applied sciences

MDPI

Review

Quick Overview of Face Swap Deep Fakes

Tomasz Walczyna and Zbigniew Piotrowski *

Faculty of Electronics, Military University of Technology, 00-908 Warszawa, Poland; tomasz.walczyna@wat.edu.pl
* Correspondence: zbigniew.piotrowski@wat.edu.pl

Abstract: Deep Fake technology has developed rapidly in its generation and detection in recent years. Researchers in both fields are outpacing each other in their axes achievements. The works use, among other methods, autoencoders, generative adversarial networks, or other algorithms to create fake content that is resistant to detection by algorithms or the human eye. Among the ever-increasing number of emerging works, a few can be singled out that, in their solutions and robustness of detection, contribute significantly to the field. Despite the advancement of emerging generative algorithms, the fields are still left for further research. This paper will briefly introduce the fundamentals of some the latest Face Swap Deep Fake algorithms.

Keywords: face deep fake; faceswap; image deep fake

Citation: Walczyna, T.; Piotrowski, Z. Quick Overview of Face Swap Deep Fakes. *Appl. Sci.* **2023**, *13*, 6711. https://doi.org/10.3390/app13116711

Academic Editors: Junhui Huang, Qi Xue and Zhao Wang

Received: 9 May 2023
Revised: 24 May 2023
Accepted: 29 May 2023
Published: 31 May 2023

1. Introduction

Facial swapping generates photos with a different individual's facial identity and qualities (e.g., pose, expression, lighting, and background). The implementation of this task is not limited to long-term training of the model. Additionally, more time-consuming preprocessing and postprocessing tasks are usually necessary to create a perceptually good conversion. Most of the described algorithms use a variety of labor-intensive external solutions to improve their performance. Algorithm developers generally use two main approaches—source-based and target-based. The target-based approach entails extracting the person's "identity" from the source image and inserting it in the target image with the characteristics contained there. The source-based approach, in contrast, involves editing the source image based on the attributes extracted from the target image. The downside of the second solution is that there is no control over the environment used. Therefore, the work will focus on target-based algorithms to make the usefulness as universal as possible.

Developments in algorithmic work are introducing more solutions to improve quality and reduce the time or amount of data required to achieve the target result [1]. Often, the work that obtains the best results is trained for specific examples that are not generalized for those outside the training set. This paper will present identity conversion pipelines. However, it should be considered that the results they achieve could change depending on changes in pre- and postprocessing tools. Therefore, most of the work will focus on how the pipeline works and the model of the generating algorithm.

Figure 1 shows a typical source-based face swapping pipeline. It consists of data preprocessing, a deep fake model (containing minor models, such as for identity extraction), and postprocessing to improve the results. Due to the frequent merging of different stages with each other depending on the algorithm and the computing power held by the developers, a new division will be introduced:

- Identity extraction—involves obtaining information about the human's identity.
- Attributes extraction—involves extraction from the image containing the person whose identity will be edited to create features and attributes such as pose, emotion, and background.
- Generator—generates the target image containing the source person's identity.

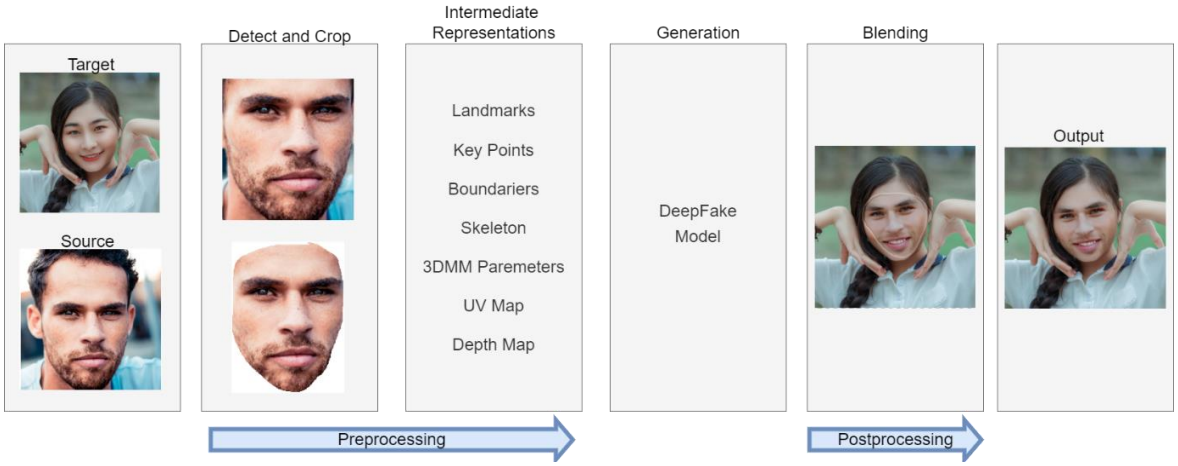

Figure 1. Typical face swapping pipeline.

