Applications of Generative Adversarial Networks for Facial Style Transfer

Kyle Rodrigues, Nathan Chapman, Nathan DuPont, Kiel Dowdle, Stuart Harley

{rodriguesk, chapmann, dupontn, dowdlek, harleys}@msoe.edu

Abstract

The evolution of AI (Artificial Intelligence) technologies has advanced the development of human society and is transforming every walk of life. With the importance of this technology in our lives, Milwaukee School of Engineering (MSOE) is collaborating with Discovery World and Rockwell Automation to develop a new interactive exhibit at Discovery World that provides an educational opportunity for AI that is not currently available to this extent in the Milwaukee area. Developing an interactive exhibit based around AI is a challenging task to accomplish for many reasons; one of the largest reasons being the diversity and age present within the target population. To provide all patrons with the best experience, a solution needs to be made such that it will not express any forms of bias towards the attendees, allowing for all individuals to get engaged and find potential interest in AI. With those considerations in mind, we developed a web-based application capable of detecting a face from an image, which then uses a Generative Adversarial Network (GAN) to apply a style transfer in real time. This web application is based on open-source code (OpenCV and CycleGAN) and could eventually be extended and deployed at Discovery World.

Considering the key user demographic at Discovery World is racially diverse and primarily younger individuals, we developed a process to test the performance of the models for all ages and races. Since many public datasets have limited or no access to photos of younger individuals for privacy reasons, we created a subset of the UTKFace dataset [1] and segmented it evenly based on age and race within our target demographic. We then performed facial detection and style transfers on these subsets using the open-source models and presented the results in our paper.

1. Introduction

Generative Adversarial Networks (GANs) are models that consist of both a generative model (generator) and a discriminative model (discriminator). The generator is used to produce a 'fake' output, such as an image produced from random noise, or a style transformation of an input image. The discriminator is then fed a set of 'real' and 'fake' images and tasked with determining what set each image belongs to. The output from the discriminator can then be used to train the generator to create better 'fake' images and train the discriminator to be better at finding these 'fake' images. Through a repeated training process, the entire GAN structure can produce a generator that is able to create realistic images, or image transformation, depending on the desired use case, and a discriminator that decides if images are real or fake. For our experiments, we will be using a pre-trained generator model.

In this project, we look to use pre-trained GAN models, which have been trained to apply a style transformation, to produce a style transfer on an image that we provide to the model. To make this accessible to a general audience of participants, we also need to produce an interface application to allow users to interact with the chosen GAN model. In this paper, we execute two primary experiments to produce a final product targeted at supporting a new exhibit in Discovery World. In our first experiment, we propose a web application architecture that supports interfacing with a backend GAN model performing style transfer. In our second experiment, we audit the performance of our chosen GAN models to determine the potential experience of participants within our target demographic. These two experiments are independent but are both pertinent to the product for our key stakeholders and are presented in detail in the following sections of this paper.

For a project this intense, there are many components that came into consideration for the team, and decisions that we made for the sake of simplicity or transparency. For example, we only use one of the three parts included in the UTKFaces dataset since the other two sections weren't useful or held too much information for it to be considered useful under the circumstances of this project. There are many topics that are included in this dissection of the engineering process, namely: Key Stakeholders, Hypotheses, Dataset Information and Analysis, Experimental Methodology, Experimental Results, Ethical Considerations, and a Conclusion with an added Future Work section. All of this is included to help better understand the processes and decisions the project involved to allow recreation of this project for improvement or experimental testing.

2. Key Stakeholders

This section highlights the clients involved with this project and provides background information about each, including motivations for being involved with this project.

2.1. Discovery World

Founded in 1983, Discovery World is a Milwaukee non-profit science and technology center designed for families, especially those with young children. The founders believed that the world would be a better place and the future would be brighter if there were more inventors and innovators in it. So, they created a place where young people could get their hands on science and technology. Originally built in the basement of a Boys & Girls Club, the center has expanded several times to its current

120,000 square feet facility located on the shore of Lake Michigan. The mission of Discovery World is to "provide fun and educational experiences through interactive exhibits and educational programs" [2].

Discovery World was the primary client for our project. They are developing a new exhibit with a focus on allowing patrons to experience and learn about AI. A successful result from this project could be included in this exhibit and put on display at Discovery World. Our contact with Discovery World went through Dr. Derek Riley, the director of the computer science program at MSOE.

2.2. Rockwell Automation

Rockwell Automation, a Fortune 500 company, is an American provider of industrial automation and information technology. Headquartered in Milwaukee, Wisconsin, the company was originally founded in 1903. They create control systems, industrial control components, information software, motor control devices, sensing devices, network technology, safety technology, industrial security and have even manufactured components for NASA that have gone to the moon.

Rockwell Automation was a primary stakeholder for this project because they are sponsoring the exhibit space in Discovery World that this project is a part of. Like Discovery World, their primary goal of this project is to inspire an interest in artificial intelligence technology through the exhibit. Additionally, this generated interest could push younger individuals into a career involving AI, which could eventually lead to these individuals working at Rockwell in the future. Our contact with Rockwell Automation also went through Dr. Derek Riley.

