LFW. Using this LFW threshold on the Original dataset resulted in a 29% false match rate and no false non-matches. Similarly, the mean threshold for the DLib mean EER on the Original dataset was 0.462, less than the published 0.6 LFW threshold for DLib [7]. This indicated that the LFW thresholds were not appropriate values to use with our datasets, probably due to the different make up of the datasets. Hence we have experimented and used the thresholds that we defined for finding new identities and determining uniqueness.

The distributions and statistics for each dataset within each facial recognition system's scores followed each other. The overall distribution for each dataset was the expected bimodal distribution. The distributions of the generated datasets followed the Original distribution. The one notable difference was that the generated datasets tend to have slightly wider score ranges than the Original dataset.

In all the distributions there was an overlap between matching and non-matching scores. This overlap may be reduced in the synthesized datasets by applying a α_M threshold similar to when searching for identities. The distance between identities may also be increased if a higher α_I threshold were used when searching for identities. Some tests were done with these thresholds and there seems some merit to experiment further, however, for these experiments this was left as is. These images may be hard samples that would help the training as the network may learn a better general identity representation with them in the training dataset, or they may be bad samples that would hinder the training due to noisy images that have managed to retain some of the identity features but potentially are another identity all together.

The distribution of the lowest non-match scores between the New Identities and the Original dataset followed that of the Original dataset. The lowest non-match score distributions can be seen in Figure 8. The lowest non-match score histograms have similar peaks. The minimum lowest non-matches between the New Identities and the Original dataset have higher scores than within the Original dataset. This is due to the $\alpha_U = 0.6$ thresholding when generating new identities. When the New Identities dataset was generated with the new identity vectors, no limits were used to reject faces too close to the Original dataset. 31 synthesized faces in the New Identities dataset were less than 0.6 away from an identity in the Original dataset.

Figure 9 shows the closest faces between the New Identities dataset and the Original dataset as seen by each facial recognition system along with an example at the $\alpha_U = 0.6$ threshold. This is a difficult task. The FaceNet scores between the closest matches for both of Dlib (0.912) and ArcFace (0.893) were not considered as matching according to the EER threshold for FaceNet (0.831) despite being well below the Dlib and ArcFace thresholds respectively. The $\alpha_U = 0.6$ threshold used when searching for new identities should be experimented with further to push the new facial images further from the Original dataset and when synthesizing the final datasets. The trade-off will be whether the face synthesis network is able to synthesize the required number of facial images with higher thresholds.

As a final test of uniqueness, the mean EER and standard deviation of the combined Original dataset and New Iden-

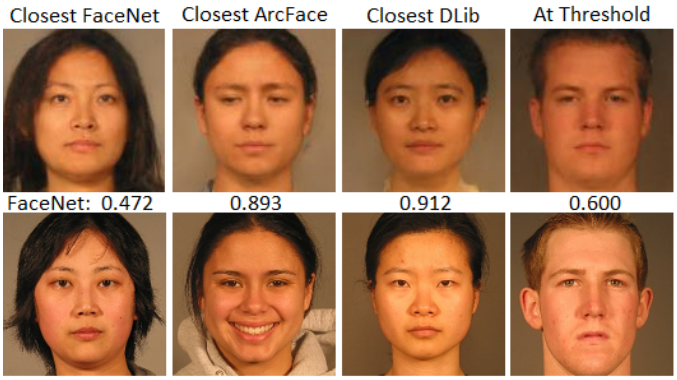

Fig. 9. Closest faces from New Identities (top row) to Original dataset (bottow row) as determined by FaceNet (1st column), ArcFace (2nd column) and Dlib (3rd column), and the $\alpha_U = 0.6$ threshold used (4th column). The FaceNet score for each column pair is shown in between the two images. This indicates how high the α_U threshold may need to be for highly convincing unique new identities. Both the ArcFace and Dlib closest faces would not be viewed as matching by the FaceNet EER threshold (0.831) found for the Original dataset.

tities dataset were found for each facial recognition system. These are 0.48% (0.08) for FaceNet, 0.56% (0.04) for Dlib and 0.21% (0.05) for ArcFace. This supports the conclusion that we have generated faces for unique new identities.

4.2 Deep Facial Recognition with Synthetic Faces

4.2.1 Datasets

The same FRGC training dataset that was used to train the synthesis network was used as the control dataset in the facial recognition training experiments. This is referred to as the Original dataset.