The presented division is general. Although some developers combine or divide the presented functionalities differently, our formulation is universally applicable and straightforward, so we will stick to it throughout this study.

2. Face Swapping Process

Although many advanced algorithms implementing face swapping have been developed, mentioning the origins of Deep Fake is necessary to gain a better perspective on the problem. One of the first approaches to face swapping based on deep learning methods is Faceswap [2]. This software first appeared in 2017 and is still being developed. Updates are still being added, with new models that users can choose and test. The basic approach presented by the developers is based on autoencoders. Face Swap is the software with which the user, without much programmer knowledge, can perform face swapping. Despite the basic settings offered, the user can make many changes to the model, among other things.

Another algorithm and publicly available software that also has its origins based on autoencoders is SOTA (state-of-the-art) DeepFaceLab [3]. The developers continue to develop the software by adding other scientific developments, such as GANs (generative adversarial networks) [4,5]. In addition to the model in the published paper, the authors mention other enhancements, including preprocessing and postprocessing. The solution is regarded as one of the best in terms of the quality of the generated content. High quality is provided because the entire pipeline is adjusted to work on only two people. There is no possibility of using people who did not appear during training. The model must be trained from scratch whenever we want to perform a face swap on a new person.

In contrast, subsequent innovations have attempted to disentangle the identification of the person portrayed in a photograph from any traits that may be present, such as hair, spectacles, or occlusions. Only the face is affected by this transformation. In addition, an effort was made to generalize them as much as possible and limit the number of samples of a person used in the generating process. One such solution is FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping [6]. The generation algorithm itself consists of two networks: the AEI-Net (adaptive embedding integration network) responsible for face transfer, which uses an encoding method to identity information in addition to GANs, and the HEAR-Net (heuristic-error-acknowledging network) network, which is added to improve the results obtained, including occlusion.

Similar to the above method, using the encoding of identity information is involved in SimSwap: An Efficient Framework For High Fidelity Face Swapping [7]. Compared to the above, it uses one model rather than two to perform the face swapping task.

A method that achieves outstanding results regarding the quality of the transformations obtained is HifiFace: 3D Shape and Semantic Prior Guided High Fidelity Face

Swapping [8]. In this task, developers employ a 3D-shape-aware identity to regulate the face form under the geometric supervision of 3DMMs (3D morphable models) and the 3D face reconstruction approach. The method achieves good results in identity transfer with the added aspect of high-quality results.

Another method that achieves excellent results is GHOST-A [9]. Its implementation uses the FaceShifter mentioned above as a base. However, it incorporates discoveries such as the eye loss function, the super-resolution block, and the creation of a Gaussian-based face mask, thereby enhancing the quality of the underlying method.

The last analyzable model is a diffusion-based architecture—DiffFace [10]. Thanks to technological advances and increased availability of computing power, diffusion models have also found their way into face substitution. Despite the many shortcomings connected with their current speed of operation, these models may be a breakthrough in terms of the realism of the generated samples, primarily through training process stability [11].

2.1. Preprocessing

One of the main aspects in both target use and training is the appropriate data processing, for example, to find a human face in an image. Not all algorithm developers refer to this issue and use publicly available training sets. In the subsection, we describe the solutions used by the developers in this regard.

Faceswap software, which is a one-to-one method, is limited in its operation to working on two people—target and source developers do not describe the datasets used but present the algorithm of procedure required to use the program. First, acquiring a considerable quantity of data, such as photographs or films depicting the source target, is essential. This data is fed to three processes: detection, alignment, and mask generation, which is often similar to the algorithms discussed further.

Face detection is the process of locating a face inside a frame. The detector examines the image and identifies face-like regions.

Alignment involves finding "landmarks" (intermediate representations in Figure 1) within the face to thus orient the face. This process takes the detector result and determines where the face's key features (eyes, mouth, nose, etc.) are located.

Mask generation removes the background and obstacles from the image area and leaves the face.

The results obtained in preprocessing serve both in the training phase and in actual use.

