Generative AI for Humanity

Final Report

Supervisor: Prof. Rapheal Phan
Email: Raphael.Phan@monash.edu

(MCS19)

Name: Yan Hao Tan
Student ID: 30721318
Email: ytan0146@student.monash.edu

Name: Jun Koo Park
Student ID: 28399722
Email: jpar0007@student.monash.edu

Name: Nikhita Peswani
Student ID: 31361552
Email: npes0001@student.monash.edu

1. Font Matter

Word Count = 7134

Table of Contents

2. INTRODUCTION

Generative AI, such as Generative Adversarial Networks (GANs), refers to a technological approach that uses deep learning models to produce content that closely resembles human-generated content, such as images and phrases. This technology can generate responses to diverse and intricate prompts, including different languages, instructions, and inquiries [1]. In recent years, there have been notable breakthroughs in Generative AI. The progress in technology has resulted in significant enhancements in the capabilities of Generative AI, enabling it to generate content of exceptional realism, including artwork and photographs [2]. The images produced by Generative AI rely heavily on extensive datasets comprising human images, which leads to substantial privacy concerns. These privacy risks become even more pronounced when considering the potential outcomes of others exploiting these databases.

Therefore, it has become essential to anonymize the datasets in ways that preserve their utility and privacy. We aimed to describe a unique identity disentanglement technique which constitutes an anonymization process achieved by modifying the gender of the input image within the latent space and pixel space. The solutions we offer are part of our framework, which is designed to: identify and manipulate identity-relevant information in a face to produce an anonymized face and maintain nonidentity-related features (such as hair, background, and pose) without compromising the naturalness of the face.

Our methodology entails the integration of StyleGAN2 and Pix2PixHD, which guarantees the production of authentic content and the effective safeguarding of privacy. The proficiency of StyleGAN2 in generating synthetic data while preserving statistical traits by anonymizing serves as a valuable complement to Pix2PixHD 's capability to proactively translate AI-generated images into high-resolution, realistic representations. The generated output includes both an anonymized image and a numerical score indicating the extent of similarity between the anonymized image and the source image.

To quantify this similarity, we employed the cosine similarity measure, a mathematical method that assesses the resemblance between two vectors based on the cosine of the angle between them [3]. This measure serves as a reliable indicator of the degree of likeness between the anonymized and source images. A higher cosine similarity score signifies a closer match, while a lower score indicates greater dissimilarity. This approach not only provides an anonymized image but also furnishes a quantitative measure, allowing for a nuanced evaluation of the effectiveness of the anonymization process in preserving key features from the source image. It also emphasizes the ethical obligations of transparency and responsible data handling, thereby paving the way for a future where technological advancements and privacy safeguards can coexist harmoniously.

3. PROJECT BACKGROUND

3.1.Project Background

The goal of privacy is to keep personal information out of the hands of prying eyes (i.e., to prevent public access). It is regarded as a fundamental human right and is necessary for individualism, autonomy, and self-esteem. Invasion of privacy can have major effects for victims, such as loss of respect, job loss, and social disgrace. As a result, most companies that use personal data prioritize privacy protection [4].

The widespread use of generative AI techniques has led to the emergence of various privacy breach approaches that threaten individuals' privacy. AI can be used to compromise privacy by creating synthetic data. In a scholarly publication, the author presented a methodology for utilizing artificial intelligence (AI) to perform the task of re-identifying individuals across various social networks. The researchers successfully re-identified a significant proportion of the anonymized records, with a maximum re-identification rate of 85% [5]. The utilization of hyperparameters in the fine-tuning process of AI models can potentially heighten the risk of privacy breaches.

Therefore, our study primarily concentrates on the anonymization of image datasets, with a particular emphasis on facial anonymization. To safeguard the privacy of individuals, it is necessary to employ de-identification techniques to remove identifying characteristics. Also, the alteration of faces should be done in a way that prevents any association with specific individuals, and this process should be irreversible. A well-designed anonymization technique should also prioritize the preservation of data utility. [6]

In recent times, image privacy concerns have garnered greater attention due to the continuous developments in information extraction technology specifically designed for image data. At present, the primary emphasis of image privacy protection strategies is on reducing the potential for human observation. The primary focus of the specific methodologies is the execution of research that is based on the social connections of users and the subject matter of their photographs. The underlying principle of our design entails implementing identity disentanglement in the latent space before adjusting the facial attributes in the pixel space in a way that strikes the optimal balance between feature preservation and

anonymity assurance. For most of this manipulation, the latent space of StyleGAN2 is utilized [7].

3.2.Project Rationale

3.2.1 Face Anonymization

The initial investigations into data anonymization centered on safeguarding the confidentiality of categorical data, yielding many widely recognized methods for de-identification. A method called k-anonymity, which Sweeney proposed, is one such technique for de-identifying entries in a relational database. Expanding upon the concept of k-anonymity, the literature presents additional methods for anonymizing categorical data. Among these, the two most widely used are l-diversity and t closeness, as described by Ashwin Machanavajjhala, Gehrke, Kifer, and Muthuramakrishnan Venkitasubramanian and Li, Li, and Suresh Venkatasubramanian, respectively [8, 9, 10]. Initial attempts at face de-identification utilized ad-hoc methods including black-box, blurring, and pixelation [11].

While ad hoc methods can hinder the reidentification of a subject in a de-identified image by humans, they do not maintain the usefulness of the data and lack the strength to deceive recognition systems [12]. Due to these challenges, Generative Adversarial Networks (GANs) are regarded as a modern category of generative models that produce visually accurate synthetic images of arbitrary objects via adversarial training. Moreover, GAN projection algorithms enable the association of images with latent vectors in latent space, not just for the generation of facial images [13]. This enables the generation of a series of interpolated images between a source and target image, e.g., between two users. Therefore, by opting for StyleGAN2 and Pix2PixHD, renowned for their proficiency in generating detailed and high-resolution images, our approach effectively addresses the challenge of ensuring clarity in image generation.

3.2.2 Disentanglement using StyleGAN2

Generative adversarial networks (GANs), specifically the StyleGAN2 series, have gained recognition for their exceptional capacity to produce face images that exhibit a high degree of realism. These images are represented as latent vectors, and the distance between these vectors plays a significant role in shaping our perception and recognition of the created images [14].

The generation process of StyleGAN2 encompasses the utilization of various latent spaces, commencing with the Z space, which follows a normal distribution. The process involves the conversion of random noise vectors from the Z space to an intermediate latent space W via a sequence of completely connected layers. The W space is known for its capability to more accurately represent the disentangled characteristics of the learned distribution [15]. Notably, the authors of StyleGAN2 suggest searching for embeddings in the W space instead of W+ to identify images generated by StyleGAN2. Moreover, it is worth noting that StyleGAN2 was selected for exhibitions based on practical considerations. This choice was made owing to its ability to strike a balance between image quality and inference time, which aligns with the goals of optimizing time efficiency and enhancing the overall visitor experience during these events [14].

3.2.3 Disentanglement using Pix2pixHD

Pix2pix [16] emerged as one of the initial conditional generative models designed for image-to-image translations, focusing on learning the mapping between input and output images. The pioneering work by Chen and Koltun [17] introduced the first model capable of generating 2048x1024 pictures, paving the way for subsequent advancements such as pix2pix HD [18] and SPADE [19]. In SPADE's generator, each normalization layer modulates activations using the segmentation mask, making it particularly suitable for translating segmentation maps.

In the realm of image-to-image translation, the integration of Pix2pixHD with StyleGAN2 has garnered attention. This union brings forth a powerful combination, capitalizing on the strengths of Pix2pixHD's high-resolution synthesis capabilities and StyleGAN2's proficiency in generating realistic and diverse images.

3.2.4 Gender Classification using ResNet

Research in computer vision has concentrated on gender classification, especially when framed as a binary task to distinguish between males and females. Ng et al. [20] conducted a comprehensive survey, underscoring the significance of gender recognition as a pivotal demographic characteristic. Their study illuminates the involvement of various facial features—including hair, neck area, lips, and eyes—in the determination of an individual's gender.

Early methodologies in gender classification involved a neural network trained on a limited set of near-frontal face images. In another approach, the combination of 3D head structure (derived from a laser scanner) and image intensities was utilized for gender classification. More recent advancements include the utilization of the Weber Local Texture Descriptor, showcasing near-perfect performance on the FERET benchmark for gender recognition. leveraged intensity, shape, and texture features with mutual information, yielding impressive results on the FERET benchmark [21].

