Deep Neural Networks for Conditional Visual Synthesis

by

© Xin Huang

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

Memorial University of Newfoundland

May 2022

St. John's Newfoundland

Abstract

Conditional visual synthesis is the process of artificially generating images or videos that satisfy desired constraints. Individual visual synthesis tasks include high-fidelity natural image generation, artwork creation, face animation, etc. Such tasks have many real-world applications, such as database expansion, face editing in beauty camera, and face effects in short videos. With advances in deep learning, methods for conditional visual synthesis have evolved rapidly in recent years. Many of these recent approaches are based on Generative Adversarial Networks (GANs), which have strong abilities to generate samples following almost any implicit distribution, allowing the synthesis of visual content in an unconditional or input-conditional manner. However, GANs still have many limitations, such as difficulty in directly approximating high-resolution image distributions, poor model generalization ability on unpaired datasets, and limited power for mimicking human actions. Hence, it is worth to tackle these limitations and investigate how to handle different conditional visual synthesis tasks.

Four conditional visual synthesis tasks are investigated in this thesis. The first task studies how to generate high-resolution images from conditioning text descriptions. The second task simulates facial changes based on desired age inputs. How to synthesize realistic talking face videos from conditioning audio inputs is investigated as the third task. Finally, the forth task generates human-like painting actions based on desired target images. Both qualitative and quantitative validations are conducted for method developed for each task. Comparisons with existing works demonstrate the respective merits of these techniques. Insights on how to design conditional visual synthesis approaches are summarized.

Acknowledgements

Finishing my Ph.D. dissertation would be impossible without the help of all the people who supported me unconditionally. There isn't any word that I can use to express my feelings and appreciation to them. I would like to offer my sincere thanks to all those individuals whom I am deeply indebted.

First of all, I would like to thank my wonderful supervisor Dr. Minglun Gong for providing me strong supports and valuable advice, which help me to fulfill my dream. I have benefited greatly from his academic skills, great patience, positive spirit, and kindness, which I will always appreciate. He provided me excellent research environment that I have sufficient freedom in choosing research topics and building research collaborations. He also spent countless hours with me patiently discussing ideas, reviewing results, refining papers, and developing my skills in writing, presentation and communication. I was greatly honourable to study under his supervision and forever thankful for all I have learned from him.

In addition, I would like to express many thanks to my collaborators and fellow friends in Memorial University of Newfoundland, Drs. Zili Yi, Wendong Mao, Jun Zhou, Messrs. Songyuan Ji, Cao Hai, Mingjie Wang, and Xue Cui. I have enjoyed working with them on various research projects.

I am very grateful to all the committee members of my comprehensive exam and the supervisory committee members: Drs. Todd Wareham, Miklos Bartha, Manrique Mata-Montero, Wlodek Zuberek, Ting Hu, David Churchill, Mohamed Shehata, Adrian Fiech, Yuanzhu Chen, and Oscar Meruvia-Pastor for the valuable and insightful suggestions that helps enhance my basic training and perfect my proposal. I would also like to thank for

the administrative supports that I received from Ms. Barbara Hynes, Mr. Carl Dohey, Ms. Cathy Hyde, Mr. Andrew Kim, Ms. Erin Manning, Ms. Darlene Oliver, Ms. Jennifer Friesenand, and Dr. Sharene Bungay.

Special thanks is dedicated to my family for their encouragement and support, I owe the wonderful opportunities and experiences in my life to their dedication and hard work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

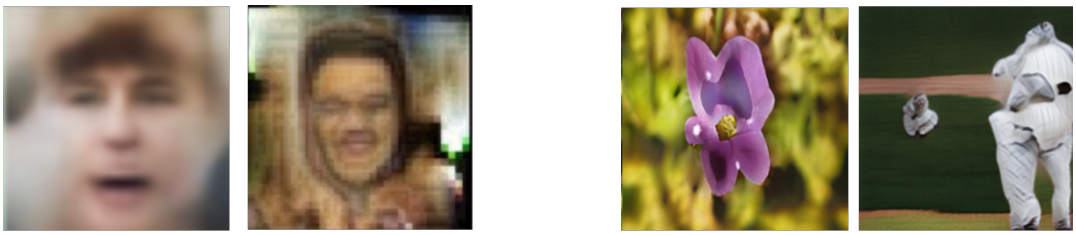
1.1 Motivation

With advances in deep learning over recent years, various computer vision techniques have enabled machines to mimic the human visual system with human-like intelligent behaviour. However, it still remains a significant challenge for those computer vision systems to truly understand the visual world.

"Just like to hear is not the same as to listen, to take pictures is not the same as to see."

——Fei-Fei Li

How to make a machine "see" really means how to make a machine "understand". In spirit of Fei-Fei Li's famous quote, training an artificial agent to have a deep understanding of visual contents and generating novel images and videos in the context of visual data can show the "intelligence" of a machine. Another noteworthy benefit of training these creative artificial agents would be in assisting and inspiring amateur users and even artists to generate new designs, unleash their creativity, and effectively express their thoughts visually. In



(a) (b)

Figure 1.1: (a) Blurred images generated by VAE [56]. (b) Defective images generated by HDGAN [163].

general, visual content synthesis is the process of converting information conveyed visually back to images and videos. There are many applications that humans can do with visual synthesis, for example, generating related images for a given label or a given paragraph. Visual synthesis requires the computer to understand the image or video, so that it can reproduce the real data distributions. This thesis is motivated by four visual content generation applications in different domains: creation of images from text descriptions; face synthesis with desired ages; generation of talking face videos from audios; and human-like painting strokes for artwork reproduction.

Research on photorealistic image generation techniques can be traced back to the 1980s. Due to the limited computing power of early computers, algorithmic models heavily relied on hand-designed features, which make the generation mainly focus on some simple, regular, and low-resolution image processing techniques, such as structure propagation based image restoration [147], Markov random field (MRF) based texture image generation [35, 15], Karhunen-Loeve transform (KL Transform) based face image synthesis [133], linear and nonlinear independent component analysis (ICA/NonlinearICA) [47, 48], and Gaussian mixed models (GMM) [153, 90].

With the development of deep neural networks, researchers have realized the great potential of deep neural networks for applications in the field of visual content generation. Auto-regressive models, such as PixelRNN and PixelCNN [135], model the conditional probability distribution between pixel values at all locations on the target image by a multilayer Long Short-Term Memory (LSTM) network and a Masked Convolution Network. For images with large spatial resolution, the sequence span between pixels at backward positions and forward positions is too large, making the modelling of conditional distribution difficult. Therefore, PixelRNN and PixelCNN do not perform well in generating high-resolution images.

Compared with PixelRNN and PixelCNN, Variational Autoencoders (VAEs) [56] and Generative Adversarial Networks (GANs) [26] better model the relationship between training images and hidden variables. The VAEs consist of an encoder and a decoder, and the decoder can be learned from a latent space to the real picture space by using a variational inference method. They have succeeded in generating images of digits and celebrity faces, but the generated images are usually blurred (see Figure 1.1(a)). GANs consist of a generator and a discriminator which are trained to learn in an adversarial way. The generator tries to synthesize the image towards the true data distribution so that the discriminator cannot distinguish the real image from the fake one, while the discriminator tries to differentiate real and fake data. Although GANs do improve the resolution of generated images, its performance is often limited by the instability of their training and the mode collapse problem [4, 107]. This leads to significant flaws in the results, such as insufficient diversity of generated images, low quality and high computational stress in generating high-resolution images. Hence, it still remains a challenge to fully optimize the GAN generator to generate realistic images that always follow the target distribution (see Figure 1.1(b)).

Video generation is an extension of image generation, which has a higher level of difficulty and is still in the early stages of research. Many research works have utilized GANs to synthesize realistic human videos from audios [12] or realize transformations from one person to another [85] for various entertainment applications. Compared with image synthesis, the output videos consists of a sequence of still images with temporal consistency. Video is difficult to model directly due to its high complexity. In particular, the cumulative error arises in continuous multi-frame prediction.

Figure 1.2: Images generated from text descriptions: (a) low-resolution images generated by DCGAN [96]; (b) Unnatural images generated by StackGAN [160].

Overall, the following three challenges have prevented generative models to generate desired images and videos. First, one of the biggest challenges comes from the lack of realism of the synthesized results. Humans are constantly exposed to real images and videos in real life and hence can easily identify unrealistic images or videos when viewing them. In practice, people are sensitive to several common synthetic artifacts, such as blurred images, images with significant borders, images with missing structural information, images with missing details, and so on. Images generated by Deep Convolutional Generative Adversarial Network (DCGAN) [96] based on text can only reflect the general shape and colour

of the flower and bird. Their results lack details and vivid object parts (e.g., textures of flowers and legs of birds; see figure 1.2(a)), which make them neither realistic enough nor have sufficiently high resolution. Simply stacking multiple generators to generate higher resolution images usually misses fine-grained information and background-object smoothness (see figure 1.2(b)). Hence, it is important to explore on how to ensure the realism of generated results.

Figure 1.3: Age Progression/Regression: the first column shows the original faces marked with their true ages on the bottom. The right 10 columns are synthesized faces generated by Conditional Adversarial Autoencoder [162].

Second, it is also a very big challenge to make synthesized images and videos satisfy given conditional inputs. For example, when synthesizing a person's face under a different age, the output face image should maintain identity features, expressions, and backgrounds of the input face picture, while satisfying the desired age condition at the same time. Many of the generated samples do not adequately follow the input conditions. As shown in Figure 1.3, faces generated by Conditional Adversarial Autoencoder [162] under different age conditions do not match ages well, especially when the target age is very young or very old. How to design a framework so that the synthesis model can adequately follow input conditions is very important in conditional synthesis tasks.

In addition, when synthesizing non-photorealistic imagery in a stroke by stroke manner, some researchers approach the problem by focusing on images with simpler structural

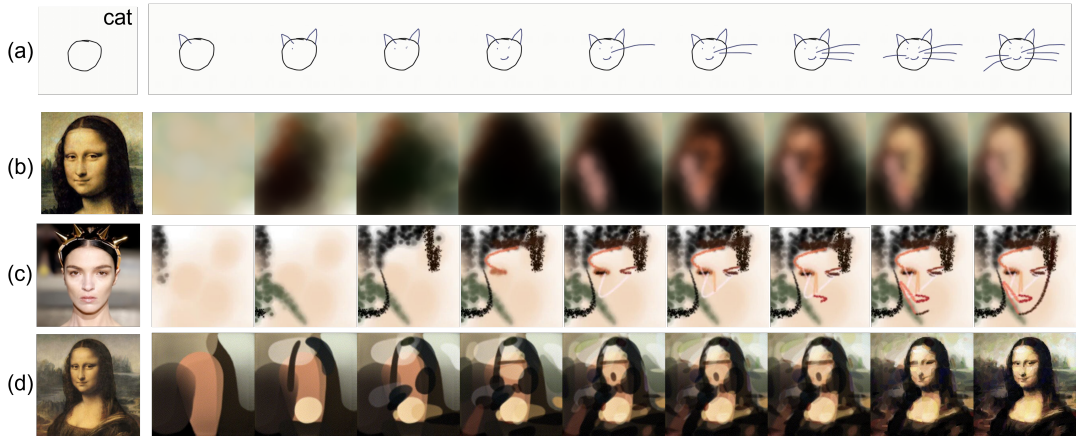


Figure 1.4: Stroke-Based Rendering: (a) Sketch-rnn-generated sketches [33]. (b) SPIRAL-generated Mona Lisa [22]. (c) Improved SPIRAL-generated doodling [76]. (d) Model-Based DRL-generated painting [45]

compositions, such as sketches and doodles (see Figure 1.4(a)), whereas others experiment on images containing richer textures and more complex structures, such as natural images and artistic paintings (see Figure 1.4(b, c and d)). Deep Reinforcement Learning (DRL) is a combination of deep learning and reinforcement learning, which uses deep learning to train an agent in the reinforcement learning process to accomplish a target task. DRL has been successfully applied into neural painting tasks, but mimicking human-like behaviour in painting is still a challenge. For example, artists usually prioritize important foreground objects rather than backgrounds details to make the painting recognizable in early stages. This behaviour is now captured by existing approaches.

Many current results related to image, video and artwork generation are mostly from the perspective of improving and studying loss functions. In this thesis, tasks for text-to-image generation, aging face generation, talking face video generation, and artwork painting are used as gates for exploring how to use improved generative models with optimized network

structures to accomplish the visual content generation tasks. The research question this thesis tries to answer is:

What are common and effective strategies for various conditional visual synthesis tasks?

1.2 Contributions

To answer the research question and to address aforementioned challenges for individual visual content generation tasks, this thesis designs new network architectures, investigates structure distributions, and modifies learning objectives. The contributions for individual tasks include:

i) **A novel Hierarchically-fused Generative Adversarial Network (HfGAN) is built to construct high-quality images from text descriptions.** Different from previous works which usually adopt multi-pairs of generators and discriminators for high-resolution image generation stage by stage, HfGAN applies hierarchical feature map fusion, which can fully extract and utilize the local and global features to synthesize photo-realistic images with only one pair of generator and discriminator. Extensive experiments demonstrate that the proposed HfGAN improves the quantity of the generated images and shows more stable training behaviour than other state-of-the-art methods.

ii) **A novel Landmark-guided Dual-learning cGAN (LDcGAN) with a multi-attention mechanism is introduced for aging face generation.** LDcGAN learns the transition pattern at different ages and performs well in preserving personal identity and keeping face-aging consistency. External landmark attention is introduced to the network for adjusting facial structure changes related to aging and a built-in attention mechanism is also adapted to emphasize the most discriminative regions relevant to aging and minimize changes that

affects personal identity and background. Conditioned with age vectors, the primal conditional GAN in LDcGAN network converts input faces to target ages, and the dual cGAN invert the previous task, which feeds synthesized target faces back to the original input age scope for enhancing age consistency.

iii) **A novel audio-to-video-to-words framework called AVWnet is presented, which can generate fine-grained talking face videos with better audio-lip consistency and higher frame quality.** A multi-scale attentive generation network and multi-purpose discriminators are applied for high-resolution videos. A reinterpretation module drives the synthesized video to align its semantic information with the input audio, resulting the improvement of audio-lip consistency.

iv) **An end-to-end attention-aware reinforcement learning approach is designed to paint like humans, which better approximates the target image under small number of strokes and capture finer foreground details in the final results.** To better mimics the painting process used by human artists, this thesis gives a detailed investigation on a RL framework with attentions on foreground objects. The experiments demonstrate its effectiveness in generating strokes sequentially to recover foreground content details with limited strokes.

Through conducting research on vast different visual synthesis tasks, this thesis finds two common and effective design strategies. The first is to apply attention mechanisms for associating input conditions with visual features. For examples, in text-to-image synthesis task, attention is used to build the connection between text descriptions and image features, whereas in neural painting, attention is used to select high priority areas to paint. The second strategy is to add feedback loop for enhancing the consistency between input conditions and generated results. For examples, both primal and dual cGANs are used for

enhancing age consistency in aging face generation, whereas lip-reading model is applied to improve audio-lip consistency in talking face generation.

1.3 Organization of the Dissertation

The rest of this thesis is organized as the following. First, several core concepts about Auto-regressive Models, Variational Auto-Encoder (VAE), Generative Adversarial Networks (GANs), variants of GANs, Deep Reinforcement Learning, and Evaluation metrics are introduced in Chapter 2. The Hierarchically-fused Generative Adversarial Network (HfGAN) is then presented in Chapter 3. The Landmark-guided Dual-learning cGAN (LD-cGAN) is introduced in Chapter 4. The audio-to-video-to-words framework (AVWnet) is discussed in Chapter 5. The attention-aware Deep Reinforcement Learning based Neural Painting is introduced in Chapter 6. Finally, conclusions and discussion on future work are presented in Chapter 7.

1.4 Co-Authorship Statements

Research conducted in this thesis has led to the following published or submitted works:

- Hierarchically-fused generative adversarial network for text to realistic image synthesis. Conference on Computer and Robot Vision, 2019 [43] (Chapter 3; Best Paper Award on Computer Vision).

- Landmark-guided conditional GANs for face aging. International Conference on Image Analysis and Processing, 2022 [42] (Chapter 4).

- Enhanced Face Aging using Dual-learning and Muti-attention Mechanism. Applied Intelligence (Chapter 4; under review).

- Fine-grained talking face generation with video reinterpretation. The Visual Computer, 2021 [44] (Chapter 5).

- Attention-Aware Neural Painting via Deep Reinforcement Learning. Neurocomputing (Chapter 6; under review).

The above publications all have my supervisor, Dr. Minglun Gong, as a co-author, who directed my research and edited the manuscripts. In addition, the publications associated to Chapters 3 and 5 have Mr. Mingjie Wang as a co-author, who provided valuable feedback through discussions. In all cases, the identification of research topics, key ideas for implementation, experimental designs, data analysis and interpretation, and manuscript preparation were performed by myself.

I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-authors to include the publication materials in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

Chapter 2

Deep Generative Models

2.1 Auto-regressive Models

The emergence of deep neural networks has greatly enhanced the feature representation capability of relevant network models and the ability to fit complex nonlinear functions in high-dimensional spaces. Many classical deep image generation models have been proposed, including auto-regressive models. As the auto-regressive class of deep image generation models, PixelRNN and PixelCNN [135] were proposed in 2016. They modelled the joint distribution of all pixels on the corresponding image as the product of a series of conditional probabilities in the pixel sequence prediction process. Specifically, as shown in Figure 2.1(a), for an input image x with $n \times n$ pixels, the model treats it as a sequence with length n^2 which is formed through reading by rows: $\{x_1, \cdots, x_{n^2}\}$. The overall joint distribution $p(x)$ is:

$$p(x) = \prod_{i=1}^{n^2} p\left(x_i \mid x_1, \cdots, x_{i-1}\right) \tag{2.1}$$

Here $p\left(x_i \mid x_1, \cdots, x_{i-1}\right)$ is the probability of the i-th pixel x_i given all the previous pixels $\{x_1, \cdots, x_{i-1}\}$.

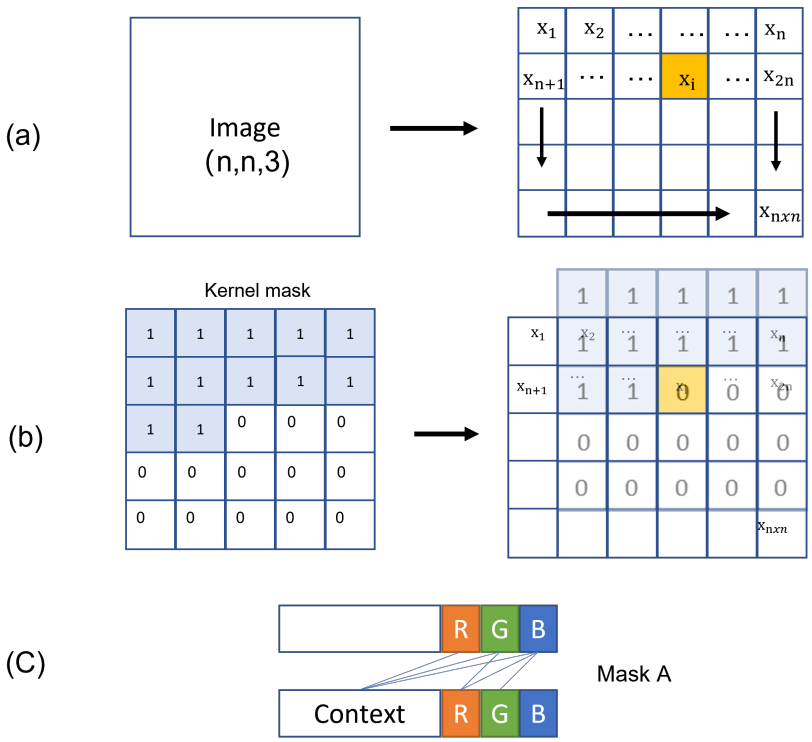


Figure 2.1: Auto-regressive Models [135]. (a) Context. (b) Apply the kernel mask in spatial layout. (c) Masked convolution in PixelCNN.

The image generation of the auto-regressive model follows a scheme of row-by-row and pixel-by-pixel. PixelRNN models this sequential generation process by a recurrent neural network (RNN) [104] built with LSTM layers [29]. LSTM has demonstrated its considerable power for sequential-data-related tasks like handwriting recognition, speech recognition, text-to-speech synthesis, currently holding the best performance for these tasks. In the improved work PixelCNN, the authors make the pixel prediction at the current position only related to the pixel value at the previous position without reference to the pixel value

behind, by using a masked convolutional layer. We can see them but we shouldn't use them because in the generation process, it won't have access to those future pixels. They use a mask over convolutional kernel in training, as shown in Figure 2.1 (b). To generate x_i, the kernel mask is applied to the convolution so that only pixels with value 1 in the past are used. As shown in Figure 2.1 (c), each colour channel should be dependent on only the previously generated colour channels and pixels that were generated before. Red channel is dependent on only the previous pixels, green channel is dependent on all previous pixels and the red channel pixel that was just generated, and blue channel is dependent on context, red channel and green channel pixels. Compared to PixelRNN, the advantage of the parallel function of PixelCNN is only available when training or evaluating the test images.

2.2 VAE

Diederik P. Kingma and Max Welling proposed a new image generation model in 2014: Variational Auto-Encoder (VAE) [56], which fits the data distribution of real images in the training set by Stochastic Gradient Variational Bayes (SGVB) [56]. The VAE consists of two parts: an encoder network and a decoder network. In the training process, the encoder network first encodes the input image x into the latent space to get the corresponding latent vector$z$, and then the decoder network maps the latent vector back into the image space to get the corresponding reconstructed image x'.

In the specific model design, the VAE assumes that the image's corresponding hidden variables follow Gaussian distribution. The encoder takes x as input and outputs the corresponding mean vector $\mu(x)$ and standard deviation vector $\sigma(x)$. Then, the VAE introduces a KL Divergence $\mathcal{KL}(N(\mu(x), \sigma(x))\|N(0, 1))$ on the latent variable space to constrain la-

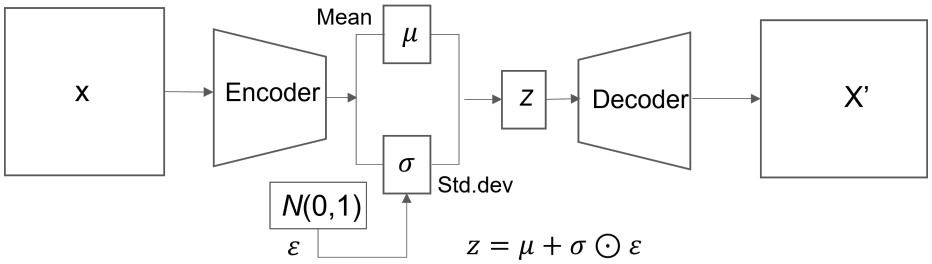


Figure 2.2: Overall model structure of the VAE [56]. VAEs are composed of an encoder e and a decoder d. The encoder defines the approximate posterior distribution $q(z \mid x)$, which takes as input an observation x and outputs two latent variables $\mu(x)$ and $\sigma(x)$, which are the parameters of a Gaussian distribution learned during the training and used for specifying the conditional distribution of the latent representation z. ε is generated from a standard normal distribution which is used to maintain stochasticity of z. The sampled vector z is passed through the decoder and obtain the predicted fake samples x'.

tent variables of the image to a standard normal distribution $N(0, 1)$. If the latent variables are directly sampled from the encoded distribution$N(\mu(x), \sigma(x))$ and input to the decoder network, the gradient of the decoder part cannot be passed back to the encoder. To solve this problem, the VAE employs a method known as the Reparameterization Trick [56]. First, a random vector ε is sampled from the standard normal distribution $N(0, 1)$, and then the latent variable z is obtained through $z = \mu(x) + \varepsilon \otimes \sigma(x)$. Finally, z is fed into the decoder network to obtain the corresponding reconstructed image x'. The overall model structure of the VAE is shown in Figure 2.2. The objective function for the model training is:

$$\mathcal{L}_{VAE} = \mathcal{KL}(N(\mu(x), \sigma(x)) \| N(0, 1)) + \|x' - x\|_2^2 \tag{2.2}$$

Here the regular term $\mathcal{KL}(N(\mu(x), \sigma(x)) \| N(0, 1))$ makes the distribution of the latent

variable obtained by the encoder approximate the standard normal distribution $N(0,1)$. $\|x' - x\|_2^2$ is the Mean Squared Error (MSE), which is used to constrain the reconstructed image x' to be as close as possible to the input image x. For binary images (e.g., MNIST dataset), the loss function can be replaced with a cross-entropy loss function.

2.3 GANs and Conditional GANs

Another classical work in deep generation models is the Generative Adversarial Networks (GANs) [26]. Basic GANs consist of two networks: a Generator (G) and a Discriminator (D). The Generator (G) is used to learn the distribution to the real data. The Discriminator (D) is a binary classifier, which is used to discriminate whether the input is real data or generated data. t is the real sample,consistent with the $P_r(t)$ distribution. z is a hidden space variable, which conforms to a $P_z(z)$ distribution, such as a Gaussian distribution or a uniform distribution. Then the data $t' = G(z)$ is generated by G after sampling from the hypothetical hidden space. G accepts a random noise vector z and learns the data distribution to generate an image $G(z)$, whereas D distinguishes the whether $G(z)$ is "real" or not. During training, D aims at misleading G when G strives to generate "real" images. Hence, these two networks compete in a two-player minmax game:

$$\begin{aligned} \min_G \max_D V(D,G) = & E_{t\sim\, p_{data}(t)}[\log D(t)] \\ & + E_{t\sim\, p_z(z)}[\log(1 - D(G(z)))], \end{aligned} \tag{2.3}$$

where V is the overall GAN objective. $D(t)$ computes the probability of t being "real", which should approach 1. For $\min_G$, $D(G(z))$ represents the probability that the generated image by G is "real". V will diminish when $D(G(z))$ grows. For $\max_D$, the better D's ability is, the higher $D(t)$ should be, thus $D(G(z))$ should be lower, and $V(D,G)$ will be

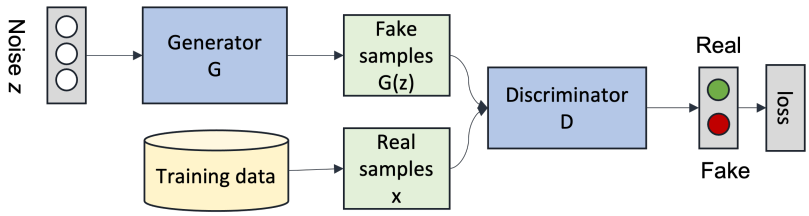


Figure 2.3: The framework of Generative Adversarial Networks (GANs).GANs consist of a generator and a discriminator. The generator G takes the random noise vectors z as inputs and generates fake samples $G(z)$. The discriminator D takes both fake samples $G(z)$ and the real samples x from the training set as inputs and distinguish whether the generated samples $G(z)$ are real or fake.

bigger. Conditional GANs [79, 25] are extension of GANs where both the generator and the discriminator receive additional conditioning variables c, yielding $G(z, c)$ and $D(x, c)$. The loss function of the conditional generative adversarial network is:

$$\min_D \max_G V(D, G) = -\mathbb{E}_{x \sim P_r}[\log D(x, c)] - \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z, c), c)] \tag{2.4}$$

This formulation allows G to generate images conditioned on variables c. In practice, the condition can be texts [160, 43], images [49, 166], audios [44], or other input contents [51, 146]. Usually, the condition variables c can either be concatenated with the random noise z in the first layer or be combined in the subsequent layers as additional channels. A later work AC-GANs [86] improved the discriminative network by adding auxiliary classifiers to increase the capability of the model.

2.4 Variants of GANs

2.4.1 Wasserstein GAN

Wasserstein GAN (WGAN) [5] provides the first theoretical explanation for the instability that occurs in the training of GANs. When the discriminative network is the optimal classifier and there is no overlapping between the generated and real image distributions, the gradient of the discriminator D back to the generator G is 0, resulting in a vanishing gradient problem. To solve this problem, WGAN proposes a training method using the Wasserstein distance, which is defined as follows:

$$W\left(P_r, P_g\right) = \inf_{\gamma \sim \Pi\left(P_r, P_g\right)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|] \tag{2.5}$$

Here $\Pi\left(P_r, P_g\right)$ is the set of all possible joint distributions for the combination of the real image distribution P_r and the generated image distribution P_g. This distance defines the cost of the optimal transport plan. Compared with KL divergence distance and JS divergence distance, the superiority of Wasserstein distance is that when the real image distribution P_r and the generated image distribution P_g do not overlap, the Wasserstein distance can still reflect their distances. However, the minimization over γ in Equation 2.5 is generally intractable, so [5] turned to the dual formulation of this optimal transport problem to obtain the value function for WGAN. The loss function of the discriminator D is:

$$\mathcal{L}_{\text{WGAN}}(D) = \mathbb{E}_{x\sim P_g}[D(x)] - \mathbb{E}_{x\sim P_r}[D(x)] \tag{2.6}$$

The loss function of the generator G is:

$$\mathcal{L}_{\text{WGAN}}\left(G\right) = -\mathbb{E}_{x\sim P_g}[D(x)] \tag{2.7}$$

2.4.2 WGAN-GP

To make the discriminator satisfy the Lipschitz constraint, WGAN restricts all parameters w_i in the discriminator within a certain range $[-0.01, 0.01]$ during the training process. However, this leads to undesirable behaviour by creating pathological value surfaces and capacity under use, as well as gradient explosion/vanishing without careful tuning of the weight clipping parameter c. Instead of restricting all parameters w_i, Wasserstein GAN with Gradient-Penalty (WGAN-GP) [32] uses a gradient penalty to make the Lipschitz constraint satisfied in the discriminator. WGAN-GP directly restricts the gradient of the loss function by:

$$\mathcal{L}_{\mathrm{WGAN-gp}}(D) = \mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{x\sim P_g}[D(x)] - \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}} \left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1\right)^2 \qquad (2.8)$$

where $\mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{x\sim P_g}[D(x)]$ is the original critic loss and $\mathbb{E}_{\hat{x}\sim P_{\hat{x}}} \left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1\right)^2$ is the gradient penalty. $P_{\hat{x}}$ is the distribution obtained by uniformly sampling along a straight line between the real distributions P_r and generated distributions P_g. With this gradient penalty term, the capability of the discriminator can be exploited better, and the overall training of the generative adversarial network is more stable, and the quality of the generated images is improved.

