Exploring Methods Used In Face Swapping

Joshua Eklund
eklun124@umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

Face swapping involves replacing the face in one image (the target) with a face in a different image (the source) while maintaining the pose and expression of the target face. Previous methods of face swapping required extensive computer power and man hours. As such, new methods are being developed that are quicker, less resource intensive, and more accessible to the non-expert. This paper provides background information on key methods used for face swapping and outlines three recently developed approaches: one based on generative adversarial networks, one based on linear 3D morphable models, and one based on encoder-decoders.

Keywords: face swapping, generative adversarial networks, encoder-decoders, linear 3D morphable models, Poisson blending, multi-band blending, face segmentation

1 Introduction

Face swapping involves replacing the face in one image with the face contained in another. Typically, there are two images: the target image and the source image. The goal of face swapping is to swap the face in the target image with the face in the source image such that the source face has the same pose and expression as the original face in the target image.

Face swapping is highly applicable to and widely used within the entertainment industry. Face swapping has been used to "resurrect" deceased actors or "de-age" them. For example, in *Rogue One: A Star Wars Story* face swapping was used to bring back the characters of Leia Organa and Grand Moff Tarkin (played by the deceased Carrie Fisher and Peter Cushing respectively). Face swapping was also used in season 2 of *The Mandalorian* to bring back a young version of Luke Skywalker.

Despite the seemingly innocent nature of face swapping, there exist concerns relating to privacy and misinformation. For example, a popular account on *Tiktok* with over 3.5 million followers (*@deeptomcruise*) has a series of videos featuring the famous actor Tom Cruise. However, the Tom Cruise in the videos is not the real Tom Cruise. Rather, the account owner(s) hired a Tom Cruise impersonator and swapped his face with Tom Cruise's. While some users were fascinated with how realistic the videos looked, others expressed concern over the potential abuse of the technology. What if this technology had been used to make a fake video of a politician or other important figure saying or doing something dangerous?

Adding to the potential dangers of this technology is how easy it is becoming to access and use it. Previous face swapping methods, such as those used in *Rogue One: A Star Wars Story*, used to require extensive knowledge, computer power, and man hours in order to produce convincing results. However, new methods of face swapping are being developed that require less computer power, time, and technical knowledge while still being able to produce quality results.

This paper provides the foundation for understanding the general face swapping process, methods used within face swapping, and examines three specific approaches to face swapping (one based on generative adversarial networks, one based on linear 3D morphable models, and one based on encoder-decoders).

Section 2 provides necessary information relating to how computers represent images. Sections 3, 4, 5, and 6 detail specific methods and technologies used in the three face swapping approaches discussed in this paper. Section 7 synthesizes the information of the previous sections to outline three specific approaches to face swapping. Section 8 compares the quality of the results of each approach, and Section 9 outlines the conclusions of this paper.

2 Computer Images

Images are stored in computers as a matrix of values. For grayscale images, there is one matrix. Matrix values, called pixels, are integers typically between 0 and 255, with 0 being black and 255 being white. A common way to represent colored images is by using three matrices (referred to as channels): one for each primary color of light (red, green, blue). To represent the fully colored image, the color channels are combined [13].

Computer images can be manipulated and transformed in several ways. For the purpose of this paper, we will discuss image operations important to Sections 5.

Two important image operations related to Section 5 are binary masks and the subtraction of images. A binary mask simply defines the particular region of interest (ROI) of an image. The mask contains pixels that have a value of either 0 or 1, with a value of 0 (typically black) meaning that the pixel is not in the ROI and a value of 1 (typically white) meaning the pixel is in the ROI. Image subtraction refers to

the operation of subtracting the values of the pixels in one image from the values of the pixels in another (pixelwise operation). For two colored images, the subtraction is done for each channel [12]. Negative pixel values are handled differently depending on the image format used and the operators used within the subtraction. For example, some implementations may set pixel values to 0 if the subtraction is negative, or the operator used may wrap the pixel values so that a pixel value of -30 would be wrapped to 226 [1].

