GANs N' Roses

Sunday, June 27, 2021 10:18 PM

- Link to paper: https://arxiv.org/pdf/2106.06561.pdf

- Abstract

  ⇒ Inputs

  1. Content code derived from human image
  2. Style code chosen randomly

  ⇒ Output = anime image taking the pose of the human in the human image.

  ⇒ New adversarial loss based on new definition of content and style

- Intro

  ⇒ The paper achieves better human-to-anime pose transfer by defining new losses based on new definition of "content" and "style"

  - Should it just use the word "pose" for "content" and "appearance" for "styele"?

  ⇒ Idea

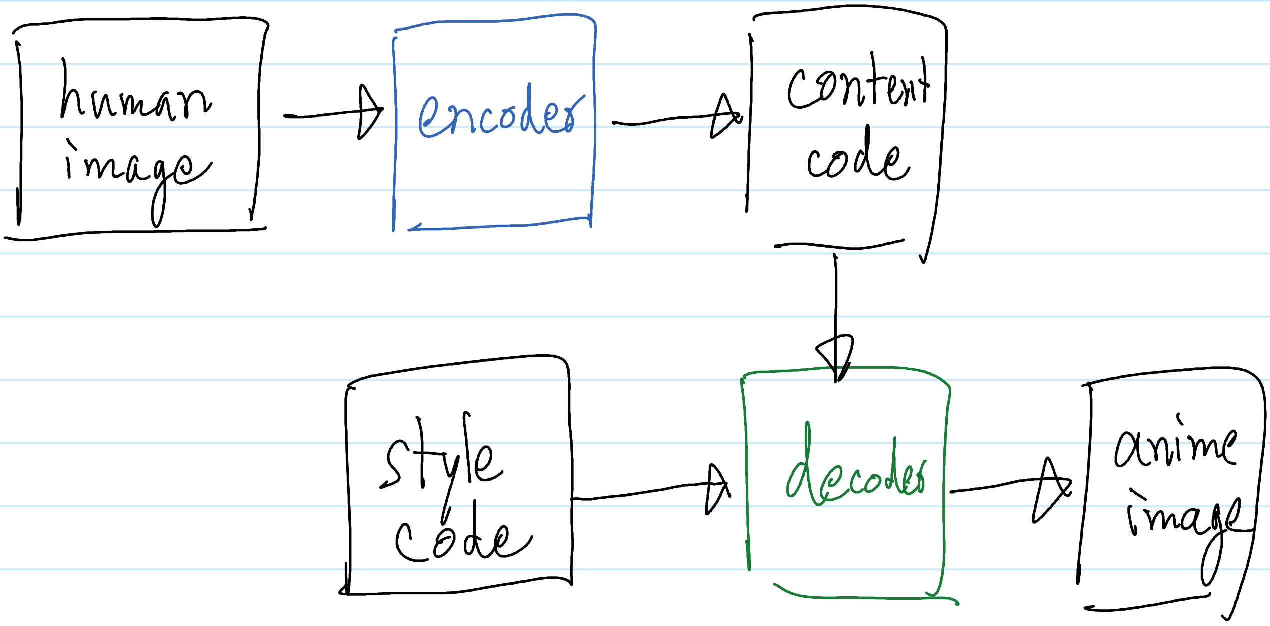
  - "Content" is what changes if face image undergoes a family of data augmentation transformations.

  - Style is what does not change

Page 2

Sunday, June 27, 2021 10:52 PM

- Augmentations = scaling, rotating, cropping.
- How this works



- To make sure that the anime image has the same pose as the human image, we must ensure that the anime image has the same content code as the input human image
- However, we cannot use cycle consistency loss because we do not want a 1-1 mapping between human and anime face.
- The paper proposes a way to ensure the content code is the same without using cycle consistency loss

Page 3

Sunday, June 27, 2021 11:08 PM

- The Related Works session has two paper related to anime.
  - CountcilGAN [LINK]
    - Generate diverse anime face from a single human face using multiple generators working in parallel
    - Problems
      - Cannot capture diverse anime styles.
      - Mode collapses.
  - AniGAN [LINK]
    - New normalizations that
      - Transfer color and texture styles
      - Maintaining global structure
    - Problem: Style not diverse enough.

- Framework

⇒ Two domains X (human face)
Y (anime face)

⇒ Goal: Given $x \in X$,
generate a subset $\hat{Y} \subseteq Y$
such that each $y \in \hat{Y}$ contains
similar semantic content as x.