2.4.3 DCGAN

In the early days when generative adversarial networks were proposed, researchers did not have guidelines for designing the structures of the generative network G and the discriminative network D, so they could not get a good generative model G in their experiments. Deep Convolutional Generative Adversarial Networks (DCGAN) [96] first studied the design of the network structures of G and D. After extensive experiments, they found that the

following recommendations should be used for the generative and discriminative networks:

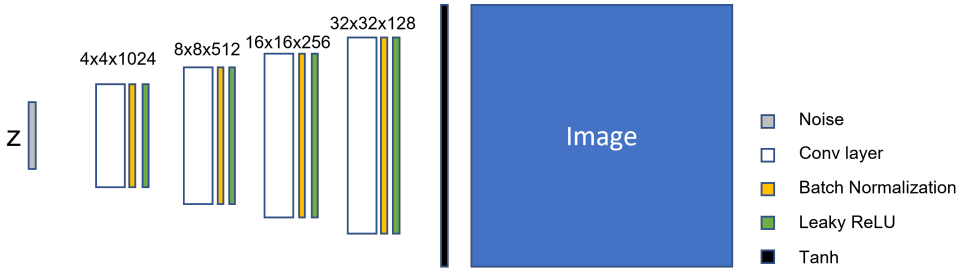


Figure 2.4: The generator of DCGAN [96]). z is the input noise.

a. Do not use pooling (including max pooling, average pooling) in the generative and discriminative models; adopt convolution with step size for up-sampling or down-sampling.

b. Do not use fully connected layers elsewhere in the network, except for the first layer of the generative model and the last layer of the discriminative model.

c. Use Batch Normalization in generative and discriminative models

d. Use Tanh as the activation function for the last layer in the generated model and the ReLU for all the rest.

e. Use the LeakyReLU activation function for all layers in the discriminator.

Following the above design guidelines, the structure of generative model is schematically shown in Figure 2.4.

2.4.4 SAGAN

In recent years, the Self-Attention mechanism has achieved very good results in many natural language processing works [136]. Zhang et al. [159] proposed Self-Attention GAN

(SAGAN) which introduces the self-attention mechanism into the generator and discriminator of generative adversarial network to improve the quality of the generated images.

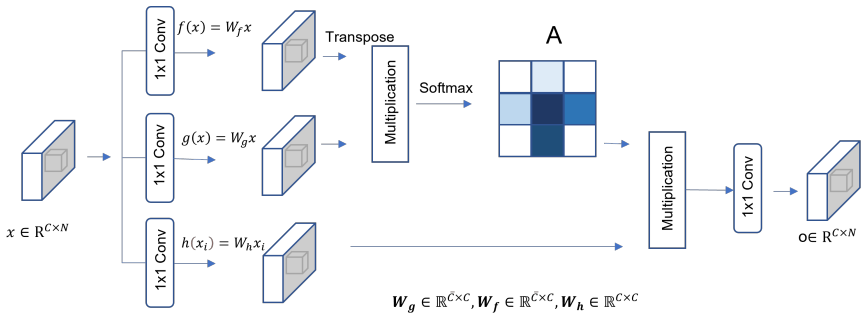


Figure 2.5: Self-Attention Generative Adversarial Networks [159]. x indicates the image features, which are transformed into two feature space f and g to get the attention. Transpose the output of f(x) and multiply it with the output of g(x), then normalize it by softmax to get an Attention map A. Multiply the obtained Attention map A and the output of h(x) pixel by pixel to get the adaptive attention feature maps O.

As shown in Figure 2.5, the feature map of the convolutional layer is linearly transformed and channel compressed using two 1×1 convolutions, and the generated two tensors are reshaped into matrix forms. After operations of transpose, multiplication, and softmax, the attention map is obtained. The original feature map is then linearly transformed with a 1×1 convolution (the number of channels is kept constant), and then multiplied with the attention map matrix to obtain the self-attention feature maps. Finally, the result of the weighted sum of the self-attentive feature map and the original feature map is considered as the final output. The self attention mechanism is able to utilize information from more distant regions, and each location is able to combine information from similar or related

Figure 2.6: Agent's action and environment's reply.

regions. Hence, the regional consistency of the generated images is improved.

2.5 Deep Reinforcement Learning

In recent years, Reinforcement Learning (RL) [121] is a technology that has received much attention in the field of intelligent control. In RL, the most important actors are the agent and the environment. As shown in Figure 2.6, the agent, in its interaction with the environment, acquires observations of the state of the environment (not necessarily the entire states of the environment) and then determines the next action. The environment may change in the interaction with the agent as a result of the action, or it may change on its own. After performing an action, the agent also receives a reward signal from the environment, and this value determines the merit of the current state. The goal of the agent in its continuous interaction with the environment is to obtain the maximum cumulative reward.

A RL agent interacts with an environment over time. At each time step t, the agent receives a state s_t in a state space $\mathcal{S}$ and selects an action a_t from an action space $\mathcal{A}$, following a policy $\pi\left(a_t \mid s_t\right)$, which is the agent's behaviour (i.e., a mapping from state s_t to actions a_t). It then receives a scalar reward r_t and transitions to the next state s_{t+1}, according to the environment dynamics, or model, for reward function $\mathcal{R}(s, a)$ and state transition probability $\mathcal{P}\left(s_{t+1} \mid s_t, a_t\right)$, respectively. In an episodic problem, this process continues until the agent reaches a terminal state and then it restarts. The return $R_t =$

$\sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the discounted, accumulated reward with the discount factor $\gamma \in (0, 1]$. The agent aims to maximize the expectation of such long term return from each state. The problem is setup in discrete state and action spaces. It is not hard to extend it to continuous spaces.

A deep reinforcement learning (deep RL) method is derived when deep neural networks are used to approximate any of the following components of reinforcement learning: value function $\hat{v}(s; \boldsymbol{\theta})$ or $\hat{q}(s, a; \theta)$, policy $\pi(a \mid s; \theta)$, and model (state transition function and reward function). Here, the parameters θ are the weights in deep neural networks. At present, Deep Reinforcement Learning (DRL) algorithms are mainly divided into value function approximation method and policy gradient method. The most representative DRL algorithm based on value function approximation is the Deep Q-network (DQN) [80] algorithm and the most representative DRL algorithm based on policy gradient is the Deep Deterministic Policy Gradient (DDPG) [62] algorithm.

2.6 Evaluation metrics

How to measure the quality of generated visual contents and to compare the advantages and disadvantages of different generation models have become important issues in image generation related research. Some intuitive comparison methods include subjective visual judgment and evaluation of image quality through user study. However, the results obtained by these comparison methods are subjective and cannot be used as the only criterion to justify the research method. In contrast, the quantitative numerical evaluation is more objective and convincing.

2.6.1 PSNR

Peak Signal-to-Noise Ratio (PSNR) [46] is often used to measure the similarity between the generated image and the Ground Truth. First, the Mean Square Error (MSE) between the generated image and the Ground Truth is calculated as follows:

$$MSE = \frac{1}{mn} \sum_{0}^{m-1} \sum_{0}^{n-1} \|f(i,j) - g(i,j)\|^2 \tag{2.9}$$

where f is the matrix data of true images, g represents the matrix data of generated images. m and n are the number of rows and columns in the input images. Then the PSNR is calculated by:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_{\text{pixel}}^2}{MSE} \right) \tag{2.10}$$

where MAX_{pixel} is the largest pixel value of the real result image. For images that use 8-bit binary representation, MAX_{pixel} usually takes the value of 255. It is measured in decibels (dB), and the larger the value, the closer the generated image distribution is to the real image distribution.

2.6.2 SSIM

In fact, it is possible that those with higher PSNR score may appear to be of worse quality than those with lower PSNR score. This is because the sensitivity of human vision to errors is not absolute, and its perception results vary depending on many factors.

Structural Similarity Index Measure (SSIM) [144] is an image quality evaluation metric that measures the similarity between the generated image and the real image in terms of brightness, contrast and structure.

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{2.11}$$

where μ_x is the mean value of x, μ_y is the mean value of y, $\sigma_x^2$is the variance of x and σ_y^2 is the variance of y. The range of SSIM is [0,1], and the larger the value of SSIM, the higher the similarity of the two images. Usually, SSIM only performs well at a specific resolution, and in order to make its performance stable for images of different resolutions, researchers have proposed the Multi-scale Structural Similarity Index Measure (MSSIM) [145], which divides the image into n blocks using sliding windows and weights the mean, variance and covariance obtained from each window, and then calculates the structural similarity result of the corresponding block, and finally the average value is used as the MSSIM result of the two images.

2.6.3 Inception score

The Inception Score [105] is the distance between the generated image distribution and the ground truth distribution by feeding the generated images into a pre-trained classification model 'Inception-V3' [123]. Inception Score considers the quality of generated images from two perspectives:(1) Reality. The generated image x is fed into the Inception-V3 model to obtain the vector y of its output dimension, and the value of each dimension of the vector corresponds to the probability that the image belongs to a certain class. For a real image, the probability that it belongs to a certain class should be very high, while the probability that it belongs to other classes should be small, i.e., the entropy of $p(y|x)$ should be small. (2) Diversity. If a model can generate enough diverse images, then the distribution of its generated images in each category should be evenly distributed, that is, the entropy of the marginal distribution $p(y)$ of the generated images in all categories is

large. The Inception Score is:

$$\mathrm{IS}(G) = \exp\left(\mathbb{E}_{\mathrm{x}\sim p_g} D_{KL}(p(y \mid \mathrm{x}) \| p(y))\right) \tag{2.12}$$

2.6.4 VS similarity

For the specific kind of generation model such as text to image generation model, how to measure the alignment between generated images and the conditioned text is very important to show the performance of models. Zhang et al. [163] proposed the Visual-Semantic similarity (VS similarity) [163] measurement which can measure the distance between synthesized images and text descriptions through a trained visual-semantic embedding model. The scoring function is $c(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|_2 \cdot \|\boldsymbol{y}\|_2}$. They train two mapping functions f_v and f_t to map real images and text embeddings into a common space, by minimizing a bi-directional ranking loss:

$$\sum_{v} \sum_{\boldsymbol{t}_v} \max\left(0, \delta - c\left(f_v(v), f_t\left(\boldsymbol{t}_v\right)\right) + c\left(f_v(v), f_t\left(t_{\bar{v}}\right)\right)\right) +$$

$$\sum_{\boldsymbol{t}} \sum_{v_{\bar{t}}} \max\left(0, \delta - c\left(f_t(\boldsymbol{t}), f_v\left(\boldsymbol{v}_v\right)\right) + c\left(f_t(t), f_v\left(v_{\bar{t}}\right)\right)\right)$$

where δ is the margin, $\{v, t\}$ is ground truth image-text pair. $\{v, t_{\bar{v}}\}$ and $\{v_{\bar{t}}, t\}$ are mismatched image-text pairs.

Chapter 3

Hierarchically-fused Generative Adversarial Network for text to realistic image synthesis

In this Chapter, we present a novel Hierarchically-fused Generative Adversarial Network (HfGAN) for synthesizing realistic images from text descriptions. While existing approaches [160, 161, 154] on this topic have achieved impressive success, to generate 256×256 images from captions, they commonly resort to coarse-to-fine scheme and associate multiple discriminators in different stages of the networks. Such a strategy is both inefficient and prone to artifacts. Motivated by the above findings, we propose an end-to-end network that can generate 256×256 photo-realistic images with only one discriminator. We fully exploit the hierarchical information from different layers and directly generate the fine-scale images by adaptively fusing features from multi-hierarchical layers. We quantitatively evaluate the synthesized images with Inception Score [105], Visual-Semantic Sim-

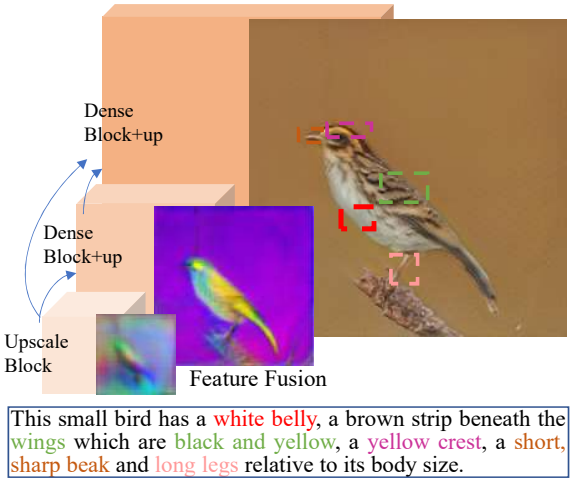


Figure 3.1: Overview of our adversarial network, which fuses the feature maps in three stages and synthesizes the output image at the last stage. Only one discriminator is used to evaluate the final output. Fine details in the output image are highlighted and colour matched with key words in the input text description.

ilarity [163] and average training time on the CUB birds [139], Oxford-102 flowers [84], and COCO datasets [63]. The results show that our model is more efficient and noticeably outperforms the previous state-of-the-art methods.

3.1 Introduction

Generating photorealistic images from text descriptions is a challenging problem that has many applications in computer vision. Current research based on Generative Adversarial Network (GAN) has shown promising results for mapping from natural language feature space to image feature space [100, 160, 161, 154]. Reed et al. [100] first proposed a GAN-based text-to-image synthesis approach, but the generated images are small (64×64) and lack of details. To overcome this limitation, Zhang et al. [160] proposed StackGAN, which

appends an additional GAN to generate low-to-high resolution images. While capable of synthesizing more details, this method needs to train two separate GANs. Later, Zhang et al. further extended StackGAN into StackGAN++ [161], which uses an tree-like structure to progressively generate images of three sizes: 64×64, 128×128 and 256×256. Based on StackGAN++, AttnGAN [154] introduces a deep attention model which can use word level information in the generative network to produce fine-grained images.

An important limitation of StackGAN, StackGAN++, and AttnGAN is the needs to train multiple generators and discriminators. For example, to generate 256×256 images, three discriminators are needed corresponding to three generative stages. This is not only computationally inefficient, but is also prone to artifacts. That is, the generator-discriminator pair at the coarse state may synthesize imperfect results due to the lack of detailed information. Since the output of a coarser level is used to constrain the result generation at finer scales, these imperfections are refined, but not eliminated, in the final result; see Figure 3.8 for example.

In this chapter, we propose a novel end-to-end framework called Hierarchically-fused Generative Adversarial Network (HfGAN), which can synthesize fine-scale images with only one discriminator in the whole network. We adopt Deep Attentional Multimodal Similarity Model (DAMSM) of AttnGAN [154], which transforms a text description to a sentence condition and two word conditions. The generative model generates images through three stages (see Figure 3.1), corresponding to one upscale block in low feature space and two dense blocks with upscale operations in high feature space, respectively. We extract multi-scale global features from different stages and adaptively fuse them together. The input and output of each dense block are directly added together to locally fuse the feature maps. Compared to existing methods, our approach makes full use of the available

convolutional layers and avoids degradation problem with skip connections. Furthermore, deep hierarchical features are used for text-to-image synthesis, allowing features extracted from coarse layers to guide the generation of fine-scale images.

Qualitative and quantitative evaluations on standard datasets [84, 139] demonstrate the advantages of the proposed approach over the current state-of-the-art methods.

3.2 Related Work

In the past few years, deep generative networks based methods have significantly advanced the field of image synthesize. Kingma [56] used stochastic backpropagation to train variational autoencoders (VAEs). The DRAW model of Gregor et al. [30] generates images by an attentional mechanism with a RNN. PixelRNN [134] has generated nice synthetic images by using neural networks to transfer pixel space to a conditional distribution. In addition, Generative Adversarial Networks (GAN) [26] and its variants [105, 78] have achieved impressive results in image modeling such as synthesis [83], image-to-image translation [49], image style transfer [59], image super-resolution [58], etc. Recently, high resolution image synthesis from text descriptions has been an interesting topic of GANs. Basic GAN models tend to generate small or low quality images. Reed et al. [100] first adopted conditional GAN to generate impressive images of size 64×64 from captions. Reed et al. later proposed GAWWN [101], which concatenates auxiliary conditions of part locations. Conditional GAN(CGAN) [79] takes special class tags as conditions to generate higher resolution images. Odena et al. [86] synthesized images with additional classifiers and generated 128×128 images that have good global coherence. S^2-GAN [143] and LAPGAN [18] focus on iteratively refining images by stacking multiple GANs. In each level of Laplacian

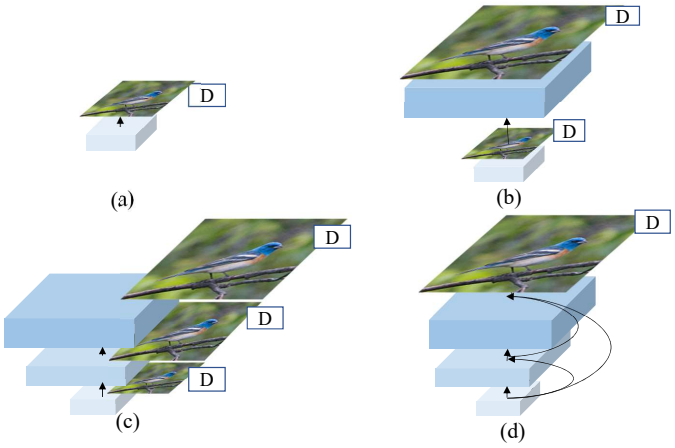


Figure 3.2: Comparison among different network models: (a) Pioneer work on text-to-image Synthesis [100] uses a single feature map and can only generate low-resolution images. (b) Stacked image pyramid approaches [160, 18] train multiple generators to synthesize images at different resolutions. The output of the generator at a coarser scale is fed to the generator at a finer scale and a discriminator is trained at each scale. (c) The hierarchically-nested framework [163] uses single-stream generator with hierarchically-nested discriminators. The coarse-scale generated images are no longer fed into the generator, but they are still needed by nested discriminators (d) Our proposed end-to-end pipeline fuses features from different hierarchies and uses only one discriminator.

Pyramid, LAPGAN trains a separate generative model for coarse-to-fine image generation using residual images as conditions. Similarly, Zhang et al. [160] took low level generated images as input conditions to the next GAN model. To get higher resolution images, Progressive GANs [53] gradually add symmetric layers to the generator and discriminator. Zhang et al. [161] further improved StackGAN to StackGAN++ by jointly training multiple generatiors and discriminators. This approach generates compelling images of three sizes: 64×64, 128×128 and 256×256, and is more stable than StackGAN. Xu et al. [154]

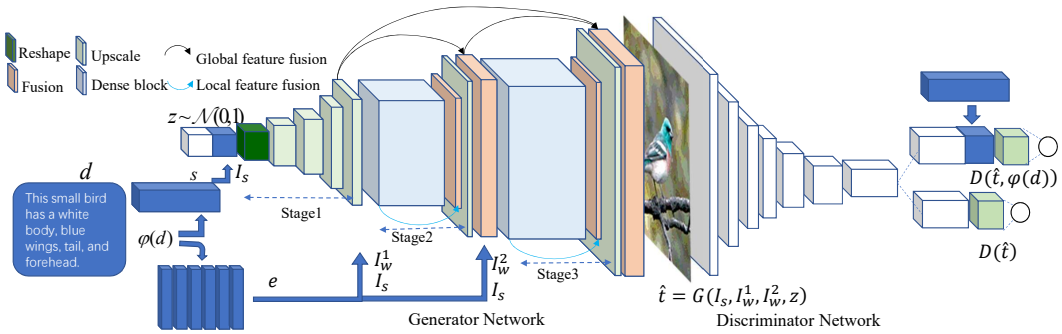


Figure 3.3: Our Hierarchically-fused Generative Adversarial Network

proposed an attention-driven model to get fine-grained images. Zhang et al. [163] proposed Hierarcahilly-nested structure to better adapt discriminators in multi-levels.

To generate detailed images, the previous advanced models all involve multiple discriminators to iteratively refine images. As a result, they are cumbrous and easily result in image collapse because of the inconsistency of these networks. Figure 3.2 highlights the differences between the presented approach and representative models [100, 160, 18, 163].

3.3 Deep Attentional-Text Condition

Text embedding is to learn a correspondence function between text and images. Akata et al. [1] first proposed Structured Joint Embedding and Reed et al. [99] improved it by using the inner features generated through deep neural encoders, and this method is widely adopted by the following text-to-image researches [100, 101, 160, 161, 163]. These methods take whole encoded sentence vector as the condition when generating images. Xu et al. [154] proposed word-level attentive text embedding which highlights more details. We adopt a similar approach as Xu et al. [154]: to pre-train a text encoder for generating sentence and word vectors, then transform words vectors into text condition through attention

model.

$$h_0 = G_0(z, E)$$
$$h_j = G_j(h_{j-1}, A_j^{attn}(e, h_{j-1})) \tag{3.1}$$

where z is random noise, E represents the conditioning vector converted from global sentence features [160]. $A_j^{attn}(e, h_{j-1})$ is the attention model at j^{th} stage of the generative network, which takes all word features $e \in R^{D \times N}$ and image features $h \in R^{\hat{D} \times N}$ from previous hidden layers as inputs. G_j is generative neural network at j^{th} stage. Calculating weighted mean $A_j^{attn}(e, h_{j-1})$ of e based on h_{j-1} can get the relevance between one word and its best matched sub-region. To better use of global sentence semantic information and local words information, we take both whole sentence features and separate word features as the conditioning augmentation at hidden feature layers.

3.4 Methodology

3.4.1 Hierarchically-fused generative adversarial network

As discussed in Section 3.2, existing GAN-based methods show different ways of combining generators and discriminators for text-to-image synthesis. Zhang et al. [161] and Xu et al. [154] utilize three pairs of generators and discriminators to enhance the synthesis. Their multi-stage training strategy has a notable limitation. As shown in Figure 3.8, many synthesized objects contain artifacts (e.g. birds have extra eyes and beaks, and objects have unnatural shapes) if the features learned in lower level are inaccurate. In inconsistent generative networks, misjudgment by low levels discriminator D will have enormous influence on the ultimate image generation. Besides, multiple G-D training processes lead to high computational cost and slow down the whole network.

To address these limitations, we propose an end-to-end architecture (Figure 3.3) which merges features from low-resolution space with high-resolution space via a bottom-up feature fusion approach. Lower-resolution feature space can also utilize the knowledge from the single discriminator D placed at the end of high resolution layer. The proposed hierarchically-fused generative network (see details in Subsection 3.4.2) is denoted by G. G is a convolutional neural network which consists of four hidden states in three stages: shallow feature generation, local feature fusion, global feature fusion, and a series of up-scale operations. There are multiple inputs and outputs from these states. Specifically,

$$\begin{aligned} s_i, e_i &= \varphi(d_i) \\ I_s^i &= F_{ca}(s_i) \\ I_w^{i,j} &= A_j^{attn}(e_i, F_j^i) \\ (F_1^i, ...F_{j+1}^i), \hat{t}_i &= G(I_s, I_w^{i,j}, z), j \in \{1, 2\} \end{aligned} \tag{3.2}$$

where d_i is text data corresponding to the i^{th} image t_i. $\varphi(\bullet)$ is a text encoder in DAMSM designed by Xu, et al [154]. F_{ca} is Conditioning Augmentation [160], which converts sentence vector $s_i \in R^D$ to global conditioning sentence vectors I_s^i. I_w is conditioning words vectors computed by the attention model A^{attn}, which has two inputs: word vectors e and image features F^i of current layer. The j^{th} conditioned word and sentence vector matrices concatenate the last hidden layer of j^{th} stage. $z \sim \mathcal{N}(0, 1)$ is randomly generated noise prior. G generates corresponding synthetic image $\hat{t}_i$. In our method, a training example is an image-text pair (t^i, d_i), I_s^i first joins with z^i to extract initial shallow features and then connects with $I_w^{i,j}$ at next two stages for high-level features generation. $F_1^i, ...F_{j+1}^i$ are feature maps generated from hidden layers at different stages of G. Each F_j merges with the next F_{j+1} for global feature fusion.

Discriminator D has two training objectives: to evaluate whether the input image is

real or fake, and to classify whether an image-text condition pair matches or not [161]. Our discriminator loss function is a combination of conditional loss and unconditional loss. We feed the discriminator with five types inputs:

t_i: real image;

$\hat{t}_i$: generated image;

$(t_i \,|I_s^i)$: real image with corresponding matching text;

$(\hat{t}_i \,|I_s^i)$: generated image with matching text;

$(t_i \,\big|\, \hat{I}_s^i)$: real image with mismatching text.

The discriminator loss function is defined by

$$\begin{underbrace{\mathcal{L}_D = \underbrace{-\frac{1}{2}\sum_i \log(D(t_i)) - \frac{1}{2}\sum_i \log(1 - D(\hat{t}_i))}_{\text{unconditional loss}}}_{\text{conditional loss}} \\ \underbrace{-\frac{1}{3}\sum_i \log(D(t_i \,\big|\, I_s^i)) - \frac{1}{3}\sum_i \log(1 - D(\hat{t}_i \,\big|\, I_s^i)) - \frac{1}{3}\sum_i \log(1 - D(t_i \,\big|\, \hat{I}_s^i))}_{\text{conditional loss}}} \tag{3.3}$$

Our generator loss function is a contextual loss which consists of conditional loss and unconditional loss. For i^{th} training example, the adversarial loss for G is:

$$\mathcal{L}_G = \underbrace{-\frac{1}{2}\sum_i^{j=1,2} \log(D(G(z^{(i)}, I_s^i, I_w^{i,j})))}_{\text{unconditional loss}} \underbrace{-\frac{1}{2}\sum_i^{j=1,2} \log(D(G(z^{(i)}, I_s^i, I_w^{i,j}), I_s^i))}_{\text{conditional loss}} \tag{3.4}$$

As mentioned above, to effectively use words information, we adopt the Deep Attentional Multimodal Similarity Model (DAMSM) [154] in our generative model. Therefore, the overall objective is the weighted sum of generator loss and image-text matching loss $\mathcal{L}_{DAMSM}$ (please refer to [154] for more details of DAMSM).

$$\mathcal{L} = arg\min_G \max_D V(D, G, I_s, I_w, z) = \mathcal{L}_G + \lambda \mathcal{L}_{DAMSM} \tag{3.5}$$

3.4.2 Feature Fusion

Lower-resolution feature maps are spatially coarser, but semantically more accurate. They reflect the overall structure and colour distribution of an image. Up-sampling these features

in a lateral pathway and merging them with feature maps generated via main pathway help to gather different receptive fields. Addition and concatenation are two commonly used feature fusion methods. Inspired by ResNet [37], we divide addition fusion method into "identity addition" and "weighted addition" performed by "shortcut connection" [37]. "Shortcut connection" can be regarded as "identity mapping" or "weighted fusion" layers inside or between different blocks. In each dense block, every layer is concatenated to subsequent layers to preserve local feature information. The input and output of current dense block are fused together by identity mapping since they have the same size of feature dimension. Each residual learning across different hierarchical stages is considered as the weighted fusion. The output of each stage attaches an up-sample layer and a 3×3 convolution to match dimensions of feature maps in other stages, forming a multi-feature addition fusion layer, and thus supporting contiguous state transition. Local and global residual feature fusions contribute to the full use of the hierarchical features.

As shown in Figure 3.4, the proposed generative network can be divided into three stages. Stage 1 serves to extract shallow feature maps F_1 from the sentence vector I_s

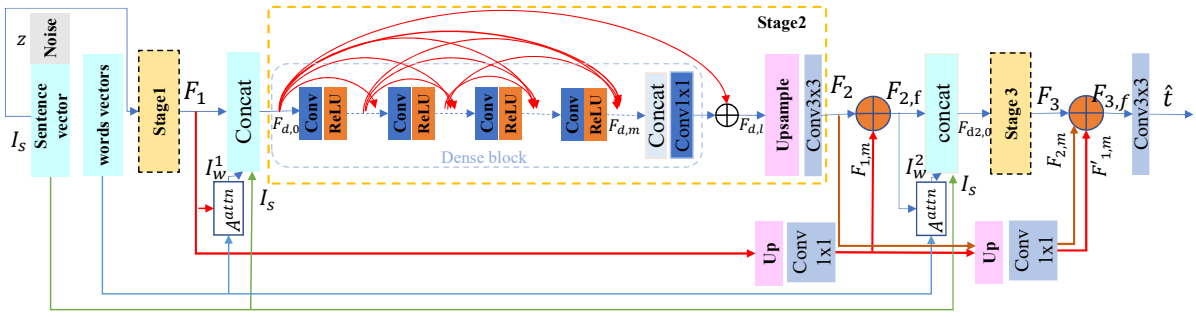


Figure 3.4: Proposed Generative Network Structure with feature fusion.

concatenated with z.

$$\begin{aligned} F_1 &= H_{s1}(z, I_s) \\ F_{d,0} &= [F_1, I_s, I_w^1] \end{aligned} \tag{3.6}$$

where H_{s1} represents four upscale operations. F_1 will be used for global feature fusion by residual learning. $F_{d,0}$ denotes the concatenation of text conditions and feature maps produced by stage 1, and it is applied on further feature generation. Inspired by DenseNet [41], Stage 2 and Stage 3 adopt dense blocks. There are three advantages to use dense blocks in our hierarchical network: to alleviate the vanishing-gradient problem, facilitate feature transfer, and make better use of features in different layers.

$$\begin{aligned} F_2 &= H_{dup}(H_{dense}(F_{d,0}) + F_{d,0}) \\ F_{2,f} &= F_2 + H_{sup1}(F_1) \\ F_{d2,0} &= [F_{2,f}, I_s, I_w^2] \end{aligned} \tag{3.7}$$

where H_{dens} denotes the operation of dense block inside Stage 2, and H_{dup} represents the up-sample layer with 3×3 stride 1 convolution. F_2 is lateral output feature maps. Every layer inside dense block has direct connections to the subsequent layers of current dense block to pass local semantic information. As mentioned above, $H_{dense}(F_{d,0}) + F_{d,0}$ is local feature fusion in an identity addition way implemented by local residual learning in the feed-forward neural networks. This local feature fusion does not add extra parameters or computational complexity. H_{sup1} is lateral upscale process, which consists of a nearest-neighbor up-sample layer, followed by a 1×1, stride 1 convolutional layer. $F_{2,f}$ is one of globally fused feature maps achieved by global residual learning. $F_{d2,0}$ is concatenated result of text conditions and fused feature maps after stage 2. Stage 3 has the same network structure with Stage 2. Thus we can get its output in a similar way by

$$F_3 = H'_{dup}(H'_{dens}(F_{d2,0}) + (F_{d2,0})) \tag{3.8}$$

To exploit hierarchical features in a global way, in addition to global feature fusion $F_{2,f}$ in the middle stage, our generator also adopts global feature fusion after last stage to adaptively fuse features from different hierarchies.

$$F_{3,f} = F_3 + H_{sup2}(H_{sup1}(F_1)) + H_{sup2}(F_2) \tag{3.9}$$

where H_{sup2} has the same upscale architecture as H_{sup1} mentioned above. $F_{3,f}$ is the final fusing output, which is fed into a convolutional layer that has 3 output channels for image generation. We directly fuse features in the weighted addition way by residual learning. This can reduce half of the parameters and computational costs required by the concatenating approach.

3.4.3 Architecture details

Our generator is composed of one upscale block, two dense blocks with upscale layers, two lateral upscale layers and a final convolutional layer. We set all convolutional layers to size 3×3 with padding 0 except in local and global feature fusion, where kernel size is set to 1×1. The initial upscale block extracts shallow features from sentence vectors by changing the size and dimension of feature maps. The initial upscale block has four upscale layers while one upscale layer consists of a nearest-neighbor up-sample and a 3×3 convolution followed by a batch normalization operation and a GLU activation.

