2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT) | 978-1-6654-5126-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/IAICT55358.2022.9887491

Using Generative Adversarial Networks for Conditional Creation of Anime Posters

Donthi Sankalpa
Computer Science and Engineering
American University of Sharjah
Sharjah, United Arab Emirates
g00062902@aus.edu

Jayroop Ramesh
Computer Science and Engineering
American University of Sharjah
Sharjah, United Arab Emirates
b00057412@aus.edu

Imran Zualkernan
Computer Science and Engineering
American University of Sharjah
Sharjah, United Arab Emirates
izualkernan@aus.edu

***Abstract*—Japanese animation, known as anime, has become one of the most accessible forms of entertainment across globe. Recent advances in generative adversarial networks (GAN) and deep learning have contributed greatly to multiple interesting applications in the domain of anime, particularly in face generation, style transfer, and colorization. However, there are no existing implementations for generating composite anime posters with a genre accompaniment prompt. This work proposes a novel application of genre to anime poster generation conditioned on BERT-tokenized binary genre-tags of *light-hearted* or *heavy-hearted* categorized based on the thematic subject content of the medium. A dataset of 9,840 image with genre tags and synopses was constructed by scraping MyAnimeList. The conditional Deep Convolution GAN with Spectral Normalization produced the best posters, achieving the quantitative scores of FID: 90.17, average IS: 3.505, 1KNN with PSNR: 0.445 across inter-label discernability, and FID: 166.4, across genuine versus generated poster distinguishability. The primary contribution of this work is to present results outlining the feasibility of various GAN architectures in synthesizing controllable and complex composite anime posters. The larger implication of this project is to provide an introductory approach showing the promise of a creativity assistant for authors, artists, and animators, where they can simply enter a key phrase representing a concept they have in mind, to generate a baseline idea as an initial phase.**

***Keywords*—Anime, Computer Generated Art, Deep Learning, Generative Adversarial Networks, Image Generation**

I. Introduction

Japanese animation, known as anime, has become one of the most accessible, forms of entertainment across globe. The dynamic, kinetic and colourful style of anime combines elements of traditional cartoons with the historical artistic styles of Japan and makes its foray into mainstream popular culture. Advances in machine learning and deep learning with regards to computer vision primarily have contributed to various interesting applications in the domain of anime. In specific, the advent of generative adversarial networks (GANs) has enabled the possibility of utilizing computers to synthesize art in a manner mirroring the subjective creative process of artists. Previous work in the animation domain can be categorized into sketch colorization, face portrait generation, and recommendation of titles based on user specified parameters. Some of the recent works exhibiting considerable success are in face generation [1], facial expression transfer [2], posture generation [3] style transfer [4], manga style superimposition [5], and line art colorization [6]-[7]. There are also interesting approaches made available via open source projects like [8] and [9] that generate random characters using tags. A few of the main challenges in this area stem from the complex nature of anime objects, where high quality stylized semantics are preserved, but separating fine features is difficult [10]. Landscapes, characters, vehicles, buildings, and varying plethora of facial expressions across different anime posters, scenes or movie stills make it difficult to capture a unified representation. As can be observed from [10], where attempts to generate an opening theme frame of an anime episode managed to capture the colour content which vaguely resembled anime, but no fine features were represented at all. Feature extraction, segmentation of image aspects and feature translation due to the varying qualities of anime in general remain an issue. While genre prediction of movie posters/covers have been found in literature, conditioned generation of such content yet remains to be introduced. Since posters are not entirely uniform but rather complex images with humans, objects, landscapes and text, it poses an interesting and worthwhile challenge to undertake. Thereby the goal of this research is to implement a GANwhere users can type in a short description or indicate a genre label pertaining to any anime scenario or thematic content and choose between genres, to generate a composite poster.

The contribution of these experiments are as follows.

1. An initial study of GAN applicability in synthesizing controllable anime posters.
2. Provide benchmarks with multiple GAN architectures for generating conditioned anime posters.

The wider implication of this project is to present a creativity assistant for authors, artists, and animators, where they can simply enter a paragraph summarizing a concept they have in mind, and an appropriate scenery is generated. This process can be to provide them with an initial idea to start from, creative inspiration, or show them how a text-summary is likely to manifest as in the form of an image.

This paper is organized as follows: Section II explores the related work; Section III discusses the methodology; Section IV discusses the results and Section V concludes the work and suggests possible future research directions.