⇒ While goal is only in direction $X \to Y$,
we need a mechanism to do $Y \to X$
as well. So, 2 directions in total.

⇒ For each direction, we need
① an encoder, denoted by $E_{X \to Y}$, $E_{Y \to X}$
② a decoder, denoted by $F_{X \to Y}$, $F_{Y \to X}$

Page 4

Sunday, June 27, 2021 11:37 PM

- image → $\boxed{\text{encoder}}$ → content code, style code (random)

- content code, style code (random) → $\boxed{\text{decoder}}$ → image.

- Generator = encoder + decoder

- Main idea: Choose a relevant collection of data augmentation. Under it:
  Content is what changes
  Style is what does not change

- Ensuring style diversity
  ⇒ Existing strategy.
  - Generate from random style code
    ⇒ The decoder might ignore style code.
  - Make sure that the style code can be recovered from the generated image
    ⇒ The decoder might hide it in a few pixels

Page 5

Monday, June 28, 2021 12:06 AM

- Force outputs from different style codes to be different.

  ⇒ No garantee that this is the right diversity.

⇒ Let $P(X)$ denote probability distribution of X.

$T(\cdot)$ = a function that applies a random augmentation that changes contend and preserve style.

$P(C)$ = distribution of content codes

$P(Y)$ = distribution of Y.

$P(\hat{Y})$ = distribution of $F_{X \to Y}(c(x), s_z)$ where $x \sim P(X)$, $s_z \sim N(0,1)$

content code of x

normal distribution

⇒ Note that $c(x_i) \sim P(C)$ if $x_i \sim P(x)$

⇒ Requirement on T: $c(T(x_i)) \sim P(C)$

(i.e. while the exact content code would change, the overall distribution does not)

Page 6

Monday, June 28, 2021 12:21 AM

⇒ The paper note that the last requirement is reasonable. Otherwise, augmentation used when training classifiers would not work.

⇒ IMHO, I doubt this. Rotation changes the pose significantly.

⇒ If you have a dataset with mainly heads in upright position, rotating the image can change the range of head angles.

⇒ After we generate $\hat{y} = F_{X \to Y}(C(x_i), s_z)$ we pass it to a discriminator D to judge whether the generated image is real or fake.

⇒ The paper proposing generating a fake batch in the following way:

(a) choose a single $x \in X$.

(b) Compute $x_1 = T(x)$, $x_2 = T(x)$, ..., $x_8 = T(x)$ (These are randomly augmented examples)

(c) compute $c_i = E_{X \to Y}(x_i)$

Monday, June 28, 2021 12:35 AM

(d) Same random style code $z_1, z_2, \dots, z_k \sim N(0, 1)$

(e) Generate $\hat{y}_i = F_{X \to Y}(c_i, z_i)$

$\Rightarrow$ The paper proposes that the batch $\{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_k\}$ should be indistinguishable from a batch of samples from $P(Y)$ of the same size.

This is the basis of its adversarial loss func

Note that this goal ensures that a single content code can be translated to multiple, diverse style.

Monday, June 28, 2021 12:44 AM

- Losses

⇒ Style consistency loss. For each batch generated as above the style must be the same. So, the variance on the style code must be low:

$$\mathcal{L}_{scon} = Var(s(x_1), s(x_2), \ldots, s(x_k))$$

⇒ Cycle consistency loss

Let $\hat{y}_i = F_{X \to Y}(c_i, z_i)$

$\hat{x}_i = F_{Y \to X}(c(\hat{y}_i), s(x_i))$ ⇐ this should equals x_i

However, the style codes can be different and this can allow information about content to leak through it. The paper thus shuffles the style codes before computing $\hat{x}_i$

So, now $\hat{x}_i = F_{Y \to X}(c(\hat{y}_i), s(x_{\pi[i]}))$ for some random permutation π,

The loss is then

https://arxiv.org/abs/1801.03924

$$\mathcal{L}_{cyc} = E\left[\| x_i - \hat{x}_i \|_2 + \lambda \cdot \mathrm{LPIPS}(x_i, \hat{x}_i) \right]$$