Dense blocks in Stage 2 and Stage 3 have the same architecture. Both of them contain four dense layers, and each dense layer has a 3×3 convolutional layer with a ReLU activation. Dense layers are connected together and then fed into a 1×1 convolution for further local feature fusion. Local feature fusion layers, placed after dense blocks and global feature fusion layers, adopt 32 filters, denoted as N_g. Feature maps are gradually transformed

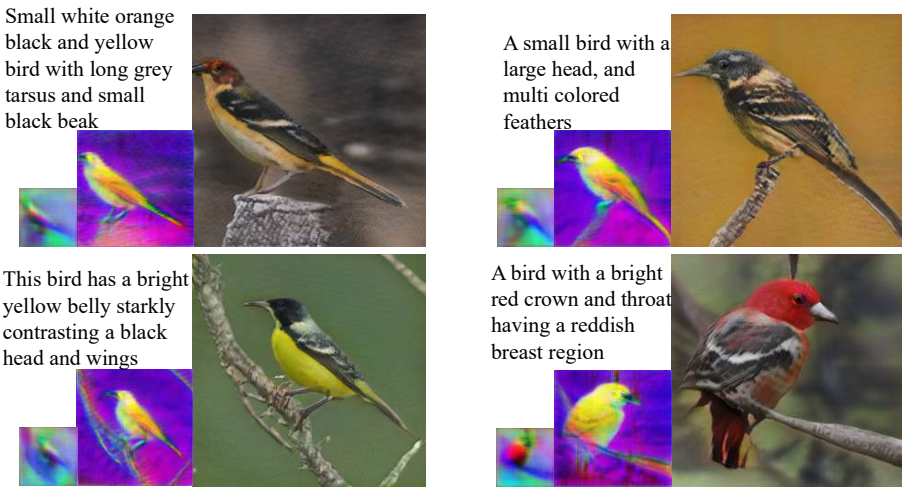


Figure 3.5: Results synthesized using model trained on the CUB dataset. Hidden feature maps from two coarse layers are visualized

from a sentence vector concatenated with a noise vector z to $4 \times 4 \times 16N_g$, $64 \times 64 \times N_g$, $128 \times 128 \times N_g$ and eventually $256 \times 256 \times N_g$ feature tensors via 3 stages of the generator. The last convolutional layer has 3 output channels for compressing feature maps to RGB colour images.

Our discriminator D consists of 3 modules. The initial module transforms input image to a $16 \times 16 \times 8N_d$ tensor by four 3×3 convolutional layers and each layer is followed by a BatchNorm and a LeakyReLU activation. We set filters number N_d to 64. Second module includes two downscale blocks which transform $16 \times 16 \times 8N_d$ to $4 \times 4 \times 32N_d$. The third module contains two identical blocks that transform proceeding maps into a $4 \times 4 \times 8N_d$ tensor. Each block consists of a 3×3 convolutional layer with stride$= 1$ and padding$= 1$.

3.5 Experiments

3.5.1 Datasets and Evaluation Metrics

Datasets We evaluate the proposed method on three widely used datasets: Oxford-102 [84] contains 8,189 flower images belonging to 102 categories; CUB [139] contains 200 bird species with 11,788 images. Both Oxford-102 and CUB contain 10 descriptions. COCO [63] contains 82,783 images in training set and 40,504 images in validation set, and each image has 5 descriptions. We use pre-trained text encoder provided by Xu et al. [154] to encode every sentence into a 256-dimensional sentence embedding vector and 15×256-dimensional words embedding vectors.

Evaluation Metrics Our proposed HfGAN approach is evaluated both qualitatively and quantitatively. Recently, [86, 105] have introduced many new evaluation metrics for GANs. We choose two quantitative measures here: Inception Score [105] and Visual-semantic similarity (VS similarity) [163]. Inception Score is a good evaluation metric to show the discriminability of the generated images. StackGAN provided pretrained inception model for CUB and Oxford-102. For COCO, we use inception model pretrained on ImageNet. VS similarity is applied to measure distance between synthesized images and their corresponding text in feature space. Higher VS similarity score indicates stronger semantic correspondence between the generated image and conditioned text. We use the pre-trained visual-semantic embedding models for three datasets provided by [163].

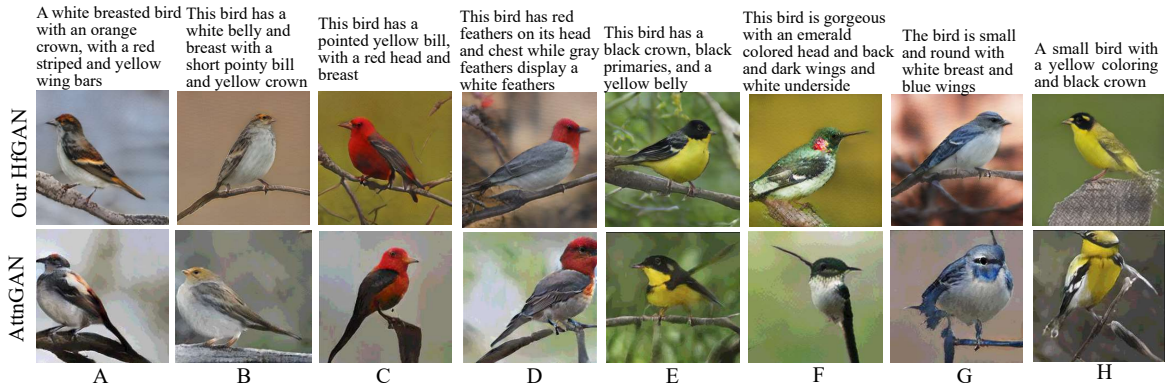


Figure 3.6: Images (256×256 pixels) generated on CUB dataset by our approach (middle) and AttnGAN (bottom) based on input text (top). Our results have much fewer artifacts and demonstrate more semantic details, more natural colour, and more realistic object structure. In comparison, birds generated by AttnGAN have overly fat shapes (A & B), poor head-body proportion (C & D), unnatural body profile (E), artifacts in detailed areas such as the eye (F) and beak (G), or areas not matching the text description (black crown in H).

3.5.2 Results and Comparison

As shown in Figures 3.5-3.10, we qualitatively assess the results generated by our HfGAN. Figure 3.5 shows images generated from input text descriptions using the CUB datasets, along with the visualization results for the middle-layer fused feature maps. The structure and part location information are reflected on the feature maps. It can be seen that our method generates photo-realistic results that are accordant with the input text. Figure 3.6 compares our results to those of AttnGAN [154] on CUB dataset using the same set of text input. It shows that our model generates much more vivid images with better object structures and smoother details, as well as following the input text description more closely.

We also trained HfGAN on Oxford-102 dataset and compared it with HDGAN [163];

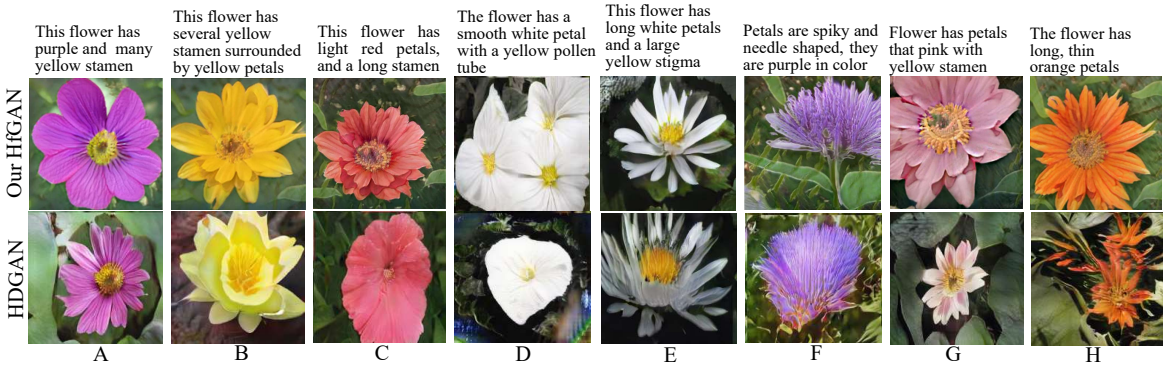


Figure 3.7: Generated 256×256 images on Oxford-102 dataset compared with HDGAN. The flowers in our results have more photo-realistic looking, whereas those generated by HfGAN often have unnatural shape (C & D), unsymmetrical petals (A & F), or poor petal structure (E & H).

see Figure 3.7. Our model generates well-structured and natural looking flowers. Figure 3.8 further compares the two approaches on four cases, where the results of AttnGAN show obvious artifacts (e.g. extra heads, eyes, beaks, or missing claws), whereas our approach provides well-structured results. Finally in Figure 3.9, we compare our results on the COCO dataset with those reported in publications[154, 163], respectively. The results shows that our approach can better learn global coherent structures and is capable of generating complex scenes.

Furthermore, to illustrate the effect of the global fusion, we visualized F_2 and $F_{2,f}$, which are the feature maps before and after fusing with the lateral output F_1 of Stage 1; see Eq. (8). As demonstrated in Figure 3.10, global feature fusion adds more details to the feature map, which indicates the effectiveness of the proposed fusing-strategy.

Table 3.1 shows the Inception score tested by our model compared with GAN-INT-CLS [100], GAWWN [101], StackGAN [160], StackGAN++ [161], TAC-GAN [16], At-

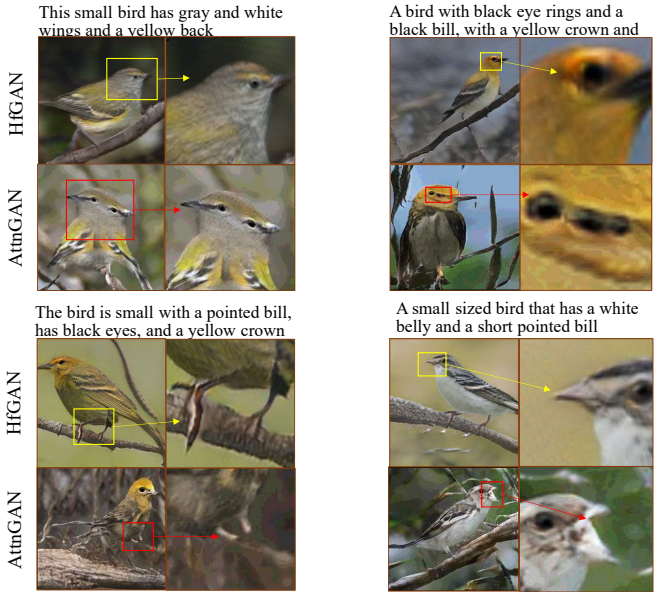


Figure 3.8: Further comparison between HfGAN and AttnGAN on the same set of input text descriptions. The zoomed-in views shows obvious artifacts in AttnGAN results, whereas our results are well-structured.

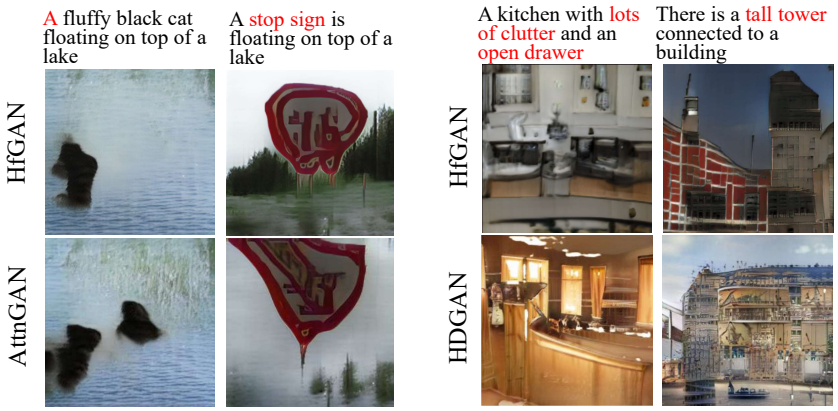


Figure 3.9: Comparison between the images synthesized by our approach on the COCO dataset with those provided in AttnGAN [154] (left) and HDGAN [163] (right). Although not perfect, our results have correct number of cat and less distorted stop sign shape than AttnGAN. It also follows the text description (e.g. drawer and tower) better than HDGAN.

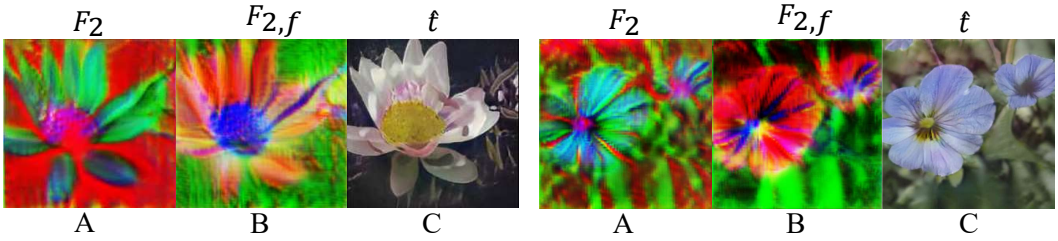


Figure 3.10: Comparison between feature maps before and after the global fusion process. The fused feature maps (B) contain more details and smoother edges than those before fusion (A), which contributes to realistic and detailed synthesis results (C).

Table 3.1: Inception scores on the three datasets obtained by previous text-to-image models and our HfGAN. The scores of existing approaches are reported in the respective publications. The highest scores are shown in **bold**.

Method	Dataset		
	Oxford-102	CUB	COCO
GAN-INT-CLS [100]	$2.66 \pm .03$	$2.88 \pm .04$	$7.88 \pm .07$
GAWWN [101]	/	$0.62 \pm .07$	/
StackGAN [160]	$3.20 \pm .01$	$3.70 \pm .04$	$8.45 \pm .03$
StackGAN++ [161]	/	$3.84 \pm .06$	/
TAC-GAN [16]	$3.45 \pm .05$	/	/
AttenGAN [154]	/	$4.36 \pm .03$	$25.89 \pm .47$
HDGAN [163]	$3.45 \pm .07$	$4.15 \pm .05$	$11.86 \pm .18$
Our HfGAN	$\mathbf{3.57 \pm .05}$	$\mathbf{4.48 \pm .04}$	$\mathbf{27.53 \pm .25}$

Table 3.2: The VS similarity score on the three datasets by previous model [160] [163] and our models

Method	Dataset		
	Oxford-102	CUB	COCO
StackGAN [160]	$.278 \pm .134$	$.228 \pm .162$	/
HDGAN [163]	$.296 \pm .131$	$.246 \pm .04$	$.199 \pm .183$
Our HfGAN	$\mathbf{.303 \pm .137}$	$\mathbf{.253 \pm .165}$	$\mathbf{.227 \pm .145}$

Table 3.3: Training time (s) / epoch

Dataset	Oxf-102	CUB
AttnGAN [154]	446.02	6891.54
Our HfGAN	**308.57**	**5614.73**

tenGAN [154] and HDGAN [163]. We generate 30,000 images using randomly selected captions for computing Inception score. The generated images are classified using the pre-trained Inception v3 model. Specifically, the probability of the image belonging to each class is predicted. These predictions are then summarized into the inception score. We can see that our HfGAN achieves higher scores than previous state-of-the-art methods on all three datasets. This suggests that HfGAN has higher discriminability. Table 3.2 further compares HfGAN with StackGAN and HDGAN on VS similarity score. It shows that HfGAN has higher scores on all three datasets as well, which suggests our model has better semantic consistency.

We also compared average training time per epoch with AttnGAN on same parameter settings and hardware environment (one GeForce GTX 1080 Ti Graphics Card; batch=10)(Table

3.3). The results show that our method is more efficient.

3.6 Summary

In this chapter, we proposed an Hierarchically-fused Generative Adversarial Network (HfGAN) for efficient text-to-image synthesis. The generative network in HfGAN adaptively fuses features from current and preceding layers based on residual learning. Feature information from middle layers are fully used by local and global fusion. Compared with other models, our HfGAN is better in generating consistent and high-quality images because the hierarchical feature maps' fusion can fully extract and utilize the local and global features. Besides, our end-to-end path generation with exclusive discriminator can effectively address the inconsistency problem that exists in previous state-of-the-art approaches (StackGAN and AttnGAN). The advantages of our model over these approaches are demonstrated through both qualitative and quantitative evaluations on three popular datasets.

Chapter 4

Landmark-Guided Conditional GANs for Face Aging

Face aging is an active research field in multimedia applications, which alters a person's facial photo to the appearance at a different age. Recently, conditional Generative Adversarial Networks (cGANs) have achieved impressive progress on this topic. However, most existing works still have challenges in generating convincing aging appearance while preserving the person's identity due to the following limitations: i) they need long-range sequential labeled faces of the same person for training, which are very rare in existing datasets; ii) they focus on texture changes (i.e., wrinkles) and ignore structural variations related to aging, making them ineffective for handling large age spans; and iii) they preserve personal identity through minimizing the differences between inputs and synthesized results, which leads to blurry artifacts and insufficient variations.

In this chapter, we address the above limitations by proposing a novel Landmark guided Dual-learning cGAN (LDcGAN) with a multi-attention mechanism. An external landmark

attention is introduced for adjusting variations of facial structures and a built-in attention is adapted to emphasize the most discriminative regions relevant to aging. Conditioned with age vectors, the primal cGAN converts input faces to target ages, whereas the dual cGAN inverts the process by feeding the synthesized results back to the original input age scope. This allows LDcGAN to enhance age consistency and to minimize changes that affects the personal identity and the background. Both qualitative and quantitative experiments show that our method can generate appealing results in terms of image quality, personal identity, and age accuracy.

4.1 Introduction

Face aging is a gradual and continuous process over time, which contains rich and complex alterations in facial features. Automatically rendering a given face under different ages has wide applications in areas such as identifying a missing child at different ages [31], special effects in entertainment [21], and person identification [141]. This makes it a hot research topic in both Computer Graphics and Computer Vision.

In the last decade, there are two traditional types of face aging methods: the prototype-based [55, 129, 88, 155] and the physical model-based [57, 97, 98, 120]. In the prototype-based methods, faces are grouped according to different ages and an average face is constructed as the prototype for each age group. Aging patterns between groups are learned and texture difference are transferred for synthesizing an aged face. Faces synthesized this way often lose personal identity information, leading to unrealistic visual results. Physical model-based methods apply a more complex parameter model to describe variations of wrinkles, muscles, hair colours, skin, etc [119, 98, 120]. However, these methods heavily

rely on face aging sequence images of the same individual, which are rare and hard to collect. In addition, these traditional methods learn a dedicated mapping between two input age groups and hence cannot transfer an input image to an arbitrary age group.

With the success of deep convolution networks in image generation [27, 49], Generative Adversarial Networks (GANs) based models become a powerful tool for age progression [67, 93, 142, 162, 146]. Features of generated face images are controlled by the age-condition in the conditional GANs (cGANs) [79], which can dramatically reduce artifacts and produce more appealing aging effects of the rendered images. Zhang et al. [162] proposed a Conditional Adversarial Auto-Encoder (CAAE) framework to learn the reasonable wrinkles and muscle variations. Wang et al. [146] proposed a cGANs-based model with a pre-trained AlexNet to keep identity consistency of generated images and produced promising aging results.

Accurate face transformation based on ages is far from being solved though, due to the difficulties of formulating the complex aging mechanism and the diversity of aging patterns. Related biology studies have shown that facial structures and certain facial parts (e.g., eyes, nose and mouth) evolve through aging [77]. The facial skeleton is generally believed to expand continuously throughout early ages [38]. This is reflected in the progressive increase in certain facial anthropometric measurements with age such as the facial shape and eye socket [6]. During early years (birth to teenager), a person's head shape and eyes location changes dramatically. When the person gets older, the hair colour, muscle distribution, and wrinkle appearance show obvious changes. Labelled data (i.e. face images of the same person at different ages) are rare and time consuming to collect. In the available data, age related variations often accompany different head poses, expressions, and lighting conditions. These lead to the challenge of feature leaning in aging patterns

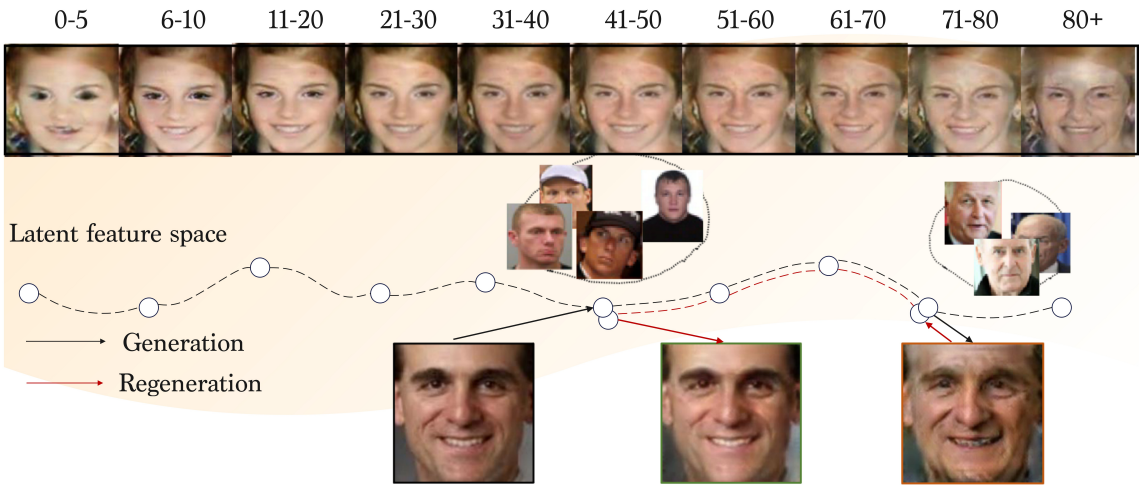


Figure 4.1: Illustration of the symmetric structure that embodies the idea of generation-then-reconstruction. After transferring an input face of age 42 to the target age group of 80+, we use the generated result to reconstruct a face of age group 41-50 and enforce its similarity with the input face.

modelling. Moreover, recent studies seek to improve aging accuracy by face age regression but pay less attention in preserving the identity of generated faces [66].

To tackle the aforementioned problems, we propose a novel Landmark-guided Dual-learning cGAN (LDcGAN) for face aging. The locations of landmark points around facial components and contours are important information for face aging and hence are used to guide the face synthesis tasks. The dual-learning framework in DualGAN [157], CycleGAN [166], and AGGAN [124] have realized significant performance in unsupervised image-to-image translation tasks. Inspired by their work, we apply dual generators and discriminators to enforce constraints on face reconstruction. We take the age which is paired with the input image as a condition and reconstruct the input face image from the generated one. As shown in Figure 4.1, if a face generated by cGAN preserves the person's identity

well and is consistent with target age group, then the reconstruction step should transform the aging results back to the input image.

Specifically, the primal side of our proposed LDcGAN consists of three modules: a cGAN module with built-in attention, a landmark prediction module, and a conditional discriminator. The inverse network structure for face rejuvenation has the same components as above. The generator G (both primal and dual) receives an input image and a target age code and learns an attention-content mask in a similar way as in [94]. The attention mask learns the modified facial regions that underline aging diverse effects, whereas the content colour mask learns pixels which focus on person identity and constant features. The final output of the generator is a combination of the attention and content masks. Since the attention mechanism only modifies the regions relevant to face aging, it preserves the background and the identity of the person well. Furthermore, to supervise long-range age progression, the generator is conditioned not only by age but also by facial landmark code to enhance differences in facial structures, features of sense organs, and poses. To encourage the synthesized faces fall into the target age group, we send the generated aged faces to a pre-trained age classifier and add an age classification loss to the objective. Additionally, the discriminator D consists of an unconditional discriminator and a conditional discriminator on age vector, aiming to make the generated face be more photo-realistic and guarantee the synthesized face lies in the target age group.

The main contributions of this study can be summarized as follows:

i) We propose a novel dual-learning network for face aging, which incorporates both age progression (primal) and face reconstruction (dual) operations. The primal and dual reconstruction processes enhance both the quality of synthesized face images and the preservation of personal identity.

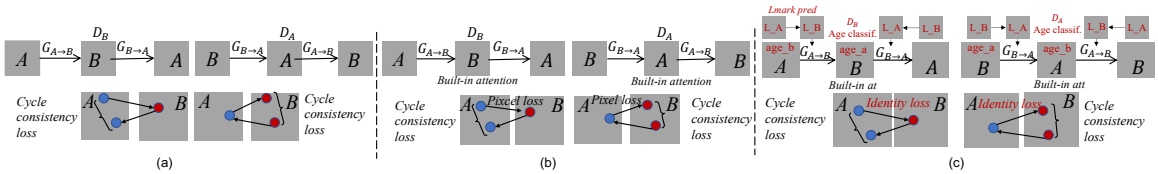


Figure 4.2: Comparing the proposed framework LDcGAN (c) with previous dual-learning frameworks such as CycleGAN/DualGAN [166, 157] (a) and AGGAN [124] (b). Besides the built in attention-content masks in AGGAN, our LDcGAN takes ages as conditions and highlight the facial structure transformation via external landmark attention module. D_A and D_B are conditional discriminators which aim to render facial images with improved age accuracy. Finally, for better preserving personal identity, we employ identity loss and cycle-consistency loss.

ii) An external landmark attention mechanism is designed and applied in the generator, which helps to generate faces with more precise facial structure and poses when applying aging over large age spans.

iii) A built-in attention-content mechanism is highlighted to better reflect the dynamic regions relevant to face aging, allowing aging accuracy and identity consistency be simultaneously achieved. Hence, the ghost artifacts between adjacent age groups can be significantly reduced.

iv) Extensive qualitative and quantitative experiment results on the UTKFace dataset demonstrate the ability of the proposed method in rendering effective aging results in terms of image quality, personal identity, and age accuracy.

4.2 Related work

4.2.1 Face Aging

As mentioned above, early face aging methods can be split into physical modeling methods [120, 126] and prototyping methods [55, 129]. Suo et al. [120] mechanically highlighted the revolution of physical features such as muscles, wrinkles, and hair. Todd et al. [130] modelled face growth by revised cardioidal strain transformation in a computable geometric progress. These modelling-based algorithms are computational expensive and rely on large amount of paired age-image sequences of the same person.

The prototype methods, on the other hand, construct a texture transformation between different age groups. The aging features can be presented differently by averaging the faces as prototypes in the same age group [21, 55]. In order to generalize, these methods discard identity information, resulting in unrealistic facial appearance. Shu et al. [111] and Yang et al. [155] adopted sparse representation to emphasize personalized attributes to some extent. However, this method often produces ghosting artifacts in the synthesized faces.

Recently, Wang et al. [142] propose a recurrent face aging framework, which can model the intermediate transition states, thus the face growth between adjacent age groups is more smooth. It requires sequential face images of the same person at different ages, which limits its applications. Zhang et al. [162] apply conditional adversarial auto-encoder (CAAE), which transforms the input image to manifold to simulate muscle sagging. The CAAE network gets rid of the requirement of paired training samples while preserving personality. However, reconstructing the face image with the age condition only lacks sufficient restraints. This transformation process ignores more basic information of various images, causing the generated images unnatural. Wang et al. [146] applied cGANs in face ag-

ing and Antipov et al. [2] employed a Local Manifold Adaptation and age normalization to better maintain identity consistency, while the representation ability of discriminator is insufficient. Shu et al. [112] proposed an editing approach to achieve face aging editing task. Although they show some positive results, the rendered faces are blurry. These works all ignore the fact that the facial skeleton structure undergoes dramatic changes with aging from young to old, leading to unsatisfactory aging results when applied over large age spans (e.g., rendering a 90 years old face from a baby face or rendering a 5 years old baby face from a senior person).

To handle the above issues, our method uses the target age condition to generate the aging results, which are then used to reconstruct faces in the original face age group. Facial landmarks are employed to guide both synthesis processes.

4.2.2 Generative Adversarial Network

Goodfellow et al. [27] first introduced Generative Adversarial Networks (GANs) in 2014. Basic GANs consist of two networks: a Generator (G) and a Discriminator (D). G accepts a random noise vector z and learns the data distribution to generate an image $G(z)$, whereas D determines whether $G(z)$ is "real" or not. During training, G strives to generate "real" images, whereas D aims accurately detect them as "fake". Hence, these two networks compete in a two-player minmax game:

$$\min_G \max_D V(D,G) = E_{t\sim\ p_{data}(t)}[\log D(t)] + E_{t\sim\ p_z(z)}[\log(1 - D(G(z)))], \tag{4.1}$$

where V is the overall GAN objective and t is an image. $D(t)$ computes the probability of t being "real", which should approach 1. For $\min_G$, $D(G(z))$ represents the probability that the generated image by G is "real". V will diminish when $D(G(z))$ grows. For $\max_D$,

the better D's ability is, the higher $D(t)$ should be, thus $D(G(z))$ should be lower, and $V(D, G)$ will be bigger.

Radford et al. [96] extended GANs to Deep Convolutional GAN (DCGAN), which boosts the application range of GANs in more tasks. To better control the output from a prior distribution, Mirza et al. [79] proposed conditional GAN (cGAN) by applying additional conditions like labels. Kingma et al. [56] introduced the variational autoencoder (VAE) and Makhzani et al. [73] proposed the adversarial autoencoder (AAE). Based on AAE and cGANs, Zhang et al. [162] proposed a Conditional AAE (CAAE) for face aging and successfully simulate identity manifolds, but it is difficult to access a large labelled database. Pyramid GAN proposed by Deton et al. [18] can generate samples through a coarse-to-fine strategy. More recently, DualGAN and CycleGAN [166, 157] promote dual learning and perform successfully in many image-to-image transform tasks.

Similar to DualGAN/CycleGAN, our method first transfers original faces into the desired age groups and then renders the aging results back to the age groups of the original faces. The reconstructed faces should therefore be as similar to the input faces as possible. Figure 4.2 shows the differences between previous classic works (CycleGAN [166, 157] and AGGAN [124]) and the proposed LDcGAN. The paired generators in the proposed LDcGAN have both a built-in content attention module and an external landmark attention module. The built-in attention module can disentangle the modified regions relevant to face aging [94]. The external attention module models facial skeleton changes as a facial landmark change problem. With the guidance of landmarks under different ages, the network can better synthesize facial structures.