The adoption of ResNet in gender classification models is motivated by its unique architecture, organized into blocks, each comprising a shortcut and convolutional layers. This design allows the shortcut to

pass and integrate information from middle layers, eliminating the need for additional weights and ensuring consistent network performance. ResNet's effectiveness lies in its ability to address challenges such as vanishing gradients, exploding gradients, and network degradation, as highlighted in recent research [22]. Additionally, the adaptability of 1x1 convolutions in ResNet facilitates the maintenance of consistent input and output dimensions, contributing to its success in training deep neural networks.

3.2.5 Summary

Our model selection is guided by the advantageous characteristics of Pix2pixHD for producing high-resolution images and StyleGAN2's proficiency in generating diverse images. As opposed to our initially proposed methodology, we have opted not to include MaskGAN in our model due to its limited available information. Instead, we have integrated a Gender Classifier trained on ResNet18 to identify the gender of the input image. Leveraging StyleGAN2, we employ a strategic approach to swap the gender of the input image. This decision is driven by the desire to optimize image quality, enhance diversity, and efficiently address privacy concerns within the framework of our unique identity disentanglement technique.

4.Outcome

In this section, we delve into the outcomes of our project, which aimed to develop a generative AI model for enhancing privacy through identity anonymization using gender-swapping techniques. Here, we meticulously explore the implementation, results, and how the project's requirements were met. We also provide justifications for critical decisions, discuss the results in detail, and acknowledge the limitations of our outcomes.

Furthermore, we propose potential improvements and future directions, while engaging in a critical discussion throughout, to highlight the impact and significance of our four core features in advancing privacy preservation in AI technologies.

4.1.What has been Implemented and the Results Achieved

The four core features include: Input User Interface, Face Alignment, Gender Classifier, Gender Swapper.

4.1.1. Input User Interface

The core functionality of our AI model pivots on processing a single human image to anonymize identity by gender swapping. To facilitate this, we've developed a user-friendly interface, streamlined for ease of use, ensuring users effortlessly understand how to interact with it.

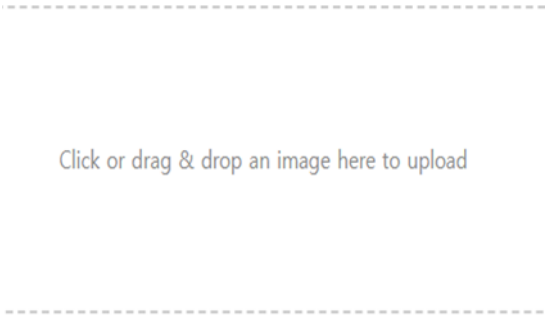


Figure 1: Input user interface

When an image is uploaded, it is stored in the 'Uploaded_Image' folder. Users can conveniently verify the upload by clicking on the image. Additionally, to assure users of successful uploads, the interface displays a confirmation message 'Image saved' once the image is correctly saved in the folder. This intuitive interface design significantly enhances the overall user experience and efficiency of the identity anonymization process.

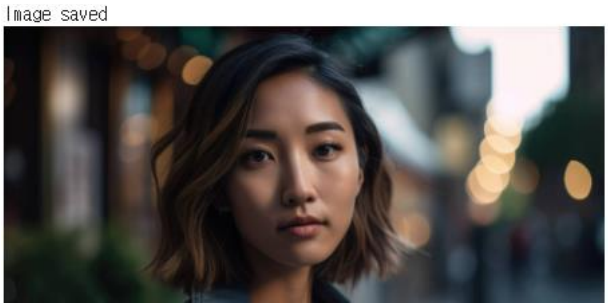


Figure 2: 'Image saved' shown when image saved successfully

4.1.2. Face Alignment

The Face Alignment feature plays a pivotal role in preparing the uploaded images for our gender classifier and swapper. It leverages a pre-trained dlib library model, 'shape_predictor_68_face_landmarks.dat', to accurately identify and extract facial features from the original image. This model transforms the image into a format suitable for further processing.

When the face alignment is complete, the system identifies the number of faces detected and saves the processed image into the 'Uploaded_Image' folder. The transformative effect of this feature is particularly notable in images with diverse backgrounds, aligning the faces to a standard format essential for the consistent performance of the gender-swapping algorithm. Figure 3 and 4 shows the face alignment of the image with a background.

Figure 3: Input image before face alignment

Figure 4: Input image after face alignment

4.1.3. Gender Classifier

Our model's success hinges significantly on the Gender Classifier feature. This crucial component automatically recognizes the gender of the input image, offering a binary prediction—male or female. We have employed the Resnet-18 CNN architecture for this classifier. During the training and validation phases, the model was refined using a substantial dataset comprising approximately 12,000 male and female images. Furthermore, we leveraged the celebA dataset, containing around 20,000 images, for testing the trained model.

The aligned input image is fed into this classifier, leading to the prediction of gender. The classifier has demonstrated impressive accuracy, ranging between 94% to 97%. The predicted gender, identified as either 'Male' or 'Female', is recorded for subsequent processing stages, reinforcing the model's ability to effectively handle critical gender-swapping tasks. Figure 5 and 6 demonstrate examples of prediction.

*Figure 5: Example of male image **(Left)***

*Figure 6: Example of female image **(Right)***

4.1.4. Gender Swapper

The Gender Swapper is the cornerstone of our model, designed to alter the gender of the original image. It operates by utilizing the gender classification data to load appropriate pre-trained weights for the swapping process. Our model integrates the advanced capabilities of StyleGAN2 for feature alteration and image generation, alongside Pix2PixHD for high-resolution output. These components have been fine-tuned using a dataset of over 20,000 images. Agile, and specifically Scrum, is well-suited for projects where the requirements and technologies are not entirely clear from the outset. In this project, there may be a need for experimentation and adaptation as new disentanglement techniques are explored and tested, which makes Agile's iterative approach a good fit.

Before being processed by the Gender Swapper, the aligned image undergoes an additional transformation. To demonstrate the efficacy of this feature, we present side-by-side comparisons of the input and output images, with the swapped image being saved for further analysis. This robust implementation underscores our model's innovative approach in redefining gender representation in images. Figure 7 and 8 show examples of gender swapper.

Figure 7: Male to female

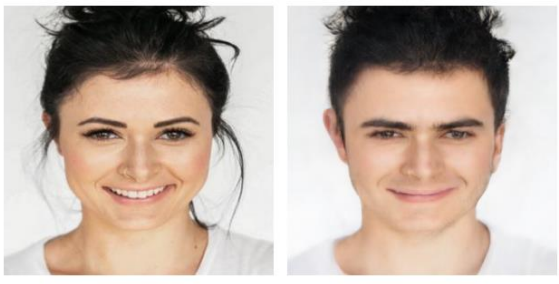


Figure 8: Female to male

4.2.How are requirements met

The primary aim of our project was to create an AI model dedicated to anonymizing personal identity. We specifically focused on gender alteration as our main feature, recognizing its potential to significantly alter the appearance in images, thereby effectively concealing identity.

This objective has been successfully achieved through the meticulous development and integration of five key features, each playing a vital role in the model's capability to transform and protect individual identity.

4.2.1. Input User Interface

This user-friendly interface effectively meets the requirement for easy interaction with our AI model. It allows users to upload images seamlessly, with a clear indication of successful uploads, thus facilitating the initial step of the identity anonymization process.

4.2.2. Face Alignment

The implementation of the Face Alignment feature fulfills the requirement for preparing images for gender classification and swapping. By accurately extracting and aligning facial features, this feature ensures that images are in the optimal format for further processing.

4.2.3. Gender Classifier

With a high accuracy rate of 94-97%, the Gender Classifier meets the critical requirement of correctly identifying the gender of the input image. This accuracy is vital for the subsequent gender-swapping phase.

4.2.4. Gender Swapper

The Gender Swapper, leveraging advanced StyleGAN2 and Pix2PixHD technologies, successfully achieves our goal of transforming gender presentation in images. This feature's ability to provide high-resolution, gender-altered images demonstrates the successful fulfillment of our core project objective.

4.3.Discussion of all results

4.3.1. Input file format

Image File Format	Result
JPG/JPEG	Image saved
PNG	Image saved
WEBP	Image saved
SVG	Image not saved
GIF	Image not saved

Table 1: Shows the result of different input image file formats

The model will only work on certain input image file type including WEBP, JPEG, JPG and PNG. Some input types will not work for the model such as gif, svg, etc. This restriction is implemented to manage image data appropriately within the specified functions, and it helps ensure that the processing and saving of image files function is as intended.

4.3.2. Detection of Facial features

Our results highlight the face aligner's efficiency in isolating and extracting only the facial features from an image, while disregarding other elements like background and body. This is evident in Figure 9, where a woman's face looking over her shoulder is precisely aligned, focusing solely on her facial features.