II. Literature Review

Scribble colour art based line art colorization is a challenging computer vision problem because lines do not necessarily add semantic information and could merely be an artistic choice. [1] integrates a Wasserstein GAN (WGAN) with gradient penalty with conditional adversarial network to synthesize colorized images that are more realistic and natural given colour hints from the users. They web crawled and constructed two datasets with 21,930 and 2,779 images respectively, one for illustration and one for authentical line arts. Images in both datasets were croped to 512 x 512 pixels with random horizontal flipping for augmentation. The generator used is based on the U-Net architecture, which is essentially an encoder decoder architecture with residual connections, except they use ResNeXt blocks instead of

ResNet. In addition, they remove all normalization layers, utilize sub-pixel convolutions to increase resolution, and added dilation to expand the receptive fields within the ResNeXt blocks. The discriminator was based on SRGAN, modified to form a condition GAN and stacked with more layers so that 512 x 512 images could be processed. The GAN layers and parameters were 250,000 training steps, batch size of 4, Adam optimizer learning rate of 0.0001. The adversarial loss was replaced with the Wasserstein approximation with a gradient penalty and perceptual loss to preserve the colour content, which was obtained using the 4th convolutional layer in the VGG16 network.

In unsupervised image translations tasks, mapping local texture is successful, but anime mapping to human faces, or cats to dogs where the global shapes vary face challenges. Usually, image cropping and alignment among other pre-processing techniques are needed to limit the complexity of data distributions. The authors of [11] implement an end-to-end image to image translation by adding an attention module and learnable normalization function called Adaptive Layer Instance Normalization, the areas of interest can be focused on. Selfie2anime consisting of human faces and anime faces was used. All images are resized to 256 x 256 pixels for training. They used two generators and two discriminators, one for local and one for global features. The generator consisted of an encoder and encoder component, with an attention feature map guiding the information about the critical regions from the discriminator to the generator. The encoder of the generator had two convolutional layers with the stride size of 2 for down sampling and four residual blocks. The decoder of the generator had four residual blocks and two up sampling convolutional layers with stride 1. Encoder used the instance normalization, where the decoder used ADALIN. The activation functions were ReLU followed by tanh in the final layer. The discriminators were based on the PatchGAN architecture for local and global features extraction and classification, with Spectral Normalization and Leaky ReLU in all layers except the last layer. The training parameters were 500,000 iterations with Gaussian weight initialization. And learning rate of all models was 0.0001. The loss functions used was adversarial loss with cross entropy, cycle consistency loss and identity loss ensure the colour transfer between the source and the target was preserved.

[12] is one of earliest works to address the mitigation of common problems such as poor image quality, low diversity generated samples, and slow model convergence to generate animated images. They constructed their own dataset by *scraping* 15,000 images from *Google*. They used a slightly modified Deep Convolutional GAN with 1 fully connected layer, 3 transpose convolutional layers with batch normalization, ReLU activation and tanh activation on the final layers. Finally, Four convolutional layers, and one full connection layer, with batch normalization and Leaky ReLU activation, and sigmoid activation on the final layer.

[2] explored facial expression transfer for anime images, in the same domain given an input image using the StarGAN architecture and achieved promising results regarding the manipulation of the facial emotion latent variables using a conditional label. The confusion matrix reveals correlation between sad, crying and neutral expressions. The model converted a given image to happy with the most certainty. They leveraged the StaGAN architecture with 1) Generator using 3 convolution layers for down-sampling, 9 residual blocks, and 3 transpose convolutional layers for up-sampling. 2) Discriminator using PatchGAN based architecture, which classified local patches independent of faces of real or fake and determined the expression using the Auxiliary Classifier (AC) GAN extension from the last discriminator layer. The losses considered were domain classification loss to maintain the anime information between input and output and reconstruction loss represents how successful the generator was in reconstructing the image from the translated one.

[13] performed automatic generation of anime characters without blurred or distortion which was a challenge, and especially when incorporating conditions. This is one of the first works to include conditions in anime portrait synthesis. They collect images from *Getchu*, where there are characters with an accompanying description. After filtering, 42,000 images in total were compiled. They used illustration2Vec for estimating tags of anime illustrations. All images are resized to128 x 128 pixels. Conditions were hair, eyes, mouth and facial expression. Accessories such as glasses, and hat were also included within the eyes and hair conditional latent space representations, and this is empirically validated by stochastic variation of the conditioned noise along different ranges. The generator was based on the SR-ResNet, and contained 16 ResBlocks with 3 sub-pixel CNN feature maps. Discriminator had10 ResBlocks, with no batch normalization. Weight Initialization from Gaussian distribution and the batch size was 64.

As put forth by [4], anime face translation is challenging due to complex variations of appearances among anime faces. This method preserved the global structure of the source and eliminated noticeable artifacts and distortions in the local shapes of generated images. The Generator consisted of content encoder, style encoder and a decoder. The encoder encoded the content, and style encoder extracted the style from the reference image. The decoder constructed an image using the content code and style code, where the local facial features were transformed while maintaining the global structure of the source. To this end, they used an adaptive stack convolutional block, fine-grained style transfer block, and two normalization functions for the content encoder, style encoder and decoder respectively. The discriminator was expanded to a double branch discriminator, as they operated under the assumption that anime faces and photo faces shared common distributions, and this implies meaningful facial information. Therefore, the discriminator consisted of a shared shallow layers that received both anime and human faces, and then domain specific feature extraction layers. This method was hypothesizing to individually learn domain specific features while preserving the global features common to both domains in the shallow/coarser layers. The loss functions used were hinge loss with gradient penalty regularization is the adversarial loss, feature matching loss for global feature preservation in the shared layers, domain aware feature matching loss for artifacts reduction in the final specific layers and reconstruction loss for maintaining global semantic structure.

