StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation [1]

Niharika Dalsania
DA-IICT, Gandhinagar, India, 382007
201701438@daiict.ac.in

Ruchit Shah
DA-IICT, Gandhinagar, India, 382007
201701435@daiict.ac.in

Zeel Patel
DA-IICT, Gandhinagar, India, 382007
201701443@daiict.ac.in

Darshan Patel
DA-IICT, Gandhinagar, India, 382007
201701436@daiict.ac.in

Abstract

Research on image-to-image translation models has shown very good results. But those are limited to two domains. For multiple domains, one has to train separate models for each pair of input domains, which is tedious. To solve that issue, this paper proposes StarGAN approach which performs image-to-image translation for multiple domains using single model. This property of StarGAN allows training the model on datasets with diverse attributes. Because of this StarGAN is able to flexibly transferring an input image to an image of target domain. We will be showing results of the model on facial attribute transfer task.

1. Introduction

Facial expressions and attributes provide a natural description of facial images and provide a perspective for understanding several facial manipulation tasks. Image-to-image translation changes particular aspect of a image to other with results ranging from changing attributes like hair color, gender, age, facial expressions, adding/removing attributes like glasses etc. Attributes typically mean the features inherent in an image (like hair color, gender etc.) and attribute values are the set of values from which the attribute takes one (like brown/black for hair color, male/female for gender etc). A set of images with the same attribute value form a domain. Given the training data from two different domains, image-to-image translation models can convert images from one domain to the another. This process is aided by several image dataset that come with labeled attributes such as facial attributes (Eg. CelebA) or facial expressions (Eg. RaFD). The benefit of true multi domain image-to-image translation is harnessed by joint training a single model over such datasets with varied attributes.

2. Related Works

In recent years, a variety of models have been developed for peculiar facial attribute transfer tasks and have given impressive results. Convolutional Neural Networks (CNNs) lies at the base of such models which has the ability for human face generation with attributes. A typical architecture leveraging this ability of CNN is Generative Adversarial Networks (GANs), which is one of the state-of-the-art approaches for image generation. A typical GAN structure consists of a Discriminator that learns to differentiate between real and fake images and a generator that learns to Generate fake image of target domain that are close to identical to the real samples. The idea is to introduce adversarial loss so that generated image is realistic and discriminator fails to differentiate between real and fake images.

2.1. Image-to-Image Translation

Following sections briefly summarises the important analysis obtained via literature review of models used for Image-to-Image Translation prior to StarGAN.

2.1.1 DiscoGAN (Discover Cross-domain relations with GAN) [2]

The aim is to learn relations among multiple domains and perform translation on cross domains along with preserving identity in images. The model consists of two GANs in pairs. Each of these two learns the mapping from one domain to another and the reverse for reconstruction. Because of this, the total generator loss is the GAN loss added to the reconstruction loss for each partial model. The main benefit of this model is that it can be trained on pair of sets of images without explicitly mentioned pair labels, because it learns to relate datasets from diverse domains. Hence, it can generate high-quality images with transferred style.

2.1.2 CycleGAN [3]

The idea is to generate an image of a source domain X to a target domain Y with no paired examples. Like all adversarial networks, train a generator $G : x \rightarrow y$ s.t $\hat{y} = G(x)$ is not differentiable from images in set y by an discriminator D_x trained to classify $\hat{y}$ apart from set of images in y. Train another translator $F : y \rightarrow x$ with adversary D_y s.t. G and F are inverse of each other, because G does not guarantee the meaningful translation from x to y as there are infinitely many such G possible(random permutation). Train both the mapping G and F simultaneously, and add a cycle consistency loss that encourages $F(G(x)) \approx x(i.e., x \rightarrow G(x) \rightarrow F(G(x)) \approx x.)$ and $G(F(y)) \approx y$. Cycle consistency losses(forward and backward) prevent the learned mappings G and F from contradicting each other.