Similarly, Figure 11 depicts a man looking to the right against a clear background, with the aligner successfully isolating and saving only his face. These instances demonstrate the reliability and accuracy of our face aligner in handling various conditions and maintaining focus on the essential facial elements.

However, the detection might struggle with accuracy under extreme variations in lighting, angles, or facial coverings. Figure 9 to 11 are shown in Appendix A.

4.3.3. Prediction of Gender

The results demonstrate the gender classifier's skillful performance in analyzing complex facial features. For instance, Figure 12 depicts a woman with heavy makeup, a hat, and a tongue out, with typical female features like long hair obscured. Yet, our classifier accurately identifies her as female.

Similarly, Figure 13 presents a challenging scenario where a short-haired woman with glasses is successfully classified as female. Additionally, Figure 14 shows a man with long hair, who is correctly detected as male. These examples highlight our classifier's ability to discern gender accurately, even when presented with characteristics commonly associated with the opposite sex.

Given the nuanced and non-binary nature of gender, as well as the varied cultural expressions of gender, there is a possibility of misinterpretation with our binary classifier. Figure 12 to 14 are shown in Appendix B.

4.3.4. Swapping Gender

The results demonstrate our gender swapper's ability to modify distinct gender-specific features during the swapping process. For example, transitioning from male to female, it alters characteristics such as beards and thicker eyebrows. Conversely, swapping from female to male, it removes features like long eyelashes and double eyelids. These changes preserve the facial structure, as illustrated in Figure 17, where the gender

swap is successful even when the original photo shows a person looking over their shoulder.

Additionally, the gender swapper effectively handles images with accessories. Figure 16 exemplifies this by transforming a woman wearing glasses into a man, still retaining the glasses. This underscores the swapper's versatility in adapting to various scenarios and incorporating additional accessories.

Our gender swapper does not categorize long hair solely as a female trait. Consequently, in gender-swapping from female to male, it typically retains the hair length while adding masculine features. Figure 15 to 17 are shown in Appendix C.

4.3.5. Complete model results

In our comprehensive model evaluation, we utilized the celebA dataset, comprising approximately 20,000 male and female faces. Figure 18 and 19 show the model's output when processing an input image. For analytical purposes, we adopted the cosine similarity approach from Pytorch which is used to calculate similarity percentage. Figure 18 and 19 are shown in Appendix D.

Table 2 presents the similarity percentage and accuracy of gender-swapped images from the celebA dataset. The similarity percentage is about 61.34%, deemed reliable as our primary aim is not complete transformation but rather modifying key features that heavily influence identity. The original image's essential characteristics are preserved. Additionally, the model achieved an impressive accuracy range of 94-97% in gender identification on output images.

Dataset Type	Similarity %	Accuracy of Gender Swap
Male images from CelebA	61.34%	94%
Female images from CelebA	61.47%	97%

Table 2: Combined model test results

4.4.Justification of choices

4.4.1. Gender Classifier Architecture

The ResNet-18 architecture serves as the foundation for our gender classifier, chosen for its efficiency given our computational and human resource constraints. While architectures like ResNet-50 and AlexNet potentially offer higher accuracy, they also demand more computational resources, making them less suitable for our project's requirements [23].

ResNet-18 was selected for its shorter training duration, reduced likelihood of overfitting due to fewer layers, and its ability to provide good generalization. This balance of depth, complexity, and resource efficiency makes ResNet-18 an optimal choice for our gender classification needs.

4.4.2. Gender Swapper Network

Our gender swapper integrates the StyleGAN2 network and Pix2PixHD method. StyleGAN2 is particularly chosen for its proficiency in creating high-quality, realistic images, excelling in detailed facial feature generation and manipulation [24]. This capability is vital for subtle and effective identity anonymization. Compared to other GANs and networks like VAEs, StyleGAN2 stands out for its superior image quality and training stability.

As StyleGAN2 generates the output image, Pix2PixHD ensures that these images maintain high clarity and detail, even at larger scales. This combination optimally utilizes StyleGAN2's feature generation capabilities and Pix2PixHD's strength in producing high-resolution images, making it highly effective for tasks requiring maintained image quality and realism.

4.4.3. Dataset Selection

The core component of our model, StyleGAN2, is trained using the FFHQ dataset for its high-resolution images (1024x1024), surpassing many other datasets in detail and quality [25]. FFHQ's broad demographic representation, including age, ethnicity, and backgrounds, offers more variety than CelebA or LFW. Its realistic nature, with a spectrum of everyday faces as opposed to the celebrity-focused CelebA, helps reduce model bias.

Conversely, we utilize the CelebA dataset for gender classification training and testing. CelebA's extensive attribute annotations, especially gender labels, are vital for supervised learning [26]. It provides a more diverse range of poses and expressions compared to FFHQ's focus on high-resolution, frontal faces. Moreover, the larger size of CelebA means more training samples, enhancing the performance of gender classification.

4.5.Limitations of Project Outcomes

4.5.1. Limited Computational Resources

Throughout the project, due to restricted computational resources, impacting our ability to extensively evaluate its performance. These constraints primarily affected the testing speed and the scope of our evaluation, potentially leaving some aspects of scalability and high-load performance unexplored.

4.5.2. Biased Dataset

While we employed a popular and diverse dataset, its inherent biases in gender representation, cultural diversity, and possibly age and ethnicity have likely influenced our model's interpretation of gender. These

biases may affect the accuracy and fairness of the model's gender classification and swapping capabilities.

4.5.3. Quality and Accuracy Measurement

Quantitatively assessing the quality of our model's gender-swapped images was challenging due to the subjective nature of gender perception. Our approach involved using a similarity percentage to gauge anonymization accuracy. However, this metric doesn't definitively determine complete anonymization, as quality and accuracy assessments are subjective. Consequently, our testing may not fully reflect the model's effectiveness or fairness in diverse real-world scenarios, possibly limiting the thoroughness of our conclusions.

4.5.4. Selection of Feature Anonymization

Due to time and resource limitations, our model primarily focuses on gender swapping for identity anonymization. While gender is a key factor, other aspects like age and race also significantly shape identity. Currently, the model's capacity is confined to gender-based identity concealment, resulting in limited user options.

4.6.Discussion of possible improvement and future works

4.6.1. Enhanced Computational Resources

To overcome current limitations, future iterations could leverage more advanced computational resources. This would enable more rigorous testing and fine-tuning, particularly in high-load scenarios, leading to a refined understanding of the model's performance under various conditions.

4.6.2. Diversifying the Dataset

A key area of improvement involves diversifying the training dataset. Incorporating a broader spectrum of ethnicities, ages, and cultural backgrounds would significantly reduce biases and enhance the model's fairness and accuracy.

4.6.3. Development of Advanced Metrics

The development of more sophisticated and objective metrics for assessing the quality and accuracy of the model's outputs is crucial. This could involve research into novel ways of quantifying anonymization and image quality, offering a more standardized evaluation framework.

4.6.4. Ethical and Privacy Considerations

Future work should also prioritize the ethical implications of identity anonymization. Establishing guidelines and considerations around consent and privacy will ensure the responsible and ethical application of this technology.

5.METHODOLOGY

While our project's methodology deviated somewhat from the initial proposal, the fundamental process of receiving an input image from the user and producing an anonymized output remained unchanged. Final architecture of our model is shown in Appendix E as Figure 20.

5.1.Image Preprocessing (Face Alignment)

The initial step in our process involves the face alignment procedure, which is essential for preparing the input image by extracting facial features. This task requires handling various image types and transforming them for subsequent classification and gender swapping.

For facial feature extraction, we utilized 'shape_predictor_68_face_landmarks.dat', a pre-trained model from dlib [27]. This open-source library is renowned for its facial landmark detection capabilities. The face aligner employs this model to detect faces in the input images, as illustrated by the 68-point model shown in Figure 21. Once detection is complete, the number of faces identified is displayed, and the aligned image is saved for further processing.

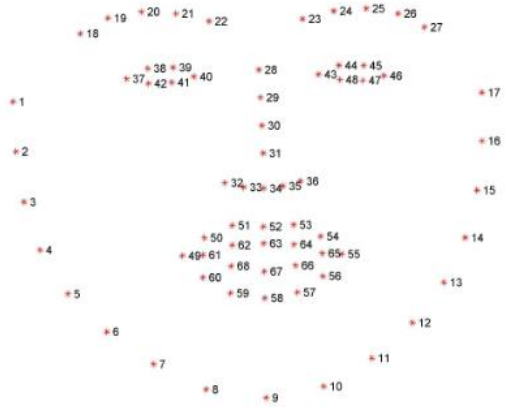


Figure 21: Example of Dlib's 68 points model [28]

In our updated methodology, a significant change is the move from converting the input image into a latent code for projection into the latent space, to now extracting features in vector form. This shift represents a new step in our process.