The authors in [14] propose an Attention-GAN (AttnGan) to generate birds based on their descriptions. They train their GAN on the two most used bird database, namely, the

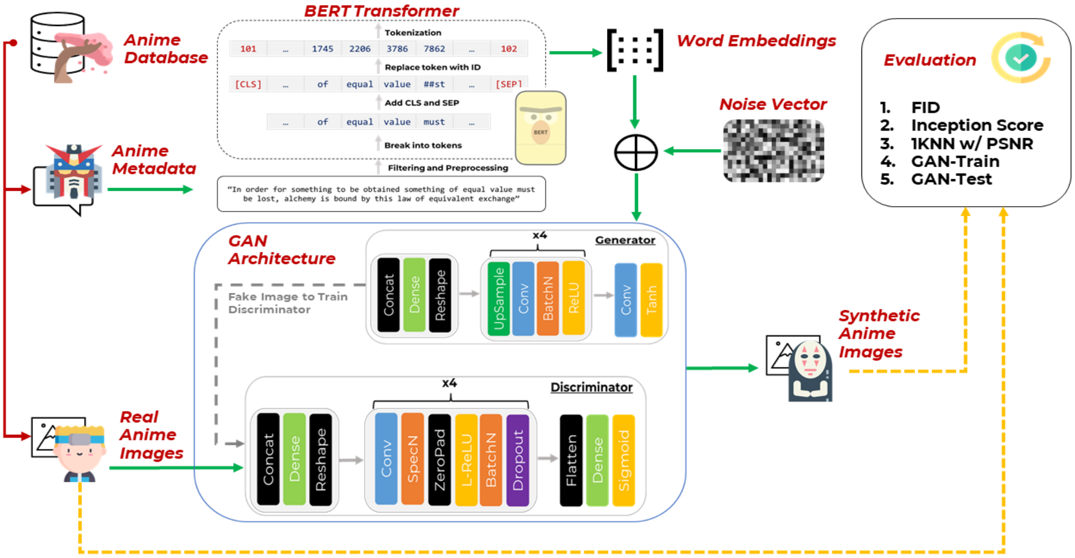

Fig 1. End-to-end dataset to evaluation procedure

Caltech-UCSD Bird (CUB) database that consists of 8855 training images and 2933 images for testing of 200 classes of birds and each has 10 descriptions and the MS COCO dataset that consists of 82783 training images for training and 40504 images for validation along with 5 descriptions for each bird. They proposed two systems, the first being a GAN that is in charge of the generation and a image-text similarity model. The input to the GAN includes the description embeddings that were generated using conditioning augmentation. The GAN consist of m number of generators with m hidden states before it. The Attn model takes in the word features and image features from the previous hidden layer. the two features are then converted into a common semantic space with a perception layer. Next a word-context is computed for each sub-region based on the hidden features. Finally, image features and word features are combined to generate images. A CNN was used to create a model for checking image-text similarity. It is used to map images into semantic vectors. It is trained on inceptionV3 with ImageNet weights. Images are rescaled to 299x299 for this. It extracts the local feature map from inceptionV3 mixed_6e (768x17x17) where 768 is the number of dimensions of local region. There are 289 subregions. The global feature vector is extracted from last average pooling layer with a perception layer that is used to convert image features into common semantic space of text features. The losses used are adversarial loss for the generator and cross entropy loss for the discriminator. Additionally, a DAMSM loss is used. It is designed to learn the attention model in a semi-supervised model (supervision is only full sentence match with full image). It is the sum of loss of each of the words which is minimized posterior probability that sentence matches with corresponding image. the authors used inception score as their metrics.

Another instance is where parts of the AttnGAN were used to build a variant of the original model over it. The authors Zhang et al. of paper [15] propose a cooperative up-sampling method to further improve the AttnGAN. The authors use the CUB dataset as well as the Oxford-102 dataset which consists of 7043 training images from 82 species of flowers and 1155 testing images from 20 other flower species with 10 descriptions each. The authors used the pre-trained LSTM text encoder from the AttnGAN to create the embeddings. As the latent space for text embeddings is high in dimensionality, they applied conditioning augmentation (CA) from StackGAN to produce a gaussian distribution over the latent representation. They had 256 dimensions over sentence level features. The architecture uses two Generators, object generator Go and background generator GB. Input to Go is all of the embedded text, input to GB is with partial dimensions removed through a dropout layer. Cooperative upsampling is used to share and upsample the features from both the generators. The low-resolution vector is split in two parts, one produces a LR object OL and mask ML, the other is sent to AttnGAN to combine with word-level features. The final image is generated using object, mask and background. The authors use three discriminator, one high level discriminator, one low resolution discriminator and a background discriminator. The resolution discriminators take in only sentence level vectors. The authors apply the log losses and the DAMSM loss from AttnGAN. For evaluation, they used the Inception score, R-precision, subjective user evaluation, with comparison to baseline and failure case analysis.

