Article

Optimizing Few-Shot Learning based on Variational Autoencoders

Ruoqi Wei [1] and Ausif Mahmood [1] *

[1] Department of Computer Science & Engineering, University of Bridgeport, CT 06604 USA; ruoqiwei@my.bridgeport.edu(R.W.); mahmood@bridgeport.edu(A.M.)
* Correspondence: mahmood@bridgeport.edu; Tel.:+1-(203) 576-4737

Abstract: Despite the importance of few-shot learning, the lack of labeled training data in the real world, makes it extremely challenging for existing machine learning methods as this limited data set does not represent the data variance well. In this research, we suggest employing a generative approach using variational autoencoders (VAEs), which can be used specifically to optimize few-shot learning tasks by generating new samples with more intra-class variations. The purpose of our research is to increase the size of the training data set using various methods to improve the accuracy and robustness of the few-shot face recognition. Specifically, we employ the VAE generator to increase the size of the training data set, including the basic and the novel sets while utilizing transfer learning as the backend. Based on extensive experimental research, we analyze various data augmentation methods to observe how each method affects the accuracy of face recognition. We conclude that the face generation method we proposed can effectively improve the recognition accuracy rate to 96.47% using both the base and the novel sets.

Keywords: Deep learning; Variational Autoencoders (VAEs); data representation learning; generative models; unsupervised learning; few shot learning; latent space; transfer learning

1. Introduction

The explosion of big data has provided us enough training samples in the real world, which will facilitate the development of deep learning performance[1-3]. Moreover, with the development of high-performance computing devices such as graphics processing units (GPUs) and CPU clusters in recent years, the training of large-scale deep learning models has been greatly improved for big data feature learning. Nowadays, deep learning models can usually be successful with millions of model parameters and a large amount of available labeled training big data. Deep learning has also fueled great strides in a variety of computer vision problems, such as object detection [4,5], motion tracking [6,7] ,action recognition [8,9], human pose estimation [10,11], and semantic segmentation [12,13]. Face recognition has also witnessed great success with convolutional neural networks (CNN) [14-16]. However, many applications of this kind of deep learning success in face recognition can only be realized on the premise of having a large amount of labeled data. Moreover, in real life, due to restrictions such as data security management and labor costs, it is impractical to obtain such a large amount of labeled data.

Humans, after learning only a few images of a target, can recognize and sometimes, they can even perceptually recognize the same target without learning the target image. Inspired by the ability of humans to learn quickly from a small number of samples, the field of artificial intelligence (AI) is currently actively researching few-shot learning, to solve the problems caused by limited data sets in recognition by imitating the process of rapid recognition of the human brain to make AI applications closer to the actual real-world scene.

The purpose of few-shot learning is to learn the classifier of new classes; each class provides only a few training examples [17-19]. For instance, when it comes to practical

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

applications of face recognition, such as surveillance and security, the face recognition system should be able to recognize people who have only seen it a few times, which is the ability of a machine to see and understand things the same way humans do. Among the existing few-sample learning methods, data augmentation is one of the important methods. It is a weakly supervised series of techniques aimed at expanding data sets with additional data points. However, it is very challenging for existing machine learning methods because this limited data set does not represent the data variance well. Unbalanced data distribution or lack of data will cause over-parameterization and over-fitting problems, resulting in a significant decrease in the effectiveness of deep learning results. Especially-in face recognition, the variance of facial attributes such as smile can cause huge intra-class differences between faces of the same person, seriously affecting face recognition performance. Therefore, a data augmentation approach is needed to generate new facial images with greater intra-class variations.

Data augmentation is a method that can significantly increase the variety of data that can be used for training models without requiring the manual collection of new data. The easiest way to generate data is to augment by simple image transformation [20] such as image translation, noise addition, color jittering, and rotation. These methods create new data by transforming the original image and can prevent over-fitting between classes. However, these methods generate only repeated versions of the original data, and the data set continues to lack intra-class variations. To solve these problems, deep generative models have been attempted to refine the data to convert the original data into features, thereby increasing the intra-class variations of the data set.

The principle of the deep generative model to generate new data is to use distribution estimation and sampling [21,22]. The traditional deep generative model is the Boltzmann series, that is, deep belief networks (DBNs)[23] and deep Boltzmann machines (DBMs) [24]. However, one of their main limitations is the high computational cost during the operation process [3]. The latest deep generative networks are VAEs [25,26] and generative adversarial networks (GANs) [27]. VAEs do not suffer problems encountered in GANs, mainly nonconvergence causing mode collapse, and are difficult to evaluate [21,27,28]. Besides, a key benefit of VAEs is the ability to control the distribution of the latent representation vector z, which can combine VAEs with representation learning to improve the downstream tasks further[29,30]. VAEs can learn the smooth latent representations of the input data [31] and can thus generate new meaningful samples in an unsupervised manner. Moreover, the generated image quality and diversity are improved by the existing VAE-variants such as β-VAE [32] and InfoVAE [33], which combine VAEs with disentanglement learning, GMVAE [34] and VaDE [35] ,which give the VAE the ability for classification with unsupervised clustering, f-VAEGAN-D2 [36] and Zero-VAE-GAN [37] , which combine VAEs with GANs and few-shot learning, *S*-VAE [38] , which combines VAEs with spherical latent representation, VQ-VAE [39] , which combines VAEs with discrete latent representation, VAE-GAN [40] , which combines VAEs and GANs to generate a high-quality image, and *S*3VAE [41], which combines VAEs with disentangled representations of sequential data. These properties have allowed VAEs to enjoy success especially in computer vision, for example, static image generation [42], zero shot learning [43-45], image super-resolution[46,47] , and semantic image inpainting [48,49]. Therefore, in our paper, we try to utilize the VAE to the few-shot learning problem due to the scarcity of labeled training data.

We employ the model proposed by [50] to train a model with a base set based on transfer learning and then build a feature extractor. Then, we fine-tune to learn the actual label of the target using a novel image data set from the data augmentation. A face data set is divided into a base set and a one-shot set. The base set implies that each person has only one picture. The one-shot set also means each person has only one picture. What needs to be emphasized here is that there is no overlap between the base set and the one-shot set. Using transfer learning as the backend, we have implemented various types of

 doi:10.20944/preprints202109.0389.v1

data generation to increase the intra-class variations of the base set, thereby achieving higher recognition accuracy.

The structure of our paper is as follows. Section 2 describes some background works about our research. Section 3 explains the research plan in detail: (1) proposed architecture overview, (2) deep convolutional networks, (3) generation networks, (4) verification networks, and (5) identification networks. Section 4 provides experiments with implementation details and results. Summary, conclusion, and future work are given in Section 5, and references have been delineated at the end.

2. Related Work

2.1. Components of Face Recognition

The complete deep face recognition system can be divided into three modules [51]: (1) a face detector to locate faces in images, (2) a facial landmark detector that can align faces with normalized coordinates, and (3) the face recognition module. We only focus on the face recognition module throughout the remainder of this paper.

Furthermore, face recognition can be divided into face verification (answers the question, is this the same person?) and face identification (addresses the question, who is this person?) [50]. Face verification calculates one-to-one similarity to determine whether two images belong to the same face, while face recognition calculates one-to-many similarities to determine the specific identity of the face. The face recognition module includes the following processes: (1) face processing, (2) deep feature extraction, and (3) face matching or face identification. In this research, we present the data augmentation method, which is facial attribute manipulation (FAM) using VAEs. Second, we adopt a pretrained architecture of Inception ResNet v1 [52] as a CNN for deep feature extraction for face data augmentation, face verification, and face identification tasks.

2.2. Few-shot Learning

The Few-shot learning was proposed to solve the problem of learning new classes in classifiers, where each class provides only a small number of training samples [17-19]. With the development of deep learning techniques, the existing FSL methods can be divided into the following aspects: (1) metric learning [53,54], which learns metrics/similarity of few-shot samples through deep networks; (2) meta-learning [55], which learns a meta-model in multiple FSL tasks, and then the meta-model can be used to predict the weight of the model in a new FSL task; (3) transfer learning [56,57], which uses pretrained weights as initialization; and (4) data augmentation [20,58], which is a form of weak supervision [59] and aims to expand the few-shot sample data set with additional data points. It should be noted that there is no absolute distinction between the four categories. In this paper, our idea (1) utilizes triplet-loss-based metric learning for few-shot face verification, (2) utilizes data augmentation to improve the few-shot face recognition, and (3) builds upon the transfer learning backend.

2.3. Data Augmentation

Data augmentation [20,58,60] is a technique to increase the amount of available training data. The difference between data augmentation and data synthesis/generation is that the augmented data is generated based on existing data. Data augmentation methods can effectively solve the challenge of few-shot face recognition. They can be used to augment not only the training data set but also the test data set. In this section, we aim to introduce the traditional and latest methods of data augmentation. These methods can be divided into: (1) basic image processing, (2) model-based

 doi:10.20944/preprints202109.0389.v1

transformation, and (3) generative approach using generative models such as GANs and VAEs.