The FRGC dataset consists of front facing images, therefore, the Validation and Test datasets were constructed from similarly front facing images. These datasets were compiled from 3 sources. The neutral and smiling front facing images from the Face Research Lab London Set [5] provided 102 identities with 2 images per identity. The CMU Multi-PIE dataset [11] included a set of high resolution front facing images consisting of 249 identities with 3 to 5 images per identity. The FRAV2D dataset [30] included 109 identities. The first 12 front facing images per identity were used from FRAV2D. The identities from these datasets were split evenly into the Validation and Test datasets. This gave a more diverse set of Validation and Test facial images. The Validation dataset consists of 230 identities with 1281 images. The Test dataset consists of 230 identities and 1230 images. Unless otherwise stated in the experiment, the non-synthetic Validation dataset is used in all training.

All images in the Validation and Test datasets were resized to 256x256 and the face was cropped with MTCNN [8] with a margin of 33 pixels and resized to be 128x128, the size expected by the identity network. Deng et al. [6] found that using specific alignment can improve the performance despite training with unaligned faces. We have not focused on the alignment and it is possible that better facial recognition training and evaluation results can be achieved through precise alignment and specific cropping sizes.

The FRGC Validation dataset was used in these results to evaluate if the trained network performed significantly better on faces of the same FRGC distribution in the data augmentation experiments. The FRGC Test dataset consists of 3629 facial images belonging to 86 identities. The minimum number of images in any identity was 4 and the maximum was 120.

A completely Synthetic Validation dataset was also created. An additional 100 new identities were created in the same way as for the New Identities dataset. 20 faces of each identity were used. This Synthetic Validation set was used when running experiments with training a facial recognition network on completely synthetic faces.

3000 matching and non-matching pairs were randomly sampled and stored for each of the Test and Validation datasets. These 3000 matching and 3000 non-matching pairs were used to determine the EERs in the results.

4.2.2 Training Detail

Each facial recognition network was trained in the same way. Dropout was set to 0.5. We followed Deng et al. [6] when calculating ArcFace loss. The batch size was set to 128.

An early stopping criteria of 10 consecutive epochs without any improvement of the best Validation dataset EER was used. Following Simonyan and Zisserman [33], after the training stopped, the learning rate was reduced (divided by 10) and training started again. Adam was used as the optimiser. The learning rate started at 10^{-3} and ended at 10^{-7}.

All networks were trained with the same random data augmentation. All image values were normalised with the PyTorch torchvision.transforms.Normalize method into the range [-1,1] on each channel. All input images were of size 128x128. Validation images were only resized and center cropped. The specific augmentation that all networks were trained with included: Gaussian blur sigma in range [1,2] (20% likely), Random Rotation [-15,15] degrees (80% likely), Brightness Jitter (50% likely), Contrast Jitter (50% likely), Saturation Jitter (50% likely), Hue Jitter (10% likely), Grayscale (5% likely), Randomly resized with range [80%, 120%] and cropped to 128x128.

4.2.3 Only Synthetic Faces

In this section we consider the question: Does training with completely synthetic faces result in similar performance to training with the original non-synthetic faces?

Networks were trained with each of the Original, Reconstructed, New Faces and New Identity datasets. The Validation dataset used in these training runs was the same and based on real faces. This enabled like-for-like comparisons across the datasets as well as in the future data augmentation experiments as the only input variable that was changing was the training dataset, though the stochastic components used in the training would influence each outcome. The mean EER and standard deviation of the EERs for the 5 trained networks are shown in the 18143 faces column of Table 3. The trained networks did not reach the same level of performance as the facial recognition systems used for the dataset evalution. Those systems were trained with one or more orders of magnitude more data with higher diversity. Despite this, our results remain useful for

comparison between training with real and synthetic front facing faces with fewer than 20k images.

In addition to training the whole dataset, a sample of the dataset is drawn to reduce the dataset size to half (9072 faces), a quarter (4536 faces) and one eighth (2268 faces) to compare training outcomes at different dataset sizes. These datasets were generated by drawing one image from each of the 482 identities in turn until the full dataset size was reached. The performance can be seen in the additional columns of Table 3.

