Latest Artificial Intelligence Paper Study

U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation

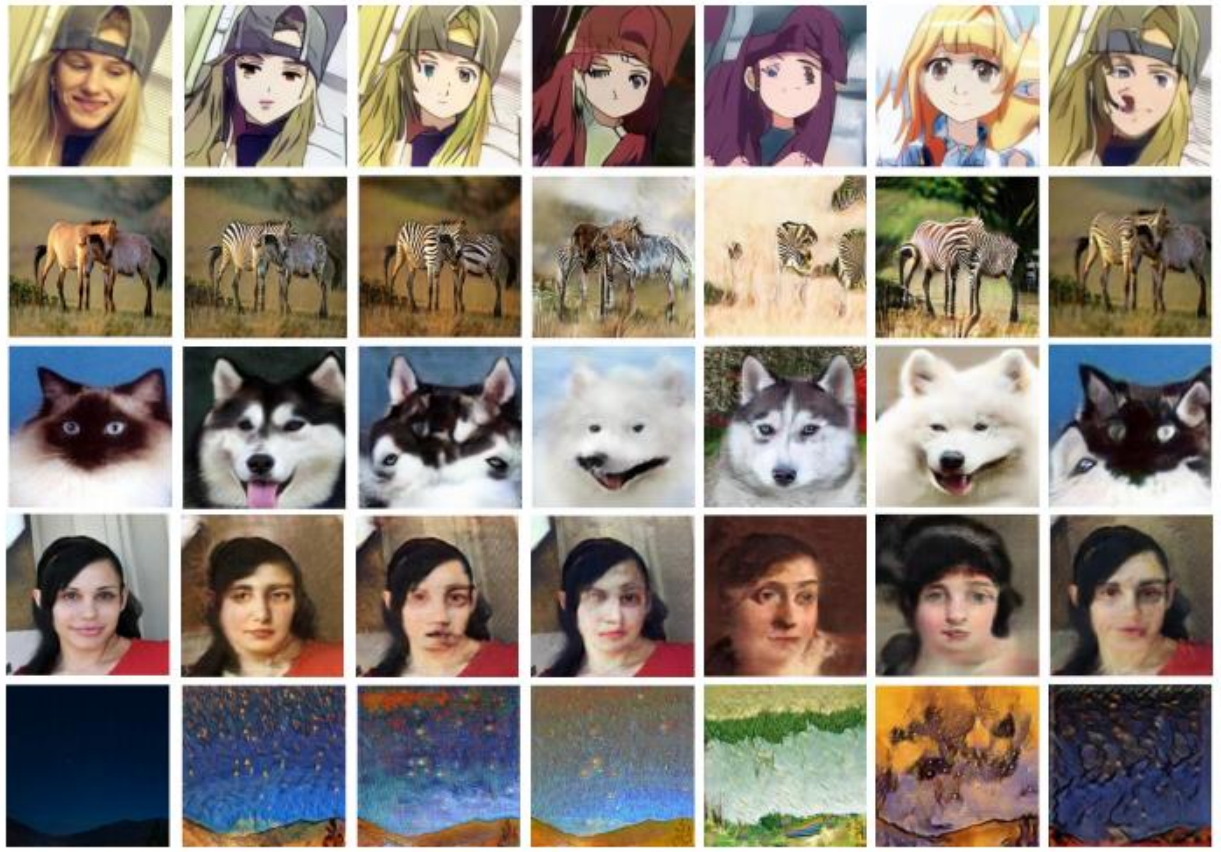
1ST WEEK (2020/09/02)

프리드리히

Contents

1. Introduction

- Introduces new normalization technique, called AdaLIN (Adaptive Layer Instance Normalization)
- Application of CAM (Class Activation Map) and attention mechanism that enables shape and texture transformation without changes in model architecture and hyper-parameter tuning.



2. Related Works: Generative Adversarial Network

- Generator synthesizes realistic images to fool Discriminator while Discriminator distinguishes real and fake data
- Applications