3 Neural Networks

All three of the approaches discussed in Section 7 utilize neural networks in some capacity. Neural networks are a subset of machine learning, which involves building algorithms that take in a dataset as input, train themselves to recognize patterns in said data, and then utilize those discovered patterns to apply their functionality to the input dataset and other similar datasets. As such, neural networks take in a dataset as input, perform a mathematical calculation, and produce an output. Additionally, a subset of the input is used to train the network. For example, a neural network could be created to identify cars within an image. The network would take in several images featuring cars, perform a mathematical calculation that determines how it identifies said cars, and output the images with the cars highlighted. The specifics of how neural networks perform said calculation is unnecessary for understanding the rest of this paper. What is important to understand is that neural networks become better at performing their specified function by iterating over the training dataset numerous times (each complete iteration is called an epoch) and optimizing parameters within said calculation. For more information regarding the specifics of neural networks, please read [5].

4 Face Segmentation

A crucial step in some face swapping approaches is face segmentation, which involves partitioning the pixels of an image into two regions: a region containing all the pixels associated with faces and a region containing all non-face associated pixels. The end result of face segmentation is a mask that represents all the visible portions of the face in the image [9]. Looking at Figure 1 as an example, we have an image of a man wearing glasses in the left panel. Face segmentation is performed and produces a mask that represents the pixels associated with the visible portions of the man's face (the red pixels in the right panel).

Several methods exist for performing face segmentation, including those based on neural networks. However, going into the specifics of these techniques is beyond the scope of this paper. For more information regarding face segmentation techniques please read [9].

Figure 1. Face segmentation example [10]. Red pixels in the right panel represent visible portions of the face in the left panel.

5 Image Blending

Image blending refers to the image composition method of seamlessly blending a source image into a target image. Usually, an object from the source image is cropped and then pasted into the desired region in the target image. The challenge is to then adjust the appearance of the cropped object such that it matches the rest of the environment in the target image and to make the cropping boundary appear seamless [17]. This problem is highly relevant to face swapping, as one must ensure that the swapped face is appropriately blended into the target image for convincing results. In the remainder of this section, we present two methods of image blending used by the face swapping approaches discussed in Section 7: multi-band blending and Poisson blending.

5.1 Multi-band Blending

Multi-band blending utilizes Gaussian and Laplacian pyramids. A Gaussian pyramid is essentially a hierarchy of images that have been blurred and reduced in size. The left-most hierarchy of images in Figure 2 is an example of a Gaussian pyramid. To construct a Gaussian pyramid, let F be a Gaussian filter (used to blur images) and G_0 be the original image. If i represents the current level of the pyramid, then let G_i be the blurred and downsampled by a factor of 2 (halved in size in both the x and y dimensions) version of the image in the level below i. G_i can be represented mathematically by the following equation (where $*$ represents a mathematical operation known as convolution and $\downarrow$ represents downsampling):

$$G_i = (F * G_{i-1})_{\downarrow 2}$$

The Laplacian pyramid of an image is constructed using its Gaussian pyramid. Let L_i be the Laplacian of the image at level i. L_i is constructed by upsampling the Gaussian pyramid image at level $i + 1$ by 2 (increase size by 2 in both the x and y dimensions) and subtracting it from the Gaussian pyramid image at level i. This process is represented visually by the middle column of Figure 2 and mathematically by the following equation (where $\uparrow$ represents upsampling):

$$L_i = G_i - (G_{i+1})_{\uparrow 2}$$

This equation is essentially subtracting each Gaussian image G_i by a blurred version of G_i. Subtracting an image by a blurred version of itself results in a new image that

captures the edges of the image, i.e. the Laplacian. The rightmost hierarchy of images in Figure 2 shows an example of a Laplacian pyramid. The original image can be reconstructed from the Laplacian pyramid by summing each image of the pyramid upsampled to its original size.

To composite a source image into a target image, Laplacian pyramids are constructed for both the source image L^S and the target image L^T. A Gaussian pyramid G is constructed for the mask of the region in the source image that is to be copied into the target image. A Laplacian pyramid of the composited image L^I is constructed using the following equation (where i is the current level of the pyramid):

$$L_i^I = G_i L_i^S + (1 - G_i) L_i^T$$

The final composited image is then constructed from L^I. At higher levels of G, mask pixels are blurred so they may be shades of gray (between 0 and 1) rather than black or white. As such, at higher levels of G, when you are constructing L^I you are taking a weighted average of L^S and L^T, producing a blend of the two. This smooths the boundary between the source object and the target image. The method described in this section assumes that both images are grayscale. For colored images, this process must be done for each color channel [12].