2.3. Milwaukee School of Engineering

Milwaukee School of Engineering (MSOE) recently added a new computer science program to its list of offered degrees. This undergraduate program is unique because of its focus on Artificial Intelligence. The members of this project are all part of the first class of Computer Science students at MSOE.

MSOE partnered with Discovery World and Rockwell Automation to create an engaging exhibit highlighting AI to be featured at Discovery World. MSOE's motivation for this project was to build a better relationship with the community and create more youth awareness and interest in AI. This could inspire more individuals to apply and/or enroll in MSOE's computer science program in the future. Dr. Derek Riley is acting as the project owner and the primary point of contact with the other stakeholders from Discovery World and Rockwell Automation.

3. Hypothesis

This section introduces the two hypotheses that the team created based on the information provided from the stakeholders. The team spent most of our time exploring the main hypothesis, and the reach hypothesis is something we could pursue if provided with enough time.

3.1. Main Hypothesis

If a face is found in an image, then we can perform a style transfer driven by a GAN on the face to output a modified image of the individual's face. Specifically, the style transfer refers to applying the style of a famous painter to the image.

3.2. Reach Hypothesis

If we can find the latent space vectors that control specific parts of a face in a GAN produced image, such as hair color, wrinkles, expression, etc., then we can allow the patrons of the exhibit to change their own faces in real-time.

4. Web Application Development

4.1. Experimental Methodology

For the development of our web application, we identified several key components that need to be implemented to have a functioning and interactable exhibit. This section identifies the key components and deployment mechanisms that we have created for this application and highlights the interactions between the components.

4.1.1. Pipeline and Deployment

The pipeline of this project consists of 3 individual components. The first component is a web application with simple capabilities of retrieving an image from a user and rendering a processed image back to the user. The second component takes the image retrieved from the web application, localizes, and extracts an individual's face in the image. That image is then sent into our third component, a GAN model, which applies a style transfer to the image. The final image is displayed on the web application back to the user.

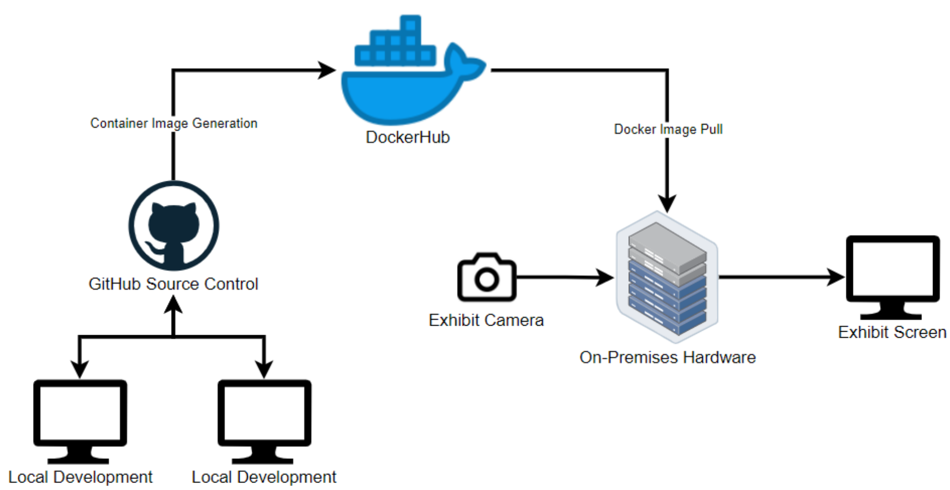

Figure 4.1: Application Deployment Workflow

Currently, we have chosen to have the user submit a single image through our web application, to produce an output. This manual process demonstrates all components of our product working and produces an augmented output for our user. Our current solution for a production deployment of this pipeline is to package the components (face detection, GAN, and web application) into a Docker container [3] through a custom Dockerfile, which will then be used to construct a Docker image that can be hosted publicly or privately on Docker Hub. This image can then be pulled down and executed on any device with docker installed, as illustrated in Figure 4.1. When the container is run, our Flask web application will be publicly exposed on a container port (ex. 3000), which will allow an end user application to display our Flask application on the chosen port through the host machine, which will display the output of the web application. Hosting our application through Docker allows for high usability of our product. Whether the final hardware is a on-premises server rack, or a server hosted in the cloud, encapsulating our product in a Docker container manages all dependency installations and overhead for our solution. Modifications to the application code can be done from anywhere through our GitHub source control [4], which is hosted publicly, which then can be used to generate an updated container image for use. Once ready for production, the exhibit can pull down the newest Docker container version and start running the updated application code. This architecture maximizes response time in the case of adverse results coming from our application, allowing for support staff to fix and deploy code fast, with minimal interaction with the exhibit hardware.