  1) Image-to-Image Translation

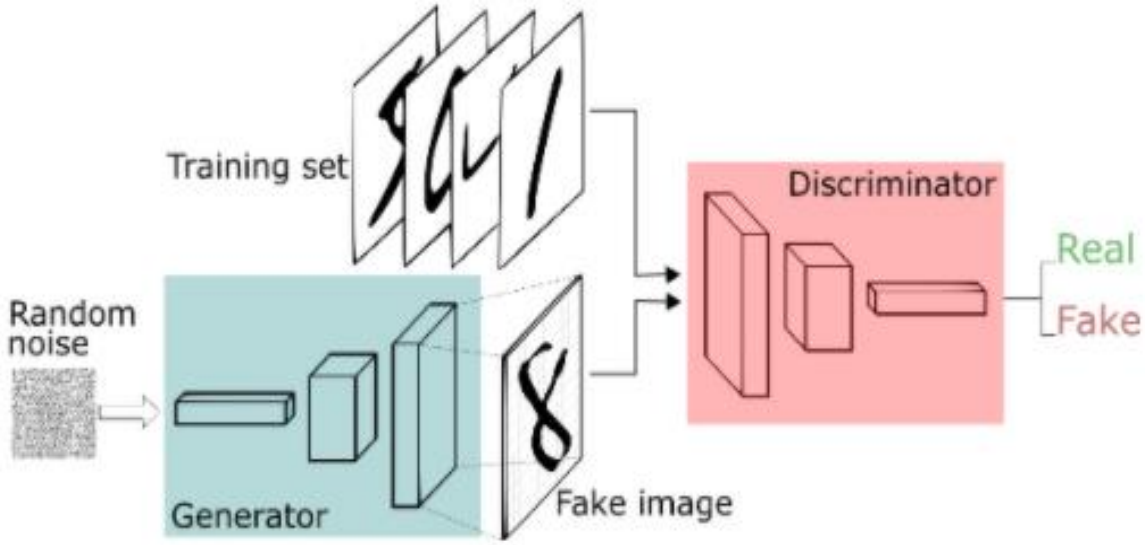
  2) Text-to-Image Synthesis

  3) Denoising, Deblurring

  4) Super Resolution

  5) Image inpainting

  6) Colorization



$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim Pdata(x)}[logD(x)] + \mathbb{E}_{z\sim Pz(z)}[\log(1 - D(G(z)))]$$

2. Related Works: Image-to-Image Translation

1. Mapping source S to target T via GANs

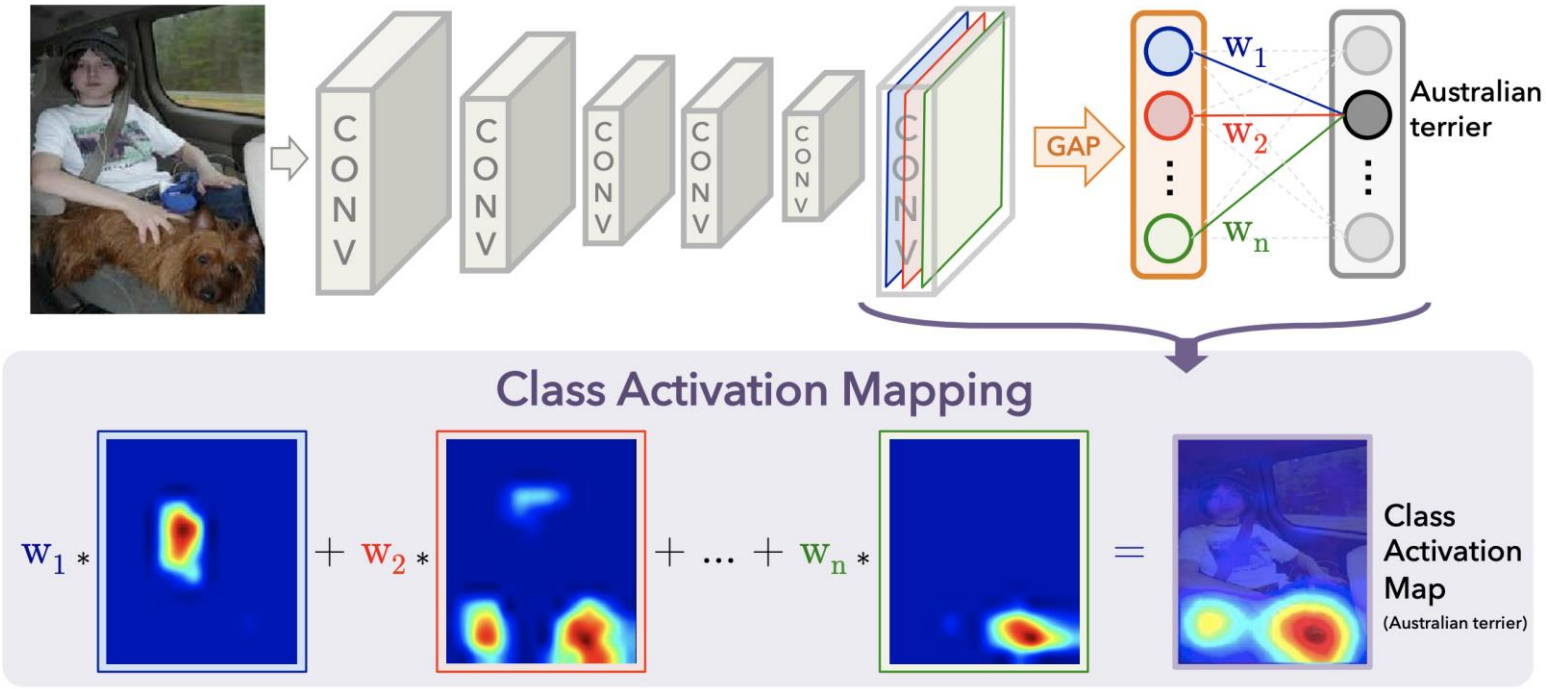
2. Supervised Learning
   - Unimodal – Pix2Pix
   - Multimodal – BicycleGAN

