

Applying Aging Effect on Facial Image with Multi-domain Generative Adversarial Network

Shuvendu Roy

Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Bangladesh
Email: bikash.shuvendu@gmail.com

Received: 06 August 2019; Accepted: 25 September 2019; Published: 08 December 2019

Abstract—Face Aging is an important and challenging application in computer vision. This is an application of conditional image generation. Until recently generative model was not good enough to generate considerable good resolution images. A generative model called generative adversarial network has introduced impressive capabilities in generating realistic images in both unconditional and conditional settings. Still, the task of generating images of different age conditioning on a given image is a very challenging task. Because there are two constraints to satisfy here in the generated images. The generated image must preserve the identity of the person in the source image and the image must have the features of the target age. In this work, we have applied the generative adversarial network in conditional settings along with custom loss function to satisfy the two mentioned constraints. The experiment has shown improved performance both in preserving the person's identity and classification accuracy of generated images in the target class compared to previous known approach to this problem.

Index Terms—Face-Aging, GAN, CNN, Generative-Model, Face-Synthesis

I. INTRODUCTION

Aging is a natural process which changes the appearance of a person over time. Although some facial feature preserves during the aging process, sometimes it is hard to recognize a person at his later age just only seeing him in his earlier age. In many cases, it becomes very important to recognize someone in their later age while images of his earlier age are available. For example, identifying a criminal who has been hiding for a long time. In some cases, reverse aging is also required. This work presents a novel method for generating the facial image of someone in their different ages. The total age range is divided into 6 age group. Fig. 1 shows a couple of examples of the generated images of a given person in different ages.

There is a lot of real-world application for this task. It is useful for forensic application. It can be used to enhance or modify a photograph of a suspect or victim or a lost person to identify his identity. This technique was used by police and investigators, where this modification

was done manually by the help of human artist, computer graphics, psychology and many more. This long and complex process can be simplified by such an algorithm that can do the aging.



Fig. 1. Generated images at different age of a given person with the proposed model. The first column shows the input image, rest are generated image of the corresponding age group.

Another important application is in entertainment. Aging is part of special virtual effects in film and magazine. The actor's approaches are changed with an extremely realistic way from aged to young and reversed. This is done with age synthesis process manually, frame by frame. While converting one image is hard this process makes it repetitive and very time-consuming. So, making the computer able to automatically convert the characters of the film with an aging effect will make the process much easier.

Face aging was also referred to as age synthesis [1] and age progression [2]. It is the base of cross-age face detection. All previous approaches to this topic can be divided into two major categories—prototyping approaches and modeling approach. Prototyping approaches [3], [4] tries to estimate the average face of a certain age group. The difference between these faces gives the age pattern of that age group which is then used to generate the aged face of a given input. Such a method does not put any constraint to keep the identity of the original person. For that reason, the generated images look unrealistic. These rules-based methods are fast but not very effective.

Whereas the modeling approach [5], [6] concentrate on the skin, muscle and skull level feature of different age group. These models are supervised and require images of the same person in a different age, which are costly to collect.

Recent advancement in the neural network has opened a much interesting research direction in computer vision. Using the neural network to generate natural-looking image has been studied for the last few years but until

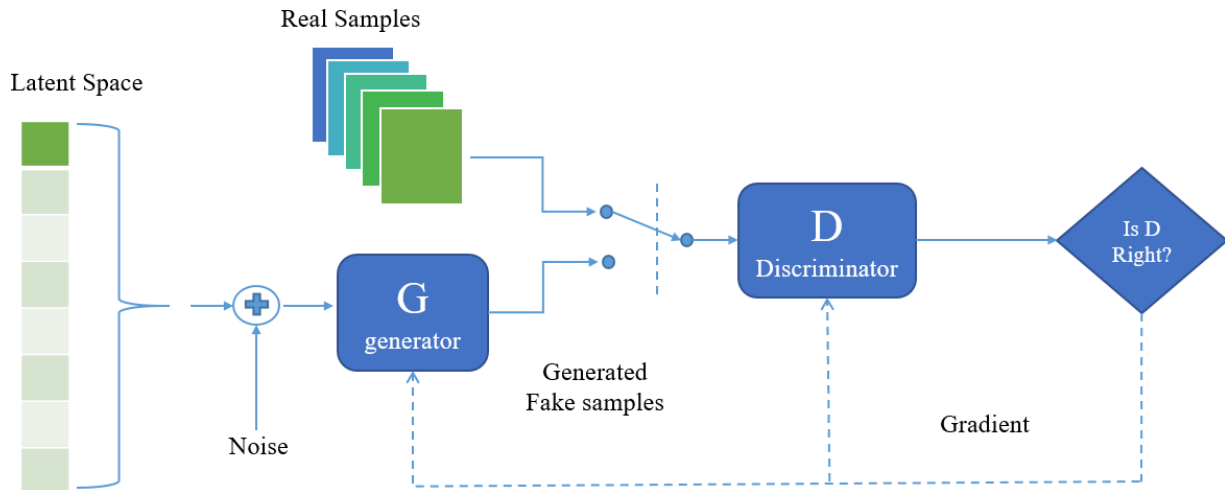


Fig. 2. Training a generative adversarial network. The generator takes a vector as input and generates an image (Generated Fake samples). This image is passed to discriminator, which tells whether the image is real or fake.