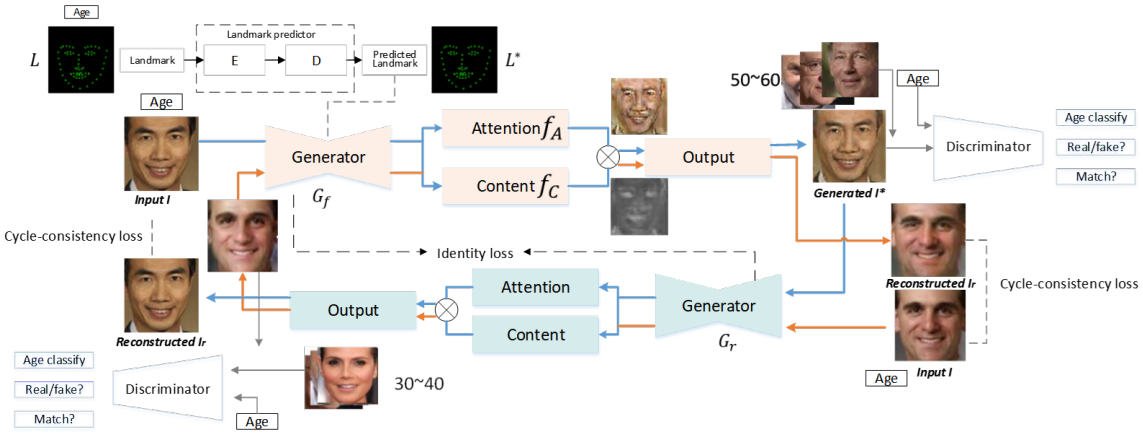


Figure 4.3: The pipeline of the proposed LDcGAN for face aging. The blue flowchart shows the primal cGAN G_f for aging face generation and the red flowchart shows the dual cGAN G_r for face reconstruction. G_f and G_r share parameters W_G. Based on the target age condition, an input face image is first transformed to the target ages before being reconstructed based on the initial input age condition.

4.3 Methodology

It is important to find a mapping between age features and face features for inferring faces with desired ages. This problem can be considered as a cGAN problem and solved by minimizing the distance between ground truth face distributions in target age groups and generated ones, while preserving the person identity at the same time.

We divide multiple sets of training faces with different ages into N non-overlapping groups: $\mathbb{I}_1, \mathbb{I}_2, \ldots, \mathbb{I}_N$ ($N = 10$). The images in different age groups can belong to different persons. These 10 groups correspond to age 0-5, 6-10, 11-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80 and 80+, respectively. Our goal is to learn an age progression model G_f for the primal task and a face reconstruction model G_r for the dual task. G_f and G_r have the same structure and share same parameters.

Assume the input facial image I belongs to age group $A_s \in \Re^{h \times w \times 10}$ and the target age condition is $A_t \in \Re^{h \times w \times 10}$, where h and w indicate the height and width of an intermediate feature map, 10 is the number of age groups. Similar to a 10-dim one-hot age vector, we reshape the vector as a 10-channel tensor and each channel represents a specific age group. Model G_f aims to generate a synthesized face $I^* = G_f(I, A_t)$ of the input person that falls into target age group A_t. The generated face I^* should not only belong to target age group A_t but also preserve the original person's identity of the input face image I. The regenerated face $I_r = G_r(G_f(I, A_t), A_s)$ should be as close to the input face I as possible and it should lie in the input age group A_s.

A high-level illustration scheme of our proposed architecture is shown in Figure 4.3. The detailed components are described in subsection 4.3.1 and the objective functions are presented in subsection 4.3.2.

4.3.1 Network Architecture

As shown in Figure 4.3, our primal face aging network consists of three main modules: i) a generator G_f that is trained to synthesize a face I^* with target age A_t; ii) an associated discriminator D that aims to make I^* looks realistic and drive I^* lie in target age group A_t; and iii) target aging face landmark predictor that works as attention mechanism for face generation. The dual procedure, which regenerates face image I_r to match the original face I, has the same three modules as the forward face aging network.

Landmark attentions: Most of the existing works [146, 142] have limited ability for handling facial structure changes and hence can handle age progression between adults only (e.g., between 20 and 80 years old). To overcome this limitation, we pretrain an external

landmark attention model as a landmark predictor, which is imposed on the internal multi-scale features; see Figure 4.3.

We first retrieve a source landmark feature $L \in \mathbb{R}^{2\times n}$ from the input face image I using 2D face alignment [8], where n is fixed at 68 in this work. The pretrained landmark predictor $\mathcal{G}_L$ is then used to convert L to L^* that falls in the target age group. The landmark predictor has an Encoder (E)-Decoder (D) structure. The final landmark prediction is achieved by fusing the age feature vectors with the latent features of the source landmark. The training loss for $\mathcal{G}_L$ is $\mathcal{L}_{lmk} = \|E(L^*) - E(L)\|_2^2 + \left\|L^* - L^{GT}\right\|_2^2$.

The attention mechanism helps to precisely emphasize the variational structure areas, thus producing more sophisticated features in a fine-grained fashion. Specifically, to calculate the latent attention layer as shown in Figure 4.4, we integrate original and predicted landmarks features to the hidden feature layers h_i and h_j in the image generation process. The overall attention features can be formulated as:

$$f_{attn1} = \left(f_c\left(H_l^1\left(L^*\right)\right) - \left(f_c\left(H_l^1\left(L\right)\right)\right)\right) \tag{4.2}$$

$$f_{attn2} = \sigma\left(H_l^2\left(H_l^1\left(L^*\right) \oplus \left(H_l^1\left(L\right)\right)\right)\right) \tag{4.3}$$

where H_l^1 is convolutional encoder to produce latent landmark vectors of landmarks and $\oplus$ is concatenation. σ is Sigmoid activation function. The difference between two latent landmark features forms a first attention maps f_{attn1} with size of $N_L \times 16 \times 16$. H_l^2 is the second convolutional encoder which can encode the concatenated landmark feature values to attention feature maps f_{attn2} with size of $N_L \times 32 \times 32$.

Generator: Basic GAN-based methods [3] for face aging firstly apply numerous down-sampling convolutional layers to learn the high-level feature distributions, and then forward

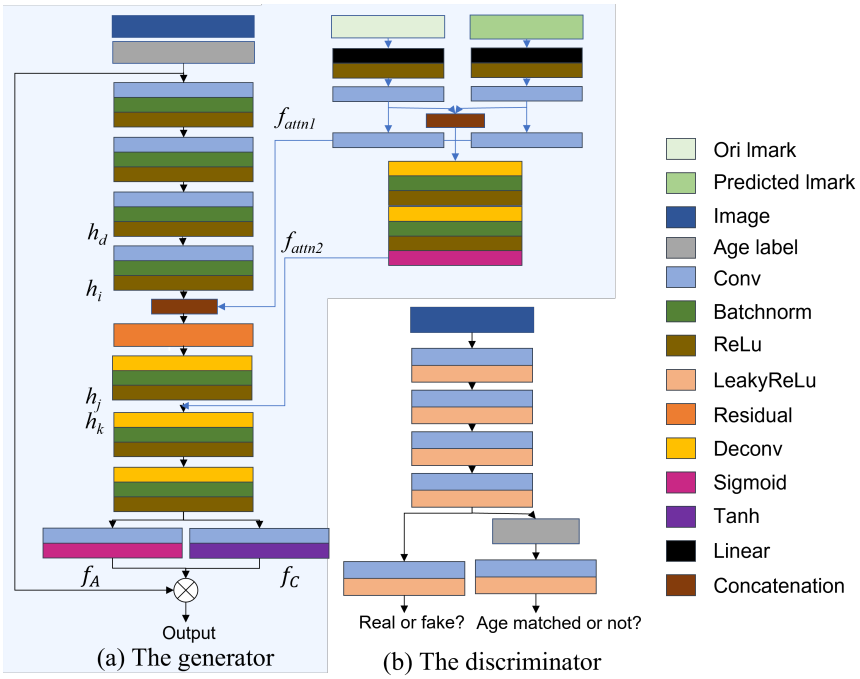


Figure 4.4: The network architectures of the generator and the discriminator. Note that the dimension of age features is 10. (64; 128; 3; 2) denotes that the input channel number is 64, the output channel number is 128, the kennel size is 3, and the stride equals 2.

the feature maps into multiple up-sampling convolutional layers to render the final image. In our work, given an input RGB face image $I \in \mathbb{R}^{h\times w\times 3}$ under an arbitrary age group $\boldsymbol{A}_s \in \mathbb{R}^{1\times 10}$ and a target age group $\boldsymbol{A}_t \in \mathbb{R}^{1\times 10}$, we need to pad the age label $\boldsymbol{A}_t$ into $\mathbb{R}^{h\times w\times 10}$ in one-hot form. Then, we form the input of generator as a concatenation vector $(I, \boldsymbol{A}_t) \in \mathbb{R}^{h\times w\times(3+10)}$. We believe that the distance between predicted landmarks $L*$ and original landmark L can reflect the distance before and after the age transformation in feature space, hence we get the current face feature $h_i \bigoplus f_{attn1}$. Conditioned on f_{attn2}, we get following image feature:$h_k = h_i \cdot f_{attn2} + h_d \cdot (1 - f_{attn2})$.

One key ingredient of our approach is to make G focus on the regions of image that are relevant to face aging and keep the remaining information unchanged to preserve identity consistency. For this purpose, we have embedded a built-in attention mechanism to the generator, which can disentangle the discriminative diverse regions and the consistent part by producing an attention feature mask f_A and a content feature mask f_C instead of regressing a full image. The final generated image can be obtained as:

$$I^* = G\left(\boldsymbol{I}, \boldsymbol{A}_t, L^*, L\right) = (1 - \boldsymbol{f_A}) \otimes \boldsymbol{f_C} + \boldsymbol{f_A} \otimes I \tag{4.4}$$

where $\otimes$ denotes the element-wise product, $f_C = G_r\left(I, \boldsymbol{A}_t\right) \in \mathbb{R}^{h\times w\times 3}$, and $f_A = G_r\left(I, \boldsymbol{A}_t\right) \in \{0, \ldots, 1\}^{h\times w\times 1}$. The mask f_C indicates how much the original image contributes at each pixel location, whereas f_A determines how much the aging condition contributes to the changes of the final image. The whole reconstruction process in training is

$$\mathbf{I} \rightarrow \mathbf{G_f}\left(\mathbf{I}, \mathbf{A}_t\right) \rightarrow \mathbf{G_r}\left(\mathbf{G}\left(\mathbf{I}, \mathbf{A}_t\right), \mathbf{A}_s\right) \tag{4.5}$$

Discriminator: Discriminator aims to distinguish between the generated image (fake) and the ground truth images (real). This combined discriminator D_G consists of both a

unconditional and a conditional part, which promotes the realism of generated faces and guides the results toward the target age group, respectively. The conditional discriminator concatenated with reshaped age condition C_t to the fifth convolution layer, which corresponds the age condition in generator network G_f.

Symmetric face regeneration: As described above, our model includes a semantic face reconstruction module, which maps generated target image back to the original feature space by a dual generation network. We share the weights between the primal and the dual generators, as well as the primal and the dual discriminators, during the training process.

4.3.2 Objective Functions

The defined loss function includes four terms: 1) an adversarial loss proposed by Gulrajani et al. [32], which pushes the distribution of the generated images to the distribution of the training images; 2) an age classification loss for encouraging accurate age identification for the generated facial images using an age classifier; 3) an identity feature loss, which helps to preserve the identity information for the generated fake face samples; and 4) a cycle-consistency loss [166], which further constrains the input and output faces correspond to the same person.

Adversarial loss: The discriminator D_G is trained alternately with the Generator G to avoid being fooled. Similar to the generator, the objective of the discriminators consists of a visual realism adversarial loss and an age-face paired consistency adversarial loss.

Mathematically, it is defined as:

$$\begin{aligned}\mathcal{L}_{D_G} = &-\tfrac{1}{2}\mathbb{E}_{I^{GT}\sim p_{I^{GT}}}\left[\log\left(D_G\left(I^{GT}\right)\right)\right] \\ &-\tfrac{1}{2}\mathbb{E}_{I^*\sim p_{I^*}}\left[\log\left(1-D_G\left(I^*\right)\right)\right] \\ &-\tfrac{1}{2}\mathbb{E}_{I^{GT}\sim p_{I^{GT}}}\left[\log\left(D_G\left(I^{GT},s_t\right)\right)\right] \\ &-\tfrac{1}{2}\mathbb{E}_{I^*\sim p_{I^*}}\left[\log\left(1-D_G\left(I^*,s_t\right)\right)\right]\end{aligned} \tag{4.6}$$

$$\begin{aligned}\mathcal{L}_G = &\,\mathbb{E}_{I^*\sim p_{\text{data}}(I^*)}\left[\log D_G\left([C_t,I^*]\right)\right]+ \\ &\mathbb{E}_{I\sim p_{\text{data}}(I)}\left[\log\left(1-D_G\left([C_t,G_f(I)]\right)\right)\right]\end{aligned} \tag{4.7}$$

where I^{GT} is from the real image distribution p_I^{GT}. s_t is reshaped age condition C_t. G tries to minimize the adversarial loss objective $\mathcal{L}_{D_G}$, while D_G tries to maximize it. The target of G is to produce a facial image I^* that looks similar to the images from GT, while D_G aims to distinguish between rendered face images I^* and real images I. A similar adversarial loss of Eq. 4.6 for inverse mapping is defined as:

$$\begin{aligned}\mathcal{L}_{\mathcal{R}D_G} = &-\tfrac{1}{2}\mathbb{E}_{I\sim p_I}\left[\log\left(D_G\left(I\right)\right)\right] \\ &-\tfrac{1}{2}\mathbb{E}_{I_r\sim p_{I_r}}\left[\log\left(1-D_G\left(I_r\right)\right)\right] \\ &-\tfrac{1}{2}\mathbb{E}_{I\sim p_I}\left[\log\left(D_G\left(I,s_r\right)\right)\right] \\ &-\tfrac{1}{2}\mathbb{E}_{I\sim p_{I_r}}\left[\log\left(1-D_G\left(I_r,s_r\right)\right)\right]\end{aligned} \tag{4.8}$$

$$\begin{aligned}\mathcal{L}_{\mathcal{R}G} = &\,\mathbb{E}_{I_r\sim p_{\text{data}}(I_r)}\left[\log D_G\left([C_t,I_r]\right)\right]+ \\ &\mathbb{E}_{I^*\sim p_{\text{data}}(I^*)}\left[\log\left(1-D_G\left([C_t,G_r(I^*)]\right)\right)\right]\end{aligned} \tag{4.9}$$

Age classification loss: Beside reducing the image adversarial loss, the generator must also reduces the age error by the age classifier D_A. The age classification loss is defined with two components: an age estimation loss with fake images used to optimize G, and an age estimation loss of real images used to learn the age classifier D_A. This loss

$\mathcal{L}_{cls}\left(G, D_A, \boldsymbol{I}, \boldsymbol{C}_t, \boldsymbol{C}_s\right)$ is computed as:

$$\mathcal{L}_{cls} = \mathbb{E}_{\boldsymbol{I}\sim\mathbb{P}_I}\left[\ell\left(D_A\left(G\left(\boldsymbol{I}, \boldsymbol{C}_t\right)\right), \boldsymbol{C}_t\right) + \ell\left(D_A\left(\boldsymbol{I}\right), \boldsymbol{C}_s\right)\right] \tag{4.10}$$

where C_s is the source age condition of the input image I, $\ell(\cdot)$ corresponds to a softmax loss. Similarly, in the dual procedure for face regeneration, the loss function is defined as:

$$\mathcal{L}_{\mathcal{R}cls} = \mathbb{E}_{\boldsymbol{I}^*\sim\mathbb{P}_{I^*}}\left[\ell\left(D_A\left(G\left(\boldsymbol{I}^*, \boldsymbol{C}_t\right)\right), \boldsymbol{C}_t\right) + \ell\left(D_A\left(\boldsymbol{I}^*\right), \boldsymbol{C}_s\right)\right] \tag{4.11}$$

Identity feature loss: Adversarial loss and age classification loss only drive the generator to generate samples that follow the target data distribution. Hence, the generated images may not looks like the same person. To better preserve the personal identity when generating face images, we introduce an identity feature loss. Since the generated I^* consists of many different characters from I in terms of texture, wrinkles, hair, etc., it is not appropriate to calculate the pixel distance between I^* and I. This problem is addressed by using a perceptual loss to shorten the distance between input and output images in the same lower feature space. That is, the identity feature loss is defined as:

$$\mathcal{L}_{\text{id}} = \sum_{I\in p_I(I)} \left\|\Phi(I) - \Phi\left(G\left(I \mid C_t\right)\right)\right\|^2 \tag{4.12}$$

where $\Phi(\cdot)$ represents the features extracted by the conv5 layer in our generative network.

Cycle-consistency loss: As we regenerate the face in a primal-dual loop, the rejuvenated face image I_r should be as similar to the input image I as possible. This is enforced using the following lost function:

$$\begin{aligned}
&\mathcal{L}_{\text{cycle}}\left(G_f, G_r\right) = \\
&\mathbb{E}_{I\sim p_{\text{data}}(I)}\left[\left\|G_r\left(G_f(I)\right) - I\right\|_1\right] + \\
&\mathbb{E}_{I^*\sim p_{\text{data}}(I^*)}\left[\left\|G_f\left(G_r(I^*)\right) - I^*\right\|_1\right]
\end{aligned} \tag{4.13}$$

where the regenerated image $\tilde{I} = G_r\left(G_f(I)\right)$ is compared against the input image I.

Full objective function: To generate desired facial image I^* with the target age and the same person identity, we linearly combine the aforementioned losses:

$$\begin{aligned}\mathcal{L}_G &= \lambda_1\mathcal{L}_G + \lambda_2\mathcal{L}_{\mathcal{R}G} + \lambda_3\mathcal{L}_{\text{cls}} + \lambda_4\mathcal{L}_{\mathcal{R}\text{cls}} \\ &+\lambda_5\mathcal{L}_{\text{id}} + \lambda_6\mathcal{L}_{\text{cycle}}\ \left(G_f, G_r\right)\end{aligned} \tag{4.14}$$

$$\mathcal{L}_D = \mathcal{L}_{D_G} + \mathcal{L}_{\mathcal{R}D_G} \tag{4.15}$$

where $\mathcal{L}_{R\cdot}$ is losses produced by the face regeneration procedure. λ is a set of hyper-parameters that control the relative importance of each term.

4.4 Experiments

We now introduce our implementation details and then evaluate LDcGAN both qualitatively and quantitatively. A large public dataset UTKFace [9] is used in the training and testing, which contains over 20,000 face images between 0 and 116 years old. All images are annotated with age. There are large variations in pose, illumination, expression, and occlusion in this dataset. The data is unpaired and non-sequential because there are no different age photos of the same person. We split UTKFace into two parts, 90% for training and the rest for testing. The number of training images are shown in Table 4.1.

Table 4.1: Image numbers in each age group of the UTKFace dataset.

Age group	0-5	6-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	80+
Numbers	2204	850	1645	7736	4316	2091	2192	1160	676	530

4.4.1 Implementation Details

Basics: Our model is implemented on PyTorch and tested on a single Nvidia GeForce GTX 1080 Ti GPU, and 50GiB memory. We apply batch normalization, set a fixed learning rate of 0.0002, and use Adam algorithm as the optimizer.

Architecture details: The network architectures of the generator G and the discriminator D are illustrated in Figure 4.4. For the generator G, it transforms input RGB image with size $128 \times 128 \times 3$ and age condition to a $16N_i \times 16 \times 16$ tensor by 4 groups of convolution- Batchnorm-Relu layers and 6 residual blocks, where $N_i = 16$ is the depth of latent image feature maps. The stride of each convolution layer is 2 and the kernel size is 3×3. The residual block includes 2 convolution layers, followed by a zero-padding layer, a stride-1 convolution layer, and a batch-normalization layer.

We reshape the age condition as a 10-dim one-hot vector to a 10-channel tensor because ages are divided into 10 groups. The reshaped tensor further concatenates the latent image feature. Each channel of the conditioned tensor indicates a specific age group. Through the pretrained landmark prediction model, we can directly get a predicted landmark with target age group. The predicted landmark vector is then fed into a landmark encoder, which has a linear layer with ReLU activation and a Conv layer, producing landmark features with size $N_i \times 16 \times 16$. Both age condition and landmark condition tensor are embedded to the image feature tensor after the residual blocks. To decode the combined latent feature maps into a single-channel attention mask and a 3-channel content mask, we adopt 3 deconvolution layer groups. All of them have the same basic structure: a deconvolution layer with 3×3 kernels, followed by a batch-normalization layer and a ReLU as the activation function.

Our discriminator D involves two parts. The first part contains four 3×3 Conv layers

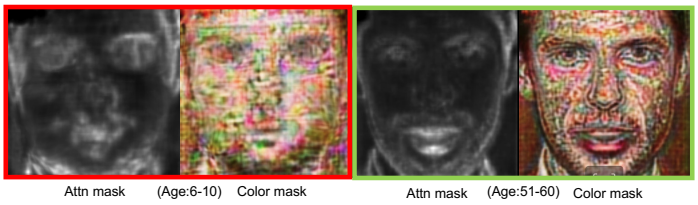


Figure 4.5: Illustration of intermediate attention masks and content colour masks for two input faces. The ones in the red box belong to the 6-10 age group, whereas ones in the green box belong to the 51-60 group.

with BatchNorms and Leaky-ReLU activation, which transforms $3 \times 128 \times 128$ images to $8N_d \times 16 \times 16$ feature maps, where $N_d = 64$. To distinguish different images belonging to different age groups, we also add age condition to our discriminator by concatenating reshaped age condition vector to the output of fourth convolution layer.

4.4.2 Qualitative Comparison

To better demonstrate the superiority in aging accuracy and preserving identity features of our methods, we have compared the two state-of-the-art methods: Conditional adversarial autoencoder (CAAE) [162] and Identity-Preserved Conditional Generative Adversarial Networks (IPCGAN) [146]. For IPCGAN, we first train the age classier on the dataset UTKFace based on AlexNet and other parameters are set according to [162]. For CAAE, we remove the gender information and use 10 age groups for fair comparison.

We select two face images from two different age groups to visualize the attention mask and content mask learned by LDcGANs; see Figure 4.5. We can tell the feature differences between young and old in terms of face shape and key features. This suggests that LDcGANs successfully learns the parts of faces that are relevant to aging.

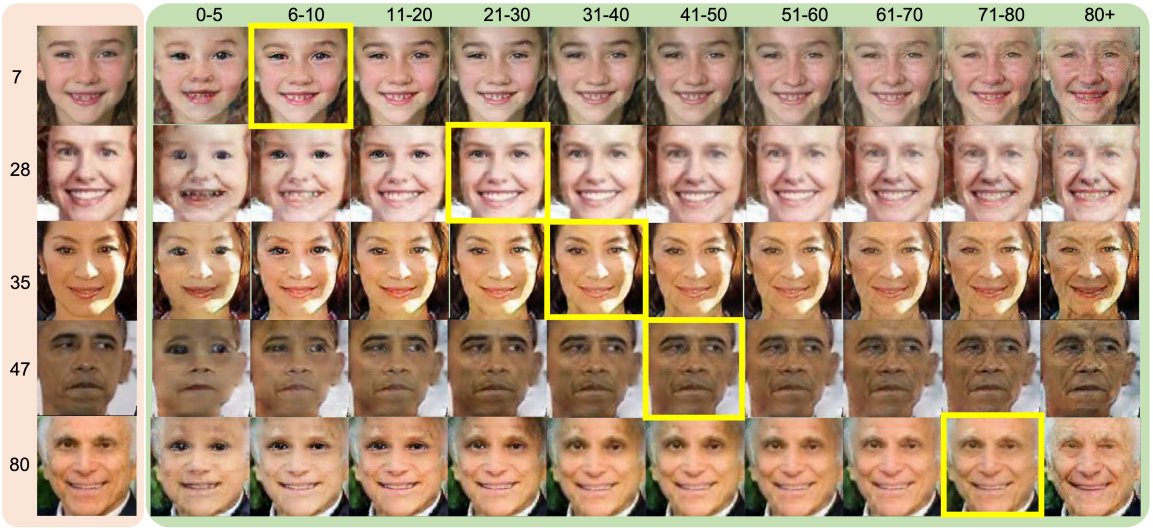


Figure 4.6: Results of our LDcGAN for age progression. The first column shows the original faces marked with their true ages on the left. The remaining 10 columns show the results synthesized for different age target groups (indicated at the top of each column). Generated faces are realistic in aspects of age and facial feature. The ones generated for the respective input age groups (highlighted in yellow boxes) have appealing similarity with the input faces.

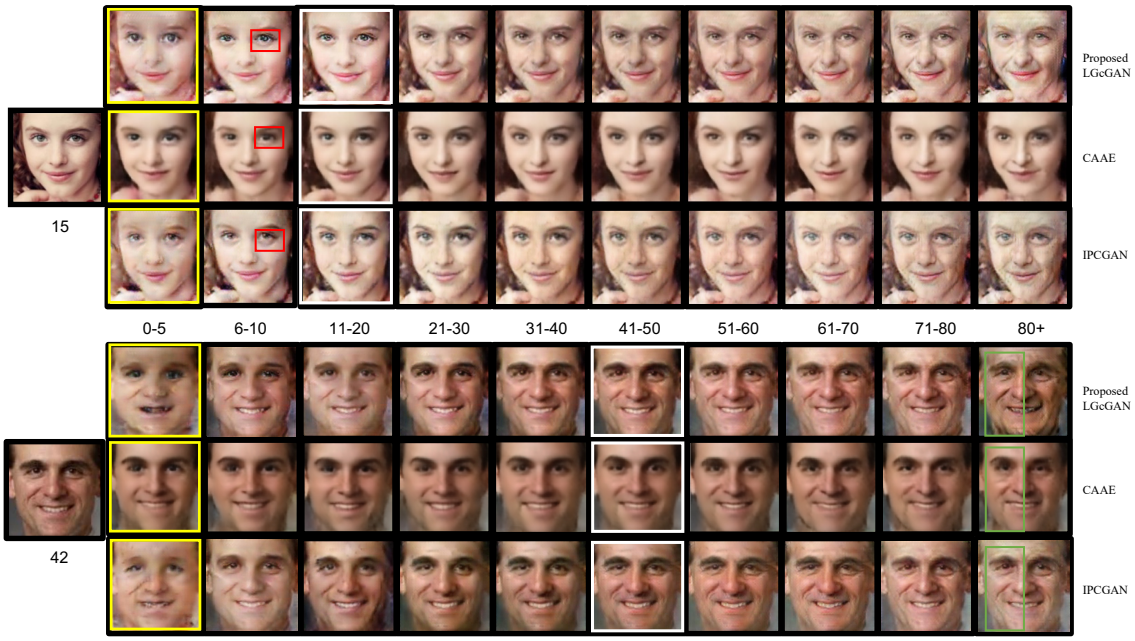


Figure 4.7: Comparisons between results synthesized by our LDcGAN, CAAE [162], and IPCGAN [146]. Two input faces images of different genders and age groups are used (shown on the left. The white boxes highlight the age groups that the initial faces belong to. Yellow boxes show that our method tend to generate rounder eyes and shorter faces at early ages. Overall, faces reconstructed by LDcGAN yield more distinct features and realistic aging effects, see for example areas highlighted green boxes. In comparison, eyes generated by CAAE and IPCGAN highlighted in the red boxes are blurry and have more artifacts.

Synthesized face manifold of our proposed LDcGAN is shown in Figure 4.6. The generated face images not only resemble faces in the target age groups but also well preserve the persons' identity over large age spans. So our method is effective in simulating translations between age groups and synthesizing elderly and young face images with high visual fidelity.

Quality comparison with prior works is provided in Figure 4.7. It shows that LDcGAN generates more visually convincing faces for different ages, providing better age accuracy and identity consistency. In comparison, the CAAE fails to generate the elder looking and the results are overly smooth, whereas IPCGAN lacks the ability to transfer faces to younger looking faces, as eye shapes and facial profiles do not change much. Furthermore, the faces generated using our model for the same age groups as inputs best match to the input faces. The result of CAAE lost the wrinkles on the male's face and the results of IPCGAN introduced unnatural gray colour to the female's face.

Additionally, LDcGAN also performs particularly well if there is a huge age gap between the original and target ages; see Figure 4.8. Such aging task is very challenging due to dramatic facial features changes. As seen in Figure 4.8, CAAE [162] and IPCGAN [146] perform poorly in terms of adjustments on facial structures, eye shapes and positions. Popular commercial apps usually have limited functions. For example, "MakemeOld" [106] provides aging range from 20 to 90 only and merely adds wrinkles to input faces without adjusting facial features. "FaceLab" [118] has two aging patterns, Yong and Old, and produces inconspicuous results. Despite the wide age gap, LDcGAN effectively synthesizes faces that match the target ages and and preserve personal identities.

We further compare local details between our method and IPCGAN [146] in Figure 4.9. The results show that our approach handles details such as earring, teeth, and eyes much

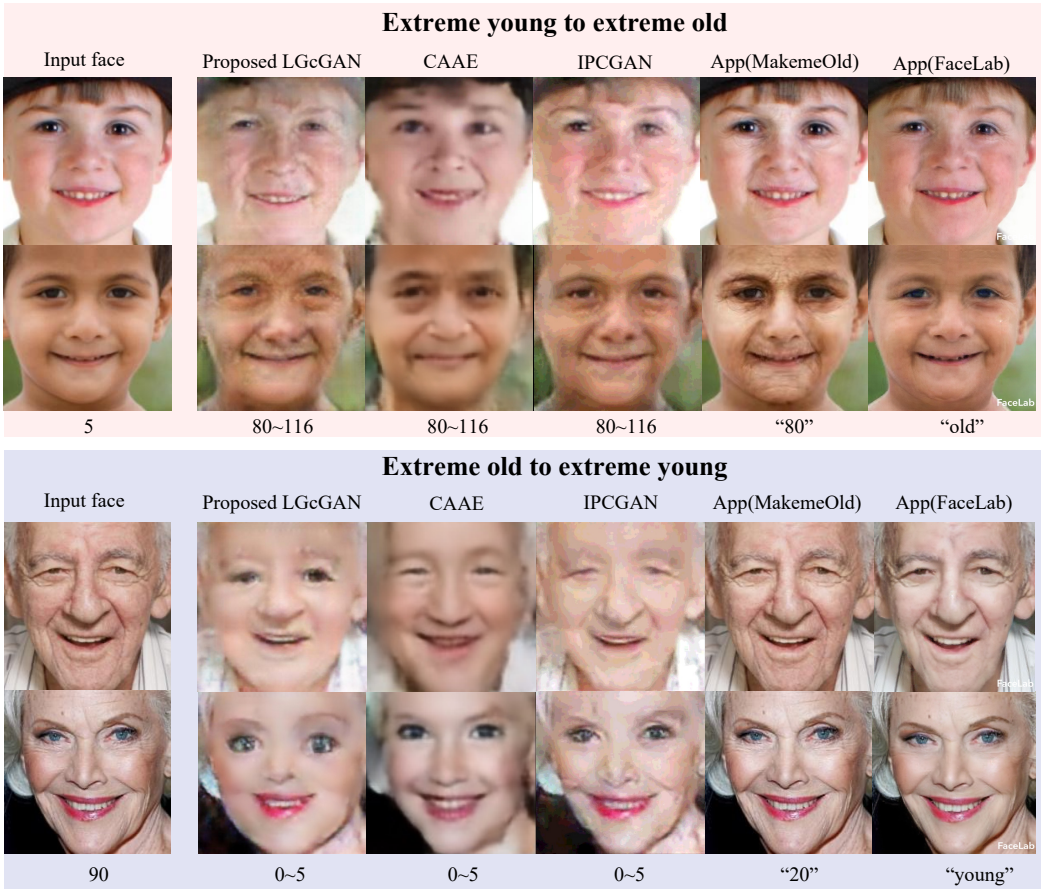


Figure 4.8: Results synthesized by LDcGAN, CAAE [162], IPCGAN [146], and aging Apps (MakemeOld [106], and FaceLab [118]) under large ages. The first row shows the synthesized results from young to very old ages, whereas the second row shows the results from old to very young ages. The corresponding ages are listed below face images. Our LDcGAN produces better results in terms of skin textures and facial structures.

