



HAL
open science

Converting Nebulous Ideas to Reality – A Deep Learning Tool for Conditional Synthesis of Character Designs

Lilian Guo, Anand Bhojan

► **To cite this version:**

Lilian Guo, Anand Bhojan. Converting Nebulous Ideas to Reality – A Deep Learning Tool for Conditional Synthesis of Character Designs. 6th International Conference on Computer, Communication, and Signal Processing (ICCCSP), Feb 2022, Chennai, India. pp.75-89, 10.1007/978-3-031-11633-9_7. hal-04388149

HAL Id: hal-04388149

<https://inria.hal.science/hal-04388149v1>

Submitted on 11 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Converting nebulous ideas to reality – a deep learning tool for conditional synthesis of character designs

Lilian Guo¹ and Anand Bhojan¹

¹ National University of Singapore, 21 Lower Kent Ridge Rd, 119077, Singapore

banand@comp.nus.edu.sg

Abstract. Characters are essential elements of games and are critical to their success. At the same time, designing good characters can be time and labor intensive, especially when developing games with thousands of characters. In such cases, procedural generation may be used to expedite the process. However, characters generated by traditional procedural generation techniques often rely on a limited pool of premade assets and may lack novelty. This work explores deep learning for the conditional generation of creative character designs with artist input. It proposes a framework which receives artists' inputs in the form of blurred character silhouettes and converts these into high resolution character designs using Generative Adversarial Networks. In addition, the paper presents a demo Graphical User Interface and user study evaluating the tool's effectiveness.

Keywords: Generative Adversarial Networks, Game Development, Creativity, Character Design.

1 Introduction

1.1 Character Design in Games and Multimedia

Characters are essential elements of productions like games, films, and books. The visual appearances of characters can affect players' perception of the gameplay narrative [1] and their ability to project themselves onto virtual avatars, and are thus important to the success of a game. However, designing good characters can be time and labor-intensive.

This is particularly true for role-playing games (RPGs), where the player assumes roles of characters in a fictional setting. In many RPGs, character designs are not only needed for the player's own avatar but also the various interactable non-playable characters which the player encounters in their gaming adventure. Another example is the trading card game, a type of collectible card game which lets players assemble decks strategically from an often large pool of character cards. Many character designers have to be involved to develop games of such genres – this is evident in titles like *Magic: The Gathering*, which contained more than 22,000 character cards as of March 2020 [2]. Developing a good character design involves many stages, including

researching on the character's background, sketching the characters, rendering the best sketches, and deciding on a final design [3]. This can amount to a lot of work for character concept artists.

Currently, artists may search for references online or look at examples of their past work to gather inspiration for new character designs. The first approach is time-consuming, and furthermore, artists have to be careful not to refer too closely to designs which do not belong to themselves for fear of copyright infringement. The second approach is limited when it comes to discovering new sources of inspiration. Moreover, neither approach attempts to generate novel character designs — they can only assist the artist in gathering inspiration, and the designs ultimately originate from the artist's own imagination.

The limitations of these approaches, together with the large number of character designs some games require, point to procedural generation as a solution to speed up designers' workflows.

1.2 Generating Creative Character Designs

Procedural generation refers to creating assets algorithmically as opposed to manually. This can be assisted, where content is generated with human guidance, or non-assisted, where the algorithm produces results with little to no human intervention. In game development, procedural generation is often used to generate terrain such as roads, rivers, cities and vegetation [4]. There also exist 3D character creator software such as Autodesk's Character Generator, Reallusion's Character Creator 3 and the open-sourced MakeHuman [5]. These applications allow users to combine selections of premade body parts, clothing and textures to produce a new character. However, designs created by such applications are limited by the number of combinations allowed by existing assets, and characters may be spotted using the same premade hairstyle or accessory. As such, it is hard to describe these forms of procedural generation as being creative or capable of novelty, and their usage is limited to less important characters who do not need to have distinctive design flairs.

In the domain of 2D concept art, computer applications like Adobe Photoshop have traditionally only been tools for realizing human creativity. They enable artists to visualize their designs more rapidly and at potentially lower cost than painting on a physical canvas, but it is ultimately the artist's own creativity which decides what is being rendered to screen.