recently it was not possible to generate good resolution images with a neural network. The key model that made it possible is the generative adversarial network (GAN) [7] which unlike other generative models (eg. variational autoencoder [8]) does not use L2 loss.

The L2 loss has a tendency of generating blurry images. Instead, GAN focuses on generating realistic images with an explicit loss for that. And that helped to generate realistic images with fine details. The earlier GAN was capable of generating random images from the data distribution of the trained images. Later, conditional generative adversarial network (cGAN) [9] brought control over the generated image. It takes a class label as input to generate images of that class. Isola et. al. [10] introduced encoder-decoder architecture to condition on a given image, not just the class. This approach is successfully used in various facial attribute editing task like changing hair color or adding sunglasses on a given image.

The objective of this research is to propose a model that can take a human image of any age and generate his predicted image at different specified age. In this work, we have used the concept GAN in conditional settings for facial aging. Our proposed method generates the facial image considering two constraints. First, the proposed method has a single generator for generating faces of different age groups. Second, it preserves the identity of the source subject while converting to the desired age. We have proposed a custom loss function along with GAN loss to import these two constraints. The experiment shows improvement in generating better images with the identity of the source person preserved.

II. METHODS

We first describe the concept of the generative adversarial network which is the backbone of our model. Then we present our model for generating the facial image in a different age of a person. Then we discuss the effect of the different components of our objective function. Finally, we describe the training procedures of the model.

A. Generative Adversarial Network

Generative adversarial network (GAN) [7] is one of the most effective and popular generative models currently available. The GAN has two components: Generator and discriminator. The GAN generator is trained to match the data distribution of the generated image to that of the target image distribution. This is done with a two-player min-max game between the generator and the discriminator. The discriminator is trained to learn the distribution of real data and distinguish it from that of generated data. At the same time, the generator is trained to generate more realistic images to fool the discriminator. The training improves the performance of both the generator and the discriminator.

In unconditional settings, GAN tries to generate a random realistic image to match the distribution of the target domain. Let z be a random variable drawn from the normal random distribution R_d , where d is the dimension of input to the generator. The generator takes the random variable z as input and generates an image of target domain $G(z)$. The objective of the discriminator is to output $D(x) = 1$ for $x \in P_x$ and $D(G(z)) = 0$. The objective of the GAN model is presented in the Equation 1, where the generator tries to minimize the loss value and the discriminator tries to maximize it.

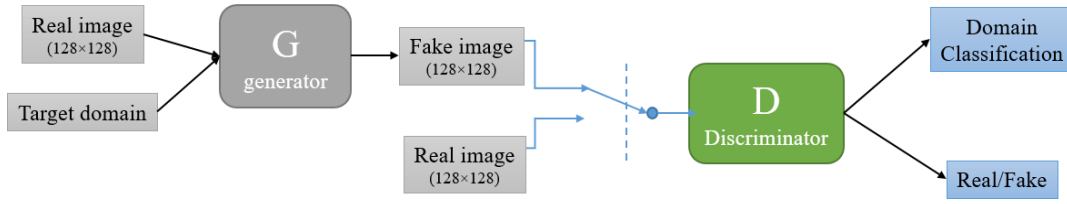


Fig. 3. Training the discriminator. The gray part of the diagram is frozen during the training of discriminator. Either real image of the fake image is send as input to the discriminator at a time.

$$\min_G \max_D V(G, D) = E_{x \sim p_x} [\log D(x)] + E_{z \sim p_z} [\log (1 - D(G(z)))] \quad (1)$$

A visual description of GAN training is given in Fig. 2. GAN can also be used in conditional settings with extra input to the generator. The conditional generative adversarial network (cGAN) was introduced by Mirza et. al. [9]. GAN can be used in the conditional setting by providing extra information like target class or attribute as input. The GAN objective in the conditional settings can be represented in Eq. 2.

$$\min_G \max_D V(G, D) = E_{x \sim p_x} [\log D(x)] + E_{z \sim p_z} [\log (1 - D(G(z|y)))] \quad (2)$$

Where y is the conditional input.

It popularized a new dimension in the generative model called the image to image translation. These models use an input image as a condition to generate a target image. First, the image-to-image translation network generating a good resolution image was proposed by Isola et. al. [10]. Because of the fact that an image is used to condition on, the generator of such model uses encoder-decoder architecture.

