Shallow Fakes

Adrian Mika Däschlein
ITU Guest Student (ID)
daeschlein@itu.dk

David Hark
ITU Guest Student (ID)
hark@itu.dk

***Abstract*—This study investigates the challenge of identity manipulation in facial editing, focusing primarily on conserving pose and expression information while transferring the identity from the source to the target image. The study investigates how both CNN-based and GAN-based models can address this challenge, prioritizing photorealism and identity-attribute tradeoffs. Key concepts include hierarchical feature encoding, image generation and loss optimization for identity preservation and visual quality. The VGGFace2 was chosen as a dataset given its broad range of both identities and images. After preprocessing to enhance the quality and reduce the size of the dataset, both models were trained numerous times. The CNN demonstrated limited photorealism and identity transfer, while the GAN model faces issues like style vector collapse and mask generation issues. Future work recommends curated datasets, improved loss functions and exploring temporal consistency for video applications.**

Index Terms— Face swapping, identity manipulation, RAFSwap, photorealism, attribute editing.

I. Introduction

Identity manipulation is a key challenge within the frames of facial attribute editing. The objective is to manipulate the perceived identity of a subject while maintaining other characteristics such as facial structure, lightning or background. Deepfake technology, which refers to synthetically generated media using advanced machine learning techniques, has gained increased attention for its potential applications ranging from entertainment content to ethical concerns regarding misinformation.

Earlier methods used convolutional neural networks (CNNs), while more recent advancements have been primarily focused on diffusion models or generative adversarial networks (GANs). These models are being developed to tackle challenges like identity preservation, attribute consistency or occlusion handling.

Korshunova et al. (2017) proposed one of the earliest CNN-based implementations for face swapping by framing the task as a style transfer problem. Their method involved the training of a feed-forward CNN to map the identity of a source face onto the attributes of a target face. The architecture utilizes the CNN in combination with preprocessing (alignment) and postprocessing (seamless cloning) for photorealistic face swapping. The method achieves high photorealism, though limited by a dependence on frontal views. The method demonstrates the potential for lightweight face swapping architectures such as CNNs [1].

In a later paper by Kim et al. (2022), a diffusion-based framework ("DiffFace") with facial guidance using pretrained models such as identity embedders, face parsers and gaze estimators. The framework introduces facial identity guidance during the diffusion process as well as a target-preserving blending technique to balance the identity-attribute tradeoff. The architecture achieved high photorealism, demonstrating its superiority over GAN-based methods regarding fidelity, controllability over the ID-attribute tradeoff and identity preservation [2].

Concerning GAN implementations, a study by Liu et al. (2023) proposed the E4S architecture, built on a GAN-inversion-based approach focusing on fine-grained control of facial attributes. The GAN inversion framework disentangles the shape and texture of facial components trough a mask-guided multi-scale encoder, feeding the swapped texture codes into a pre-trained StyleGAN generator with a mask-guided injection module to generate the swapped face. The model also enabled selective swapping of components, supporting various applications such as face beautification, hair transfer, or controlling the swapping extent of face swapping [3].

Another GAN-based implementation was proposed by Xu et al. (2022) with the RAFSwap, a GAN implementation utilizing transformer-based local and global feature branches. By specifically disentangling facial regions, RAFSwap preserves source identity while transferring the target's pose and expression. The hierarchical encoder ensures robust performance across a variety of face orientations and occlusions [4].

Concerning the evaluation of such models, a multitude of metrics can be utilized. Most authors resign to a qualitative evaluation, sometimes including a human study. Quantitative metrics include identity similarity, using embedding models like ArcFace or CosFace, as well as attribute distances for pose or expression. Lastly, also image quality can be assessed using metrics like Fréchet Inception Distance (FID).

II. METHODS

A. CNN Model

For the first model, we implemented a similar multi-scale transformation network, as proposed by Korshunova et al. [1] to perform identity-preserving face swapping while maintaining the target's pose, expression, and lightning.