5.2.Gender Classification

In the second step of our process, the aligned image is fed into our gender classifier to determine its gender. Our final model, which aims at anonymizing identity through gender swapping, the role of gender classification is crucial. For this, we have trained the gender classifier using the efficient Resnet-18 architecture.

Resnet's methodology involves updating the weights during training based on the error function's

partial derivative. ResNet introduces residual blocks, where each block has skip connections that bypass one or more layers. These connections allow gradients to flow directly through the network, enabling efficient training of deep networks with 150+ layers [29]. Figure 21 shows the connection flow of Restnet.

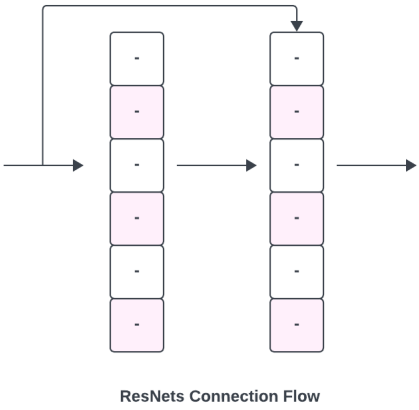


Figure 21: Resnet connection flow

The gender classifier is trained with over 10,000 male and female images annotated with its gender. In the training phase, Cross Entropy was used as the loss function, and Stochastic Gradient Descent (SGD) with momentum was chosen for its efficiency, especially with large datasets. We settled on 3 epochs for training, as additional epochs did not yield improvements in accuracy, suggesting that the model reached its learning potential within this timeframe.

5.3.Gender Swapper

In the third step, the aligned image, along with its gender classification, is processed by our gender swapper. The swapper then loads the appropriate weights based on the identified gender and executes the swapping process. This results in the generation of the output image, effectively completing the gender transformation.

Our final model, unlike the initial plan of projecting the image into the w+ latent space and generate the image, we integrated a separate pre-trained face classification network after realizing that StyleGAN2's latent space does not explicitly encode face attributes like gender.

Hence, for our final model, Stylegan2 was used to generate datasets for gender swapping. This dataset is created in seven detailed steps:

1. Generate random latent vectors $z_1 \ldots z_n$, map them to intermediate latent codes $w_1 \ldots w_n$, and generate corresponding image samples $g(w_i)$ with StyleGAN2.

2. Get attribute predictions from pretrained neural network f, $c(w_i) = f(g(w_i))$ as shown in the Figure 22.

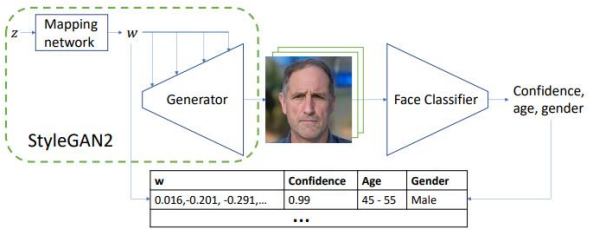


Figure 22: Method of finding correspondence between latent codes and facial attributes [7]

3. Filter out images where faces were detected with low confidence. Then select only images with high classification certainty. Low confidence helps to reduce generation artifacts in the dataset, while maintaining high variability as opposed to lowering truncation-psi parameter.

4. Find the center of every class $Ck = 1/n_{c=k} \sum_{c(wi)=k} w_i$ and the transition vectors from one class to another Δc_i, $c_j = C_j - C_i$.

5. Generate random samples zi and pass them through mapping network. For gender swap task, create a set of five images $g(w-\Delta)$, $g(w-\Delta/2)$, $g(w)$, $g(w+\Delta/2)$, $g(w+\Delta)$.

6. Get predictions for every image in the raw dataset. Filter out by confidence.

7. From every set of images, select a pair based on classification results. Each image must belong to the corresponding class with high certainty.

Figure 23 shows how the dataset is generated.

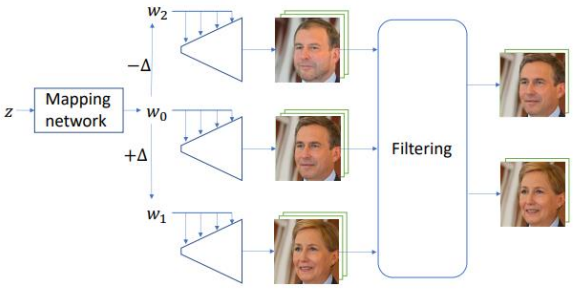


Figure 23: Dataset generation [7]

We begin by sampling random vectors z from a normal distribution. For each vector z, a series of images is generated along the vector Δ, each

associated with a specific facial attribute. Subsequently, from each set of generated images, the best pair is selected based on the results of a classification process. This methodical approach ensures precise attribute representation in the image generation process.

These steps show how the dataset for gender swapping can be prepared using StyleGAN2. Prepared dataset is then trained with pix2pixHD, an advanced image-to-image translation framework that specializes in high resolution image generation [7].

When Pix2pixHD and StyleGAN2 are used together, StyleGAN2 can generate detailed and diverse facial images, which Pix2pixHD can then translate into different styles or apply specific transformations at a high resolution. This combination leverages the strengths of both models: realism and detail of StyleGAN2's outputs and the high-resolution, detailed translation capabilities of Pix2PixHD.

5.4.Output Result Analysis

In the final step, the output image is evaluated using two methods: calculating the similarity percentage and classifying the gender of the output image.

As in project proposal, we continued to use cosine similarity to assess image similarity, but its application differs from our initial approach. Previously, lower similarity percentages indicated greater feature changes, implying better anonymization. Now, we aim for a balance, using similarity to ensure that while gender is accurately swapped, key features preserving the essence of the original image are retained, targeting a similarity percentage around 60%.

Additionally, we employ gender classification to validate the output image's gender, combining it with the expected similarity percentage to demonstrate effective identity anonymization by our model.

5.5.Conclusion

Our approach to identity anonymization has undergone a significant shift. While the initial method was not incorrect, the team collectively decided to focus on altering a single feature - gender - rather than changing all facial features. This strategic choice allows us to preserve certain key facial attributes, ensuring the essence of the original identity is maintained. This is in contrast to completely altering all features, which we realized is more akin to creating a random individual rather than anonymizing an existing one.

6.Software Deliverables

6.1.Summary of software deliverables

Please refer to Appendix G to J for the sample source code of our software deliverables.

6.1.1. Description of Deliverables

The software deliverables for this project include Jupyter Notebooks, compatible with Google Colab, designed for gender-swapping using generative AI to enhance privacy. These notebooks contain Python code using deep learning frameworks for image processing tasks like face alignment and gender swapping. The notebook will contain detailed explanations of the processing step, the code snippets, and visualizations of the output with its cosine similarity with the input image, providing transparency into the generative process. Table 3 in Appendix F has pinpointed all the libraries that have been used for the project.

6.1.2. Description of Usage

Our model, accessible in Jupyter Notebook, leverages state-of-the-art technologies in image-to-image translation, particularly specializing in gender swap transformations while prioritizing user privacy. The users will provide their input image and upload it into UI we created in the Jupyter notebook itself, it been shown in Figure 24.

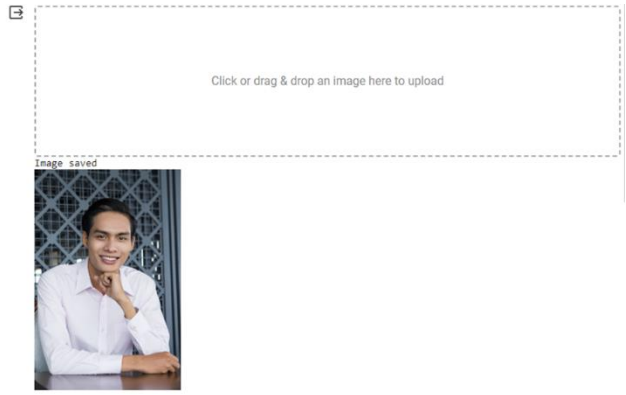


Figure 24: Image uploaded by user interface

The model streamlines the process of achieving gender swap transformations through a simple and efficient sequence of steps. Figure 25 shows the expected output generated from the input. The main features of the model included face alignment, gender swapping, gender classification, and similarity analysis.

Face alignment: Our model incorporates advanced face alignment techniques to ensure precise and accurate positioning of facial features. This step is crucial for maintaining the integrity of facial structures during the gender swapping process.