Unlike the above-mentioned tools and character generation techniques, advances in deep learning have made it possible for artificial intelligence to come much closer to being true agents of creativity. State-of-the-art neural networks can generate high-quality, previously unseen images matching or surpassing those produced by human creators. With deep learning, it is possible to rapidly generate new character designs which may inspire artists or serve as a base for their final concept art. This method of procedural generation hopes to assist artists by making their design workflow less cumbersome and more efficient, allowing them to refer directly to generated designs and reducing the need to gather inspiration from various online sources.

1.3 A Deep Generative Tool for 2D Character Designs

In this work, we propose and study the use of deep generative frameworks like pix2pixHD and StyleGAN2 for assisting artists in producing character designs. The availability of a reference image which closely matches the final design can help to reduce much of the time spent on ideating, and generated images which are of adequate quality may only require minimal post-processing to be used in production. Furthermore, basing a final design off a computer-generated image avoids any potential copyright issues associated with reference to designs drawn by human artists.

Our work focuses on methods which allow artists to control the model’s output in order to finetune results to match their desired aesthetic, instead of generating completely random designs. The main contributions of this work are as follows: (1) We propose a method to convert character artists’ nebulous ideas in the form of blurred silhouettes to coherent character designs in high resolution. (2) We conduct experiments on two different datasets to compare the effectiveness of different generative frameworks for realizing the proposed method. (3) We develop a demo GUI application which enables artists to paint nebulous, blurred forms of their intended design on a drawing canvas and generate a high-resolution output image conditioned on the blurred input.

2 Related Works

2.1 Image Generation using Deep Learning

A Generative Adversarial Network (GAN) [6] is a class of deep learning network architectures comprising a generator and a discriminator which contest against each other in a zero-sum game. For instance, in an image generation task, the generator attempts to synthesize a realistic image while the discriminator attempts to detect flaws which point to the synthesized image being fake.

GANs have proven to be highly competent at image synthesis tasks. Unconditional GANs are capable of producing random novel images belonging to classes within the dataset. On the other hand, a conditional GAN (cGAN) is a subtype of GAN which receives input from the user and conditions the synthesized output image on the user input. While cGANs may appear more suited to the task of generating character designs in a manner that is controlled by the artist, it is also possible to do the same by manipulating the latent space of an unconditional GAN.

Existing works on conditional image synthesis can be broadly classified into two categories: firstly, involving direct image-to-image translation using a cGAN; and secondly, involving a prior step of training an unconditional GAN model before introducing other means of finetuning the output to the user’s desired look.

2.2 cGAN Image-to-Image Translation

Image-to-image translation involves changing an image’s domain. For instance, given a sketch of an object and a photograph of the same object, the goal of image-to-image translation is to translate the sketch to the photograph or vice versa. To attain this goal, the neural network learns a mapping from the image’s input domain to its output domain.

pix2pix is an image-to-image translation framework [7] which performs domain translation using a cGAN trained on image pairs of both domains. The original pix2pix produced images with obvious artefacts when the resolution was raised to 512x512 and above. This was corrected by pix2pixHD [8], which improves results on resolutions as high as 2048x1024. Other notable image-to-image translation works include BicycleGAN [9] and SPADE [10].

2.3 Controlling Outputs of Unconditional GANs

The second common method for conditional image generation relies on first training an unconditional model, and then guiding the synthesis of output images by discovering latent space controls or other means.

StyleGAN [11] is known for being able to synthesize high resolution images of subject matter like faces, animals, and cars. The architecture of StyleGAN encourages the automatic, unsupervised separation of high-level attributes, enabling intuitive control of the synthesized results by modifying latent space vectors. Additionally, “style mixing”, which refers to applying different styles to the content of an image, can be performed by manipulating the W latent space. A later work, StyleGAN2 [12], proposed changes in model architecture and training methods to address the common artefacts found in images generated by StyleGAN, and in doing so redefined the state-of-the-art in unconditional image synthesis.