	Validation results			
Total Faces	18143	9072	4536	2268
Original	1.49 (0.12)	1.92 (0.23)	2.41 (0.30)	4.44 (0.63)
Reconstructed	2.30 (0.14)	2.56 (0.30)	3.51 (0.18)	4.82 (0.78)
New Faces	4.61 (0.49)	4.62 (0.31)	5.53 (0.22)	5.89 (0.71)
New Identities	4.80 (0.78)	4.81 (0.53)	5.28 (0.77)	5.82 (0.98)
	Test results			
Total Faces	18143	9072	4536	2268
Original	2.21 (0.39)	2.83 (0.23)	3.54 (0.23)	5.99 (0.73)
Reconstructed	2.95 (0.25)	3.28 (0.32)	4.41 (0.67)	6.21 (0.94)
New Faces	5.13 (0.34)	4.93 (0.43)	5.80 (0.63)	6.43 (0.56)
New Identities	5.09 (1.14)	5.25 (0.53)	6.01 (0.76)	7.02 (0.79)

TABLE 3
Mean EER (standard deviation) results over 5 trained networks for All (18143) faces, Half (9072) faces, Quarter (4536) faces and Eighth (2268) faces per dataset.

To understand the overall stability of the different datasets, each dataset was divide into 3 equal parts of identities and each subset was trained as previously. The mean and standard deviations for each of the 3 parts are show in Table 4.

	Validation results		
Segment	1	2	3
Original	3.47 (0.78)	3.76 (0.48)	3.36 (0.40)
Reconstructed	3.74 (0.43)	4.50 (0.13)	4.40 (0.34)
New Faces	7.18 (0.41)	7.45 (0.60)	7.83 (0.94)
New Identities	8.22 (0.54)	7.73 (0.16)	7.65 (0.62)
	Test results		
Segment	1	2	3
Original	4.51 (0.56)	4.26 (0.55)	4.34 (0.45)
Reconstructed	4.54 (0.59)	5.47 (0.54)	4.78 (0.27)
New Faces	7.62 (0.32)	7.13 (0.46)	7.84 (0.65)
New Identities	8.21 (0.46)	7.63 (0.37)	7.64 (0.89)

TABLE 4
Mean EER (standard deviation) results over 5 trained networks: Training each third of identities of each dataset.

To understand the change in results for a completely synthetic dataset, the training was repeated with the whole New Identities dataset using the Synthetic Validation dataset. This resulted in a mean EER of 3.86% (0.46) on the Synthetic Validation dataset and a mean EER of 5.27% (0.34) on the Test dataset.

Similarly, the results of training each third of the New Identities dataset with the Synthetic Validation dataset were 8.94% (0.65), 8.95% (0.39) and 9.72% (0.96) mean EER on the Test dataset. This had a slightly higher mean EER than training with the non-synthetic Validation dataset however the variation over each third of the identities showed a similar stability.

Training with the synthetic faces resulted in a higher mean EER than training with the Original dataset. The Reconstructed dataset performs the closest to the Original dataset. This was anticipated as the Reconstructed dataset most closely matches the main training objective of the synthesis network. Training with the New Faces dataset performed very slightly better than with the New Identities. The synthesis network was trained to generate new faces of the known identities and hence this may be why the New Faces performed slightly better than new identities it was never trained to synthesize. From the view of the 3 facial recognition networks, the New Identities dataset was the most stable with respect to the mean EER, but this did not result in better training results.

The variance in the Validation and Test results was generally slightly higher when training with the New Faces and New Identities, though all results had higher variance at the smallest size dataset. It may be that the New Identities and New Faces introduced more noise in image quality or lower intra-identity consistency and hence the outliers seen previously in the wider facial recognition score ranges may have been noisy samples rather than useful hard samples.

The performance was similar for each third of the identities in each dataset. Across the datasets, different thirds had higher and lower variance. The mean EERs, while varying a bit, were in comparable ranges. The New Face and New Identities datasets had consistently higher mean EERs.

The mean EER on the Test dataset when training with the New Identities dataset and the Synthetic Validation dataset was slightly higher than when training with the non-synthetic Validation dataset. The Synthetic Validation dataset was closer to the FRGC dataset distribution as that is what the face synthesis network was trained on that generated the Synthetic Validation dataset. This may explain the lower mean EER achieved on the Synthetic Validation dataset as it is closer to the training distribution. The non-synthetic Validation and Test datasets were drawn from the same, non-FRGC distribution hence it is not unexpected that the mean EER increases slightly when facing a different facial image distribution. How effective the synthetic faces are will be limited by the distribution of faces that can be synthesized, which may be limited to the distribution of facial images of the original face synthesis network's training dataset.

Training with fully synthetic faces was possible, but it was not as effective as training with the original faces despite the distributions of the facial recognition properties following each other. However, if privacy were a highly important property, using synthetic faces of new identities may be good enough.