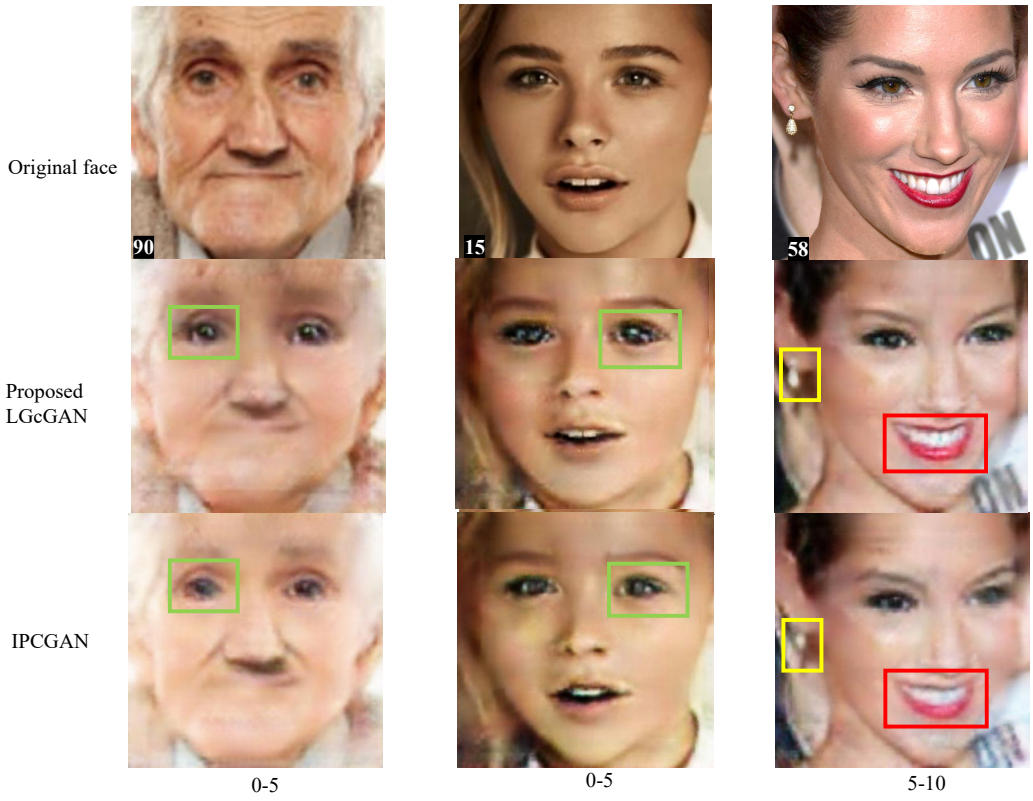


Figure 4.9: Detailed comparisons of our proposed LDcGAN and IPCGAN [146]. Original input images are shown in the top row. Our LDcGAN produces bigger eyes (highlighted in green boxes) and rounder facial structures which match the target age group indicated at the bottom of respective columns) better. The adornments (yellow boxes) and the details (red boxes) are better preserved by our method.

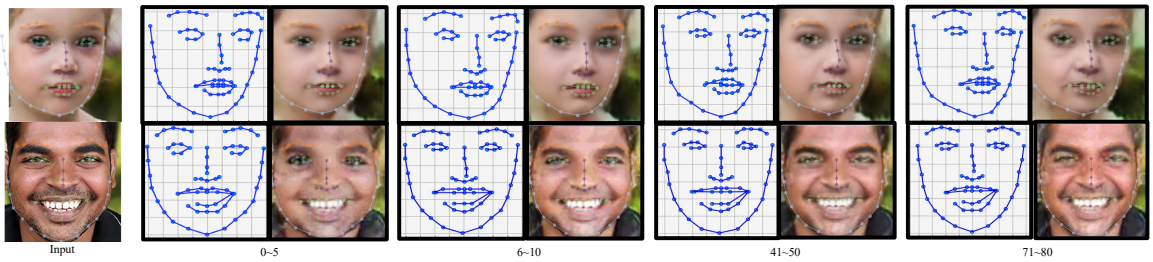


Figure 4.10: Conditional landmark visualization. The input faces are shown in far left and have landmarks retrieved by 2D face alignment [8] displayed on the top. The remaining columns show the landmark prediction results (left) and the corresponding rendered aging faces (right) for different age groups, respectively. One can notice that the eyes are bigger and the faces are chubbier at young ages, suggesting the effectiveness of the proposed landmark attention module.

better. The results are sharper, rich in details, and having higher fidelity.

Facial landmarks can reflect the changes in the facial skeleton that occurs with aging and hence have impact on facial appearance. The landmark changes could be dramatic between children and adults, but much less obvious among young and senior adults because the skeleton itself undergoes minimal changes with aging after the growth and development of a person. Figure 4.10 illustrates the effect of facial landmark prediction. For the two input faces, their facial landmarks are retrieved by 2D face alignment [8]. Changes to these landmark locations are predicted for different age groups, which guide the synthesized faces to follow. Hence, the external landmark attention contributes the accuracy of face progression.

To better validate the advantages of using landmark attention, we also compare our method with IPCGAN [146] in a scenario that the input face depicts a side pose. As show in Figure 4.11, IPCGAN [146] fails to transform the aging appearance such as changes in

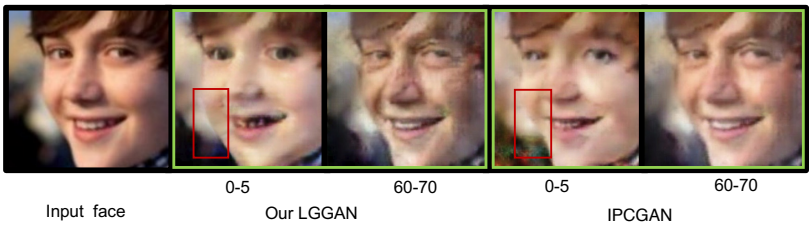


Figure 4.11: Comparisons between faces synthesized by LDcGAN and IPCGAN [146] when the input face has a side view. IPCGAN only makes slight changes in face texture and hence the aging effects are unconvincing. Our approach adjusts facial profile even under the side view (chubbier face as highlighted in red box), making the results more realistic.

facial structure. Our approach is robust against different input pose and successfully adjusts the facial profile and eye shape, making the results much more realistic.

4.4.3 Quantitative Comparison

To quantitative evaluate our approach, we compare it with CAAE [162] and IPCGAN [146] on both aging accuracy and identity preservation, which are important evaluation metrics in face aging.

Aging accuracy: To evaluate the age accuracy of synthesized faces, we use young faces as input to generate elder faces. On UTKFace, faces with age under or equal to 20 are considered as testing samples, and the target aging faces in other seven age groups are generated. For fair comparison, the Face++ API [75] is then used to estimate ages for generated results. Since the algorithms are trained using the UTKFace datasets, we first use Face++ API to estimate all real faces in different age groups of the datasets. The mean age of each group serves as the respective ground truth age value. The mean value of all

generated fake faces in the same age group is then compared with the ground truth. The less discrepancy between the two, the better synthesis accuracy in terms of aging effects. As shown in Table 4.2, our method outperforms the two baseline approaches in all but the 21-30 age group, demonstrating the effectiveness of our network.

Table 4.2: Estimated ages between real and synthesized faces. "Real" is the mean value of ages estimated for real photos from different age groups of the UTKFace datasets, which serves as ground truth. The following rows show the mean values of ages estimated for faces synthesized by different approaches, with values in brackets showing their absolute differences from the ground truth. Best results (smaller discrepancy) are shown in boldface.

Age group	21-30	31-40	41-50	51-60	61-70	71-80	80+
Real	25.03	38.01	46.12	54.63	65.40	73.66	87.29
CAAE*	24.31(**0.72**)	32.43(5.58)	42.21(3.91)	51.49(3.14)	60.17(5.23)	70.57(3.09)	82.68(4.61)
IPCGAN*	22.74(2.29)	31.74(6.27)	39.93(6.19)	50.04(4.59)	58.32(7.08)	68.42(5.24)	80.33(6.96)
Ours	26.18(1.15)	36.91(**1.10**)	44.68(**1.44**)	51.79(**2.84**)	62.52(**2.88**)	72.05(**1.61**)	88.24(**0.95**)

Additionally, evaluating synthesizing performance through a significance test is shown in Figure 4.12. In statistics, P-value is the probability that reflects the likelihood of an event occurring. A p-value of 0.05 or lower is generally considered statistically significant. As shown in the figure, the p-values obtained for different age groups are much greater than 0.05. This suggests that then there is no significant different between the age distribution between the ground truth image and the generated images.

Identity preservation: To determine whether the identities of input faces have been properly preserved during face aging process, face verification check is also performed.

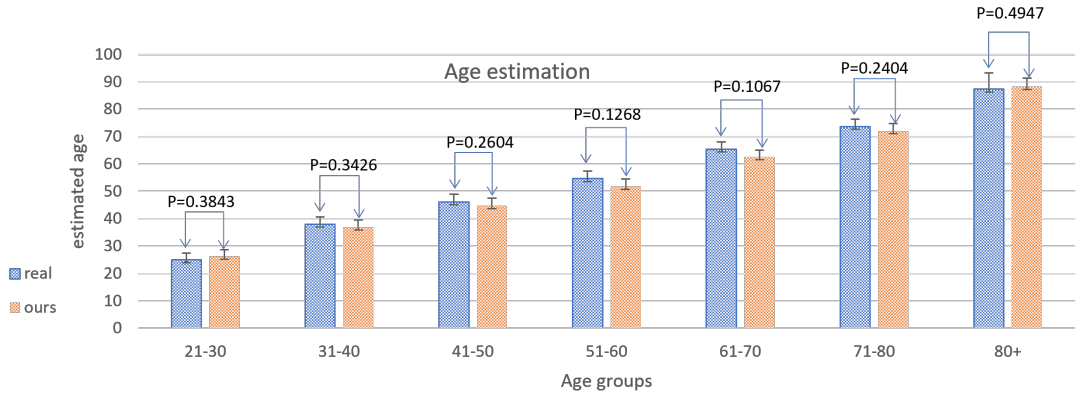


Figure 4.12: Significance test on age values estimated from both the ground truth and our generated images. The P-values for different age groups are also shown.

Here we use scores from Face++ [75] to compute not only the similarity between the input faces and each of the synthesized aging results, but also the similarity among aging results generated for different age groups. We again select images with age under 20 as the input and generate images in four other groups (21-30, 31-40, 41-50 and 50+). Comparisons between the input images and aging results in each target group give us the following four average confidence scores: [10-20 →21-30], [10-20→31-40], [10-20→41-50] and [10-20→50+]. we evaluate the verification rates between testing faces and its corresponding aging results to show the accuracy that they are the same person. We adopt $thresholds = 73.5$ and $FAR = 1e-5$ in our face verification experiments. Table 4.3 shows the identity verification check results. The top part lists the aforementioned verification confidence scores generated by LDcGANs. The bottom part compares the verification rates among three methods.

The above two tests show that, IPCGAN [146] is good at preserving identity information but lacks ability of generating aging faces with accurate target age appearance.

Table 4.3: Face identify verification results. The top shows the verification confidences among the input images and results synthesized by LDcGAN. The bottom compares the verification rates among three methods, with best verification rates shown in bold.

Age group	21-30	31-40	41-50	51+
	Verification Confidence			
10-20	95.76	94.78	94.65	93.28
21-30	-	95.74	94.54	93.77
31-40	-	-	95.12	94.32
41-50	-	-	-	94.64
	Verication Rate(%)			
CAAE [162]	87.05	81.07	73.36	60.25
IPCGAN [146]	**100**	**100**	**100**	**100**
ours	**100**	**100**	**100**	**100**

CAAE [162] performs better than IPCGAN [146] in age accuracy but fails to preserve personal identity. Our LDcGANs consistently outperforms CAAE [162] and IPCGAN [146] in both evaluation metrics.

4.5 Summary

In this chapter, we developed a novel Landmark-guided Dual-learning cGAN (LDcGAN) for enhanced face aging. We first introduce external landmark attention and built-in attention mechanisms to the generator to focus aging effects at important regions. A dual-learning framework is also used to both generation aging results and reconstruct the input photos. By minimizing multiple loss terms, our model can learn the transition pattern

at different ages and performs well in preserving personal identity. Both qualitative and quantitative experiments validate the effectiveness of our approach over the existing ones. However, the generated children's face is still limited compared to the results for adults. For example, we didn't consider the change of hair colours. Future work will include attribute-aware techniques and using hand-crafted landmarks from the original database, which may bring even higher accuracy results. Additionally, we believe that the proposed landmark-based attention module can provide better performance for other face synthesis problems, such as facial expression dynamics and facial image-image translations.

Chapter 5

Fine-Grained Talking Face Generation with Video Reinterpretation

Generating a talking face video from a given audio clip and an arbitrary face image has many applications in areas such as special visual effects and human-computer interactions. This is a challenging task, as it requires disentangling semantic information from both input audio clips and face image, then synthesizing novel animated facial image sequences from the combined semantic features. The desired output video should maintain both video realism and audio-lip motion consistency. To achieve these two objectives, we propose a coarse-to-fine tree-like architecture for synthesizing realistic talking face frames directly from audio clips. This is followed by a video-to-word regeneration module to translate the synthesized talking videos back to the words space, which is enforced to align with the input audios. With multilevel facial landmark attentions, the proposed audio-to-video-to-words framework can generate fine-grained talking face videos that are not only synchronous with the input audios but also maintain visual details from the input face images.

Multi-purpose discriminators are also adopted for adversarial learning to further improve both image fidelity and semantic consistency. Extensive experiments on GRID and LRW datasets demonstrate the advantages of our framework over previous methods in terms of video quality and audio-video synchronization.

5.1 Introduction

Automatically generating talking face videos under different conditions, such as audio speech, text, and sketch, is a problem of interests in both Computer Vision and Graphics. A talking face contains rich and complex semantic information and humans are sensitive to subtle artifacts shown on faces. Hence, generating high-quality, audio-corresponding videos based on diverse conditions is a very difficult task. Although significant progress has been made in generating videos using temporal-dependency models [11, 116, 12], realizing photo-realistic visual contents and optimizing generated videos by lip semantic alignment remain challenging.

The key issue is to learn the shared representation of two modalities (e.g. the given audio and an arbitrary image). To achieve this, we explored coarse-to-fine learning module for generating fine-grained talking face video, as well as designed an end-to-end neural architecture built upon temporal-dependent GAN framework, which is conditioned on a face image, facial landmarks, and audio information. The generated video results are then reinterpreted to semantic features and transformed to words information, which is expected to align with the ID of the word associated with the input audio (see Fig. 5.1).

Current still image generation algorithms based on Generative Adversarial Network (GAN) have shown promising results for mapping from natural language feature space to

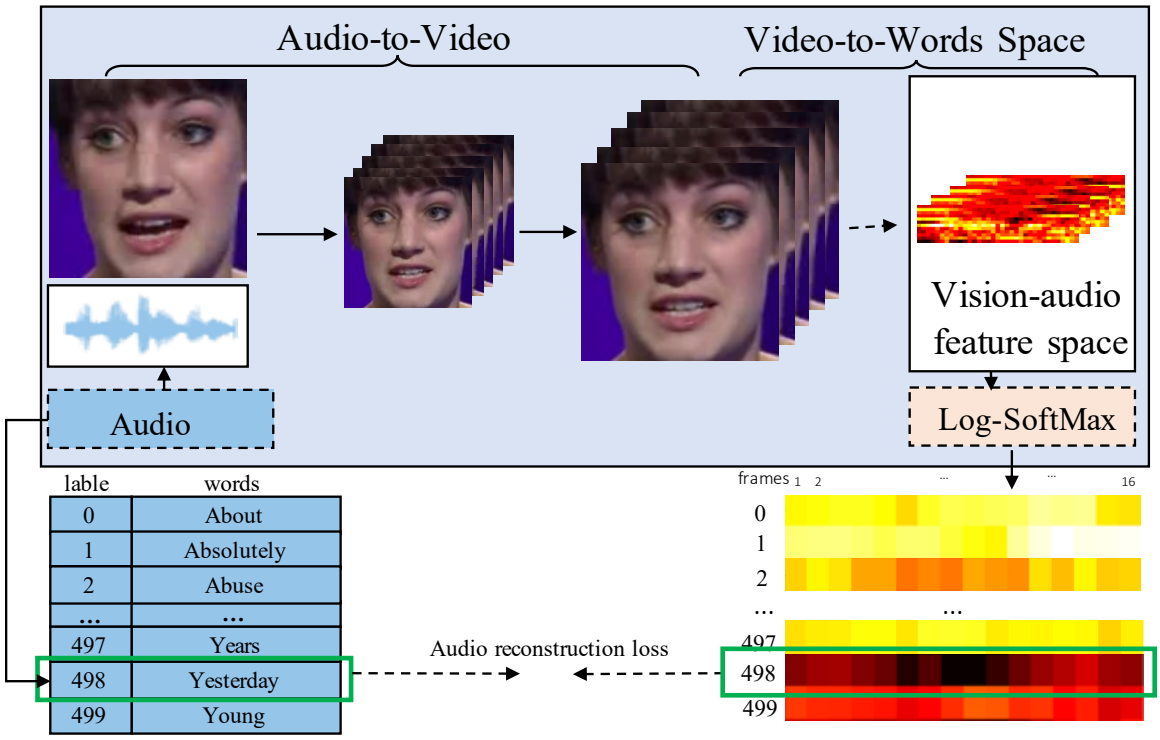


Figure 5.1: Overall architecture of the presented network. Taking an image of the target face and an audio signal as inputs, the network first generates a target face video in a coarse-to-fine manner. The obtained video is then reinterpreted into a vision-audio feature space by a lip reading model [117], which computes word probability distribution through a log-SoftMax layer. The word label with the highest probability is expected to align with the ground truth word that associates to the input audio.

image feature space. Zhang *et al*. [161] extended StackGAN [160] into StackGAN++ [161], which uses a tree-like structure to progressively generate images from small scale to large scale. Compared to still image generation, animating a still facial image to talking videos in a controllable way is far more challenging, due to the difficulties in bridging domain gap between audio and image sequences, as well as in eliminating artifacts between adjacent frames.

Most existing works on audio-to-talking-face generation can be primarily categorized into two classes: temporal independent methods [52, 12, 149, 165] and temporal dependent ones [122, 116, 11]. For example, Chung *et al*. [12] adopted an encoder-decoder model to generate one image for every 0.35-second of the input audio. A common failure phenomenon of temporal-independent model is that the generated image sequences are not always smooth, causing obvious pixel jittering among frames. To address this issue, Song *et al*. [116] incorporated valuable temporal information into the Recurrent Neural Network (RNN), whereas Chen *et al*. [11] divided the training into two successive steps: training an audio-conditioned face landmark generation model and a landmark-conditioned image generation model. This method establishes a bridge between audio and video by future landmarks prediction. Although it can generate temporally-consistent frames in an audio-driven manner, it lacks direct association between input audio and final images. Since the output is constrained using predicted landmarks only and through a single-purpose regression discriminator, the output images can be blurry. Therefore, the lip movements can roughly match input video, but the individual frames lack sharp edges and vivid textures. This is because motion dynamic regions are guided by features which correspond to pixel intensity and the whole quality of each frame image is insufficiently constrained by mean absolute error, i.e. $\mathcal{L}_1$ loss [150] .

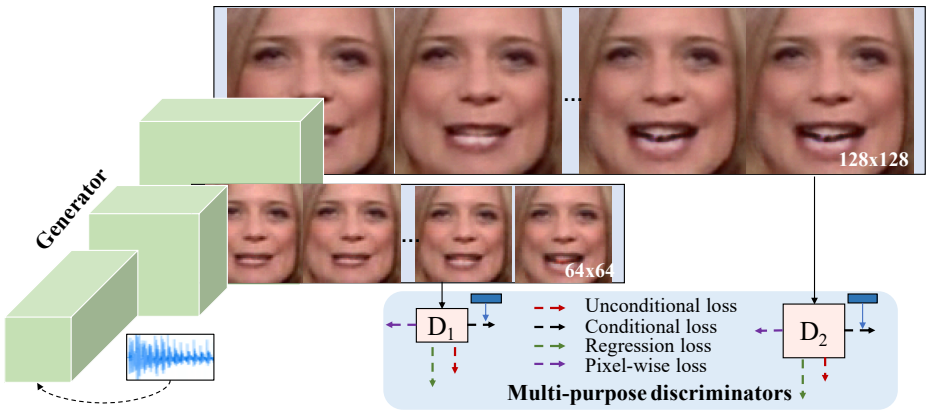


Figure 5.2: The schema of our coarse-to-fine adversarial network, which generates face frames with growing resolutions. Images at each resolution level are associated with a multi-purpose discriminator.

To tackle these problems, our proposed model adopts sketch-refinement and lip recognition steps to generate the photo-realistic talking face video, which precisely reflects the speech semantics. As illustrated in Fig. 5.2, at each scale level, we adopt multi-purpose discriminators for adversarial learning. The discriminator for each image frame is conditioned on a clip inside the audio that matches to this frame to enhance the semantic association. Our discriminators are trained to distinguish the generated image frames from the ground truth image frames based on the corresponding semantic audio clips. Furthermore, we apply multi-level attention mechanism in our model to weight image attribute features at different levels so that the generator is guided to emphasize informative phonic clips when generating various regions in each frame. For example, Fig. 5.5 shows that at low level, the attention information focuses on rough shapes such as pose, positions of facial features, etc., whereas at higher level, the attention information is refined and details such as lip textures, skin wrinkles, and eyeballs become the focus.

Lip reading process can be considered as an inverse problem of the audio-to-lip generation. Given a talking face video, the lip-reading network is performed to output words corresponding to the dynamic lip shapes of the input video. As shown in Fig. 5.1, if a talking video generated by the audio-to-video module is consistent with phonetic meaning of the input speech, the reinterpretation by the video-to-words module should output the highest probability for the word that is associated to the input speech. By integrating a lipreading network, the generator is impelled to produce more precise mouth motion to further narrow the gap between generated results and and the ground truth. Motivated by this observation, we propose a novel audio-to-video-to-words framework called AVWnet which exploits the idea of optimizing generation by re-description.

The contributions of our work can be summarized as follows: 1) establish a multi-level attentive generation network to generate fine-grained talking face, which is conditioned on the embedding of audio clips, example image and landmarks; 2) the extended reinterpretation module re-encodes the synthesized video to align its semantic information with the input audio; and 3) multi-purpose discriminators are designed as part of our adversarial learning framework to consistently constrain the quality and semantic correlation of generated video.

5.2 Related work

Talking face modelling. Research on talking face modelling was first studied in 1990s [156], which establishes the mapping between acoustic speech features and facial motions. Since then, various approaches have been proposed for audio-driven [52], video-driven [128] or text-driven [140] generations. Many traditional approaches on this topic are built upon hid-

den Markov models which reflect the dynamic features of audio and video sequences [7]. Deng *et al*. [17] combined target phoneme instances and expressive features to construct an best-matched motion-path. Ma *et al*. [72] proposed a model to generate speech animations through searching and concatenating best motions. Recently, methods based on deep learning have made great progress. Xie *et al*. [151] introduced a Bayesian network based model to synthesize videos with speaking mouth. Taylor *et al*. [125] adopted a deep neural network for transferring phonetic context to a dynamic lower half of the face. Karras *et al*. [52] presented an end-to-end training network to learn the mapping between 3D meshes and raw speeches. Fan *et al*. [20] generated the lower half region of the face by a bi-directional LSTM. Konstantinos *et al*. [138] trained a temporal GAN to generate talking faces from raw audios directly. Suwajanakorn *et al*. [122] trained a one-person based model to generate the dynamic mouth region, which is then restitched to the original video.

Video generation. With the booming studies of high-level representation of images [43], researchers extend techniques for image synthesis to videos. Currently, generating videos based on different prerequisites has been extensively exploited. For example, how to predict video frames from previous frames has been studied in [74, 87]. Motivated by the success of Generative Adversarial Nets (GANs) [27], some researchers implement video generation based on adversarial learning [103, 137] using 3D convolutional layers. Tulyakov *et al*. [131] decomposed video features to motion and content via a recurrent neural network. Li *et al*. [61] presented a text-to-video generation based on mlVariational Auto Encoders (VAE) and GANs. Suwajanakorn *et al*. [122] learned lip landmarks from audios, then synthesize textures from lip landmarks, and finally merge lip textures into the original face.

Attention models. Attention mechanism has been a hotspot in many research communities ranging from computer vision [159, 71] to natural language processing [70]. Xu *et al*. [154]

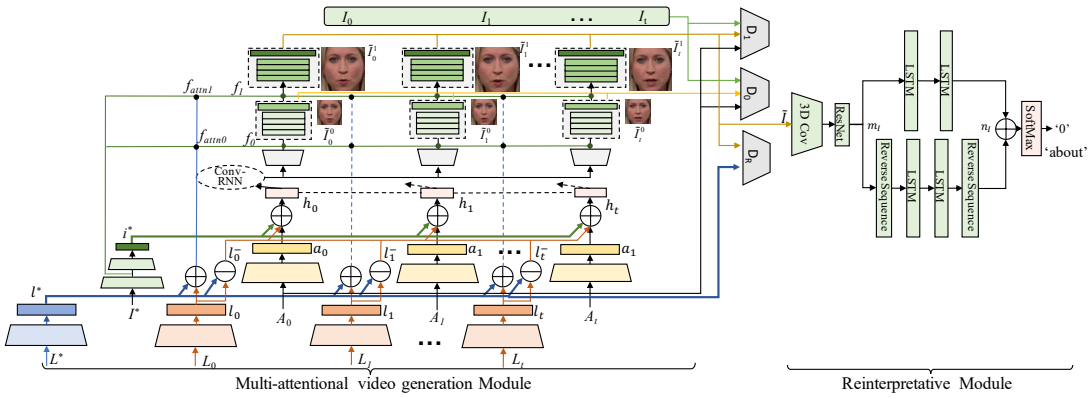


Figure 5.3: Overview of our proposed network structure, which consists of a multi-attentional video generation module for talking face video generation and a reinterpretative module for lip reading from the synthesized video.

applied the word level attention to multi-level layers, which underlines different words in different subregions. Pumarola *et al*. [94] and Chen *et al*. [11] adopted the facial attention masks and base colour features to generate the final RGB images. The attention mask determines how much the original image will contribute at each pixel location. We use a similar attention mechanism to discriminatively filter the audiovisual regions.

Antitropic structure. Our approach is also inspired by existing work on text-to-image generation [95], which re-describes the synthesized image using an image captioning model and aims to align the re-described text with the input text. Nevertheless, applying this idea to audio-to-video generation is a more challenging problem.

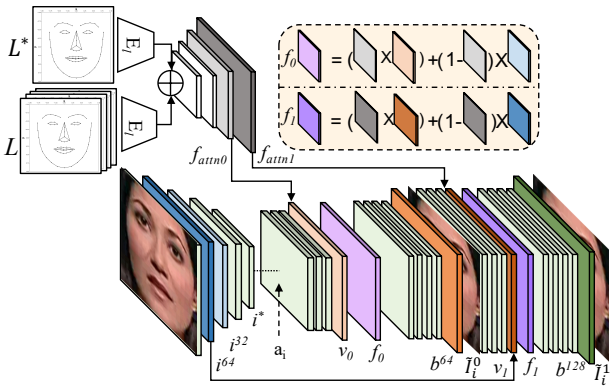


Figure 5.4: Multi-level landmark attention mechanism. The subfigure at top-right corner shows how the landmarks features (f_{attn0} and f_{attn1}) are used to blend between input image features (i^{32} and i^{64}) and synthesized hidden layer feature maps (v_0 and v_1). Different components in the subfigure are colour coded, where the colours match the corresponding layers in the network.

5.3 Methodology

To infer the emotions and desired mouth motions of a talking face from a given audio, we need to find a mapping between audio features and face features. This problem can be considered as a conditional temporal GAN problem and solved by minimizing the distance between ground truth video distributions and generated ones [27]. As shown in Fig. 5.3, our Audio-to-Video-to-Words network (AVWnet) consists of a multi-level attentional generative network and a reverse lip reading network. The video generation process exploits the idea of Stackgan++ [161], with multi-stage generator and multiform discriminators arranged in a tree-like structure. Inspired by Chen *et al*. [11], we leverage pixel-wise landmark attentions in different stages of the generator. Once all face frames are generated, a reinterpretative module is cascaded to predict word IDs, which are constrained to be consis-

tent with the input audio. In the subsections below, we first elaborate the video generation part, which is followed by explanation on the semantic word regeneration and alignment module.

5.3.1 Multi-level attention-based generative network

Inputs. The generative network receives three kinds of inputs which are hybrid features extracted by the following branches.

Audio feature processing: The inputs to audio encoders are Mel-Frequency Cepstral Coefficients (MFCC) values. Similar to methods in [12, 116, 11], for every audio sample, we use the last 12 coefficients out of original calculated 13 values. The successive window is set to 0.01s and each audio segment is 280 ms, resulting in 28 time steps. Every input vector can be formulated as a 28×12 feature map, where each row embodies MFCC features at each time step. We select a consecutive time step audio sequence $A = \{A_0, A_1, ...A_t\}$ that matches the selected video frames and feed them into a sequence of audio encoders. Each element A_i represents an inner clip in this audio sequence. Every audio encoder E_a outputs one audio clip feature: $a_i = E_a(A_i)$ for $i = \{0, 1, ...t\}$.

Landmark feature processing: In the training and testing process, given a sequence of facial landmarks $L = \{L_0, L_1, ...L_t\}$, the landmark encoder E_l generates latent landmark features $l = \{l_0, l_1, ...l_t\}$. We take the distance $l^- = \{l_0^-, l_1^-, ...l_t^-\}$ between real landmark distribution l and the example landmark distribution $l^* = E_l(L^*)$ as one of the condition value for the generative network, where $l^- = (l - l^*)$,L^* is the example landmark. Real landmark is generated face landmark from real image sequence and example landmark is generated face landmark from the example image. Given a random face image and a speech

signal, we first employ the landmark generation model from Chen's landmark prediction model [11] to get the sequence of predicted landmarks instead of real landmarks, then take the audio signal, example image, and divergent landmarks as input conditions.