3. Unsupervised Learning
   - Unimodal – CycleGAN, DiscoGAN
   - Multimodal – StarGAN

2. Related Works: CAM (Class Activation Maps)

1. Visualizing feature map that affects classification of a model

2. Visualization is derived by multiplying weights, that are yielded when calculating probability of being belonged to a specific class c, to feature maps and add them.



2. Related Works: Attention Mechanism

1. Mimic how people attend; we concentrate on a specific point instead of a whole image

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

2. Related Works: Normalization

1. Batch Norm
   - 1) Reduce the internal covariance shift
   - 2) Significantly stabilize training by normalizing input batch data via whitening (N~(0, 1))
2. Layer Norm
   - 1) Normalize features
   - 2) Irrelevant to batch size and shows better performance when applied to RNNs
3. Instance Norm
   - 1) Similar to Layer Norm but normalize filters as well
   - 2) Shows the best performance to generative models amongst three normalization

2. Related Works: Normalization

4. Adaptive Instance Normalization
   - 1) Receives content input x and style input y
   - 2) Matching the channel-wise mean and variance of x with the mean and variance of y
   - 3) Adaptively calculate using style input y
   - 4) Widely used to style-transfer algorithms

2. Related Works: Normalization

5. Adaptive Layer Instance Normalization
   - 1) Adaptive Instance Normalization + Layer Normalization
   - 2) Balance between style and content

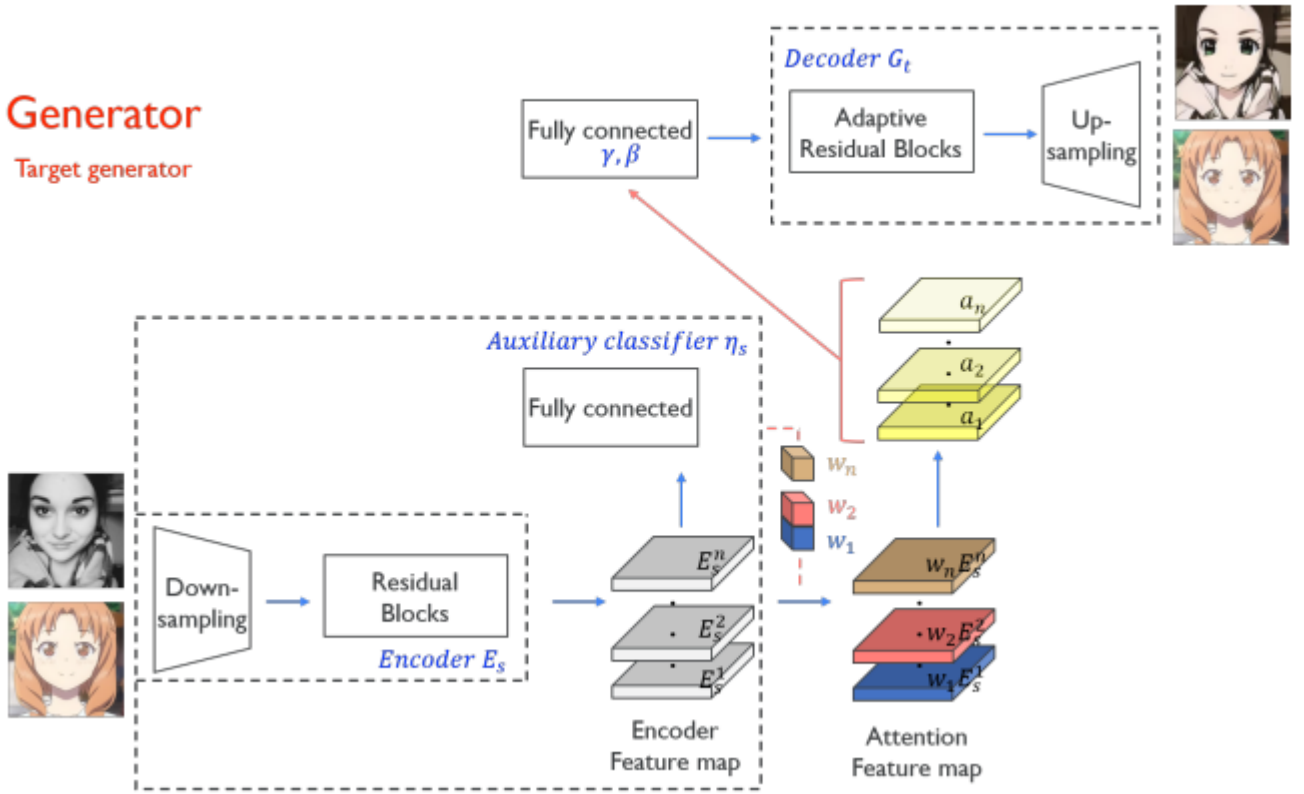
$$AdaLIN(a,\gamma,\beta) = \gamma \cdot (\rho \cdot \hat{a}_I + (1-\rho) \cdot \hat{a}_L) + \beta,$$