4.2.4 *Supplementing Training data with Synthetic Faces*

In this section we consider the question: Does training with the original non-synthetic faces supplemented with synthetic faces improve the performance of a facial recognition network?

Several scenarios were considered for augmenting a dataset. The first scenario considers if we could augment a small subset of the Original dataset and improve the outcome. The largest reduction in performance was seen when training with an eighth of the Original dataset, thus we trained with these 2268 images supplemented with 3 different datasets of synthetic faces - the New Identities dataset, the New Faces dataset and finally both the New Identities and New Faces datasets. The second scenario considers if we could improve the performance when training with the whole Original dataset by augmenting with

Original Faces / Identities	Synthetic Faces / Original Identities / New Identities	Validation	Test	FRGC Test
2268 / 482	0 / 0 / 0	4.44 (0.63)	5.99 (0.73)	8.41 (0.86)
2268 / 482	18143 / 0 / 482	2.93 (0.39)	3.48 (0.52)	4.45 (1.68)
2268 / 482	18143 / 482 / 0	3.55 (2.37)	3.99 (2.42)	4.25 (3.82)
2268 / 482	36286 / 482 / 482	4.45 (1.94)	4.94 (2.18)	5.36 (2.13)
18143 / 482	0 / 0 / 0	1.49 (0.12)	2.21 (0.39)	2.40 (0.32)
18143 / 482	18143 / 0 / 482	1.42 (0.20)	2.04 (0.19)	1.97 (0.15)
18143 / 482	18143 / 482 / 0	3.34 (2.01)	3.75 (2.08)	2.20 (1.19)
18143 / 482	36286 / 482 / 482	3.40 (1.61)	4.17 (1.82)	3.40 (1.52)
2268 / 482	96400 / 0 / 482	1.90 (0.09)	2.32 (0.09)	3.54 (0.29)
2268 / 482	78257 / 482 / 0	2.01 (0.16)	2.26 (0.13)	2.16 (0.31)
2268 / 482	174657 / 482 / 482	1.93 (0.27)	2.26 (0.27)	2.37 (0.14)
18143 / 482	96400 / 0 / 482	4.31 (2.26)	4.85 (2.44)	4.58 (2.21)
18143 / 482	78257 / 482 / 0	1.69 (0.27)	2.17 (0.31)	1.61 (0.26)
18143 / 482	174657 / 482 / 482	1.81 (0.11)	1.84 (0.20)	1.96 (0.26)

TABLE 5
Mean EER (standard deviation) over 5 trained networks when training with different additional synthetic face sets. Top rows represent training no additional data as well as with New Identities and New Faces datasets. Bottom 6 rows represent training with the Big New Faces and Big New Identities datasets.

the same datasets of additional synthetic faces. The results of the training are shown in Table 5. In these results we also report on the FRGC Test dataset to determine if the data augmentation leads to overfitting to the FRGC face distribution.

As an additional experiment in augmenting with many synthetic faces, 2 additional larger datasets were synthesized in order to test augmentation with a large number of images. The Big New Faces dataset augmented the Original dataset until there were 200 faces per identity. Training with the Big New Faces dataset resulted in a mean EER of 2.47% (0.06) on the Test dataset. The Big New Identities dataset contained 200 synthetic faces for each the 482 new identities. Training with the Big New Identities dataset resulted in a mean EER of 4.94% (1.86) on the Test dataset and when trained with the Synthetic Validation dataset a Test dataset mean EER of 8.24% (3.66) was achieved. The results of training with these augmented datasets can be seen in the bottom 6 rows of Table 5.

Training with 2268 faces from the Original dataset augmented with synthetic faces improved the results. The synthesis network was trained on the full Original dataset, so this experiment may not be too informative beyond there is clearly potential in using synthetic faces to expand a dataset.

It is notable that adding real faces to the synthetic faces datasets and training resulted in lower mean EERs than when training with only synthetic faces. This implies the real faces were providing features that the synthetic faces do not have or the training was unable to find. It is also notable that when training with the combined New Identities, New Faces and 2268 real faces there was no improvement over using either synthetic dataset on their own.

Training with the whole Original dataset and the New Identities did improve the mean EER on Validation, Test and FRGC Test datasets over training with only the Original dataset. Using the Big datasets, there was no improvement in the Validation dataset however there were improvements in the Test datasets when training with the Big New Faces and both Big datasets. These are promising results and highlight the potential value in data augmentation with synthetic faces.