Image feature processing: To realize arbitrary face video generation, our proposed conditional temporal GAN framework needs to be conditioned on image features, so that the facial features and audio information can be further fused together. An example image $I^* \in I \equiv \{I_0, I_1, ...I_t\}$ is randomly chosen from the source frames and transformed into latent variables i^* through an image encoder, i.e. $i^* = E_i(I^*)$.

Outputs. We concatenate the audio features $a = \{a_0, a_1, ...a_t\}$, example image feature i, and landmark difference

$l^- = \{l_0^-, l_1^-, ..., l_t^-\}$ together, forming a hybrid feature $h = \{h_0, h_1, ..., h_t\}$ as the condition, which is performed by:

$$h^{'} = E_a(A) \oplus (E_l(L) - E_l(L^*)) \tag{5.1}$$

$$h = E_i(I^*) \oplus h^{'} \tag{5.2}$$

where $\oplus$ is concatenation operation. The network attempts to generate talking face frames $\tilde{I}^0 = \{\tilde{I}_0^0, \tilde{I}_1^0, ...\tilde{I}_t^0\}$ with size of 64×64 in the 1st stage, $\tilde{I}^1 = \{\tilde{I}_0^1, \tilde{I}_1^1, ...\tilde{I}_t^1\}$ with size of 128×128 in the 2nd stage, so that the conditioned results distributions of $\tilde{I}$ and ground truth frames I are as close as possible, i.e., $p(\tilde{I}\,|A, L\,, I^*) \approx p(I\,|A, L\,, I^*)$.

Multi-level landmark attentions. In most of the existing works, single level-based attention is widely utilized to recalibrate the features [11], which usually limits the ability to capture more informative representations for coarse-to-fine process. To overcome this limitation and further reduce jitters between adjacent frames, we design a multi-level attention mechanism, which is imposed on the internal multi-scale features; see Fig. 5.4. The

coarse-to-fine attention mechanism helps to precisely rescale and emphasize the dynamic areas that encode speech-related face motions, thus helping produce more sophisticated multi-level features in a fine-grained fashion. Specifically, to calculate the latent attention layers f_0 and f_1, we integrate landmarks features (f_{attn0}, f_{attn0}), hidden-layer features (i^{32}, i^{64}) of example image, and features v_0 decoded using a Convolutional Recurrent Neural Network (Conv-RNN) along with an upsample process. The overall attention mechanism can be formulated as:

$$
\begin{aligned}
f_{attn0} &= \sigma(H_{d0}(E_l(L^*) \oplus E_l(L))) \\
f_{attn1} &= \sigma H_{d1}(E_l(L^*) \oplus E_l(L)) \\
v_0 &= H_D(H_{\mathrm{cov}-rnn}(h)) \\
f_0 &= (f_{attn0} \odot v_0) + (1 - f_{attn0}) \odot i^{32} \\
b^{64} &= H_{dup1}(f_0) \qquad (5.3) \\
v_1 &= H_{res}(b^{64}) \\
f_1 &= (f_{attn1} \odot v_1) + (1 - f_{attn1}) \odot i^{64} \\
b^{128} &= H_{dup2}(f_1) \qquad (5.4)
\end{aligned}
$$

where $\odot$ is multiplication operation. H_{d0} is landmark decoder which can decode the concatenated landmark values to attention feature maps with size of $N_l \times 32 \times 32$, whereas H_{d1} produces attention feature maps with size of $N_l/2 \times 64 \times 64$. $H_{\mathrm{conv}-rnn}$ is a Conv-RNN network, which generates temporal sequential vectors from the input sequence. Through a deconvolutional operation H_D, we can get a latent feature maps v_0 for future generation purpose. After upsample transaction H_{dup} and residual network H_{res}, we obtain the final feature results b^{64} and b^{128}, which will be used to generate realistic image frames $\tilde{I}^0$

of 64×64 resolution and $\tilde{I}^1$ of 128×128 resolution, respectively. The implementation process is as follows:

$$\begin{aligned} \tilde{I}_i^0 &= f_{gray}^0 \odot f_{colour}^0 + (1 - f_{gray}^0) \odot I^{0*} \\ \tilde{I}_i^1 &= f_{gray}^1 \odot f_{colour}^1 + (1 - f_{gray}^1) \odot I^{1*} \end{aligned} \tag{5.5}$$

where the attention mask f_{gray} is 1 channel activated convolutional result of feature b and the colour content f_{colour} is 3 channel activated convolutional result of b. Based on above computations, the model produces image frames that not only retain semantically irrelevant information from the given example face image, but also generate new semantically consistent information according to the regions in the colour content, which acts on the positive part of the grayscale attention mask.

5.3.2 Semantic video reinterpretation

As described above, our proposed model includes a semantic video reinterpretation module, which maps generated image frames to the word space. We here adopt a popular lipreading framework proposed by Stafylakis et al. [117] for its simplicity. Other more advanced lipreading models [92, 91] can also be integrated in our network and potentially yield better results.

Relying on a 3D spatiotemporal convolutional network and a Bi-LSTM, the audiovisual speech recognition process is as follows:

$$m_I = H_{res}(H_{STC}(\tilde{I})) \tag{5.6}$$

$$p = \sigma^{'}(H_{lstm}(m_I) \oplus H_{re\_lstm}(m_I)) \tag{5.7}$$

where the input $\tilde{I}$ is the frame sequences synthesized by our proposed generative network,

H_{STC} is the spatiotemporal convolutional network and H_{res} is a residual block. H_{lstm} and H_{re_lstm} are combined to a two-layer Bi-LSTM network. A following linear layer has been omitted for clarity. $\sigma^{'}$ is a log-SoftMax applied in the last layer, which realizes word-level recognition by producing words probability distribution p. To help AVWnet achieve a more stable training process and converge faster, we pre-train this model rather than jointly optimizing it with generative network. The parameters in the pre-trained model are fixed during training of the generative model.

5.3.3 Multi-purpose adversarial losses

To produce videos with increasing resolution, hierarchical adversarial losses are associated with the tree-like structure generator, which plays a dominant role in the performance of our AVWnet. Apart from the conventional single $\mathcal{L}1$/$\mathcal{L}2$ loss, other loss functions can be incorporated as constraints on the semantic consistency and to boost the generation quality. Specifically, we first transform the matching-aware pair loss [161] to improve the semantic consistency into the video generation problem. The pixel loss $\mathcal{L}_{pix}$ imposed on mouth region by the mean absolute error and regression-based loss $\mathcal{L}_R$ [11] are also combined together to enforce the high-quality generation of mouth region and structural consistency. Note that, to learn coarse-to-fine consistent features at multiple layers, all of these loss terms are independently computed at different generative stages.

The generator G and discriminator D are trained alternately at each stage of AVWnet. G_i is the i^{th} stage of generative network and it has a corresponding discriminator D_i. The discriminator D_i takes frame-phonetic clip pairs as its inputs and jointly approximates conditional and unconditional distributions. Discriminator D_i has two training objectives: to

distinguish whether the input video is real or fake; and to classify whether an video-audio fragment condition pair matches or not [161]. Our discriminator loss function is a combination of conditional loss and unconditional loss. We feed the discriminator with five types of input:

I_j: the j^{th} frame of real video;

$\hat{I}_j^i$: the j^{th} frame of generated video frame in the i^{th} stage;

(I_j^i, A_j): real j^{th} video frame with matching j^{th} clip of input audio in the i^{th} stage.

$(\hat{I}_j^i, A_j)$: generated j^{th} video frame with matching j^{th} clip of audio in the i^{th} stage.

$(I_j^i, \hat{A}_{j+1})$: real j^{th} video frame with mismatching $(j+1)^{th}$ clip of audio in the i^{th} stage.

The cross-entropy loss for D_i function is defined by:

$$\begin{aligned}\mathcal{L}_{D_i} = &\underbrace{-1/2\mathbb{E}_{I_j^i \sim pdata_i}[\log(D_i(I_j^i))] - \frac{1}{2}\mathbb{E}_{\hat{I}_j^i \sim pG_i}[\log(1 - D_i(\hat{I}_j^i))]}_{unconditional-loss} + \\ &\underbrace{-1/3\mathbb{E}_{I_j^i \sim pdata_i}[\log(D_i(I_j^i, A_j))] - \frac{1}{3}\mathbb{E}_{\hat{I}_j^i \sim pG_i}[\log(1 - D_i(\hat{I}_j^i, A_j))]}_{conditional-loss} \\ &\underbrace{-\frac{1}{3}\mathbb{E}_{I_j^i \sim pdata_i}[\log(1 - D_i(I_j^i, \hat{A}_{j+1}))]}_{conditional-loss}\end{aligned} \tag{5.8}$$

By combining the cross-entropy loss $\mathcal{L}_{D_i}$ and the regression loss $\mathcal{L}_{R_i}$, the final loss $\mathcal{L}_D$ for the conjoint discriminators can be formulated as:

$$\mathcal{L}_D = \mathcal{L}_{D_i} + \lambda \mathcal{L}_{R_i} \tag{5.9}$$

where λ is a hyperparameter to balance the two terms.

Our generator loss function is a contextual loss, which consists of conditional and unconditional losses. The i^{th} training stage G_i is trained by minimizing the loss as follows:

$$\mathcal{L}_{G_i} = \underbrace{-1/2\mathbb{E}_{\hat{I}^i_j \sim pG_i}[\log(D_i(\hat{I}^i_j))]}_{unconditional-loss} - \underbrace{1/2\mathbb{E}_{\hat{I}^i_j \sim pG_i}[\log(D_i(\hat{I}^i_j, A_j))]}_{conditional-loss} \tag{5.10}$$

We further utilize an aggregated per time step loss to align the reinterpretated words and the given audios. Mathematically, this loss is defined as:

$$\mathcal{L}_w(p, C_{t\arg et}) = -\sum_{k=0}^{t} p[C_{t\arg et}] \tag{5.11}$$

where p is words probability distribution after the log-SoftMax operation (see Eq. 5.7). To generate realistic videos with phonetic conditions, the final objective function of the AVWnet is weighted summation of generator loss, audio reconstruction loss $\mathcal{L}_w$ and mouth region pixel loss $\mathcal{L}_{pix}$:

$$\mathcal{L} = \mathcal{L}_G + \partial\mathcal{L}_w + \beta\mathcal{L}_{pix} \tag{5.12}$$

where ∂ and β are hyperparameters to balance the three terms.

5.4 Experiment

Datasets. We evaluate the proposed method on two widely used datasets: LRW [13] contains 500 classes of words spoken by hundreds people. In each word class, there are 1,000 training video samples, 50 test samples and 50 validation samples. GRID [14] contains 33 speakers, each uttering 1,000 short phrase. We extract image frames with a sampling rate of 25 FPS, which leads to 31 frames for each LRW video and 75 frames for each GRID video. All face images are cropped to 128×128 by aligning to the landmarks. We train on ground truth landmarks that are generated by 2D face alignment library [8] and test under generated example landmark and future landmark sequences predicted by a pre-trained

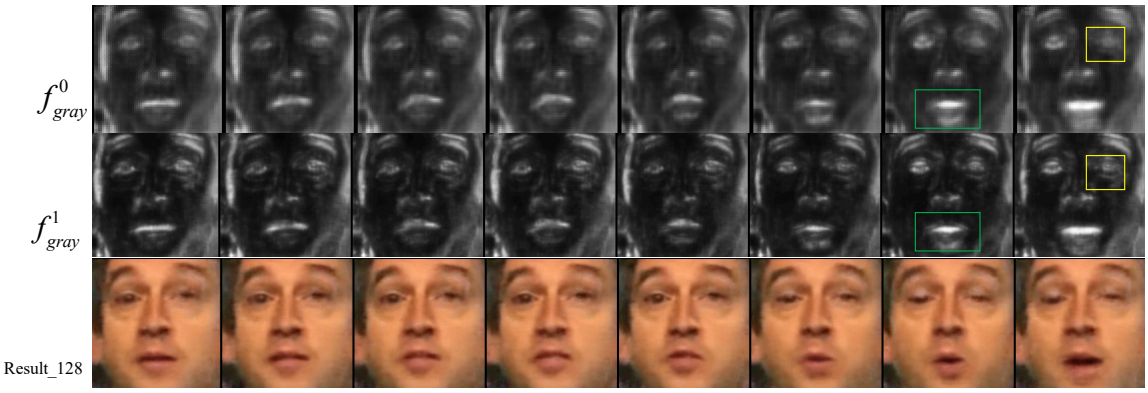


Figure 5.5: Attention masks generated for the coarse (top) and fine (middle) levels and the resulted image frames (bottom). The image resolution is 64×64 pixels for f^0_{gray} and 128×128 for f^1_{gray}. The green boxes highlight that fine-level masks are sharper and more precise around the mouth area, whereas the yellow boxes demonstrate that fine-level masks learned in a latter stage contain more detailed textures.

landmark-prediction model [11]. To get audio signal, we transform videos to raw audio format first, then extract MFCC features with window size being 0.01s. Moreover, we pre-train the lip reading model [117] based on image frames with size 128×128, which are cropped in the same way with our generative model.

Evaluation metrics. Our proposed approach is evaluated both qualitatively and quantitatively. The quantitative measures we used are: Structural Similarity Index (SSIM) [144], Peak Signal-to-Noise Ratio (PSNR) [46], and Landmark Distance Error (LMD) [10]. SSIM and PSNR aim to assess the quality of our generated videos. LMD calculates the distance between landmarks in generated video and ground truth video, and it is used to evaluate audio-video semantic consistency.

5.4.1 Implementation details.

Basics. Our model is implemented on PyTorch and tested on a single Nvidia GeForce GTX 1080 Ti GPU with 50GiB memory. We apply the batch normalization, set a fixed learning rate of 0.0002, and use Adam algorithm as the optimizer.

Architecture details. As illustrated in Fig. 5.3, our generator consists of multiple threads encoders and temporal two-stages decoders with residual connections. The example image encoder transforms input image to a $4N_i \times 8 \times 8$ tensor by five convolutional (Conv) layers with residual connections and padding$= 1$ after the second Conv layer, where N_i is the depth of latent image feature maps. The residual block and the third Conv layer produce feature maps with shapes of $N_i/2 \times 64 \times 64$ and $N_i \times 32 \times 32$, respectively. These feature maps will be used in the decoders to constrain the final video generation.

The audio encoder is decomposed to same 16-streams encoders to manage selected 16 audio segments that correspond to 16 frames of the video. Each audio encoder involves two stages: The first stage consists of five standard Conv layers (kernel size $= 3 \times 3$, stride$= 1$, padding$= 1$) and two maxpool layers, which are placed after the 2nd and 5th Conv layers to increase the scale invariance and non-linearity of the features; The second stage employs three upsample operations, extracting audio feature maps with size of $N_a \times 8 \times 8$, where $N_a = 2N_i$.

Similarly, the landmark encoder consists of the same 16-streams landmark frame encoders and one example landmark encoder. Each landmark encoder has a linear layer with Relu activation and a Conv layer, producing landmark features with size $N_l \times 8 \times 8$, where $N_l = 2N_i$. The subtraction result between example landmark features and frame landmark features is concatenated with the audio clip features, forming a hybrid condition vector

for the generator. Associating the condition with example image features, the multiple encoders produce 16-stream inputs for a Conv-RNN layer.

The Conv-RNN layer predicts 16 frame feature maps and each frame feature map is fed into 16 identical branched decoders. Each decoder can be divided into four sections. The initial section consists of a four layer residual block and transposed convolution layers to improve the resolution of images, producing feature maps with size $N_i \times 32 \times 32$. The second section aims to generate image frames with scale 64×64. It has a residual block and an upsample operation, producing feature maps with size $N_i/2 \times 64 \times 64$. The third section is a pure residual block containing tensors that have the same size with preceding maps. The last layers of the first section and the third section accept multi-level attentions as discussed in Sec. 5.3.1. The last section further improves the image resolution to 128×128 by a residual block and an upsampling operation.

Our discriminator D involves 3 parts. The first part contains four 3×3 Conv layers with BatchNorms and LeakyReLU activationtrans, which transforms $3 \times 128 \times 128$ image frames to $8N_d \times 16 \times 16$ feature maps, where $N_d = 64$. The second part produces $16N_d \times 4 \times 4$ maps using downsampling blocks. The third part has a Conv layer of kernel size 3×3, stride$= 1$, padding$= 1$ with a BatchNorm and LeakyReLU activation, generating a tensor with size $16N_d \times 4 \times 4$.

5.4.2 Results and comparison

Qualitative results. As illustrated in Fig. 5.6-5.8, we qualitatively compare the results generated by the presented AVWnet with those of two state-of-the-art approaches: Chen *et al*. [11] and Chung *et al*. [12]. All three methods are trained on LRW dataset and output

Figure 5.6: Video frames (128×128 pixels) generated using the same audio clip but different face images sampled from synthetic images, real world images, celeA, LRW datasets and cartoon characters, respectively. Top row shows the ground truth video and the left column in the remaining rows shows the input faces. The lip movements of generated video frames match the ground truth effectively.

videos with 128×128 resolution. Chung *et al*. [12] adopt a pre-trained VGG-M model on the VGG face dataset [89].

Fig. 5.5 displays the attention masks obtained for both coarse and fine levels. The fine level attention masks contain more detailed textures on skin winkles, eye details, and the vermilion border. As a result, the multiple generation stages enrich the details in the result frames.

Fig. 5.6 shows the outputs generated by AVWnet on different input faces. The input audio reads "what happened to her", which is randomly selected from the testing set of LRW dataset. We show inner 18 frames out of 31 generated frames. Despite the wide variety among input faces, AVWnet effectively synthesizes the lip movements that are synchronized to the input audio and nicely matched to the mouth shapes in ground truth video frames.

Figure 5.7: Comparisons of videos frames generated by AVWnet, Chen *et al*. [11] and Chung *et al*. [12] based on two sets of input audios and faces. The first comparison uses the same face from the ground truth for direct comparison with ground truth frames, whereas the second comparison is tested on a cartoon character shown on the left. Red boxes highlight frames that AVWnet generates lip movements that best match to the ground truth. Yellow boxes highlights frames that AVWnet yields best fidelity (sharper details and more realistic colours).

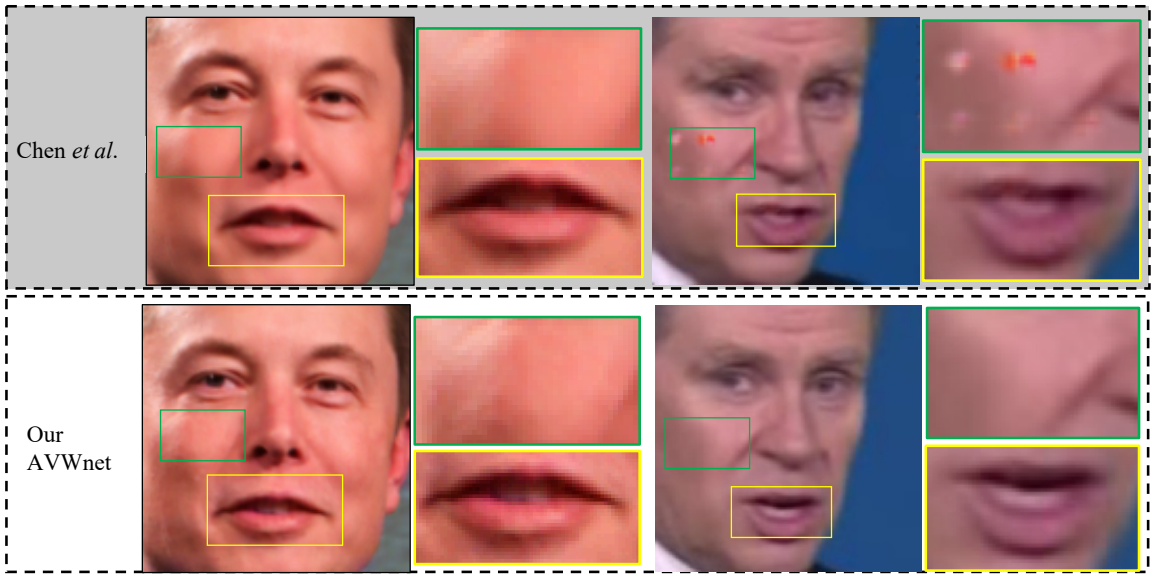


Figure 5.8: The detailed comparison between Chen *et al*. [11] (top) and our approach (bottom). Green boxes highlight the areas that AVWnet generates more detailed and realistic textures, whereas Chen *et al*.'s approach yields artifacts since the attention masks are not accurately learned. Yellow boxes highlight the mouth region that AVWnet produces sharper details. In contrast, the mouth region produced by Chen *et al*. [11] are blurry.

Fig. 5.7 compares the results generated by AVWnet with Chen *et al*. [11] and Chung *et al*. [12]. Both an identity face image from the ground truth and a cartoon figure from the Internet are used for testing. Visual inspection over different frames demonstrate that our model produces sharper talking faces with discriminative lips, teeth, eyes and detailed skin textures compared with the other two methods (e.g., frames in yellow boxes). The mouth motions are also better matched with the ground truth, which means our outputs can better reflect the input audio (e.g., frames in red boxes). The outputs of existing approaches [11, 12], on the other hand, are more blurry and have less noticeable lip movements. Fig. 5.8 further compares our work with Chen *et al*. [11] on two cases where their results contain blurry regions and obvious artifacts (e.g. spots on the face), whereas our approach outputs sharp and artifact-free images.

Quantitative results. Using the aforementioned evaluation metrics, we quantitatively compare the performance of our AVWnet with state-of-the-art approaches [11, 12] on two datasets: LRW and GRID. The values of SSIM and PSNR reflect the visual frame quality. The higher the score, the better the visual results are. LMD is the distance between the landmarks of generated frames and the ground truth landmarks, which indicates the semantic-visual consistency. The lower the LMD, the better audio-video synchronization is. For fair comparison, different approaches are evaluated under the same test algorithm and our customized test datasets. Table 5.1 provides the evaluation results, which shows that AVWnet outperforms both existing approaches on all three metrics and for both datasets.

We also compare the computation time needed by AVWnet and Chen's approach [11] for generating each frame under the same parameter settings and hardware environment (one GeForce GTX 1080 Ti Graphics Card). We apply both methods to produce the same talking video with 31 frames respectively. Method [11] can generate one frame in 0.018

Table 5.1: Quantitative evaluation on LRW and GRID testing datasets. Best scores are shown in boldface.

Method	LRW			GRID		
	SSIM	PSNR	LMD	SSIM	PSNR	LMD
Chung [12]	0.71	28.31	3.19	0.74	28.46	3.03
Chen [11]	0.75	30.04	2.97	0.77	31.61	2.88
Our AVWnet	**0.82**	**31.24**	**2.84**	**0.84**	**32.03**	**2.79**

seconds, whereas our method takes 0.025 seconds. However, as our model involves extra computation of multi-level landmark attention and multi-level discriminators to improve the video quality and accuracy, the slight increase of computational cost is acceptable.

5.4.3 User studies

Sec. 5.4.2 evaluates all synthesized frames individually and hence may not be sufficient for evaluating the whole video. To address this limitation, we conduct a user study between our model and Chen *et al*. [11], which has better performance than Chung *et al*. [12].

We randomly select 16 example faces from LRW, CelebA, GRID, and cartoon characters to generate 16 talking videos for each method, which are shuffled before being shown to participants. Same as Chen et al [cite], we use 10 participants. The group that we picked has equal numbers of males and females and involves both undergraduate and graduate students between the ages of 20 and 40. To make the survey result more robust, every pair of videos is shown to each participant twice. The participants are asked to score between 0 (worst) and 5 (best) with interval of 0.1 on three aspects: the consistency between lip movements in the videos and input audio (synchronization), the smoothness of the overall

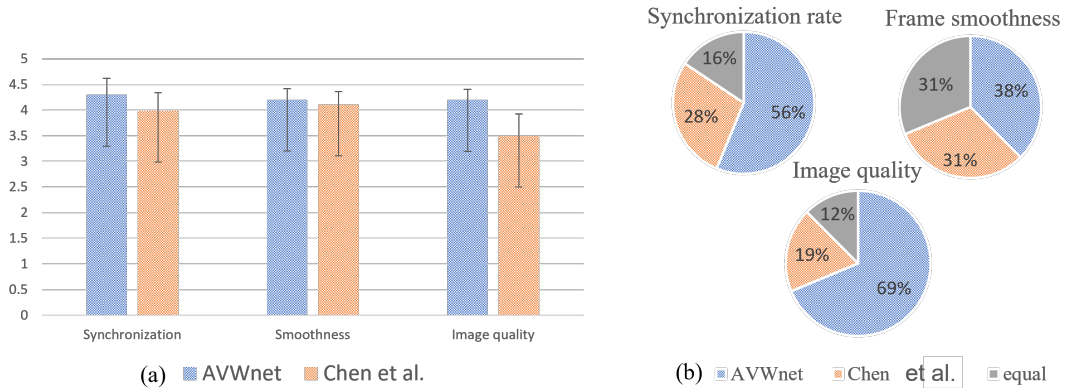


Figure 5.9: User study on videos generated using the proposed AVWnet and the state-of-the-art method [11], which shows AVWnet outperforms [11] in synchronization and image quality.

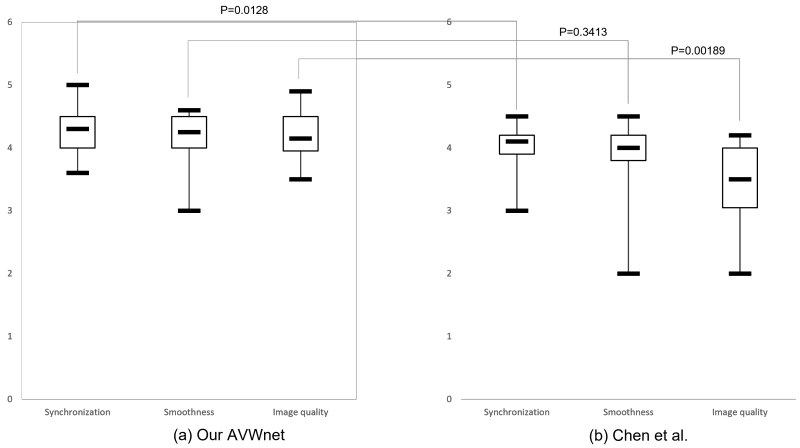


Figure 5.10: Comparison between two boxplots of scores on two models. (a) scores of videos generated by our AVWnet; (b) sores of videos generated by Chen et al. [11].The P-values for different scoring aspects are also shown.

video (video smoothness), and the fidelity of the each image frame (image quality). The average scores of the two methods on the three aspects are then computed; see Fig. 5.9 (a). For the statistical performance comparison fo the scores, we draw two boxplots to show the performance of our proposed AVWnet and the Chen's method. (see Fig. 5.10). A p-value of 0.05 or lower is generally considered statistically significant. As shown in the figure, the p-values obtained for synchronization and image quality are much lower than 0.05. This suggests that then there are significant different between the distributions between the scores of our model and Chen's model. As for the video smoothness, the p-value is greater than 0.05, this suggest that there is no significant difference between the two models in terms of video smoothness. The comparison shows that AVWnet outperforms Chen's method in terms of synchronization and image quality, whereas the video smoothness scores for the two methods are comparable.

Additionally, we count the three possible evaluation outcomes for each pair of videos generated by the two methods: AVWnet gets higher scores, Chen *et al.*'s method gets higher scores, and both methods get equal scores. Fig. 5.9 (b) shows the percentages of each outcome on the three aspects. In terms of the image quality, 69% users gave higher scores to AVWnet, compared to 19% gave higher ones to Chen *et al.*'s model. In terms of synchronization, 56% scored higher for AVWnet and 28% scored higher for Chen *et al*. [11].

5.4.4 Ablation studies

To study the effects of different components and learn the contributions of different losses in our method, we perform an ablation study on the LRW dataset. We investigate three

Table 5.2: Ablation studies on LRW. The performances of the algorithm after removing different components are evaluated. Scores with obvious changes when compared with the AVWnet are shown in boldface.

LRW		Variant without component:			
	AVWnet	AM_0AM_1	$\mathcal{L}_{D_0}\mathcal{L}_{D_1}$	$\mathcal{L}_R$	RIM
SSIM	0.839	0.831	**0.796**	0.836	0.839
PSNR	32.425	32.418	**31.120**	32.421	32.423
LMD	2.775	**2.820**	2.784	2.801	**2.802**

components: Attention Maps f_{attn0} and f_{attn1} (AM_0 and AM_1), Reinterpretative Module (RIM), and three loss functions (see Sec. 5.3): $\mathcal{L}_{D_0}$, $\mathcal{L}_{D_1}$, $\mathcal{L}_R$ in this section. Table 5.2 shows the comparison results by removing each element at a time. To accelerate the process, we simplify the LRW datasets from 500 words to 24 words and test these variant models on these 24 words only. For the same reason, the ground truth facial landmarks, instead of the predicted ones, are used, which leads to higher evaluation scores in some cases.

Table 5.2 confirms that the best performance is achieved when all components are used. Specifically, removing the twofold temporal-spatial adversial loss $\mathcal{L}_{D_0}$ and $\mathcal{L}_{D_1}$ leads to declines in SSIM and PSNR measures, which indicates that $\mathcal{L}_{D_i}$ is crucial for visual quality. Similarly, removing the RIM and AM causes the LMD values to increase, which means the RIM and multi-attention mechanism are important for lip synchronization.

5.5 Summary

A fine-grained Audio-to-Video-to-Words network (AVWnet) is presented in this chapter for efficient audio-to-video talking face synthesis. Compared to previous models, our AVWnet can generate talking face videos with better audio-lip consistency and higher frame quality. This is achieved because the generative network in AVWnet jointly approximates multi-scale conditional and unconditional video distributions, and gradually produces videos in a coarse-to-fine manner. In order to further improve the audio-lip consistency, a reinterpretation module is used to supervise the generator by remapping the generated video to words and enforcing the reconstruction loss. The advantages of our method over existing state-of-the-art methods are demonstrated through both qualitative and quantitative evaluations on two classic datasets, as well as user studies.

Chapter 6

Attention-Aware Neural Painting via Deep Reinforcement Learning

Neural painting, which aims to produce realistic artworks with stroke sequences, has drawn considerable attention from both academia and industry. Previous methods are often designed to minimize the total colour distance of all pixels between the target image and painting results, making no efforts to distinguish between foreground and background objects. As a result, the stroke sequences generated differ greatly from those used by human artists, who generally pay more attention to the important content painting rather than details of backgrounds in limited strokes.

Motivated by this observation, we propose a new attention-aware end-to-end deep reinforcement learning framework for stroke planning, which better mimics the painting process of human artists. This architecture consists of two components: 1) an attention-guided policy network learning for generating stroke parameters and 2) stroke rendering network for canvas updates. The first component uses a dual-branch network to compute an attention

map for the current canvas, which is used to guide the generation of stroke parameters. A feature masked reward function is also adopted to train the reinforcement learning network to prioritize important regions identified on the attention map. Experiments demonstrate that our method can generate paintings that render foreground objects with great details and high fidelity but approximate the backgrounds using relatively coarser strokes. As a result, more visually appealing results are generated using fewer strokes than the existing approaches.