Faceswap, as a software, allows the user to select the algorithms he wants to use in individual processes. In the detection phase, several detection models can be selected: the CV2 DNN Detector (based on the OpenCV2 library, it is a detector using the pre-trained single-shot detection model—ResNet [12]), MTCNNs [13] (multi-task cascaded convolutional networks), and the S3FD [14] (Single-Shot Scale-invariant Face Detector—the best detector among those proposed. It can detect more faces and fewer false positives than the others but is much more resource intensive). The S3FD face detector is an advanced algorithmic system that detects human faces within images or video content. It has been developed to address the challenges typically associated with anchor-based detection methods, particularly regarding their reduced efficiency when identifying smaller objects or faces. The system is built upon a scale-equitable framework, which employs a comprehensive range of anchor-associated layers and a series of carefully considered anchor scales. This method allows the system to handle faces of various scales effectively, from large to very small. One key characteristic of the S3FD detector is its implementation of a scale compensation anchor-matching strategy. This strategy is specifically designed to enhance the recall rate of small faces, thereby improving the system's ability to identify smaller faces within the content accurately. Furthermore, the S3FD detector employs a max-out background label. This feature aims to decrease the false positive rate associated with small face detection, which essentially means it helps to prevent the system from incorrectly identifying non-face objects or regions as faces.

In the alignment process, the user can choose two algorithms: the CV2 DNN Aligner [15] (low resource intensive, but also low accuracy) and the FAN [16] (slower, but more accurate). The authors of FAN have devised a face alignment algorithm that addresses both 2D and 3D datasets. They combined advanced architecture for landmark localization with a modern residual block, trained it on an expanded 2D facial landmark dataset, and evaluated it across multiple 2D datasets. To address 3D face alignment scarcity, a 2D-guided CNN was introduced to convert 2D annotations into 3D, thus creating the largest and most challenging 3D facial landmark dataset to date. The performance of these networks is remarkable, which suggests that they may be nearing peak accuracy on the used datasets.

As part of the mask development, the developers implemented BiSeNet [17] (a relatively lightweight mask based on neural networks, which provides more precise control over the masked area, including masking the entire head), VGG Clear [18] (a mask designed to provide intelligent segmentation of mostly frontally aligned faces without occlusion), VGG Obstructed [18] (a mask designed to provide intelligent segmentation of mostly frontally aligned faces. The masked model has been specially trained to recognize some facial obstructions), Unet-DFL—TernausNet [19] (implemented by the developers of the DeepFaceLab software described later and trained accordingly. The mask is designed to provide intelligent segmentation of mainly frontal parts of the face).

In addition to the treatments mentioned above, Faceswap allows a choice of normalizing methods to better find faces in different lighting conditions. To reduce "micro-jitters," a Re Feed function has been added, which repeats the detection–alignment process several times and then averages the results.

The generated data must then be sorted accordingly to catch erroneous samples. The developers have implemented several sorting algorithms; however, they suggest using one involving the VGG face descriptor [20]. It uses a pairwise clustering algorithm to check the distance between the 512 features on each face in the set and order them accordingly.

DeepFaceLab, which is, likewise, a one-to-one approach and, hence, has no publicly available dataset regarding the number of user-configurable default settings, is more modest than its predecessor. Nevertheless, it achieves better results and is more automated. Developers have focused on providing the best preprocessing algorithms to the users. Instead of customizable settings, they have introduced several modes that determine which parts of the face to manipulate and extract: full face (default mode, whole face), half face (better resolution, but includes a smaller cheek area), mid-face (30% wider than half face), whole face (whole face area plus forehead), and head (whole head).

As part of face detection, they suggest the S3FD detector also be used in Faceswap [14].

The second step to keep the performance stable over time is face alignment. For this purpose, the developers suggest two algorithms: 2DFAN (2D Face Alignment Network [16], also used in Faceswap and applied to faces with a standard pose) and PRNet [21] (used in exceptional cases where one side of the face is out of view).

In addition, the developers added an optional feature with a configurable time step to ensure the stability of the detected points. They use Umeyama's [22] traditional point pattern mapping and transformation approach to produce the similarity transformation matrix needed for face alignment. After applying the above steps, to remove potential errors, they use one of the many proposed sorting algorithms and eliminate samples that do not represent the face of the person for whom the model will be trained.

In a further step, TernausNet [19] was used for segmentation and masking. TernausNet can effectively remove irregular occlusions. However, in some shots, the model may not cope with generating precise masks. Therefore, the developers developed a face segmentation tool, XSeg, which can be customized with few-shot learning. The user can create a mask label and train a customized segmentation model, thus creating a self-defined mask.