III. Methodology

As illustrated in Figure 1, the end-to-end pipeline process consists of 5 core steps. First, a compendium of original anime poster images and its associated metadata in terms of descriptions and genre-tags, was collected and compiled into a dataset. Secondly, the metadata was preprocessed and fed into a language transformer to derive embeddings. Thirdly, a subset of these embeddings was meshed with random noise and passed along with its respective images into a GAN model for training. Then, through exhaustive experimentation, the best performing GAN architecture was identified. Lastly, the results of the GAN model were evaluated with the standard metrics for this domain using the remaining subset of testing embeddings and its images.

A. Data Collection

The dataset for this work was constructed using metadata of hyperlinks acquired from the *Kaggle* dataset utilized for user rating score prediction [16]. The UIDs pointed to records in *MyAnimeList*, which is an online database of organized anime and manga with user scores. It is reported to have 4.4 million anime and 775,000 manga entries. We scraped the list of popular anime from the last 30 years with the *BeautifulSoup* library [17]. Initially, this yielded a total of 13,000 records. After filtering for duplicates, incompletes (i.e., image but no description, or vice versa), and removing invalid (grey images with NO IMAGE FOUND watermarks), our final dataset had 9840 images with associated genre tags and descriptions/synopses. All images were resized to a resolution of 256 x 256 x 3. Normalization with 255.0, ad 127.5-1 scaling yielded pixel value ranges in [0,1] and [-1, 1] respectively. The latter scaling was observed visually to produce better results. We categorized two classes, *light-hearted* and *heavy-hearted*, based on the leading tags of "fantasy, comedy, school, slice-of-life" versus "action, drama, horror, thriller". The genre-tags and the accompanying text descriptions were subject to and following preprocessing methods in order: 1) non-ASCII (tags and HTTP codes) character removal, 2) Stop words removal, 3) Lemmatization, 4) Punctuation removal. 5) Trailing and repeated spaces removal, 6) Placeholder text removal, 7) Tokenization with class and separator tags, 8) Uniform padding to length of maximum. The small, uncased BERT [18] model, which does not differentiate between uppercase and lower cases, was employed to convert the preprocessed tags/descriptions into embeddings. The embeddings are a semantic representation in the form of a vector with 768 dimension.

B. Models

The following models were used in this work and are outlined in terms of their key architectural design variations. The key layers composing the models are depicted in Fig.1.

Conditional DCGAN (CDCGAN) with spectral normalization: This involves additions to the vanilla GAN architecture in terms of the use of strided convolutions/transpose convolutions instead of pooling layers, the use of batch normalization for both generator and discriminator, the use of ReLU and tanh in generator and Leaky ReLU in the discriminator to get a DCGAN [19]. Additionally, Spectral normalization (SN) is a technique used to stabilizes the training process by constraining the Lipschitz constant associated with the layer weights of the discriminator and reduces the impacts of exploding gradient as well as mode collapse [20]. Essentially, the weights are bounded at 1 every time the weights are updated. Since this is a computationally expensive process, the power iteration algorithm is used to approximate the updated eigen values. This is an enhancement to our baseline models, with the addition of the concatenated conditional label.

Conditional LSGAN (CLSGAN): The discriminator final layer activation layer of sigmoid is removed, while the binary cross entropy loss is replaced with mean square error. The key idea in this change is to provide smoother and non-saturating gradients, as MSE is more tolerant than BCE in terms of the gradient information propagated to generator for updating its weights. BCE accounts for right or wrong, but MSE will penalize images based on their distance from the arbitrary decision boundary, thereby providing a resolving vanishing gradient [21].

InfoGAN: The motivation behind using this model is to disentangle and control the individual features of the generated images. By separating the auxiliary information, they become control variables which can elicit different properties from the latent space. In the previous architectures, we directly concatenate the labels or embeddings with the noise, where they come entangled with the latent manifold representation of images. This is undesired as smaller dimensional information such as labels may get ignored, or higher dimensional information such as our text embeddings be too disparate in relation to the images. The key idea in this architecture is mutual information, referring to the amount of information that can learned about one variable (embeddings, labels), given the image generated using noise and control variables. The mutual information loss minimizes the conditional entropy of the control variables, calculated as the difference between joint probability of generated image and control variables and marginal probability of control variables. This in theory, should disentangle our labels, and the embeddings from the images, and conferring some notion of independence to them. In terms of implementation, we simply take extend the feature extraction layers of the discriminator and predict the control variable as an additional output [21].

AC + InfoGAN: By adding an additional classification layer to the discriminator to minimize the error of classification alongside the standard adversarial loss, we can utilize the utility of labels to generate our *heavy-hearted and light-hearted classes.* This proved to generate better images than Info-GAN only [22].