2.1.3 IcGAN (Invertible Conditional GAN) [4]

Aim here is to Identify the latent representation of image using encoder, and modify any attribute to get desired results. They train the conditional GAN to learn the corresponding features of the image with respect to its label. Adversarial loss is used in this paper also, to map between multiple domains and train the model. Taking the help from condition GAN IcGAN will learn the mapping of latent space along with attributes to the actual image. New images are conditioned on latent vectors and the attribute vectors. The encoders are trained as well, using a neural networks to learn the mapping. Another decoder is also used for mapping the attribute vectors based on the probability distribution of labels. Combining both encoder IcGAN is able to convert images from one domain to another.

2.1.4 DIAT (Deep Identity-Aware Transfer of Facial Attributes) [5]

To alleviate the issue of identity loss during image-to-image translation using prior models, the major objective of this model was to generate an image that has the reference attribute while retaining the identity of the input image. It consists of an attribute transform network to generate photo-realistic facial image with the reference attribute and a mask network to not allow the incorrect editing on the regions where a specific attribute is irrelevant. Its superior performance corresponds to multiple loss functions like adversarial attribute loss and identity loss as well as regularisation such as suppressing artifacts in transfer result and attribute ratio regularization to constraint size of attribute relevant region. This provides a unified solution for several representative facial attribute transfer tasks and gives impressive results even for identity related attribute like gender, while retaining most identity aware features.

2.2. Shortcomings of existing approaches

The models discussed are highly inefficient for multi-domain image translation tasks because to learn all mappings among k domains, k(k-1) trainings are required. Yet, they are not effective because instead of having global features, generators cannot fully exploit the entire training data individually.

2.3. Proposed solution

Resolving these issues, StarGAN model, will be able to learn mappings among multiple domains simultaneously. Instead of a fixed translation, the generator now learns to flexibly translate the image among domains. This simple idea is presented in Figure 1.

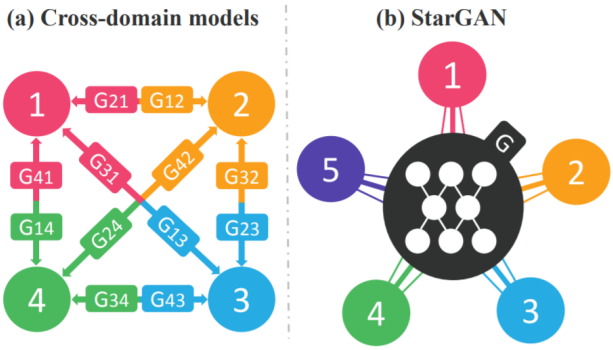

Figure 1. *Comparison between cross-domain models and StarGAN. [1]*

This architecture will overcome the drawbacks in previous models by providing robustness in learning through a multi-task learning framework from very big and different feature datasets. It will help improve sharpness of features and reduce efforts of training for each source target pair. Additionally, we will not train the model to do fix type of translations but we will give the target domain as an input and the model will be able to flexibly translate input image to any target domain.

The following section looks into the network architecture of StarGAN and explore the loss function it leverages for multi-domain image-to image translation.

3. Star Generative Adversarial Networks

The architecture overview of StarGAN reveals a generator. Input of generator will be image and target domain and it will learn the translation to that domain. We use labels to specify domain attribute. While training, the model is trained to translate an input image into a randomly generated target domain. This enables translating images into

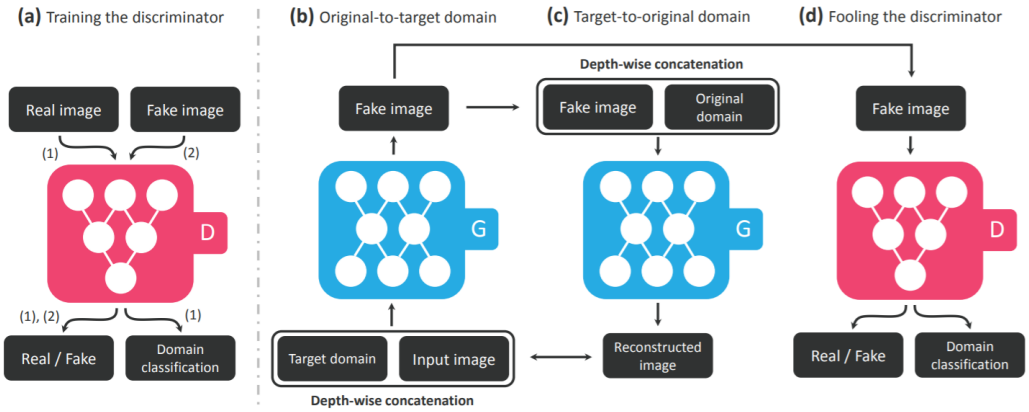

Figure 2. *Visualisation of training approach of StarGAN[1], D is discriminator and G is generator*