FaceShifter utilizes CelebA-HQ [23], FFHQ [24], and VGGFace [20] to train the AEI network. In contrast, the HEAR network is trained using only the facial regions with the highest percentage of heuristic errors in these datasets, together with synthetic occlusions.

The occlusion pictures are sampled at random from the EgoHands [25], GTEA Hand2K [26], and ShapeNet [27] object rendering sets. Each facial image is first aligned using five extracted landmarks and cropped using the approach [28]. The cropped image contains the whole face and portions of the surroundings.

SimSwap was trained on the VGGFace2 dataset [29], and images smaller than 250 × 250 were removed to improve quality. They were then aligned and de-essentially cropped. The SimSwap developers generated their own VGGFace2HQ dataset [30] for quality enhancement, which was obtained by applying two tools on the dataset mentioned above: GFPGAN [31]—for image restoration—and InsightFace [32]—for data preprocessing.

HifiFace was trained on VGGFace2 [29] and Asian-Celeb [33]. The developers created two models for 256 × 256 resolution and 512 × 512 resolution. For the former, they used the same method as FaceShifter [28], and for the latter, they implemented a portrait enhancement network [34] to increase the resolution of the training set to 512 × 512.

The developers of GHOST-A used CelebA-HQ [23] and VGGFace2 [20] datasets in their work and used the InsightFace library [32] to crop and align faces properly.

DiffFace developers used the FFHQ [24] dataset to train the diffusion model, while they used the FaceForenics++ dataset [35] to test the algorithm's performance.

2.2. Identity Extraction

One of the pipeline components responsible for face replacement is the identity extractor. Its task is to represent a person's face in the appropriate hidden space to eliminate unnecessary attributes that are not universal for photos of the same person taken at different moments and situations.

FaceSwap and DeepFaceLab are one-to-one algorithms, so the identity extractor training should also occur during the training of the whole pipeline. FaceSwap is based on the idea of using autoencoders. To utilize autoencoders to swap faces, the authors train two auto-encoders: one for the person whose face is to be moved (the source) and one for the person whose face will be replaced (the target). The identity features are assigned to the model in such a solution because of the bottleneck used. Autoencoders ultimately try to reconstruct the identity suitable for the model based on only non-universal attributes.

The developers of DeepFaceLab use a standard encoder in their solution, and the transfer of identity features is done through a decoder that is individual to the person.

FaceShifter and the following algorithms are already many-to-many algorithms—they are intended to work for multiple instances of individuals, including not necessarily found in the training set. A component called "Identity Encoder" was used to extract identity features. It is responsible for encoding the identity embedding, which will contain the information necessary for the generator to reconstruct the identity. For the encoder, the developers of FaceShifter, SimSwap, and Ghost-A used a pretreated model for face recognition [36]. The model was trained on a vast quantity of 2D faces.

Within HiFiFace, the developers used a 3D-shape-aware identity extractor to extract information about the target's identity. In this task, unlike previous solutions, a pre-trained 3D face reconstruction model [37] is added in addition to a pre-trained network for face recognition (Curricularface [38]) to generate an identification vector carrying identity information. It is used on both the image-containing attributes and the identity. Based on the generated information, expression and pose information is extracted from the first model, while identity information is extracted from the second.

In DiffFace, the developers tested two models for extracting identity features introduced as a condition during face synthesis: ArcFace [36] and CosFace [39]. The introduction of these features into training is described in the generation subsection.

ArcFace, CosFace, and CurricularFace are state-of-the-art methods for face recognition that significantly enhance feature discriminability by adopting margins in the softmax loss function to maximize class separability.

ArcFace introduces an additive angular margin loss, which has a clear geometric interpretation and significantly enhances the discriminative power. It is susceptible to

massive label noise. Hence, a variant called sub-center ArcFace has been proposed. In sub-center ArcFace, each class contains multiple sub-centers, and training samples only need to be close to any of these. This technique boosts performance by automatically purifying raw web faces under real-world noise. The pre-trained ArcFace model can also generate identity-preserved face images using network gradient and batch normalization priors.

CosFace addresses the lack of discriminative power in the traditional softmax loss by proposing a large margin cosine loss (LMCL). It reformulates the softmax loss as a cosine loss using L2 normalizing features and weight vectors to remove radial variations. A cosine margin term is introduced to maximize the decision margin in the angular space. As a result, minimum intra-class variance (differences within the same identity) and maximum inter-class variance (differences between separate identities) are achieved.

