AIGS 538 Homework Assignment 5 20222421 심재윤 (JAEYOON SIM)

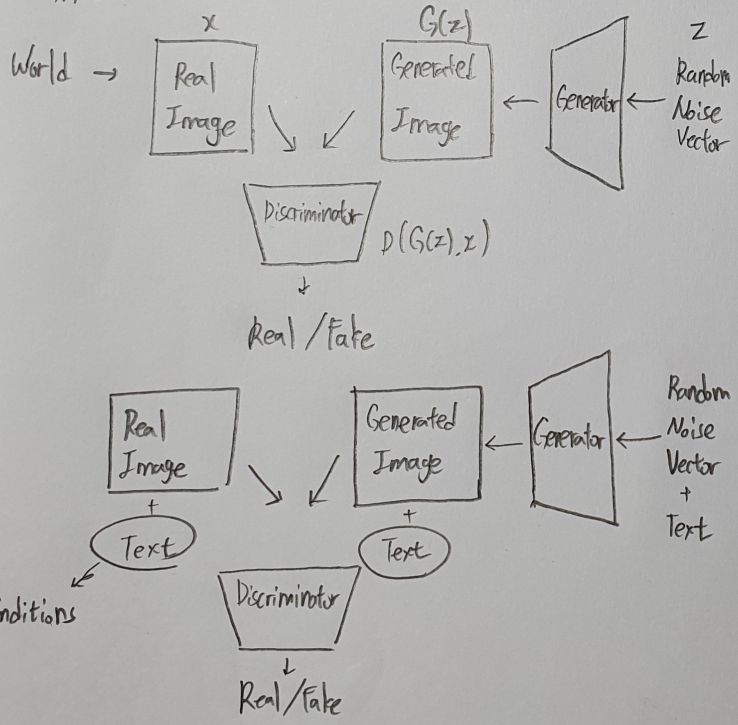
#1. Generative Adversarial Text to Image Synthesis (2016)

- Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak lee

[1] Introduction & Related Works

Automatic synthesis of realistic images from text would be interesting and useful, but current AI systems are still far from this goal. However, in recent years generic and powerful recurrent neural network architectures have been developed to learn discriminative text feature representations. Meanwhile, deep convolutional generative adversarial networks (GANs) have begun to generate highly compelling images of specific categories, such as faces, album covers, and room interiors. In this paper, they develop a novel deep architecture and GAN formulation to effectively bridge these advances in text and image modeling, translating visual concepts from characters to pixels. They demonstrate the capability of their model to generate plausible images of birds and flowers from detailed text descriptions.

- GAN



Based GAN model architecture is illustrated on the left. GAN has also benefited from convolutional decoder-networks. Also, there is conditional GAN which has one more input (conditions) for both generator and discriminator. And the authors has used CGAN as a based model. The distinction of their work from conditional GAN is that their model conditions on text description instead of class labels. The bottom illustrates GAN with joint distributions. The generator generates plausible image based on the conditions text - explaining about the image which generator will generate and both this image and text are inputed to the discriminator.

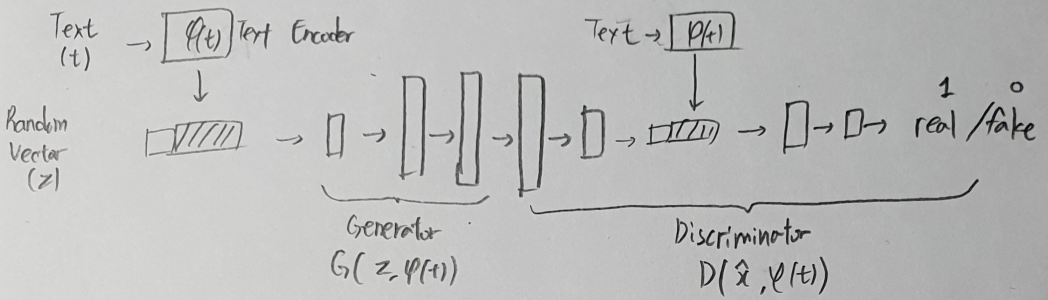
- Text Embedding

Then how is the text inputted to generator and discriminator? In order to represent the text in a vector which can be an input for both generator and discriminator, text embedding is used and for text embedding recurrent text encoder is used.