$$\hat{a}_I = \frac{a - \mu_I}{\sqrt{\sigma_I^2 + \epsilon}}, \hat{a}_L = \frac{a - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}},$$

$$\rho \leftarrow clip_{[0,1]}(\rho - \tau \Delta \rho)$$

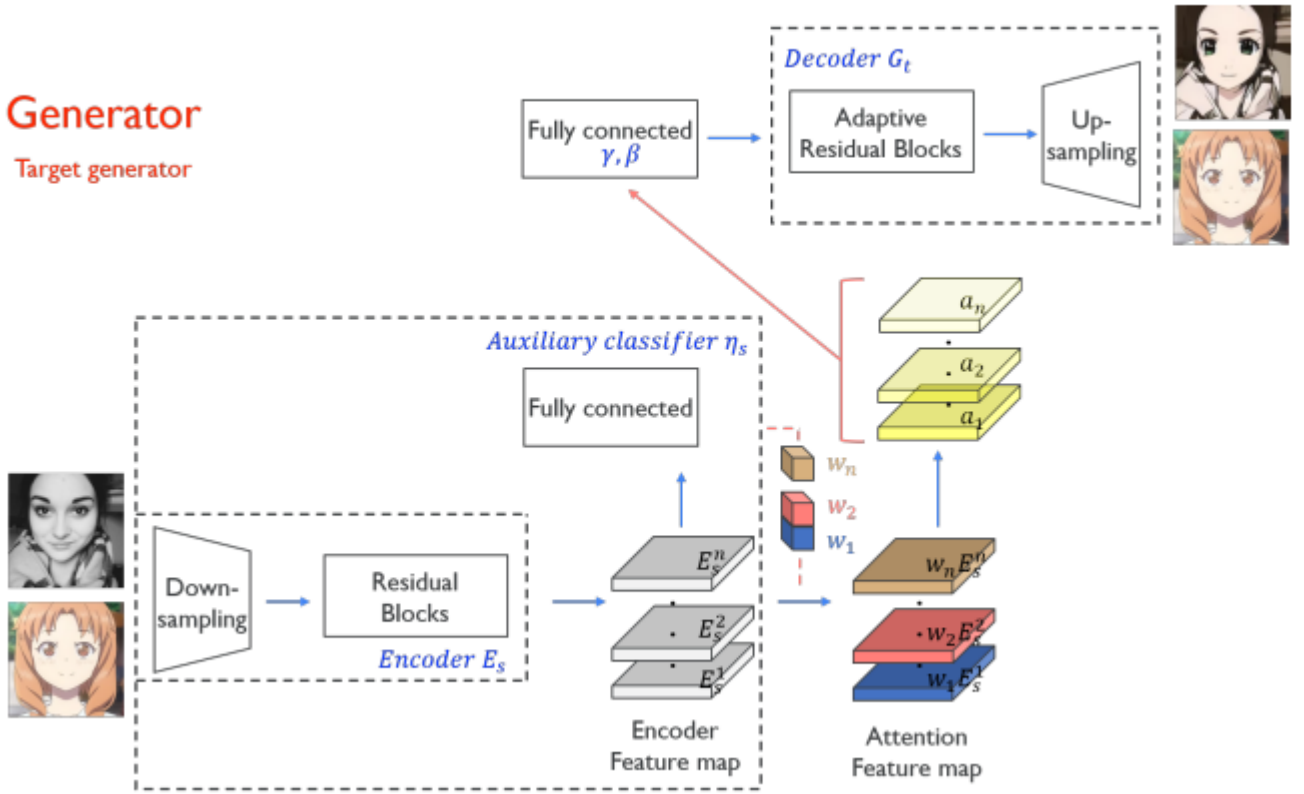
3. Model Architecture

1. Generator (Overview)
   - Generates translated images
   - Comprised of Encoder, Auxiliary Classifier, Fully Connected and Decoder
   - Similar to typical Encoder-Decoder architecture
   - But applied attention mechanism and AdaLIN to the Encoder and applied AdaLIN to the Decoder