desired domain at testing phase by controlling the domain label. Also, mask vector is introduced for joint training between different domains. This makes the approach even more effective by ensuring that the model only considers the labels of given dataset while training and ignores the additional labels. These training/testing phases are presented in Figure 2.

We now look into the actual neural network architectures of Generator and Discriminator.

3.1. Network Architecture

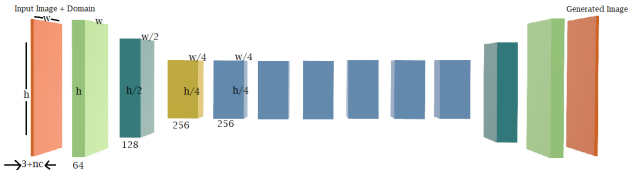

Figure 3. *Generator architecture*

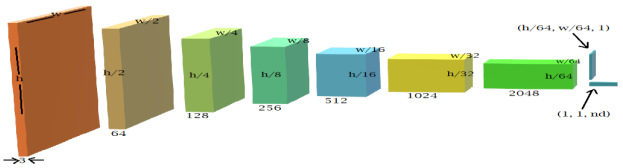

Figure 4. *Discriminator architecture*

StarGAN generator is like an autoencoder, which is has of two convolutional layers with the stride size of two for downsampling, six residual blocks, and two transposed convolutional layers with the stride size of two for upsampling. Instance normalization is applied in generator but not in discriminator. Discriminator is built using two convolutional layers with the stride size of two for downsampling. So this is like any simple CNN that does a classification task, but just that now we have an extra auxiliary classifier as well.

3.2. Multi-Domain Image-to-Image Translation

We will train a single Generator and a single Discriminator. G will learn mapping between multiple domains and will able to translate input image x into output domain y based on the given label c. For training purpose, c will be randomly chosen.So, $G(x, c) = y$ D will generate probability distribution of image being real or fake D_{src} and probability distribution over all the domains D_{cls}.

3.2.1 Adversarial Loss

This loss will be reduced so that generated images looks realistic. So, D will be trained so that it makes this loss value large for generated images(as they are not real) and small for real images. G will be trained so that it reduces this loss value for generated images to make generated images realistic. The same functionality is conveyed in the following equation.

$$L_{adv} = E_x[log(D_{src}(x))] + E_{x,c}[log(1 - D_{src}(G(x, c)))] \quad (1)$$

3.2.2 Domain Classification Loss

Given input image x and target domain c, G will generate $y = G(x, c)$. To make y properly classified to domain c, extra classifier is added on top of D and it's classification loss will be used as following: Domain classification loss of real images will be used to train D and domain classification loss of generated images will be used to train G.

$$L^r_{cls} = E_{x,c'}[-log(D_{cls}(c'/x))] \quad (2)$$

$$L^f_{cls} = E_{x,c}[-log(D_{cls}(c/G(x, c)))] \quad (3)$$

Here, $D_{cls}(c'/x)$ is the probability of the image x classified to domain c'. D will try to minimize L^r_{cls}, so that it can classify original images into their correct domain and G will try

to minimize L_{cls}^f, so that generated images can be classified to target domain c.

3.2.3 Reconstruction Loss

Cycle consistency loss just like cycle GAN will be introduced so that reconstructed images preserve the property of the original image. So, reconstruction loss will be,

$$L_{rec} = E_{x,c,c'}[||x - G(G(x,c),c')||_l] \tag{4}$$

Here, from original image x of domain c', image $y = G(x,c)$ of domain c will be generated using G. And with the same G, y will be reconstructed back to original domain image x. L_1 norm of original and reconstructed images will be used for the reconstruction loss here.