The variance was increased when using the New Faces dataset for data augmentation. Early experiments were done that combined real and generated faces of the same identities which showed the score distributions to be similar, but possibly it was not always the case. When evaluating some of the high variance results, it could be seen that one or two of the outcomes had a significantly higher EER than the mean. In a similar manner, the variance decreased when training with the Big New Identities and 2268 original faces while it increased when training with the Big New Identities and whole Original dataset. More training runs may be required to understand the variance and mean better for this dataset but higher variance implies that the Big New Identities dataset might be less stable for training.

When augmenting with the Big datasets for training, the most improvement was seen in the FRGC Test dataset while the Validation dataset mean EER was not improved. This may show a tendency to overfit to the FRGC dataset distribution as the number of synthetic faces increases in the training dataset. This may be because the face synthesis network was trained on the FRGC distribution and will synthesize faces close to that distribution.

Based on these results, using the synthetic faces for augmenting an existing dataset was possible and can improve the EER. Using the synthetic data was not guaranteed to improve the performance, but it may do so for a given dataset. In the scenario where multiple runs are made searching for the one best trained network, training with synthetic faces can improve the performance.

5 Conclusion

The initial research sub-questions have been evaluated and discussed within the context of training facial recognition with synthetic faces. Some methods for generating synthetic faces have been presented and a method implemented that focused on the problem of synthesizing front facing faces. The synthetic faces generated with this method followed the distributions of the original face dataset when evaluated with the 3 facial recognition systems.

More work is needed on the criteria of uniqueness between the New Identity dataset and the Original dataset. The results are good enough but could be better. Experimenting with the thresholds and evaluating the resulting distributions would provide more insight into if the dataset can be truly unique with the given training data. This could then lead to publishing the synthetic faces with no privacy issues linking to the Original dataset.

When training with the Original dataset of real faces a mean EER of 2.21% was found on the Test dataset. When

training with only the synthetic faces of the New Identities dataset, the mean EER increased to 5.27%. The New Identities dataset distribution followed the Original dataset's distribution and had a stable mean EER when viewed from the 3 pretrained facial recognition systems. It would seem that some properties for identity have been embedded into these synthetic faces, but they are not clearly learnable by a new system being trained on them alone. It may be of benefit to further analyse the distributions with other statistical methods, such as the Kolmogorov–Smirnov method, to get a more detailed picture of the differences in the distributions. However, we have shown that a dataset of only synthetic faces can be used to train a facial recognition system which has potential privacy benefits.

When using the synthetic faces for data augmentation, the New Identities dataset improved the mean EER on the Test dataset to 2.04% and the combined Big New Identities and Big New Faces datasets improved the mean EER to 1.84%. However this improvement is not guaranteed with some datasets having higher mean EER after training with more synthetic faces.

Using synthetic faces for data augmentation is a common experiment. Our results show that finding synthetic faces to be beneficial in augmenting data for facial recognition training is not the same as using only synthetic faces for facial recognition training. This is a useful result to be aware of.

There is more to understand in generating synthetic faces to train a facial recognition system to the same performance as with real faces. A future research question is to determine what the relevant difference between the real faces and the synthetic faces is. Is it more valuable to focus on more identity preserving loss functions to make the identity features in the synthetic faces more findable? Or is there something other than the identity features, such as facial image quality, that is impacting the performance when training with only synthetic faces? Early experiments with removing outliers that are farthest away from the mean identity vector show a slight improvement in results and that the image quality of the outliers may be lower. This is a good avenue to understand better as hard, good quality faces may be useful but noisy, low quality images may not.

Many research directions can branch out from our work. A face synthesis network that can synthesize new front facing faces in the general distribution of the FRGC dataset has been trained and could be extended with more facial images and new identities to make the synthetic face distribution more diverse. Introducing the ability to control attributes such as pose, expression and other facial features more deliberately would be highly beneficial and would potentially improve general facial recognition training further.

REFERENCES

[1] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. CVAE-GAN: Fine-grained image generation through asymmetric training. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2764–2773, 2017. doi: 10.1109/ICCV.2017.299.

[2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6713–6722, 2018. doi: 10.1109/CVPR.2018.00702.

[3] Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. *CoRR*, abs/1806.08472, 2018. URL http://arxiv.org/abs/1806.08472.

[4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified Generative Adversarial Networks for multi-domain image-to-image translation, 2018. URL https://arxiv.org/abs/1711.09020.

[5] Lisa DeBruine and Benedict Jones. Face Research Lab London set, May 2017. URL https://figshare.com/articles/dataset/Face_Research_Lab_London_Set/5047666/5.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019. doi: 10.1109/CVPR.2019.00482.