This model has two parts; the encoder takes the image as input and compresses it into a lower-dimensional matrix which is the input to the decoder. The decoder upsamples it to generate the target image. Sometimes the decoder takes a random vector as input along with the encoder matrix. The purpose of the random vector is to introduce randomness in the generated image.

The original image-to-image translation model proposed a supervised method for training conditional image generator. The model consisted of two losses. The regular GAN loss was responsible for generating realistic-looking images. And an L1 loss was considered between the generated image and the target image reduce the difference between the generated and target image. Given an image x and a random variable z , it learns to generate the image of the target domain, $G: \{x, z\} \rightarrow y$.

$$L_{cGAN} = E_y [\log D(y)] + E_{x,z} [\log (1 - D(G(x, z)))] \quad (3a)$$

$$L_{L1}(G) = E_{x,y,z} [\|y - G(x, z)\|_1] \quad (3b)$$

In the final objective, both losses is weighted with a weighting factor λ , which balance the importance of both losses.

$$\min \max V(G, D) = L_{cGAN}(G, D) + \lambda L_{L1}(G) \quad (4)$$

In real life, most of the image translation task does not have paired data for supervised training. Zhu et. al. [11] proposed a cycle-consistent adversarial network that can transfer images between two domains with unsupervised training. It had two copies to each network to convert from and to both domains. Now it can convert from source to target and back to the source where L1 loss can be applied. It does not require paired supervised data to train with this objective. If F is the generator from domain Y to X and G is the generator from domain X to Y . The cycle consistency loss can be represented as follow.

$$L_{cyc}(G, F) = E_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + E_{x \sim p_{data}(x)} [\|F(G(y)) - y\|_1] \quad (5)$$

This approach has a major disadvantage. It cannot be extended with more than two domain efficiently. Even having two different generators and discriminator is costly to train. We have to deal with 6 different domain for different age groups. Also applying the two important constraints for preserving the person's identity and the age-specific feature is not possible here. So, we adopted a single generator model with the class as a condition to generate the images of the target domain and added two custom objective to apply the mentioned constraint.

B. Face-Aging Model

Our objective is to train a multi-domain image-to-image translation model. Along with the adversarial loss, we applied the custom loss to make the image looks like the target domain and keep the identity of the person of the source. The generator takes the class of the target image c , along with the input image x to generate the image; $G(x, c) \rightarrow y$. The model generates images of all classes from a given image.

The discriminator acts as an auxiliary classifier [12] so that one discriminator can handle images on multiple domains. For that, the discriminator has two outputs-class probabilities and discriminator validity. Like regular

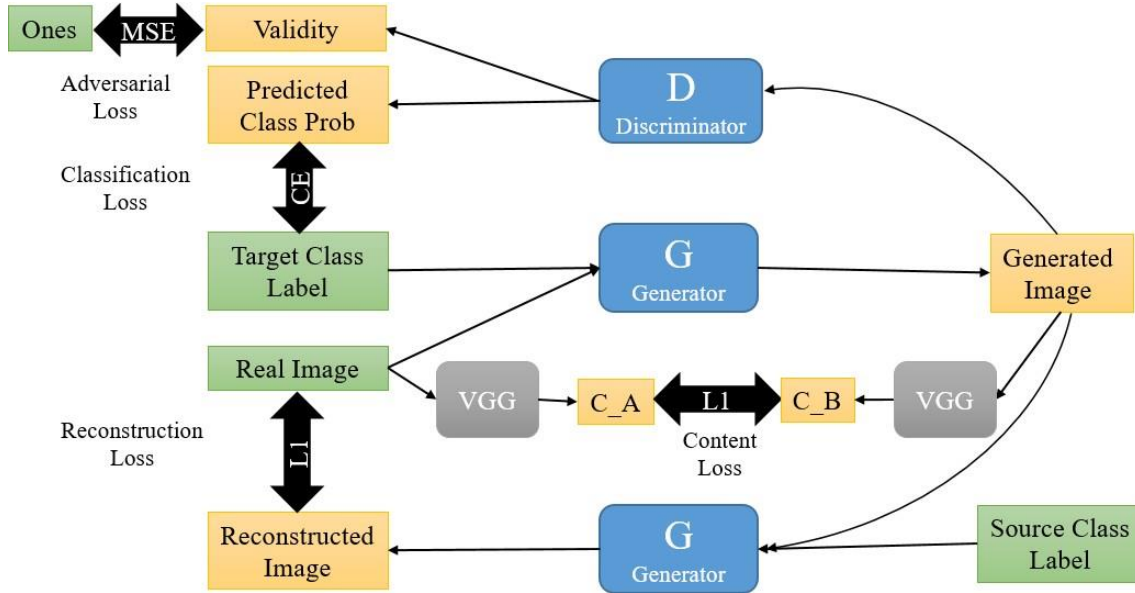


Fig. 4. Training the generator. Here, green represents the inputs to the model. Blue is the trainable model and black is trained frozen model. Yellow are the output of the model.