Gender Swapping: Like what been described above, this is the core functionality of our model, allowing

users to transform the gender attributes of facial images. The model intelligently modifies facial features while preserving privacy.

Gender Classification: Our model includes a gender classifier, enabling users to train and evaluate a model for gender identification. This classifier is used for selecting the model weight for gender swapping. For example, if the classifier predicted the image as female, the wight for female turn into male will be loaded. It is also a tool to analysis the success of the output, where the gender has been preserved after the image gender has been swapped by the model. Figure 26 shows the gender prediction in set of data.

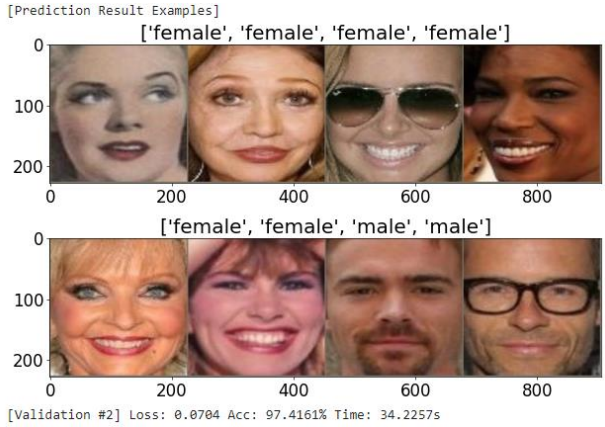


Figure 26: Gender prediction

Similarity Analysis: The model incorporates similarity analysis, specifically utilizing cosine similarity, to quantify the resemblance between the input and output images. This is also used for the outcome analysis to determine the success of the generated output.

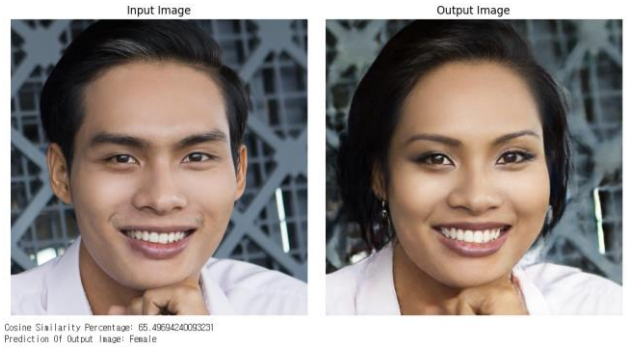


Figure 25: Final result of the software

6.2.Software Quality Handling

The software qualities, robustness, security, usability, scalability, documentation, and maintainability stand as key pillars. In the upcoming discussions, we will delve into each of these aspects, exploring strategies, best practices, and considerations to illustrate our approach to achieve high quality of our software deliverables.

6.2.1. Robustness

Thoughtful considerations in handling a wide range of input images address the robustness of the software deliverables. The model accommodates various image file types, explicitly supporting the commonly used formats such as jpeg, jpg, png, and webp. However, it is essential to note that gif and SVG file types are unsupported, and attempting to use them may result in incorrect behavior. This restriction is communicated to users, setting clear expectations about the acceptable file formats.

Despite the limitations on file types, the software deliverable maintains its robustness within the accepted image file types. The model demonstrates resilience in processing input images regardless of their size, offering flexibility to users. This capability contributes to the robust nature of the software, allowing for the analysis and transformation of images with diverse dimensions and resolutions.

Additionally, the software connects its robustness with the infrastructure it operates on, particularly Google Colab. Since the notebook is accessible within the Google Colab environment, the overall robustness of the software is influenced by the reliability and stability of Google Colab. Google Colab is recognized for its robustness and scalability, providing a secure and well-supported platform for running computational tasks.

6.2.2. Security

Our software deliverables address security with a focus on data handling and privacy preservation. Although security is not the project's primary emphasis, given the reliance on open-source models and datasets, fundamental security measures have been implemented to safeguard operations. A crucial aspect of this security approach involves carefully handling input and output data.

A robust strategy is in place for handling images processed by the gender-swapping model to ensure data confidentiality and user privacy. Google Colab cloud storage securely stores all images in a controlled environment that restricts access to other model users. This prevents unauthorized users from accessing or compromising the processed images. Utilizing secure cloud storage aligns with industry best practices for protecting sensitive data.

Furthermore, we have taken a proactive approach to limit the exposure of output images. Users can download the output images during their session, but a key security measure is in place. The notebook does not retain or make these images available once the user exits. This temporary availability ensures that processed images are not stored indefinitely, mitigating the risk of unauthorized access or unintended exposure.

6.2.3. Usability

The usability testing conducted on the software deliverables provides valuable insights into the user-friendliness and effectiveness of the User Interface (UI) and overall model usage. The 'End_User.ipynb' notebook, designed to require no additional setup procedures, facilitates a straightforward and intuitive user experience. This approach aims to align with user expectations, ensuring ease of use during model interactions.

The test involved four Monash and Sunway University participants, creating a diverse user group. The testing environment, set in an isolated environment within the campus library, simulated a distraction-free, real-world usage scenario. The participants were assigned specific tasks, from setting up the model to processing images and assessing the visual similarity between input and output images.

The usability test results showcase positive outcomes, with participants successfully completing tasks and providing constructive feedback. The test plan covered aspects such as setup, uploading images, processing steps, visual similarity assessments, and gender-swapping outcomes. Participants rated the usability positively, highlighting the success of the UI in terms of clarity and functionality.

Feedback from participants revealed notable observations, including the impressive nature of the gender-swapping capability and the potential for AI in image transformation. Some participants mentioned their unfamiliarity with Google Colab, suggesting a preference for a more user-friendly interface. These insights provide valuable considerations for future enhancements, emphasizing the importance of user familiarity with the tools employed.

Nevertheless, users unfamiliar with Google Colab might encounter challenges in navigating its interface and identifying the specific buttons mentioned in our guide.

6.2.4. Scalability

Scalability, a crucial aspect of software quality, is effectively addressed in the software deliverables using a Jupyter Notebook hosted on Google Colab. The inherent scalability of Google Colab, a cloud-based platform, ensures the software can handle varying workloads and user demands without compromising performance. As a result, the software deliverables benefit from the scalability features inherent in the underlying Google Colab infrastructure.

The choice of Google Colab as the hosting platform contributes significantly to the scalability of the software. Google Colab provides access to computational resources in the cloud, allowing users to execute code and run resource-intensive tasks seamlessly. The scalable nature of Google Colab enables the software to adapt dynamically to changing requirements, making it well-suited for processing diverse image inputs and accommodating users with varying computational needs.

However, the free version of Google Colab has limitations on GPU usage, which might affect performance with large datasets. For faster anonymization with substantial data, upgrading to the Pro version may be necessary.

6.2.5. Documentation and Maintainability

The documentation within the Jupyter Notebooks provides comprehensive and clear explanations of the code, processes, and functionalities. This meticulous documentation is a valuable resource for users, developers, and stakeholders, offering insights into the inner workings of the gender-swapping model and the associated gender classifier. It facilitates understanding and serves as a foundation for future development or modifications.

Including detailed test plans, participant feedback, and outcomes from usability testing demonstrates a commitment to documenting the testing process and the user-centric evaluation of the application. This user feedback, coupled with a thorough discussion of participant experiences, provides valuable insights for potential enhancements and improvements, contributing to the ongoing maintainability of the software.

6.3.Sample Source Code

The source code included face alignment, gender classification, gender swapping, similarity analysis will be provided in appendix for reference and transparency.

7.Critical Discussion on the Software Project as a whole

7.1.How well the project was executed

Our project was successfully managed and implemented, aligning closely with our initial plans. However, we encountered unexpected challenges, particularly in time management, due to the need to redo a core feature. This situation tested our adaptability and required a re-evaluation of our approach.

Additionally, the six-week summer semester imposed a challenging time constraint for our project. Learning the fundamentals and building a complete, application-level model in deep learning, a field that requires a strong understanding of mathematics, was a demanding task. This led to an unpredictable project trajectory, where continuous effort and commitment were essential, and the outcome was uncertain.

Despite these hurdles, we managed to complete our assessments on time. This experience highlighted the importance of flexibility and resilience in project execution, and how our strategic thinking evolved to navigate unforeseen obstacles. Please refer to section ***3. Outcome*** to read on how the project was executed.

7.2.Deviation from the initial project

In our six-week project timeline, we initially had a detailed plan focusing on efficient time management. However, an unexpected need to modify our core feature led to a significant disruption in our schedule, compelling the team to expedite the remaining tasks and assessments.

