Coarse-to-Fine Clothing Image Generation with Progressively Constructed Conditional GAN

Youngki Kwon[1], Soomin Kim[1], Donggeun Yoo[2], and Sung-Eui Yoon[1]
[1]*School of Computing, KAIST, Daejeon, Republic of Korea*
[2]*Lunit Inc, Seoul, Republic of Korea*

Keywords: Generative Adversarial Networks, Image conditional image generation, cloth image generation, Coarse-to-Fine

Abstract: Clothing image generation is a task of generating clothing product images from input fashion images of people dressed. Results of existing GAN based methods often contain visual artifact with the global consistency issue. To solve this issue, we split the difficult single image generation process into relatively easy multiple stages for image generation process. We thus propose a coarse-to-fine strategy for the image-conditional image generation model, with a multi-stage network training method, called rough-to-detail training. We incrementally add a decoder block for each stage that progressively configures an intermediate target image, to make the generator network appropriate for rough-to-detail training. With this coarse-to-fine process, our model can generate from small size images with rough structures to large size images with details. To validate our model, we perform various quantitative comparisons and human perception study on the LookBook dataset. Compared to other conditional GAN methods, our model can create visually pleasing 256 × 256 clothing images, while keeping the global structure and containing details of target images.

1 INTRODUCTION

When we see pictures of celebrities, we often want to know what clothes he or she wears and where we can buy those clothes. For this, we first need to perform the image search with pictures of celebrity as queries. However, results might contain irrelevant images, fundamentally because pictures of celebrities and cloth product images belong to different domains. Generally, a picture of celebrities consists of a clothing object, that we are looking for, and unnecessary regions such as background. A clothing product image, however, contains only clothing objects themselves. This semantic and visual gap between two domains can be obstacles for searching intended clothing product images. To avoid this, we utilize clothing image generation.

In this paper, we define a clothing image generation as a task of creating clothing images (product images) from any input images of people dressed. The generated images must contain an apparel-like object with details consistent with the input images. The resulting images must be realistic and visually plausible, as well (Figure 1).

Our problem of clothing image generating can be approached in the perspective of image-conditional image generation. In image-conditional image generation problem, the conditional Generative Adversarial Network (GAN) (Mirza and Osindero, 2014; Goodfellow et al., 2014) based approaches have shown remarkable results (Pathak et al., 2016; Iizuka et al., 2017; Isola et al., 2017; Lassner et al., 2017; Ledig et al., 2017; Zhu et al., 2017). In practice, however, result images generated by GAN often contain visual artifacts with a global consistency issue; objects in an image are structurally collapsed (Goodfellow, 2016); see Figure 3 (e). It can be worse in high-resolution images (Arjovsky and Bottou, 2017; Arjovsky et al., 2017).

To mitigate these artifacts, many studies have applied various computer vision techniques to GAN. The coarse-to-fine strategy is one of the classical approaches in computer vision (Szeliski, 2010) for structured prediction, and GAN with coarse-to-fine approaches have shown acceptable results, even when generating a high-resolution image (Zhang et al., 2016; Zhao et al., 2017; Denton et al., 2015; Karras et al., 2017; Mathieu et al., 2015). These works have split the difficult single image generation process into easier multi-stages for the image generation process. An image generation model with this strategy generates target images from smaller scale images with rough structures and then gradually generates bigger scale images with details. As a result, these ap-

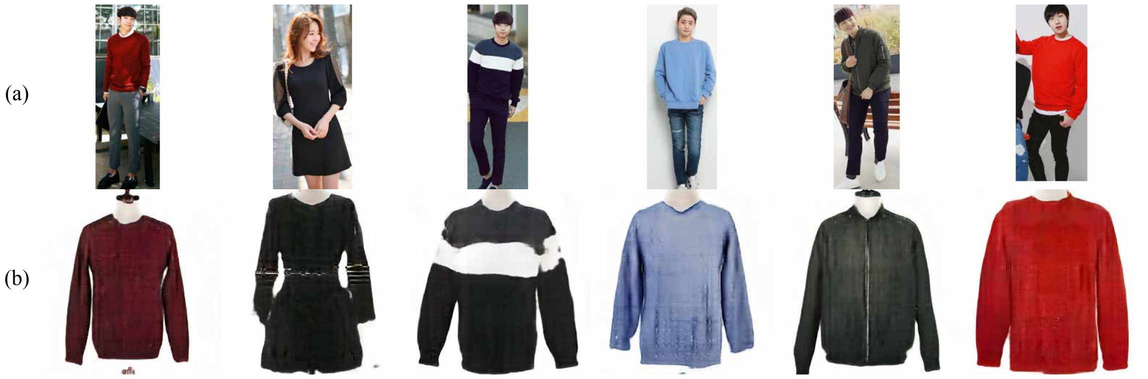


Figure 1: Examples of clothing images generated by our model. (a) are fashion model images as input. (b) are product images generated by our model conditioned on the input images (a).

proaches can avoid generating structurally collapsed result images.

Unfortunately, previous studies (Zhang et al., 2016; Zhao et al., 2017; Denton et al., 2015; Mathieu et al., 2015) have used multiple pairs of generators and discriminators for stages in order to implement this strategy, causing an excessive amount of network parameters.