4.1.2. Web Application Component

The web application is built using Flask [5], a lightweight templating framework for Python. This allowed us to keep our entire stack in Python instead of developing our web components with a JavaScript alternative. An entirely Python-based stack allows for easy integration with the Face Detector and CycleGAN implementations, as they both have Python backends. Choosing a JavaScript framework for our web application would have required creating a service-based architecture to communicate between the components, adding much more overhead to our project.

The web application manages interactions with our end user; in specific, the application allows the user to submit an image, and renders the style transferred image back to the user. When the application is first run, it also performs the required initialization of our application dependencies, including the Face Detector and CycleGAN models. This allows for runtime operations to perform much faster, as the required initialization is no longer factored into the inference time, instead occurring on application startup.

4.1.3. Face Detection Component

OpenCV [6] supplies open-source libraries for computer vision. OpenCV provides a pretrained AI model for detecting faces from a given image frame, which then can be used to return an array with a collection of “faces” found in an input image, with varying confidence values. We developed a simple implementation of this library which detects a face in a single image. In the instance of multiple faces being detected which have a confidence greater than our threshold, only the largest (closest to the source) face is captured. In the case that no face is detected in the image, the model returns the source image with no extracted faces.

4.1.4. GAN-Driven Style Transfer Component

Our core facial augmentation functionality is built around the CycleGAN architecture, proposed by Jun-Yan Zhu et. al. in August of 2020 [7]. This model uses earlier research based around style transfer and super resolution, which allows for the model to produce output images in comparable resolution to the input, with the desired style transfer applied. Similar models in the field often suffered from resolution loss, as shown in the architecture discussion of the DC-GAN architecture [8]. The model’s focus is to work with a “cycle” of images, where the style is derived from the first image in the cycle and applied to the second image in the cycle. This can be applied in both directions, to produce distinctive style transfer effects. Our specific implementation uses pre-trained models produced by the researchers of the original CycleGAN model to perform art-based style transfers on the images by applying the forward cycle onto our chosen input images.

We have extracted much of the logic of the CycleGAN implementation into a custom class, which allows the user to execute inference through a simplified function call. The custom class has built-in configurations to allow for external applications to have different and variable interactions with the class, such as changing the style transfer of the output image. Calling to the inference function takes in an image and applies the cycle to it to render a style transfer.

4.2. Performance Results

With the CycleGAN model being integrated into the Flask app, we were able to configure all our models during the initial configuration of the webpage. This setup was critical to improving our processing time, from when the user submitted an image to the time that the user obtained the resulting image from the webpage.

Figure 4.2: Flask Interface After Image Upload

To test the processing time of the application, we used the image provided on the left in Figure 4.2 to test the face detector and the style transfer GAN. The application would then produce the image on the right, passing it through the style transfer GAN produces the results as shown. The original image has a size of roughly 50 KB, while the facial extraction has a size of around 8 KB. To test performance, we ran this image and others of varying sizes (see Table 4.1) through the model 10 times manually and tracked the time from the application receiving the HTTP Request to the time the application saved the modified photo and updated the template. This situation serves to test the full application execution time (including face detection, cropping, and CycleGAN model inference), and represents the time of all interactions performed by our application.

Image Number	Face Present?	Original Size	Reduced Size
1	Yes	27 KB	3 KB
2	Yes	50 KB	8 KB
3	Yes	28 KB	26 KB
4	No	325 KB	257 KB

Table 4.1: Image Size Reduction During Processing

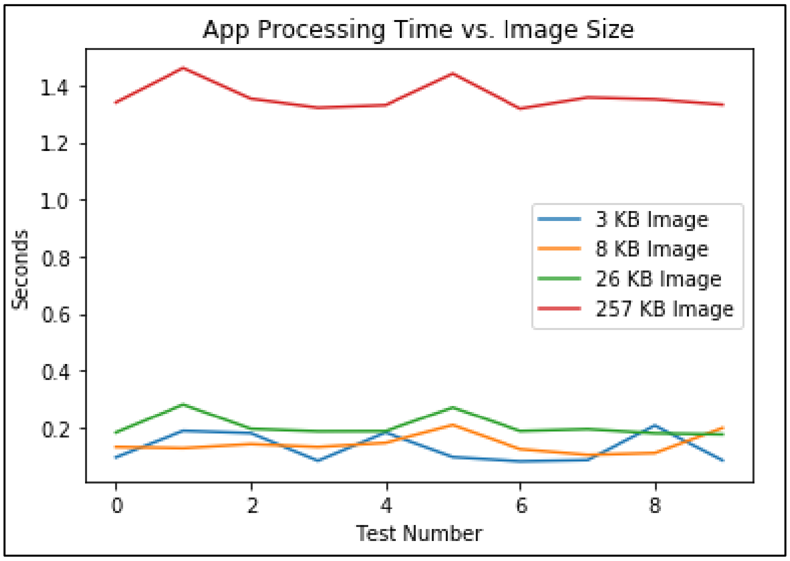

Figure 4.3: Comparing Application Processing Time and Image Size

Our tests were performed against four different images, outlined in Table 4.1. All four images had people, but the face detector only was able to detect, and crop faces in images 1, 2, and 3. There was no faces detected in image 4, so the entire image underwent style transfer. The size reduction was dependent on the size of the face found in the image, and its proportion of the total image size. For instance, image 3 was a single individual's face, close enough in size to make it roughly the same size of the face detector's crop, making the total reduction minimal.