As StyleGAN2 is an unconditional GAN, the model is unable to directly condition the output to the user’s needs. However, this can be done with the help of Pixel2Style2Pixel (psp) [13]. Building upon StyleGAN and StyleGAN2, the authors of psp introduce an encoder network which can be trained to embed real images into an extended $W+$ latent space. By feeding the artist’s input image into the psp encoder and leveraging an existing StyleGAN generator for image synthesis, image-to-image translation can be accomplished. This approach also facilitates multi-modal synthesis via style-mixing and is capable of yielding high quality outputs without local pixel-to-pixel correspondence.

Finally, the problem of identifying latent space controls has been explored by works like SeFa [14] and GANspace [15]. The former proposes a closed-form approach for unsupervised latent semantic factorization in GANs. The latter proposes applying Principal Component Analysis in the latent space or feature space to identify important latent directions. Both works contribute to discovering interpretable GAN controls and includes results from experiments conducted on notable frameworks like BigGAN [16], StyleGAN and StyleGAN2. These controls enable the targeted manipulation of attributes such as viewpoint, aging, lighting and time of day in the GAN’s synthesized outputs.

2.4 Applications in Art and Design

GANs have been popularized by sites like thispersondoesnotexist.com, which displays photorealistic portraits of people generated by StyleGAN2. Many variations of this site have sprung up and employ unconditional GANs for generating images of subject matter such as cats and bedrooms. While these can be interesting and useful for niche purposes like populating a site with mock data, they are unlikely to be helpful to designers as the images synthesized are completely random.

The rest of this section will focus on applications which are able to respond to inputs from designers and attempt to integrate deep learning into an art or design workflow. In particular, both ArtBreeder [17] and Google Chimera Painter [18] provide an interface for users to submit input to the model, which conditions its synthesized output on the given inputs.

ArtBreeder is a deep learning based collaborative web application which allows users to upload, generate and modify images from various categories. Its character design category was the main source of inspiration for this work. Currently, users can modify (“breed”) character design images by adjusting a list of sliders loosely corresponding to character attributes such as “Female”, “Clothing”, and “Armor”. Other features include morphing two images while varying the contribution of content and style influence from each image, and generating random images resembling the input image. ArtBreeder can be a powerful tool for character designers. However, the platform still contains limitations – such as the input slider controls not feeling intuitive to designers – which this work hopes to address. This is explored in the user study in section 5.3 and is one of the considerations when designing the GUI demo.

Another related work is Google Chimera Painter, a web application which converts segmentation maps of fantastical creatures into fully fleshed out renderings. The model was trained on input pairs comprising images of 3D creature models and their corresponding artist-labelled segmentation maps indicating the various creature body parts. In the interactive demonstration, users are able to paint on a canvas and control the generated output by adding strokes of colors corresponding to different creature body parts. A major limitation of this method is that training requires a large and high quality dataset of images paired with accurate segmentation maps, which is costly to produce. Furthermore, artists do not have control over the final output colors because segmentations may not correspond to the RGB color of the corresponding regions in the generated output. This work aims to address these limitations by proposing a method which greatly simplifies creation of the training dataset, and also enables control over the generated colors.

3 Data Collection

3.1 Overview

To the best of our knowledge, there is no existing public dataset for full-body character designs apart from an anime style dataset [19], which is limited to a specific art style. Hence, we decided to build a new character designs dataset to compare the models’ performance on different styles of character designs and to evaluate the transferability of the approach to different styles. Compared to the anime style dataset, the new dataset is more challenging as the poses, lighting conditions and general contents of the designs are more diverse.

To build the character designs dataset, images were first downloaded in bulk from web browser DuckDuckGo [20] image search results. The results were fetched using 20 different keywords related to common fictional genres for character designs, such as “warrior character design”, “archer character design”, and “elf character design”. Images were filtered for portrait dimensions to increase the probability of them depicting single full body characters.