WGAN: Wasserstein GANs use an approximation of the Wasserstein distance instead of the standard adversarial loss. It is a mathematical representation of the Earth Mover distance, which quantifies the minimum cost for transporting mass of one distribution given to another distribution. In terms of GAN applicability, it is stated that gradients during learning are smoother and mode collapse is reduced. One of the known issues is the difficulty in enforcing the Lipschitz constraint, i.e., selecting optimal parameters to bound the derivate range, to train discriminator). If bounding parameter is large, it takes a long time for weights to approach limit, and if it is small vanishing gradients occur. The original paper used weight clipping, which was later shown to be not very effective in terms of convergence. An addition of the gradient penalty was proposed, which constraints the Lipschitz by penalizing only if the gradients diverge from the target value of norm at most 1 almost under the two distributions. Batch normalization was removed for the discriminator and replaced with Layer Normalization [23].

IV. Results

All models are trained in the following experimental scenarios: 1) Images with Genre-tags, 2) Images with Descriptions, and 3) Images with Genre-tags and Descriptions. The best performing model was CDCGAN with SN, when the binary genre-tags of *light-hearted* and *heavy-hearted* were used and its results are depicted in Figure 2. They were obtained after approximately 39 hours of training and was stopped when the improvement in FID score became negligible and the generator and discriminator loss stabilized in the range ~0.25±0.07 and ~3.00 of ±0.19 respectively. The results from the other models when using only genre-tags is shown in Figures 3 and 4. Note that only

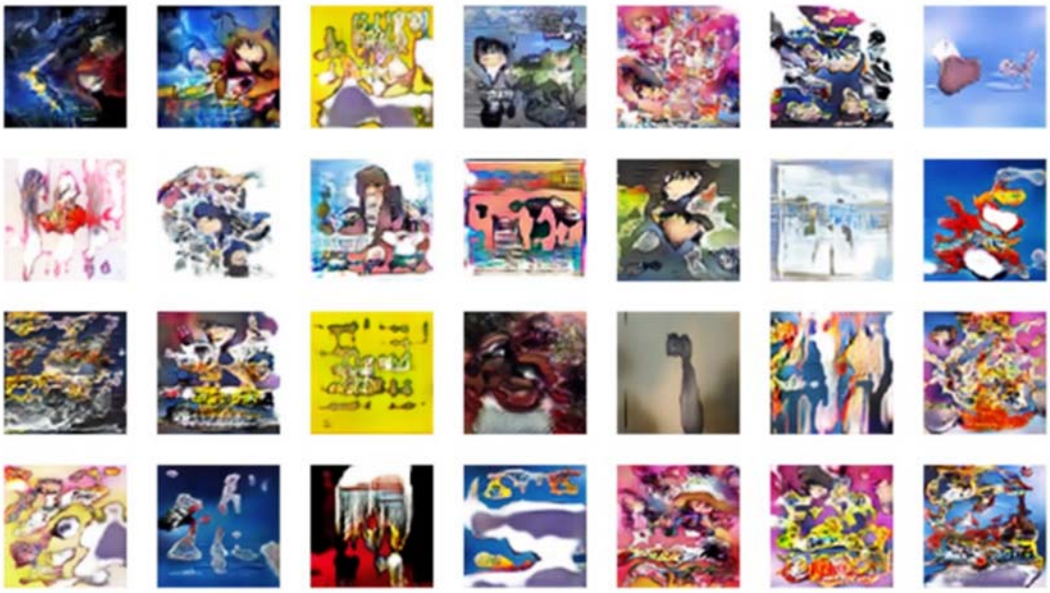

Fig 2. Posters generated in the last epoch by the best model

AC+InfoGAN synthesized images using both genre-tags and descriptions, as the other models did not achieve relatively satisfying outcomes in this experimental scenario.

The WGAN captured the textual positioning reasonably well but failed at the other aspects of the poster composition. Noticeably, the AC + InfoGAN model gave a result almost like the best model mentioned previously but required almost twice the training time. The InfoGAN exhibited a tendency to create posters with big spots of white areas in the image, while on the other hand, the CLSGAN created darker images. The training parameters were varied as follows in Table I.

TABLE I. PARAMETERS UTILIZED FOR THE TWO MODELS

Batch Size	***Optimizers***	***Learning rate***	***Depth***
16-64	ADAM, RMSPROP, SGD	0.0002 *beta_1*=0.5, *beta_2*=0.999	Up to ~20 layers

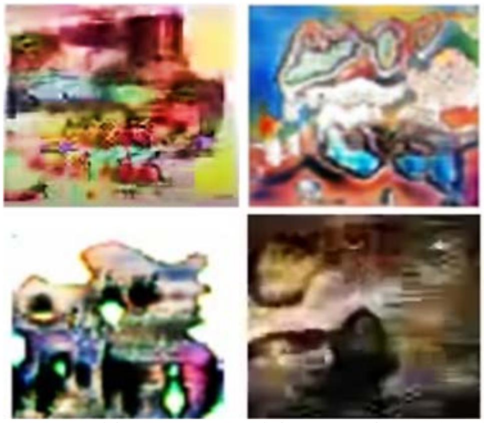

Fig 3. Posters generated from WGAN (top-left), AC + InfoGAN (top-right), InfoGAN (bottom-left) and CLSGAN (bottom-right)