Main contributions. In this paper, we propose a novel image-conditional image generation model, rough-to-detail conditional GAN, for clothing image generation. Our model is designed to utilize the coarse-to-fine approach to produce visually pleasing clothing images in a high resolution. During network training, our model progressively constructs a generator for a target image via adding decoder blocks sequentially (Section 3.3). In this way with only a single pair of a generator and a discriminator, we can use network parameters in a compact way, and thus allow to use a large minibatch size during optimization for accurate gradients, resulting in high-quality image generation (Salimans et al., 2016; Salimans et al., 2018).

Compared to other conditional GAN models, result images generated by our model both look like realistic and contain detailed apparel-like objects consistent with the input images (Section 5.2). As a result, our result image achieves better performance in quantitative evaluation with various metrics such as RMSE, SSIM, and Recall@K (Section 5.1) as well as human evaluation (Section 5.3).

2 RELATED WORKS

We review prior approaches that are directly related to our work.

2.1 Generative Adversarial Network

Generative Adversarial Network (GAN) (Goodfellow et al., 2014), as an unsupervised learning method, learns the mapping from a latent space Z to a target data distribution. If the target data distribution of interest is the image, this framework takes an arbitrary latent vector $z \in Z$ as input and maps the input to a point as an image on the target distribution. A typical GAN is composed of a generator G and a discriminator D. While D learns to distinguish whether the image y or $G(z)$ is real or not, G learns to generate a realistic image $G(z)$ that is hard to identify by D.

Generating a user-intended image, however, is a difficult problem, because we do not know how z is mapped to the image that the user intends to generate. For finding the appropriate z, we should exhaustively explore the vector space of z, then it becomes an expensive and impractical optimization problem. To control the image generation process for a user to generate an intended image, the conditional GAN is proposed (Mirza and Osindero, 2014), which takes an additional user input (e.g. a user image) along with a noize vector z. It learns a mapping from a pair of the random noise vector z and input image x to the target image y. The conditional GAN helps to make the generated image $G(x,z)$ to be correlated with the input image x.

Training the GAN model, however, is unstable, because the objective function of GAN is the minimax problem that D tries to minimize the loss, while G tries to maximize the loss of D simultaneously. To make GAN training stable, many variations have been proposed in terms of architectures (Radford et al., 2016), loss functions (Arjovsky et al., 2017; Gulrajani et al., 2017; Mao et al., 2017), and training algorithms (Salimans et al., 2016; Karras et al., 2017).

2.2 Image-Conditional Image Generation

In the field of image-conditional image generation, conditional GAN based approaches have been dominant. They show remarkable results for various applications: image inpainting (Pathak et al., 2016; Iizuka et al., 2017), interactive image editing (Brock et al., 2016), super-resolution imaging (Ledig et al., 2017), domain-transfer (Kim et al., 2017b), and image-to-image translation (Zhu et al., 2017; Isola et al., 2017).

(Isola et al., 2017) have proposed a general purpose image-conditional image generation model called pix2pix, which supports the relatively high resolution result images (256 × 256) and has become a widely-used model for this problem. (Yoo et al., 2016) have proposed a clothing image generation model, which generates clothing images at 64 × 64 resolution.

CycleGAN (Zhu et al., 2017) and DiscoGAN (Kim et al., 2017b) conduct image-conditional image generation with unpaired image datasets. CycleGAN supports up to 256 × 256 resolution images. It works well when changing the style, while keeping a shape of an object in an input, but it is difficult to change shape itself. DiscoGAN is relatively easy to change shape itself, unlike CycleGAN. However, it supports a relatively low resolution (64 × 64).

In this paper, we propose a clothing image generation model based on pix2pix. Our method is designed by adopting a coarse-to-fine strategy to cope with clothing image generation where a large-shape change is required.

2.3 Coarse-to-fine Strategy

Similar to ours, GAN approaches adopting the coarse-to-fine strategy to generate detailed images have been proposed. (Denton et al., 2015) have proposed a multi-stage image generation process consisting of several GANs. Each GAN conditioned on the previous GAN results generates a residual image; the result residual image is added to the previous GAN result to create the input of the next GAN. This iterative generation process can produce sharper images. (Mathieu et al., 2015) have proposed a multi-scale network to predict future video frames with the similar approach.

(Zhao et al., 2017) have shown image-conditional image generation from an input cloth image to a cloth image in a different-view via two-stage image generation process. At the first stage, they generate a coarse image by using a Variational Autoencoder (VAE), which is a generative model and relatively easy to train compared to GAN, but generates a blurry image. At the second stage, they generate a fine image through conditional GAN that has a pair of the coarse image from the first stage and the input image. Huang et al. (Huang et al., 2017b) have shown two-stage text-to-image generation with a sequence of GANs in a similar manner.

(Karras et al., 2017) have proposed GAN training method, called progressive growing, which is similar to supervised pre-training (Goodfellow et al., 2016; Bengio et al., 2007). This method progressively adds a block on the generator and discriminator to generate the target resolution image. Based on this concept, it can generate high-resolution face images from a noise vector. However, this approach produced images from a noise vector, so it was not directly designed for image-conditional constraints like our clothing image generation problem.

