International Journal of Internet, Broadcasting and Communication Vol.15 No.2 31-38 (2023)
http://dx.doi.org/10.7236/IJIBC.2023.15.2.31

IJIBC 23-2-4

Real-Time Arbitrary Face Swapping System For Video Influencers Utilizing Arbitrary Generated Face Image Selection

Jihyeon Lee†, Seunghoo Lee†, Hongju Nam† and Suk-Ho Lee††

†*Bachelor Degree Candidate, Dept. Artificial Intelligence Appliance, Dongseo University, Korea*
††*Professor, Dept. Computer Engineering, Dongseo University, Korea*
E-mail: petrasuk@gmail.com

Abstract

This paper introduces a real-time face swapping system that enables video influencers to swap their faces with arbitrary generated face images of their choice. The system is implemented as a Django-based server that uses a REST request to communicate with the generative model, specifically the pretrained stable diffusion model. Once generated, the generated image is displayed on the front page so that the influencer can decide whether to use the generated face or not, by clicking on the accept button on the front page. If they choose to use it, both their face and the generated face are sent to the landmark extraction module to extract the landmarks, which are then used to swap the faces. To minimize the fluctuation of landmarks over time that can cause instability or jitter in the output, a temporal filtering step is added. Furthermore, to increase the processing speed the system works on a reduced set of the extracted landmarks.

Keywords: *Face Swapping, Video Influencer, Landmarks, Stable Diffusion, Mediapipe*

1. Introduction

In recent years, there has been a growing trend of video influencers who choose not to reveal their identities on their channels[1][2]. These content creators are commonly known as "faceless influencers" or "anonymous influencers". They employ various methods such as wearing masks, applying digital masks or using voice-changing software to conceal their appearance. There are several reasons why YouTubers may opt for anonymity. A first reason is that it serves as a means of safeguarding their privacy and personal lives, especially if they tackle sensitive or controversial topics. Another reason might be the one that the anonymity is a creative choice, adding an element of intrigue and mystery to their content.

Despite not showing their faces, these video influencers have managed to build a dedicated following and forge connections with their audiences through their content and personalities. Some of the most successful faceless video influencers have developed personas or characters to represent themselves on their channels,

Manuscript Received: February. 12, 2023 / Revised: February. 14, 2023 / Accepted: February. 17, 2023
Corresponding Author: petrasuk@gmail.com
Tel: +82-51-320-1744, Fax: +82-51-327-8955
Professor, Department of Computer Engineering, General graduate school, Dongseo University, Korea

setting themselves apart from others and establishing a unique brand.

The advent of deep learning has led to the creation of numerous technologies that allow for digital face masking. One such technique is deepfake, which has the ability to superimpose one person's face onto another's[3][4]. However, most deepfake technologies do not operate in real-time and require a certain processing time to output a video where the face has been replaced, usually when a video is inputted offline. The computer requires sufficient resources to perform the necessary processing to hide your face, which may lead to delays.

Another digital masking method is face swapping which is a computer vision technique that involves replacing the face of one person in an image or video with the face of another person. This technique involves manually selecting facial features such as eyes, nose, and mouth, and replacing them with corresponding features from another face. There are several platforms which offer face swapping services. “Reface” is an app which allows users to swap their faces with celebrities, movie characters[5]. “Snap Camera” is a desktop application that allows users to add augmented reality lenses and also offers a face swapping feature that allows users to switch their faces with other people or even objects[6]. “Face Swap Live” allows users to swap faces with their friends or even celebrities in real-time[7]. “MSQRD” is an app that offers a face swapping feature that allows users to switch their faces with other people or even animals[8]. “DeepArt.io” is a website uses deep learning algorithms to create art by merging two images together[9]. Users can upload a photo of themselves and an image of their choosing to create a unique face-swapped artwork.

However, in all of the above platforms, users are either limited to pre-defined faces offered by the system or are required to prepare the faces they want to swap with their own faces. In this paper, we propose a platform that enables users to generate faces using a generative model, and then select the desired face image to replace their own. The selected face is then applied to the user's face based on landmark detections. To accelerate the face swapping process, we perform landmark selection on the original landmarks extracted by a landmark extraction network. Additionally, we apply temporal filtering on the selected landmarks to reduce the shaking of the swapped face.

2. Preliminaries

The following techniques were employed to implement the proposed system.