A. Evaluation

Table II shows the results of the metrics applied on the two (relatively) best models. The evaluation metrics used in accordance with recent literature are the Fréchet Inception Distance (FID), Inception Score (IS), 1K-Nearest Neighbour with PSNR (1KNN w/ PSNR) and GAN-Train/GAN-Test. Their adaptation for this work is briefly explained as follows:

In the case of FID, lower values indicate better quality of generated images. This metric quantifies the distance between two distributions of images. A pre-trained Inception-v3 model is loaded without the head component, where is the classifier and output are taken from the last pooling layer. this pooling layer has 2048 activations which indicate that each image is predicted to have 2048 activation features. Next the F-distance is calculated using the 2048 features of each group of images, which can be interpreted as ascertaining if the features of real images are close the features of the fake images.

In the case of IS, higher values indicate better quality of generated images. This metric is used to check for image quality and diversity across classes. It has a lowest value of 1.0 and highest value of the number of classes supported by the classification model. Like the previous approach, the pre-

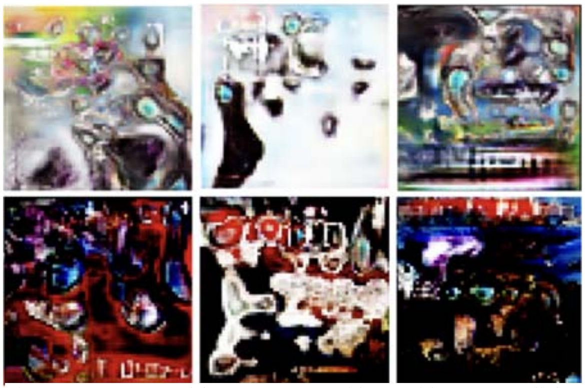

Fig 4.Posters generated from CDCGAN with SN with genre-tags light-hearted (top-row) and heavy-hearted (bottom-row)

TABLE II. QUANTITATIVE RESULTS FOR ANIME POSTER GENERATION FROM 2 BEST MODELS

Model	***FID***	***IS***	***1KNN w/ PSNR***	***GAN-Train Accuracy***	***GAN-Test Accuracy***
CDCGAN with SN	*Genre-tags*= 90.17 *Real vs Fake* = 166.40	*Light-hearted* = 3.56 *Heavy-hearted* = 3.45	0.445	Train = 45.44% Validation = 50.00%	Test = 50.4% Validation = 70%
AC+InfoGAN	*Genre-tags* = 23.92 *Real vs Fake* = 224.12	*Light-hearted* = 2.53 *Heavy-hearted* = 2.48	0.43	Train = 45.68% Validation = 52.20%	Test = 50.24% Validation = 65%

trained Inception-v3 model is used to compare each label distribution against the marginal distribution, and the ideal case elicits a high KL divergence. If it is medium, it means that there is good diversity but bad specificity with respect to labels, and low indicates there is no diversity or no distinct labels or even both.

The 1-KNN w/ PSNR method allows for checking intra-label diversity. This metric trains KNN of n=1 with real and fake images. It uses a leave one out (LOO) algorithm for the KNN. The distance is calculated using PSNR between the points as the points represent images. The best score is a LOO accuracy of approximately 0.5 as that shows the two distributions are very similar. That is, the real and fake images are clustered around each other with a proper amount of distance to show a proper amount of diversity. If the LOO accuracy is lesser than 0.5 then the GAN is overfitting towards real images as the generated as the generated images are very close to the real samples. If the LOO accuracy is greater than 0.5 then that means the two distributions are separable and that is bad as it goes against the goal of a GAN that is to generate real like fake images.

Finally, the GAN-Train/GAN-Test was performed. For this metric the process was divided into two parts: 1) GAN-Train: this is where we train a classifier on generated images and test then on real images, and 2) GAN-Test: this is where we train a classifier on real images and test them on generated images. Transfer learning was leveraged with the pre-trained MobileNet and ImageNet weights to load the model without the head component and attached it to a dense layer for classification. A GAN is said to be imperfect when the GAN-train accuracy is lower than the validation accuracy. This could happen because mode dropping reduces the diversity of the generated images, generated samples are not realistic enough or the GAN has mixed up classes and is confusing the classifier. If the GAN-test accuracy is higher than the validation accuracy, it shows that the GAN is overfit and is simply memorizing the training set, on the other hand, if the GAN-test is significantly lower than the validation accuracy, it shows that the GAN is not capturing information well and the images generated are of poorer quality.