Except for (Karras et al., 2017), aforementioned studies (Denton et al., 2015; Mathieu et al., 2015; Zhao et al., 2017; Zhang et al., 2016) require a multi-network configuration using pairs of generators and discriminators for stages. As a result, it causes a large model size. (Karras et al., 2017) have implemented a coarse-to-fine approach with a single pair of a generator and a discriminator. It is, however, not designed for image-conditional constraints. So, it is unclear to apply the model to our target task without modification.

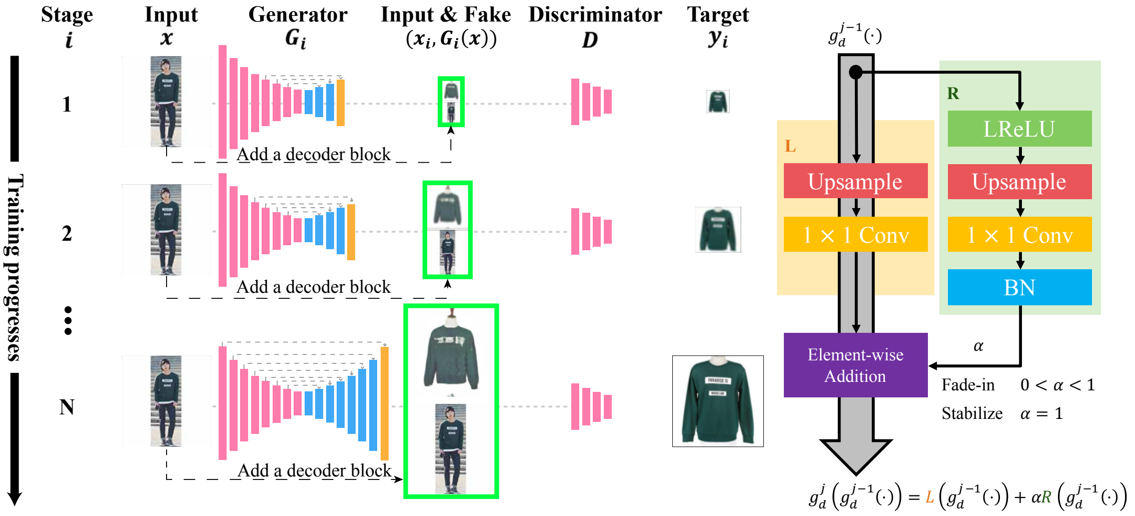
Instead of using multiple, separate pairs of generators and discriminators, our model progressively configures the network to be appropriate for each stage. Furthermore, we design our approach for respecting image-conditional constraints.

3 Rough-to-Detail GAN

We propose a new image-conditional image generation model, named rough-to-detail GAN (rtdGAN). The rtdGAN is a conditional GAN based image generation model that is trained in a coarse-to-fine manner, in order to solve the global consistency problem (Goodfellow, 2016). This problem causes inconsistent structures on generated images, especially in high resolution. In this section, we introduce the architecture of rough-to-detail GAN, objective function, and rough-to-detail training.

3.1 Architecture Design

Our model is based on a conditional GAN, which consists of a generator $G = \{G_E, G_D\}$ and a discriminator D. G consists of an encoder G_E and a decoder G_D, where $G_E = \{g_e^1, \dots, g_e^M\}$, $G_D = \{g_d^1, \dots, g_d^M\}$, g_e^j is



(a) (b)

Figure 2: (a) shows overview of rough-to-detail training. At the first stage, the generator G_1 works with a target image y_1, which has the coarse-structure of the image. As the training goes on, we incrementally add a decoder block g_d^{M-N+i} on the decoder part of G_i for generating larger images with finer details. (b) shows the structure of a decoder block. The flow of a previous result, $g_d^{j-1}(\cdot)$, is divided into two flows. The left-hand flow is used to resize and enhance previous results. The right-hand flow is to generate details of the image. a is a weighted term introduced for stably using the newly added decoder block. For the smooth fade-in, a increments from 0 to 1. After a fade-in, a is fixed to 1.

an j-th encoder block and g_d^j is an j-th decoder block, and M is the number of blocks in each encoder and decoder.

The encoder maps an input image x to a latent vector. Each encoder block g_e^j produces a down-sampled feature map, which contains higher level information as an output of its prior block g_e^{j-1}. We use a general stride-convolution block as the encoder block.

The decoder generates an image from the latent vector. Each decoder block g_d^j is designed to produce an up-sampled and refined result with the result of its prior block g_d^{j-1}. Therefore, a number of decoder blocks determines a size of an image generated by G. We use a modified version of the residual block (He et al., 2016) as the decoder block. The detailed information of the residual block is provided in Figure 2b. The entire structure of our G is similar to U-Net (Ronneberger et al., 2015), which can preserve contents of the input image x via skip-connection between g_e^j and g_d^j. The skip-connection is used for reducing the information loss caused by the bottleneck between the encoder and the decoder.

We implement a coarse-to-fine strategy through manipulating the structure of the decoder G_D. Our generator G can control the size of result image via adding decoder blocks. Given N stages of our rough-to-detail training, let G_i to be a generator G at the stage i. At stage $i = 1$, G_1 (the generator of the first row in Figure 2a) generates a small image, aiming to achieve the coarsest version of the target image via an asymmetric encoder-decoder structure, where G_E has M encoder blocks and G_D has only $M - N + 1$ decoder blocks. As we have more stages, we have additional decoder blocks on the generator. In the end, at stage $i = N$, G_N (the generator of the last row in Figure 2a) creates a larger image containing details of the target image via the symmetric encoder-decoder structure, where G_E and G_D have M blocks.