GAN the validity represents the probability score of a generated image as real/fake, predicted by the discriminator and the class probability is the probability of the generated image belonging to a specific domain. So, the adversarial loss of the proposed model can be represented as Eq. 6

$$L_{adv} = E_x[\log D_{adv}(x)] + E_{x,c}[\log(1 - D_{adv}(G(x, c)))] \quad (6)$$

Domain Classification Loss. The objective of our model is to generate an image of a target domain that is classified as an image of that domain. For that, we have adopted a classification loss and implemented with a second output of the discriminator. During the discriminator training, the classifier is trained to classify the images properly. The classification loss of the generated image is used as a signal to optimize G for generating a better image that fit the distribution of the target class.

$$L_{cls} = E_{x,c}[-\log D_{cls}(c/x)] \quad (7)$$

Here, $D_{cls}()$ gives the class probability of each class. The discriminator combinedly optimizes the adversarial and the classification loss. The importance of the two losses is adjusted with a constant factor λ_{cls} . The final objective of the discriminator is,

$$L_D = -L_{adv} + \lambda_{cls}L_{cls} \quad (8)$$

The negative value of the adversarial loss represents the opposite objective of the generator and the discriminator. Rather than the discriminator tries to maximize the adversarial loss it now minimizes the negative of adversarial loss and classification loss. The discriminator training process is illustrated in Fig. 3.

Reconstruction Loss. By minimizing the adversarial loss the generator learns to construct a realistic image that follows the data distribution of the target image. With the classification loss, it generates images that look like the images of the target class. However, none of this loss imposes any constraint to keep the details of the source image of the person or to preserve the identity of the person of the source image. To add this constraint we optimize two more loss on the generator. The reconstruction loss [13] is the L1 loss between the reconstructed image of the translated image back to its original domain, and the source image itself.

$$L_{rec} = E_{x,c,c'}[||x - G(G(x, c), c')||_1] \quad (9)$$

The reconstruction loss imposes half the objective of CycleGAN [11]. As we have the class level as an input to the generator we do not need two generators to impose this loss. This loss imposes an indirect constraint to the generated image to preserve some feature of the source image in the target image so that it can be converted back to the original image with the class level of the original image. Although this loss showed an impressive result in many application reported by many papers. But if the generator finds any shortcut to generate the image back to original it might successfully convert it back without preserving the desired feature in the generated image with the target level. To ensure the content of source image preserved in the generated image we have added the content loss.

Content Loss. The purpose of content loss is to force the generator to preserve the underlying content of the source image in the generated image. It is the difference between the content of the source and the generated image. Similar losses have shown good result in other applications, like image super-resolution [14]. The content vector is extracted from the high-level representation learned from a pre-trained deep

convolutional neural network. It is well known that the deep CNN trained with large dataset are good in extracting the high-level features from an image. In this work, we have used the VGG-19 [15] model trained with ImageNet [16] as a feature extractor to find the content loss.

The content loss is extracted from an earlier layer of VGG-19 model other than the output layer. The content loss of our model is represented in the following equation.

$$L_{con} = E_{x,c}[||VGG_l(x) - VGG_l(G(x,c))||_1] \quad (10)$$

Here, l represents the specific layer chosen for taking the feature from. In this model, it is conv44 of VGG-19 model. The $l1$ is chosen instead of $l2$ because various previous work has shown that the $l2$ loss has a tendency of generating blurry images.

The final objective function of the model consists of all the described losses along with individual weight for each loss.

$$L_G = L_{adv} + \lambda_{cls}L_{cls} + \lambda_{rec}L_{rec} + \lambda_{con}L_{con} \quad (11)$$

Here, λ_{cls} , λ_{rec} , λ_{con} are the hyper-parameters of the model. We have conducted an experiment with different values of these parameters found the best result with $\lambda_{cls} = 1$, $\lambda_{rec} = 10$ and $\lambda_{con} = 0.25$.

The generator training procedure is shown in Fig. 4. Here, Target class label and the real image is the input to the generator. The generated image then goes to the discriminator and the generator with the original source class label. The discriminator outputs the validity and the predicted class probability. From this, we calculate two of the objective loss- the adversarial loss and the classification loss. The generator outputs the reconstruction of the original image. We calculate the difference between the original image and the reconstructed image with L1 loss. To calculate the content loss, we pass both the real image and all corresponding generated image to a pre-trained and froze VGG-19 model. Then we calculate the content loss from the output of both models.