3.2.4 Final Objective

So, finally for Generator G and discriminator D objctive function will be,

$$L_D = -L_{adv} + \lambda_{cls} L_{cls}^r \tag{5}$$

$$L_G = L_{adv} + \lambda_{cls} L_{cls}^f + \lambda_{rec} L_{rec} \tag{6}$$

3.2.5 Gradient Penalty

The paper[1] is using Wasserstein GAN's adversarial loss functions instead of using the one explained above.

At the beginning discriminator is trained first and then the generator. So, initially the discriminator will of course be able to distinguish between real and fake and it will give D(G(z)) a value 0. And hence log(1-D(G(z))) term from which generator learns will be 0 and generator will not learn anything at all. Also, given an optimal D, G tries to minimize JS divergence between real and generated data distributions and if real and generated data distributions are far apart from each other, JS will give high value $\approx 2\log(2)$.

One solution for that problem is to use reverse JS divergence and adding noise. But that has the problem of very high gradient. Instead, we use losses proposed by WGAN which uses Wasserstein distance. Wasserstein distance just like transportation problem. It looks at the horizontal distance between 2 distributions. So, if 2 distributions are far apart, Wasserstein will still give +ve result. But this equation is not tractable. So, transformation is done using some duality. Now D will not be telling real and fake probability distribution anymore. Instead, it will be trained to become 1 K-Lipschitz continuous function to compute Wasserstein distance. For that transformation weight clipping was done. But authors of original WGAN paper said that weight clipping is terrible solution because of the issues of slow convergence after weight clipping and vanishing gradients. So now clipping is not done. Loss is penalized if the norm of gradient norm is not in range 1. [6]

$$\begin{aligned} L_{adv} = E_x[D_{src}(x)] - E_{x,c}[D_{src}(G(x,c))] \\ -\lambda_{gp} E_{\hat{x}}[E(||\nabla_{\hat{x}} D_{src}(\hat{x})||_2 - 1)^2] \end{aligned} \tag{7}$$

4. Training the model

To extensively train the entire model, original paper says that it 1 day is needed for a single NVIDIA Tesla M40 GPU. For the purpose of this study, such massive computations are infeasible. Such training is not feasible on loacl machines, although we present the results obtained from partial self training. Then for understanding the actual training process of the model, we look at the training process used in background for building the pretrained model.

4.1. The Self-trained Model

Apart from using the pretrained model for analysis, we built our own StarGAN model. Following are the details about our attempts to train the same and challenges encountered during that.

4.1.1 Setting up the environment

We restructured the original model code with updates such as upgrading the script from Tensorflow v1 to v2, replacing the deprecated libraries, converting the modular code into notebook format etc. We set up GPU on Google Colab for training this model.

4.1.2 Infeasibility of sufficient training

As far as the architecture is concerned, we used the same layer structure and hyperparameters for training. Even after reducing the dataset to just 2000 images and training the model over 20 epochs (100 iterations each) with batch size 20, the estimated time for training was above a day. Given the available computing resources, it is impossible to complete even this small amount training in an uninterrupted session. Considering the fact that even though successfully trained, this model will be massively undertrained for the complex task that I has to undertake. So, for demonstration purposes we will present the results and analysis obtained using the pretrained model.

4.2. The Pre-trained model

The model is trained using Adam with hyperparameters $\beta 1 = 0.5$ and $\beta 2 = 0.999$. Data is augmented by flipping the images horizontally with a probability of 0.5. The generator updates once every five discriminator updates. A

batch size of 16 is choosen for all experiments. The two main datasets used here, CelebA and RaFD, are presented in the sections below.

4.2.1 Datasets

1. **CelebA:** This dataset has 202,599 face images of celebrities, each with 40 binary attributes. The images are resized to 128×128 with 2,000 images as test images and remaining train images. We train using seven domains formed using the attributes hair color (black, blond, brown), gender (male/female) and age (young/old). While training, the learning rate is 0.0001 for the first 10 epochs (10000 iterations per epoch) and then linearly decays to 0 over the next 10 epochs.