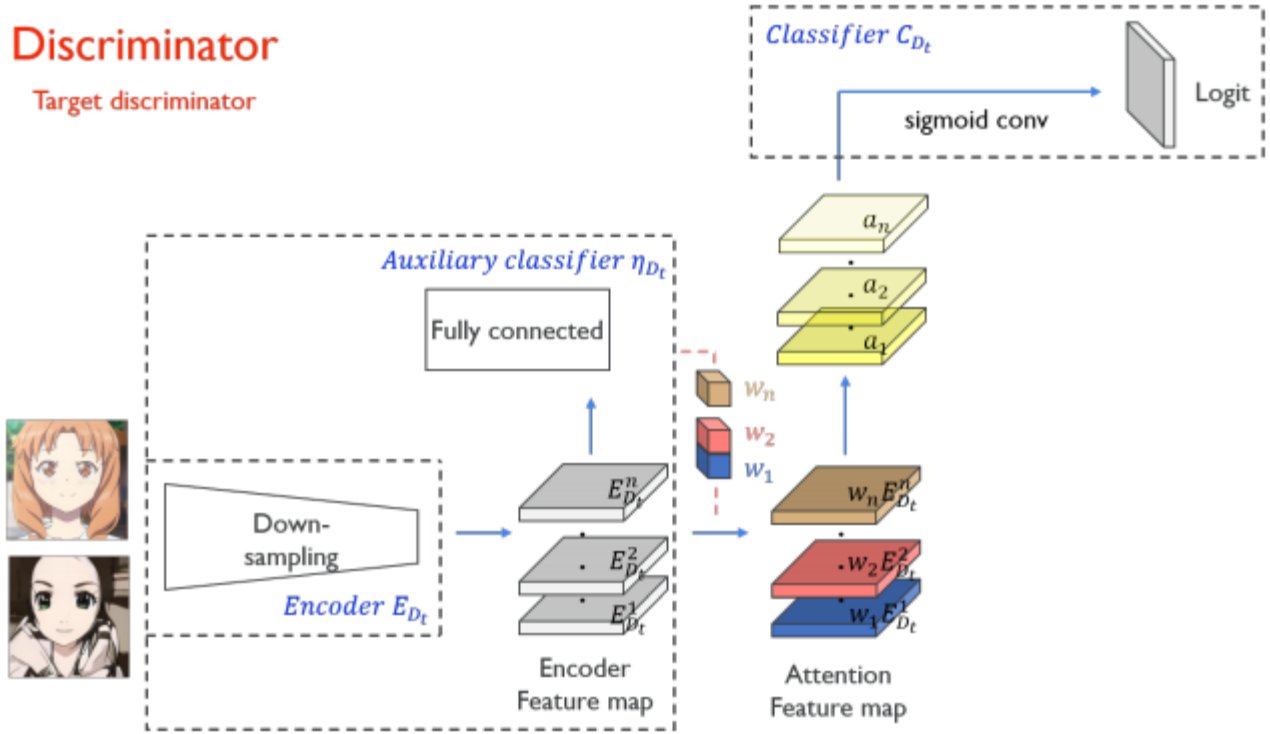
3. Model Architecture

2. Generator (Auxiliary Classifier)
   - Calculate probability of feature map of input image is the same as feature map of source image
   - CAM is also derived in order to learn important weights of feature map
3. Generator (Fully Connected)
   - Calculate γ and β for de-normalization process
4. Generator (Decoder)
   - Generate Images using weighted feature map