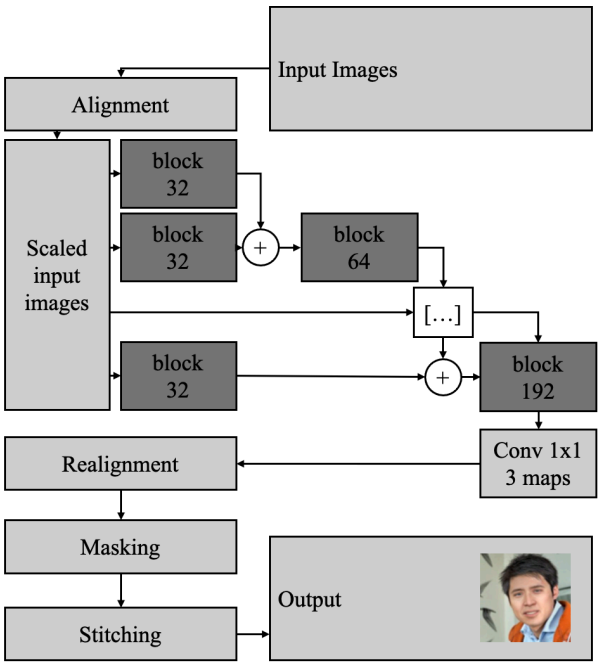


Fig. 1. **Architecture of CNN Model.** The pipeline begins with image alignment via facial landmark detection. Hierarchical residual networks at six scales (8x8 to 256x256) encode facial details. Outputs from these branches are upscaled and concatenated through a Multi-Scale Combiner. The combined features pass through a final convolution lawyer, yielding the transformed face. Masking the face of the target ensures fidelity to the target's pose and expression, while stitching integrates the transformed region into the target's background.

The architecture begins with aligning the source and target images using facial landmarks obtained via Mediapipes's face mesh model. This alignment ensures that both images are spatially consistent for the following processing. The aligned images are then fed into a hierarchical multi-scale architecture, with each branch handling a different resolution of the same image. Each branch processes features independently through a series of residual blocks, which capture localized identity-relevant details while preserving spatial consistency.

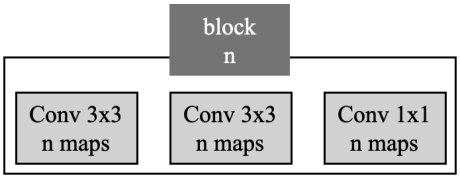


Fig. 2. **Residual Block used in Multi-scale Network.** The block consists of three sequential convolutional layers. The first two layers use a kernel size of 3x3, followed by ReLU activation and batch normalization. The final convolutional layer applies a 1x1 kernel. *n* refers to the number of feature maps at each convolutional layer.

The output of each branch is upscaled using nearest-neighbor interpolation and concatenated along the channel axis via a Multi-Scale Combiner. This step effectively merges features from all resolutions, facilitating the integration of both coarse-grained and fine-grained information. The combined features are passed through a series of convolutional layers to produce the final transformed image.

To ensure only region-based transformations, we introduce a Mediapipe-derived facial mask, which restricts the transformation to facial areas, leaving the background and non-facial regions untouched. This approach prevents artifacts and enhances the realism of the output image. It also shows an improvement to the hand-drawn masks proposed by Korshunova et al. [1]. The transformed features are blended into the target image using Poisson blending, for a natural integration of the facial region with the background.

The network is trained using a combination of loss functions that emphasize both visual quality and identity preservation. The content loss ensures the target's pose and expression are retained, while the style loss aligns the transformed image with the source identity. To maintain realism, a luminance consistency loss preserves the lighting conditions between the source and the target. A total variation loss is employed to encourage spatial smoothness in the output image. The losses are weighted and added to a total loss.

B. StyleGAN-based Model

For the second model, we chose to follow the proposed architecture by Xu et al. [4] for their RAFSwap. Figure 1 below shows the architecture of the RAFSwap model:

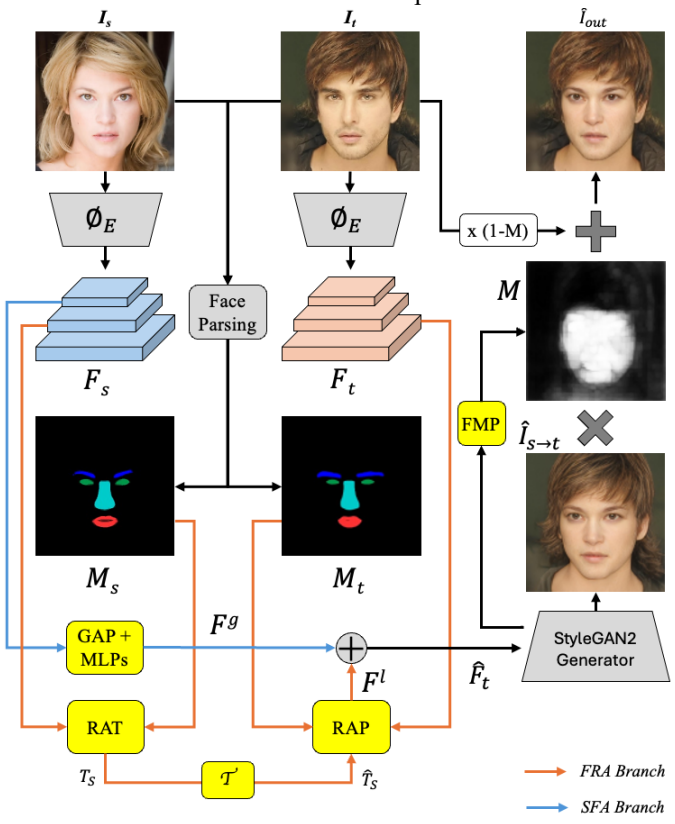


Fig. 3. **Architecture of RAFSwap.** Both the source face $\boldsymbol{I_s}$ and target face $\boldsymbol{I_t}$ are encoded hierarchically by a weight-sharing encoder $\emptyset_E$ to obtain $\boldsymbol{F_s}$ and $\boldsymbol{F_t}$. A pre-trained face segmentor creates four binary masks $\boldsymbol{M_s}$ and $\boldsymbol{M_t}$. The global *Source Feature-Adaptive Branch* in blue and the local *Facial Region-Aware* branch are used to integrate identity information of the source image with the target attributes of the target image in a local-global manner. The fused hierarchical feature $\hat{\boldsymbol{F}}_t$, the combination of $\boldsymbol{F^l}$ and $\boldsymbol{F^g}$, is mapped into different style vectors to control the generation process of a StyleGAN2. The *Face Mask Predictor* uses the GANs feature maps to produce soft masks simultaneously, The final output is the result of blending the target image with the generated image using the soft mask [4].

The *Region-Aware Identity Tokenizer* (RAT) of the regional branch indicated in orange, equipped with an attention mechanism, converts the source face features $\boldsymbol{F_s}$ into sets of important local identity-relevant tokens $\boldsymbol{T_s}$, with the shape $\mathbb{R}^{N\,x\,L\,x\,512}$. N is the number of feature map scales, being three, and L is the number of facial regions, in this case four (lips, nose, brows, and eyes). $\boldsymbol{F_s}$ and $\boldsymbol{M_s}$ are encoded using a region-wise average pooling layer $\emptyset$ to obtain the local semantic representations. Afterwards, each region's features are aggregated into a token, being embedded once more by a linear layer. The tokenizer operation can be defined as:

$$T_s^n = Linear\big(\emptyset(F_s, M_s^n)\big), \quad (1)$$
$$where\ M_s^n \in \{M_s^{lips}, M_s^{nose}, M_s^{brows}, M_s^{eyes}\}$$

Next, the transformer layer T models the interaction between the tokens over different scales, utilizing a Multi-head Self-Attention (MSA) layer, a Feed-Forward Network (FFN) as well as some Layer Normalization (LN). The final transformed tokens $\hat{T}_s$ can be formulated as:

$$\hat{T}_s = T_s + [MSA|FFN](LN(T_s)) \quad (2)$$

On the other side, the *Region-Aware Identity Projector* (RAP) tokenizes the source face's key regions (e.g., eyes, nose, and mouth) and updates the target face's feature map with this localized identity information, preserving the target's attributes like pose and expression. The combined masked target feature map F_t^m is updated using the token $\hat{T}_s$. After flattening F_t^m, using each scale as Query, and the tokens $\hat{T}_s$ as Key and Value, the identity-relevant tokens are transferred to F_t^m, which is then reshaped to match the size of Ft to allow combination:

$$F^l = F_t + RS\big(A^P \hat{T}_s W^P\big), \quad (3)$$

where A^P is the attention matrix computed using the Key, Query and Value, W^P is a learnable weight, and RS is a reshape operation.