The simplest method in data augmentation is the expansion of the data set through basic digital image processing [61]. Digital image processing [62] including photometric transformations [63] and geometric transformations [64]. Geometric transformation changes the geometry of an image by transferring the pixel values of the image to new locations. This kind of transformation includes flipping, cropping translation, rotation, zooming, perspective transformation, reflection, scaling, cropping, padding, mirroring, elastic distortion, lens distortion, and other such processes. Photometric transformation changes the RGB channel by altering the pixel color to new values, which include gray scaling, color jittering, filtering, contrast adjustment, lighting perturbation, random erasing, noise adding, vignetting, and so on. Image processing is a traditional but powerful image augmentation method. However, these methods generate only repeated versions of the original data, and yet the data set lacks intra-class variations. Therefore, its application is primarily to transform the entire image uniformly, rather than transforming specific attributes of the face.

Model-based face data augmentation is to fit a face model to the input face and then generate faces with different attributes by changing the parameters of the fixed model. Commonly used model-based face data augmentation can be divided into 2D active appearance models (2D AAMs)[65] and 3D morphable models (3DMMs) [66]. The common feature of AAMs and 3DMMs is that they both are composed of a linear shape model and a linear texture model. However, the shape components are different between them, which are 2D for AAMs and 3D for 3DMMs. Although 2D- and 3D model-based methods can generate more accurate and diverse 2D and 3D faces, these methods, however, have some drawbacks. One of the biggest challenges is the difficulty in generating the teeth and mouth of the human face because these models can only generate the surface of the skin, not the eyes, teeth, and mouth cavity [58]. Another shortcoming is that when the head posture changes, the lack of occlusion area causes the artifacts [20]. Therefore, it is difficult to reconstruct a complete and accurate face model from a single 2D image through model-based face data augmentation, and the computation cost in this method is also very expensive. In this research, our generative model-based transformation method does not do only with pose transformation and expression transfer but also with realistic facial attributes transformation at an affordable cost.

2.4. Generative Models

In recent years, with the rapid development of big data and GPUs, deep generative models have greatly improved the performance of data generation. Among them, VAEs [25,26,29] and GANs [27] are the two most popular models. The basic idea of generative models is to generate new data from modeled distribution by learning the data distribution of the training set. The rationale behind GANs is to learn the mapping from a latent space to real data distribution through adversarial training. After learning such a nonlinear mapping, GAN is capable of producing photo-realistic images by sampling latent code from a random distribution [67]. Conversely, VAEs have a key benefit that is their ability to control the distribution of the latent representation vector z that can combine VAEs with representation learning to enhance the downstream tasks further. However, compared with GAN, the samples VAEs generate tend to be blurred and of lower quality. Recently, a breakthrough has been made in

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

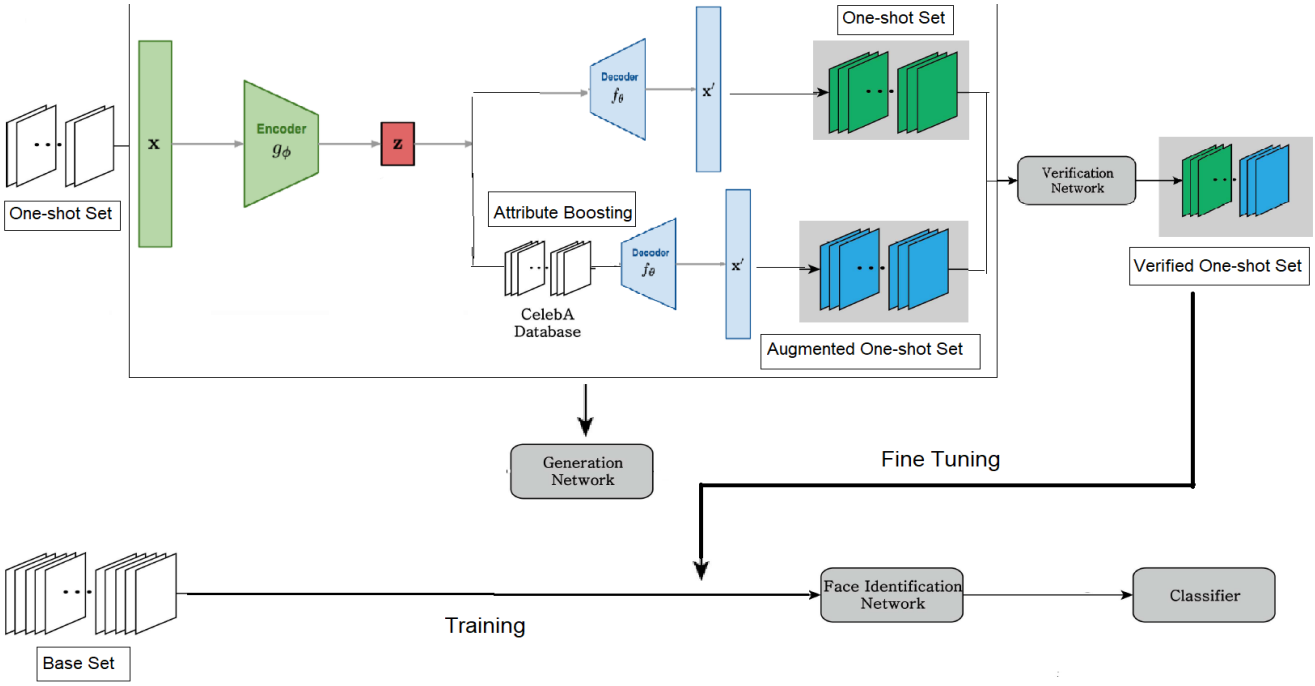

Figure 1. Proposed Architecture Overview.

VAEs by employing perceptual loss function instead of reconstruction loss. The perceptual loss function is based on the high-level features extracted from pretrained deep CNNs to train feed-forward networks. It captures perceptual differences and spatial correlation between output and ground-truth images and solves the problem of blurry figure, thus resulting in high image quality [68]. Therefore, in our research, we employ VAEs using perceptual loss to generate networks.

2.5. Transfer Learning

The methods of transfer learning can be divided into three categories according to the different situations of the source and the target domains and the corresponding tasks [69]. These are: (1) inductive transfer learning, (2) transductive transfer learning, and (3) unsupervised transfer learning (Figure 6). Furthermore, the inductive transfer learning method can be summarized into four situations based on the "what to transfer":

(1) instance-based transfer learning: It refers to reweighting samples in the source domain and correcting the marginal distribution difference between it and the target domain. Then these reweighted source sample instances are directly trained in the target domain. These methods work best when the conditional distributions in the two domains are the same.

(2) Feature-representation-transfer approach: It is suitable for homogeneous and heterogeneous problems. For heterogeneous problems, the goal is to narrow the gap between the source and target feature spaces. This method converts both the target and the source domains into a low-dimensional common latent feature space so that potentially meaningful structures can be discovered. For homogeneous problems, the goal is to narrow the gap between the marginal and conditional distributions of the source and the target domains. This method works best when the source and the target domains have the same label space.

(3) Parameter-transfer approach: This type of transfer learning transfers knowledge through the shared parameters of the source and the target domain

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

learner models. As the pretrained model on the source domain has learned a well-defined structure, the pretrained model can be transferred to the target model if the two tasks are related. Since fine-tuning requires much less labeled data, this approach can potentially save time, reduce costs, and help improve robustness.

(4) Relational knowledge-transfer problem [50]: The basic assumption of this method is that there are some common relations between the data in the source and the target domains. Therefore, the knowledge to be transferred is the common relationship between the source and the target domains.

In this research, we focus on parameter-transfer-approach-based transfer learning, which means fine-tuning the CNN parameters from a pretrained model using a target training data set is a particular form of transfer learning.

2.6. Facial Attribute Manipulation

In the last few years, face attribute analysis has made considerable progress along with deep learning. Facial attribute analysis based on deep learning includes two research directions: (1) facial attribute estimation (FAE), which is used to identify whether there are facial attributes in a given image, and (2) FAM, which is used to synthesize or remove specific facial attributes [70]. In this research, we focus on FAM. The latest progress of the FAM method is primarily built around the deep learning generative models, of which GANs and VAEs are the two most popular models. There are two main methods to get the FAM on generative models: (1) model-based methods and (2) extra-condition-based methods [70]. Furthermore, there are two kinds of extra-condition-based methods: (1) attribute vectors as extra conditions, which, with extra input vector, rely on simple linear interpolation, and (2) reference exemplars as extra conditions, which directly learn the image-to-image translation along with attributes that is popular for unsupervised disentanglement learning. However, disentangling is not easy to achieve, and to obtain better disentangling, the quality of reconstruction must be sacrificed [71]. Therefore, in this research, we focus on the first method that takes an attribute vector as the guidance to manipulate the desired attribute. Specifically, by changing a specific face attribute vector, the attributes of the face can be updated accordingly, referred to as deep feature interpolation (DFI) [72].

3. Research Plan

3.1. Proposed Architecture Overview

In this study, we analyzed the data augmentation method based on variational Autoencoder (VAE) in order to improve the accuracy and robustness of few-shot face recognition. We also increased the size of the training dataset in various ways to observe how data augmentation affects identification accuracy.