3.2 Data Cleaning

A list of requirements was curated to increase consistency across images in the dataset. Characters should be humanoid and be shown in full body. In addition, each image should only contain one character in a straightforward pose. The art style should also avoid being too “cartoonish”, i.e. rendered with flat colors and lacking detailed shading. This is done to distinguish the new dataset from the anime style public dataset and to ensure consistency in the art style. After downloading, images were manually inspected to ensure that they conform to the requirements, and the non-conforming images were discarded. Duplicates were also removed by computing the MD5 hash of each image and discarding the ones with hash collisions. After filtering, the dataset contains 5119 images of full-body character designs.

To further improve the dataset quality, we initially used a pre-trained segmentation model [21] to identify background regions and color them white. This was ineffective as many images lacked a clear distinction between foreground and background. Hence, all images were re-processed using another method which applies a strong Gaussian blur to backgrounds instead of coloring them white. Thereafter, images were resized to a standard 512x512 dimension while preserving aspect ratio, and the empty areas padded with blurred, repeated sections of the image. This method produced more coherent-looking results even if large sections were segmented wrongly. For instance, body parts which were mistakenly identified as background are now blurred instead of being completely blanked out. Samples of the dataset are shown in Figure 1.

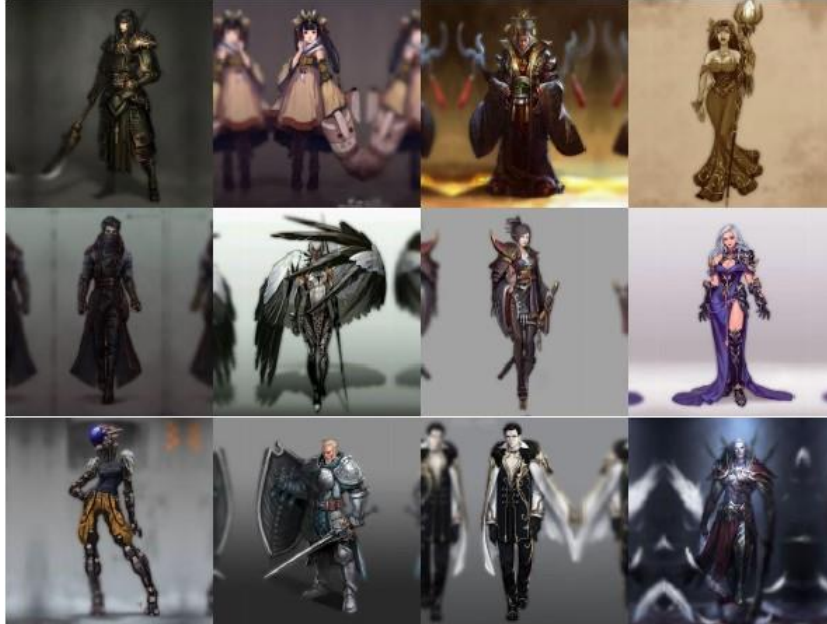


Fig. 1. Examples from our new dataset. Original (unprocessed) images from Pinterest, Artstation, DeviantArt.

4 Experiments

4.1 Overview and Training Details

In our experiments, we compared the results from using pix2pixHD to perform domain translation directly and using the psp framework together with a StyleGAN2 generator for image-to-image translation. This was done on both a public dataset of anime style character designs [19] and on our own collected data. Training was conducted on two Titan Xp GPUs at a standard size of 512x512 pixels.

4.2 Input Domain Selection

When approaching the problem of generating character designs as an image to image translation task, an important question is what input domain would be most suitable. In order to maximize the usefulness of the tool for character designers, we try to fulfil the following criteria in our selection of the input domain: (1) The input domain allows for a balance between artist control and the neural network’s ability to generate creative and realistic outputs. (2) When using the tool, it is easy for artists to modify the input domain. (3) There exists an easy, automated way to produce training data in the form of input output image pairs.

For instance, if the input domain very closely matches the output, (1) may not be satisfied because the network may be trained to simply reproduce the input rather than

generating a novel interpretation. (2) is also unsatisfied because significant effort is required to edit a fully rendered character design.