2. **RaFD:** This dataset has 4,824 images collected from 67 participants, each making eight facial expressions in 3 gaze directions captured from 3 angles. The images here as well are resized to 128×128. To alleviate the issue of less training data in RaFD dataset, we use 100 epochs in place of 10.

4.2.2 Training with multiple datasets

While taking advantage of multiple datasets containing different types of labels, StarGAN faces an issue that the label information is only partially known to each dataset. While CelebA has labels for attributes like *hair color* and *gender*, there are no labels for facial expressions like *happy* and *angry* as in RaFD, and vice versa. This creates an issue because the complete information on the target domain label vector is needed while reconstruction of input image from the translated image. To alleviate this problem, a mask vector is used that allows StarGAN to simply focus on the explicitly known labels in the particular dataset while ignoring the others. For this, an n-dimensional one-hot vector is taken as the mask vector m for n datasets. The label vector now looks like $\bar{c} = [c_1, ..., c_n, m]$, where c_i is a vector for of labels from the i^{th} dataset. For the n-1 datasets except the dataset under consideration, the unknown labels in c_i's are set to all zeros. For this study, n = 2 (CelebA and RaFD). An overview of StarGAN learning from both CelebA and RaFD datasets is shown in Figure 4.

While training on multiple dataset with input domain label $\bar{c}$ to the generator, unspecified labels with zero vector will be ignored. Rest of the generator is unchanged, as in training with a single dataset. Also, the classifier on top of the discriminator generates probability distributions over labels for all datasets while it tries to minimize only the classification error associated to the known label. This way it learns all of the distinctive features from all available datasets, while the generator learns to control all the labels in both datasets.

5. Results

We hereby present the results obtained using our self trained model. For illustrative purposes we will then present the results and analysis obtained using the pretrained model. Finally, we will compare the results with the results from the baseline models.

5.1. Results from self-trained model

Following are the results obtained by training the model on local machine. These trainings were carried out using different number of training examples and iterations as shown in captions below the following images.

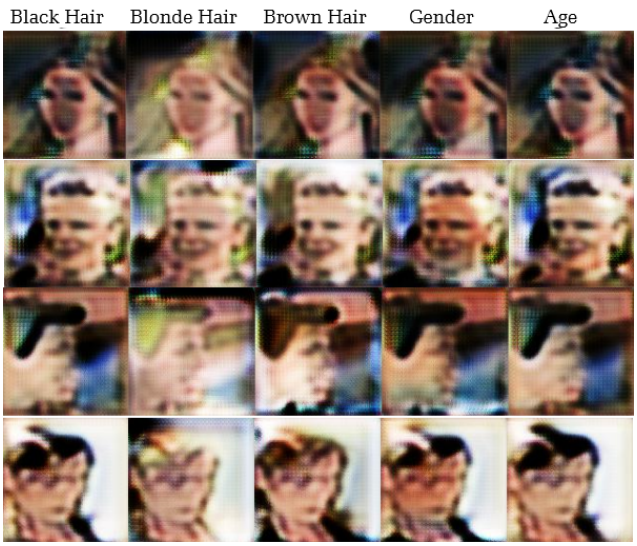

Figure 5. *Training images= 4000, Total Iterations = 2000*

Clearly, these images are very blurry and distorted, but they are gradually moving towards the target domain. So this validates that given sufficient training resources, the model will give accurate better visual results. Slightly improved results obtained after over 6 hours of training are hereby presented.