Note that we did not make our encoder structure to grow during the training process. If the encoder network grows, it also suggests that the input image should start with a very small input image, indicating that the information of the pixel area required for creating the clothes image in the input image can be lost compared to a bigger size input. Therefore, there is a possibility that the error created by this lost information might spread through the network as the stage progressed. To prevent this potential loss of pixel information, we freeze the encoder structure so that it can deal with as large images as possible.

We utilize the patch discriminator D, which determines whether the local patch of an image is real or not, while a general discriminator examines the entire image. This approach is more beneficial for describing high-frequency details (Isola et al., 2017; Li and Wand, 2016; Zhu et al., 2017). The detailed architecture of each network is summarized in the supplementary material.

3.2 Objective Function

Our objective function consists of three loss terms: Adversarial loss, Content loss, and Laplacian loss. The adversarial loss is used for generating realistic images and content loss has beneficial to force low-frequency correctness between the result image and the target image (Isola et al., 2017). The Laplacian loss is utilized to sharpen the result image.

The adversarial loss is used to generate an image indistinguishable with a real image. The loss is the same to the objective of the conditional GAN, which is expressed as:

$$\mathcal{L}_{adv}(G_i, D) = \mathbb{E}_{x_i, y_i}[D(x_i, y_i)] - \mathbb{E}_{x, x_i}[D(x_i, G_i(x))], \quad (1)$$

where y_i is a target real image for the stage i, x is the original input image, $G_i(x)$ is the fake image, x_i is a resized image of x whose size is same to y_i.

The content loss is used for generating a near ground-truth target image. The content loss is an L1 loss between a real image and a generated image, and is defined as follows:

$$\mathcal{L}_{con}(G_i) = \mathbb{E}_{x, y_i}[\| y_i - G_i(x) \|_1]. \quad (2)$$

We use a Laplacian loss to generate a sharper image. The Laplacian loss is an L1 loss between the Laplacian filtered real image and the generated image. Note that the Laplacian filtered image has been widely used for applications related to high-frequency information such as edge detection (Marr and Hildreth, 1980) and edge-preserving inpainting (Kim et al., 2017a). The Laplacian loss is defined as the following:

$$\mathcal{L}_{lap}(G_i) = \mathbb{E}_{x, y_i}[\| Lap(y_i) - Lap(G_i(x)) \|_1], \quad (3)$$

where $Lap(\cdot)$ is a Laplacian filtered image, which is approximated with the difference of Gaussians (DoG) (Szeliski, 2010) in our case.

Our final objective is then defined as follows:

$$\begin{aligned} G_i^* = \arg\min_{G_i}\max_{D} \lambda_{adv}\mathcal{L}_{adv}(G_i, D) + \\ \lambda_{con}\mathcal{L}_{con}(G_i) + \lambda_{lap}\mathcal{L}_{lap}(G_i), \end{aligned} \quad (4)$$

where λ_{adv}, λ_{con}, λ_{lap} are parameters that balance three loss terms.

3.3 Rough-to-detail Training

To realize our goal, we use rough-to-detail network training that performs a coarse-to-fine image generation through N stages. Through this training algorithm, our model gradually creates multiple scales of the target image from a coarse-scale to a fine-scale.

At a stage i, the model upsamples and refines the result of the previous stage $i-1$ to produce an intermediate target image y_i of the stage i. In this manner, the network learns the overall structure of the target image and then learns its details gradually. By repeating this process, our model finally generates the target image y_N. An overview of rough-to-detail is shown in Figure 2a. We first explain how to generate intermediate target images, followed by our learning process at each stage.

Intermediate target images. The goal of a stage i is to create representative structural characteristics at its chosen scale from the original target image y. To do this, we prepare an intermediate target image y_i for the stage i. For this purpose, we utilize the Gaussian image pyramid representation.

The Gaussian image pyramid representation is widely used in computer vision area, and is a useful tool for analyzing an image in various scales (Szeliski, 2010; Lindeberg, 1994). Normally, the image pyramid consists of various sizes of images, from the smallest coarse-level (the top of the pyramid) image to the largest fine-level image (the bottom of the pyramid).

Let the total pyramid level to be N and the Gaussian image pyramid representation $y_g = \{y_g^0, \ldots, y_g^{N-1}\}$ given the $H \times W$ original target image y. Each level of pyramid y_g^i is generated by a sequence of the Gaussian blur and down-sample on y_g^{i-1}. As a result, the top of the pyramid is the $\frac{H}{2^{N-1}} \times \frac{W}{2^{N-1}}$ smallest image y_g^{N-1}, which contains the coarsest structure of the input image, and the bottom of the pyramid is the $H \times W$ largest image y_g^0, which is the original image y.

Learning process at each stage. Because the size of a target image is different at every stage, we should setup G_i to generate a target image y_i for the stage i. As we mentioned in Section 3.1, the number of decoder blocks determines the size of an image generated by G_i. So, we setup G_i via adding a block g_d^{M-N+i} on the decoder G_D.