Both the models performed similarly except with a marginal difference in performance. For both the models, we can conclude firstly, from the FID that the real and fake images were far apart, but the two class labels were close to each other. Ideally, we would like this to be the opposite. The primary reason could be is because the generator was quite confused between the two labels and hence generated some overlapping images between the two classes. Secondly, since the inception score is medium, we can say that there is some good diversity but bad specificity. This additionally suggests that the generator is in fact getting the two class labels mixed up. Thirdly, both models had a 1KNN accuracy of slightly lower than 0.5, which means that the fake and real image distributions were slightly similar but yet had a chance of overfitting. Lastly, from the GAN test we can see that the GAN test accuracy was much lower for CDCGAN and slightly lower for AC + InfoGAN than the validation accuracy and hence shows that the GAN does not capture the target distribution to well. The GAN train accuracy for both the models was weaker, this could be because the generated samples are not realistic enough or the GAN mixed up classes. Further reasons will be elucidated in the next subsection.

B. Discussion

The disentanglement in the auxiliary information made possible by the architecture of (ACGAN+InfoGAN), provided the best for the multimodal case with both genre-tags and description/synopses. As for only using genre-tags, which yielded the most realistic images relatively in the CDCGAN, spectral normalization helped in constraining the Lipschitz constant, thereby making the likelihood of mode collapse lower. We hypothesize that there is some inherent inclination of the feature extraction layers to focus on colour/textures, more so than linear design. It is evident from our results that no outlines, or shapes were produced in any of the images, and any discernible shape is merely colorized blobs. This is consistent with the findings of [24], where the researchers utilized style transfer to create the Stylized Image Net and noted that ImageNet CNNs were "lazy" and tended to focus on textures only. In hindsight, another problem could be the fact that anime in general tends to be more heterogenous when compared to human faces in the CELEB set or the objects in the CIFAR set. This is because the artistic style of anime varies more than just watercolour, as different artists, across different genres, tend to have their own unique look. Additionally, there are clear differences between anime produced in the 90s vs 2000s vs 2010s. Also, the overlapping classes i.e., action, fantasy are different from each other. Observing the dataset, we noticed that not all were posters, because there a mix of individual characters, characters scenery, only scenery, title only with coloured screen, title with additional taglines, or staff names, studio logos. Diverse and overlapping not enough of each sample to discriminate and learn well. Because both images and text, there seems to be a sort of warped nature to the generated images, due to difficulty in separating. Mathematically regularity in texts could be captured, but finer details such as facial features were never effectively captured. Another open-source project tried to generate album art covers given lyrics, which had a more expanded dataset, and ran for longer, but still was not of high quality. It is fairly an easy task to discern an artificial attempt at an album cover [25]. Two key techniques stood out here, which is the content loss which preserves the image during style transfer, and the "edge" promoting loss which to boost the outline definition of the anime versions.

Ultimately, the GANs capture the general appearance of the anime in terms of color and theme, but do not produce satisfactory finer, granular details. We purport that based on empirical experimental results, this is because of batch normalization mitigating the likelihood of mode collapse, Leaky ReLU preventing vanishing gradients and SN conferring stable convergence. The AC+InfoGAN appears to be producing relevant images, and this is likely because This architecture disentangles both the labels and the embeddings from the image, while keeping the DCGAN architecture with SN as its foundation. The WGAN gives some notion of colour and position with respect to the title text, when using genre-tags. Mode collapse, and the inapplicability of the Wasserstein metric for our images could be the likely scenario for failure in the scenario with descriptions. CLSGAN and InfoGAN did not produce images of quality, which we posit is due to the unsuitability of the loss functions for this specific task, and their tendency to capture contrasts or a single monotone color scheme in the posters.

V. Conclusion

The work presented above is an initial attempt at laying the groundwork towards resolving the problem of personalized anime poster generation. The insights and intuitions from training such models have been put across with the intent of benefiting the research community. In terms of additional work, besides the direct approach of bolstering the dataset count, it is likely that leveraging multi-label and multi-class losses in addition to augmenting the existing dataset with image patches can aid in better results. Furthermore, for improving image quality, style based, or perceptual losses can be introduced for preserving the "anime-ness" of the generated image in relation to the original and create more realistic posters. In addition, restricting a particular style and era of anime may help as well. Lastly, a reconstruction loss may be useful, where the generated posters can be segmented and individual components such as faces, landscapes or vehicles can be identified. Future work can expand by consider the rationale behind each selected approach in this work, to reach a fully realized version of a composite anime art generator.

References

[1] J. Oshiba, "Automatic Landmark-Guided Face Image Generation for Anime Characters Using C2GAN.," pp. 236–249, 2020, doi: 10.1007/978-3-030-68780-9_21.

[2] M. Mobini and F. Ghaderi, "StarGAN Based Facial Expression Transfer for Anime Characters," in 2020 25th International Computer Conference, Computer Society of Iran (CSICC), Jan. 2020, pp. 1–5. doi: 10.1109/CSICC49403.2020.9050061.