2022242| 심재윤 (JAEYOON SIM)

2 Architecture & Architecture Comparison.

The overall architecture of the proposed networks are similar to the illustration on page 1, GAN with join distribution. Addition to this architecture there are text embedding networks. The below illustration shows the overall architecture of the proposed model method.



Discriminator only outputs 1 (real) if the generated image looks like real and text and image matches. Text is embedded through text encoder and is compressed through fully connected layer with leaky ReLU and then concatenated with the random noise vector z.

- GAN-CLS (matching-aware discriminator)

If the model use the general GAN discriminator, it will be hard for the model to find out whether the generated image matches the input text. This is because general discriminator only focus on whether the generated image looks real or not. Therefore, in this paper, authors introduced "GAN-CLS" which is aware of the matching.

$$L_D \leftarrow \log(S_r) + [\underbrace{\log(1-S_w)}_{new} + \log(1-S_f)]/2 \quad \begin{bmatrix} S_r \leftarrow D(x,h) \\ S_w \leftarrow D(x,\hat{h}) \\ S_f \leftarrow D(\hat{x},h) \end{bmatrix}$$

where x is real image, h is true text, $\hat{h}$ is fake text, $\hat{x}$ is fake image. With this objective function, discriminator will be train to give "fake" to both S_w (real image & fake text) and S_f (fake image and real text).

- GAN-INT (learning with manifold interpolation)

Deep networks have been shown to learn representations in which interpolations between embedding pairs tend to be near the data manifold. Authors were motivated by this property and generated a large amount of additional text embedding by simply interpolating between embeddings of training set captions. And this interpolation is not done by human so it is free of labeling cost. The objective cost which should minimize is as follows.

$$\mathbb{E}_{t_1,t_2 \sim p_{data}}\left[\log(1-D(G(z,\beta t_1+(1-\beta)t_2)))\right]$$

They train interpolating between different text embedding vector (t_1, t_2). So the generator learns to fill gaps on the data manifold.

- Inverting the generator for style transfer

If the text embedding $\varphi(t)$ captures the image content like flower shape and colors, then in order to generate a realistic image the noise sample z should capture style factors such as background color and pose. To transfer the style of query image onto the content of a particular text description, one can train CNN to invert G to regress from samples $\hat{x} \leftarrow G(z, \varphi(t))$ back onto z. In this paper, they used a simple squared loss to train the style encoder.

$$L_{style} = \mathbb{E}_{t, z \sim \mathcal{N}(0,1)} \| z - S(G(z, \varphi(t))) \|_2^2$$

where S is the style encoder network. With the trained generator G and S, style transfer from query image x onto text t proceeds as follows.

$$[s \leftarrow S(x), \quad \hat{x} \leftarrow G(s, \varphi(t))$$

[3] Training Details.

In this paper, experiments were done with various datasets like Oxford-102 (flower images), CUB dataset (bird image) and MSCOCO. The writer used the same GAN architecture for all datasets. The training image size was set to 64x64x3. The text encoder produced 1024-dimensional embeddings that were projected to 128 dimensions in both the generator and discriminator before depth concatenation into convolutional feature map. They used alternating steps of updating the generator and discriminator network. They used the same base learning rate of 0.0002 and used the ADAM solver with momentum 0.5. The generator noise was sampled from a 100-dimensional unit normal distribution. They used a minibatch size of 64 and trained for 600 epochs.

[4] Performance Comparison.

On CUB datasets, both GAN and GAN-CLS got some color right but results didn't look real. On the other hand, GAN-INT and GAN-CLS-INT show plausible images that usually matched all or at least of the caption. On the Oxford-102 datasets all the results from diverse GAN model was better than that of CUB datasets. This might be because birds have strong structual regulation across species that make it easier for D to spot fake bird than to spot a fake flower.

2022242 심재윤 (JAEYOON SIM)

#2. Image-to-Image Translation with Conditional Adversarial Networks (2017)

- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros

[1] Introduction & Related Works.

Image to image translation is one application of conditional GAN (CGAN). CGAN is very similar to GAN but get one more input (condition) in both generator and discriminator. In use of CGAN in image to image translation, this 'condition' is an image and this image can be architecture label, map and others which is suited for particular tasks.

- Structured losses for image modeling.

Image to image translation are often formulated as per-pixel classification or regression and these formulations outputs "unstructured" output space. Unstructured output space pixels are independent from all the other pixels given the input image. However, in this paper, the CGAN is used which learns structured loss. This structured loss penalize the joint configuration of the output.