As shown in Figure 2a, in the first stage, our training starts with an asymmetric encoder-decoder network G_1 which consists of an encoder with M encoder blocks and a decoder with $M-N+1$ decoder blocks. In the last stage, our training works with the symmetric encoder-decoder network G_N, which consists of the encoder with M encoder blocks and an incrementally modified decoder with M decoder blocks. Our learning process for each stage progresses with a sequence of three steps: Preparation, Fade-in, and Stabilization.

• **Preparation** is the process of setting up the network to generate a target image y_i for the stage i. We set $(N-i)$-th level of the Gaussian pyramid representation y_g^{N-i} as the intermediate target image y_i. We add a residual block g_d^{M-N+i} to the decoder G_D of the generator G_i for increasing the resolution.

• **Fade-in and Stabilization** are introduced for stably updating network parameters. Fade-in is performed for avoiding a sudden shock caused by a newly added decoder block g_d^{M-N+i} on G_D of G_i. To avoid such a problem, we use a weighting term α to regulate the influence of the decoder block, which is added for generating details of a result image. α increments linearly from 0 to 1 per every epoch. After the fade-in, the network is further trained for stabilization. The detail of training algorithm is described in the Algorithm 1.

Algorithm 1: Rough-to-detail training

```
Let current training epoch $t$, current stage $i$
Build the Gaussian pyramid of the target image
  $y : \{y_g^n\}_{n=0}^{N-1}$
for $i$ in $[1, \ldots, N]$ do
    Set $y_g^{N-i}$ as an intermediate target image $y_i$
    Add a decoder block on the decoder $G_D$ in
      the generator $G_i$
    Initialize $\alpha$ as 0
    while iterations $t < t_{fade} + t_{stab}$ do
        Sample a minibatch of image $x$, $y_i$ from
          training data
        if $t < t_{fade}$ then
            Increment $\alpha$
        end
        Update the generator $G_i$ with the loss
          gradients of (Eq. 4)
        Update the discriminator $D$ with the
          loss gradients of (Eq. 1)
    end
end
```

4 EXPERIMENT SETTING

In this section, we explain various experiment settings used for validating the effectiveness of our proposed rtdGAN model. We conduct qualitative and quantitative evaluations on the LookBook dataset (Yoo et al., 2016). We compare the quality of result images with two other methods: pix2pix and PLDT. pix2pix is an image conditional image generation model proposed by (Isola et al., 2017), which is the base model for our rtdGAN. PLDT is a clothing image generation model proposed by (Yoo et al., 2016). We trained PLDT and pix2pix on the LookBook dataset using the source codes released by authors. We follow the training protocols described in their papers.

Dataset. LookBook (Yoo et al., 2016) is a dataset for the clothing image generation problem. It is made up of pairs of images of people dressed and clothing product images that they are wearing. LookBook includes a total of 9,732 top product images and 75,016 fashion model images; see Figure 3(a) and (f). Each product image is associated with eight fashion model images on average. For training, we resize all images to 256×256. We use ten percents of clothing images and its associated model images as the test split, and the remaining images are used as the train split. In the test split, we did data cleaning by removing redundant images that are in both splits. As a result, in the test split, the total number of clothing images are 939 and the total number of its fashion model images is 7,307.

4.1 Implementation details

A decoder block g_d has two ad-hoc blocks: ToRGB and Skip. The ToRGB block converts an intermediate generator result into an RGB image. We use this ad-hoc block for every stage except the last stage N, because the results of the generator in those stages are not RGB images. The ToRGB block consists of LeakyReLU (He et al., 2015) with 0.2 slope, 1×1 Convolution, and Tanh. Skip is for the channel reduction before the element-wise addition. Skip consists of upsampling by a factor of 2 and 1×1 convolution.

To generate result images (Figure 1), we use the total stage number N as 3 given the input resolution of 256×256; one can use more stages for higher resolutions. In fade-in, we train D and G_i for 40 epochs (t_{fade} in Algorithm 1). In stabilization, we train D and G_i for another 40 epochs (t_{stab} in Algorithm 1). All models are trained using the Adam optimizer (Kingma and Adam, 2015), where initial learning rate is 0.0002, momentum parameters β_1 is 0, and β_2 is 0.99. Mini-batch sizes of each stage are 60, 40, and 20 from the stage 1 to the stage 3, respectively. Also, target resolutions from the stage 1 to 3 are 64, 128, and 256.

We use the conditional version of the Wasserstein loss (Arjovsky et al., 2017; Gulrajani et al., 2017) as the adversarial loss. In our settings, the weight for gradient penalty is 10 and the number of critic is 1. All of balancing parameters (λ_{adv}, λ_{con}, λ_{lap}) in Equation 4 is 1.

Table 1: RMSE, SSIM, and Recall@60 results of our model with other conditional GAN methods of PLDT and Pix2Pix.

Method	RMSE	SSIM	Recall@60
PLDT (Yoo et al., 2016)	0.2921	0.4096	0.1787
Pix2Pix (Isola et al., 2017)	0.3009	0.5570	0.1873
Ours (3 stages)	**0.2590**	**0.5967**	**0.3373**

5 RESULTS

We use there different evaluation metrics to compare tested methods. We also conduct user study for evaluating human perception on different results.