The global branch in blue helps to avoid spatial misalignment between the source and target faces by passing the source feature map $\boldsymbol{F_s}$ with the smallest size trough a global average pooling (GAP) layer, followed by MLPs. The obtained global features is expanded to all three scales and then added to the local features F^l to obtain the final features $\hat{F}_t$:

$$F^g = MLPs\big(GAP(F_s^0)\big), \quad (4)$$
$$\hat{F}_t = F^g + F^l$$

The trainable components of the model include the RAP and RAT, the style vector extractors or the global branch, while the pre-trained components, such as the GAN, segmentor, and encoder, remain fixed. The model is trained using a combination of losses, including an identity loss comparing embeddings from a pre-trained model (ArcFace) between the source and output faces, an L2 reconstruction loss for pixel-level alignment when the source matches the target, and a perceptual loss (LPIPS) calculated using a pre-trained VGG16 network to evaluate semantic differences between the output and target.

In this paper, we employ two different GAN models: one where we implemented our own integration by building all trainable

parameters of the RAFSwap model from scratch, and another where we directly incorporated the pre-trained RAFSwap model provided in the authors GitHub repository [5].

III. Experiments

A. Data source

To train both models, the VGGFace2 dataset was used, after combining two separate downloads [6, 7]. Figure 2 below shows the data pre-processing steps:

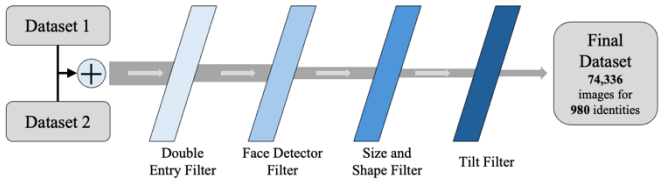


Fig. 4 **Pre-Processing steps**

The first dataset included 980 different identities with a total of 345,794 pictures. First, some images were duplicate entries, such as *n001/001_01.jpg* and *n001/001_02.jpg*, resulting from the face identifier detecting two images in one picture. If only one image was present, it was kept; however, if both entries were present, both were omitted. Next, a face detector was used to remove any images with zero or more than one detected face. A filter was then applied to ensure decent image quality by removing the lower 45% in terms of size. Additionally, a shape filter was applied to retain only images with a height-to-width ratio below 0.85 or above 1.15. Lastly, a tilt filter was applied, removing all images tilting too far in either direction. In the end, the dataset comprised 74,336 images for 980 identities.

B. CNN Results

The results for our CNN-based approach to face swapping could not show the results of the initial proposal of Korshunova et al. [1], which can be seen in Appendix D. Initial training runs yielded poor performance due to several reasons, including misaligned input images, inadequate masking, and insufficient training duration. This resulted in outputs that were far from photorealistic and could only recreate outlines of human faces.

Fig. 5. **Early results of own CNN**

Subsequent modifications significantly improved the performance of the CNN model. These changes included the proper alignment of all training images before model training and proper masking applications.

Fig. 6. **Results of final own CNN**

Further optimization of loss function weights also contributed to better results. In particular, the light loss was further finetuned. A final run over 120 epochs and loss weights of 1, 50, 100, and 10 for content loss, style loss, light loss, and total variation loss, respectively, showed the results in Figure 6.

Fig. 7. **Results of final own CNN with changes in loss weights**

Despite these improvements, challenges remained. The model struggled with details and extreme changes in facial expression, pose, and lighting. Figure 6 illustrates the ability of the model to transfer first identity features to the target face.