[7] DLib. Face recognition, accessed 2021-08-18. URL http://dlib.net/face_recognition.py.html.

[8] Tim Esler. Face recognition using pytorch, accessed 2021-08-18. URL https://github.com/timesler/facenet-pytorch.

[9] Adam Geitgey. Face recognition, accessed 2021-08-18. URL https://github.com/ageitgey/face_recognition.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Advances in neural information processing systems*, volume 27, 2014.

[11] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-PIE. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–8, 2008. doi: 10.1109/AFGR.2008.4813399.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[13] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[14] Deep Insight. Insightface: 2d and 3d face analysis project, accessed 2021-08-18. URL https://github.com/deepinsight/insightface.

[15] Animesh Karnewar and Oliver Wang. MSG-GAN: Multi-scale gradients for Generative Adversarial Networks, 2020. URL https://arxiv.org/abs/1903.06048.

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation, 2018. URL https://arxiv.org/abs/1710.10196.

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for Generative Adversar-

ial Networks, 2019. URL https://arxiv.org/abs/1812.04948.

[18] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2261–2268, 2019. doi: 10.1109/CVPRW.2019.00279.

[19] Lingjun Li, Yali Peng, Guoyong Qiu, Zengguo Sun, and Shigang Liu. A survey of virtual sample generation technology for face recognition. In *Artificial Intelligence Review*, volume 50, pages 1–20, 2018. doi: 10.1007/s10462-016-9537-z.

[20] Mu Li, Wangmeng Zuo, and David Zhang. Convolutional network for attribute-driven and identity-preserving human face generation, 2016. URL https://arxiv.org/abs/1608.06434.

[21] Tingting Li, Ruihe Qian, Chao Dong, Si Liu, Qiong Yan, Wenwu Zhu, and Liang Lin. BeautyGAN: Instance-level facial makeup transfer with deep Generative Adversarial Network. In *Proceedings of the 26th ACM International Conference on Multimedia*, page 645–653, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356657. doi: 10.1145/3240508.3240618.

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[23] Jiang-Jing Lv, Xiao-Hu Shao, Jia-Shui Huang, Xiang-Dong Zhou, and Xi Zhou. Data augmentation for face recognition. *Neurocomputing*, 230:184–196, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.12.025. URL https://www.sciencedirect.com/science/article/pii/S0925231216315016.

[24] Iacopo Masi, Anh Tuan Tran, Jatuporn Toy Leksut, Tal Hassner, and Gerard Medioni. Do we really need to collect millions of faces for effective face recognition?, 2016. URL https://arxiv.org/abs/1603.07057.

[25] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. doi: 10.23915/distill.00003. URL http://distill.pub/2016/deconv-checkerboard.

[26] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press, September 2015. ISBN 1-901725-53-7. doi: 10.5244/C.29.41.

[27] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, Jin Chang, K. Hoffman, J. Marques, Jaesik Min, and W. Worek. Overview of the face recognition grand challenge. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 947–954 vol. 1, 2005. doi: 10.1109/CVPR.2005.268.

[28] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional Generative Adversarial Networks, 2016. URL https://arxiv.org/abs/1511.06434.

[29] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. doi: 10.1109/CVPR.2015.7298682.

[30] Ángel Serrano, Isaac Martín de Diego, Cristina Conde, Enrique Cabello, Linlin Shen, and Li Bai. Influence of wavelet frequency and orientation in an SVM-Based parallel gabor PCA face verification system. In *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning*, IDEAL'07, page 219–228, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3540772251.

[31] Yujun Shen, Ping Luo, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaoou Tang. FaceID-GAN: Learning a symmetry three-player GAN for identity-preserving face synthesis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2018. doi: 10.1109/CVPR.2018.00092.

[32] Yujun Shen, Bolei Zhou, Ping Luo, and Xiaoou Tang. FaceFeat-GAN: a two-stage approach for identity-preserving face synthesis, 2018. URL https://arxiv.org/abs/1812.01288.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. URL https://arxiv.org/abs/1409.1556.

[34] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1283–1292, 2017. doi: 10.1109/CVPR.2017.141.

[35] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes, 2017. URL https://arxiv.org/abs/1611.05507.

[36] Xiang Wang, Kai Wang, and Shiguo Lian. A survey on face data augmentation for the training of deep neural networks. *Neural Computing and Applications*, 32(19):15503–15531, Mar 2020. ISSN 1433-3058. doi: 10.1007/s00521-020-04748-3.