RMSE and SSIM. We measure a quantitative performance via measuring the similarity between generated images and its target ground-truth product images. We use two well-known metrics: Root Mean Square Error (RMSE) and Structural Similarity (SSIM) (Wang et al., 2004). RMSE measures dis-similarity between two images, in a range from 0 (the same) to 1. SSIM measures a perceived quality of digital images in a range from 0 to 1, where bigger SSIM values mean higher similarity between two tested images. Before measuring SSIM, we convert RGB images to grayscale images, because SSIM supports only grayscale image.

Recall@K. If a generated image is similar to a target image, it should be easy to find the target image in image search when we use the generated image as a query. Assuming this property, we perform image search for evaluating our model. We use the query image generated from a fashion model image to find the corresponding ground-truth clothing image in the test split. For measuring the quality of image search, we use recall@k as metric. To perform image search, we extract image features via pre-trained densenet (Huang et al., 2017a).

5.1 Quantitative evaluation

A quantitative comparison is reported in Table 1. Our model with three stages achieves better RMSE, SSIM, and recall@60 results over the prior methods. Compared with PLDT, our model with three stages achieves 12.7% improvement in RMSE (from 0.2921 to 0.2590), 45.6% improvement in SSIM, and 88.7% improvement in recall@60. Based on these results, we can conclude that our model can generate more similar images to target clothing images than other models.

Examples of product image search are shown in Figure 4. In the second and third rows, the ground-truth target clothing images are located in the top-1 among retrieved results. This result is achieved by the high similarity between our generated images and their ground-truth images.

5.2 Qualitative evaluation

We also conduct qualitative comparisons between ours and other methods, which are shown in Figure 3. PLDT (e) tends to generate blurry images, because its target resolution is 64×64, while our model and Pix2Pix (d) can generate 256×256 resolution images. Pix2Pix (d) results do not have fine patterns nor colors contained in the target image, even if they are quite realistic.

We also test our method even with one stage, which adopts the symmetric encoder and decoders for the generator and thus does not contain our rough-to-detail training that is guided by our intermediate target images. Our method with a single stage (c) can generate clothing images with an appropriate pattern and colors based on an input image. However, all of these results contain blurry silhouette compared to input images.

On the other hand, our method with three stages (b) shows visually pleasing results, while producing global structures with fine details. Especially, in the second row of Figure 3, our result satisfies the color pattern that horizontally splits black and gray. Furthermore, our model can generate various type of clothing images. In the last row of Figure 3, ours can generate a skirt image, whereas two prior techniques (d) and (e) still generate sweater-like clothing images.

5.3 Human evaluation

Although RMSE, SSIM, and Recall@K measure similarity between generated images and target images, they cannot fully reflect the quality according to the human perception. To complement this limitation of the quantitative measures, we evaluate the quality of result images through human perception, as well.

We randomly select 70 model images that are associated with different product images in the test split. For each model image, three clothing images are generated by ours with three stages, Pix2Pix, and PLDT. All result images are evaluated in at its original size without any resizing. Given model images and their resulting images by different methods, 30 users are asked to perform two tasks related to realism and similarity aspects, as follows:

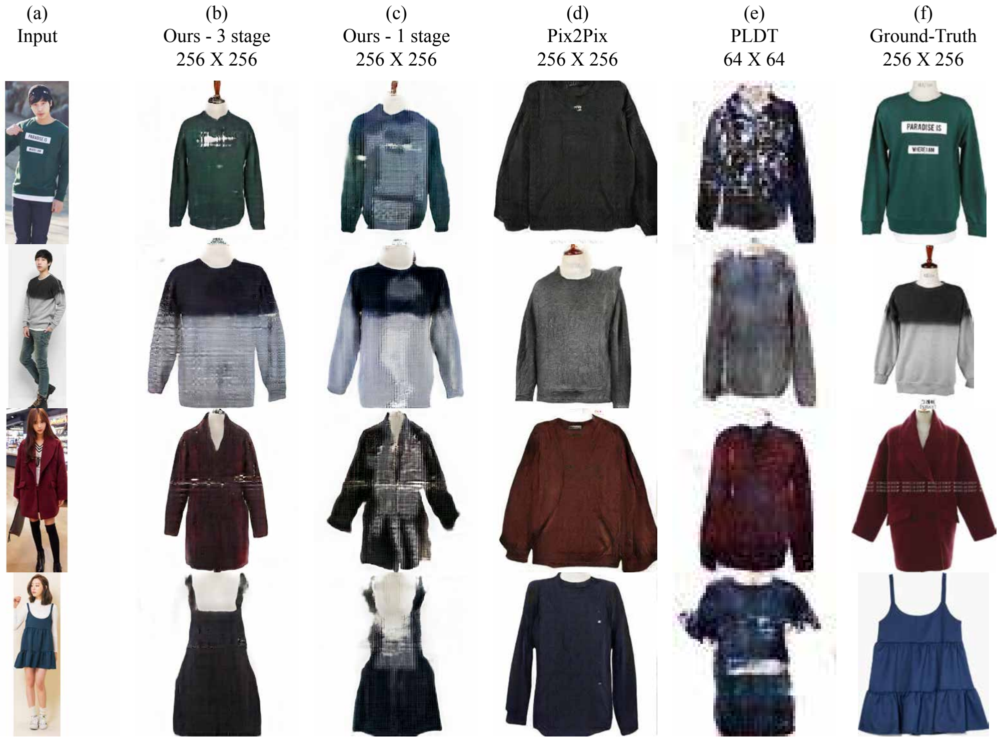


Figure 3: Examples of clothing image generation results. (a) Input fashion model images from the LookBook test split. (b) Results by our model with three stages of rough-to-detail training. (c) Results by our model with only a single stage of rough-to-detail training. (d) and (e) show results of other conditional GANs methods, Pix2Pix (Isola et al., 2017) and PLDT (Yoo et al., 2016), respectively. (f) Ground truth target clothing images. Our results with three stages show well-constructed structures with fine-details.