C. GAN Results

The pre-trained RAFSwap from the author's GitHub repository [5] was quite easy to implement, achieving the expected results, as shown in Figure 7 below:

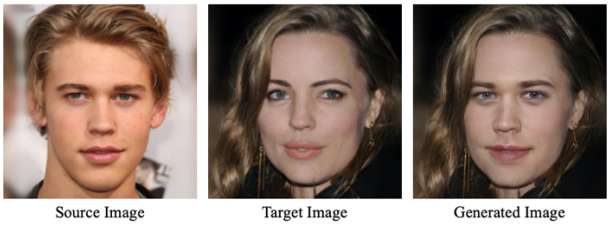


Fig. 8. **Results of pre-trained RAFSwap**

To also gain insights into the actual learning process, we rebuilt or own RAFSwap implementation, which achieved somewhat

different results. Usually, the different test runs had batch size set to 12, with each epoch's dataset containing 360 image pairs. Every fifth epoch copied the target image from the source image and enabled the reconstruction loss to enable the model to learn rebuilding the target images background and style. In total, nine different training runs where completed.

The first few runs yielded very bad results, probably because of too short training runs covering 5k to 10k pairs, whereas the authors trained their RAFSwap for a total of 50k pairs. The biggest issue however was that the style vectors passed to the GAN generator had a mean of around 0 and a very small standard deviation. This led to the generator always generating the *zero-face*, which is obtained after passing all zero style vectors to the GAN. This hindered any relevant learning, as the mask collapsed to maximize the perceptual loss. The results of an earlier model can be seen in Figure 8 below:

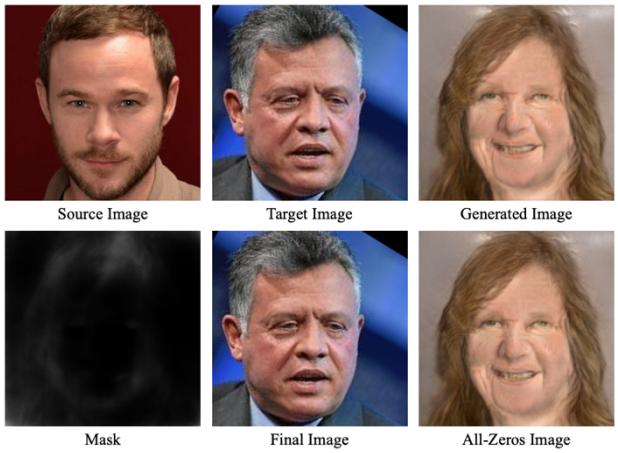


Fig. 9. **Results of early own RAFSwap model**

To combat the style vectors being close to zero and the mask also collapsing two zero, two further losses not mentioned in the report [4] but in the GitHub repo [5], namely the Wnorm loss as well a mask loss were added. The Wnorm loss ensures the style vectors take the same distribution as the latent space of the GAN generator. A multitude of mask losses such as a binary loss centering the values around a mean of 0.5 or a mask entropy loss were used. In the later stages, another mask loss based on the average facial region of the dataset (i.e. the area we would expect to be swapped) was used as a further condition in the mask generation.

Appendix A and B show the model outputs over 1000 training epochs of the 7th training run (without last mentioned mask loss) and the logged losses, whereas Figure 9 below shows the output of the 9th training run after 1000 epochs:

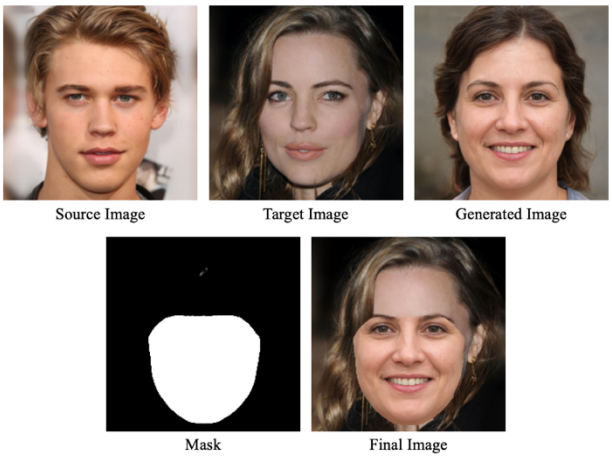


Fig. 10. **Results of final own RAFSwap model**

While the masks looked somewhat cleaner and the generated image before masking deviated from the "all-zero" face, the generated images still did not differentiate significantly from one example to the next.