Each image was passed into the application a total of 10 times, as shown in Figure 4.3. Overall, each plot stayed close to its mean processing time, with the facial images producing results in roughly 200 milliseconds per run. Scaling up to image 4, the application was able to run inference on the entire scene in roughly 1.4 seconds, or 1400 milliseconds. Comparing the image size with the inference time on all our samples yields a linear relationship with an R-Squared coefficient of determination of 0.9993. This relationship has strong implications for future scalability of the application, such as incorporating a video feed into the application instead of single image processing and can allow us to predict application performance based on input resolution.

5. Model Accuracy Auditing

5.1. Dataset

This section introduces the UTK Face dataset used in the project along with information on how the data was collected, some sample distributions of the data and any excluded data.

5.1.1. Sources

The primary dataset for this project will be the UTKFace dataset [1] which was introduced as a byproduct of Zhifei Zhang's work in his paper on Age Progression/Regression by Conditional Adversarial Autoencoders [9]. The dataset's original purpose was to be used to find if a model could take input images of faces and either increase or decrease the age of the person in the image.

5.1.2. Collection Process

The dataset was originally gathered from images of people on the internet. The dataset includes labels for age, gender, race, and the datetime of when the image was collected. Races are categorized as either White, Black, Indian, Asian, and other. The DEX algorithm [10], a CNN trained to identify age, race, and gender of an individual, was used to estimate the ground truth of the age, gender and race labels and was double checked by a human annotator. While it does not appear like there are any clear biases in the data, since the data was found on the internet, some of the images have watermarks on them which is something to consider as an obstacle for the model. The dataset contains images of faces from a large variety of different settings, so age, race, gender, lighting, and background biases among others should be minimized in this dataset.

5.1.3. Sample Sizes

The dataset, after some cleaning explained in Section 5.1.4, consists of 24,105 labeled images. Some basic distributions of the data for each of the labels, excluding datetime, are plotted below on histograms. It is important to note that the data does overlap across the labels, for example an image of a man of age 20 can be found in the male gender category and the age category along with whatever race category he belongs to.

Figure 5.1 presents age distribution. Age is distributed in a range from 0 to 116 years old with much of the data focusing on young adults in their 20s and babies.

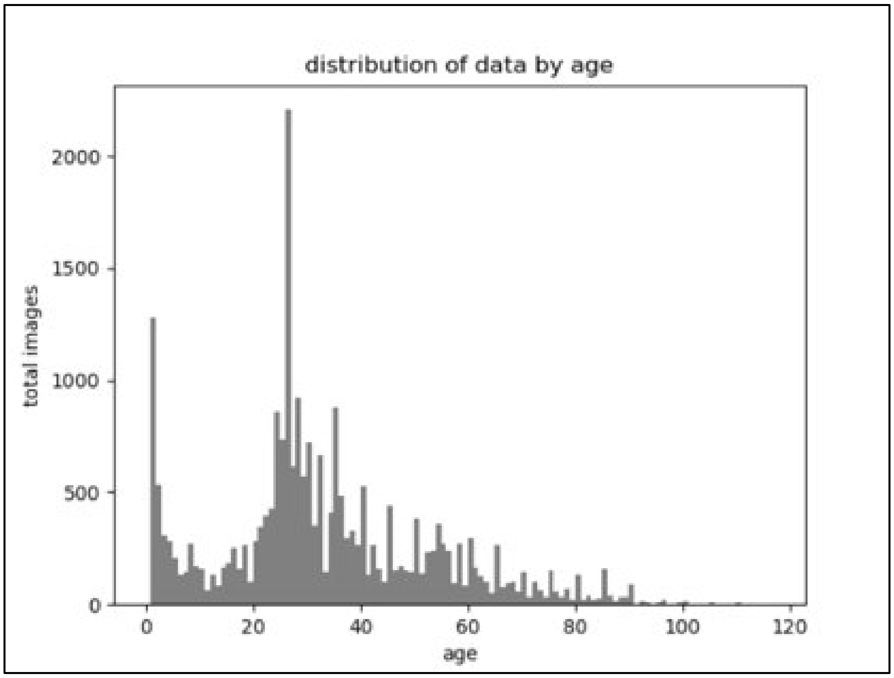

Figure 5.1: Distribution of UTKFace Dataset by Age

Figure 5.2 presents gender distribution. As shown below, there is a near equal distribution of male and female images.