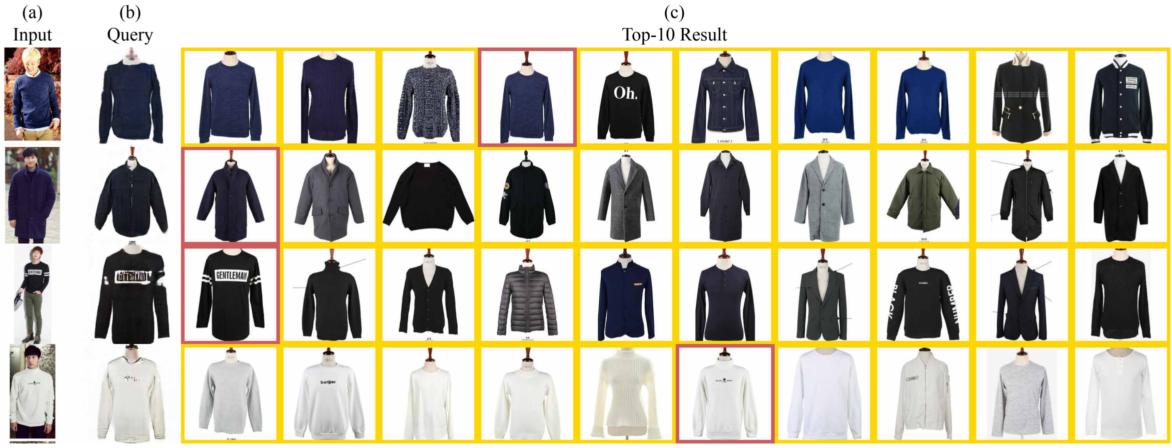
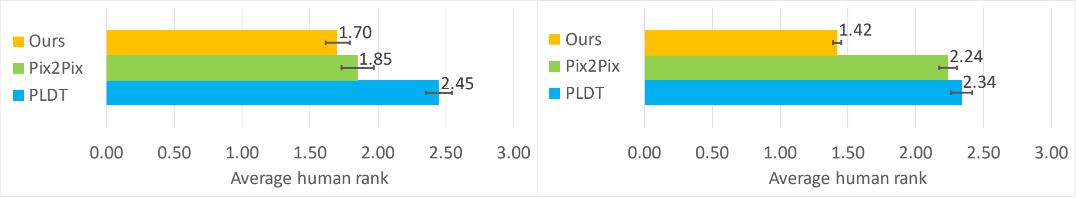


Figure 4: Examples of clothing product search results. (a) Input fashion model images from the LookBook test split. (b) Generated clothing images by our model. (c) Top-10 image search result. Results in the red box indicate the ground-truth clothing images.

1. Realism: Rank result images in the order that they look like real clothing images.
2. Similarity: Rank result images in the order that they reflect details from input model images.

To compare results of different methods, we calculate the average human rank computed by ranks given by users. Figure 5 shows the 95% confidence interval of the average human rank in each task. Our model achieves the best average human rank on



(a) Realism rank

(b) Similarity rank

Figure 5: Average human rank about image quality of ours and other conditional GAN methods, Pix2Pix (Isola et al., 2017) and PLDT (Yoo et al., 2016), with 95% confidence intervals.

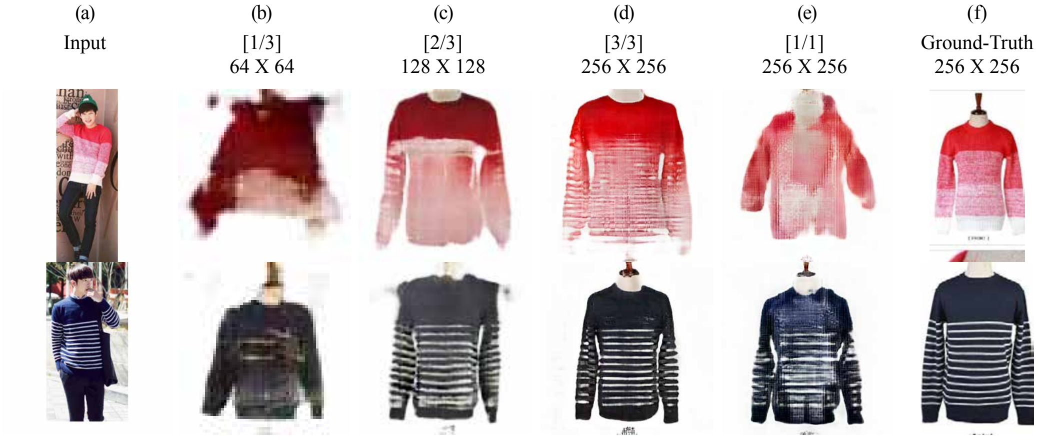


Figure 6: The progress of image generation in our model. [*i*/*N*] means results after *i*-th stage finished, when the total number of stage in rough-to-detail training is *N*.