IV. Discussion

The CNN implementation, modelled after Korshunova et al. [1] diverged significantly from the original results, which can be explained by several key aspects of their training procedure. Firstly, the original study utilized a very controlled and selective dataset. Korshunova's model was trained exclusively on two source faces- Nicolas Cage and Taylor Swift- both with highly curated datasets. Additionally, the training process involved handpicking subsets of images from a larger dataset that closely matched the target identity's pose and style.

In contrast our approach lacked such stringent curation. We relied on automated alignments, and employed no style-matching subset selection, leading to greater variability and inconsistency in the input data. Additionally, Korshunova's model leveraged a specific lighting loss term in their objective function, penalizing variations in illumination between the input and output images. This lighting loss - computed using a separate lighting network - enables their model to handle lighting conditions more effectively. Our implementation left our results more vulnerable to illumination inconsistencies.

Going forward, this model could still prove useful in controlled environments where the variability in pose, lighting, and expression is minimized, such as for specific applications requiring consistent datasets. Additionally, introducing dynamic control of the loss weights during training could enable the model to better balance identity preservation and

photorealism, adapting to different stages of training or specific challenges in the dataset.

Regarding the GAN implementation, using the pre-trained RAFSwap model proved to be straightforward, offering a great integration to produce substantial results with minimal effort. In contrast, our own implementation of the model did not achieve comparable success, likely due to several factors. Firstly, the datasets differed: The authors using the CelebA-HQ dataset, while we utilized the VGGFace2. Secondly, their model was presumably trained way longer, while we reached limitations around 1000 epochs, as well as financial constraints for renting out A100 GPUs. Other differences include model parameter initialization as well as variations in loss functions, particularly regarding the mask generation. Ultimately however, our inability to generate meaningful output from the GAN generator hindered progress; when the generator fails to produce even marginally distinct outputs for different examples, the losses cannot guide meaningful optimization.

Despite the challenges, integrating the model from scratch was a valuable learning experience. Notably, gaining deeper insights into the mask conditioning was quite insightful, seeing how the model evolved based on the losses. This iterative process allowed us to observe the interplay between different loss components and their impact on the model training. A notable issue that persisted during training, however, was the disparity between batch-averaged losses and individual example losses. While identity loss, for instance, appeared to converge to low levels on batch averages, single example losses often remained significantly higher. This discrepancy, potentially caused by batch normalization or aggregation, hindered the analysis of the model training.

IV. Conclusion

This study explored identity manipulation in facial editing using two fundamentally different architectures: a CNN-based model inspired by Korshunova et al. [1] and a GAN-based model built on Xu et al. [4]. The CNN model aimed to incorporate identity information from the source face into the target face while preserving pose and expression. Despite some improvements In training and loss fine-tuning, results remained rather limited in photorealism and style transfer, mostly due to insufficient curation or training images and architectural limitations compared to the authors model.

The GAN-based model following the RAFSwap architecture [4], highlighted the utility of using pre-trained models in achieving great outputs in a short manner. However, our own implementation of the RAFSwap architecture faced some challenges such as collapsing style vectors and mask generation issues, which ultimately resulted in poor differentiations between training runs. Introducing further losses such as a Wnorm or further mask losses somewhat alleviated the issues, but overall output remained subpar compared to the pre-trained model.

Concerning future work, efforts should be focused on introducing curated datasets for the CNN model, as well as adding further lightning loss components to enhance photorealism. The GAN model should rather be trained on the same dataset as the authors model, namely the CelebA-HQ. Further addressing the issue of locked style vectors disregarding the input images is also important. In a broader picture, exploring temporal consistency for video applications can widen the field of applications.

References

[1] I. Korshunova, W. Shi, J. Dambre and L. Theis, “Fast Face-Swap Using Convolutional Neural Networks”, *IEEE International Conference on Computer Vision (ICCV)*, pp. 3697-3705, 2017.

[2] K. Kihong, Y. Kim, S. Cho, J. Seo, J. Nam, K. Lee, S. Kim and K. Lee, “DiffFace: Diffusion-based Face Swapping with Facial Guidance”, *ArXiv,* abs/2212.13344, 2022.

[3] Z. Liu, M. Li, Y. Zhang, C. Wang, Q. Zhang, J. Wang and Y. Nie, “ Fine-Grained Face Swapping Via Regional GAN Inversion”, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8578-8587, 2023.