Using the above criteria, we decide to apply a simple Gaussian blur transformation to the output images in our training dataset to obtain the corresponding input pairs. After training, a blurred input image can be passed to the psp encoder and StyleGAN2 generator to obtain a high-resolution result. Artist control exists to a rather high degree as the image encoded by the network will be similar in both pose and color to the input. Users can also tweak the input easily by painting blurred blobs of color without having to worry about details. Moreover, it is convenient to produce training image pairs – the blur transformation can be easily applied to any new dataset in a completely automated process. This is not true for some input domains like sketches or segmentation maps, which usually require extensive manual labelling to produce.

4.3 Results

pix2pixHD models were trained to convert blurred images of character designs into their high-resolution versions. This was done with batch size 8 for 165 epochs, producing results such as in Figure 2, selected randomly from the test set.

Next, we trained StyleGAN2 models to generate unconditional samples of character designs. This was done using batch size 4 over 400,000 iterations. Sample outputs are shown in Figure 3.



Fig. 2. Examples randomly selected from the test set of pix2pixHD model trained on image pairs from anime-style dataset (top rows) and new dataset (bottom row).



Fig. 3. Example character designs produced by StyleGAN2 models trained on anime-style dataset (top row) and new dataset (bottom row).

Finally, we trained a psp encoder which accepts blurred images and encodes them into the $W+$ latent space. This was done using batch size 2 for 500,000 iterations. Together with the StyleGAN2 generator, this framework performs domain translation from a blurred image into a high resolution image (Figure 4). Style mixing can be used to generate varied outputs from a single input image, as shown in Figure 5.

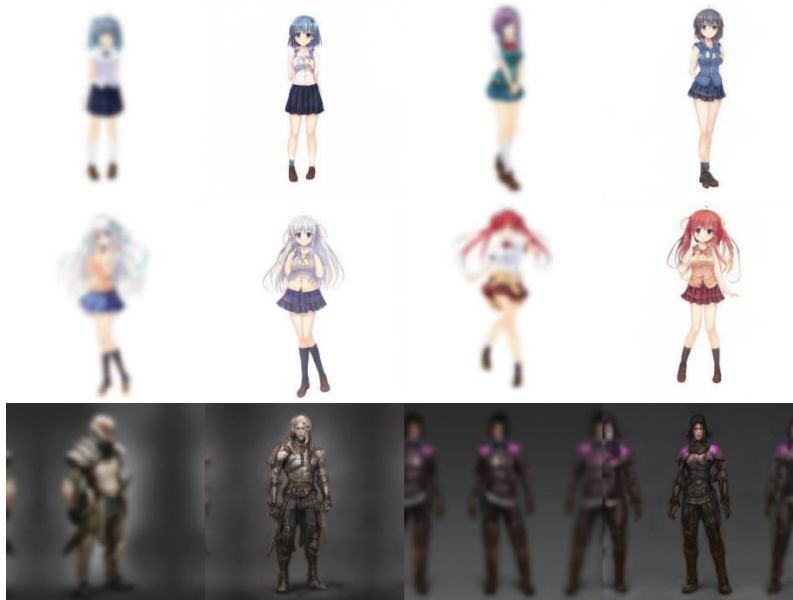


Fig. 4. Examples randomly selected from the test set of psp model trained on image pairs from anime-style dataset (top rows) and the new dataset (bottom row). For each pair, the left image is the input and the right image is the generated output.



Fig. 5. Examples of style-mixing on outputs of psp model from Fig. 4. The leftmost column shows the input image, and the remaining columns show random style-mixed examples.

Comparing the results from both frameworks, we observed that pix2pixHD produces outputs with more local pixelwise correspondence to the input but tends to produce undesirable artefacts on details such as the eyes. While psp outputs do not match the input as closely, they often possess more convincing details. Furthermore, psp has the advantage of being able to easily synthesize varied outputs using a single input by performing style mixing on the StyleGAN2 generator. Hence, we eventually chose to use psp for our final application.

5 GUI Application

5.1 Methods Evaluation

In section 2, we described two approaches for conditional image synthesis: the first involving direct image-to-image translation using a conditional GAN, and the second involving training an unconditional GAN followed by making adjustments in the latent space to produce outputs of desired styles. Section 3 elaborated on the model training conducted to evaluate different frameworks for both approaches, and it was decided that psp would be more suitable for the final application.