6.1 Introduction

As a well-known form of visual art, painting is a special and powerful way that humans used to reflect emotions, express thoughts, and preserve memories. Although great improvements have achieved in painting tools and techniques, it is still a hard skill to master since it requires long-term learning and practising. Hence, designing an agent that can efficiently mimic the human painting process will be a challenging but interesting task.

Early works of artistic painting generation mainly focus on style transfer based on generative modelling [23, 50, 81, 166]. These methods learn pixel-wise mapping between two images by continuous optimization. In contrast, painting creation by real artists is a step-by-step sequential process, which can not be realized by the pixel-wise image style transfer.

Recently, many Stroke Based Rendering (SBR) works have achieved great success in simulating paintings in a manner similar to human artists [22, 33, 45, 76, 152, 164]. Painterly rendering, especially Stroke-Based Rendering (SBR) [40] techniques have achieved remarkable success. The optimization methods and the greedy methods are adopted in au-

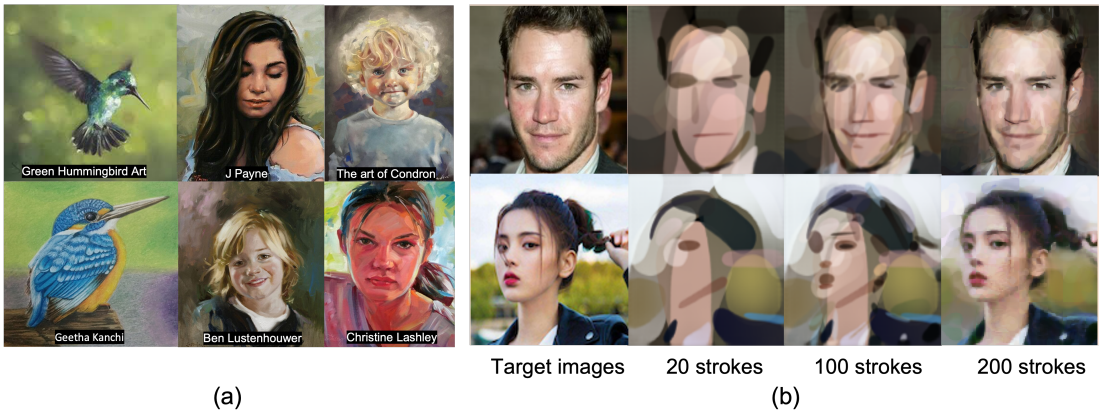


Figure 6.1: (a) In real paintings, artists (credited below) focus on their strokes on important foreground subjects and paint backgrounds with few rough strokes. (b) Our neural painting process achieves similar effects. To paint the target images shown on the left, it progressively applies strokes to produce paintings with detailed foreground subjects.

tomatic stroke placement [40]. The optimization algorithms iteratively place and adjust strokes to minimize or maximize certain objective energy functions [132] and the greedy algorithms try to place the strokes to match specific targets in every single step [39, 64]. Reinforcement Learning (RL) and Markov Decision Process (MDP) are broadly adopted to generate stroke sequences through an agent and determine where the strokes should be placed on the canvas. With enough strokes, the final canvas shows a painting that visually resembles the target image. Most of these RL models use generative adversarial network (GAN) [27] to learn the reward of the agent. The adversarial losses are designed to measure the similarity between the final painting and the target image. The planning of the strokes can find its way to minimize the loss. Some researchers aim to guide the RL painting to draw like humans by collecting real stroke samples from painters. This method requires significant human and material resources for data collection, which makes the process highly

time-consuming.

In this chapter, we treat neural painting as an action prediction process, which places a sequence of strokes on the canvas. Our goal is to mimic the stroke sequences used by humans, who often apply strokes from back to front, select brushes from big to small, and draw objects from coarse to detail. To achieve this goal, we summarize two guiding principles used by artists. First, artists constantly compares what is already drawn on canvas against the target image. The difference between the two will guide them to decide the ensuing strokes, which allows them to produce recognizable subjects at early painting stages that can minimize the difference. Secondly, artists pay more attention to the primary subjects in the scene than the background scenes. Hence, they tend to use more detailed strokes on important foreground objects than on backgrounds; see Figure 6.1(a).

To follow the above two guiding principles, we design a dual-branch attention module to produce foreground enhanced images and apply two different feature masked losses to prioritize stroke selections. The attention module and feature masked rewards direct strokes to areas that the current canvas differs from the target image the most and areas that contain high saliency foreground subjects. As a result, our approach can approximate the target image under a small number of strokes and capture fine foreground details in the final results. Particularly, we conducted comprehensive experiments under different methods and conditions to demystify the capacity of our approach to train painting agents.

In summary, our contributions are as follows: i) we adopt an attention module in policy networks that use rewards to direct generated strokes to selected areas; ii) we formulate two feature masked losses, which give higher rewards to high saliency areas on the canvas; and iii) experiment results show that the proposed painting agent can decompose images with complex scenes into stroke parameters and regenerate artistic paintings in a human-like

manner.

6.2 Related work

Neural style transfer. Style transfer approaches convert natural images to different painting styles while preserving the contents in the images. The task can be treated as an image-to-image translation problem. Unsupervised learning approaches were developed, so that a neural network can be trained on unlabelled datasets [166, 157]. Arbitrary style transfer approaches are also developed to stylize input images using a single image as the reference style [24, 115]. These approaches are powerful tools for generating raster artwork, but they cannot be used to output painting stokes.

Stroke based rendering (SBR). SBR overlays a bunch of individual elements such as strokes, lines and points on the canvas, forming a non-photorealistic artistic image. The goal of SBR is to create sequences with proper information (e.g., locations, shapes, thickness, etc.) [40] so that the final canvas resembles a target image [158]. Most traditional SBR algorithms are based on step-by-step greedy search [158, 64], energy function optimization by heuristics [132], and supervision of stroke positions realized by user interaction [34, 127]. Haeberli et al. [34] provided description of a simple, semi-automatic painting algorithm for single-point strokes rendering, which requires users to click and drag on the canvas. The colour is automatically extracted from the source image. Litwinowicz et al. [64] improved Haeberli's algorithm to single layer painterly rendering. The algorithm randomly put a set of strokes on canvas and the colour of each stroke is extracted from the source image and the orientation is set by the input orientation field.

Recurrent neural drawing. Recurrent neural networks play an important role in gener-

ating simple drawings, such as stick figures and handwritings. Existing RNN-based models such as SketchRNN [33] and Graves et al. [28], require labelled image-to-stroke-sequence datasets. Due to the lack of labelled datasets and the complexity of stroke sequence used by human, these methods can hardly handle complicated paintings.

Reinforcement Learning for SBR. Most recent SBR works solve the stroke decomposition problem by training an agent which learns whole paintings rather than a single stroke. Xie et al. [152] designed oriental ink strokes by Reinforcement Learning (RL). Based on the feedback, the agent is able to model strokes which are believed made by humans. Yaroslav et al. [22] proposed the SPIRAL which can train a deep RL agent by adversarial training. However, the agent can only capture the coarse structural feature and miss details in the images. Traditional rendering models is non-differentiable since they mostly involve rasterization, which is a discrete operation. Deep learning-based SBR [22, 45, 82] has become very popular recently. Neural Render, which breaks the gap between graphic rendering and deep neural networks, successfully applies differentiable neural networks so that the back-propagation and derivative calculation can be achieved [54, 60, 65, 69, 19, 109]. This allows Huang et al. [45] to decompose target images into sequences of strokes. The algorithm can perfectly regenerate the input image as long as there are enough strokes, but the stroke sequence generated often differ from those used by human artists.

6.3 Methodology

Our painting network aims to decompose an input image into brush strokes and render a new painting through these strokes by intelligent painting machines. We first introduce the painting agent of artistic drawing along with the settings of Markov Decision Process

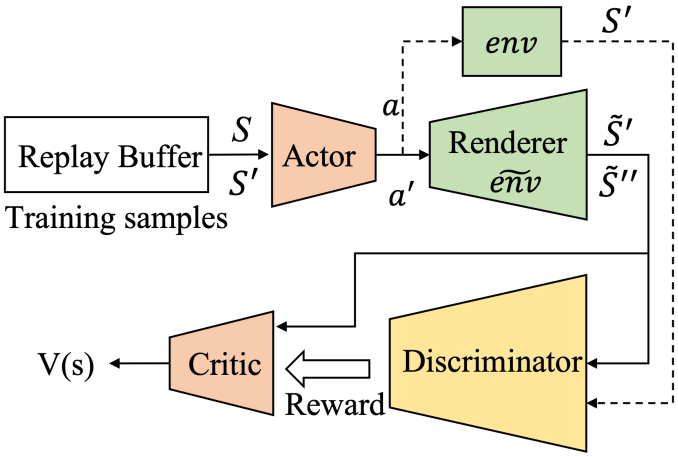


Figure 6.2: The overall architecture of model-based DDPG. S': next states of the environment; a: actions; a': next actions; env: real environment of the agent; $\tilde{env}$ environment model which is being modelled by generator; $\tilde{S}'$: predicted (generated) states after current states of environment; $\tilde{S}''$: predicted(generated) states after next states of environment.

(MDP) to show how the deep RL works. Then, we introduce the proposed end-to-end attention-aware architecture and the whole training procedure.

6.3.1 DDPG and model-based DDPG

Deep Deterministic Policy Gradient (DDPG) [62] improves Deterministic policy gradients (DPG) [113] by adopting deep convolutional neural network (DNN). It is an off-policy algorithm that concurrently learns a policy and a Q-function designed for the tasks that are continuous action spaces and high dimensional.

In generative RL agents, traditional rendering pipelines of DDPG involve rasterization, which makes the rendering non-differentiable. To break the non-differentiable limitation, Model-based RL [121] is adopted to efficiently predict how an action affects the state of the canvas environment and significantly accelerates the training time. Recently, Huang et al. [45] and Zou et al. [167] adopted a differentiable Neural Stroke Renderer to simulate

the rendering process through a neural network and formulate the image painting as an MDP problem (Figure 6.2). At the testing stage, action a from an actor network is an input of the real environment env and renderer $e\tilde{n}v$ and the next state S' is the output from the real environment env. Then the state S' is feed into the actor and to the discriminator. Here states S represents the canvas from blank to completion. The discriminator applies adversarial learning and update the environment model $e\tilde{n}v$ more accurately. At the training stage, the actor and critic are being trained and updated together using distribution distance between the generated data and the target data. The required reward for reinforcement learning is given by the discriminator at each step, and the training samples are randomly sampled from the replay buffer.

We improved the architecture in [45] with attention module and the feature masked loss. Coordinate with feature masked loss, the foreground focused state drives the actor to generate stroke parameters focus on the important part.

6.3.2 Painting Agent

We denote the state space of MDP as $\mathcal{S}$, the action space as $\mathcal{A}$ and the transition function between the two spaces as $\mathcal{F}(s_t, a_t)$, where $s_t \in \mathcal{S}$, $a_t \in \mathcal{A}$. We superimpose those strokes at each step to the canvas iteratively. At each drawing step t, the pretrained neural render network $\mathcal{H}_r$ takes in a set of stroke parameters a_t, and produces a foreground colour stroke a_c and an alpha matte a_r. We then use a soft blending to mix them with the previous canvas C_t by: $C_{t+1} = a_c + (1 - a_r)\, C_t$.

State $s_t = (C_t, I, f_{mask}, t) \in \mathcal{S}$ is defined as the current state s_t with canvas C_t rendered by strokes, the target image I, and target image feature masks f_{mask} extracted from different

layers of VGG-16 [114]. Here the feature masks are probability maps $\left\{\in [0,1]^{H\times W}\right\}$.

Action A set of parameters of a stroke at step t is denoted as an action a_t. The parameters involve information of position, shape, colour and transparency, which can effectively create a stroke by a render network. We use Quadratic Bézier curves to model strokes, thus we define a stroke as a 13 dimensional tuple:

$$a_t = (x_0, y_0, x_1, y_1, x_2, y_2, z_0, v_0, z_1, v_1, r, g, b)_t \tag{6.1}$$

$$Q(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, 0 \leq t \leq 1 \tag{6.2}$$

where $(x_0, y_0, x_1, y_1, x_2, y_2)$ indicates parameters of control points, (z_0, v_0) and (z_1, v_1) represent the thickness and transparency of two ends of a stroke, respectively. (r, g, b) are the parameters to control the colour of strokes. $Q(t)$ is the formulated quadratic Bézier curve, which serves as the ground truth stroke during the render network training.

State transition The transition function $s_{t+1} = \mathcal{F}(s_t, a_t)$ is simulated by pretrained neural render model $\mathcal{H}_r$. Based on current canvas C_t, the neural render model transforms the generated parameters a_t to strokes and updates canvas C_t to C_{t+1}.

Action Bundle As mentioned in Model-based RL [45], environment can be observed every k frames at one step to better learn the planning process. Experiments show that $k = 5$ can efficiently improve the performance and accelerate the learning speed [45]. Hence we set $k = 5$ as well to predict parameters of 5 strokes at each step.

6.3.3 Attention-aware neural painting

In this section, we introduce the whole pipeline of our attention-aware neural painting model (see Figure 6.3). The proposed network consists of an attention-guided policy net-

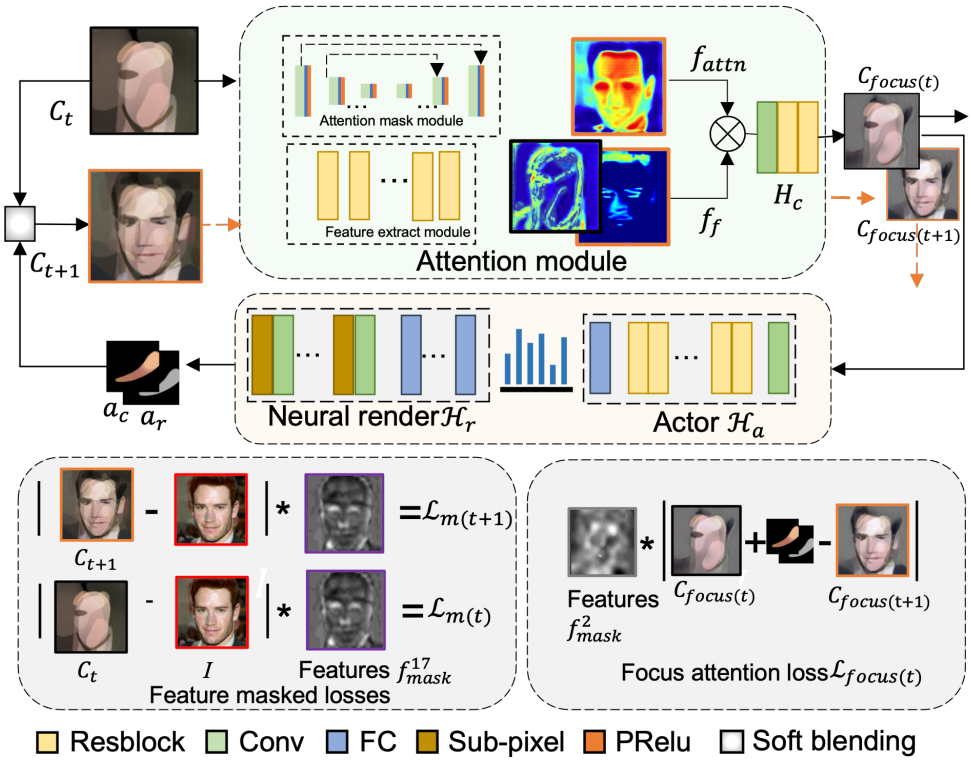


Figure 6.3: Architecture of our attention guided policy network and neural render network. At each timestep, the attention module takes the current canvas as input and outputs a foreground enhanced attention canvas, which is sent to the actor to predict a set of stroke parameters. The render network then transform the generated stroke parameters to canvas. The reward is computed using a feature masked losses at each step.

work and a neural render network. The former embeds an attention module before the actor $\mathcal{H}_a$, which is used for stroke parameters generation. One key ingredient of our approach is to enforce stroke parameters to focus on important object regions rather than to treat all regions on the canvas indiscriminately. Therefore, it is crucial that how to guide the agent to train the actor network properly. In this section, we achieve this goal through an attention module and feature masked reward functions.

6.3.3.1 Attention module

The foreground rewards are computed using an attention network. If the positions and scales of foreground objects vary significantly in different input images, then the inputs to the attention network would have high variance. Such high variance poses a direct challenge to the reward performance when training on complex real world datasets. To address this issue, we propose a differentiable neural attention model, which combines an attention mask network and a feature extract network to zoom into the foreground object, thereby providing a standardized input for the actor.

To quickly update the canvas to match the target image, we train the whole network end-to-end by jointly optimizing the attention module and the actor network. The attention module aims to prioritize areas that the current canvas differs from the target image the most and areas that contain high saliency foreground subjects. It consists of an encoder-decoder based attention mask module and a residual blocks-based feature extract module. The encoder-decoder attention mask module has shown great success in many segmentation [102] and generation [110] tasks. We adopt a fully convolutional model with skip connections to produce an attention mask map f_{attn}. To accelerate the convergence, we initialize the decoder with pretrained layers of ResNet34 [37]. The residual blocks module

Figure 6.4: Comparison on different rewards. When limited number of strokes are used, both VGG and WGAN yield blurry results. Our feature-masked reward encourages fine details, such as eyes and mouth, be represented at early stage.

extracts essential high-dimensional features f_f, which help to keep spatial information of the input canvas. We take every canvas as the input and learn an intermediate foreground focused image. The learning process and back-propagation of this attention module are as follows:

$$C_{focus(t)} = H_c(f_{attn} \odot \alpha f_f); \min_{\Theta} \mathcal{L}_{rec}\left(C_{focus(t)}, C_t \odot M\right) \tag{6.3}$$

where $\odot$ represents the element-wise product, H_c is reconstruction process with two convolutional layers to produce image with size $3 \times 64 \times 64$. We adopt the pixel-wise MSE loss to minimize the distance between reconstructed image and the masked canvas. Here M is the binary ground truth mask which is corresponding to the subject of the target image I. The generated foreground alerted image will contribute to the feature masked rewards which is discussed in the next subsection.

6.3.3.2 Reward Functions

For canvas updates, Huang et al. [45] applied WGAN reward based on *Wasserstein-l* distance to distinguish visual difference between the target image and the canvas. Compared with pixel-wise l_2 distance, WGAN loss shows some broad abstract information. How-

ever, the discriminator with objective $\max_D \mathbb{E}_{C\sim\mu}[D(C)] - \mathbb{E}_{I\sim\nu}[D(I)]$ based on WGAN distance only considers global features during the adversarial learning and does not focus enough on detailed but important features of foreground objects. For example, as shown in Figure 6.4, the WGAN-based agent approximates global information such as structure and colour very well, but misses characterizing fine details, such as eyes and hair at early steps. Besides, perceptual loss is also widely used in computer vision. We use selected layers of pretrained VGG-16 network [114] to calculate the distance between canvas feature maps and the target image feature maps by $\mathcal{L}_{vgg} = (VGG_l(c) - VGG_l(y))^2$. The agent can hardly produce satisfactory paintings under pure VGG reward especially at early stroke steps (see Figure 6.4). The reason is that the VGG is trained on real images rather than artistic paintings, hence the features extracted from canvas can not give enough strong signals and tend to make the final rendering rough and blur.

Different from Huang's work [45], we adopt feature masked losses in our attention aware model-based RL. The purpose of feature masked losses are to increase the importance of high saliency subject regions on the target image. As shown in Equation 6.4, feature masks act as a weight for pixel distances between paintings and the target image and hence guide the agent to accurately capture the appearances of important foreground objects and ignore minor differences in backgrounds. The feature masks are extracted feature maps of the target image from VGG-16 model [114]. 0-1 normalized features from different layers of VGG-16 represent different information of the image. To train the actor network for producing strokes in the recognition-related regions, we adopt two reward strategies.

First, we minimize the pixel distance between current canvas C_t and the target image I to encourage C_t to quickly approximate I in a coarse-to-fine manner. The canvas is initiated

with pixel value 0 and hence strokes with brighter colours tend to receive a higher reward since they can reduce pixel distances more dramatically. To weaken the bias, we use l_1 loss rather than l_2 loss before weighting it by feature masks. The feature masked loss $\mathcal{L}_m$ is the feature weighted differences between current canvas and target image. The feature masked reward $\mathcal{R}_m$ of canvas is calculated by computing the differences between canvas C and the target image I at timestep t and $t+1$. The loss and reward functions are defined as follows:

$$\begin{aligned}\mathcal{L}_{m(t)} &= \min_{\Theta}(|C_t - I|, \lambda) \odot f^l_{mask} + \mathcal{L}_{vgg(t)} \\ \mathcal{L}_{m(t+1)} &= \min_{\Theta}(|C_{t+1} - I|, \lambda) \odot f^l_{mask} + \mathcal{L}_{vgg(t+1)}\end{aligned} \tag{6.4}$$

$$\mathcal{R}_{m(t)} = \text{mean}\left[\mathcal{L}_{m(t)}(C_t, I) - \mathcal{L}_{m(t+1)}(C_{t+1}, I)\right] \tag{6.5}$$

where feature mask $f^l_{mask} = \text{norm}(VGG_l(I))$ is normalized feature maps extracted from the l^{th} layer of VGG-16 when inputting the target image I. f^l_{mask} indicates important regions on the canvas. By maximizing the reward, our agent tends to produce strokes that refine these important regions.

Second, we calculate another reward using foreground focused image C_{focus}, which is generated by the attention model. At each step, a canvas C_t is transformed to C_{t+1} by blending with the generated strokes. The C_{t+1} at timestep $t+1$ will be mapped to an intermediate focused image $C_{focus(t+1)}$. If the generated strokes at timestep t attempt to locate on the foreground subject, then the composition of the strokes and the last focused image $C_{focus(t)}$ should have smaller distance with $C_{focus(t+1)}$. That is, applying strokes on the foreground region will receive more rewards. We denote the stroke-blended focused image as: $C^{blend}_{focus(t)} = a_c + (1 - a_r)C_{focus(t)}$ and calculate a new attention loss and a

corresponding reward by:

$$\mathcal{L}_{focus(t)} = \min_{\Theta}(|C^{blend}_{focus(t)} - C_{focus(t+1)}|, \lambda) \odot f^{l}_{mask} \tag{6.6}$$

$$\mathcal{R}_{focus(t)} = \text{mean}\left[\mathcal{L}_{focus(t)} - \mathcal{L}_{focus(t+1)}\right] \tag{6.7}$$

To fit with the focused image, the stroke tensor a and the feature map f^{l}_{mask} are both downsized from 128×128 to 64×64.

It is a known fact that lower level layers capture overall subject shapes (e.g. edges and texture) and higher level layers reflect details (e.g. mouth and eyes). Hence, we choose features from layer-2 of VGG-16 to maximize the foreground focused reward and select layer-17 which can balance both high and low features to maximize the whole canvas reward.

The final reward function $\mathcal{R}(C_t, a_t)$ is weighted combination of the feature masked reward and the focus reward:

$$\mathcal{R}(C_t, a_t) = \sigma\mathcal{R}_{m(t)} + \eta\mathcal{R}_{focus(t)} \tag{6.8}$$

where σ and η are hyperparameters, which are fixed at $\sigma = 2$, $\eta = 1$ to balance the contributions of the two rewards. $\mathcal{R}\left(C_t, a_t\right)$ is the reward for executing action a_t at state C_t.

An expected value function $V(s)$ is predicted by the critic:

$$V\left(C_t\right) = \mathcal{R}\left(C_t, a_t\right) + \gamma V\left(C_{t+1}\right) \tag{6.9}$$

where critic is a network to predict new expected reward for the state in model-based Deep Deterministic Policy Gradient (DDPG) [45]. We use neural render to map the stroke parameter a_t to canvas: $C_{t+1} = \mathcal{F}\left(C_t, a_t\right)$.

6.4 Experiments

6.4.1 Datasets and training

In our task, we evaluate the proposed method on two widely used datasets: CelebA-HQ [68] and CUB-200-2011 Birds [139]. CelebA contains about 30,000 examples and we directly use the officially released cropped faces and masks. CUB contains 200 bird species with 11,788 images.

All training images are resized to 128×128. Our model is implemented on PyTorch and tested on a single nVidia GeForce GTX 1080 Ti GPU with 50GiB memory. We apply the batch normalization, set a fixed learning rate of $2e^{-4}$ for the attention module, and use Adam algorithm as the optimizer. Action bundle is set to $k = 5$. We stop training after 400 epochs. It took about 70 hours for training on CelebA and 24 hours to train on CUB Birds data. At each iteration, we update the attention module, critic and actor in turn.

6.4.2 Final Painting Results and comparisons

We first show the results on selected images from both the CelebA and CUB Bird datasets; see Figure 6.5. The results show that our model can effectively generate the appearance of the target images in a coarse-to-fine manner. In the final results obtained using 400 strokes, high saliency features (eyes, mouth, feathers, claws, etc.) are nicely captured, whereas the backgrounds are kept blurry. The results suggest that we achieved the design goal of our attention-aware neural painting.

Figure 6.6 illustrates increased saliency of intermediate foreground focused objects. By focusing attention on the foreground features, we try to reduce a detrimental influence

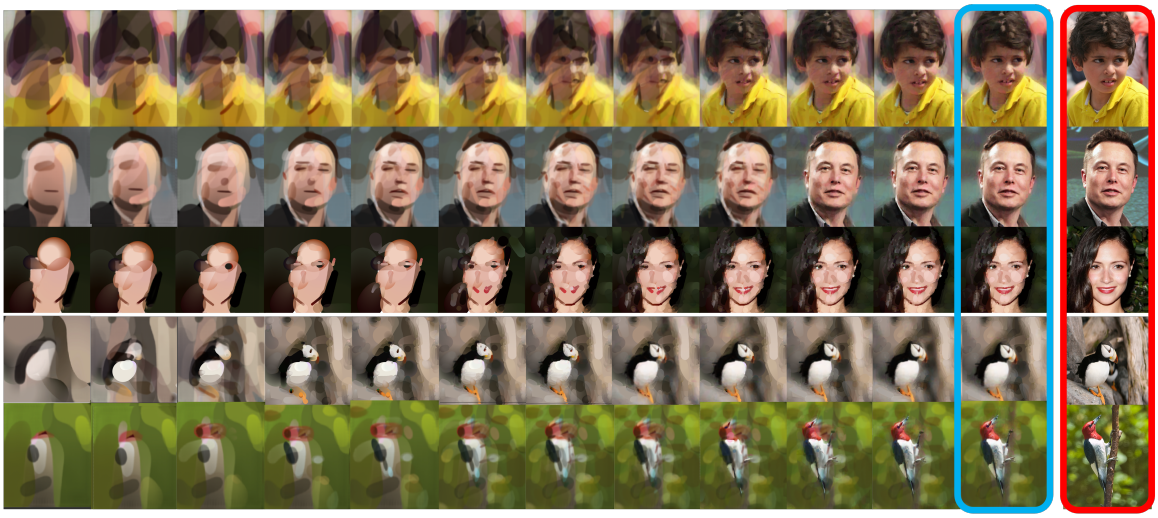


Figure 6.5: Painting results under different stroke numbers by our method. The blue box shows the results with 400 strokes and the red box shows the original target images. Our results nicely captures high saliency foreground details, such as eyes and mouths of persons and feathers, beaks, and claws of birds. The backgrounds are relatively blurry.

possibly present on other elements in the image during the focused reward guidance.

We further compare our method with state-of-the-art work Huang et al. [45] and Zou et al. [167] in Figure 6.7. In both cases, our approach better approximates the target image under small number of strokes and capture finer foreground details in the final results, thanks to its attention-aware painting strategy. The rough sketch and edges of mouth, eyes and hairs are more distinct which is benefit from the use of feature masked rewards. In contrast, Huang et al. [45] and Zou et al. [167] treat every pixel of target image equally resulting in the lack of the priority of key objects.

To quantitatively compare our model with Huang et al. [45], we feed paintings produced by both under different number of strokes into a facial detection model (FaceNet) [108]. 800 example images are used and the detection success rates are plotted in Figure. 6.8. The

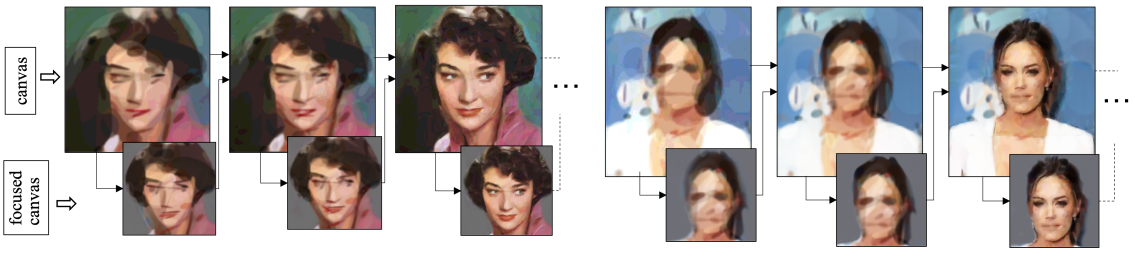


Figure 6.6: Illustration of generated canvas with the corresponding intermediate object focused results. The foreground attention module provides a differentiable enhancement for the foreground object regions in the updated canvas C. These focused object images are then used to compute the foreground focused reward (Eq. 6.7).

plot suggests that, with the same number of strokes, our model generates more recognizable faces.

To evaluate whether our approach can better mimic human artists preference of selecting brushes from big to small, we also compare the average brush thicknesses used by both our proposed model and Huang et al. [45] at different stages of the painting. The thickness of a given brush stroke is extracted from the 7^{th} and 9^{th} parameters in a_t (see Eq. 6.1), as they specify stroke sickness at its two ends. We randomly select 10 CelebA images and 10 CUB Birds examples from the two datasets and calculate the average stroke thickness for every 10 strokes on both models. The results, shown in Figure 6.9, suggests that both models follow the course-to-fine manner, but our actor changes to finer brushes early. The differences in stroke thickness is more noticeable for the bird images, which contains more detailed features. Figure 6.10 qualitatively compares the results generated by our approach with those manually painted by human artists (credited on the artworks). We can see that artists like to focus on the key characters and weaken the unimportant background during artistic creation. Our approach successfully mimics this behaviour, even though the paint-

Figure 6.7: Step-wise comparisons of painting canvas generated by our method (red boxes), Huang et al. [45], and Zou et al. [167] under the same amount strokes. Our results show enhanced foreground saliency and more details in terms of facial appearance and birds features at early stages, as well as offering better details in important regions such as eyes and feathers.