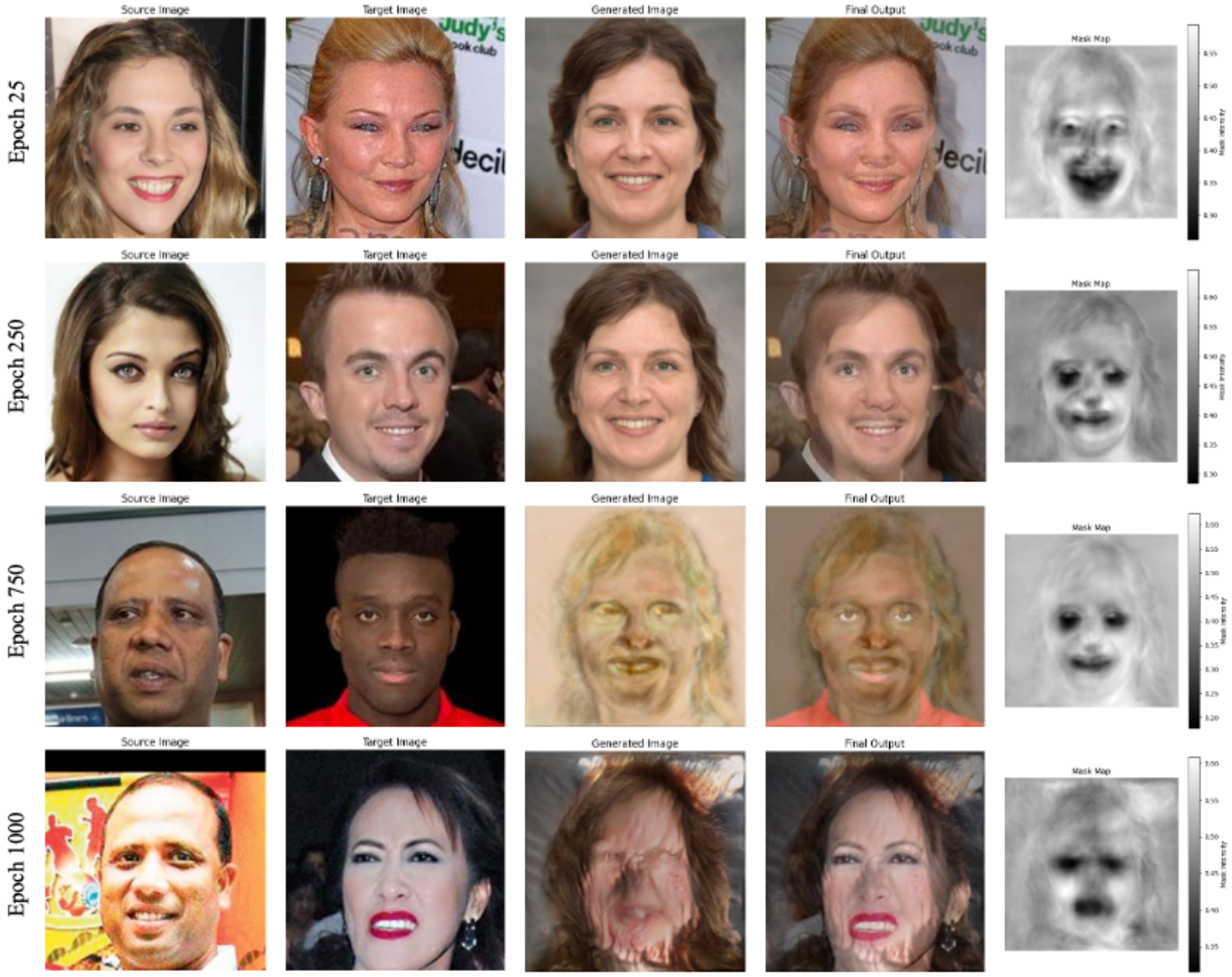
[4] C. Xu, J. Zhang, M. Hua, Q. He, Z. Yi and Y. Liu, “Region-Aware Face Swapping”, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7622-7631, 2022.