The proposed idea is as follows (as shown in Fig.1): A face dataset is divided into base set and one-shot set. The base set means that each person has just one picture. The one-shot set also means that each person has only one picture. It should be noted here that there is no overlap between the basic set and the one-shot set. The face identification network is first pre-trained on the base set, and then fine-tuned through the augmented one-shot set. However, the face identification accuracy may not be good enough. Since the accuracy of individual recognition is proportional to the number of training images for each person, we can increase the accuracy of one-shot set by adding augmented data. We, thus, use the proposed data augmentation method to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

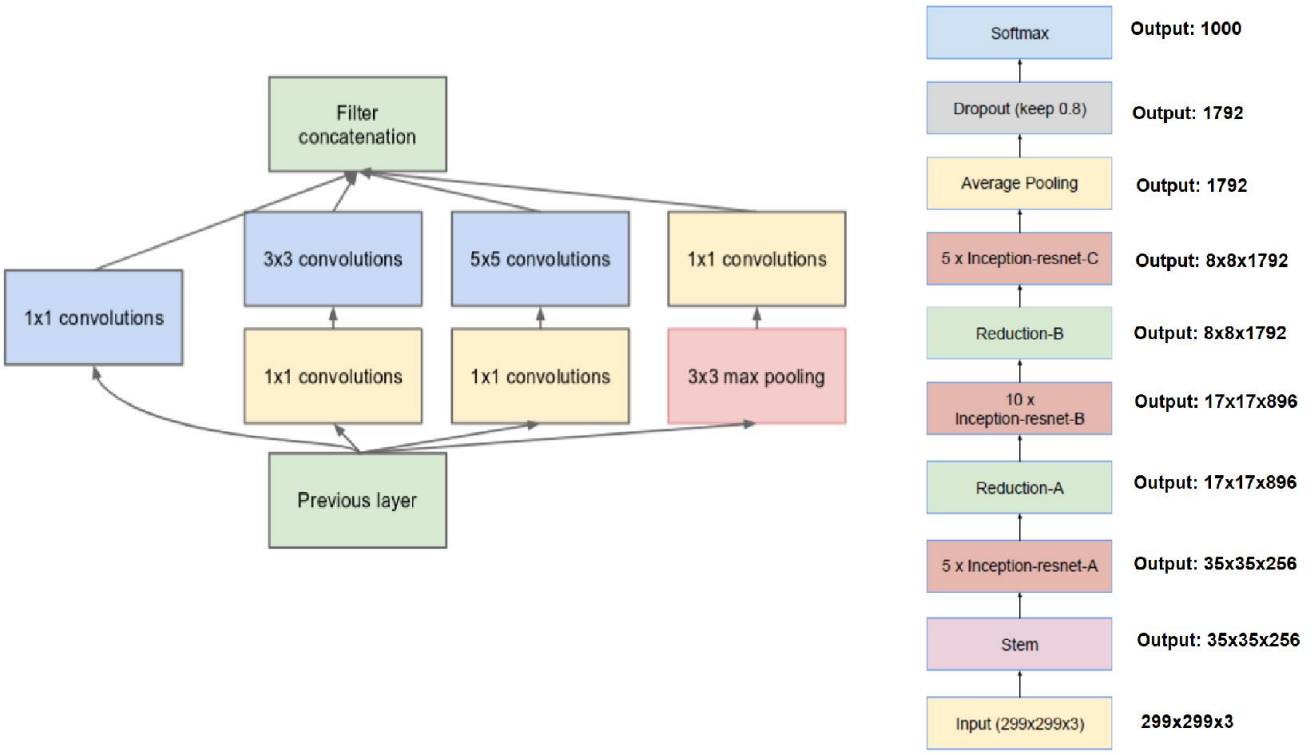

Figure 2. Inception module with dimension reductions(left) and Schema for Inception-ResNet-v1(right)

increase the one-shot set and study the change of identification accuracy by this method. However, the problem with this method is that the performance of face identification may decrease with several augmented data because the identity information is not sufficient, especially those images generated by VAEs. In order to solve this problem, we use a verification network to filter images that cannot be recognized, that is, if the augmented data can successfully pass the verification network, we pick out a subset from it. Finally, we can use these qualified augmented one-shot sets to fine-tune the face identification network with original base set and one-shot set. We hope to use this method and transfer learning as the backend to achieve higher accuracy of few-shot face recognition.

3.2. Deep Convolutional Networks

We chose the best architecture for the results of Computation Accuracy Tradeoff [50],]—Inception Resnet V1 as the Deep Convolutional Networks for the FaceNet system(as shown in Fig.2).

This deep neural network is almost the same as described in [73]. The main difference between them is that the L_2 pooling is used in a local specific area instead of the maximum pooling (m). The pooling layer can reduce the number of parameters in the subsequent operation. The idea of L_2 pooling is to use L_2 regularized for pixel values in a local specific area, i.e., except for the final average pooling, the pooling is always 3×3 and is parallel to the convolution modules in each Inception module. If the dimensionality is reduced after pooling, it is represented by p. We then utilize 1×1, 3×3, and 5×5 pooling to concatenate and get the final output. Table 1 describes CNN network in detail. Note that all of our specific networks described in the next sections are based on this CNN framework.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

3.3. Generation Network

In our research, we employ VAE using FaceNet-based [50] perceptual loss similar to the paper [74] for face image generation with boosting attributes. Specifically, the pixel-by-pixel reconstruction loss of the deep convolutional VAE is replaced by a feature perceptual loss based on a pre-trained deep CNN. The feature perceptual loss is to calculate the difference between the hidden representations of two images extracted from a pretrained deep CNN such as AlexNet [75] and VGGNet [76] trained on ImageNet [14].

For the results of Computation Accuracy Tradeoff [50], we choose the best architecture, which is Inception ResNet V1, as the Deep Convolutional Networks for the FaceNet system.This method attempts to improve the quality of the image generated by the VAE by ensuring the consistency of the hidden representation of the input image and the output image. It also imposes the spatial correlation consistency of the two images. The generative model consists of two parts—one is the autoencoder network, which includes an encoder network $E(x)$ and a decoder network $D(z)$, and the other is a pre-trained deep CNN, which is used to calculate the feature perceptual loss network ϕ. The encoder maps an input image x to a latent vector $z = E(x)$, then, the decoder maps the latent vector z back to image or data space $x' = D(z)$. After the VAE is trained, the decoder network can use the given vector z to generate a new image. We need two loss functions to train VAE: the first is KL divergence loss $\mathcal{L}_{KL} = D_{kl}[q(z|x)||p(z)]$ [25], which is used to ensure that the latent vector z is a Gaussian random variable. The other is feature perceptual loss that computes the difference between hidden layer representations, i.e., $\mathcal{L}_{Rec} = \mathcal{L}_1 + \mathcal{L}_2 + \ldots + \mathcal{L}_n$,where $\mathcal{L}_n$ is the feature loss at the n^{th} hidden layer. During the training process, the pre-trained CNN network is fixed and is only used for advanced feature extraction. KL divergence loss $\mathcal{L}_{kl}$ is only used to update the encoder network, and feature perception loss $\mathcal{L}_{Rec}$is used to update the encoder and decoder parameters.

3.3.1. Variational Autoencoder Network Architecture

The neural networks of the encoder and decoder are both constructed from deep CNN models such as AlexNet [75] and VGGNet [76]. As shown in Fig.3,this structure includes fully connected (FC) layers as well as convolutional (Conv) layers. The input image passes through 4 Conv layers and the last FC layer till the latent variable space is reached. The two convolutional layers in the encoder network achieve feature maps' dimensionality reduction using stride of 2 and a kernel size of 4 x 4. After each Conv layer, there will be a batch normalization layer and a LeakyReLU activation layer. Finally, two fully connected output layers (for μ and σ^2) are used to calculate KL divergence loss and sample latent variable z. The generative model $p(x|z)$ takes the sampled latent variables z received by μ and σ^2 and, using the reparameterization trick, feeds it through one FC layers and 4 Conv layers until a reconstructed output is obtained. Finally, it uses a stride of 1 and a kernel size of 3×3 in the deconvolution to obtain the reconstruction image. For upsampling, compared to the fractional-strided convolutions used in other works[13,22], we use the nearest neighbor method at a scale of 2. After each Conv layer, there will also be a batch normalization layer and a LeakyReLU activation layer to help stabilize training.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

