LANGUAGE DRIVEN OUTFIT STYLE ENHANCEMENT

Snehal Reddy Koukuntla, Deepank Agrawal, Kanishk Singh, Shrey Srivatsava

Indian Institute of Technology, Kharagpur

ABSTRACT

We attempt to present an approach for generating stylised clothing on a subject image through generative adversarial learning. Given an input image of a person with a particular base outfit and textual description of a style, our model "fashionises" the person as desired, without altering the subject's base outfit and background. Generating new outfits with precise regions conforming to a language description while retaining wearer's body structure is a challenging task.

1. INTRODUCTION

A big challenge in the fashion and graphics industry is for any person to stylise their own clothes according to their wishes without actually using existing complex software tools to design clothes and render them before actual production. The texture and pattern of any clothing item/accessory is an indispensable and crucial part in deciding the aesthetics of the fashion item. Every person has very subjective view on what seems attractive, but not has the expertise to graphically visualise how the items will turn out with the help of those complex software tools. It is not feasible to produce the fashion items without rendering them. A natural way for a lay man to describe his idea is through words and hence the best way to take input is natural language in form of textual descriptions.

We aim to impose the texture as described by the user on a given fashion item. To formally present the problem statement we seek to solve, given an input image of a fashion item I, and a textual description consisting of a set of key-words $\{w_i \mid w_i \in [1, n]\}$, we aim to generate transformed image T, with the texture corresponding to the textual description imposed on top of the input image I.

We divide the problem statement into three major parts which ease task formulation. The first task is to generate the texture from the textual description input from the user. This step is followed by segmentation of targeted cloth item from input image. Then the generated texture is super-imposed on the segmented region.

Converting text to image is an important problem and has a lot of applications, including photo-editing, computer-aided design, etc. Recently, Generative Adversarial Networks (GAN) have shown credible results in generating plausible real-world images. The main difficulty in generating textures by GANs is that supports of natural image distribution and implied model distribution may not overlap in high dimensional pixel space. StackGANs have produced amazing results in this field. They propose a Stacked Generative Adversarial Networks for synthesizing photo-realistic images from text descriptions. It decomposes the difficult problem of generating high-resolution images into more manageable sub-problems and significantly improves the state of the art. StackGAN for the first time generates images of 256*256 resolution with photo-realistic details from text descriptions. A new Conditioning Augmentation technique is proposed to stabilize the Conditional GAN training and also improves the diversity of the generated samples. Since the textual description from the user is very general and has no particular genre, we aim to use COCO dataset for the purpose of training StackGAN.

Over the past year, the vision community has produces many methods which are conceptually intuitive and offer flexibility and robustness, together with fast training and inference time. Many segmentation networks have shown promising results in the past. In particular Mask-RCNN is a very simple, flexible, and general framework. A lot of datasets pertaining specifically to fashion items have been made public.

This process of filling texture is difficult for a deep network to learn for several reasons:

- It is undesirable to leave an object partially textured or to have the texture spill into the background.
- The network should additionally learn to foreshorten textures as they wrap around 3d object shapes, to shade textures according to ambient occlusion and lighting direction.
- Existing deep networks aren't particularly good at synthesizing high resolution texture details even without user constraints. Typical results from recent deep image synthesis methods are at low resolution (e.g. 64x64) where texture is not prominent or they are higher resolution but relatively flat.

Recent research investigating deep image synthesis guided by sketch, color, and texture have led to image synthesis methods that can mitigate such challenges. TextureGAN Xian et al. [2017] is such a method that shows promising results for texture augmentation.