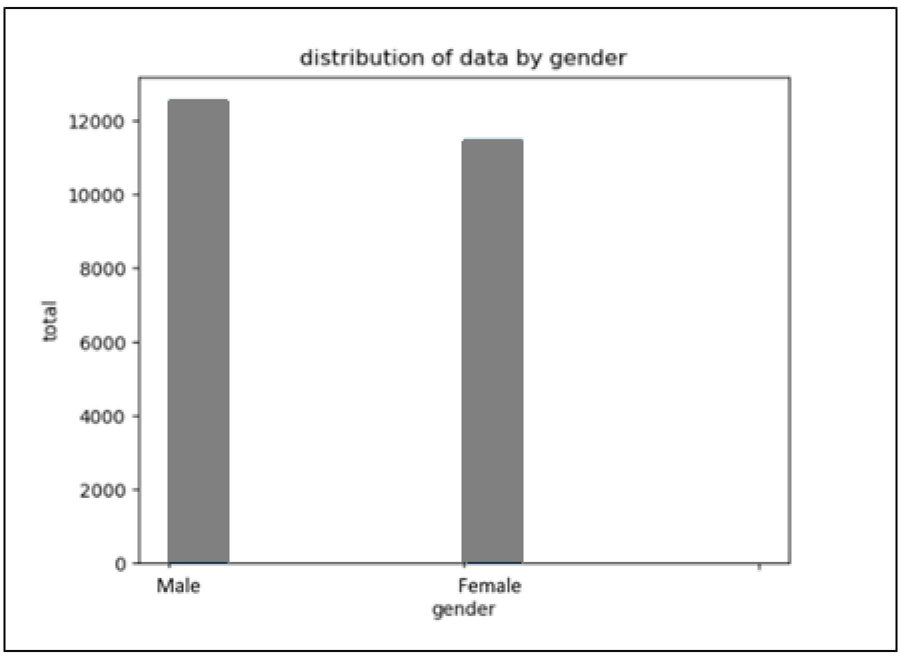

Figure 5.2: Distribution of UTKFace Dataset by Gender

Figure 5.3 presents race distribution. Race is segmented into White, Black, Indian, Asian, and Other. This dataset primarily has individuals of White race, while it has a more equal distribution for the other races. “Other” refers to other races that are not in one of the preceding categories, such as Hispanic, Latino, and Middle Eastern.

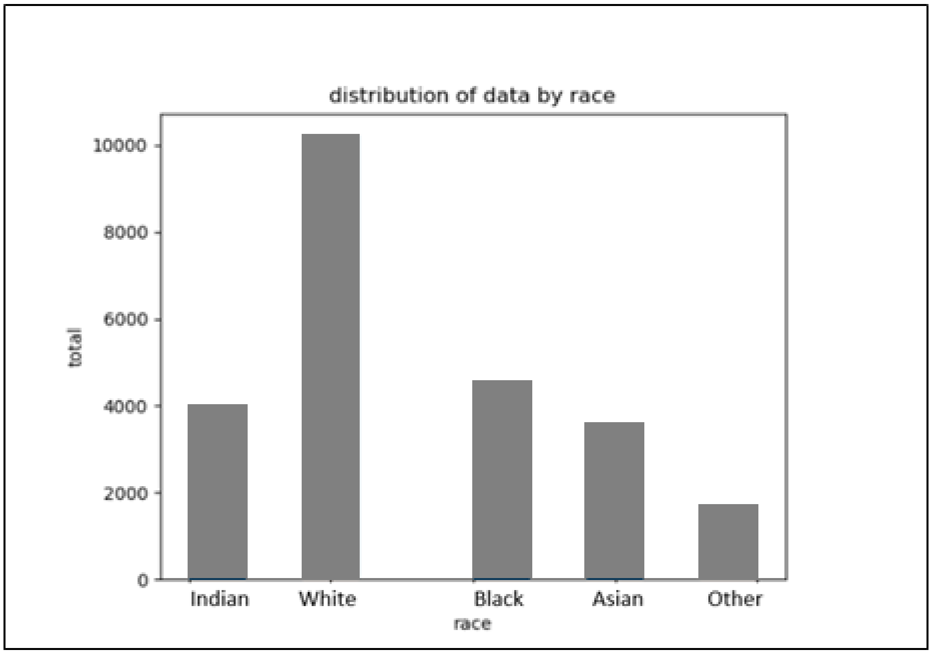

Figure 5.3: Distribution of UTKFace Dataset by Race

5.1.4. Excluded Data

The UTK dataset includes three different sets in it. It includes faces from the internet that have been unedited, a cropped and aligned dataset of faces, and landmarks (datapoints on facial structure, cheekbones, nose, mouth, etc.) of the faces that have been cropped and aligned. We choose to exclude the cropped and landmark datasets since we do not expect the need to crop or align faces and do not need the landmarks data since we are not analyzing parts of the face. When going through the dataset, there were some images that were missing labels, such as having no race, or an arbitrary string value that was not expected in place of a label. It is important to have accurately labeled data to use for verifying model accuracy across all the labels. Therefore, we removed the mislabeled data from our copy of the dataset. While it could have been possible impute or fix the labels, we elected to simply remove the mislabeled data since our dataset is large enough that removing the mislabeled data still left us with a sizeable dataset.