"Realism", indicating that users thought that our results are more realistic compared to other methods. Moreover, Figure 5b also shows that our model also achieves the best average human rank on "Similarity", suggesting that our model can generate clothing images that have details from input model images, compared to other methods.

Table 2: RMSE, SSIM, and Recall@60 results of our model with different N, where N is the total number of stages in rough-to-detail training.

Method	RMSE	SSIM	Recall@60
Ours ($N = 1$)	0.2660	0.5256	0.2738
Ours ($N = 2$)	0.2594	0.5782	0.2869
Ours ($N = 3$)	**0.2590**	**0.5939**	**0.3373**

6 ABLATION STUDY

To investigate the effectiveness of the rough-to-detail training, we conduct an ablation study. To see how our rough-to-detail training improves the quality of images, we train our model with various values for N up to 3 stages given our target image resolution.

Generated images with different numbers of stages are shown in Figure 6. Our model with $N = 3$ progressively generates higher quality images. After the first stage, the model generates blurry clothing images (Figure 6(b)). After the second stage, it starts to reflect the pattern and shape of the target (Figure 6(c)). Finally, the result becomes more clear and sharp (Figure 6(d)). Our model only with the single stage training produces structurally collapsed results, exhibiting the global structure problem, compared to three stage training results. This is mainly because using only the single stage does not adopt the rough-to-detail training and thus it is difficult to generate the global structure with fine details.

We examine quality improvement quantitatively as well. Table 2 shows that RMSE, SSIM, and recall@60 results of our model according to N. We find a tendency that the more stages the model goes through, the higher performance is achieved. This demonstrates the benefits of our rough-to-detail training, where a lower stage captures higher level structures, while a higher stage depicts finer details. Compared to our model with $N = 1$, our model with $N = 3$ achieves 2.7% improvement in RMSE, 12.9% improvement in SSIM, and 23.1% improvement in recall@60.

7 CONCLUSION

In this paper, we have proposed rough-to-detail conditional GAN (rtdGAN) for the clothing image generation problem. Image generation with high-resolution (e.g., 256 × 256 resolution) has been regarded as a difficult task. To solve the problem, we have split the difficult single stage image generation process into a relatively easy multi-stages image generation process. We have applied the coarse-to-fine strategy on the image-conditional image generation model and proposed a new training method called rough-to-detail training. We have also designed a generator network that is suitable for the proposed training method. The generator in our model is progressively configured for an intermediate target image at each stage by adding a decoder block. Via this process, our model can generate from small size images with global structures to large size images with details. To validate our proposed model, we have conducted extensive evaluations on the LookBook dataset. Compared to other conditional GAN models, our model can generate visually pleasing 256 × 256 clothing images while keeping global structures and containing details of target images.

Limitations and Future Work. We have shown the effectiveness of the coarse-to-fine strategy for the image-conditional image generation model. Through this multi-stage process, our model can achieve the quality improvement. Nonetheless, choosing the optimal number of total stages N rigorously is left for future study. In the current work, we simply choose the value of N depending on the resolution of target images, but the more thorough analysis is required for handling a wider range of image resolutions. In addition, we also plan to test our model for different applications, to broadly investigate the effectiveness of our model on image-conditional image generation, not only on clothing image generation.

ACKNOWLEDGEMENTS

We are thankful to the anonymous reviewers for their comments. This work is supported by SW StarLab program (IITP-2015-0-00199), and NRF (NRF2017M3C4A7066317). Prof. Yoon is a corresponding author of the paper.

REFERENCES

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations (ICLR)*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan. In *International Conference on Machine Learning (ICML)*.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160.

Brock, A., Lim, T., Ritchie, J. M., and Weston, N. (2016). Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*.

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

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017a). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708.

Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., and Belongie, S. (2017b). Stacked generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5077–5086.

Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*.

Kim, S., Kim, T., Kim, M. H., and Yoon, S.-E. (2017a). Image completion with intrinsic reflectance guidance. In *Proc. British Machine Vision Conference (BMVC 2017)*.

Kim, T., Cha, M., Kim, H., Lee, J. K., and Kim, J. (2017b). Learning to discover cross-domain relations with generative adversarial networks. In *ICML*.

Kingma, D. and Adam, J. B. (2015). Adam: A method for stochastic optimization. *In International Conference on Learning Representations (ICLR)*.

Lassner, C., Pons-Moll, G., and Gehler, P. V. (2017). A generative model of people in clothing. In *The IEEE International Conference on Computer Vision (ICCV)*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A. P., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4681–4690.

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

Lindeberg, T. (1994). Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. (2017). Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proc. R. Soc. Lond. B*, 207(1167):187–217.

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

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *In International Conference on Learning Representations (ICLR)*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer.

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

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. (2018). Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*.

Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.

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

Yoo, D., Kim, N., Park, S., Paek, A. S., and Kweon, I. S. (2016). Pixel-level domain transfer. In *European Conference on Computer Vision*, pages 517–532. Springer.

Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., and Metaxas, D. (2016). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242*.

Zhao, B., Wu, X., Cheng, Z.-Q., Liu, H., Jie, Z., and Feng, J. (2017). Multi-view image generation from a single-view. *arXiv preprint arXiv:1704.04886*.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*.