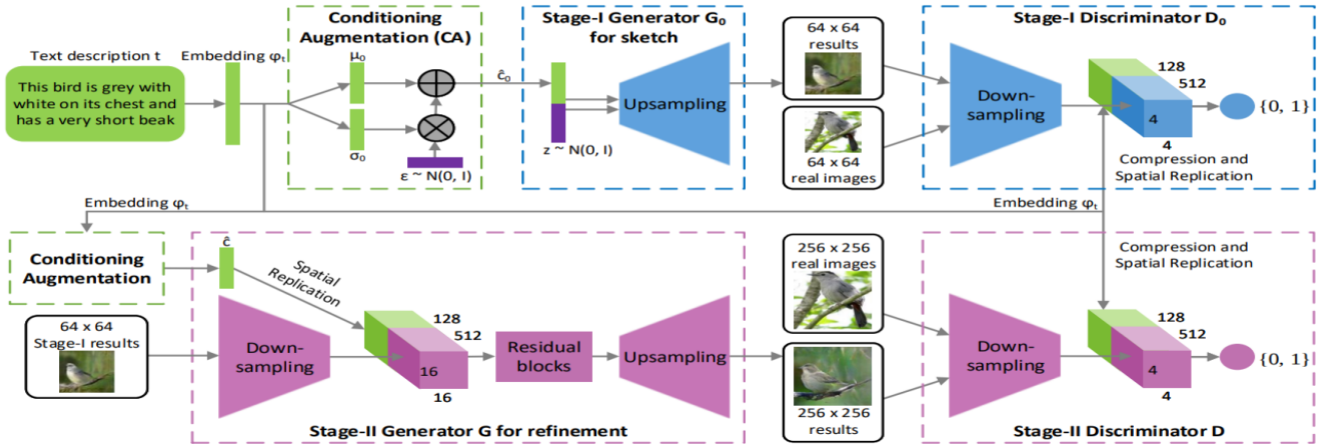

Fig. 1. Architecture of StackGAN.

2. PREVIOUS WORK

There has been a lot of related work in this field. Generating plausible visual representation of thoughts has always fascinated humans. But synthesis of realistic images is a intriguing and challenging task in fields like graphics, vision, animation industries and, for our concern, machine learning research. All the existing approaches can be grouped into parametric and non-parametric paradigm.

Non-parametric approaches excel at synthesis of visually appealing images but are often limited due to restricted data availability. On the other hand, parametric approaches are gaining wide popularity and acceptance in recent times. Generative Adversarial Networks(GANs) are a type of parametric method. It has been widely applied and experimented for visual generation recently. GANs pits two networks against each other—a generator and a discriminator. The generator is trained to confuse the discriminator. While, discriminator tries to distinguish them from realistic visuals. GAN-based methods generate more plausible images with intensive detailing and higher resolution.

The generator G is optimized to reproduce the true data distribution p_{data} by generating images that are difficult for the discriminator D to differentiate from real images. Meanwhile, D is optimized to distinguish real images and synthetic images generated by G. Overall, the training procedure is similar to a two-player min-max game with the following objective function,

$$\min_G \max_D V(D,G) :=$$

$$E_{x\sim p_{data}}[\log(D(x))] + E_{z\sim p_z}[\log(1 - D(G(z)))$$

where x is a real image from the true data distribution p_{data}, and z is a noise vector sampled from distribution p_z which is a Gaussian.

In our task, the practical visual synthesis software requires human interpretable controls, i.e., a textual description of the task. But, conventional GAN is not controllable or modifiable by external human interference after training process. On the other hand, Conditional GANs are models that generate visuals based on human control variables. Conditional GANs, in addition to the conventional losses, introduce extra losses and models to ensure controlled generator results.

Here, we are presenting two related approaches for controlled visual generation task. We used StackGAN for the generation of high resolution photo-realistic images from text descriptions. The method proposed the solution by solving two subproblems in which a low resolution image is first generated using GAN-I and then GAN-II is trained while being conditioned on generated low resolution image. The Stage-I GAN learns to draw rough shape and basic colors of the generated object conditioned on the given text description and generate background regions from a random noise vector sampled from a prior distribution. By conditioning on the text again, Stage-II GAN learns to capture text information that are omitted by Stage-I GAN and draws more details for the object.

TextureGAN takes as input a sketched object, a user drag one or more example textures onto the sketched object. The TextureGAN realistically generate the texture on the target objects. This approach is the first deep synthesis method which allows controlled generation of texture on objects. The texture fill step is challenging to learn for a neural network for various reasons, like—neural networks often fail poorly at generating high-resolution texture details, the filling operation should be constrained within the object boundaries—this demands for the sketched object to be segmented out of the background.

3. PROPOSED APPROACH

Figure 2 shows an overview of the proposed style enhancement algorithm. Our algorithm is divided into three submodules: Textual Description to Texture, Image Segmentation and Texture Augmentation. We first use the Textual Description to Texture module to generate a texture patch based on the textual description as described in Section 3.1. Thereafter, the Image Segmentation module is used to segment out the clothing region in the image. Section 3.2 describes the generation of the image with only edges corresponding to clothing region. Section 3.3 explains our Texture Augmentation module which inputs the texture patch generated by the first module and edge extracted image from the second module to generate an image with texture patch enlarged on the clothing region.

3.1. Textual Description to Texture

We employ StackGAN Zhang et al. [2016] for the task of transducing textual descriptions to texture images. StackGAN is capable of producing high resolution photo-realistic images using its two stage GAN architecture. However, the presented methodology requires only a small texture patch to be scaled-up, so generating a high resolution image as output and then scaling it down would only lead to waste of computation.

But StackGAN is highly accurate in its generated image thus to savour the benefits of StackGAN and still be computationally efficient we use the pre-trained parameters corresponding to two stage complete architecture but only deploy the Stage-1 GAN in our algorithm. The image output from the first stage is a low resolution image consisting of rough shape and basic colors conditioned on the given textual description.