5.2. Experimental Methodology

The product we created for this project is a containerized solution with three components: A front end GUI (Graphical User Interface) application for retrieving an image from the user, a component to find and crop a face from the user's image and a style transfer GAN for applying a style transfer to the cropped image of a face. The biggest concern from our stakeholders was that our project would not be able to perform a style transfer successfully on people of all ages, races, and genders without discriminating any one group. Finding an evaluation metric for this idea of success is hard because it is difficult to calculate how discriminating an image of a face is, and even harder to explain why an image of a face is inappropriate or offensive.

Our team could not find a straightforward way to evaluate the success of our model with previously used accuracy methods and knowing this, we created a subjective experiment to evaluate types of style transfers by having human evaluators decide if an image was a successful style transfer or not. We separated our UTKFace dataset into groups based on age and race and took a subset of 200 images from each of these groups. Our team separated the age groups based on child, adolescence, adult, and senior adults. We additionally created more specific groups from age 10 to 30 to accommodate noticeable changes in facial appearance throughout puberty. The groups for race were split following the UTKFace race split, with there being a group the following races: White, Black, Asian, Indian, Other (ex. Hispanic, Latino, Middle Eastern).

We performed the face cropping on the groupings and sent those images into different style transfers from the CycleGAN pretrained artwork models and had one team member evaluate on whether the style transfer was a success or not for each group. The goal of using CycleGAN pretrained models was to evaluate if the pretrained models can be used for facial style transfers and present any learnings we found about the outcomes. Our team choose to perform four different artwork-oriented style transfers for the experiment, based on the art styles of Cezanne, Monet, Ukiyo-e, and Van Gogh. The human observer looked through 3000 images across all the image groups of a style transfer model and determined an accuracy for each group in that style transfer. This was repeated for each pretrained model, but one model was considered a failure, so that model's 3000 images were skipped (see the results section for more information). The accuracy was determined by evaluating each image in a group as being a successful style transfer and not insulting in any clear way in comparison to its original image. The evaluator was asked to evaluate the images' success on how well the style transfer was applied to the image and whether the output image could be found offensive or insulting. This was repeated across all the style transfers. This is subjective way to analysis our results, but as stated above, it is hard, if not impossible, to determine if an adjustment to an image is insulting to anyone's demographic. Section 7 supplies more information about the shortcoming of this approach to evaluating accuracy and Section 8 supplies information on steps that could be taken to make this accuracy approach more robust.

5.3. Model Accuracy Results

From the tests run on each of the style transfer classes, the observer noted that the Monet model produced pixelated images when compared to the input. This was due to the model generating lower resolution images from the input, and due to that, the Monet model is excluded from the results. Figure 5.4 (below) shows the model accuracy for each age group.

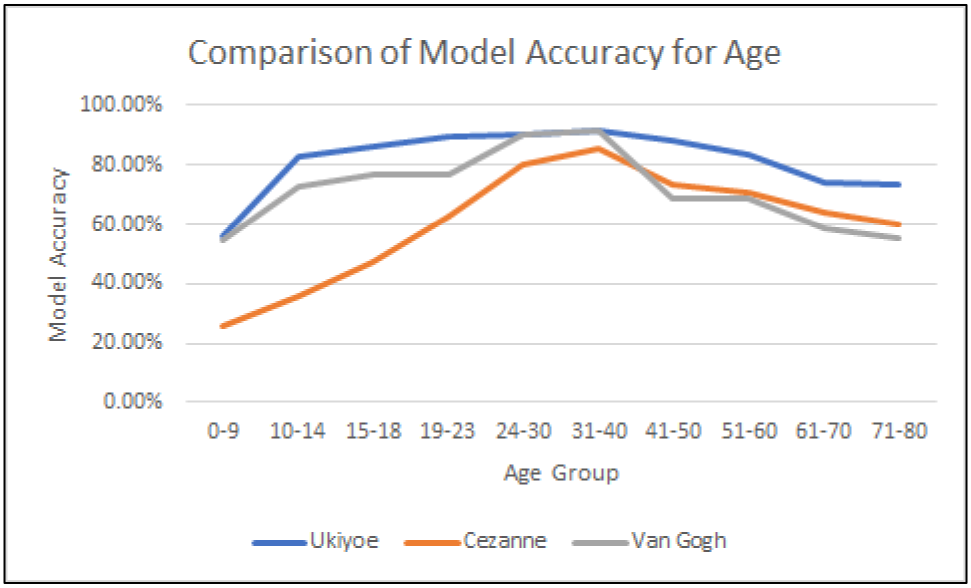

Figure 5.4: Model Accuracy for Age Groups - Divided by type of style transfer.

A few trends can be seen for the model accuracy in comparison to age from Figure 5.4. All the models suffered with the youngest age group, 0 to 9 years old, and the models increased in accuracy towards early adulthood where they peaked in performance with images in the 31- to 40-year-old classification. The models then decreased in accuracy as the age classifications increased. Cezanne overall performed worse than the other models and Ukiyo-e performed the best.