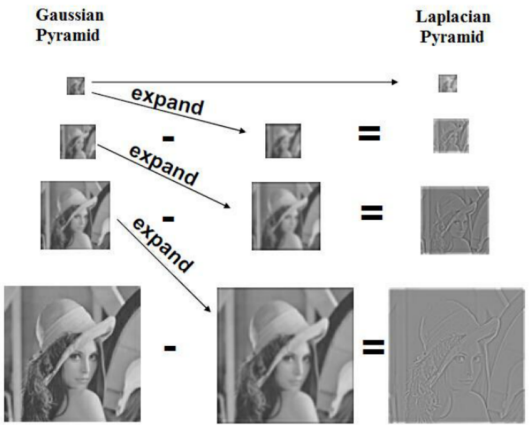

Figure 2. Example of Gaussian and Laplacian pyramids [4].

5.2 Poisson Blending

To begin explaining how Poisson blending works, we can examine Figure 4. Figure 4a is the target image (beach scene) and 4b is the source image (plane scene). 4c shows the object from the source image (the planes) that we want to paste into the target image. 4d highlights the region in the target image that we want to paste the planes into. 4e shows the composited image using multi-band blending. Although the cropping boundary between the planes and the target image has been smoothed, there is a color mismatch between the sky surrounding the planes and the sky of the rest of the target image. This type of coloring/lighting issue is a limitation of multi-band blending and is what Poisson blending attempts to fix. 4f demonstrates the final composited image using Poisson blending.

To resolve the color mismatch between the source and target images, Poisson blending operates in the gradient domain [12]. Image gradient refers to the directional change in the lighting or color of an image. Image gradients can be used for the detection of edges in images. Pixels with the largest gradient values in the direction of the gradient are identified as possible edge pixels [15]. Figure 3 provides an example of image gradients and how they can be used for edge detection.

To composite a source image into a target image using Poisson blending, let Ω be the region of the source that we want to paste into the target. In Figure 4, Ω would be the region encompassed by the circle in 4c. Let $\partial\Omega$ be the boundary of Ω. The goal of Poisson blending is to get the gradient of the composited image inside Ω to be as close as possible to the source image's gradient while having the composited image match the target image on the boundary $\partial\Omega$. If $C(x, y)$, $S(x, y)$, and $T(x, y)$ represent the pixels of the composited image, source image, and target image respectively, this problem can be represented mathematically by the following equation (where ∇ is the gradient operator):

$$\min_{C(x,y)\in\Omega} \iint_{\Omega} \|\nabla C(x, y) - \nabla S(x, y)\|^2 \, dx \, dy$$
$$\text{s.t. } C(x, y) = T(x, y) \text{ on } \partial\Omega$$

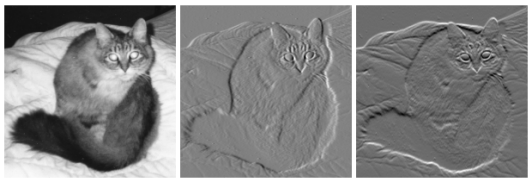

Figure 3. Example of image gradients. The middle and right images are the horizontal and vertical measures of the gradient respectively. White or black colored pixels represent a large gradient value, indicating a possible edge [15].

6 Generative Models

Generative models are a class of statistical models that are capable of creating new instances of data that resemble an already existing dataset [2]. The remainder of this section discusses three types of generative models that are utilized in the face swapping approaches discussed in Section 7.