C. Network Architecture

After the introduction of the generative adversarial network with low-resolution images, Radford et. al. [17] introduced a deep convolutional generative adversarial network. We followed the architecture from this model with other changes and variations introduced in later papers. Both generator and discriminator use the module of the form Convolution-Batch Normalization-ReLU [18].

Generator with skip connection

The problem of image-to-image translation deals with images of high resolutions. In most of the applications,

the source and destination have similar image structure in both source and generated image, although they differ in appearances. In our application, the identity of a person at his different age must preserve and we have designed the loss function for that purpose. We consider this property to build the generator.

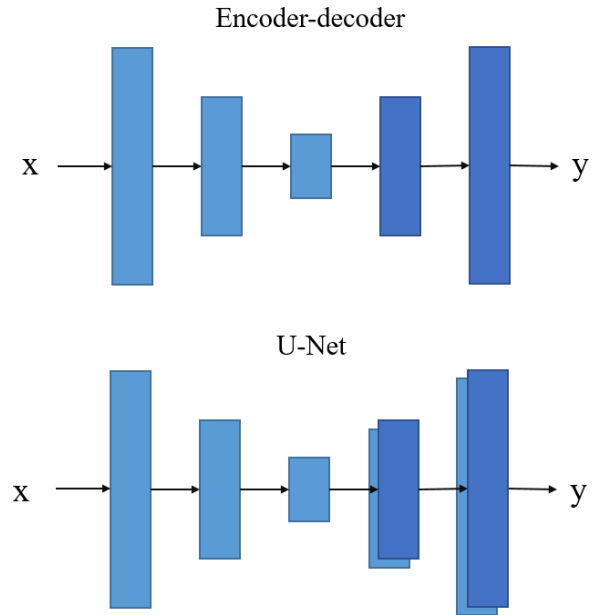


Fig. 5. Two variations of architecture choice for generator. The regular encoder-decoder architecture in the top image and the U-net [20] with skip connection between mirrored layers of encoder and decoder in bottom.

Many application in this domain has used encoder-decoder architecture [19]. In such a network, the input is passed to the encoder part of the network to compress the input to a low dimensional representation. This is called bottleneck representation. Then it is passed into the decoder which upsamples it to the image dimension. In many application, much of these low-level features are shared between both networks. For that reason, it would be beneficial to shuttle this feature directly between the networks.

To give the generator the means to share this information we have followed the intuition introduced in U-Net [20]. This network has skip-connection between the layers of encoder and decoder. Skip-connection is a concept popularized by ResNet [21]. It used skip-connection between its layers. The idea is to add the input of a layer to the output of that layer or a sequence of layers. It passes direct information in between layers of the network that help the model to understand the transition as well as the change in a transition. In U-Net the skip connection is added in between the encoder and decoder which is a skip connection between layer i and the layer $(n-i)$, where n is the total number of layers in the generator.

Table 1. Architecture of Generator

Layer	Input size \rightarrow Output size	Layer type
Down-sampling	$(r,c,3+n_c) \rightarrow (r,c,64)$	Conv(64, (7×7), 1, 3), InstNorm, ReLU
	$(r,c,64) \rightarrow (\frac{r}{2}, \frac{c}{2}, 128)$	Conv(128, (4×4), 1, 3), InstNorm, ReLU
	$(\frac{r}{2}, \frac{c}{2}, 128) \rightarrow (\frac{r}{4}, \frac{c}{4}, 256)$	Conv(256, (4×4), 1, 3), InstNorm, ReLU
Bottleneck	$(\frac{r}{4}, \frac{c}{4}, 256) \rightarrow (\frac{r}{4}, \frac{c}{4}, 256)$	Residual Connection: Conv(256, (3×3), 1, 1), InstNorm, ReLU
	$(\frac{r}{4}, \frac{c}{4}, 256) \rightarrow (\frac{r}{4}, \frac{c}{4}, 256)$	Residual Connection: Conv(256, (3×3), 1, 1), InstNorm, ReLU
	$(\frac{r}{4}, \frac{c}{4}, 256) \rightarrow (\frac{r}{4}, \frac{c}{4}, 256)$	Residual Connection: Conv(256, (3×3), 1, 1), InstNorm, ReLU
	$(\frac{r}{4}, \frac{c}{4}, 256) \rightarrow (\frac{r}{4}, \frac{c}{4}, 256)$	Residual Connection: Conv(256, (3×3), 1, 1), InstNorm, ReLU
	$(\frac{r}{4}, \frac{c}{4}, 256) \rightarrow (\frac{r}{4}, \frac{c}{4}, 256)$	Residual Connection: Conv(256, (3×3), 1, 1), InstNorm, ReLU
Up-sampling	$(\frac{r}{4}, \frac{c}{4}, 256) \rightarrow (\frac{r}{2}, \frac{c}{2}, 128)$	DeConv(128, (4×4), 2, 1), InstNorm, ReLU
	$(\frac{r}{2}, \frac{c}{2}, 128) \rightarrow (r,c,64)$	DeConv(64, (4×4), 2, 1), InstNorm, ReLU
	$(r,c,64) \rightarrow (r,c,3)$	Conv(3, (7×7), 1, 3), Tanh