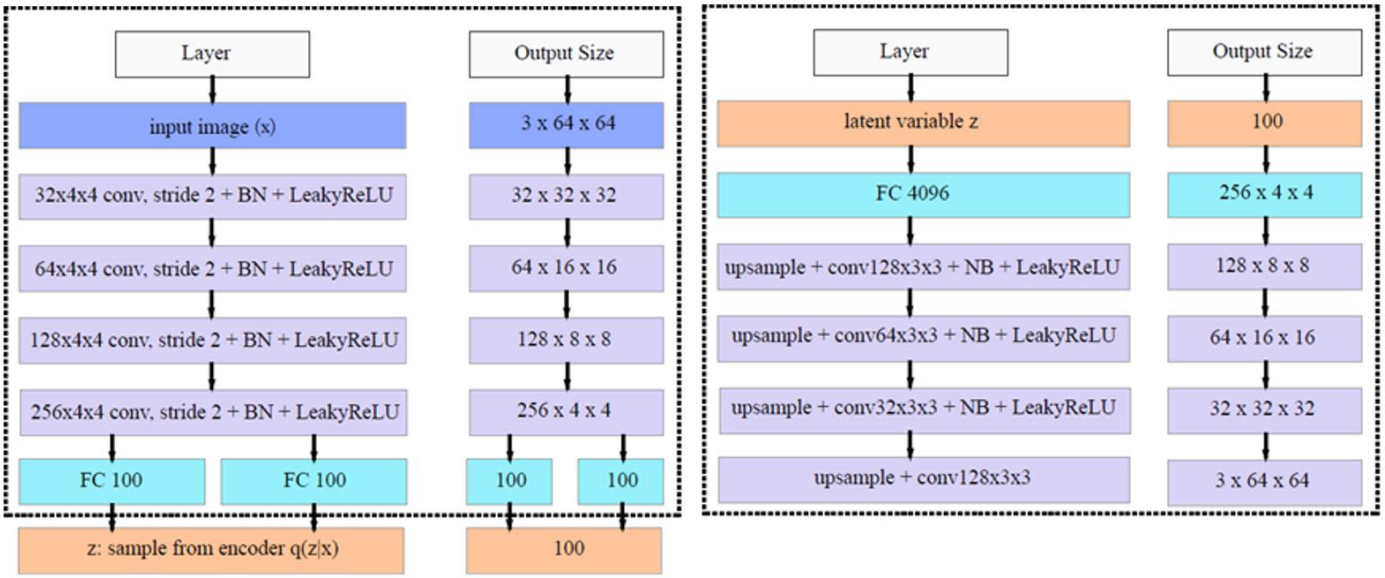

Figure 3. VAE Architecture for the encoder network(left) and the decoder network (right).Source: Adapted from [77].

3.3.2. Feature Perceptual Loss

The feature perception loss of two images is defined as the difference between hidden representations in the pre-trained deep CNN ϕ. We use Inception ResNet V1 as the Deep Convolutional Networks for the FaceNet system in our experiment (as shown in Fig.4).

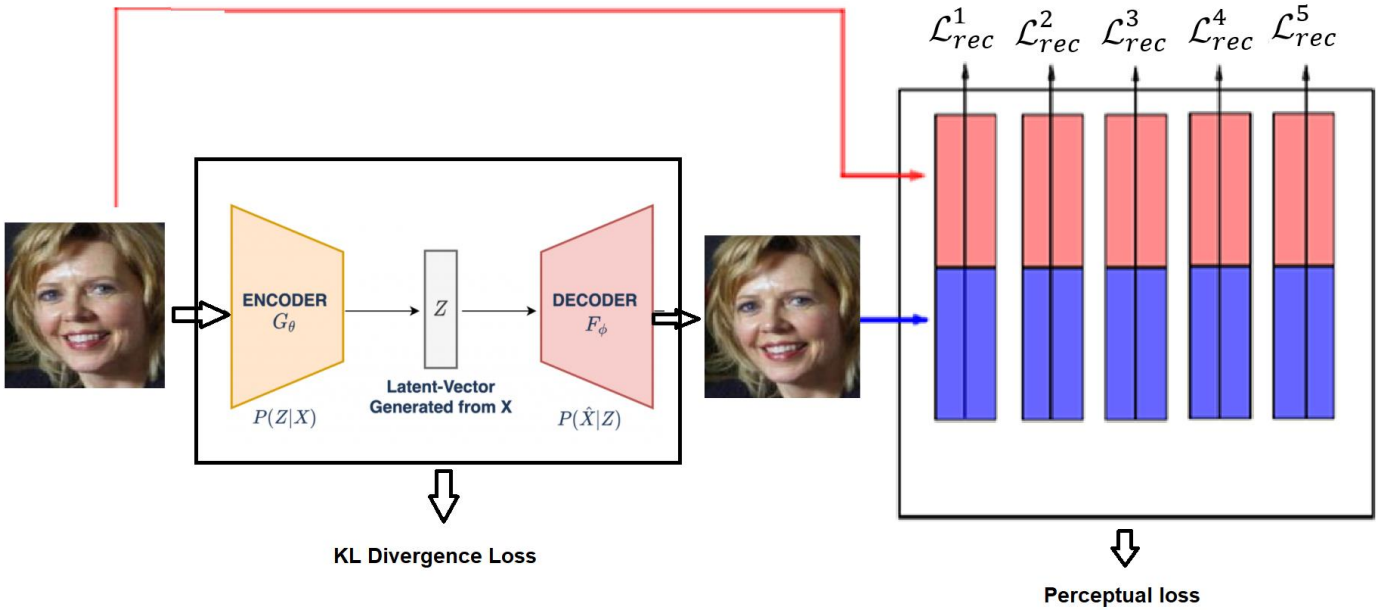

Figure 4. VAE with Perceptual Loss Architecture Overview.

The idea of feature perceptual loss is to ensure the consistency of the hidden representation of the input image and the output image and imposes the spatial correlation consistency of the two images. The reason for using feature perceptual loss to obtain better visual quality of the output image is that the hidden representation can capture important perceptual quality representations, and that the smaller difference in the hidden representation indicates the consistency of the spatial correlation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

between the input and the output. Specifically, the feature perception loss of a layer ($\mathcal{L}_{rec}^{n}$) between two images x and x' is the squared Euclidean distance between them. When the input image x is fed to network ϕ, the n^{th} hidden layer can be represented by $\phi(x)^n$. $\phi(x)^n$ is a 3D volume block array of a shape [$C^n \times W^n \times H^n$]; W^n and H^n are the width and height of each feature map of the n^{th} layer; C^n represents the number of filters.

$$\mathcal{L}_{rec}^{n} = \frac{1}{2C^n \times W^n \times H^n} \sum_{c=1}^{C^n} \sum_{w=1}^{W^n} \sum_{h=1}^{H^n} (\phi(x)_{c,w,h}^{n} - \phi(x')_{c,w,h}^{n})^2$$

The final reconstruction loss is the total loss obtained by adding the losses of the different layers of the deep CNN network, that is $\mathcal{L}_{rec} = \sum_n \mathcal{L}_{rec}^{n}$. In addition, we must add KL divergence loss $\mathcal{L}_{kl}$ to ensure that the latent vector z is a Gaussian random variable. Therefore, the training mode of this VAE model is to jointly minimize the KL divergence loss $\mathcal{L}_{kl}$ and the total feature perceptual loss $\mathcal{L}_{rec}^{n}$ of the Deep CNN network, that is,

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{kl} + \beta \sum_{i}^{n} \mathcal{L}_{rec}^{n}$$

where α and β are weighting parameters for KL divergence and feature perceptual loss [78].

3.3.3. Attribute Boosting using VAEs

In order to be able to manipulate attributes such as face attributes and gender, the attribute vector needs to be calculated first, which can be achieved through simple vector arithmetic operations [79] , thereby showing a rich linear structure in the representation space. A well-known example of vector arithmetic operations [80] is vector('King')-vector('Man') + vector('Woman') resulting in a Queen's vector. In this study, we performed a similar arithmetic on the Z representation of our generators for visual concepts. In this paper, we investigate facial attributes smiling. This is done by finding all images where the attribute 'Smiling' is not present and where the same attribute is present. The attribute vector is then calculated as the difference between the two average latent variables. Specifically, the images of two different attributes are sent to the encoder network to calculate the latent vectors, and the average latent vectors is calculated for each attribute respectively, which are represented as $z_{pos\_smiling}$ and $z_{neg\_smiling}$. We can then use the difference $z_{pos\_smiling} - z_{neg\_smiling}$ as the latent vector $z_{smiling}$ of the smiling attribute. Finally, applying this smiling attribute latent vector to different latent vectors z to calculate a new latent vector z, such as $z + \alpha z_{smiling}$,where α= 0,0.1 … 1, and then feeding the new latent vector z to the decoder network generates new face images.

3.4. Verification Network

After generating new faces with specific attributes, we can select the new faces if it can successfully pass the verification network. The verification network also uses the CNN architecture of FaceNet [50] (as shown in Fig.5) . The loss function we use in the face verification model is the triplet loss, that is, we embed *f(x),* from the image x into the feature

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

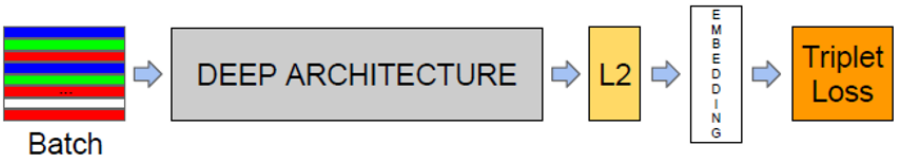

Figure 5. Face Verification Network Architecture. Adapted from [50].

space R^d, and then minimize the squared distance between faces from the same identity and maximize the squared distance between faces from different identities[81] .

The following sections describes 1) Learning face embedding with triplet loss, 2) Triplet selection, and 3) Face verification task.