2.1 Stable Diffusion

Stable Diffusion is a deep learning model in the field of artificial intelligence, developed in collaboration with Stability AI and Runway ML, and based on the research proposed by the Machine Vision & Learning Group (CompVis) lab at the University of Munich[10]. The stable diffusion technique involves the use of three artificial neural networks: CLIP, UNet, and VAE (Variational Auto Encoder). When a user inputs text, the text encoder (CLIP) converts the text into a tokenized format that can be understood by UNet. The UNet then generates a series of noise patterns based on the tokens, and these patterns are progressively denoised to produce a high-quality image. The role of the VAE is to convert the denoised image into pixel values. Compared to previous diffusion probability image generation models, stable diffusion is more efficient as it employs autoencoders at the front and back to insert and remove noise in a smaller latent space, rather than the entire image. This results in a significant reduction in computational resources required, particularly when generating images at high resolutions.

2.2 Mediapipe

MediaPipe is an AI framework developed by Google that provides a set of tools for performing vision-

related AI tasks using video data[11]. It offers a variety of functions and models, such as face recognition, pose estimation, object detection, and motion tracking, that target the human body.

MediaPipe is provided as a library, making it convenient to use with various programming languages and environments, in addition to Python. The framework includes pre-trained AI models, which eliminates the need for developers to build and train models from scratch. Instead, developers can simply call the library's functions to implement vision AI functions.

2.3 Django Framework

Django is an open-source web framework written in Python that follows the model-template-view (MTV) pattern[12]. It is currently maintained by the Django Software Foundation. Django's primary goal is to simplify the creation of database-driven websites. It emphasizes the reuse of components, the ability to add plug-ins, and rapid development. By providing a set of pre-built components and a clear architecture, Django enables developers to focus on building specific features and functionality, rather than worrying about low-level details.

3. Real-time Face Swapping with Arbitrary Generated Face Image

3.1 Text to Image Generation through a Web-based Interface

In order to generate an arbitrary face image based on the user's text query input, we implemented a Django server which connects to a pre-existing image generation system, i.e., a Stable Diffusion Model API, through a REST API. The Django server will act as a client and send HTTP requests to the Stable Diffusion Model API. The front-end interface allows the user to input a text query. Then the back-end logic handles this text query input and sends a request to the image generation system through an API. After the image generation process is completed, the image is stored at a specific address on the internet. The server then sends this address back to the client-side code, which uses this address to request the image. Once the image is retrieved by the client-side code, it is displayed to the user in the HTML template in the Django server. Then the user can decide whether he wants to use this image or not by clicking on the accept or decline button. When the user accepts the generated image by clicking on the accept button, the generated image is sent to the Landmark detection module for extracting and processing the landmarks. Figure 1 shows the operation flow of the Django Web Server which interacts via the REST API with the pretrained generation model.

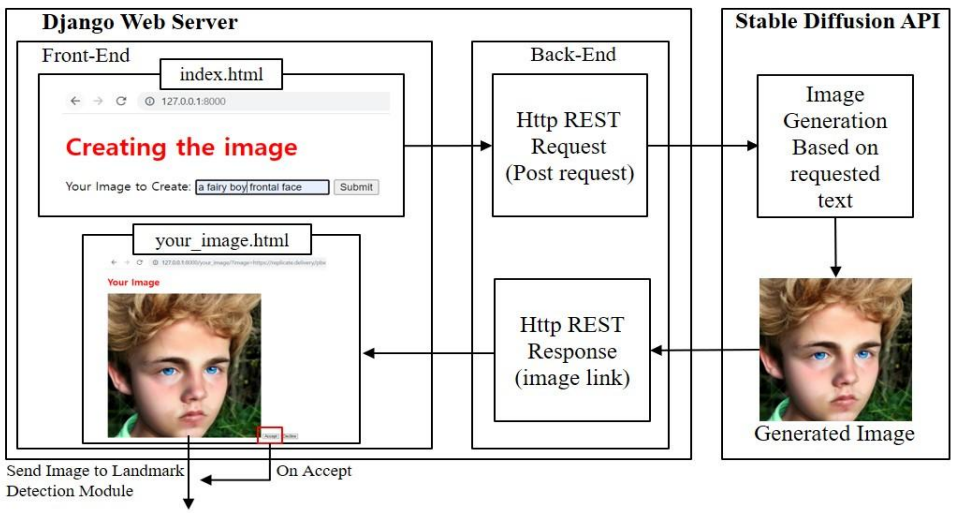

Figure 1. Django Web Server which sends a REST Request to the pre-trained image generation API and gets the generated image as a REST Response