- CGAN

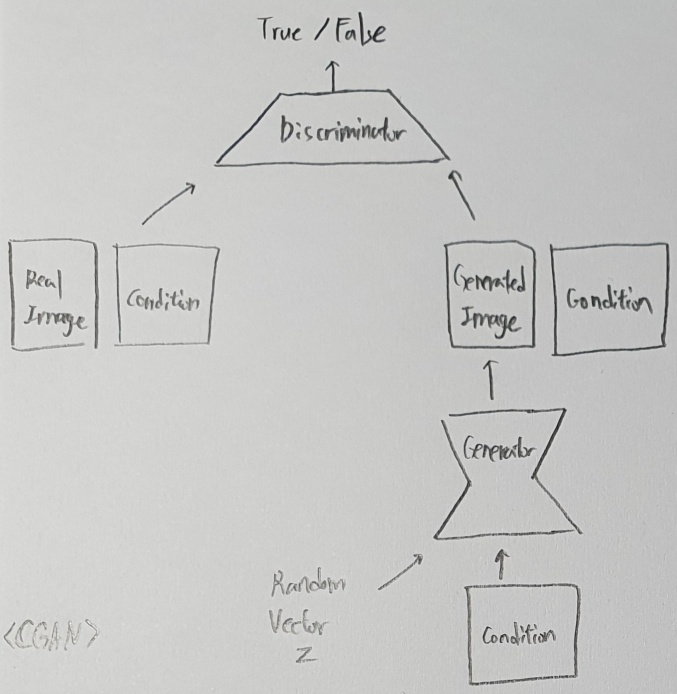
Before this paper, CGAN was already used to have image condition with discrete label condition, and text condition. However, the paper used CGAN for image to image translation for the first time. The novel framework of this paper can be used in many task and is not application specific. The novelty lies on choice of generator they used U-Net based generator and choice of the discriminated they used Patch GAN classifier.

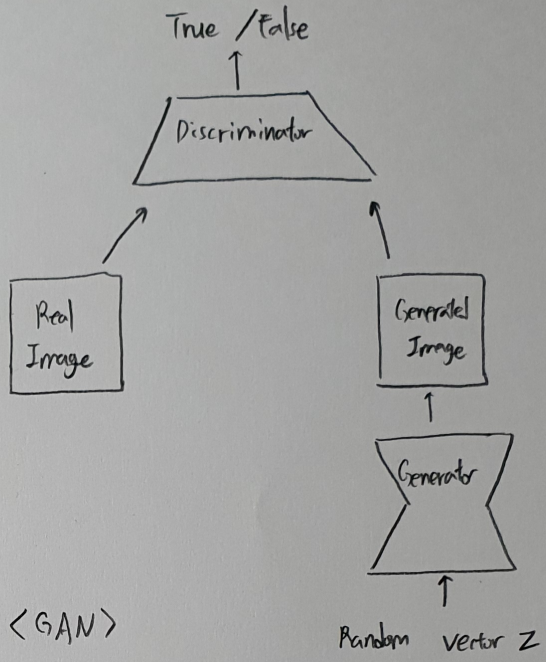
[2] Architecture & Architecture Comparison.

- CGAN

Before explaining about the CGAN which is used for the proposed network, lets first compare GAN and CGAN. GAN has generator which gets random vector as an input and outputs image and discriminator which get the generated image and real image as an input and output whether or not the generated image is true or false.

However, condition GAN has one more input for both generator and discriminator. Belows are the illustration of GAN and CGAN.

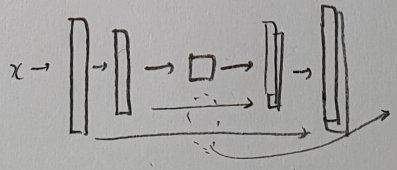




In this paper, the authors used CGAN for generating image from image.

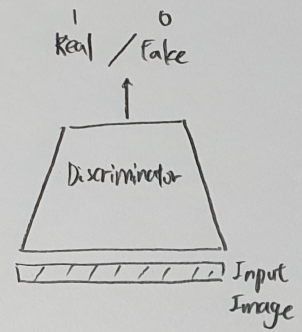
- Generator - UNet

In the CGAN generator, they used UNet which has skip connections on encoder-decoder model. This skip connections give generator a means to circumvent the bottle-neck for low-level information shared between input and output. like for image colorization or location of prominent edges.