Figure 5.5 (below) shows the model accuracy in comparison to race. Once again, Ukiyo-e outperforms the other models for all races. It appears that all the models performed better on images of black individuals, but generally performs around the same for all other races, except for Ukiyo-e who outperforms on all races except for white.

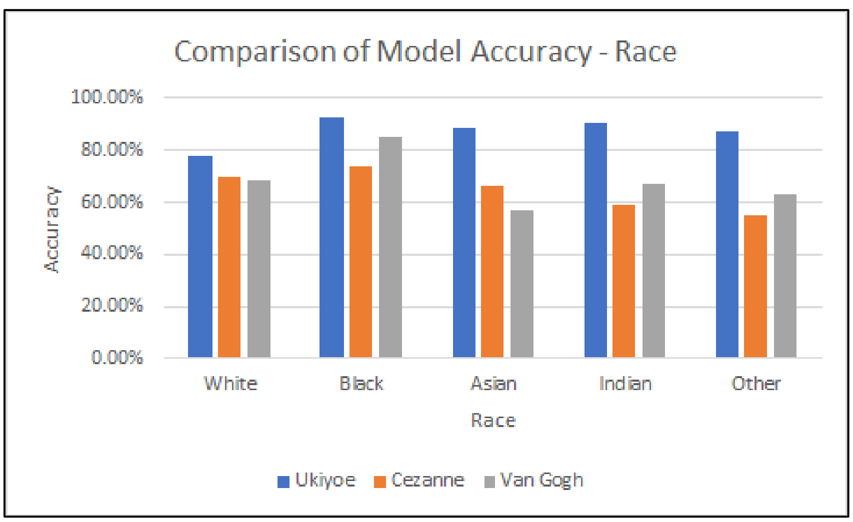

Figure 5.5: Model Accuracy for Race Groups

For a reference of what our model outputs look like, we have included images that were considered a success on all models (Figure 5.6) and images that were all considered to be failures

(Figure 5.7) from our three successful models. Figure 5.6 shows a high-resolution image that had been evaluated as a success from all style transfers and is one of the best examples of the transformations the observer could find. Figure 5.7 shows an example of a lower resolution image that was determined to be a failure for all models. Notice how in Figure 5.7, the last two style transfers from the Ukiyo-e and Van Gogh both clearly have too much emphasis on red coloring in the image. The human observer noted that if images had a shade of red anywhere on or near the face, it likely would fail due to the models overcompensating on the style transfer. The Cezanne image in Figure 5.7 on the other hand, looks like it was barely transformed which was why it was considered a failure.

Figure 5.6: From left to right - Original, Cezanne, Ukiyo-e, Van Gogh

Figure 5.7: From left to right – Original, Cezanne, Ukiyo-e, Van Gogh

6. Ethical Considerations

This project revolves around the model changing an individual's face. This brings up many problems that the stakeholders noted for this project. For example, changing skin colors, eye shapes, and genders could all be considered offensive. This means that we need to stay away from making changes to individual faces that could have unintended consequences for Discovery World. If someone is offended by the way their face was changed it could reflect negatively with Discovery World. If the project performs offensive changes to individuals in a certain demographic consistently, then the exhibit will likely be taken down to avoid further negative consequences for Discovery World. This case is very severe, as the possible implications of violating it could result in all our work being undone, with potential future repercussions.

Another ethical implication to consider revolves around the changes we are making to individual's faces, and whether the feature works equally for everyone. Much like with the prior point mentioned above, if one group is ostracized due to the project not detecting their faces, or not applying the changes correctly, then they likely will not find entertainment in the exhibit and will not be interested in Artificial Intelligence. This is problematic since the focus of the exhibit is to get the user interested with AI.

Knowing the ethical considerations for this project our team analyzed our models with these ethical considerations in mind. Our experimental setup to find a model's accuracy was created to address these ethical considerations by trying to assess whether a model was successful in its style transfer of an image based on two points: the success of the image transfer to an artwork style and if the style transfer was not insulting to any demographic group. While our experiment was flawed in many ways (which we address in Section 8), the core goal of the experiment was to consider these ethical implications in finding the accuracy of a model as well as developing a process that could be repeated with analyzing future models.

7. Conclusion

In conclusion, our final product accepts an image file and locates the face of an individual in it. It then crops the image around the face and sends that through our GAN network where we apply an Ukiyo-e painting style transfer on it. This modified image is then displayed back to the user. Of the four pretrained painting-style transfers that were available to us, the Ukiyo-e style produced the best looking and best quality results. Our model works well regardless of age and gender but performs better on individuals of black skin color and performs better on young adults. However, these results are subjective to the scorer and will not be judged the same by everyone. Because of these issues in our model, it would not be advisable to be put into place at Discovery World.