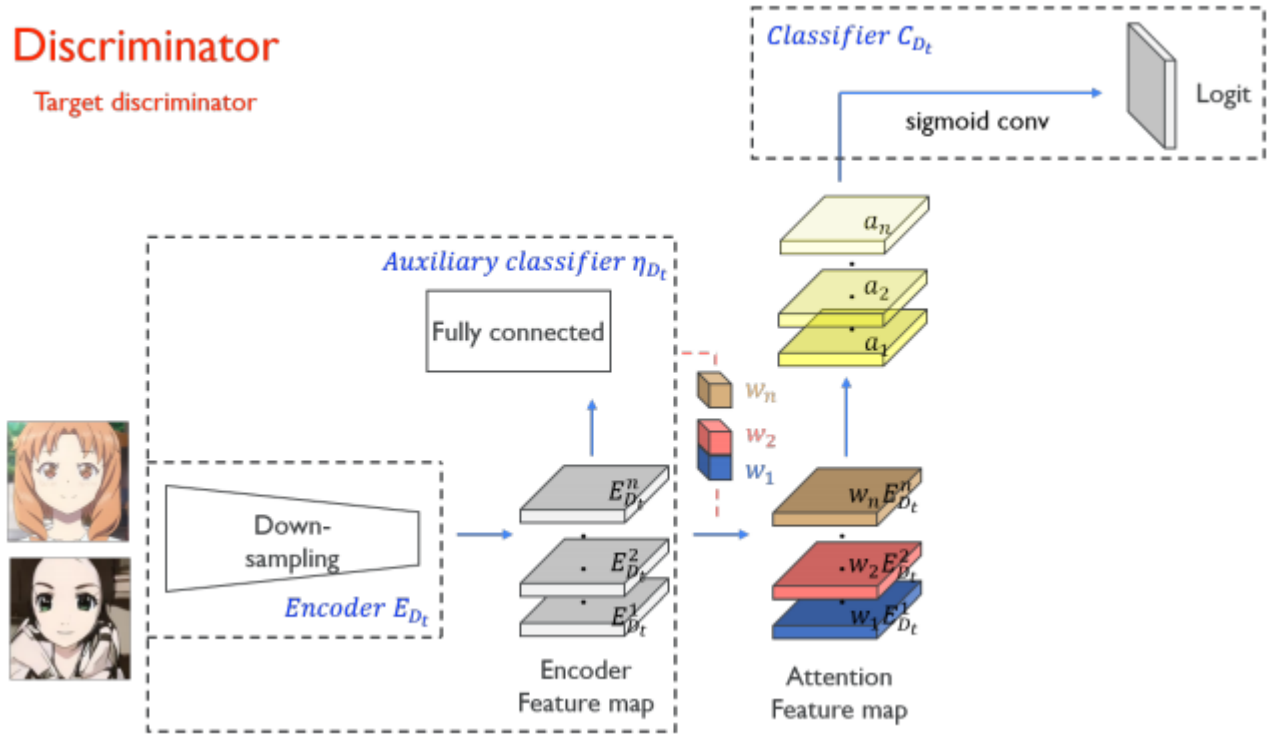
3. Model Architecture

1. Discriminator (Overview)
   - Distinguishes real (source) and fake (translated) images
2. Discriminator (Encoder)
   - Structure and purpose are the same as a Generator
3. Discriminator (Auxiliary Classifier)
   - Structure is the same classifies generated (fake) images and domain images.
   - Purpose is different: to distinguish real and fake Images



3. Model Architecture

4. Discriminator (Decoder)
   - Creates a logit that classifies from which feature map is.
   - Two types of discriminator: Local and Global
   - Utilizes PatchGAN



4. Objective Function

1. Full Objective
   - $$\min_{G_{s\to t}, G_{t\to s}, \eta_s, \eta_t} \max_{D_s, D_t, \eta_{D_s}, \eta_{D_t}} \lambda_1 L_{lsgan} + \lambda_2 L_{cycle} + \lambda_3 L_{identity} + \lambda_4 L_{cam}$$
2. Adversarial Loss
   - LSGAN (L2Loss)
   - To match probability distribution of translated images and that of target image distribution

2. Cycle Loss (L1 Loss)
   - To ensure that the reconstructed (translated back to original) image is similar to original image
   - Also constraints mode collapse problem

3. Identity Loss (L1 Loss)
   - To ensure the color distribution of input image and that of output image

4. CAM Loss
   - This enables networks to know where and what makes the most difference between two domains

5. Qualitative Evaluation: Ablation Study of CAM

1. Absence of CAM deteriorates results: see (e) and (f).

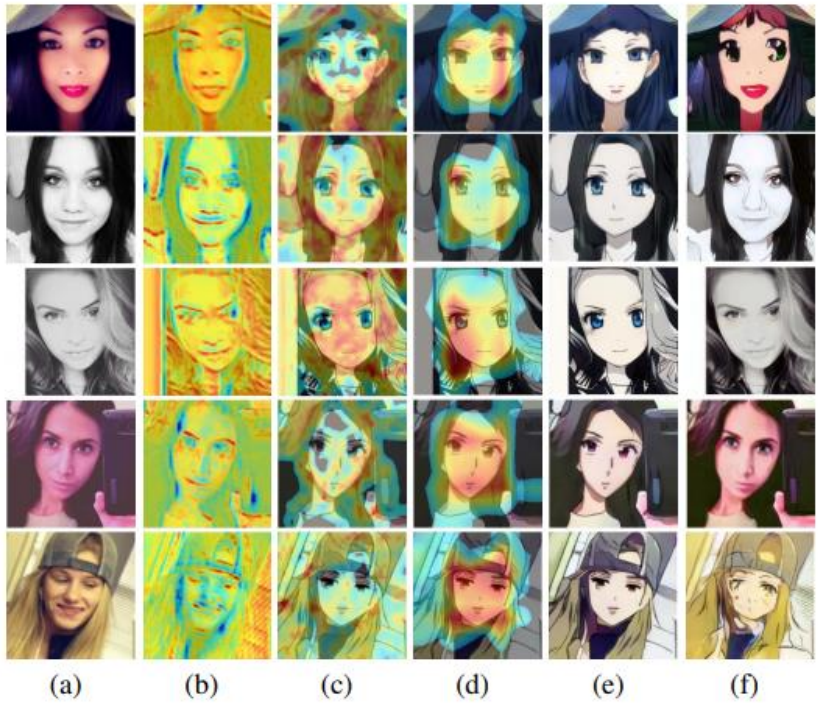


Figure 2: Visualization of the attention maps and their effects shown in the ablation experiments: (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminator, respectively. (e) Our results with CAM, (f) Results without CAM.