This not only altered our task distribution and time management strategies but also fundamentally shifted our approach to identity anonymization. While the basic concept of anonymizing identity while retaining key features remained constant, our methodology evolved considerably. For a detailed explanation of these changes, please refer to the ***4. Methodology*** section.

8.Conclusion

In summary, our project aims to tackle the crucial issue of anonymizing datasets generated by Generative AI, especially those containing human images. While Generative AI has made remarkable strides in creating lifelike content, the associated privacy concerns with vast datasets of human images are alarming. Our project emphasizes the pressing need to safeguard privacy while ensuring the continued utility of the data.

In our model, we address the challenge by using a ResNet18-trained gender classifier that accurately identifies gender in images, and we subsequently leverage the capabilities of StyleGAN2 and Pix2pixHD to modify the identified gender. Additionally, we employ the cosine similarity metric to quantitatively measure the resemblance between the input and output images, providing a robust and quantifiable measure of the anonymization process's effectiveness.

Our gender prediction model, a key component of our approach, has demonstrated remarkable accuracy, achieving a 97% and 94% success rate in tests conducted on a female dataset and male dataset respectively. This impressive result underscores the reliability and effectiveness of our model in accurately discerning gender attributes. Furthermore, the average cosine similarity percentage, calculated across a diverse dataset comprising both male and female images, stood at 61.34%. This outcome highlights our model's proficiency in retaining essential facial features and the crucial mask while successfully altering gender—a key objective of our project, aptly named 'Generative AI for Privacy Prevention.'

By combining cutting-edge technologies, our model represents a thoughtful and practical response to the ethical and privacy considerations posed by the evolving landscape of Generative AI. As technology continues to advance, our commitment to balancing innovation with responsible data handling remains at the forefront, fostering a future where privacy and technological progress coexist harmoniously.

REFERENCES

[1] Weng Marc Lim, Asanka Gunasekara, Pallant, J. L., Pallant, J., & Ekaterina Pechenkina. (2023). Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators. The International Journal of Management Education, 21(2), 100790–100790.

[2] Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., … Guo, B. (2022). StyleSwin: Transformer-based GAN for High-resolution Image Generation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr52688.2022.01102

[3] W. H. Gomaa, "A Survey of Text Similarity Approaches," vol. 68, no. 13, pp. 13–18, 2013.

[4] Majeed, A., & Seong Oun Hwang. (2023). When AI meets Information Privacy: The Adversarial Role of AI in Data Sharing Scenario. Retrieved October 30, 2023, from ResearchGate website: https://www.researchgate.net/publication/372570446_When_AI_meets_Information_Privacy_The_Adversarial_Role_of_AI_in_Data_Sharing_Scenario

[5] Ding, X., Zhang, H., Ma, C., Zhang, X., & Zhong, K. (2022). User Identification Across Multiple Social Networks Based on Naive Bayes Model. IEEE Transactions on Neural Networks and Learning Systems, 1–12. https://doi.org/10.1109/tnnls.2022.3202709

[6] Le, M.-H., & Carlsson, N. (2022). StyleID: Identity Disentanglement for Anonymizing Faces. Retrieved October 28, 2023, from arXiv.org website: https://arxiv.org/abs/2212.13791

[7] Viazovetskyi, Y., Ivashkin, V., & Kashin, E. (2020). StyleGAN2 Distillation for Feed-forward Image Manipulation. Retrieved October 28, 2023, from arXiv.org website: https://arxiv.org/abs/2003.03581

[8] SWEENEY, L. (2002). k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05), 557–570. https://doi.org/10.1142/s0218488502001648

[9] Ashwin Machanavajjhala, Gehrke, J., Kifer, D., & Muthuramakrishnan Venkitasubramaniam. (2006). l-Diversity: Privacy Beyond k-Anonymity. Retrieved October 30, 2023, from ResearchGate website: https://www.researchgate.net/publication/220964843_lDiversity_Privacy_Beyond_k-Anonymity

[10] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-Diversity: Privacy Beyond k-Anonymity," Research Gate, Jan. 2006. https://www.researchgate.net/publication/220964843_l-Diversity_Privacy_Beyond_k-Anonymity

[11] Li, N., Li, T., & Suresh Venkatasubramanian. (2007, May 15). tCloseness: Privacy Beyond k-Anonymity and l-Diversity. Retrieved October 30, 2023, from ResearchGate website:https://www.researchgate.net/publication/4251020_tCloseness_Privacy_Beyond_k-Anonymity_and_l-Diversity

[12] Ribaric, S., Ariyaeeinia, A., & Pavesic, N. (2016). De-identification for privacy protection in multimedia content: A survey. Signal Processing: Image Communication, 47, 131–151. https://doi.org/10.1016/j.image.2016.05.020

[13] Newton, E., Sweeney, L., & Malin, B. (2005, March). Preserving privacy by de-identifying face images. Retrieved October 30, 2023, from ResearchGate website: https://www.researchgate.net/publication/3297373_Preserving_privac y_by_de-identifying_face_images

[14] Karras, T., Laine, S., & Aila, T. (2020). A Style-Based Generator Architecture for Generative Adversarial Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–1. https://doi.org/10.1109/tpami.2020.2970919

[15] Isola, P., Zhu, J.Y., 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. pp. 1125–1134 (2017)

[16] Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: Proceedings of the IEEE international conference on computer vision. pp. 1511–1520 (2017)

[17] Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: Highresolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)

[18] Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2337–2346 (2019)

[19] C. B. Ng, Y. H. Tay, and B.-M. Goi, "Recognizing Human Gender in Computer Vision: A Survey," in PRICAI 2012: Trends in Artificial Intelligence, Berlin, Heidelberg, 2012, pp. 335–346, doi: 10.1007/978-3- 642-32695-0_31.

[20] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "StyleGAN2 distillation for feed-forward image manipulation," arXiv.org, https://arxiv.org/abs/2003.03581 (accessed Jan. 19, 2024).

[21] Y. Jiang, "Exploring The Efficiency of Resnet and Densenet in Gender Recognition," Highlights in Science, Engineering and Technology, vol. 38, pp. 1033–1037, Mar. 2023, doi: https://doi.org/10.54097/hset.v38i.5992.
the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2287–2296.

[22] Singh, P. (2023, November 22). *A brief dive into classic convolutional neural networks: Lenet, Alexnet, VGG, ResNet, InceptionNet...* Medium. https://medium.com/@2003priyanshusingh/a-brief-dive-into-classic-convolutional-neural-networks-lenet-alexnet-vgg-resnet-inceptionnet-4ce40fc25fb1#:~:text=Performance%3A%20AlexNet%20demonstrated%20the%20potential,for%20broader%20image%20classification%20tasks

[23] Brownlee, J. (2020, May 10). *A gentle introduction to stylegan the style generative Adversarial Network.* MachineLearningMastery.com. https://machinelearningmastery.com/introduction-to-style-generative-adversarial-network-stylegan/

[24] *FFHQ dataset: Usage in gan research and alternatives*. Datagen. (2023b, May 23). https://datagen.tech/guides/image-datasets/ffhq-dataset/#:~:text=Flickr%2DFaces%2DHQ%20(FFHQ)%20is%20an%20image%20dataset,%2DNC%2DSA%204.0%20license.

[25] *CelebA: Overview of datasets and a VAE tutorial*. Datagen. (2023a, May 23). https://datagen.tech/guides/image-datasets/celeba/#:~:text=It%20provides%20rich%20annotations%20for,and%20face%20attribute%20recognition%20models.