6.1 Generative Adversarial Networks

Generative adversarial networks (GANs) are a type of generative model that utilize an adversarial process to train the

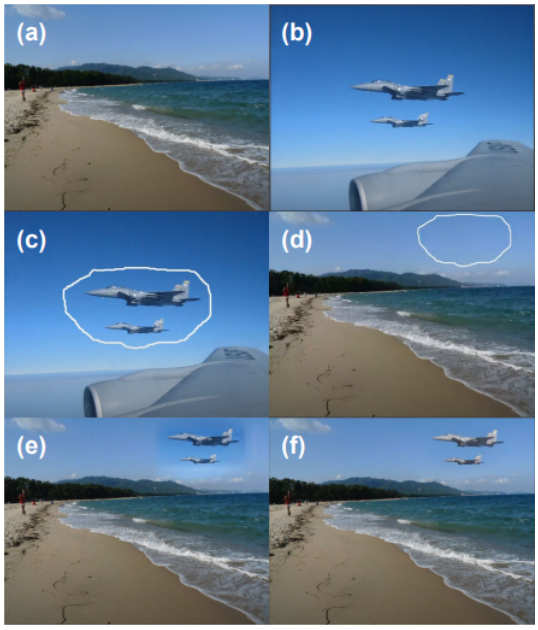

Figure 4. Example of Poisson blending [11].

model to create new instances of data that could have plausibly come from the input dataset [3]. By "adversarial process" we mean that a neural network G attempts to generate new data instances and another neural network D functions as a discriminator and estimates the probability that the example generated by G came from G or originated from the original dataset. G uses the feedback from D to generate better examples and D is presented with examples from G until D is no longer able to determine whether the example came from G or came from the input dataset [7].

6.2 Linear 3D Morphable Models

Linear 3D morphable models (3DMMs) are another type of generative model that aim to generate a 3D representation of any given face. Specifically, linear 3DMMs are a vector space of 3D shapes and textures of a class of objects. Each linear 3DMM contains a collection of 3D face shapes represented by a shape vector S and a corresponding texture vector T. New faces are generated by forming linear combinations of S and T [6]. Figure 5 shows of the shape and texture vectors of a face generated by a linear 3DMM.

$$\mathbf{S} = \sum_i \alpha_i \mathbf{S}_i = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \ldots$$

$$\mathbf{T} = \sum_i \beta_i \mathbf{T}_i = \beta_1 + \beta_2 + \beta_3 + \beta_4 + \ldots$$

Figure 5. Shape vector S and texture vector T of a face generated by a linear 3DMM [6].

6.3 Encoder-Decoders

Encoder-decoders are another type of generative model that aim to reconstruct the data given to it as input. To reconstruct the dataset, the model starts with the encoder. Given input data, the encoder attempts to compress the data into the lowest dimensional representation it can. This forces the model to discover important patterns in the data and learn how to represent it using only the most essential portions. The decoder then attempts to reconstruct the data from the lower-dimensional representation. The model is trained through what is known as the reconstruction error, which is the difference between the reconstructed data and the original data. The goal of the model is to minimize the reconstruction error so that the reconstructed data is as close as possible to the original dataset [14].

7 Face Swapping Approaches

In this section, we cover three recently developed approaches to face swapping. The approaches covered in this section include one based on GANs, one based on linear 3DMMs, and one based on an encoder-decoder model architecture.

7.1 GAN Based Approach

Nirkin et al. [16] propose a face swapping approach utilizing GANs, which they named *FSGAN*. The face swapping pipeline for *FSGAN* can be found in Figure 6. To begin the process of transferring the source face F_s in the source image I_s onto the target face F_t in the target image I_t, *FSGAN* starts with generator G_r. Given the facial landmarks of F_t, G_r generates a new version of the source image such that it depicts F_s in the same pose and expression as F_t. G_r then produces the segmentation mask of this reconstructed source face F_r. Generator G_s then produces the segmentation mask S_t of F_t.

Due to occlusions that may block the view of F_s (e.g., hand in front face, glasses, hair, etc.), F_r may contain missing parts relative to F_t. To remedy this problem, *FSGAN* utilizes a facial inpainting generator G_c to fill in the missing facial parts of F_r such that it matches F_t. G_c rerenders F_r based on the segmentation mask S_t to estimate these missing facial parts. Since S_t is a representation of the visible facial parts of F_t, rendering F_r based on S_t ensures that F_r and F_t match in terms of visible facial parts. After G_c fills in the missing parts of F_r, *FSGAN* now has a rendered face F_r that has the same pose, expression, and visible face portions of F_t. The final step is to then blend the reconstructed source face into the target image with a blending generator G_b that utilizes Poisson blending [16].