3.4.1. Learning Face Embedding with Triplet Loss

The idea of face embedding in this section is to embed the image x into a d-dimensional Euclidean space as $f(x) \in R^d$. It should be noted that this embedding is limited to the d-dimensional hypersphere, that is, $||f(x)||_2 = 1$. This loss is caused by nearest neighbor classification[82]. Here, the network is trained to ensure that the output distance between an image x_i^a (anchor) and all other images x_i^p (positive) that belong to a known person is small, and that the distance between an image x_i^a (anchor) and any image x_i^n (negative) that belongs to an unknown person is large(as shown in Fig.6). Thus,

$$||f(x_i^a) - f(x_i^p)||_2^2 + \alpha < ||f(x_i^a) - f(x_i^n)||_2^2, \ \forall((f(x_i^a), f(x_i^p), f(x_i^n) \in T$$

Where the threshold α is a margin that is enforced between positive and negative pairs. T is the set of all possible triplets. Then, the triplet loss is as follows:

$$\mathcal{L} = \sum_i^N [||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha]$$

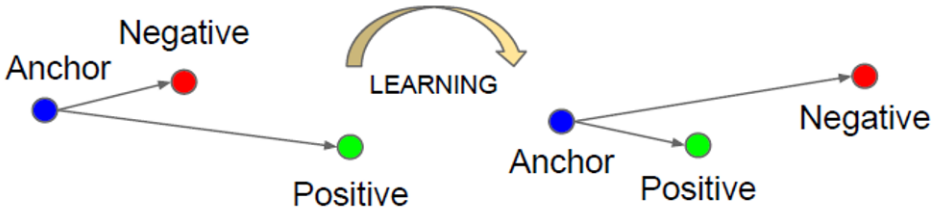

Figure 6. Triplet Loss Overview. Adapted from [50].

3.4.2. Triplet Selection

There are two ways to achieve the selection of triplets as follows:

1. Offline triplet mining: For example, at the beginning of each epoch, we calculate all embeddings on the training set, and then select only hard or semi-hard triples. Then, we can train the epoch with these triplets. However, this technique is not very effective because we need a complete pass to the training set to generate triplets. It also requires regular updates of triplets offline.

2. Online triplet mining: The idea here is to dynamically calculate useful triples for each batch of input. This technique allows you to provide more triples for a single

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

batch of input without any offline mining and is more efficient. Therefore, we focus on the online triplet mining. For example, if we want to generate triplets from these B embeddings, whenever we have three indexes $i, j, k \in [1, B]$, if examples i and j have the same label but different images, and example k has different labels, then we say (i, j, k) is a valid triple.

3.4.3. Face Verification Task

After learning the embedding using triplet loss with FaceNet's CNN, we can utilize this embedding to the FaceNet verification tasks. For example, the input is the paired faces with smiling and unsmiling faces, and the output is the L_2 distance through the verification network between the paired faces. If their L_2 distance is less than 1.1 [50], the pair of images are from the same person, otherwise they are not.

3.5. Identification Network

Finally, we can use these qualified augmented one-shot sets to fine-tune the face identification network with original base set and one-shot set. Similar to the verification network, our face identification network also combines the architecture and training strategy of the FaceNet with the deep convolutional networks and Softmax classifier.

The Softmax function takes a vector z of K real numbers as input and normalizes it to a probability distribution consisting of K probabilities that are proportional to the exponent of the input number. Before the z vector component is input to Softmax, the input z will not be in the interval (0,1). After Softmax is applied, however, each component will be in the interval (0,1), and the sum of the components is 1, so Softmax can convert the input z into probability.

The Softmax function is defined by the formula:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}}$$

for $i = 1,\ldots,K$ and $z = (z_1,\ldots,z_K) \in \mathbb{R}^K$.

4. Experiments and Results

4.1. Datasets

We use the dataset of Labeled Faces in the Wild (LFW) [83] for training face identification network. The LFW dataset contains 13,233 face images with a total of 5,749 identity labelled [84]. Among them, 1,680 people have two or more images. The generative model VAE is trained by CASIA-WebFace dataset [85] , which has a total of 10,575 people and a total of 494,414 images. In the attribute boosting process, the CelebFaces Attributes Dataset (CelebA) [86] is utilized to extract attribute variables vectors. There are about 10k people in total, including 202,599 face images, and each image has 40 binary attribute annotations.

4.2. Implementation Details

4.2.1. Data Preprocessing and Face Alignment

Before the start of each training phase, each image data must be pre-processed to adapt to the face alignment network. We crop the rectangular image into a square with the side length of the short side of the original image. Then, we roughly align the image according to the eye position and adjust the image to 224×224-pixel RGB images.

The Multi-task Cascaded Convolutional Networks (MTCNN) is a face detection and alignment method based on deep convolutional neural networks. This method

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

can be used to complete face detection and alignment tasks at the same time. Pre-processing the images into 224×224-pixel RGB images is required before using the MTCNN. All face images are then detected, five key points are utilized to align the face images, and finally, all images are cropped and uniformly scaled to 160×160-RGB pixels.

4.2.2. Face Generation using VAE

This section describes how to reconstruct face images with boosting attributes by the VAE using FaceNet-based perceptual loss.

a) **Train a VAE:** This section describes how to train a VAE using perceptual loss. The VAE generation model is trained on CASIA-WebFace Dataset. We use a batch size of 128 and 50000 epochs to train the VAE model, and the Adam method used to optimize [87] the initial learning rate is 0.0002. The Inception ResNet CNN is used as the loss network ϕ to calculate the feature perception loss for image reconstruction. The size of the generated images is decided by the VAE implementation that generates 64x64-pixel images. The hardware specifications for executing implementations use Tesla P100 GPU with 25 GB RAM. Table 1 shows some values of hyper-parameters which are used in this experiment.

Table 1. Hyper-parameters used in all experiments

HYPER-PARAMETERS	VALUES
EPOCHS	5000
BATCH-SIZE	128
LEARNING RATE	0.01
OPTIMIZER	Nadam
DATASET	LFW

b) Calculate Attribute Vectors: In this step, the CelebA dataset is used to calculate vectors in latent variable space for several attributes. The CelebA dataset contains ~200k images annotated with 40 different attributes such as Blond Hair and Mustache. This is done by finding all images where the attribute is not present and where the same attribute is present. The attribute vector is then calculated as the different between the two average latent variables. The VAE model checkpoint should point to the checkpoint trained in Step 1. Before running this, the CelebA dataset should be aligned as well. The list_attr_celeba.txt file of CelebA dataset contains the 40 attributes for each image and is available for download together with the dataset itself.

c)Attribute Boosting Using VAEs: To demonstrate the usage of the VAE, after which we can modify attributes of an image, we apply the attribute-specific vector (calculated in step b) to different latent vectors z to calculate a new latent vector z, such as $z + \alpha z_{attribute}$, where $\alpha = 0,0.1 \ldots 1$, and then feed the new latent vector z to the decoder network to generate new face images. Specifically, we select a few faces in one-shot set where the specific attribute is not present. We then feed these images to the encoder of VAE, and then calculate the latent variables. We then add different amounts of the specific attribute vector (calculated in Step b) to different latent variables z and generate new faces with specific attribute images.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

Figure 7. Face reconstruction by VAE with feature perception loss. Top row: Input images. Bottom row: Generated images from VAE with feature perceptual loss.

4.2.3. Face Verification and Face Identification Experiments

The face verification function under the deepface interface offers to verify face pairs as same person or different persons. This is a pre-trained network so there is no need to retrain. Table 1 shows some values of hyper-parameters to train the softmax classifier for Face Identification Network which are used in all experiments. For comparison purpose, some parameters for all experiments have been set to the same values to perform fair comparison.

4.3. Results

4.3.1. Quality of The Reconstruction

In this research, we first use VAE based on feature perception loss to reconstruct face images. The Fig.7 shows the result of the reconstruction. Top row: Input images. Bottom row: Generated images from VAE with feature perceptual loss. As shown in the Figure 7, we can observe the difference between the face reconstructed by the VAE based on the feature perception loss and the original input face: the VAE based on the feature perception loss can not only generate human-like faces and the reconstructed face is similar to the original input face, but it can preserve the overall spatial face structure. We know that ordinary VAE is difficult to generate clear facial parts, such as eyes, nose, and mouth. This is because ordinary VAE tries to minimize the pixel-by-pixel loss between two images, while pixel-based loss does not contain perceptual and spatially related information. However, VAE based on the loss of feature perception can generate clear facial parts, such as eyes, nose, and mouth. This point of view is also confirmed in our experiments.