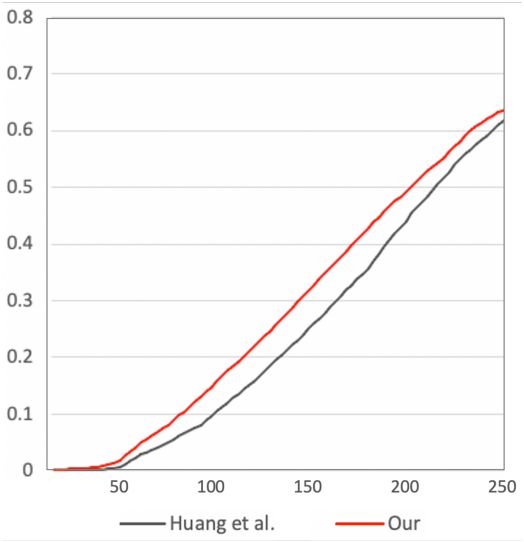


Figure 6.8: Face detection success rates of paintings produced by our model (red line) and Huang et al. [45] (black line). When limited number of strokes are used, paintings produced by our model are more successfully recognized by FaceNet [108]

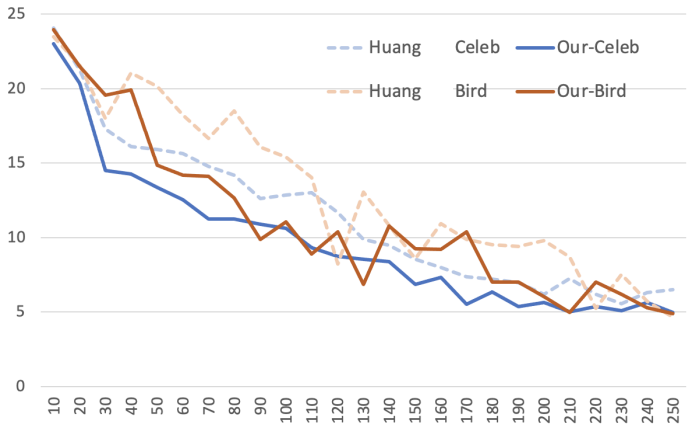


Figure 6.9: Average thickness of strokes applied by our model and Huang et al. [45] during the painting process. Our model tends to apply finer strokes earlier, which implies better handling of fine details.

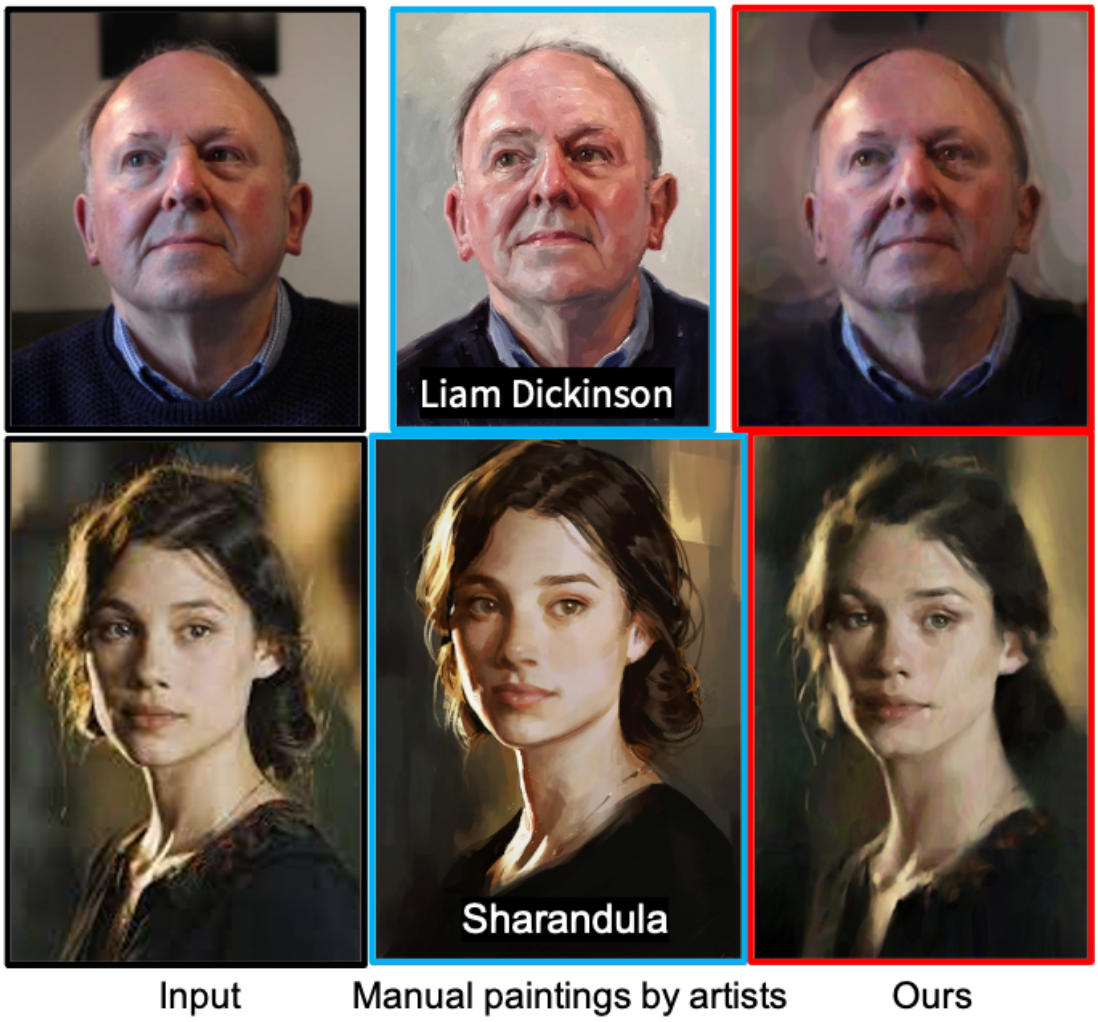


Figure 6.10: Visual comparison between paintings drew by manually artists and generated by our method. Our method handles foregrounds and backgrounds differently, which is similar to human artists.

ings by artists were not used for training. In addition, our computer generated results have the advantage in faithfully following the structures of the input image. For example, the location of the top of the lady's head and the shape of her collar are more accurate in our results than in artist's drawings.

6.4.3 Ablation study

We perform an ablation study on the key components in our deep reinforcement learning network and the following results show that all of them are essential to the efficacy of our proposed method and they jointly produce high-quality results of salient regions. **Effectiveness of foreground focused reward.** As a human painter paints, the process of the painting should follow course-to-fine scheme and apparent foreground saliency in early steps. Overall, Figure 6.11 shows the intermediate sequential results by different loss functions. The impact of focus reward proposed in Section 6.3.3.2 reflects the effectiveness of the attention module. To isolate the effect of the focus reward, we conduct a baseline experiment in which the attention module is removed so that the focus reward is eliminate. Perceptual losses are widely used in image generation in the computer vision area. As seen in Figure 6.11 (yellow box), the sequential results on the first two rows show the effect on the resulting canvas with/without the focus reward. Without the foreground loss, we clearly see that the model trained with VGG reward only captures the rough shape and colour. The agent behaves similar with that in SPIRAL [22] but it fails to accurately pay attention to finer features. The focus reward based on the pixel-level attention loss gives strong signal to the foreground information and hence improves the content saliency.

Effectiveness of different losses. In the rendering process, a strong and direct reward

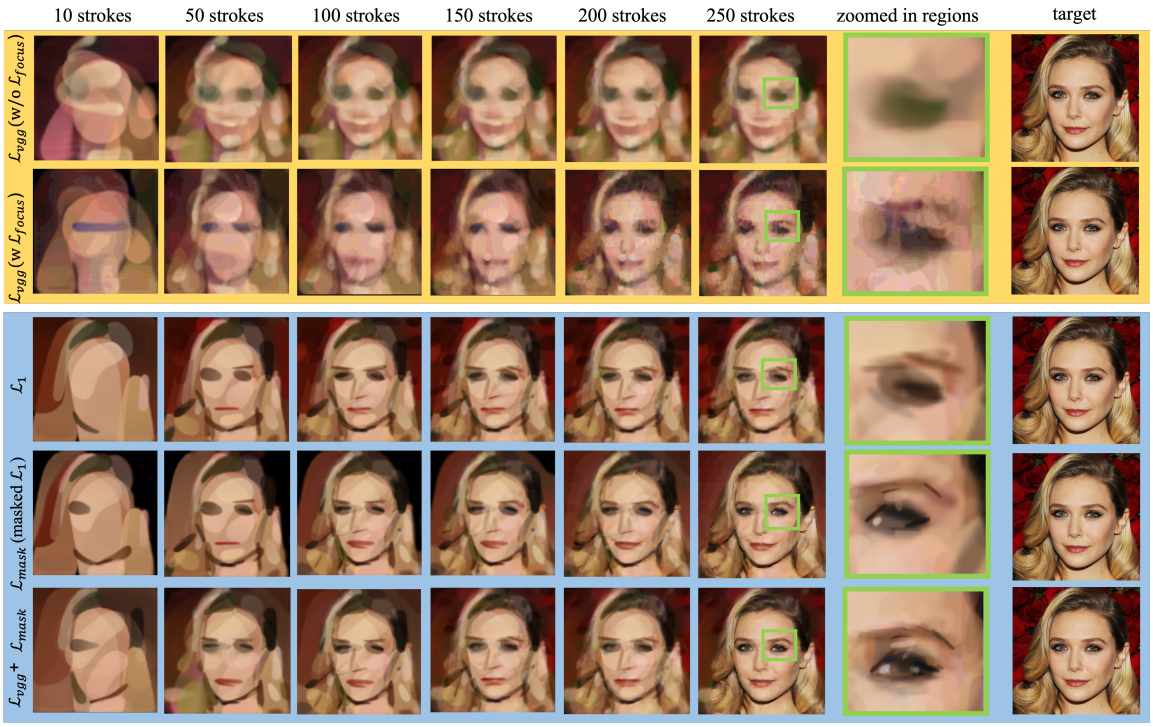


Figure 6.11: The canvas based on different strokes for models trained using different rewards. Zoomed in views on the eye region are provided in the green boxes. Overall, VGG reward only generates very coarse appearance, whereas L1 reward provides more details in comparison. Detailed features of the foreground object are added when VGG reward accompanies with the foreground object focus reward. Feature masked loss L_1 (L_{mask}) is L_1 weighted by features extracted from selected layer of VGG16 fed by the target image. Feature masked reward yields strong signals to the details such as edges, eyes and mouth, which allows the foreground content to be more recognizable faster and achieves better granularity of key object features. In conjunction with the VGG reward, both rough background and enhanced foreground are generated as a whole.

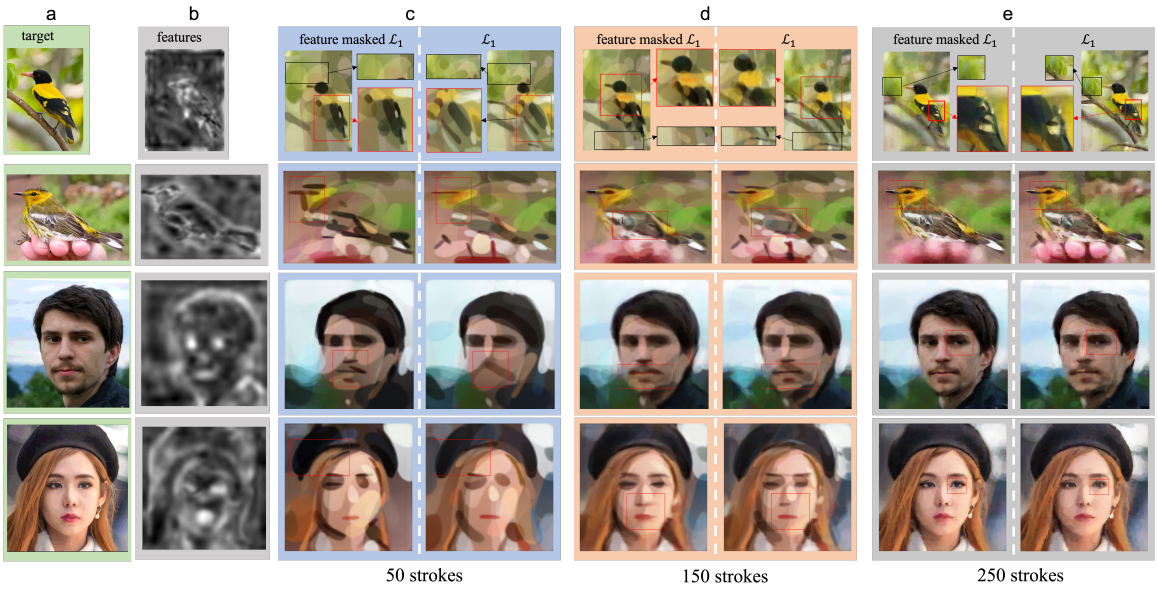


Figure 6.12: Ablation results for featured masks. From column a to column e, there are target images, features extracted from target images, comparison pairs of generated canvas with feature masked L_1 and generated canvas with L_1 under stroke 50, 150, 250, respectively. It is clear that a higher quality detailed image with finer feature details like edges, wing texture, density of eyes is generated with the feature masked reward included.

signal from pixel level comparison is important to such painting environment. As seen in Figure 6.11 (blue box), L_1 reward have good performance but it has no special treatment on the foreground instance in the target image. The $\mathcal{L}_{\text{mask}}$ reward based on feature masked L_1 gives strong signals to the important regions and crucial details for recognizing the object in the image. To guarantee the integrality of the rendered canvas, We combine both VGG reward and masked feature reward to capture both coarse background and the fine important regions at the same time.

To specify the influence from feature masked reward, we use birds images from CUB-200-2011 dataset and face images from CelebA to compare the foreground and background

differences. VGG reward is used in both method for fair comparison. Figure 6.12 shows the training results. It is clear that the model trained with feature masked L_1 captures more image information in limited strokes and has strong ability to capture finer details such as textures, beaks, eyes and the hair (red boxes). In comparison, model trained without feature masks ignores the stroke priority of different regions. It applies more strokes on the background (black boxes) at the expense of generating less detail on the key objects.

6.5 Summary

In this chapter, we propose a new end-to-end attention-aware reinforcement learning approach for painting like humans. Our method incorporates a dual-branch attention module and two feature masked losses to prioritize the handling of high saliency regions. In the attention module, we fuse attention feature maps with structure feature details, forming a new intermediate canvas where the details of foreground subjects are locally emphasized. Based on the intermediate canvas and real canvas which is to be updated, we adopt two masked feature rewards to promote the training of the actor. The results show that our approach better approximates the target image under small number of strokes and capture finer foreground details in the final results. The overall painting process uses coarse-to-fine strokes more aggressively and the results generated nicely resembles those painted by human artists.

Chapter 7

Conclusion and Future Work

Using four different visual synthesis tasks as targets, this thesis presents novel deep learning-based algorithms, which involves feature fusion, different attention mechanisms, and inverse data distribution learning. The experimental results show that, compared to the existing state-of-the-arts, the techniques developed can produce higher quality images and videos under challenging conditions, higher condition-object matching, and more human-like artistic creation process.

More specifically, the proposed HfGAN [43] suggests that local and global feature fusion can avoid training instability and nonsensical outputs caused by simply adding more upsampling layers in the state-of-the-art one-stage GAN model. When it comes to face generation under different ages, the presented LDcGAN [42] trained on the facial landmark attention is capable of generating faces with more accurate features related to age information. For the talking face videos generation, the presented AVWnet [44] adopts multi-level audio/landmark attentions and a reinterpretation process to generate high fidelity talking face videos with more accurate mouth movements. Finally, an attention-aware DRL ap-

proach is discussed to better mimic human painting behaviours, which can considerably enhance the details of important regions in early painting steps.

Conducting research on four highly different tasks also provides insights on how to design visual synthesis methods in general. Two strategies are found to be effective across-applications. Attention mechanisms are applied in all four tasks: to associate text descriptions with image features, to focus on facial landmarks, and to select high priority areas to paint like humans. In both aging face image synthesis and talking face video generation tasks, adding feedback loop is found to be effective for enhancing the consistency between input conditions and generated results.

Using GANs to stably generate high-resolution images and videos has important real-world applications. Two potential future directions are discussed below.

"High fidelity of small-scale details in visual synthesis". The attribute features of the image come from each layer of the decoder. After multiple layers of convolutional down-sampling, some small-scale objects (e.g., earrings) in the original image are ignored because the features of the decoder part have a large perceptual field. This makes the model lose some details, although it can keep the macro attributes such as expression and pose well. Therefore, how to find a better way to characterize attribute information is a direction worthy of future research. The features extracted from face images at different resolutions should have different levels of importance for the retention of attribute information and should not be independent of each other. Some existing works [36] introduced Transformer [136] structure into feature extraction process to effectively fuse global multi-scale information through self-attention mechanism. Such an approach can help addressing the aforementioned problem and is a worthy direction to pursue.

"Attack and Defense". While this thesis aims at synthesizing more realistic images

and videos, we should note that fake content creation can be a threat. With enhancements made in content creative systems in specific domains like human faces and the emergence of DeepFakes [148], it has been a challenge to rely on what we see on the web. Fake images, videos, and sounds generated by deep learning can often fool humans, thus raising security and privacy concerns. Therefore, robust and explainable detectors would be studied to defend against fake information.

Bibliography

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015.

[2] G. Antipov, M. Baccouche, and J.-L. Dugelay. Boosting cross-age face verification via generative age normalization. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 191–199. IEEE, 2017.

[3] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. In *2017 IEEE international conference on image processing (ICIP)*, pages 2089–2093. IEEE, 2017.

[4] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

[5] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

[6] S. P. Bartlett, R. Grossman, and L. A. Whitaker. Age-related changes of the craniofacial skeleton: an anthropometric and histologic analysis. *Plastic and reconstructive surgery*, 90(4):592–600, 1992.

[7] C. Bregler, M. Covell, and M. Slaney. Video rewrite: driving visual speech with audio. In *Siggraph*, volume 97, pages 353–360, 1997.

[8] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017.

[9] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *European conference on computer vision*, pages 768–783. Springer, 2014.

[10] L. Chen, Z. Li, R. K Maddox, Z. Duan, and C. Xu. Lip movements generation at a glance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 520–535, 2018.

[11] L. Chen, R. K. Maddox, Z. Duan, and C. Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7832–7841, 2019.

[12] J. S. Chung, A. Jamaludin, and A. Zisserman. You said that? *arXiv preprint arXiv:1705.02966*, 2017.

[13] J. S. Chung and A. Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, pages 87–103. Springer, 2016.

[14] M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424, 2006.

[15] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–II. IEEE, 2003.

[16] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal. Tac-GAN-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017.

[17] Z. Deng and U. Neumann. Expressive speech animation synthesis with phoneme-level controls. In *Computer Graphics Forum*, volume 27, pages 2096–2113. Wiley Online Library, 2008.

[18] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[19] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[20] B. Fan, L. Xie, S. Yang, L. Wang, and F. K. Soong. A deep bidirectional LSTM approach for video-realistic talking head. *Multimedia Tools and Applications*, 75(9):5287–5309, 2016.

[21] Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 32(11):1955–1976, 2010.

[22] Y. Ganin, T. Kulkarni, I. Babuschkin, S. A. Eslami, and O. Vinyals. Synthesizing programs for images using reinforced adversarial learning. In *International Conference on Machine Learning*, pages 1666–1675. PMLR, 2018.

[23] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[24] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[25] J. Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, 2014(5):2, 2014.

[26] 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*, pages 2672–2680, 2014.

[27] 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*, pages 2672–2680, 2014.

[28] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[29] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

[30] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471. PMLR, 2015.

[31] P. J. Grother, M. L. Ngan, and K. K. Hanaoka. Ongoing face recognition vendor test (frvt) part 2: Identification. 2018.

[32] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[33] D. Ha and D. Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

[34] P. Haeberli. Paint by numbers: Abstract image representations. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 207–214, 1990.

[35] P. Harrison. A non-hierarchical procedure for re-synthesis of complex textures. 2001.

[36] A. Hatamizadeh, D. Yang, H. Roth, and D. U. Xu. Transformers for 3d medical image segmentation. arxiv 2021. *arXiv preprint arXiv:2103.10504*.

[37] 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*, pages 770–778, 2016.

[38] M. Hellman. Changes in the human face brought about by development. *International Journal of Orthodontia, Oral Surgery and Radiography*, 13(6):475–516, 1927.

[39] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460, 1998.

[40] A. Hertzmann. A survey of stroke-based rendering. Institute of Electrical and Electronics Engineers, 2003.

[41] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[42] X. Huang and M. Gong. Landmark-guided conditional GANs for face aging. In *2021 21th International Conference on Image Analysis and Processing(ICIAP)*. IEEE, 2022.

[43] X. Huang, M. Wang, and M. Gong. Hierarchically-fused generative adversarial network for text to realistic image synthesis. In *2019 16th Conference on Computer and Robot Vision (CRV)*, pages 73–80. IEEE, 2019.

[44] X. Huang, M. Wang, and M. Gong. Fine-grained talking face generation with video reinterpretation. *The Visual Computer*, 37(1):95–105, 2021.

[45] Z. Huang, W. Heng, and S. Zhou. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8709–8718, 2019.

[46] Q. Huynh-Thu and M. Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[47] A. Hyvarinen, J. Karhunen, and E. Oja. Independent component analysis. *Studies in informatics and control*, 11(2):205–207, 2002.

[48] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

[49] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[50] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[51] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018.

[52] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *TOG*, 36(4):94, 2017.

[53] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[54] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018.

[55] 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*, pages 3334–3341, 2014.

[56] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[57] A. Lanitis, C. J. Taylor, and T. F. Cootes. Toward automatic simulation of aging effects on face images. *IEEE Transactions on pattern Analysis and machine Intelligence*, 24(4):442–455, 2002.

[58] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2017.

[59] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[60] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

[61] Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin. Video generation from text. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[62] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[63] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[64] P. Litwinowicz. Processing images and video for an impressionist effect. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 407–414, 1997.

[65] S. Liu, T. Li, W. Chen, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019.

[66] S. Liu, Y. Sun, D. Zhu, R. Bao, W. Wang, X. Shu, and S. Yan. Face aging with contextual generative adversarial nets. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 82–90, 2017.

[67] Y. Liu, Q. Li, and Z. Sun. Attribute-aware face aging with wavelet-based generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11877–11886, 2019.

[68] Z. Liu, P. Luo, X. Wang, and X. Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[69] M. M. Loper and M. J. Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

[70] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[71] S. Ma, J. Fu, C. Wen Chen, and T. Mei. DA-GAN: Instance-level image translation by deep attention generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5657–5666, 2018.

[72] X. Ma and Z. Deng. A statistical quality model for data-driven speech animation. *IEEE transactions on visualization and computer graphics*, 18(11):1915–1927, 2012.

[73] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[74] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[75] I. Megvii. Face++ research toolkit, 2013.

[76] J. F. Mellor, E. Park, Y. Ganin, I. Babuschkin, T. Kulkarni, D. Rosenbaum, A. Ballard, T. Weber, O. Vinyals, and S. Eslami. Unsupervised doodling and painting with improved spiral. *arXiv preprint arXiv:1910.01007*, 2019.

[77] B. Mendelson and C.-H. Wong. Changes in the facial skeleton with aging: implications and clinical applications in facial rejuvenation. *Aesthetic Plastic Surgery*, 44(4):1151–1158, 2020.

[78] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[79] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[80] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[81] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. 2015.

[82] R. Nakano. Neural painters: A learned differentiable constraint for generating brushstroke paintings. *arXiv preprint arXiv:1904.08410*, 2019.

[83] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4467–4477, 2017.

[84] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.

[85] Y. Nirkin, Y. Keller, and T. Hassner. FsGAN: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7184–7193, 2019.

[86] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[87] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pages 2863–2871, 2015.

[88] U. Park, Y. Tong, and A. K. Jain. Age-invariant face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):947–954, 2010.

[89] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *bmvc*, volume 1, page 6, 2015.

[90] H. Permuter, J. Francos, and I. H. Jermyn. Gaussian mixture models of texture and colour for image database retrieval. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 3, pages III–569. IEEE, 2003.

[91] S. Petridis, Z. Li, and M. Pantic. End-to-end visual speech recognition with LSTMs. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2592–2596. IEEE, 2017.

[92] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic. End-to-end audiovisual speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6548–6552. IEEE, 2018.

[93] Q. Pham, J. Yang, and J. Shin. Semi-supervised FaceGAN for face-age progression and regression with synthesized paired images. *Electronics*, 9(4):603, 2020.

[94] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2018.

[95] T. Qiao, J. Zhang, D. Xu, and D. Tao. MirrorGAN: Learning text-to-image generation by redescription. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019.

[96] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[97] N. Ramanathan and R. Chellappa. Modeling age progression in young faces. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 387–394. IEEE, 2006.

[98] N. Ramanathan and R. Chellappa. Modeling shape and textural variations in aging faces. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–8. IEEE, 2008.

[99] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.

[100] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.

[101] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *Advances in Neural Information Processing Systems*, pages 217–225, 2016.

[102] 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*, pages 234–241. Springer, 2015.

[103] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2830–2839, 2017.

[104] H. Sak, A. Senior, and F. Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[105] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[106] N. Savaliya. Faceold : Old ageing app. `https://apps.apple.com/us/app/make-me-old-old--face/id1476921217`, 2021.

[107] D. Saxena and J. Cao. Generative adversarial networks (GANs) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 54(3):1–42, 2021.

[108] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[109] T. Shi, Y. Yuan, C. Fan, Z. Zou, Z. Shi, and Y. Liu. Face-to-parameter translation for game character auto-creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 161–170, 2019.

[110] A. Shocher, N. Cohen, and M. Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018.

[111] 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*, pages 3970–3978, 2015.

[112] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5541–5550, 2017.

[113] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

[114] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[115] C. Song, Z. Wu, Y. Zhou, M. Gong, and H. Huang. Etnet: Error transition network for arbitrary style transfer. *Conference on Neural Information Processing Systems (Proceedings of NeurIPS 2019)*, pages 668–677, 2019.

[116] Y. Song, J. Zhu, X. Wang, and H. Qi. Talking face generation by conditional recurrent adversarial network. *arXiv preprint arXiv:1804.04786*, 2018.

[117] T. Stafylakis and G. Tzimiropoulos. Combining residual networks with LSTMs for lipreading. *arXiv preprint arXiv:1703.04105*, 2017.

[118] L. Studio. Facelab old face, photo editor. `https://apps.apple.com/us/app/facelab-old-face-photo-editor/id1530776865`, 2020.

[119] J. Suo, X. Chen, S. Shan, W. Gao, and Q. Dai. A concatenational graph evolution aging model. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2083–2096, 2012.

[120] 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*, 32(3):385–401, 2009.

[121] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[122] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017.

[123] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[124] H. Tang, D. Xu, N. Sebe, and Y. Yan. Attention-guided generative adversarial networks for unsupervised image-to-image translation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[125] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, 36(4):93, 2017.

[126] Y. Tazoe, H. Gohara, A. Maejima, and S. Morishima. Facial aging simulator considering geometry and patch-tiled texture. In *ACM SIGGRAPH 2012 Posters*, pages 1–1. 2012.

[127] D. Teece. 3d painting for non-photorealistic rendering. In *ACM SIGGRAPH 98 Conference abstracts and applications*, page 248, 1998.

[128] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[129] B. Tiddeman, M. Burt, and D. Perrett. Prototyping and transforming facial textures for perception research. *IEEE computer graphics and applications*, 21(5):42–50, 2001.

[130] J. T. Todd, L. S. Mark, R. E. Shaw, and J. B. Pittenger. The perception of human growth. *Scientific american*, 242(2):132–145, 1980.

[131] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. MocoGAN: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[132] G. Turk and D. Banks. Image-guided streamline placement. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 453–460, 1996.

[133] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587. IEEE Computer Society, 1991.

[134] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with PixelCnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

[135] A. Van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016.

[136] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[137] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[138] K. Vougioukas, S. Petridis, and M. Pantic. End-to-end speech-driven realistic facial animation with temporal GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 37–40, 2019.

[139] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[140] V. Wan, R. Anderson, A. Blokland, N. Braunschweiler, L. Chen, B. Kolluru, J. Latorre, R. Maia, B. Stenger, K. Yanagisawa, et al. Photo-realistic expressive text to talking head synthesis. In *INTERSPEECH*, pages 2667–2669, 2013.

[141] H. Wang, D. Gong, Z. Li, and W. Liu. Decorrelated adversarial learning for age-invariant face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3527–3536, 2019.

[142] W. Wang, Z. Cui, Y. Yan, J. Feng, S. Yan, X. Shu, and N. Sebe. Recurrent face aging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2378–2386, 2016.

[143] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016.

[144] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[145] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[146] Z. Wang, X. Tang, W. Luo, and S. Gao. Face aging with identity-preserved conditional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7939–7947, 2018.

[147] Z. Wang, Y. Yu, and D. Zhang. Best neighborhood matching: An information loss restoration technique for block-based image coding systems. *IEEE Transactions on Image Processing*, 7(7):1056–1061, 1998.

[148] M. Westerlund. The emergence of deepfake technology: A review. *Technology Innovation Management Review*, 9(11), 2019.

[149] O. Wiles, A. Sophia Koepke, and A. Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–686, 2018.

[150] C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

[151] L. Xie and Z.-Q. Liu. Realistic mouth-synching for speech-driven talking face using articulatory modelling. *IEEE Transactions on Multimedia*, 9(3):500–510, 2007.

[152] N. Xie, H. Hachiya, and M. Sugiyama. Artist agent: A reinforcement learning approach to automatic stroke generation in oriental ink painting. *IEICE TRANSACTIONS on Information and Systems*, 96(5):1134–1144, 2013.

[153] L. Xu and M. I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

[154] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018.

[155] H. Yang, D. Huang, Y. Wang, H. Wang, and Y. Tang. Face aging effect simulation using hidden factor analysis joint sparse representation. *IEEE Transactions on Image Processing*, 25(6):2493–2507, 2016.

[156] H. Yehia, P. Rubin, and E. Vatikiotis-Bateson. Quantitative association of vocal-tract and facial behavior. *Speech Communication*, 26(1-2):23–43, 1998.

[157] Z. Yi, H. Zhang, P. Tan, and M. Gong. DualGAN: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.

[158] K. Zeng, M. Zhao, C. Xiong, and S. C. Zhu. From image parsing to painterly rendering. *ACM Trans. Graph.*, 29(1):2–1, 2009.

[159] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[160] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stack-GAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.

[161] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stack-GAN++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.

[162] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5810–5818, 2017.

[163] Z. Zhang, Y. Xie, and L. Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[164] N. Zheng, Y. Jiang, and D. Huang. Strokenet: A neural painting environment. In *International Conference on Learning Representations*, 2018.

[165] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9299–9306, 2019.

[166] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[167] Z. Zou, T. Shi, S. Qiu, Y. Yuan, and Z. Shi. Stylized neural painting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15689–15698, 2021.