In examining model output, we found that lower resolution images generally failed for all models, while higher resolution images generally all passed, even if the style transfer was applied minimally. Another interesting finding was that our models encountered many out of RAM errors, resulting in us being unable to send many high-resolution images through the model. Additionally, as mentioned in the results and shown in Figure 5.7, many images failed due to having a color (typically red) take over the style transfer. It is important to note that all the above findings are subjective because they were found by our one observer, and ideally if we had more time, we would have had multiple observers look at the data and average the results. We also wanted to address that the idea of a style transfer being successful or not is a highly subjective question, and thus our results are skewed in the eyes of our one observer. The key takeaway of this project is that while the style transfers worked incredibly well on a select few high-resolution images, most of the time, the style transfer only worked partly, and could not be considered a true style transfer.

We believe that this inconsistency is largely in part due to the prebuilt models we were using being based on paintings of landscapes, buildings, etc. more so than people. While the artists that the models are based on, Monet, Cezanne, Van Gogh, and Ukiyo-e style, all have paintings which contain people, they are generally not the focus of the painting if there are people in them at all. The Ukiyo-e paintings most commonly feature people in them, which is one reason we believe this style transfer performed the best. Therefore, we concluded that while using prebuilt style transfer models can work for images of faces, it cannot truly be considered a success, and a future model would need to be trained specifically on faces for it to be more successful.

8. Future Work

There are quite a few future steps for this project. Training a style transfer model from scratch that is specifically trained on faces would likely increase model accuracy. This model could be trained on existing data, such as another painting style. Or it would be possible to create a training set from scratch that matches the output that you would want your model to have. In this case, you would not be limited by any shortcomings in the training data because you would create it yourself.

Another step for a future in this project would be to improve the accuracy determination process. Our determination of accuracy was very subjective for this project. This was partly due to the nature of the project but also due to its limited time frame. Even if the style transfer was applied correctly, it is still subjective whether the output looks “good” or not. Therefore, for any future development, a more robust way of deciding accuracy should be used. Defining a more rigorous definition of success and failure would help to limit the subjectivity of the process. A scorer should go through the entire testing set individually so there is no bias between different scorers. Also using more human scorers could supply a better estimate on the model's accuracy so no single scorer’s bias would affect the results. It is also possible that another method of determining success could be used. It may be possible to use the discriminator from the model to judge the generated output, however our team did not explore this possibility because a discriminator does not address ethical implications in its accuracy metric.

Future iterations of the MVP could also investigate using the previously mentioned StyleGAN and GANSpace [11] for image augmentation instead of style transfer. Our team investigated this route for our reach hypothesis but considered it to be too complex for the first iteration of our project. This architecture would allow for an individual’s face to be modified instead of changing the appearance of the entire image passed in. These augmentations will focus on the person’s appearance, and create augmentations such as adding or removing glasses, hats, earrings, or other small customizations. This will require the use of GANSpace to define and visualize vectors produced in the latent space of the StyleGAN model that perform the desired transformations. With desirable latent space vectors identified, we can then perform desirable transformations on input images. This will make sure that only small tweaks are made to the image to not produce any undesired results, but as mentioned in our earlier work, extensive auditing would still be required to determine the quality of the transformations.

9. Bibliography

[1] Z. Zhang, Y. Song and H. Qi, "UTKFace: Large Scale Face Dataset," 2017. [Online]. Available: https://susanqq.github.io/UTKFace/.

[2] "Discovery World: Science and Technology Center," [Online]. Available: https://www.discoveryworld.org/about/.

[3] "Docker," [Online]. Available: https://www.docker.com/.

[4] N. Chapman, K. Dowdle, N. DuPont, S. Harley and K. Rodrigues, "CycleGAN Server," 2020. [Online]. Available: https://github.com/NathanDuPont/CycleGAN-Server.

[5] "Flask: Web development, one drop at a time," Pallets, [Online]. Available: https://flask.palletsprojects.com/en/2.0.x/.

[6] "OpenCV," OpenCV Team, [Online]. Available: https://opencv.org/.

[7] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," 24 August 2020. [Online]. Available: https://arxiv.org/pdf/1703.10593.pdf.

[8] A. Radford, L. Metz and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," 7 January 2016. [Online]. Available: https://arxiv.org/pdf/1511.06434.pdf.

[9] Z. Zhang, Y. Song and H. Qi, "Age Progression/Regression by Conditional Adversarial Autoencoder," 28 March 2017. [Online]. Available: https://arxiv.org/pdf/1702.08423.pdf.

[10] R. Rothe, R. Timofte and L. Van Gool, "DEX: Deep Expectation of Apparent Age from a Single Image," 2015. [Online]. Available: https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/.

[11] E. Härkönen, A. Hertzmann, J. Lehtinen and S. Paris, "GANSpace: Discovering Interpretable GAN Controls," 14 December 2020. [Online]. Available: https://arxiv.org/pdf/2004.02546.pdf.

10. Appendix

The source code for this project is hosted publicly on GitHub [4]. This includes all code required to get the program working locally, as well as providing the Dockerfile used to create the Docker Image from the supplied code.