Table 2. Architecture of Discriminator

Layer	Input size \rightarrow Output size	Layer type
Hidden Layers	$(r,c,3) \rightarrow (\frac{r}{2}, \frac{c}{2}, 64)$	Conv(64, (4×4), 2, 1), L. ReLU
	$(\frac{r}{2}, \frac{c}{2}, 64) \rightarrow (\frac{r}{4}, \frac{c}{4}, 128)$	Conv(128, (4×4), 2, 1), L. ReLU
	$(\frac{r}{4}, \frac{c}{4}, 128) \rightarrow (\frac{r}{8}, \frac{c}{8}, 256)$	Conv(256, (4×4), 2, 1), L. ReLU
	$(\frac{r}{8}, \frac{c}{8}, 256) \rightarrow (\frac{r}{16}, \frac{c}{16}, 512)$	Conv(512, (4×4), 2, 1), L. ReLU
	$(\frac{r}{16}, \frac{c}{16}, 512) \rightarrow (\frac{r}{32}, \frac{c}{32}, 1024)$	Conv(1024, (4×4), 2, 1), L. ReLU
	$(\frac{r}{32}, \frac{c}{32}, 1024) \rightarrow (\frac{r}{64}, \frac{c}{64}, 2048)$	Conv(2048, (4×4), 2, 1), L. ReLU
Outout Layer (D_{adv})	$(\frac{r}{64}, \frac{c}{64}, 2048) \rightarrow (\frac{r}{64}, \frac{c}{64}, 1)$	Conv(1, (3×3), 1, 1)
Outout Layer (D_{cls})	$(\frac{r}{64}, \frac{c}{64}, 2048) \rightarrow (1,1,n_d)$	Conv(n_d , ($\frac{r}{64} \times \frac{c}{64}$), 1, 0)

Markovian Discriminator

The concept is also known as PatchGAN. Previous work has shown that L1 and L2 loss has a tendency to generate blurry images [10]. It fails to generate high-frequency crispness but captured the low frequencies. To capture the high-frequency structure we can restrict the discriminator to local image patches. The PatchGAN discriminator restricts the output of the discriminator into local patches. Rather than outputting a single number to indicate if an image is real or fake the PatchGAN tries to classify if each N patch of an image is real or fake.

Our PatchGAN discriminator is implemented with convolution over the image to generate the output for the patches. The output of the discriminator is much smaller than the size of the image and this generate good quality images.

Architecture

The layer description of the generator and discriminator models are given in table 1 and 2. It follows the generator and discriminator layers from StarGAN[13] implementation. We have used instance normalization in all layers except the output layer. For discriminator, we have used Leaky ReLU. In the tables, nd: number of domains, nc: the dimension of domain labels, Conv: convolution with parameters(output channels, kernel size, stride, padding), InstNorm: instance normalization layer, L. ReLU: leaky relu.

D. Training

The Wasserstein GAN [22], [23] has shown that it is difficult to optimize the GAN objective proposed in the

original GAN paper [7]. To stabilize the GAN training and to generate high-quality image it proposed a modified training objective.

$$L_{adv} = E_x[\log D_{adv}(x)] - E_{x,c}[\log D_{adv}(G(x, c))] - \lambda_{gp} E_x[\|\nabla_x D_{adv}(\hat{x})\|_2 - 1] \quad (12)$$

The latter proposed WGAN is known as WGAN-GP. It used gradient plenty instead of clipping to enforce Lipschitz constant. A differentiable function f is a 1-Lipschitz if it has a gradient with the norm at most 1 everywhere.

So, instead of applying clipping like WGAN, WGAN-GP penalizes the model if the gradient norm moves away from its target norm of 1. In the training of our model, we have used the Eq. 12 instead of previously mentioned Eq. 6.

III. EXPERIMENTS

We have implemented the model with PyTorch [24] in python. All experiments were done in NVIDIA 1060 GPU.

A. Dataset

To train the model we need facial images with the age as a label. For this, we have used UtkFace [25] dataset. This dataset contains more than 23,000 images of people in the age range 0 to 116. We divided the images into 6 age groups. They are: 0-18, 19-30, 31-40, 41-50, 51-60, 60+. Dividing the data into age groups leads to an unequal amount of data into the different age group. To

balance this we have manually augmented the data by cropping the exact facial area with a face detector.