5. Qualitative Evaluation: Various Normalization

1. (c) and (d) shows well-maintained shape and more stylization, respectively.

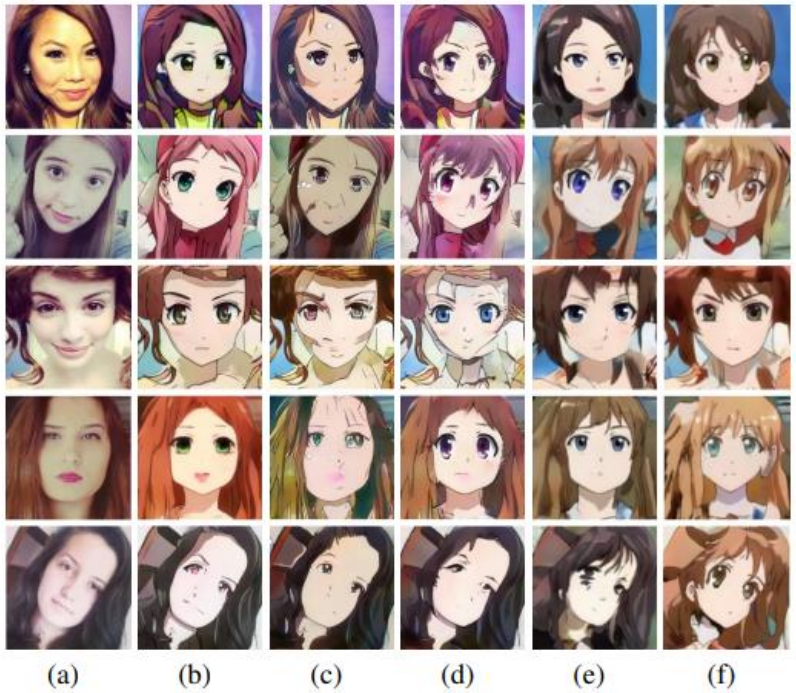


Figure 3: Comparison of the results using each normalization function: (a) Source images, (b) Our results, (c) Results only using IN in decoder with CAM, (d) Results only using LN in decoder with CAM, (e) Results only using AdaIN in decoder with CAM, (f) Results only using GN in decoder with CAM.

5. Qualitative Evaluation: With Other Papers

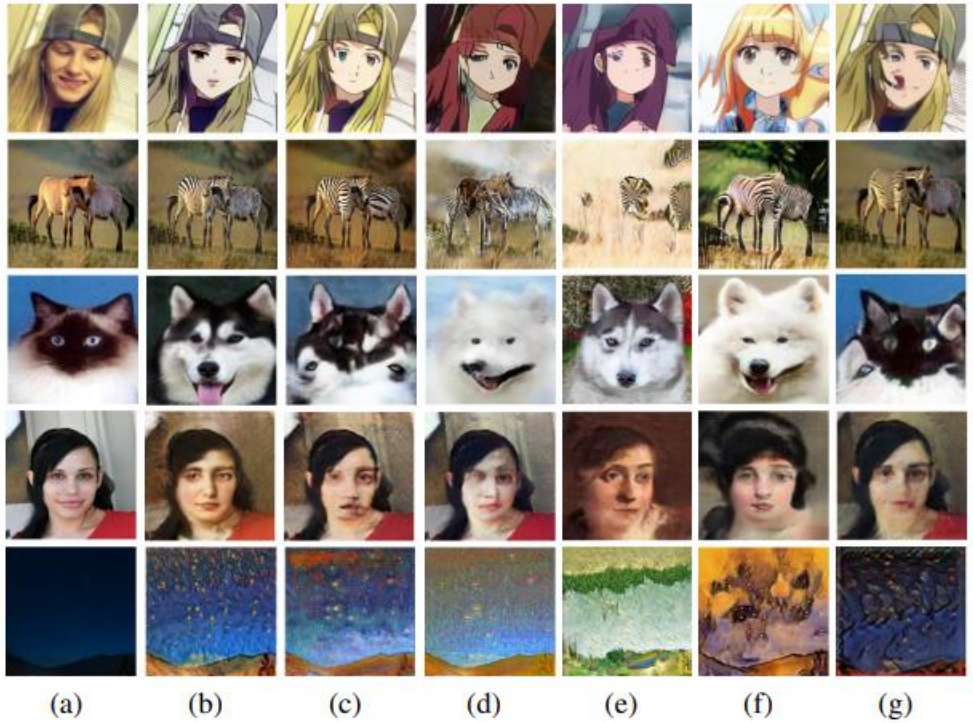


Figure 4: Visual comparisons on the five datasets. From top to bottom: selfie2anime, horse2zebra, cat2dog, photo2portrait, and photo2vangogh. (a)Source images, (b)U-GAT-IT, (c)CycleGAN, (d)UNIT, (e)MUNIT, (f)DRIT, (g)AGGAN