4.3.2. Pose Transition

In this experiment, we also tried to augment data through modifying images by rotating, flipping, adding noise, and jittering color. Our model includes these conventional schemes to provide prevention to overfitting that can occur within an interclass , increase robustness, and to achieve higher scores. The effectiveness of classical data augmentation is empirically shown in several previous studies. As shown in the Fig 8., the first row is the original face, the second row is the augment data by flipping , the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

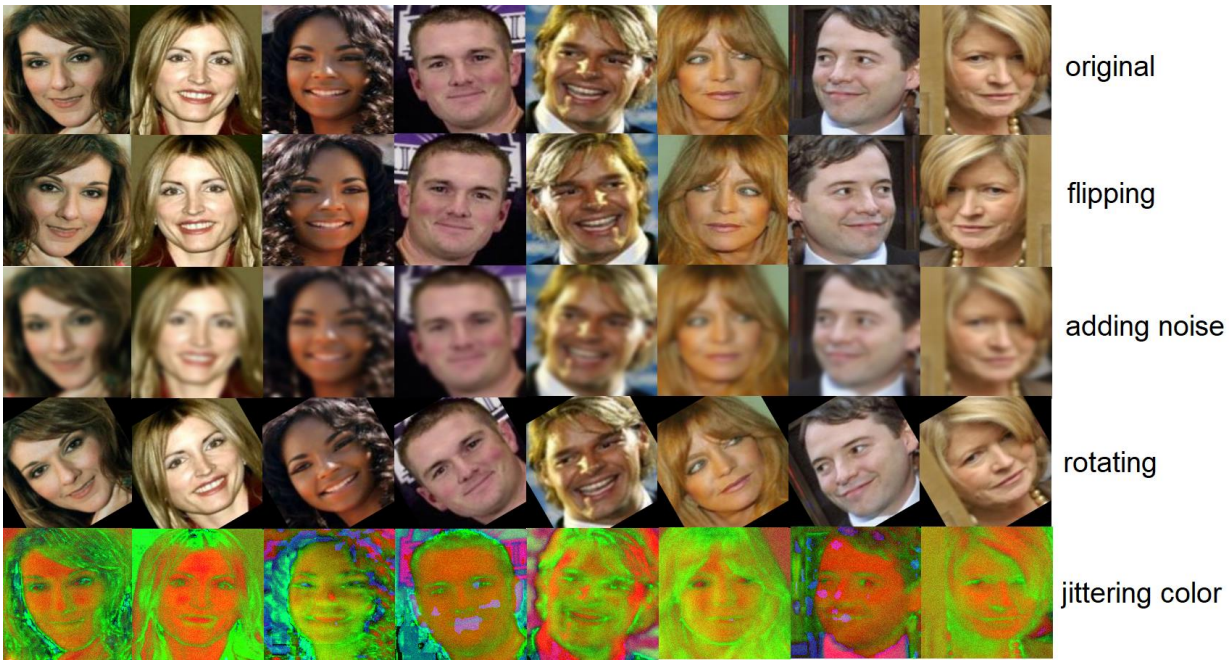

Figure 8. Face augment data through modifying images by rotating, flipping, adding noise, and jittering color

third and the fourth rows are the results of - adding noise and rotating, respectively, and the last row shows augment data by jittering color. In this way, each person has 4 different pose transition, so in this step of the experiment, each person will generate a total of 4 new pictures by basic image processing.

4.3.3. Attribute Manipulation

In this experiment, we are not seeking to manipulate the overall face image, but we want to control the specific attributes of the face image and generate a face image with specific attributes. By adding the vector of attributes to the face in the latent variable space, we can get the smooth transition process of adding this attribute to the face. As shown in the Fig.9 (first row), by adding a smiling vector to the latent vector of non-smiling women, we can get a smooth transition from non-smiling face to smiling face. When the factor α increases, the appearance of the smile becomes more obvious, while other facial attributes can remain unchanged. Similarly, the second row is the transition process of adding a sunglasses vector from left to right, the third line and the fourth line are the results of adding a goatee vector and a heavy makeup vector , respectively, and the last line shows adding an attractive vector. In this way, each person has 40 different attributes, and each attribute will generate 10 pictures, so in this step of the experiment, each person will generate a total of 400 new pictures by VAE.

4.3.4. Correlation Between Attribute-specific Vectors

There are often correlations between different facial attributes. For example, heavy makeup and lipstick are often related to women. In order to study the correlation between different facial attributes in the CelebA dataset, we selected 15 facial

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

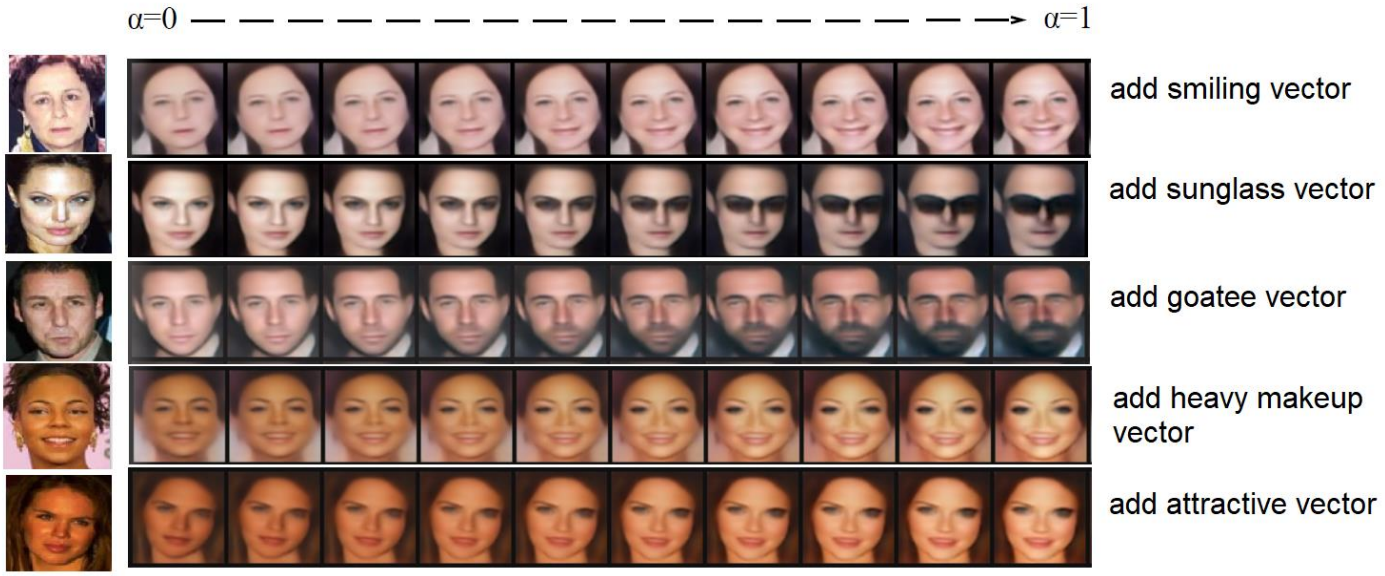

Figure 9. The Vector arithmetic for visual attributes.

attributes and calculated their attribute-specific latent vectors respectively. After that, we used Pearson correlation to calculate the correlation matrix of these 15 attributes-specific vectors. Figure 10 shows the weight visualization of the learned correlation matrix. Red indicates a positive correlation, blue indicates a negative correlation, and the intensity of the color indicates the strength of the correlation. From the visualization results, we can find many related attribute pairs and many mutually exclusive attribute pairs. For example, the attributes of arched eyebrows and heavy makeup are given relatively high weights, indicate a positive correlation between these two attributes. And the attributes of arched eyebrows and male are given relatively low weights, indicate a negative correlation between these two attributes. It makes sense that women are generally considered using more cosmetics than men. In addition, wearing necklace seems to have no correlation with most other attributes, and only has a weak positive correlation with wearing necklace, wearing lipstick, and wearing earring. This can also be explained well that wearing necklace, lipstick and earring are all facial decorations that often appear together.

4.3.5. Data Verification

Faces generated by deep generative models may occasionally fail to be recognized. For example, certain attributes (such as blond hair, young) added to the faces in the LFW data set will have a serious decrease in the face recognition rate. However, some attributes (such as pale skin, smiling) added to the faces in the LFW data set can increase the face recognition rate. Therefore, to filter out unqualified augmented images and increase the face recognition rate, we use the verification network to verify the augmented data from the generation network, instead of manually selecting the appropriate attributes for the generation results. Table 2 shows the verification success rate of the generated images in the 1-shot experiment. At the same time, we also carried out the verification success rate of basic image processing data. It can be seen from the results that the verification rate of basic image processing data is very high, further confirming that the use of basic image processing augmented data can enhance the robustness of our model, and to achieve higher scores. After this step, we can construct the final combined few-shot data set: the novel image processing