B. Experimental Setup

The input image size of the model is 128×128 . To train the model we have used the Adam optimizer [26] instead of gradient descent optimizer. It has proven to be a better choice for working with GANs. The Adam optimizer has two parameters- beta1 and beta2 which control the momentum of gradient descent. We trained the model for 50 epochs with the batch size of 8. The model has few other hyper-parameters mentioned in the method. All parameters of training the model are summarized in table 3.

Table 3. Values of training parameters

Parameters	Values
Image input size	128×128
Adam: beta1	0.5
Adam: beta2	0.999
λ_{cls}	1
λ_{rec}	10
λ_{con}	0.25
λ_{gp}	10
Epochs	50
Batch size	8
Learning rate	0.0002
LR Decay rate	0.99

C. Evaluation Matrix

It is hard to evaluate the generated image in any context. There is no common well-known method to judge the quality of generated images. GAN is a generative model which is used for generated various kind of images in the last few years. Different papers have reported performance in different ways. Most of them the human evaluation [10]. It is done by presenting a volunteer with generated and a real image for a small fixed amount of time. The task is to say which one is generated and which one is real. This score is converted into accuracy to report the performance of the generated image. This type of evaluation is neither reliable nor comparable.

In this work, we have generated images of different age group with an object of preserving the original person's identity. Previously on this domain called FaceAging [27] had introduced an evaluation matrix called face recognition score to evaluate the performance on preserving the original person's identity. For this, a pre-trained face verification algorithm is used. We have used OpenFace [28] for face verification. OpenFace is open-source face recognition and verification library.

The face verification algorithm takes two images as input and returns a score corresponding to the similarity between two faces. Lower the score the more confidence the algorithm that both images contain the same person. Commonly distance less than 0.7 is considered a good distance for positive verification. For all generated images of different ages, we have calculated the distance

of the person with the original images. For all images with this distance less than 0.7 we have treated it as positive verification and calculated the overall accuracy of the model.

As we have generated images of different age group, the image must preserve the features of that age group. For this evaluation, we have trained a classifier of these age groups. From the generated image we have calculated the predicted class for those images and calculated the overall accuracy of each class.

D. Result Analysis

Face Recognition Score

As mentioned in the previous section, we have used the pre-trained OpenFace library for calculating face recognition score. We have compared the result with known previous results. Anitpov et. al. [27] proposed a method for face aging using explicit Identity preserving optimization for preserving the identity of the person. They did not focus on the classification rate or the images containing content of the image. This two optimization has made our model yielding a better result in the previous works.

Table 4. Comparison of Face recognition score with previous works

Method	FR values
Pixelwise optimization	59.8%
Identity preserving optimization [13]	82.9%
Our proposed method	93.75%

Table 4 compared the performance of proposed approach compared with known previous approaches. For the comparison purpose, we have implemented the mentioned approaches and trained with the same dataset. It shows a significant improvement in preserving the facial identify of the person in the generated images. Table 5 shows the face recognition score of each age range.

Table 5. Comparison on Face recognition score with previous works

Age range	FR values
0-18	91.67%
19-30	91.67%
31-40	95.83%
41-50	100.0%
51-60	95.83%
60+	87.5%

Table 5 shows that the face recognition score is comparatively better for middle-range ages giving the best result for age range 41-50, but not that good for very young and very old. This is acceptable because of the fact that the human face has the most amount of changes between these age ranges. A face preserves fewer similarities in this age range from the rest of his life. That is why it is hard for generating the perfect image of very young and very old person.

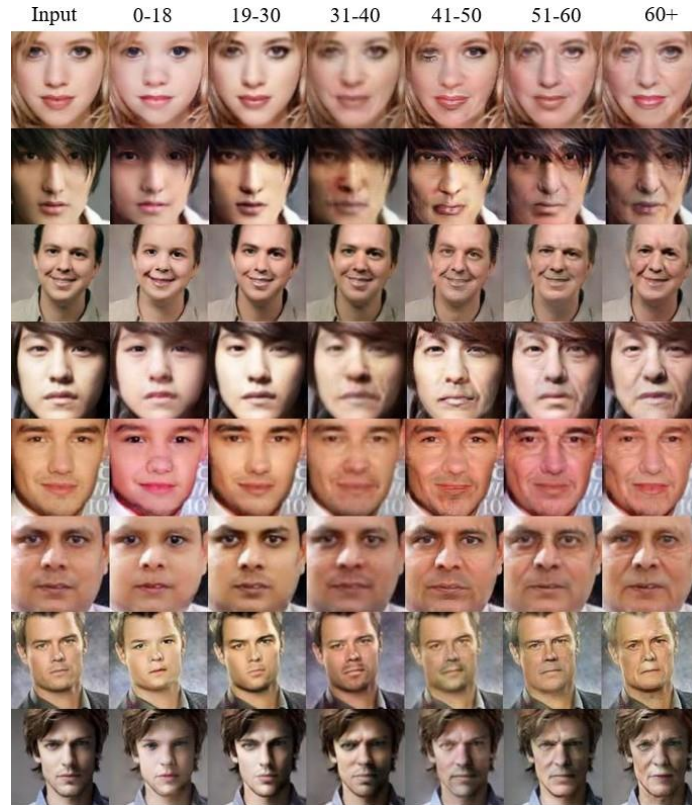


Fig. 6. Examples of generated images with the proposed mode.