6. Quantitative Analysis

KID (Kernel Inception Distance)

- Similar to FID (Frechet Inception Distance)
- FID exploits Wasserstein-2 Distance while KID uses samples drawn independently from two distributions.
- More appropriate to measuring distance between feature map that are passed activation functions.
- AdaLIN and CAM are in Inter-dependent relationship; Applying both outputs better results.

Table 1: Kernel Inception Distance$\times 100 \pm$std.$\times 100$ for ablation our model. Lower is better. There are some notations; GN: Group Normalization, G_CAM: CAM of generator, D_CAM: CAM of discriminator

Model	selfie2anime	anime2selfie
U-GAT-IT	**11.61 ± 0.57**	**11.52 ± 0.57**
U-GAT-IT w/ IN	13.64 ± 0.76	13.58 ± 0.8
U-GAT-IT w/ LN	12.39 ± 0.61	13.17 ± 0.8
U-GAT-IT w/ AdaIN	12.29 ± 0.78	11.81 ± 0.77
U-GAT-IT w/ GN	12.76 ± 0.64	12.30 ± 0.77
U-GAT-IT w/o CAM	12.85 ± 0.82	14.06 ± 0.75
U-GAT-IT w/o G_CAM	12.33 ± 0.68	13.86 ± 0.75
U-GAT-IT w/o D_CAM	12.49 ± 0.74	13.33 ± 0.89

6. Quantitative Analysis

1. Shows relatively weak performance when applied to photo2portrait and photo2vangoh dataset

2. But outperforms other papers when applied to other domain adaptation dataset.

Table 3: Kernel Inception Distance×100±std.×100 for difference image translation mode. Lower is better.

Model	selfie2anime	horse2zebra	cat2dog	photo2portrait	photo2vangogh
U-GAT-IT	**11.61 ± 0.57**	**7.06 ± 0.8**	**7.07 ± 0.65**	1.79 ± 0.34	4.28 ± 0.33
CycleGAN	13.08 ± 0.49	8.05 ± 0.72	8.92 ± 0.69	1.84 ± 0.34	5.46 ± 0.33
UNIT	14.71 ± 0.59	10.44 ± 0.67	8.15 ± 0.48	**1.20 ± 0.31**	**4.26 ± 0.29**
MUNIT	13.85 ± 0.41	11.41 ± 0.83	10.13 ± 0.27	4.75 ± 0.52	13.08 ± 0.34
DRIT	15.08 ± 0.62	9.79 ± 0.62	10.92 ± 0.33	5.85 ± 0.54	12.65 ± 0.35
AGGAN	14.63 ± 0.55	7.58 ± 0.71	9.84 ± 0.79	2.33 ± 0.36	6.95 ± 0.33
CartoonGAN	15.85 ± 0.69	-	-	-	-
Model	anime2selfie	zebra2horse	dog2cat	portrait2photo	vangogh2photo
U-GAT-IT	**11.52 ± 0.57**	**7.47 ± 0.71**	**8.15 ± 0.66**	1.69 ± 0.53	5.61 ± 0.32
CycleGAN	11.84 ± 0.74	8.0 ± 0.66	9.94 ± 0.36	1.82 ± 0.36	**4.68 ± 0.36**
UNIT	26.32 ± 0.92	14.93 ± 0.75	9.81 ± 0.34	**1.42 ± 0.24**	9.72 ± 0.33
MUNIT	13.94 ± 0.72	16.47 ± 1.04	10.39 ± 0.25	3.30 ± 0.47	9.53 ± 0.35
DRIT	14.85 ± 0.60	10.98 ± 0.55	10.86 ± 0.24	4.76 ± 0.72	7.72 ± 0.34
AGGAN	12.72 ± 1.03	8.80 ± 0.66	9.45 ± 0.64	2.19 ± 0.40	5.85 ± 0.31

7. Conclusion

1. A new normalization technique called AdaLIN (Adaptive Layer Instance Normalization)

2. A new method of applying both CAM and attention mechanism that enables flexible shape and texture transformation without changing the model structure or hyperparameter-tuning with varying domain.