7.2 Linear 3DMM Based Approach

On Face Segmentation, Face Swapping, and Face Perception [10] proposes a face swapping approach based on a linear 3DMM. Given a source image I_S and a target image I_T, the model first generates 3D shape representations of the source

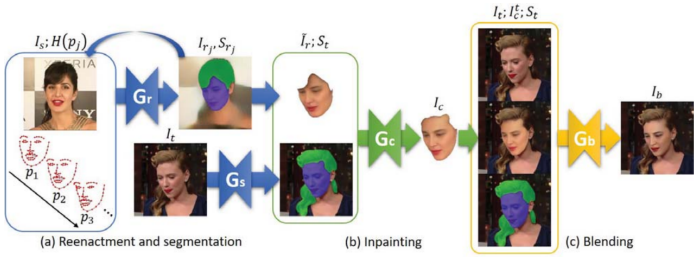

Figure 6. Face swapping pipeline of *FSGAN* [16].

and target faces. The 3DMM used in this paper is only capable of generating 3D face shapes with neutral expressions and pose, so each 3D shape is then modified using facial landmarks to match the corresponding face's pose and expression. This is represented in step b of Figure 7. A neural network is then used to produce the segmentation masks of the faces in the source and target images. Once the segmentation masks have been produced, the face swap is ready to be performed.

To swap the target face with the source face, the 3D shape of the source face V_S is projected onto I_S. Bilinear interpolation is then used to assign 3D vertices to the segmentation mask of the source face and sample the intensities of the source image based on those vertices. Bilinear interpolation is also used to assign 3D vertices to the segmentation mask of the target face (step c of Figure 7). Since all 3D faces generated by 3DMMs correspond in the indices of their vertices, the sampled intensities from the vertices of V_S can be directly transferred to the vertices of V_T. Transferring the sampled intensities to V_T provides texture to the vertices corresponding to the segmentation mask of the source face, which is represented visually by step d of Figure 7. V_T is then rendered onto I_T using the segmentation mask of the target face to mask the rendered intensities. Finally, the rendered source face is blended into the target image using Poisson blending (step c of Figure 7) [10].

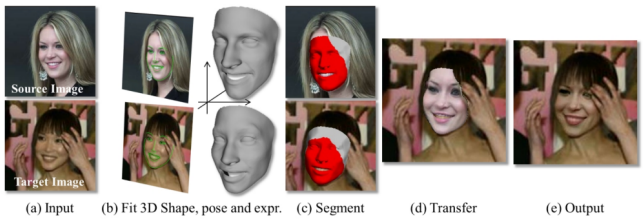

Figure 7. Face swapping pipeline of the linear 3DMM based approach [10].

7.3 Encoder-Decoder Based Approach

Researchers at Disney Research Studios recently published a paper describing their method of performing face swaps based on an encoder-decoder network architecture [8]. The researchers refer to their model structure as a *comb network* due to their model having a single encoder and p amount of decoders (one for every source face that the model has been trained for). Having multiple decoders enables the researchers to perform face swaps between any pair of faces that the model has been trained on and produces higher quality face swaps. To begin the face swapping process, the model starts by localizing the facial landmarks of the face in the target image x_t (step 1 in Figure 8). The model then normalizes the face to a 1024x1024 resolution and saves the normalization parameters (step 2 in Figure 8). The normalized face is then fed into the comb model and the p-th decoder is used to reconstruct the desired source face $\tilde{x}_s$ such that it has the same pose and expression as the target face (step 3 of Figure 8). Reverse image normalization is then performed on $\tilde{x}_s$ and $\tilde{x}_s$ is then blended into the target image, completing the face swap [8].

While Poisson blending may be able to produce passable results when the lighting of the target and source images are similar, artifacts may begin to appear in the composited image if the lighting between the source and target images is drastically different. As such, the researchers instead use a modified version of multi-band blending that is contrast-preserving to blend the source face into the target image.