[26] Y Studios. (2020, September 29). *Insights: People: Y studios - what factors really influence identity?* https://ystudios.com/insights-people/influence-on-identity

[27] Shekharpandey. (2023, May 22). *Dlib 68 points face landmark detection with opencv and python*. Studytonight.com. https://www.studytonight.com/post/dlib-68-points-face-landmark-detection-with-opencv-and-python

[28] Classification of age and gender using ResNet - deep learning - philarchive. (n.d.). https://philarchive.org/archive/MANCOA-4

APPENDIX A

Figure 9: Image looking over the shoulder with background

Figure 10: Image wearing glasses with background

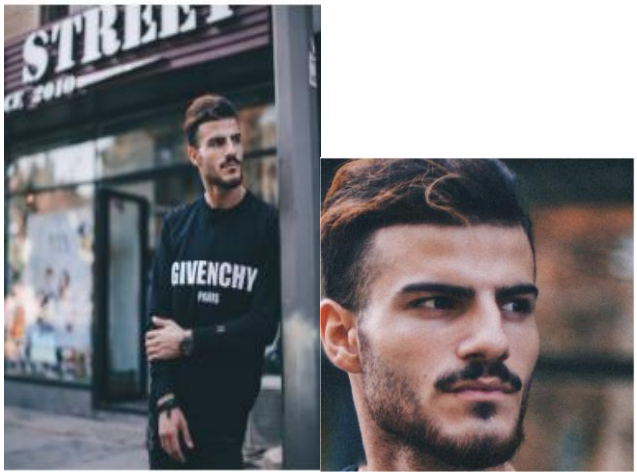
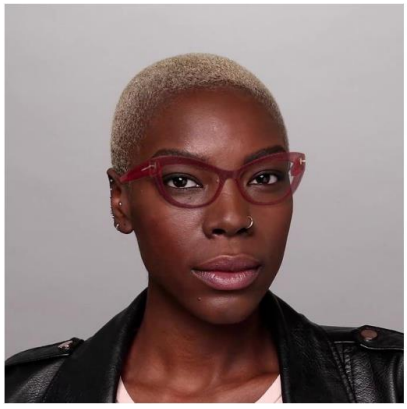


Figure 11: Image looking right with background

APPENDIX B

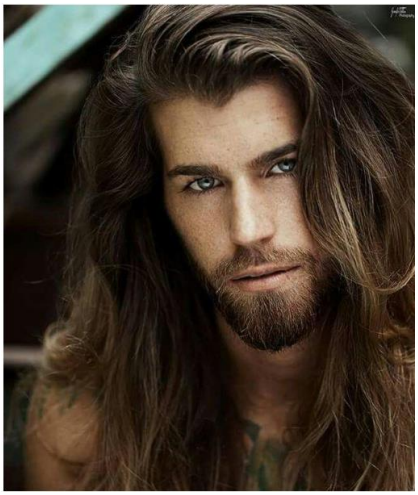
Predicted class: Female

Figure 12: women with heavy makeup and hat



Predicted class: Female

Figure 13: women with short hair and glasses



Predicted class: Male

Figure 14: man with long hair

APPENDIX C

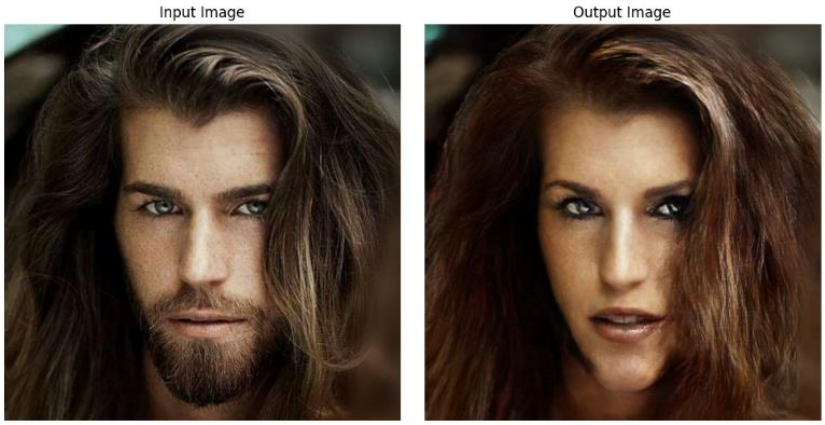


Figure 15: Male to female

Figure 16: Female to male with glasses

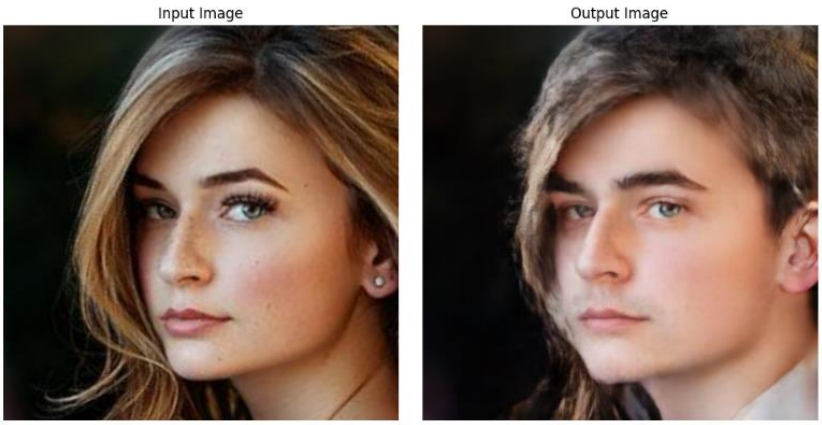


Figure 17: female to male looking over the shoulder

APPENDIX D

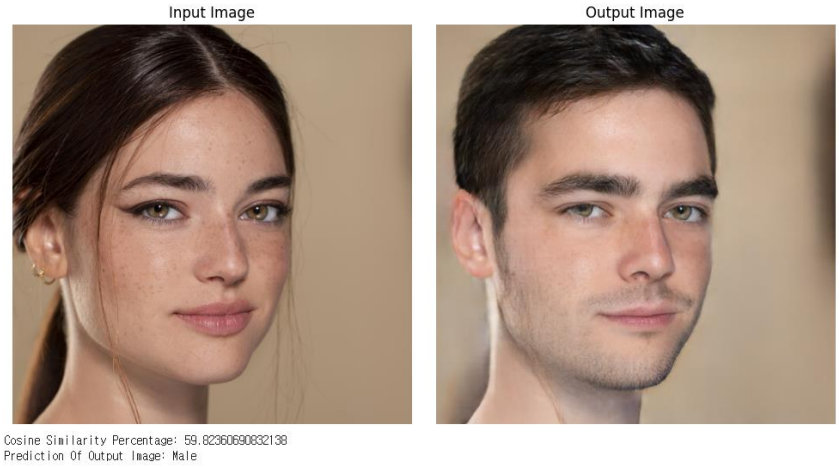


Figure 18: Result of female input image

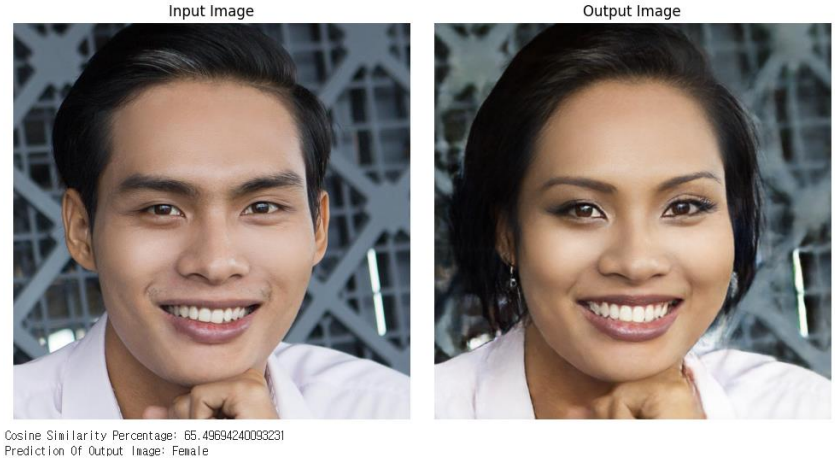


Figure 19: Result of male image input

APPENDIX E

Figure 20: Final proposed architecture of the model

APPENDIX F

Library	Description
torch	It is a fundamental library for building and training deep learning models, and it forms the backbone of our model. Used for defining and training neural networks, handling tensors, and facilitating various machine learning operations.
dlib	It is primarily utilized for facial landmark detection, aiding in accurate face alignment. Enables the identification of key facial features, crucial for precise alignment and subsequent facial attribute manipulation.
numpy	It is a powerful library for numerical operations, essential for handling arrays and matrices efficiently. Used extensively for processing and manipulating image data, especially in the context of face alignment and transformations.
matplotlib	It is employed for creating visualizations and plots, facilitating a clear presentation of results. Utilized to generate graphical representations of data, including visualizing gender-swapped images, similarity metrics, and classifier evaluation.
PIL	PIL, or Pillow, is utilized for image processing tasks such as opening, manipulating, and saving various image formats. Essential for loading and saving images, as well as applying transformations during the gender swap process.
imageio	Imageio is a versatile library for reading and writing images in different formats. Facilitates handling image files during various stages of the model, from input image loading to saving the final gender-swapped output.
requests	It is used for making HTTP requests, particularly for fetching external resources or datasets. May be used to download or access external datasets or models needed for training or evaluation.
bz2	The bz2 module provides functionality for working with BZ2-compressed files. Could be used for handling compressed files, depending on the requirements of the model or dataset.
shutil	Shutil is a utility module for high-level file operations. Used for tasks like copying, moving, and deleting files or directories.
aligner	The aligner module, though not a standard library, could be a custom implementation or external module used for face alignment tasks. Likely employed for aligning facial features during the gender swap process, contributing to precise transformations.