[37] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Learning face representation from scratch, 2014. URL https://arxiv.org/abs/1411.7923.

[38] Seyma Yucer, Samet Akçay, Noura Al-Moubayed, and Toby P. Breckon. Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation, 2020. URL https://arxiv.org/abs/2004.08945.

[39] Haoyu Zhang, Marcel Grimmer, Raghavendra Ramachandra, Kiran Raja, and Christoph Busch. On the applicability of synthetic data for face recognition, 2021. URL https://arxiv.org/abs/2104.02815.

Attribute Encoder		Discriminator	
Input image	3x128x128	Input Image	3x128x128
Conv 4x4 (stride 2x2)	128x64x64	Conv 4x4 (stride 2x2)	128x64x64
LReLU (0.2)	-	LReLU (0.2)	-
Conv 4x4 (stride 2x2)	256x32x32	Conv 4x4 (stride 2x2)	256x32x32
Batch Normalisation	-	Batch Normalisation	-
LReLU (0.2)	-	LReLU (0.2)	-
Conv 4x4 (stride 2x2)	512x16x16	Conv 4x4 (stride 2x2)	512x16x16
Batch Normalisation	-	Batch Normalisation	-
LReLU (0.2)	-	LReLU (0.2)	-
Conv 4x4 (stride 2x2)	512x8x8	Conv 4x4 (stride 2x2)	512x8x8
Batch Normalisation	-	Batch Normalisation	-
LReLU (0.2)	-	LReLU (0.2)	-
Conv 4x4 (stride 2x2)	1024x4x4	Conv 4x4 (stride 2x2)	1024x4x4
Batch Normalisation	-	Batch Normalisation	-
LReLU (0.2)	-	LReLU (0.2)	-
Conv 4x4 (stride 1x1)	2048x1x1	Conv 4x4 (stride 1x1)	512x1x1
Tanh	-	Fully Connected	1x1x1
Fully Connected (mean)	128x1x1	Sigmoid	1x1x1
Fully Connected (var)	128x1x1		

TABLE 6
Attribute Encoder and Discriminator DNN layers

Generator	
Input Vector	640x1x1
Conv 4x4 (stride 1x1)	1024x4x4
LReLU (0.2)	-
Conv 3x3 (stride 1x1)	1024x4x4
Batch Normalisation	-
LReLU (0.2)	-
Output: Conv 1x1 (stride 1x1), Tanh	3x4x4
Upsample	1024x8x8
Conv 3x3 (stride 1x1)	512x8x8
LReLU (0.2)	-
Conv 3x3 (stride 1x1)	512x8x8
Batch Normalisation	-
LReLU (0.2)	-
Output: Conv 1x1 (stride 1x1), Tanh	3x8x8
Upsample	512x16x16
Conv 3x3 (stride 1x1)	512x16x16
LReLU (0.2)	-
Conv 3x3 (stride 1x1)	512x16x16
Batch Normalisation	-
LReLU (0.2)	-
Output: Conv 1x1 (stride 1x1), Tanh	3x16x16
Upsample	512x32x32
Conv 3x3 (stride 1x1)	256x32x32
LReLU (0.2)	-
Conv 3x3 (stride 1x1)	256x32x32
Batch Normalisation	-
LReLU (0.2)	-
Output: Conv 1x1 (stride 1x1), Tanh	3x32x32
Upsample	256x64x64
Conv 3x3 (stride 1x1)	128x64x64
LReLU (0.2)	-
Conv 3x3 (stride 1x1)	128x64x64
Batch Normalisation	-
LReLU (0.2)	-
Output: Conv 1x1 (stride 1x1), Tanh	3x64x64
Upsample	128x128x128
Conv 3x3 (stride 1x1)	64x128x128
LReLU (0.2)	-
Conv 3x3 (stride 1x1)	64x128x128
Batch Normalisation	-
LReLU (0.2)	-
Output: Conv 1x1 (stride 1x1), Tanh	3x128x128

TABLE 7
Generator DNN layers

APPENDIX A
MODEL ARCHITECTURES

Table 6 shows the DNN layer configuration for the encoder and discriminator networks used in the training of the face synthesis network. These are based on the DCGAN architecture.

Table 7 shows the DNN layer configuration for the face generator network used in the training of the face synthesis network. These are based on a modified DCGAN architecture. Upsampling and the additional depth were used in order to achieve higher image quality.