Multi-band blending on its own does not guarantee that the source object that is being pasted into the target image will match the target image's lighting. To guarantee that the source face matches the lighting of the rest of the target image, the researchers copy the two highest levels of the Laplacian pyramid of the composited image and only the remaining levels are smoothed and blended. However, if the lighting between the source and target images varies greatly, then copying the two highest levels of the Laplacian pyramid is not enough to overcome the difference in contrast. To accommodate instances where the difference in lighting between the source and target images is significant, the researchers utilize what is called the Global Contrast Factor (GCF), which is essentially a measure of the overall contrast of an image. To get the reconstructed source face's contrast to match the contrast of the target image, the researchers calculate the ratio of the GCF of the target image and the GCF of the composited image and multiply each pixel of the composited image by this ratio [8].

8 Comparing Approaches

In this section, we examine the results of each face swapping approach and highlight some differences and similarities between the approaches. Before diving into specific comparisons, it is worth noting that both *FSGAN* and the encoder-decoder based approach can be used for video face swapping and image-to-image face swapping while the 3DMM based approach can only be used for image-to-image face swapping [16][8].

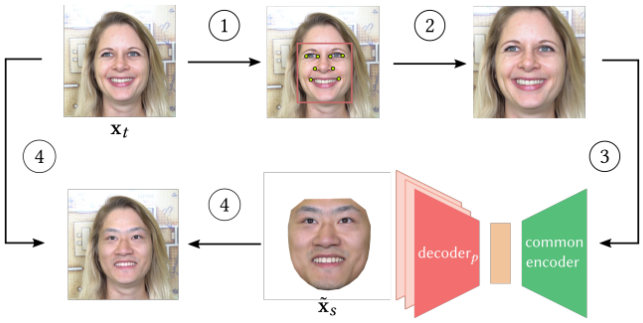

Figure 8. Disney Research Studios' pipeline for performing face swaps [8].

8.1 Encoder-Decoder Approach vs Linear 3DMM Approach

Figure 9 demonstrates face swaps for two pairs of people performed using the encoder-decoder and linear 3DMM based approaches. From Figure 9, we can see that the encoder-decoder based approach appears to produce higher quality face swaps than the 3DMM based approach. While the 3DMM based approach is capable of producing face swaps at a relatively high resolution, it struggles to accurately match the pose and expression of the target face. This is especially noticeable when looking at the eyes and mouth of the composited image. The quality of the blending is also not as good in the 3DMM face swaps, with this being especially noticeable in the first face swap pair where the lighting between the source and target image differs significantly. This highlights the superiority of the modified multi-band blending method used by Disney Research Studios over the Poisson blending method used by the 3DMM based approach.

Figure 9. Face swap pairs using the encoder-decoder based approach (ED) and the 3DMM based approach [8].

8.2 Generative Adversarial Network Approach vs Linear 3D Morphable Model Approach

Figure 10 demonstrates two face swaps performed using *FSGAN* and the 3DMM based approach. From Figure 10, we can see that since both *FSGAN* and the 3DMM based approach used Poisson blending, they are very similar in terms of blending quality. However, once again the 3DMM based approach struggles to accurately match the pose and expression of the target face. When examining the 3DMM face swaps closely, one can see artifacts around the eyes and mouth of the composited image.

Figure 10. Face swap pairs done using *FSGAN* and the 3DMM based implementation [16].

8.3 Encoder-Decoder Approach vs Generative Adversarial Network Approach

We were unable to find a direct comparison between the encoder-decoder approach and *FSGAN*. However, we can gain some insight into how the two approaches compare by examining the encoder-decoder face swaps in Figure 9 and the *FSGAN* face swaps in Figure 10. From the two figures, we can see that both *FSGAN* and the encoder-decoder based approach are capable of producing high-fidelity face swaps. However, it does appear that the modified multi-band blending method results in higher-quality blending compared to the Poisson blending method used by *FSGAN*.

9 Conclusion

As face swapping techniques become more powerful and accessible, the potential for misuse becomes more likely and dangerous. In this paper, we provide the reader with information regarding methods used within face swapping and three specific face swapping approaches with the goal of raising awareness of the topic and preventing readers from being fooled by potential misuse of the technology.