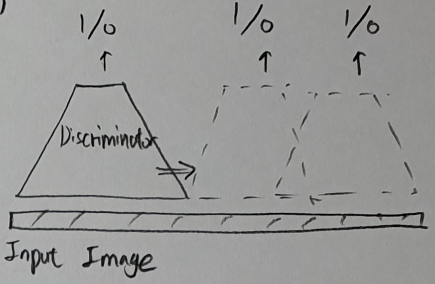
- Discriminator - Patch GAN

In the objective function of the proposed model, there is L1-norm term. This L1-norm term fails to encourage high-frequency crispness, it accurately capture the low frequencies. Therefore, they used Patch GAN to catch the high-frequency crispness. The discriminator classifies if each NxN patch which is smaller than the original input image for the discriminator. in the image is real or fake. This Patch GAN still produce high frequency results even though N are much smaller than the original image. This discriminator has fewer parameters, runs faster, and can be applied to arbitrarily large image.



The original discriminator (Image GAN) covers whole image.

The Patch GAN discriminator covers one patch at a time.



- Optimization Loss

The optimization loss, objective, for the proposed model are combination of two objectives. One is objective of CGAN (L_{cGAN}) and another one is traditional loss, L1 loss (L_1).

· $L_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1-D(x,G(x,z))]$; where G tries to minimize this loss against an adversarial D that tries to maximize it.

· $L_{L_1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]$; the discriminator's job remains unchanged, but the generator is tasked to not only fool the discriminator but also to be near the ground truth output in an L2 sense. They also explore this option, using L1 distance rather than L2 as L1 encourages less blurring. Their final objective is $G^* = \arg\min_G \max_D L_{cGAN}(G,D) + \lambda L_{L_1}(G)$.

3 Training Details

They tested the method on variety set of tasks and datasets including both graphics tasks, like photo generation, and which tasks like semantic segmentation to explore the generality of CGAN. In all cases, the input and the output are simply 1-3 channel images.

4 Performance Comparison.

The evaluation was done by comparing ground truth, output of L1, CGAN and output of L1 + CGAN. They used AMT perceptual studies and FCN score. Comparison with FCN score, L1+CGAN had the higest perpixel accuracy, per-class accuracy and class IoU between L1, GAN, CGAN, L1+GAN, L1+CGAN. And L1+GAN had the second highest per-pixel accuracy whereas CGAN had second highest per-class accuray and class IoU. In semantic segmentation, using L1 loss resulted to have highest accuracy and L1 + CGAN follows.

2022242| 심재윤 (JAEYOON SIM)

#3. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (2017)

- Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros.

[1] Introduction & Related Work.

Before this paper was published there was Image-to-Image translation with conditional adversarial networks. and the proposed model by this paper did similar task with this paper. However, they needed image for training. To accumulate these paired images are quite costy. So this paper introduced a model that does not need paired image for training. = GANs.

GANs have achieved impressive results in image generation, image editing, and representation learning. The key to this success is the idea of an adversarial loss that forces the generated image to be, in principle, indistinguishable from real photos. The writers adopted an adversarial loss to learn the mapping such that the translated images cannot be distingushed from images in the target domain.

- Image-to-Image Translation.

The idea of image to image translation goes back to employment of non-parametric texture model on a single input-output training image pair. The proposed model had build on the pix2pix framework. The difference of the proposed model to prior work, is that this model learn the mapping without paired training examples.

- Unpaired Image-to-Image Translation.

There were several other methods which also tackle the unpaired setting, where the goal is to relate two data domains: X and Y. For example, CoGAN and cross-modal seene networks use a weight-sharing strategy to learn a common representation across domains. Concurrent to the proposed model encourages the input and output to share specific content features even though they differ in style. These methods also use adversarial networks, which additional terms to enforce the output to be close to the input in a predefined metric space, such as class label space image pixel space, and image feature space. However,

the proposed framework does not relay on any task-specific, predefined similarity function between input and output, nor do the author assume that the input and output have to lie in the same low-dimensional embedding space.

- Cycle Consistency

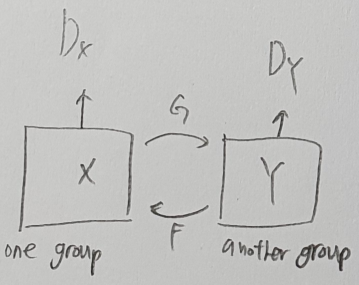
There a lot of work done using transitivity as a way to regularize structured data. And work done by zhou et al and Godard et al. use cycle consistency loss as a way of using transitivity to supervise CNN training. This is similar to what is proposed in this paper work as the writer introduce a loss to push G and F to be consistent with each other.