The conditioning text description t is first encoded by an encoder, yielding a text embedding φ_t.Then latent variables are randomly sampled from an independent Gaussian distribution $\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$, where the mean $\mu(\varphi_t)$ and diagonal co-variance matrix $\Sigma(\varphi_t)$ are functions of the text embedding φ_t.

To further enforce the smoothness over the conditioning manifold and avoid overfitting, the following regularization term is added to the objective of the generator during training, $D_{KL}((\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t)))||N(0, I))$, which is the Kullback-Leibler divergence between the standard Gaussian distribution and the conditioning Gaussian distribution.

Conditioned on Gaussian latent variables c_0, Stage-I GAN trains the discriminator D_0 and the generator G_0 by alternatively maximizing L_{D_0} and minimizing L_{G_0} by,

$$\mathcal{L}_{D_0} = E_{(I_0,t)\sim p_{data}}[\log(D_0(I_0, \varphi_t))] + E_{z\sim p_z,\, t\sim p_{data}}[\log(1 - D_0(G_0(z, c_0), \varphi_t)]$$

$$\mathcal{L}_{G_0} = E_{z\sim p_z,\, t\sim p_{data}}[\log(1 - D_0(G_0(z, c_0), \varphi_t)] + \lambda D_{KL}((\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t)))||N(0, I))$$

where the real image I_0 and the text description t are from the true data distribution p_{data}. z is a noise vector randomly sampled from a given distribution p_z (e.g., Gaussian distribution used in this paper). is a regularization parameter that controls the balance between the two terms.

3.2. Image Segmentation

We employ Mask R-CNN He et al. [2017] for the task of image segmentation. Mask RCNN is a deep neural network aimed to solve instance segmentation problem. There are two stages of Mask RCNN. First, it generates proposals about the regions where there might be an object based on the input image. Second, it predicts the class of the object, refines the bounding box and generates a mask in pixel level of the object based on the first stage proposal.

We initialise Mask R-CNN with ResNet-101 pretrained for segmentation on MS COCO Lin et al. [2014]. DeepFashion2 Ge et al. [2019] is a comprehensive fashion dataset. It contains 491K diverse images of 13 popular clothing categories from both commercial shopping stores and consumers. It totally has 801K clothing clothing items, where each item in an image is labeled with scale, occlusion, zoom-in, viewpoint, category, style, bounding box, dense landmarks and per-pixel mask.There are also 873K Commercial-Consumer clothes pairs. We use this dataset to further finetune the Mask RCNN.

After the generation of segmented image by Mask RCNN, the segmented regions corresponding to classes other than clothes are removed from the input image. The remaining image is then passed through Canny edge detectors which outputs a gray scale image with clothing boundary. The texture patch generated using the first module is affixed on the centre of clothing region in this image which is then passed to the Texture Augmentation module.

3.3. Texture Augmentation

We seek an image synthesis pipeline that can generate natural images based on an input sketch and a text defined texture patch. We employ TextureGAN Xian et al. [2017] for the purpose of scaling up the texture patch to the entire image. A major challenge for a network learning this task is the uncertain pixel correspondence between the input texture and the unconstrained sketch regions. To encourage the network to produce realistic textures, a local texture loss based on a texture discriminator and a style loss is proposed. This not only helps the generated texture follow the input faithfully, but also helps the network learn to propagate the texture patch and synthesize new texture retaining wearer's body structure.

We used pre-trained models of TextureGAN present at https://github.com/janesjanes/Pytorch-TextureGAN.

Fig. 2. Input image

Fig. 3. The StackGAN output with input "Purple Flower".

Fig. 4. The final output with input "Purple Flower".

4. RESULTS

We have presented an approach for language guided style enhancement with input textual description and target clothing image. With this system, a user can textually describe and precisely control the details of generated stylised clothing. Our results show that the pipeline is able to handle a wide variety of textual inputs and generate texture compositions that follow the sketched contours satisfactorily. The proposed methodology is not end-to-end trainable and is done on purpose. To verify the feasibility of the defined problem statement, authors used pre-existing models. After critical analysis of related research works, it can be concluded that for an end-to-end model, extra losses and degenerator modelling is required in vanilla C-GAN to achieve the desired results.

References

Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. *CoRR*, abs/1901.07973, 2019. URL http://arxiv.org/abs/1901.07973.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Wenqi Xian, Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Texturegan: Controlling deep image synthesis with texture patches. *CoRR*, abs/1706.02823, 2017. URL http://arxiv.org/abs/1706.02823.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1612.03242, 2016. URL http://arxiv.org/abs/1612.03242.