APPENDIX B
FACIAL RECOGNITION SCORE DISTRIBUTIONS

Figure 10 shows the different score distributions for the datasets for each of FaceNet, ArcFace and DLib using a euclidean distance measure. Each column represents the distribution from the perspective of a facial recognition system. Each row represents the distribution of a different dataset. Looking down each column, it can be seen that the distributions follow each other and have similar bimodal distributions.

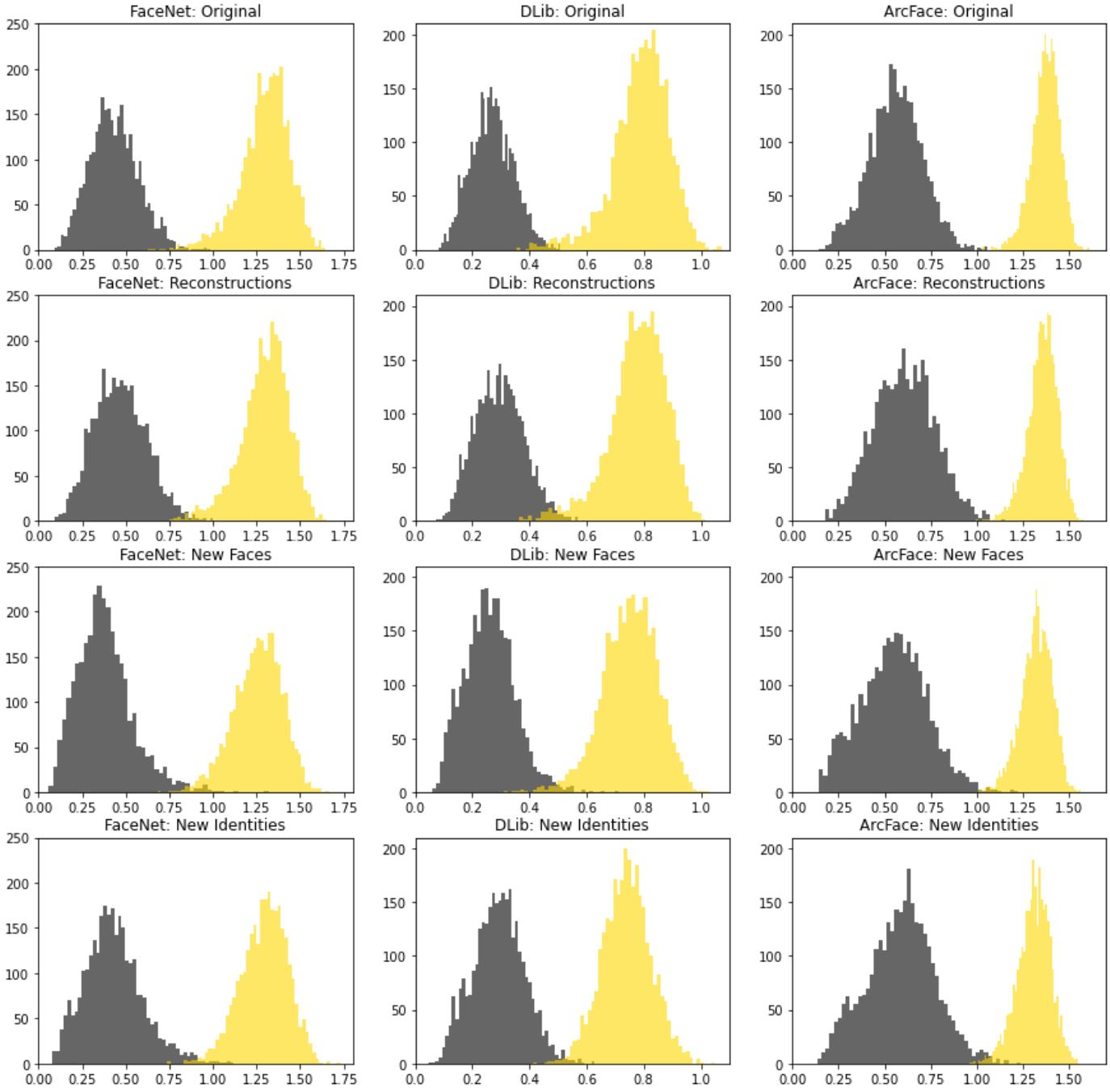

Fig. 10. Histograms of euclidean distance scores for 3000 matching and 3000 non-matching pairs for each of FaceNet, DLib and ArcFace. Each column represents the distribution from the perspective of a facial recognition system. Each row represents the distribution of a different dataset.