[3] K. Hamada, K. Tachibana, T. Li, H. Honda, and Y. Uchida, "Full-Body High-Resolution Anime Generation with Progressive Structure-Conditional Generative Adversarial Networks," in Computer Vision – ECCV 2018 Workshops, Cham, 2019, pp. 67–74. doi: 10.1007/978-3-030-11015-4_8.

[4] B. Li, Y. Zhu, Y. Wang, C.-W. Lin, B. Ghanem, and L. Shen, "AniGAN: Style-Guided Generative Adversarial Networks for Unsupervised Anime Face Generation," ArXiv210212593 Cs, Mar. 2021, Accessed: May 30, 2021. [Online]. Available: http://arxiv.org/abs/2102.12593

[5] H. Su, J. Niu, X. Liu, Q. Li, J. Cui, and J. Wan, "MangaGAN: Unpaired Photo-to-Manga Translation Based on The Methodology of Manga Drawing," ArXiv200410634 Cs, Dec. 2020, Accessed: Jun. 02, 2021. [Online]. Available: http://arxiv.org/abs/2004.10634

[6] Y. Hati, G. Jouet, F. Rousseaux, and C. Duhart, "PaintsTorch: a User-Guided Anime Line Art Colorization Tool with Double Generator Conditional Adversarial Network," in European Conference on Visual Media Production, London United Kingdom, Dec. 2019, pp. 1–10. doi: 10.1145/3359998.3369401.

[7] Y. Ci, X. Ma, Z. Wang, H. Li, and Z. Luo, "User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks," in Proceedings of the 26th ACM international conference on Multimedia, New York, NY, USA, Oct. 2018, pp. 1536–1544. doi: 10.1145/3240508.3240661.

[8] "This Anime Does Not Exist." https://thisanimedoesnotexist.ai/ (accessed Jun. 02, 2021).

[9] "This Waifu Does Not Exist v3.5 (TWDNEv3.5) - Gwern." https://www.thiswaifudoesnotexist.net/ (accessed Jun. 02, 2021).

[10] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in Proceedings of the 34th International Conference on Machine Learning - Volume 70, Sydney, NSW, Australia, Aug. 2017, pp. 2642–2651.

[11] J. Kim, M. Kim, H. Kang, and K. H. Lee, "U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation," presented at the Eighth International Conference on Learning Representations, Apr. 2020. Accessed: Jun. 02, 2021. [Online]. Available: https://iclr.cc/virtual_2020/poster_BJlZ5ySKPH.html

[12] W.-F. Zheng and W.-L. Xie, "A Comic Head Images Generation Algorithm Based on Improved Deep Convolutional Generative Adversarial Networks," in 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Apr. 2020, pp. 270–273. doi: 10.1109/AEMCSE50948.2020.00065.

[13] Y. Jin, J. Zhang, M. Li, Y. Tian, and H. Zhu, "Towards the High-quality Anime Characters Generation with Generative Adversarial Networks," p. 13.

[14] T. Xu et al., "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 1316-1324, doi: 10.1109/CVPR.2018.00143.

[15] H. Zhang, H. Zhu, S. Yang and W. Li, "DGattGAN: Cooperative Up-Sampling Based Dual Generator Attentional GAN on Text-to-Image Synthesis," in IEEE Access, vol. 9, pp. 29584-29598, 2021, doi: 10.1109/ACCESS.2021.3058674

[16] "Anime Recommendations Database." https://kaggle.com/CooperUnion/anime-recommendations-database (accessed Jun. 02, 2021).

[17] W. Limberg, waylan/beautifulsoup. 2021. Accessed: Jun. 02, 2021. [Online]. Available: https://github.com/waylan/beautifulsoup

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),

[19] K. Georgiev, "DCGANs — Generating Dog Images with Tensorflow and Keras," Medium, Jan. 10, 2020. https://towardsdatascience.com/dcgans-generating-dog-images-with-tensorflow-and-keras-fb51a1071432 (accessed May 05, 2021).

[20] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral Normalization for Generative Adversarial Networks," ArXiv180205957 Cs Stat, Feb. 2018, Accessed: Jun. 02, 2021. [Online]. Available: http://arxiv.org/abs/1802.05957

[21] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least Squares Generative Adversarial Networks," ArXiv161104076 Cs, Apr. 2017, Accessed: Jun. 02, 2021. [Online]. Available: http://arxiv.org/abs/1611.04076

[22] X. Chen et al., "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," p. 9.

[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," ArXiv170400028 Cs Stat, Dec. 2017, Accessed: Jun. 02, 2021. [Online]. Available: http://arxiv.org/abs/1704.00028

[24] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs Created Equal? A Large-Scale Study," ArXiv171110337 Cs Stat, Oct. 2018, Accessed: Jun. 02, 2021. [Online]. Available: http://arxiv.org/abs/1711.10337

[25] G. Branwen, "Making Anime Faces With StyleGAN," Feb. 2019, Accessed: May 30, 2021. [Online]. Available: https://www.gwern.net/Faces