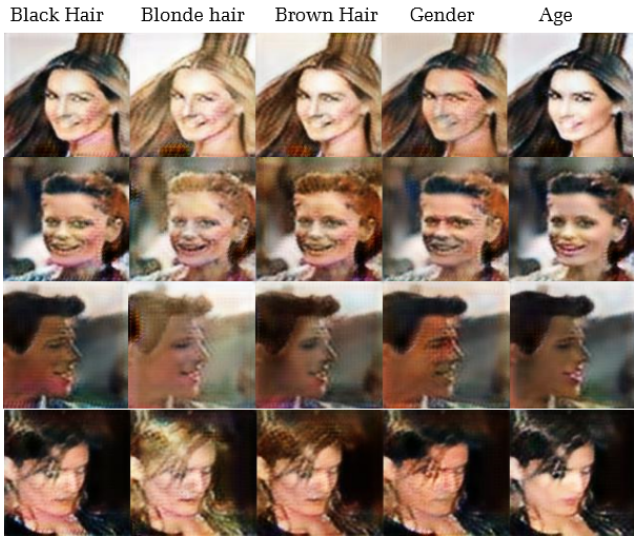

Figure 6. *Training images= 10000, Total Iterations = 20000*

5.2. Results from pre-trained model

Following are few results obtained by running the pre-trained model. Clearly, since it is a fully trained model, the results have high superior quality and show the actual power of this StarGAN model.

Figure 7. *Sample results from pretrained model*

5.3. Comparison with baselines models

The following section shows how StarGAN gives clearly better results than prior results. These have been presented from existing literature [2].

5.3.1 Results generated from celebA

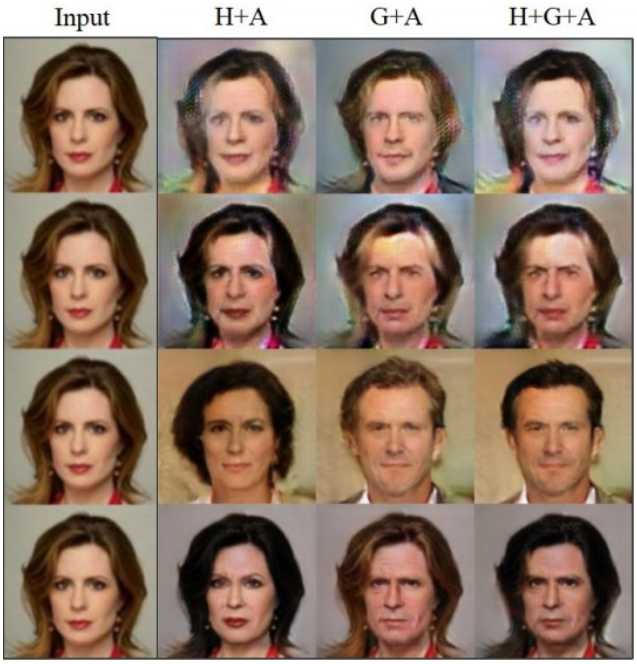

Figure 8. *Sample from training on CelebA of changing to target attribute Hair-Gender, Gender-Age and Hair-Gender-Age. Results show the comparison between DIAT, CycleGAN, IcGAN and StarGAN in that order. [2]*

These are the comparative visual results for Gender and Age change translation. We can clearly see that starGAN outperforms all the previous models. The reason is that, instead of training a model to perform only a fixed domain translation (e.g., black to blond hair and vice versa), which is prone to over-fitting towards only two type of translations, model is trained to transform image of given domain to any other domain equally. This brings in the implicit regularization effect and qualitative test results for starGAN. Further more, IcGAN is able to translate images directly from one domain to any other domain, but starGAN outperforms it by storing spatial information of latent space by using convolutional layer as latent layers followed by activation map functions, whereas IcGAN uses low-dimensional latent vectors.

5.3.2 Results generated from RaFD

Looking at the results shown below, we clearly see that our model Visually most natural-looking expressive images. Not just that but starGAN also maintains the personal identity properly and keeps the sharpened facial features of input

image. Cross domain models introduced earlier are showing blurry results and Also when it comes to sharp features of the image they fail to preserve them, while starGAN is able to do so. Lets understand the reason by an example: 2000 images for each domain and there are total 8 domains, Hence total number of images = 16,000. When we train 2 domains, Cross-domain models can only use 4,000 training images simultaneously, whereas out model can use 16,000 images during training as we saw earlier. This is the major reason that model can learn maintaining content and sharp features of the actual images properly.