3.2 Landmark Processing

After the generated face image is sent to the image processing module, the image is put as the input to a pre-trained landmark detecting deep leaning model, i.e., the Mediapipe model, which detects the landmarks. The computation cost of the following processes become prohibitively high when using all landmarks. To address this issue, we apply a process R to reduce the number of landmarks in the original set L_{n+1}^{total}, resulting in a new set L_{n+1} with a smaller size:

$$L_{n+1} = R(L_{n+1}^{total}) \tag{1}$$

However, as each frame is processed independently, the landmarks in the generated images may fluctuate significantly over time, causing instability or jitter in the output. Therefore, we have to apply also a temporal filtering on the landmarks to reduce the fluctuation. To reduce the fluctuation of the landmarks over time, we employ a method of smoothing known as the cumulative moving average, which averages the landmark positions across multiple frames in the temporal axis[13].

Let denote by L_{n+1} the position vector of the landmarks in the current frame and denote by CA_n the cumulative average of the position vectors up to the previous frame. Then, the equation for the cumulative moving average of the current frame becomes:

$$\mathrm{CA}_{n+1} = \frac{L_{n+1} + n \cdot \mathrm{CA}_n}{n+1} \tag{2}$$

Figure 2 shows the process flow of the landmark processing steps.

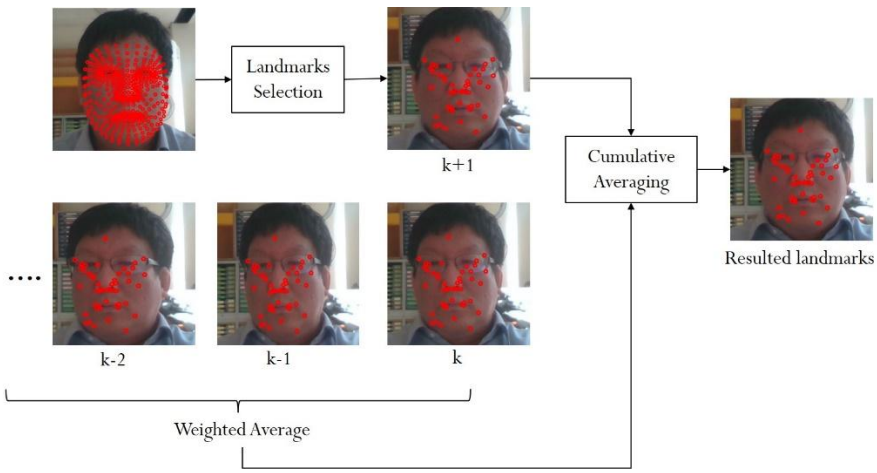

Figure 2. Landmark processing which includes landmarks extraction, landmarks selection and temporal filtering steps

3.2 Face Swapping

After we extracting and processing the landmarks, we use them to swap the face in the incoming frame image with that of the generated image. The process proposed in a lecture in [14] outlines the following steps:

1. Apply Delaunay triangulation to the landmarks of both the generated face and the face in the incoming frame.

2. Identify corresponding local triangles in both images and compute an affine transform matrix between them.
3. Copy all the pixels in the generated face image into the local triangles in the face region of the incoming frame, using the computed affine transform matrix.
4. Apply seamless blending to the swapped face to reduce the color difference between the swapped region and the non-swapped region of the face.

One common issue is the inability to extract all the landmarks of the face in the incoming frame due to factors such as face motion or changes in illumination. In such cases, the YouTuber's face may become exposed, which is a critical problem since the YouTuber who desires to remain anonymous can be identified by the viewers. Therefore, we add a test process in the above algorithm which confirms that the landmarks are all extracted. If not all the landmarks can be extracted, the corresponding frame is treated as an exception and becomes discarded. Instead, the previous frame is used to replace the current frame which helps the preservation of the anonymity of the YouTuber. Figure 3 illustrates the complete process of the arbitrary face swapping system that employs selection of arbitrary generated face images.