[5] xc-csc101, “RAFSwap GitHub Repository”, *Retrieved from:* https://github.com/xc-csc101/RAFSwap, 2022.

[6] hear fool and lisfool, “VGGFace2 Train”, *Retrieved from* https://www.kaggle.com/datasets/hearfool/vggface2?select=train, 2023.

[7] Greatgamedota, “VGGFace 2 Test”, *Retrieved from* https://www.kaggle.com/datasets/greatgamedota/vggface2-test, 2021.

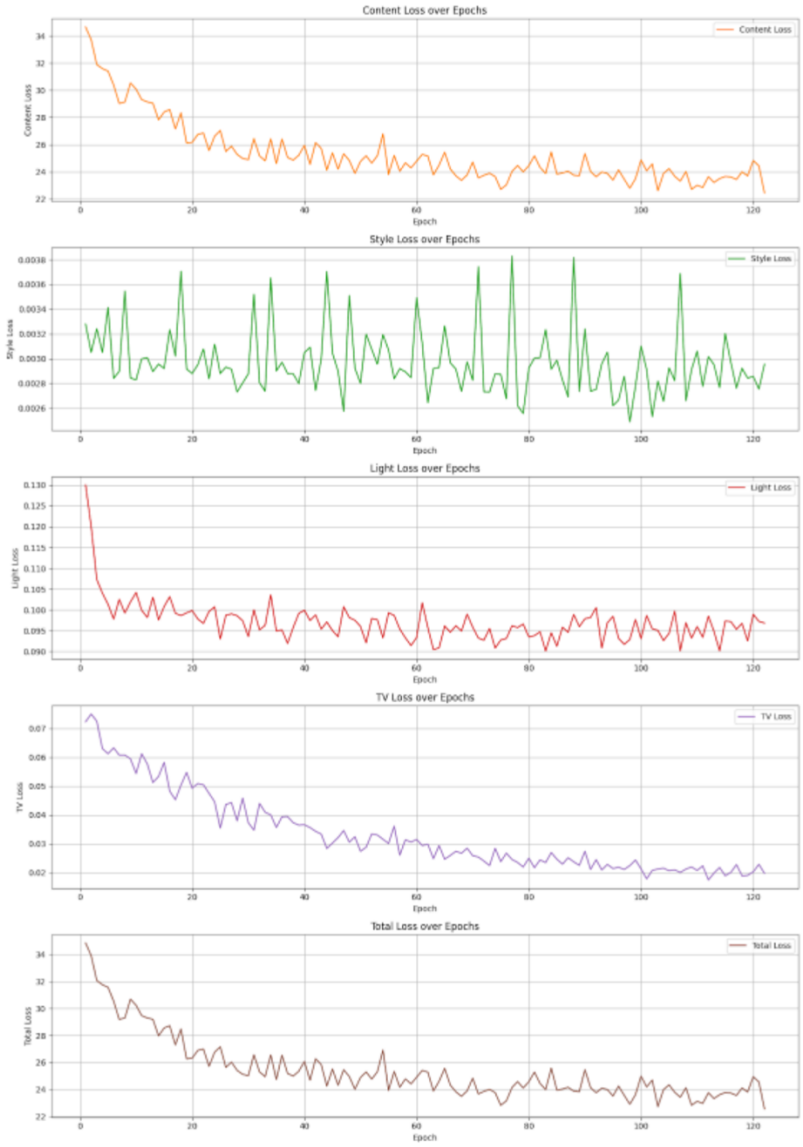
APPENDIX



App. A. *Results of 7th train run of own RAFSwap model*

Epoch	Total Loss	ID Loss	Reconstruction Loss	Perceptual Loss	Mask Loss	Wnorm Loss	Similarity Improvement
25	1.96	0.88	0.31	0.57	0.71	3.59e-05	0.0
250	1.08	1.07e-07	0.31	0.57	0.70	1.04e-05	0.0
750	1.07	0.0	0.29	0.57	0.69	2.81e-06	0.0
1000	1.08	3.73e-09	0.32	0.57	0.68	5.90e-06	0.0

App. B. *Results of 7th train run of own RAFSwap model – losses*



App. C. *Loss graph of 4th run*

(a) (b)

App. D. *Exemplary face-swaps by Korshunova et al. [1]*