Table 3: Libraries used in the model

APPENDIX G

A. Face Alignment

```
# Define the paths
shape_model_drive_path = '/content/Final_Modal/shape_predictor_68_face_landmarks.dat.bz2'
shape_model_extracted_path = '/content/Final_Modal/shape_predictor_68_face_landmarks.dat'

# Extract the contents in Google Drive
with open(shape_model_extracted_path, 'wb') as new_file, bz2.BZ2File(shape_model_drive_path, 'rb') as file:
    shutil.copyfileobj(file, new_file)

# Load the model
shape_predictor = dlib.shape_predictor(shape_model_extracted_path)
```

Figure 21: Extract shape predict weight and load the model

```
# Image align with the trained landmark face shape predictor
aligned_img = align_face(img_filename, shape_predictor)[0]

input_image.jpg: Number of faces detected: 1
```

Figure 22: Align the image with the face shape predictor

APPENDIX H

B. Gender Classifier

```
# Import necessary modules and libraries for classification
import torch
import torch.nn as nn
import torch.optim as optim

import torchvision
from torchvision import datasets, models, transforms

import numpy as np
import matplotlib.pyplot as plt

import time
import os


device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu") # device object
```

Figure 23: Load modules and libraries for classification

```
transforms_train = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomHorizontalFlip(), # data augmentation
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]) # normalization
])

transforms_val = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

data_dir = './gender_classification_dataset'
train_datasets = datasets.ImageFolder(os.path.join(data_dir, 'Training'), transforms_train)
val_datasets = datasets.ImageFolder(os.path.join(data_dir, 'Validation'), transforms_val)

train_dataloader = torch.utils.data.DataLoader(train_datasets, batch_size=16, shuffle=True, num_workers=4)
val_dataloader = torch.utils.data.DataLoader(val_datasets, batch_size=16, shuffle=True, num_workers=4)

print('Train dataset size:', len(train_datasets))
print('Validation dataset size:', len(val_datasets))

class_names = train_datasets.classes
print('Class names:', class_names)
```

Figure 24: Load dataset and initialize class name for the classifier

```
model = models.resnet18(pretrained=True)
num_features = model.fc.in_features
model.fc = nn.Linear(num_features, 2) # binary classification (num_of_class == 2)
model = model.to(device)

criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)
```

Figure 25: Define model

```
num_epochs = 3
start_time = time.time()

for epoch in range(num_epochs):
    """ Training Phase """
    model.train()

    running_loss = 0.
    running_corrects = 0

    # load a batch data of images
    for i, (inputs, labels) in enumerate(train_dataloader):
        inputs = inputs.to(device)
        labels = labels.to(device)

        # forward inputs and get output
        optimizer.zero_grad()
        outputs = model(inputs)
        _, preds = torch.max(outputs, 1)
        loss = criterion(outputs, labels)

        # get loss value and update the network weights
        loss.backward()
        optimizer.step()

        running_loss += loss.item() * inputs.size(0)
        running_corrects += torch.sum(preds == labels.data)

    epoch_loss = running_loss / len(train_datasets)
    epoch_acc = running_corrects / len(train_datasets) * 100.
    print('[Train #{}] Loss: {:.4f} Acc: {:.4f}% Time: {:.4f}s'.format(epoch, epoch_loss, epoch_acc, time.time() - start_time))
```

Figure 26: Training Phase

```
""" Validation Phase """
model.eval()

with torch.no_grad():
    running_loss = 0.
    running_corrects = 0

    for inputs, labels in val_dataloader:
        inputs = inputs.to(device)
        labels = labels.to(device)

        outputs = model(inputs)
        _, preds = torch.max(outputs, 1)
        loss = criterion(outputs, labels)

        running_loss += loss.item() * inputs.size(0)
        running_corrects += torch.sum(preds == labels.data)

    epoch_loss = running_loss / len(val_datasets)
    epoch_acc = running_corrects / len(val_datasets) * 100.
    print('[Validation #{}] Loss: {:.4f} Acc: {:.4f}% Time: {:.4f}s'.format(epoch, epoch_loss, epoch_acc, time.time() - start_time))
```

Figure 27: Validation Phase

```
[Train #0] Loss: 0.1468 Acc: 94.6159% Time: 1830.8382s
[Validation #0] Loss: 0.0722 Acc: 97.5620% Time: 2567.7587s
[Train #1] Loss: 0.0869 Acc: 97.0601% Time: 3542.1155s
[Validation #1] Loss: 0.0720 Acc: 97.1500% Time: 3632.9338s
[Train #2] Loss: 0.0720 Acc: 97.5664% Time: 4531.9619s
[Validation #2] Loss: 0.0726 Acc: 97.4590% Time: 4614.4637s
```

Figure 28: Loss, accuracy, and processing time

```python
from PIL import Image
import torchvision.transforms as transforms

model.eval()

# Load the image using PIL
input_image = Image.open('/content/Final_Modal/Uploaded_Image/aligned_img.jpg')

# Define the preprocessing steps
preprocess = transforms.Compose([
    transforms.Resize(224),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]) # normalization
])

# Apply the preprocessing to your input image
input_tensor = preprocess(input_image).unsqueeze(0)  # Add a batch dimension

# Move the input tensor to the appropriate device (GPU if available)
input_tensor = input_tensor.to(device)

# Pass the preprocessed image through the model
with torch.no_grad():
    output = model(input_tensor)

# Get the predicted class
_, predicted = torch.max(output, 1)

class_names = ["Female", "Male"]

# Map the predicted class index to the class label
predicted_class = class_names[predicted.item()]
print("Predicted class:", predicted_class)
```

```
Predicted class: Female
```

Figure 28: Use classifier to predict the result

APPENDIX I

C. Gender Swapping

```
def get_eval_transform(loadSize=512):
    transform_list = []
    transform_list.append(transforms.Lambda(lambda img: __scale_width(img,
                                                                      loadSize,
                                                                      Image.BICUBIC)))
    transform_list += [transforms.ToTensor()]
    transform_list += [transforms.Normalize((0.5, 0.5, 0.5),
                                            (0.5, 0.5, 0.5))]
    return transforms.Compose(transform_list)

transform = get_eval_transform()
```

Figure 29: Image transform function

```
config_G = {
    'input_nc': 3,
    'output_nc': 3,
    'ngf': 64,
    'netG': 'global',
    'n_downsample_global': 4,
    'n_blocks_global': 9,
    'n_local_enhancers': 1,
    'norm': 'instance',
}

if predicted_class == 'Female':
  weights_path = '/content/Final_Modal/Gender_Weight/to_male_net_G.pth'  # to_male_net_G.pth
else:
  weights_path = '/content/Final_Modal/Gender_Weight/to_female_net_G.pth'  # to_female_net_G.pth

model = define_G(**config_G)
pretrained_dict = torch.load(weights_path)
model.load_state_dict(pretrained_dict)
model.cuda();
```

Figure 30: Generative model configuration

```
img = transform(aligned_img).unsqueeze(0)

with torch.no_grad():
    out = model(img.cuda())

out = util.tensor2im(out.data[0])
```

Figure 30: Process gender swapping

APPENDIX J

D. Similarity Analysis

```
# using torch
def find_cosine_similarity(input_tensor, generated_image):
  # Convert the NumPy array to a PyTorch tensor
  output_tensor = torch.from_numpy(generated_image).permute(2, 0, 1).unsqueeze(0).float()
  # Flatten the tensors to 1D vectors (if needed)
  tensor1_flat = input_tensor.view(1, -1)
  tensor2_flat = output_tensor.view(1, -1)

  # define a method to measure cosine similarity
  cos = torch.nn.CosineSimilarity(dim=1)
  output = cos(tensor1_flat, tensor2_flat)

  # display the output tensor
  return output

def get_similarity_percentage_using_torch(input_tensor, generated_image):
  cos = find_cosine_similarity(input_tensor, generated_image)
  percentage= ((cos.item() + 1) / 2) * 100
  return percentage
```

Figure 31: Cosine similarity analysis

APPENDIX K

Section	Person In Charge
Front Cover	Tan Yan Hao
Introduction	Nikhita Peswani
Project Background	Nikhita Peswani
Project Outcome	Park Jun Koo
Methodology	Park Jun Koo
Software Deliverables	Tan Yan Hao
Critical Discussion	All Members
Conclusion	Nikhita Peswani
Appendix & References	Park Jun Koo & Tan Yan Hao
Style & Presentation	Park Jun Koo & Tan Yan Hao

Table 4: Members Contribution (ANNEX)