Classification Accuracy

We have conducted experiments on classification accuracy of each class. We trained an explicit classifier for this job. We trained this model with transfer learning on DenseNet[29] model. This method is proven for generating the best result for any classification task. The initial model had an accuracy of over 98%.

The following table represents the accuracy of each class of generated images.

Table 6. Classification accuracies of generated images.

Age range	Classification accuracy
0-18	89.58%
19-30	85.41%
31-40	83.33%
41-50	85.41%
51-60	87.50%
60+	91.66%

As we can see from table 6 that better classification accuracies are achieved on the early and very old ages. And fewer accuracies in the middle age. The reason behind this is the fact that the human face goes through fewer changes in this period of ages. There are more similarities between faces of ages in middle age so the generated image is more misclassified in this range of age.

Finally there are few examples of the generated images in different age group with the proposed model shown in

Fig. 6. The first column is the input image of a person. Next 6 column shows the generated image of that person in different age.

IV. CONCLUSION

Generating facial images of different age given an image and its current age has a lot of practical applications. In this work, we have proposed a method for this job. Our GAN based method can generate facial images of different age groups. The experiment has shown that our method does better in term of preserving the person's identity in the generated image compared to known previous approaches. Also, our method generates images that introduce features of target age, which is justified with the classification accuracies. Possible improvements for this work will include improving the quality of the images with possible higher resolution images and improving the recognition feature in the generated images.

REFERENCES

- [1] Y. Fu, G. Guo, and T. S. Huang, "Age synthesis and estimation via faces: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 11, pp. 1955–1976, 2010.
- [2] X. Shu, J. Tang, H. Lai, L. Liu, and S. Yan, "Personalized age progression with aging dictionary," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3970–3978.
- [3] B. Tiddeman, M. Burt, and D. Perrett, "Prototyping and transforming facial textures for perception research," *IEEE*

- computer graphics and applications, vol. 21, no. 5, pp. 42–50, 2001.
- [4] I. Kemelmacher-Shlizerman, S. Suwajanakorn, and S. M. Seitz, "Illumination-aware age progression," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3334–3341.
 - [5] J. Suo, S.-C. Zhu, S. Shan, and X. Chen, "A compositional and dynamic model for face aging," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 3, pp. 385–401, 2010.
 - [6] Y. Tazoe, H. Gohara, A. Maejima, and S. Morishima, "Facial aging simulator considering geometry and patch-tiled texture," in ACM SIGGRAPH 2012 Posters. ACM, 2012, p. 90.
 - [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.
 - [8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.
 - [9] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.
 - [10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," CVPR, 2017.
 - [11] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017.
 - [12] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," arXiv preprint arXiv:1610.09585, 2016.
 - [13] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," arXiv preprint, vol. 1711, 2017.
 - [14] C. Ledig, L. Theis, F. Husza, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang et al., "Photo-realistic single image super-resolution using a generative adversarial network," in CVPR, vol. 2, no. 3, 2017, p. 4.
 - [15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
 - [16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.
 - [17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.
 - [18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.
 - [19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," science, vol. 313, no. 5786, pp. 504–507, 2006.
 - [20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
 - [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
 - [22] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in Advances in Neural Information Processing Systems, 2017, pp. 5767–5777.
 - [23] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," arXiv preprint arXiv:1701.07875, 2017.
 - [24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
 - [25] S. Y. Zhang, Zhifei and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017.
 - [26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
 - [27] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in Image Processing (ICIP), 2017 IEEE International Conference on. IEEE, 2017, pp. 2089–2093.
 - [28] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.
 - [29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in CVPR, vol. 1, no. 2, 2017, p. 3.

Authors' Profiles



of the world for taking a complex decision.

Shuvendu Roy has completed his B.Sc in Computer Science and Engineering from Khulna University of Engineering & Technology, Khulna, Bangladesh. His current research interest includes using deep learning and artificial intelligence in Computer vision application and build a robust system that understand the view

How to cite this paper: Shuvendu Roy, "Applying Aging Effect on Facial Image with Multi-domain Generative Adversarial Network", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.11, No.12, pp. 14-22, 2019.DOI: 10.5815/ijigsp.2019.12.02