⇒ Adversarial loss

- Normal discriminator

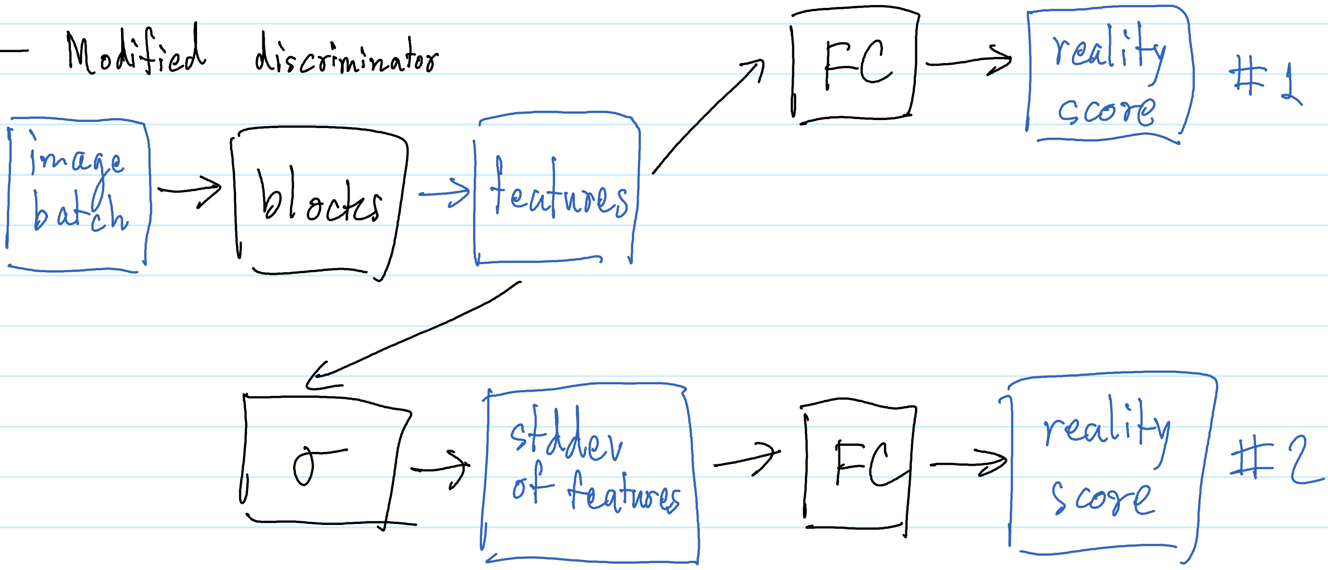
$\boxed{\text{image batch}} \rightarrow \boxed{\text{blocks}} \rightarrow \boxed{\text{features}} \rightarrow \boxed{\text{FC layers}} \rightarrow \boxed{\text{reality score}}$

- The paper uses the minibatch standard deviation trick used in the progressive GAN paper.

Page 10

Monday, June 28, 2021 1:06 AM

- Modified discriminator



- Both reality scores are then use in the standard non-saturating GAN loss.

- The paper also uses the R1 regularization term from [Mescheder et. al. 2018] (https://arxiv.org/pdf/1801.04406.pdf) on both reality scores. This involves restricting gradients w.r.t. to inputs of the discriminator output.

Page 11

Monday, June 28, 2021 1:15 AM

⇒ The whole loss

$$\mathcal{L} = \lambda_{adv} \mathcal{L}_{adv} + \lambda_{scon} \mathcal{L}_{scon} + \lambda_{cyc} \mathcal{L}_{cyc}$$

1 10 20

- Implementation details
  - ⇒ Architecture = StyleGAN 2
  - ⇒ Style code $\in \mathbb{R}^8$ (Huh?)
  - ⇒ Batch size = 7
  - ⇒ Adam with learning rate 0.002 for 300k iterations.
  - ⇒ Augmentations
    - horizontal flip
    - rotation between $(-20^\circ, 20^\circ)$
    - scaling between $(0.9, 1.1)$
    - translation up to $(0.1, 0.1)$
    - shearing up to 0.15
    - Upscale to 286×286 then crop to 256×256 randomly.

Page 12

Monday, June 28, 2021 1:23 AM

=> Datasets — selfie2anime
AFHQ = animal faces

- Experiments

=> GANs N' Roses (GNR) produced more divese images from same human face + random style codes than DRIT++, CouncilGAN, and AniGAN.

http://vllab.ucmerced.edu/hylee/DRIT_pp/

=> The paper observed that the batch standard deviation trick is important to ensure diversity

=> The paper also outperformed others in several metrics: FID, DFID (original), LPIPS pairwise distances.