Section 2.4 reviewed existing applications which integrate deep learning into an art or design workflow, namely ArtBreeder and Google Chimera Painter, and mentioned some of their current limitations. The GUI application proposed in the following sec-

tion shall attempt to address these limitations. In particular, it distinguishes itself from ArtBreeder by introducing a more intuitive form of control for the artist, allowing them to directly draw on a canvas as opposed to solely adjusting sliders to finetune the generated image. Compared to Google Chimera Painter, the proposed application offers the artist more control over generated colors. It also allows users to freely color the input image rather than having to block in regions of color to form segmentation maps. The effectiveness of the application is evaluated in a user study (section 5.3) involving artists and character designers.

5.2 Application Concept and Features

A demo version of the proposed GUI application has been implemented in Python using the Tkinter library. As the goal of the application is to help artists transform their initially nebulous ideas into coherent character designs, it is named ‘Lucid’.

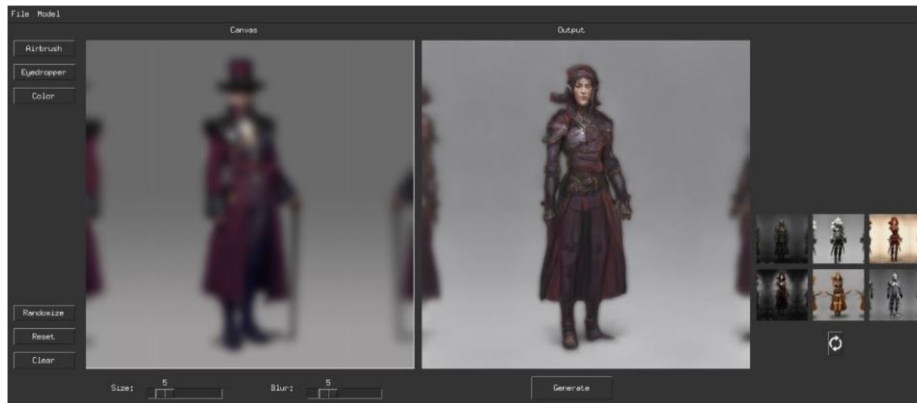


Fig. 6. Screenshot of our demo GUI. A video demo is included [here](#).

In summary, Lucid contains the following features (refer to Figure 6):

1. Canvas widget – Displays the blurred input image which represents the artist’s initial nebulous conception of their design. Users can freely edit the canvas using the app’s painting tools. This image will be sent to the model when users click the Generate button.

2. Output label – Displays the resultant image generated by the model. The output can be refreshed anytime by clicking the Generate button.

3. Artist painting tools – Buttons on the left side panel serve as basic painting tools for artists to modify the canvas. They include an airbrush, eyedropper (color picker), and color chooser (RGB color selection), as well as a slider to control the brush size. There are also buttons for clearing and resetting the canvas to the originally imported image.

4. File utilities – Users can choose to either start drawing on a blank canvas or to import an image into the program using the Import Canvas function in the file menu. New users can also use the Randomize button to load a random image from a directo-

ry of example images. After importing, the app automatically blurs the image to fit the model’s input domain. The menu options also support saving the canvas and generated output for future reference.

5. Checkpoint switcher – Lucid allows different StyleGAN2 checkpoints to be loaded via a menu option. There are currently two pretrained generators available: an anime style model trained on the public dataset and a semi-realistic model trained on the newly assembled character designs dataset.

6. Style randomizer – This injects random vectors into the latent space of the generated image to produce variants with different styles.

7. Style-mix gallery – A collection of thumbnails displays the variants generated by the style randomizer. Users can hover over a thumbnail to preview the image on the output label, or click on the thumbnail to set it as the current output.

The sample workflow for a character designer is as follows: The designer opens the app and either starts drawing on the blank canvas or imports an existing design they are inspired by. The canvas now contains their initial nebulous conception of the design. The user can then click on the Generate button and evaluate the image synthesized by StyleGAN2 on the output label (example shown in Figure 6). If they are not satisfied, they can continue making adjustments to the canvas using the airbrush tool and click Generate to refresh the output. The style-mixer can also be used to produce variants of the current design. Once the user is satisfied with the design, they can save it for future reference using the file utilities.