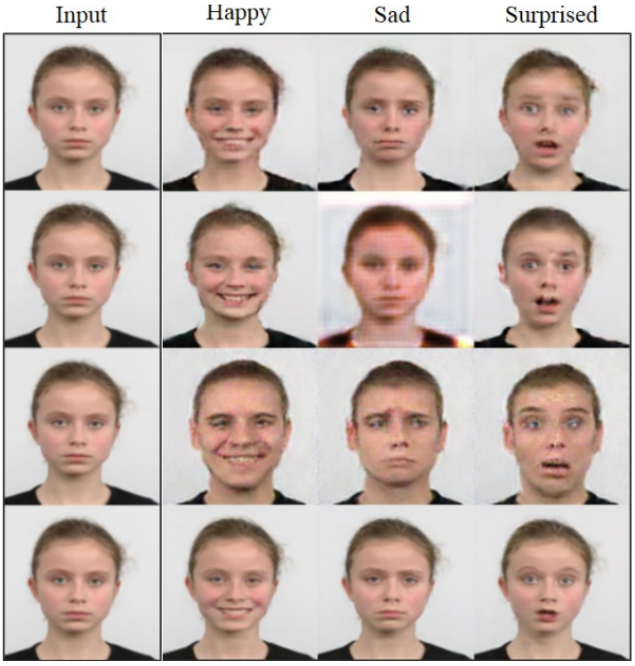

Figure 9. *Sample from training on RaFD with target domains Happy, Sad and Surprised. Results show the comparison between DIAT, CycleGAN, IcGAN and StarGAN in that order. [2]*

6. Limitations of StarGAN

StarGAN is a novel model for multidomain image to image translation, however it still has certain drawbacks. StarGAN tends to make unnecessary changes during cross-domain translation. For e.g alters the face colour, Unnecessarily changes the background etc. It fails to competently handle same-domain translation. For e.g adds a moustache to the face, adds extra hair etc. Methods used in StarGAN assume binary-valued attributes and thus cannot yield satisfactory results for fine-grained control. Methods used in StarGAN require specifying the entire set of target attributes, even if most of the attributes would not be changed. Using binary valued attribute results in StarGAN changing certain attributes of an image which were not supposed to change.

6.1. Rectifications in StarGAN

The problem of handling same domain translation could be rectified by introducing an additional identity loss equation in the loss of generator which could be given as $L_{id} = E_{x,c}[\|G(x,c) - x\|_1]$ if $c = c_x$; 0 otherwise. Complete loss functions are as follows: [7]

$$\mathcal{L}_{adv} = \sum_{c\in\{c_x,c_y\}} \mathbb{E}_{x,c}\left[\log\left(1 - D_{r/f}(G(x,c))\right)\right] + \mathbb{E}_x[\log D_{r/f}(x)]$$
$$\mathcal{L}^r_{domain} = \mathbb{E}_{x,c_x}\left[\log D_{domain}(c_x|x)\right]$$
$$\mathcal{L}^f_{domain} = \sum_{c\in\{c_x,c_y\}} \mathbb{E}_{x,c}[-\log D_{domain}(c|G(x,c))]$$
$$\mathcal{L}_{cyc} = \sum_{c\in\{c_x,c_y\}} \mathbb{E}_{x,c_x,c}[\|G(G(x,c),c_x) - x\|_1]$$
$$\mathcal{L}_{id} = \mathbb{E}_{x,c}\ [\|\boldsymbol{G(x,c) - x}\|_1] \textbf{ if } \boldsymbol{c = c_x}\textbf{; 0 otherwise}$$

In order to avoid the extra unnecessary changes in the image translation, instead of using binary values for target attributes, using relative attributes improves the result as shown in the images in the below sub-sections.

6.2. Comparison of Rectified results with StarGAN

Following results presented in existing literature shows how the limitations of StarGAN overcome by other models.