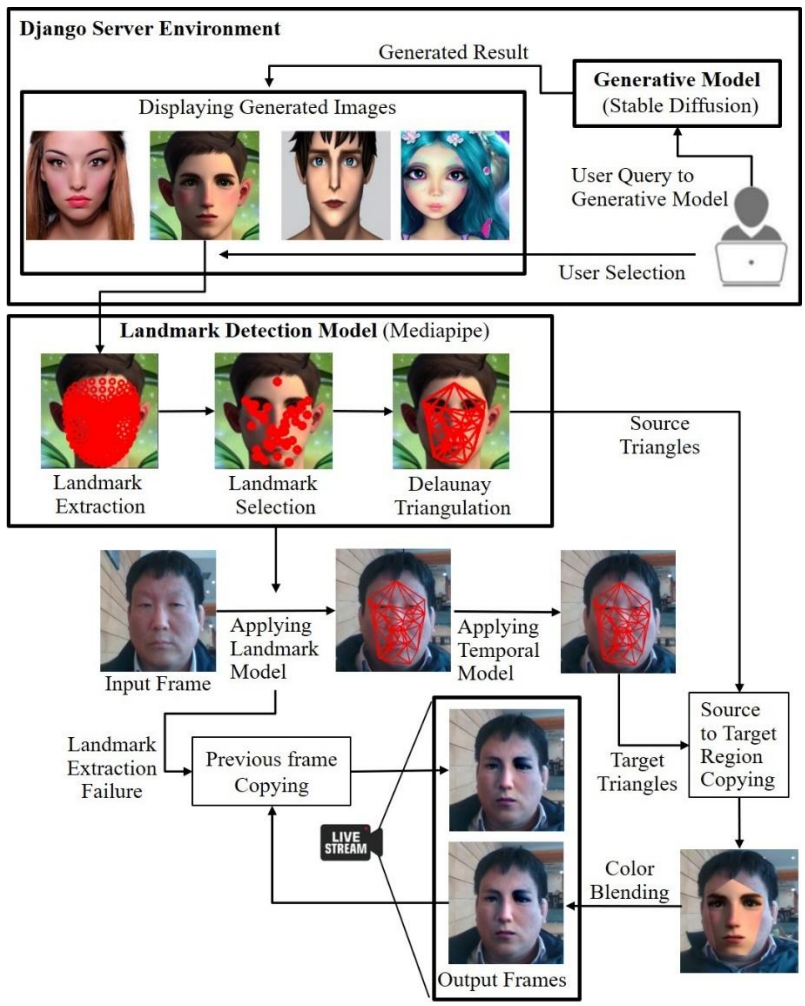

Figure 3. Overall System Diagram of the proposed arbitrary generated face swapping system

4. Experimental Results

The implemented software's operation is demonstrated in Figure 4. The process begins by inputting a text prompt into the front-end page of our server's input interface. The server then transmits this prompt to a pretrained generative model through an API, which returns the results displayed on a separate front-end page of our server. When the user selects the generated image, the face swapping algorithm initiates, resulting in a real-time face swap. Figure 4 illustrates swapped face images, with the upper row displaying the generated images and the lower row showing the corresponding swapped results.

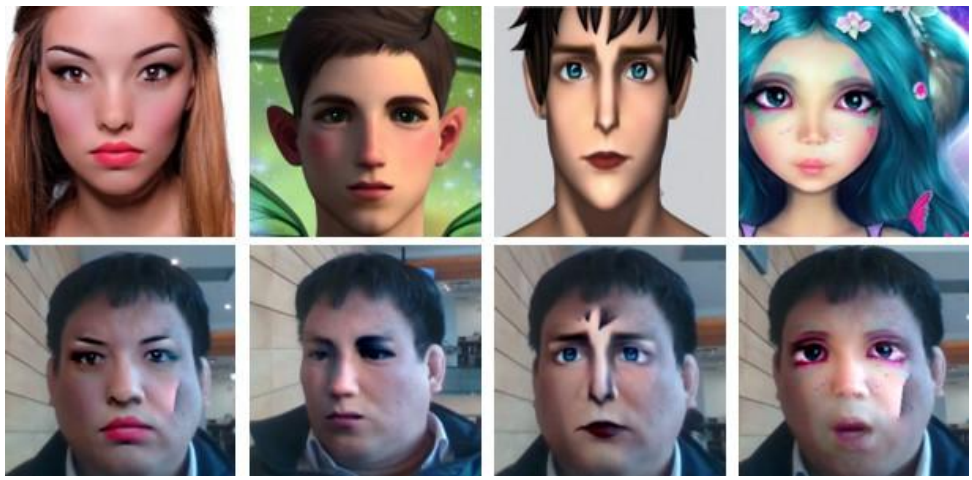

Figure 4. Face swapping results with different arbitrary generated face images. Upper row: arbitrary generated face image, Bottom row: face swapped result

Figure 5 shows the effect of using a cumulative averaging on the landmarks. We used a distance measure which measures the L2 difference of the landmark position vectors in the previous and the current frames when the face is motionless. It can be seen from Fig. 5 that the L2 difference is significant when not using the cumulative averaging than when using it. This verifies that the landmarks are less unsteady when using the cumulative averaging.