5.3 User Study

To assess the usefulness of the GUI application, we conducted a user study involving a survey sent to messaging groups interested in art and design. Participation in the survey was voluntary. As the app can currently only be run locally, the questions are based on videos, images, and descriptions of the interface. The survey had a total of 27 respondents, 15 of whom indicated they draw or design characters “often” or “all the time”.

The main objectives of the user study are to determine how helpful the application is to artists and character designers, compare it with existing platforms like ArtBreeder and Google Chimera Painter, as well as gather feedback for improvements.

In general, the responses were very positive, with 77.8% stating they would use the proposed application for generating ideas or references for character design. A few options were provided for justification, and the main options selected were “It can potentially speed up my workflow”, “The ability to control the output generated by drawing on the canvas is useful for ensuring the design doesn’t stray too far from my intention”, and “It is easier to select and modify AI generated ideas than to generate them myself”. Of the 22.2% who would not use the application, the main reasons given involve the generated outputs being of insufficient quality or too generic for specific design intentions.

The next section of the survey asked respondents to compare existing platforms to the proposed application. Links to the platforms were provided and respondents were given the option to try them if they had not done so before. Of the 9 respondents who

have used ArtBreeder, 100% felt that the proposed application’s method of drawing on a canvas directly was more intuitive than the slider modifiers provided by ArtBreeder. The reasons given include it being “simpler to draw a vague idea” as opposed to choosing from Artbreeder options, which can be “a little confusing”. Respondents also felt that the canvas interface made the deep learning assisted workflow more closely resemble the natural process of drawing.

Subsequently, respondents were asked to compare the application with Google Chimera Painter based on visual and text descriptions of the web app, as well as a link to the demo page. 59.3% of respondents felt that drawing the blurred form of a character, as proposed by this project, was more intuitive than drawing segmentation maps in Google Chimera Painter. Those who preferred this project’s method stated that it had a “simple user experience”, required “less knowledge on how elements interact with each other”, and “allows more control over the colors”. On the other hand, those who preferred Google Chimera Painter answered that segmentation maps provided better control over details of the design.

In summary, the surveyed target users expressed significant interest in using the application to enhance their existing design workflows. 100% of respondents indicated that they would like to try out an interactive web demo if it was made available. Compared with existing platforms like ArtBreeder and Google Chimera Painter, the majority of respondents felt that this proposed method was more intuitive. One of the main reasons provided for not choosing to use the application relate to the quality and distinctiveness of the model output, which we believe can be further improved with a larger training dataset.

6 Conclusion and Future Work

6.1 Summary

Conditional synthesis of full body character designs in varied poses and clothing is a challenging task, but one with promising applications in game development and other media industries. This work has proposed a method for helping artists transform the nebulous ideas they may have at the start of a design process into coherently rendered designs. It has also showcased an interactive demo providing artist friendly features for engaging in the proposed workflow, which uses StyleGAN2 and the psp framework to convert blurred images into high resolution character designs of varying styles. The user study indicates that most of the respondents surveyed would like to use this application and find the proposed user experience more intuitive than existing platforms. Additionally, we have assembled a rare new dataset of full body character designs and will make it available for future research on related topics.

6.2 Limitations

While a majority of respondents in the user study felt that the proposed user experience (drawing the blurred form of a character on a canvas) was more intuitive than adjusting slider controls or drawing segmentation maps, there was still a significant

minority who felt the opposite and pointed out problems about the input method. In particular, a few respondents felt that drawing the blurred form of a character was difficult and preferred blocking in silhouettes such as in Google Chimera Painter. To address this issue, future works can explore alternative input domains and may take into consideration the criteria suggested in section 4.2.