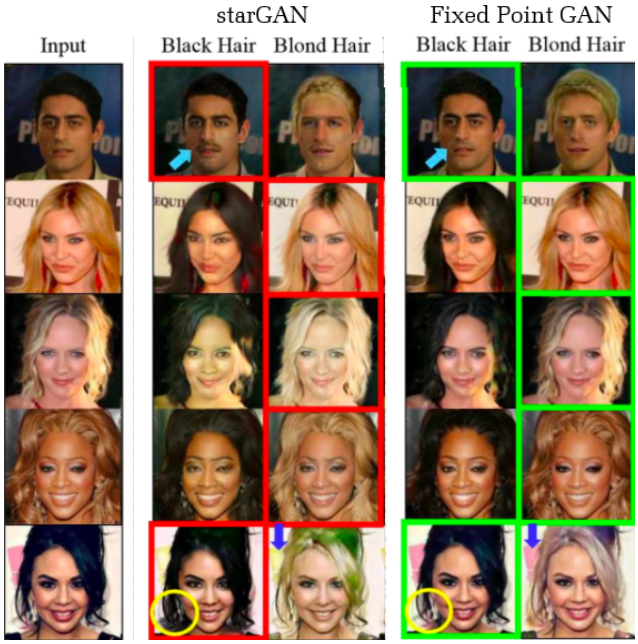

Figure 10. *Comparison of results of StarGAN and Fixed-point GAN[7]*

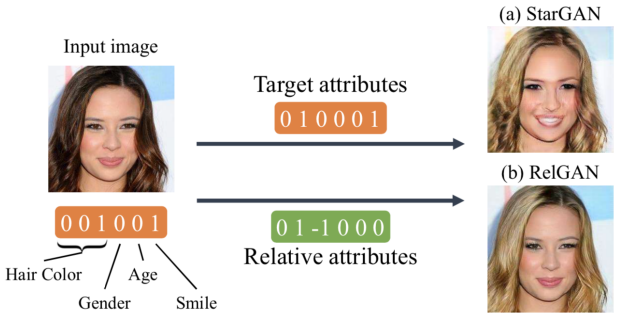

Figure 11. *Effectiveness of using relative attributes in Rel-GAN [8] instead of binary attributes used in StarGAN*

7. Conclusion

Although there existed approaches for image to image translation, they only translated within two domains and hence gave blurry distorted results due to inefficient training. The novel approach of StarGAN proposed here works for multi-domain translation as we saw. The real power of this model comes from its joint training strategy where it learns different attributes and features from multiple datasets simultaneously using the idea of mask vector. Hence, it gives superior results compared to any other previous models. Thus, StarGAN is an extremely scalable model with high visual quality generated image because of it's capability of multi-task transformation. Yet, there are some flaws addressing which researchers can develop superior image translation applications across multiple domains.

8. Contributions

Following are the individual contributions of each of the team member. Apart from that, the final report was compiled as a group.

Ruchit Shah (201701435) - Proposed the devil's advocates arguments by showing inferiority of StarGAN as comapred to Fixed-point GAN and RelGAN. Worked on pre-processing code for images, train-test functions and generated outputs from the custom code and pretrained model.

Darshan Patel (201701436) - Proposed the advocates arguments by emphasizing on joint training and concept of mask vector. Worked on the code for building the model and commented the entire code.

Niharika Dalsania (201701438) - Proposed the model architecture and later summarised the major takeaways of the project by including the advocates and devil's advocates arguments to highlight the power and shortcomings of the model. Worked on building the model architecture and related utility functions.

Zeel Patel (201701443) - Proposed the advocates arguments by showing various loss function that lie beneath the actual power of this model. Worked on coding the loss functions.

References

[1] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," 2018.

[2] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," 2017.

[3] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2020.

[4] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional gans for image editing," 2016.

[5] M. Li, W. Zuo, and D. Zhang, "Deep identity-aware transfer of facial attributes," 2018.

[6] *https://jonathan-hui.medium.com/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490*.

[7] M. M. R. Siddiquee, Z. Zhou, N. Tajbakhsh, R. Feng, M. B. Gotway, Y. Bengio, and J. Liang, "Learning fixed points in generative adversarial networks: From image-to-image translation to disease detection and localization," 2019.

[8] P.-W. Wu, Y.-J. Lin, C.-H. Chang, E. Y. Chang, and S.-W. Liao, "Relgan: Multi-domain image-to-image translation via relative attributes," 2019.