- Neural Style Transfer

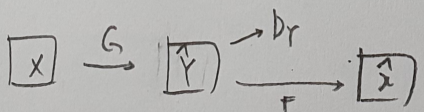
Synthesizing a novel image by combining the content of one image with the style of another image based on matching the Gram matrix statistics of pre-trained deep feature. Neural Style transfer, is another way to perform image-to-image transfer. Hoever, in this paper they focus on learning the mapping between two image collections rather than between two specific images.

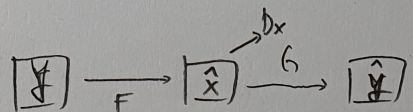
[2] Architecture & Architecture Comparison

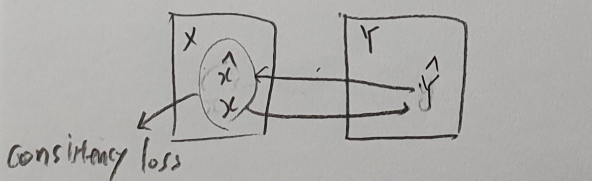
The proposed model contains two mapping functions : $G: X \rightarrow Y$ and $F: Y \rightarrow X$ and associated adversarial discriminators D_Y and D_X. The illustrations is on the left side.



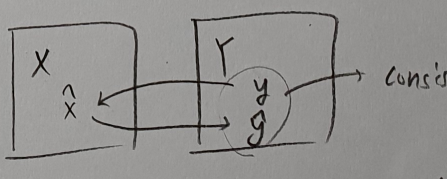
D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X. The writers introduced two cycle consistency losses. that capture the intuition that if they translate from one domain to the other domain and back again the should arrive at where they started.







Forward cycle-consistency loss



backward cycle-consistency loss

The goal of the proposed model is to learn mapping functions between two domain X and Y given training samples $\{x_i\}_{i=1}^{N}$ where $x \in X$ and $\{y\}_{i=1}^{N}$ where $y \in Y$.

2022421 심재윤 (JAEYOON SIM)

- Objective

The objective function is made of two losses i) adversarial loss ii) cycle consistency loss.

$$L(G, F, D_X, D_Y) = \underbrace{L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X)}_{\text{adversarial loss}} + \underbrace{\lambda L_{cyc}(G, F)}_{\text{cycle consistency loss}}$$

i) Adversarial Loss

The adversarial loss is both applied to two mapping function F and G. Adversarial loss for G is $L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}}[\log(1 - D_Y(G(x)))]$ and vice versa for F.

ii) Cycle Consistency Loss

GAN does not force output to correspond to input and therefore mode collapse occurs. Thus, adversarial losses alone cannot guarantee that the learned function can map and individual input x_i to a desired output y_i. To further reduce the space of possible mapping functions, they argue that the learned mapping functions should be cycle-consistent. Therefore, they use cycle consistency loss as follow. $L_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1]$. Their aim is to solve $G^*, F^* = \arg\min_{G,F} \max_{D_X, D_Y} L(G, F, D_X, D_Y)$.

[3] Training Detail

For L_{GAN}, they replace the negative log likelihood objective by a least square losses. This loss is more stable during training and generates higher quality results. They train the G to minize $\mathbb{E}_{x \sim p_{data}(x)}[D(G(x)) - 1)^2]$ and train D to minimize $\mathbb{E}_{y \sim p_{data}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x \sim p_{data}(x)}[D(G(x))^2]$. Additionally, to reduce model oscillation, they follow Shrivastava et al.'s strategy and update the discriminator using a history of generated images rather than the ones produced by the latest generators. For all the experiments they set $\lambda = 10$ in objective function and used the Adam solver with a batch size 1. All networks were trained from scratch with a learning rate of 0.0002.

[4] Performance Comparison

They use AMT perceptual studies and FCN scores for evaluation and compared per pixel accuracy, per-class accuracy and class IoU between CoGAN, BiGAN, SimGAN, feature loss GAN, and Cycle GAN (proposed). except for $y \rightarrow x$ (labels to photo) the proposed model was the best.