Some respondents also felt that the input method did not allow the artist to specify sufficient details and gave the model too much freedom for interpretation. This may sound ironic as the purpose of the application is to help transform vague ideas into more coherent designs, and having the artist draw more details in the input seems to defeat the point of using the application. However, it is possible that the respondents were seeking ways to guide the model towards more specific design intentions without having to specify details directly on the canvas, or to inform the model to preserve certain details which they already have in mind while exercising creativity on other aspects of the design. The former can be addressed by introducing features such as slider controls in the style of platforms like ArtBreeder, which would allow artists to fine-tune specific attributes of the generated output. The latter is more challenging as it requires a way of segregating important details from other more flexible attributes. For instance, if the artist requires a red bow decal to be present in the final design, the framework will have to somehow relay this information to the neural network. It may be possible to do so by selectively varying the strength of the blur filter on different parts of the input image, such that important details are sharper and thus emphasized.

6.3 Recommendations for Future Work

Apart from addressing the limitations raised above, future work may focus on making the proposed interface more accessible to users by developing it as a web application. Thereafter, a larger scale user study can be conducted to gather more representative opinions on the effectiveness of this tool.

In addition, extensions to this work can examine ways to expand the character designs dataset and improve data quality. This can enhance StyleGAN2's generation as well as prevent overfitting.

References

1. Pradantyo, R., Birk, M. V., & Bateman, S.: How the Visual Design of Video Game Antagonists Affects Perception of Morality. *Frontiers in Computer Science*, 3 (2021).
2. Magic: The Gathering, <https://magic.wizards.com/en>, last accessed 2021/11/21.
3. Filho, J., Machado, L., Chiccha, N., Franco, A., & Maia, J.: Character Design: a new Process and its Application in a Trading Card Game. In: *Proceedings of SBGames, São Paulo*, (2016).
4. Freiknecht J. & Effelsberg W.: A Survey on the Procedural Generation of Virtual Worlds. *Multimodal Technologies and Interaction*, 1(4):27 (2017).
5. MakeHuman. <https://windows.softwsp.com/makehuman/>, last accessed 2021/11/21.
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y.: Generative adversarial nets. *Advances in neural information processing systems*, pp. 2672-2680 (2014).
7. Isola, P., Zhu, J., Zhou, T., & Efros, A. A.: Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2017).
8. Wang, T., Liu, M., Zhu, J., Tao, A., Kautz, J., & Catanzaro, B.: High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2018).
9. Zhu, J., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., & Shechtman, E.: Toward Multimodal Image-to-Image Translation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2017).
10. Park, T., Liu, M., Wang, T., & Zhu, J.: Semantic Image Synthesis with Spatially-Adaptive Normalization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2019).
11. Karras, T., Laine, S., & Aila, T.: A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2019).
12. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T.: 2019. Analyzing and Improving the Image Quality of StyleGAN. *arXiv:1912.04958 [cs.CV]* (2019).
13. Elad R., Yuval A., Or P., Yotam N., Yaniv A., Stav S., and Daniel C.: Encoding in Style: a StyleGAN Encoder for Image to Image Translation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2287-2296 (2021).
14. Shen, Y. & Zhou, B.: Closed-Form Factorization of Latent Semantics in GANs. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2021).
15. Härkönen, E., Hertzmann, A., Lehtinen, J., & Paris, S.: GANSpace: Discovering Interpretable GAN Controls. *arXiv:2004.02546 [cs.CV]* (2020).
16. Brock, A., Donahue, J., & Simonyan, K.: Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv:1809.11096 [cs.CV]* (2018).
17. This Person Does Not Exist. <https://thispersondoesnotexist.com/>, last accessed 2021/11/21.
18. Simon, J.: ArtBreeder, <https://www.artbreeder.com/>, last accessed 2021/11/21.
19. Toor, A. S., & Bertsch, F.: Using GANs to Create Fantastical Creatures. *Google AI Blog*, <https://ai.googleblog.com/2020/11/using-gans-to-create-fantastical.html/>, last accessed 2021/11/21.
20. DuckDuckGo. <https://duckduckgo.com/>, last accessed 2021/11/21.
21. Zepeng L.: Generating Full-Body Standing Figures of Anime Characters and Its Style Transfer by GAN. Thesis, Graduate School of Fundamental Science and Engineering, Waseda University (2020).

22. Breheret, A.: Github repository. <https://github.com/abreheret/PixelAnnotationTool>, last accessed 2021/11/21.