From the examination of the three approaches discussed in this paper, we conclude that the encoder-decoder approach to face swapping proposed by the researchers at Disney Research Studios produces the highest quality face swaps compared to the GAN and 3DMM based approaches. The encoder-decoder approach's superiority over the other two approaches can largely be attributed to its use of a modified multi-band blending method rather than Poisson blending.

Acknowledgments

I would like to thank my advisor Peter Dolan for his help during the research and development process of this paper. I would also like to thank Kristin Lamberty and alumni student Ariel Cordes for providing feedback on this paper.

References

[1] [n. d.]. Image Arithmetic - Pixel Subtraction. https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixsub.htm [Online; accessed 24-November-2022].

[2] Safinah Ali, Daniella DiPaola, Irene Lee, Jenna Hong, and Cynthia Breazeal. 2021. Exploring Generative Models with Middle School Students. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 678, 13 pages. https://doi.org/10.1145/3411764.3445226

[3] Jason Brownlee. 2019. A Gentle Introduction to Generative Adversarial Networks (GANs). https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/ [Online; accessed 1-October-2022].

[4] Papers With Code. [n. d.]. Laplacian Pyramids. https://paperswithcode.com/method/laplacian-pyramid [Online; accessed 18-October-2022].

[5] IBM Cloud Education. 2020. Neural Networks. https://www.ibm.com/cloud/learn/neural-networks [Online; accessed 18-October-2022].

[6] Bernhard Egger. 2021. SIGGRAPH2021 - 3D Morphable Face Models - Past, Present and Future - Presentation. https://www.youtube.com/watch?v=UGtIwWv1dds [Online; accessed 1-October-2022].

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[8] C. Schroers J. Narunic, L. Helminger and R.M. Weber. 2020. High-Resolution Neural Face Swapping for Visual effects. *Computer Graphics Forum* 39, 4 (2020), 173–184.

[9] Khalil Khan, Rehan Ullah Khan, Kashif Ahmad, Farman Ali, and Kyung-Sup Kwak. 2020. Face Segmentation: A Journey From Classical to Deep Learning Paradigm, Approaches, Trends, and Directions. *IEEE Access* 8 (2020), 58683–58699. https://doi.org/10.1109/ACCESS.2020.2982970

[10] Yuval Nirkin, Iacopo Masi, Anh Tran Tuan, Tal Hassner, and Gerard Medioni. 2018. On Face Segmentation, Face Swapping, and Face Perception. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. 98–105. https://doi.org/10.1109/FG.2018.00024

[11] Rich Radke. 2015. DIP Lecture 22: Image Blending. https://www.youtube.com/watch?v=UcTJDamstdk [Online; accessed 2-October-2022].

[12] Richard J. Radke. 2012. *Computer Vision for Visual Effects*. Cambridge University Press. https://doi.org/10.1017/CBO9781139019682

[13] Himanshi Singh. 2021. How Images are stored in the computer? https://www.analyticsvidhya.com/blog/2021/03/grayscale-and-rgb-format-for-storing-images/#:~:text=Images%20are%20stored%20in%20the%20form%20of%20a%20matrix%20of,black%20and%20255%20represents%20white. [Online; accessed 31-October-2022].

[14] WelcomeAIOverlords. 2019. Simple Explanation of AutoEncoders. https://www.youtube.com/watch?v=3jmcHZq3A5s [Online; accessed 2-October-2022].

[15] Wikipedia. 2022. Image Gradient. https://en.wikipedia.org/wiki/Image_gradient [Online; accessed 2-October-2022].

[16] Tal Hassner Yuval Nirkin, Yosi Keller. 2019. FSGAN: Subject Agnostic Face Swapping and Reenactment. In *2019 IEEE/CVF International Conference on Computer Vision*. IEEE, 7183–7192.

[17] Lingzhi Zhang, Tarmily Wen, and Jianbo Shi. 2020. Deep Image Blending. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 231–240. https://doi.org/10.1109/WACV45572.2020.9093632