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

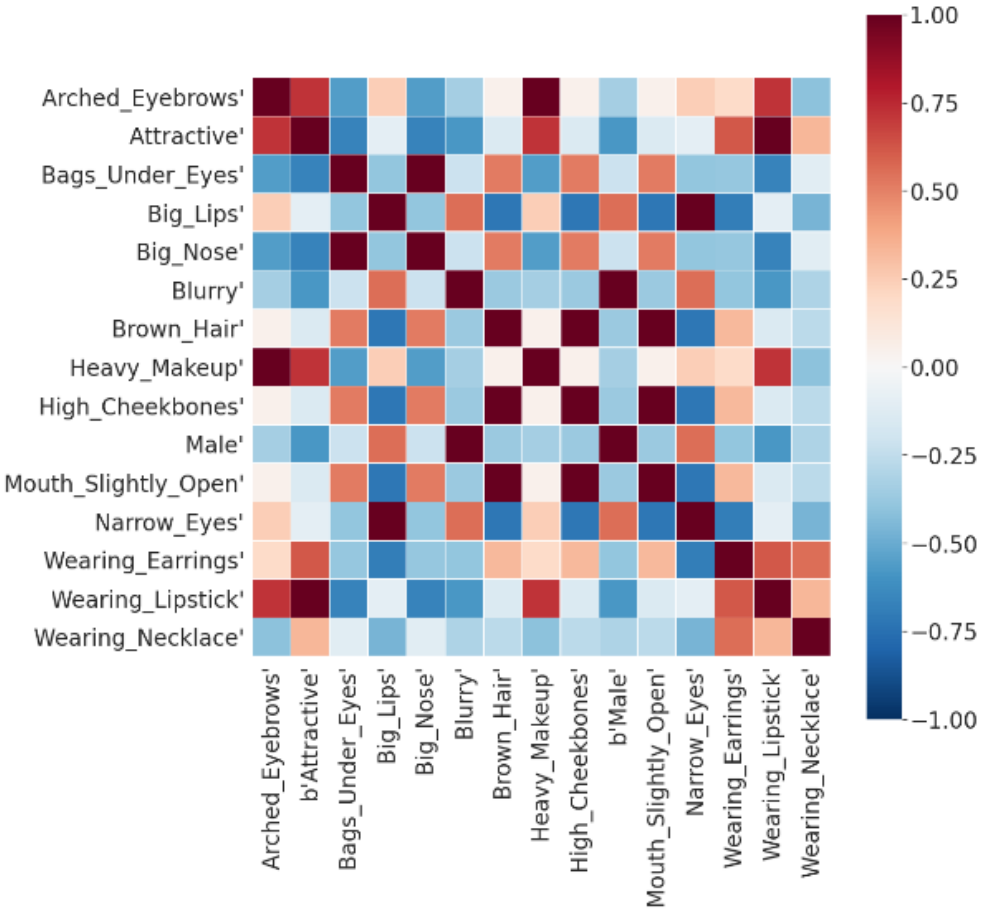

Figure 10. Pearson Correlations between specific facial attributes

augmented data set, the novel verified attribute augmented data set, and the original basic data set.

Table 2. Data verification results

	Verification rate (%)
Attribute boosting	10
Basic image processing	98.3

4.3.6. Data Identification

Our final face identification network is pre-trained by using the basic set, and then fine-tuned using the novel image processing augmented data set and the novel verified attribute augmented data set, so that individuals in the no overlap one-shot test set can be recognized. For evaluation, we measure identification accuracy. That is, suppose there are N images available in the test set, and C images are correctly recognized. Then, the accuracy is defined as C/N. We propose two methods of data augmentation and a combination of these two methods: 1) basic image processing 2) attribute boosting using VAE 3) the combination of the above two methods. In these

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

Table 3. Face identification results.

Training method	Test set	Identification Accuracy(%)
basic one-shot set	no overlap one-shot set	88.52
basic one-shot set + verified VAE attribute augmented set	no overlap one-shot set	92.99
basic one-shot set + image processing augmented set	no overlap one-shot set	93.67
basic one-shot set + image processing augmented set + verified VAE attribute augmented set	no overlap one-shot set	96.47

three methods, we gradually add new data to study the changes in identification accuracy.

The results of face identification are shown in Table 3. From these experiments, we have observed that before data augmentation, the basic one-shot set is used to train the face identification network and the no overlap one-shot test set is used for testing. The final face identification accuracy rate is 88.52%. This shows that there is still a lot of room for improvement in the accuracy of one-shot face identification. The best accuracy of one-shot face identification using VAE attribute boosting augmentation is 92.99%, indicating that although using VAE attribute boosting augmentation can improve the performance of one-shot face identification, it still lacks robustness and causes overfitting that can occur within an interclass. On the other hand, the best accuracy of one-shot face identification using basic image processing is 93.67%.This proves that basic image processing methods can and prevent possible over-fitting between classes. In many applications, they can be combined to improve the performance of the model[88-90]. Therefore, our model can utilize these basic image processing methods to provide prevention to overfitting that can occur within an interclass, increase robustness, and to achieve higher scores. We have observed that the combination of basic image processing and attribute boosting using VAE is more effective in improving performance than using classic basic image processing and VAE attribute boosting respectively. Please note that the identification accuracy has increased from 92.99% and 93.67% to 96.47%, respectively. This result shows that the use of VAE to produce more diverse training faces that certainly makes up for the insufficiency of intra-class variation. Table 4 shows that when the L_2 distance of the verification network is set to less than 1.2 and 1.1 respectively, the results will be sorted by L_2 distance from small to large, and then we pick top1 to top5 faces and add them to the training set, and observe the results of the effect of the added faces on the identification accuracy of no overlap one-shot set. From the results, we can observe that no matter whether the distance of L_2 is set to less than 1.2 or 1.1, the identification accuracy is always higher than the result before the data augmentation ,which is 88.52%. This proves that our verified VAE attribute augmented set has improved the accuracy of identifying one-shot set. In addition, although the identification accuracy is higher when the L_2 distance is set to 1.2 than set to 1.1, but after the image processing augmented set is added, the accuracy of the L_2 distance set to 1.1 is higher than set to 1.2. This shows that setting the L_2 distance to 1.1 makes the combination of the verified VAE attribute augmented set and image processing augmented set more

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 September 2021 doi:10.20944/preprints202109.0389.v1

Table 4. Face identification results with different L_2 distances.

Training method	Test set	Identification Accuracy(%)				
		Top1-shot	Top2-shot	Top3-shot	Top4-shot	Top5-shot
basic one-shot set + verified VAE attribute augmented set with 1.1 L_2 distance	no overlap one-shot set	89.60	90.36	**90.69**	90.36	90.27
basic one-shot set + image processing augmented set+ verified VAE attribute augmented set with 1.1 L_2 distance	no overlap one-shot set	95.70	95.52	95.67	**96.30**	95.36
basic one-shot set + verified VAE attribute augmented set with 1.2 L_2 distance	no overlap one-shot set	90.66	**90.99**	90.83	90.40	90.87
basic one-shot set + image processing augmented set+ verified VAE attribute augmented set with 1.2 L_2 distance	no overlap one-shot set	**95.83**	95.30	90.73	95.49	95.43

effective, thereby improving the robustness and generalization ability of the face identification model.

5. Conclusions

At present, few-shot learning technology based on image generation is very attractive in various computer vision applications, especially because of the lack of labeled data during training. In this research, we try to use VAE with feature perception loss to generate better visual quality face images with boosting attributes and expand the training set to improve the performance of few-shot face identification task. As the data set increased, we verified the performance of increasing the use of Inception Resnet V1 for few-shot face recognition. Since the generated data points at the boundary between the different classes are very easy to be misclassified. Therefore, in the future, we plan to determine decision boundaries using adversarial examples between the different classes to further improve the few-shot face identification accuracy.

Author Contributions: This research is part of the R.W. dissertation work under the supervision of A.M. All authors conceived and designed the experiments. R.W. performed the experiments. Formal analysis, R.W. Investigation, R.W. Methodology, R.W. Software, R.W. Supervision, R.W. Writing—original draft, Ruoqi Wei. Writing—review & editing, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the UB Partners CT Next Innovation Grant 2019–2020.Also, this research work was funded in part by the Department of Computer Science and Engineering, University of Bridgeport, CT, USA

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://vis-www.cs.umass.edu/lfw/

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep learning*; MIT press: 2016.
2. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436-444.
3. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience* **2018**, *2018*.
4. Ouyang, W.; Zeng, X.; Wang, X.; Qiu, S.; Luo, P.; Tian, Y.; Li, H.; Yang, S.; Wang, Z.; Li, H. DeepID-Net: Object detection with deformable part based convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2016**, *39*, 1320-1334.
5. Diba, A.; Sharma, V.; Pazandeh, A.; Pirsiavash, H.; Van Gool, L. Weakly supervised cascaded convolutional networks. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp. 914-922.
6. Doulamis, N.; Voulodimos, A. FAST-MDL: Fast Adaptive Supervised Training of multi-layered deep learning models for consistent object tracking and classification. In Proceedings of the 2016 IEEE International Conference on Imaging Systems and Techniques (IST), 2016; pp. 318-323.
7. Doulamis, N. Adaptable deep learning structures for object labeling/tracking under dynamic visual environments. *Multimedia Tools and Applications* **2018**, *77*, 9651-9689.
8. Lin, L.; Wang, K.; Zuo, W.; Wang, M.; Luo, J.; Zhang, L. A deep structured model with radius–margin bound for 3D human activity recognition. *International Journal of Computer Vision* **2016**, *118*, 256-273.
9. Cao, S.; Nevatia, R. Exploring deep learning based solutions in fine grained activity recognition in the wild. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), 2016; pp. 384-389.
10. Toshev, A.; Szegedy, C. Deeppose: Human pose estimation via deep neural networks. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2014; pp. 1653-1660.
11. Chen, X.; Yuille, A.L. Articulated pose estimation by a graphical model with image dependent pairwise relations. In Proceedings of the Advances in neural information processing systems, 2014; pp. 1736-1744.
12. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the Proceedings of the IEEE international conference on computer vision, 2015; pp. 1520-1528.
13. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2015; pp. 3431-3440.
14. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *International journal of computer vision* **2015**, *115*, 211-252.
15. Jin, A.; Yeung, S.; Jopling, J.; Krause, J.; Azagury, D.; Milstein, A.; Fei-Fei, L. Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018; pp. 691-699.
16. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical image computing and computer-assisted intervention, 2015; pp. 234-241.
17. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)* **2020**, *53*, 1-34.
18. Shu, J.; Xu, Z.; Meng, D. Small sample learning in big data era. *arXiv preprint arXiv:1808.04572* **2018**.