CurricularFace introduces adaptive curriculum learning loss, which incorporates curriculum learning into the loss function to develop a novel training strategy for deep face recognition. It starts with easy samples in the early training stage and progresses to harder ones in the later stages. The method adapts the relative importance of easy and hard samples during different training stages, thereby assigning different importance values to different samples based on their difficulty level.

In essence, these methods strive to maximize inter-class variance and minimize intra-class variance, thus ensuring more effective facial feature discrimination. The primary difference lies in the techniques employed to introduce the margin into the loss function and to manage the learning process.

2.3. Attributes Extractor

The second component of face swap is the extraction of attributes that are not the same for different photos of the same person, i.e., pose, facial expression, and lighting.

With both FaceSwap and DeepFaceLab, attributes are extracted using encoders and encoded in the corresponding hidden space. By manipulating this space's size, it is possible to increase the resemblance to the original at the expense of universal features and identity. The developers of DeepFaceLab also developed an extension of the model with additional intermediate representations that were combined and transferred to the corresponding decoder to gain a better representation of lighting and color.

FaceShifter and GHOST-A utilized a multi-level attributes encoder that was responsible for encoding attributes that embedded information such as pose, light, background, or expression. In order to preserve as many attributes as possible outside the increased dimensions of the space, the developers proposed to represent these attributes on different dimensional feature maps. The network architecture used for this purpose is U-Net [40]. Attribute extraction is implemented in self-supervised training through specially selected cost functions based on spaces from different levels of the extractor model.

In the SimSwap framework, the behavior of attributes is imposed in the generator through an appropriate cost function using the discriminator. This process will be described in the following subsection.

In HiFiFace, the extraction of attribute information from a photo ultimately uses a modified encoder containing residual blocks in its structure [41].

The developers of DiffFace used three solutions for extracting/transferring attributes to the model. Two of them are in a solution called Facial Guidance, which is responsible for guiding the process of face generation in training and direct synthesis in the trained model. In this process, authors use a face parser [14] and a gaze estimator [39]. The face parser guides the face generation process so that the generated face structure matches the person to whom the source face will be applied. Using these guidelines guarantees consistency in placing facial features such as eyes, eyebrows, lips, etc. A gaze estimator guides the generation process regarding the position of the eyeballs. This component is often a critical and problematic issue in face swap models, so an additional model in this area overcomes

this problem. The third solution is to reuse the face parser, but in the final generation stage, to reconstruct elements such as the background and hair.

2.4. Generator

A vital component of the face submodels is the generator, which is an encoder that combines and processes the representations obtained in the previous steps.

In the case of FaceSwap, as well as DeepFaceLab, the process of generation, as well as extraction, as could be seen in the previous chapters, is closely intertwined. There are no clear boundaries as to where a particular split takes place. However, it results from the methodology adopted and the final swap of the decoders.

No additional enhancements were made to Faceswap, so the cost function proves that the result is based only on image reconstruction. The developers implemented such cost functions as L1 (mean absolute error), L2 (mean squared error), Logcosh, generalized loss [42], L-inf norm, DSSIM (difference of structural similarity), GMSDLoss (gradient magnitude similarity deviation loss) [43], and GradientLoss [44].

GAN models have been added to DeepFaceLab to produce more realistic results; nevertheless, their use is only suggested towards the end of training. By default, DFL utilizes a mixed loss of the DSSIM and L2. This combination is motivated by the desire to obtain both advantages: DSSIM generalizes human faces more quickly, while MSE delivers greater precision. This combination of losses aims to strike a balance between generalization and expressiveness. In addition, the cost function described previously is applied when utilizing a GAN for the generator and discriminator networks.