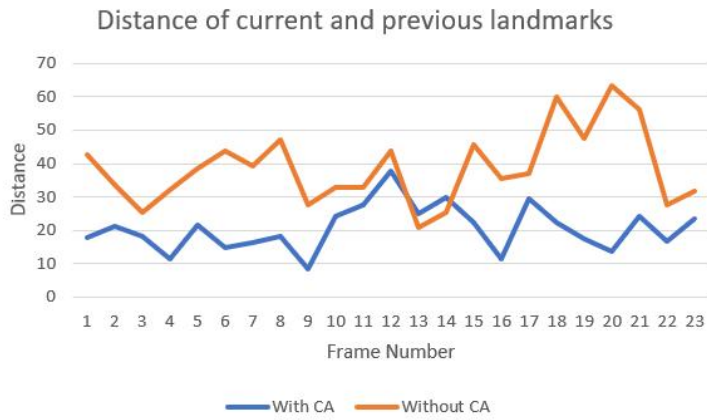

Figure 5. Comparison between the L2 differences of the current and previous landmarks with and without using the cumulative averaging

Table 1 presents a comparison of computation time between the original and reduced landmark sets. The table demonstrates that using the reduced landmark set reduces computation time by more than half.

Table 1. Comparison of computation time with and without landmark reduction

Method	Computation Time (Seconds per Frame)
With Landmark Reduction	0.1489
Without Landmark Reduction	0.3605

5. Conclusion

This paper presents a system that swaps the face of a YouTuber with an arbitrary generated face image from a pretrained generative model. We implemented a Django-based server that allows users to input a text to generate a face and can select the preferred face, which is sent to the landmark extraction module for processing upon acceptance. The resulting landmarks are used to replace the user's face with the generated face. To reduce computation time, we implemented a landmark reduction step and used a cumulative average method to minimize the fluctuation of landmarks over time. Additionally, we proposed a solution for the exceptional case when landmark detection fails, which involves detecting the failure and using the previous successfully detected frame to prevent the YouTuber's face from becoming exposed.

Overall, our system offers a practical solution to the challenge of generating realistic face swaps for online content creators, and has potential applications in fields such as entertainment, advertising, and education. Our work contributes to the growing body of research in the area of real-time computer vision techniques for real-time anonymization, and we hope that our approach will inspire further research and development in this field.

Acknowledgement

This work was supported by Dongseo University, "Dongseo Cluster Project" Research Fund of 2022 (DSU-20220001).

References

[1] I. Dunham, (2020). “Faceless Youtubers: How Content Creators Shape Audience Expectation,” in *Proc. The 21st Annual Conference of the Association of Internet Researchers*, Oct. 27-31, 2020. DOI:https://doi.org/10.5210/spir.v2020i0.11205

[2] C. Clark-Gordon, N. Bowman, A. Goodboy, A. Wright, (2019) “Anonymity and Online Self-Disclosure: A Meta-Analysis,” *Communication Reports*, Vol. 32, Issue 2, pp.98-111, DOI: https://doi.org/10.1080/08934215.2019.1607516

[3] G. David, J. Edward , “Deepfake Video Detection Using Recurrent Neural Networks,” *Proc. of 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Nov. 27-30, 2018. DOI: https://doi.org/10.1109/AVSS.2018.8639163

[4] L. Yuezun, L. Siwei, “Exposing DeepFake Videos By Detecting Face Warping Artifacts,” *Proc. of Conference on Computer Vision and Pattern Recognition(CVPR 2019)*, pp. 46-52, June 16-20, 2019. DOI: https://doi.org/10.48550/arXiv.1811.00656

[5] Reface. *https://reface.app*.

[6] Snap Camera. *https://snapcamera.snapchat.com*

[7] Face Swap Live. *https://faceswaplive.com*

[8] MSQRD. *https://msqrd.kr.uptodown.com*

[9] DeepArt.io. *https://creativitywith.ai/deepartio/*

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR*

2022), pp. 10684-10695, June 18-24, DOI: https://doi.org/10.1109/CVPR52688.2022.01042
[11] Mediapipe. *https://google.github.io/mediapipe*
[12] Django rest framework. *https://www.django-rest-framework.org*
[13] W. Enders*, Stationary Time-Series Models, Applied Econometric Time Series (Second ed.)*. Wiley. pp. 48–107, 2004
[14] Pysource. *https://pysource.com*