19. Lu, J.; Gong, P.; Ye, J.; Zhang, C. Learning from Very Few Samples: A Survey. *arXiv preprint arXiv:2009.02653* **2020**.
20. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *Journal of Big Data* **2019**, *6*, 60.
21. Mi, L.; Shen, M.; Zhang, J. A Probe Towards Understanding GAN and VAE Models. *arXiv preprint arXiv:1812.05676* **2018**.
22. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* **2015**.
23. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *science* **2006**, *313*, 504-507.
24. Salakhutdinov, R.; Hinton, G. Deep boltzmann machines. In Proceedings of the Artificial intelligence and statistics, 2009; pp. 448-455.
25. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* **2013**.
26. Doersch, C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* **2016**.
27. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in neural information processing systems, 2014; pp. 2672-2680.
28. Goodfellow, I. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* **2016**.
29. Wei, R.; Garcia, C.; El-Sayed, A.; Peterson, V.; Mahmood, A. Variations in Variational Autoencoders-A Comparative Evaluation. *IEEE Access* **2020**, *8*, 153651-153670.
30. Wei, R.; Mahmood, A. Recent Advances in Variational Autoen-coders with Representation Learning for Biomedical Informatics: A Survey. *IEEE Access* **2020**.
31. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **2013**, *35*, 1798-1828.
32. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *Iclr* **2017**, *2*, 6.
33. Zhao, S.; Song, J.; Ermon, S. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262* **2017**.
34. Dilokthanakul, N.; Mediano, P.A.M.; Garnelo, M.; Lee, M.C.H.; Salimbeni, H.; Arulkumaran, K.; Shanahan, M. Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. **2016**.
35. Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; Zhou, H. Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering. **2016**.
36. Xian, Y.; Sharma, S.; Schiele, B.; Akata, Z. F-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning. **2019**, 10267-10276, doi:10.1109/CVPR.2019.01052.
37. Gao, R.; Hou, X.; Qin, J.; Chen, J.; Liu, L.; Zhu, F.; Zhang, Z.; Shao, L. Zero-VAE-GAN: Generating Unseen Features for Generalized and Transductive Zero-Shot Learning. *IEEE Transactions on Image Processing* **2020**, *29*, 3665-3680.
38. Davidson, T.R.; Falorsi, L.; De Cao, N.; Kipf, T.; Tomczak, J.M. Hyperspherical Variational Auto-Encoders. **2018**.
39. Oord, A.v.d.; Vinyals, O.; Kavukcuoglu, K. Neural Discrete Representation Learning. **2017**.
40. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. **2015**.
41. Zhu, Y.; Min, M.R.; Kadav, A.; Graf, H.P. S3VAE: Self-Supervised Sequential VAE for Representation Disentanglement and Data Generation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020; pp. 6538-6547.
42. Walker, J.; Doersch, C.; Gupta, A.; Hebert, M. An uncertain future: Forecasting from static images using variational autoencoders. In Proceedings of the European Conference on Computer Vision, 2016; pp. 835-851.

43. Mishra, A.; Krishna Reddy, S.; Mittal, A.; Murthy, H.A. A generative model for zero shot learning using conditional variational autoencoders. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018; pp. 2188-2196.
44. Schonfeld, E.; Ebrahimi, S.; Sinha, S.; Darrell, T.; Akata, Z. Generalized zero-and few-shot learning via aligned variational autoencoders. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019; pp. 8247-8255.
45. Wang, W.; Pu, Y.; Verma, V.K.; Fan, K.; Zhang, Y.; Chen, C.; Rai, P.; Carin, L. Zero-shot learning via class-conditioned deep generative models. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
46. Sønderby, C.K.; Caballero, J.; Theis, L.; Shi, W.; Huszár, F. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490* **2016**.
47. Bruna, J.; Sprechmann, P.; LeCun, Y. Super-resolution with deep convolutional sufficient statistics. *arXiv preprint arXiv:1511.05666* **2015**.
48. Yeh, R.A.; Chen, C.; Yian Lim, T.; Schwing, A.G.; Hasegawa-Johnson, M.; Do, M.N. Semantic image inpainting with deep generative models. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp. 5485-5493.
49. Xu, J.; Teh, Y.W. Controllable semantic image inpainting. *arXiv preprint arXiv:1806.05953* **2018**.
50. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2015; pp. 815-823.
51. Masi, I.; Wu, Y.; Hassner, T.; Natarajan, P. Deep face recognition: A survey. In Proceedings of the 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), 2018; pp. 471-478.
52. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261* **2016**.
53. Kulis, B. Metric learning: A survey. *Foundations and trends in machine learning* **2012**, *5*, 287-364.
54. Bellet, A.; Habrard, A.; Sebban, M. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709* **2013**.
55. Vanschoren, J. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548* **2018**.
56. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the International conference on artificial neural networks, 2018; pp. 270-279.
57. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *Journal of Big data* **2016**, *3*, 9.
58. Wang, X.; Wang, K.; Lian, S. A survey on face data augmentation. *arXiv preprint arXiv:1904.11685* **2019**.
59. Ratner, A.J.; Ehrenberg, H.; Hussain, Z.; Dunnmon, J.; Ré, C. Learning to compose domain-specific transformations for data augmentation. In Proceedings of the Advances in neural information processing systems, 2017; pp. 3236-3246.
60. Wang, X.; Wang, K.; Lian, S. A survey on face data augmentation for the training of deep neural networks. *Neural Computing and Applications* **2020**, 1, doi:10.1007/s00521-020-04748-3.
61. Hartig, S.M. Basic image analysis and manipulation in ImageJ. *Current protocols in molecular biology* **2013**, *102*, 14.15. 11-14.15. 12.
62. Pratt, W.K. *Introduction to digital image processing*; CRC press: 2013.
63. Bartoli, A. Groupwise geometric and photometric direct image registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2008**, *30*, 2098-2108.
64. Holden, M. A review of geometric transformations for nonrigid body registration. *IEEE transactions on medical imaging* **2007**, *27*, 111-128.

65. Cootes, T.F.; Edwards, G.J.; Taylor, C.J. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2001**, *23*, 681-685, doi:10.1109/34.927467.
66. Volker, B.; Thomas, V. A morphable model for the synthesis of 3D faces. **1999**, 187-194, doi:10.1145/311535.311556.
67. Grover, A.; Dhar, M.; Ermon, S. Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models. **2017**.
68. Zhou, W.; Bovik, A.C. A universal image quality index. *IEEE Signal Processing Letters, Signal Processing Letters, IEEE, IEEE Signal Process. Lett.* **2002**, *9*, 81-84, doi:10.1109/97.995823.
69. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* **2010**, *22*, 1345-1359, doi:10.1109/TKDE.2009.191.
70. Zheng, X.; Guo, Y.; Huang, H.; Li, Y.; He, R. A Survey of Deep Facial Attribute Analysis. *International Journal of Computer Vision* **2020**, *128*, 2002, doi:10.1007/s11263-020-01308-z.
71. Kim, H.; Mnih, A. Disentangling by factorising. In Proceedings of the International Conference on Machine Learning, 2018; pp. 2649-2658.
72. Upchurch, P.; Gardner, J.; Pleiss, G.; Pless, R.; Snavely, N.; Bala, K.; Weinberger, K. Deep Feature Interpolation for Image Content Changes. **2016**.
73. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. **2014**.
74. Hou, X.; Shen, L.; Sun, K.; Qiu, G. Deep Feature Consistent Variational Autoencoder. **2017**, 1133-1141, doi:10.1109/WACV.2017.131.
75. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in neural information processing systems, 2012; pp. 1097-1105.
76. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. **2014**.
77. Hou, X.; Shen, L.; Sun, K.; Qiu, G. Deep feature consistent variational autoencoder. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017; pp. 1133-1141.
78. Gatys, L.A.; Ecker, A.S.; Bethge, M. A Neural Algorithm of Artistic Style. **2015**.
79. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. **2013**.
80. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. **2015**.
81. Sun, Y.; Wang, X.; Tang, X. Deep Learning Face Representation by Joint Identification-Verification. **2014**.
82. Weinberger, K.Q.; Saul, L.K. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research* **2009**, *10*.
83. Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments.
84. Erik, L.-M.; Gary, B.H.; Aruni, R.; Haoxiang, L.; Gang, H. Labeled Faces in the Wild: A Survey. **2015**, 189, doi:10.1007/978-3-319-25958-1_8

10.1007/978-3-319-25958-1.

85. Yi, D.; Lei, Z.; Liao, S.; Li, S.Z. Learning Face Representation from Scratch. **2014**.
86. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. **2014**.
87. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. **2014**.
88. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **2012**, *25*, 1097-1105.

89. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.

90. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; pp. 770-778.