FaceShifter consists of two networks. The first is the Adaptive Embedding Integration Network (AEINet). It aims to transfer a person's identity from one image to another while preserving the other person's attributes. In accomplishing this task, AEINet uses the two components described in the previous sections, and a third component, the Adaptive Attentional Denormalization Generator, is responsible for integrating the above embeddings. In order to avoid generating fuzzy results resulting from simple concatenation, the developers proposed a layer called the AAD (Adaptive Attentional Denormalization). Their work was inspired by the SPADE [45] and AdaIN [46] mechanisms. SPADE introduces a new layer, called spatially adaptive normalization, for synthesizing photorealistic images from a semantic layout. In traditional models, the semantic layout is directly fed into the deep network and processed through convolution, normalization, and nonlinearity layers. However, this often results in the "washing away" of semantic information. SPADE resolves this by using the input layout to modulate the activations in normalization layers through a spatially adaptive, learned transformation. This results in images that better retain the semantic input and allows for user control over the generated image's semantic and style aspects. AdaIN provides a way to achieve style transfer in real-time by using a novel layer that aligns the mean and variance of the content features with those of the style features. Traditional style transfer methods often require a slow iterative optimization process and are usually tied to a fixed set of styles. AdaIN overcomes these limitations by allowing arbitrary style transfer without being restricted to a predefined set of styles. Moreover, it offers flexible user controls such as content-style trade-offs, style interpolation, and color and spatial controls, all with a single feed-forward neural network. One of the core concepts of the AAD layer is to dynamically change the effective regions of identity embedding and attribute embedding so that they can synthesize distinct facial features. This modification is made by using denormalization to integrate characteristics at several levels. For instance, identity embedding should concentrate on synthesizing the visual features that are most crucial for distinguishing identification, including the eyes, lips, and facial shape. The AAD layer employs a sigmoid-mask-based attention mechanism as a result.

The cost function for this network includes four functions. The first is adversarial loss resulting from using an additional discriminator that resolves the realness of the synthesized sample. For this purpose, a multi-scale discriminator was used [45]. Another

is calculating the cosine similarity of two vectors obtained from an identity encoder for an image containing the swapped identity and an input image with the given identity. In the same way, by only using L2, the output- and input-containing attribute information for different space dimensions from the attribute extractor are compared. The final cost added if the same samples are given to the input during training is the reconstruction cost function—the L2 distance between the output and the input; otherwise, this cost is not included.

In the case of FaceShifter, a second model—HEARNet (Heuristic Error Acknowledging Refinement Network)—was also introduced, which could be presented as postprocessing. It was implemented because the results did not handle occlusions adequately and instead frequently obscured them. Its architecture is U-net, and the output picture from AEINet is fed to the input. If only two images containing attributes are fed to the input, the difference between the input image whose attributes will be kept and the output image from AEINet is fed to the input.

$$Y_{s,t} = \text{HEARNet}(\text{AEINet}(X_s, X_t), (X_t - \text{AEINet}(X_t, X_t))), \quad (1)$$

where $Y_{s,t}$ denotes the output image with swapped identity, X_s denotes the input image containing identity information, and X_t denotes the input image containing attribute information. The result is a synthesized face that works better with occlusions. Three cost functions were used in the implementation of this part. As in the previous one, a cost function responsible for preserving identity—cosine similarity—was used. There is a change cost function, calculating the L1 between input and output, aimed at controlling against too much change, as well as the reconstruction function also used previously, which works on the same principle—L2—when $X_t = X_s$; otherwise, it is 0. In addition, randomly generated objects extracted from the datasets described in the preprocessing subsection are superimposed on the images to train the network, typically for occlusions.

The developers of another algorithm—SimSwap—use an encoder, decoder, and discriminator as part of the generator. The encoder input is given an image on which the face will be superimposed. The identity features obtained with the aforementioned pre-trained ArcFace network are fed into the attribute vector between the encoder and decoder. A modified version of residual blocks [41] and AdaIN [46] was used for this purpose. A multi-scale discriminator [47] based on patchGAN [48] was used as an adversarial. PatchGAN is a discriminator architecture for specific generative adversarial networks (GAN) types. It calculates whether each $N \times N$ patch in an image is real or fake. Instead of assessing the entire image at once, the PatchGAN discriminator runs convolutionally across the image, thereby penalizing the structural inconsistencies at the scale of patches. The responses from all patches are then averaged to provide the final output of the discriminator. In addition to distinguishing whether a sample is true or false, the discriminator mentioned above also has another task. In order to better preserve the attributes of a modified person's photo, weak feature matching loss based on feature matching originated in pix2pixHD [47] was introduced—this involves a photo containing attributes, and a reconstructed photo is provided to the discriminator's input. Then, using only the last few layers from the discriminator structure, the L1 between the two cases is measured. By increasing the layers used, it is possible to increase the emphasis on attribute preservation, and, conversely, by decreasing the layers used, this increases the emphasis on identity preservation.

In addition to the adversarial cost and weak feature matching loss mentioned above, there are three more cost functions. Identity loss counts the cosine similarity between the vector obtained from the identity encoder for the reconstructed image and the input containing the identity information. Another is reconstruction loss occurring when the input photos belong to the same person, thereby counting between the reconstructed photo and the one containing the attributes. The last is the gradient penalty cost [49], which is designed to prevent gradient explosion in discriminators. The gradient penalty (GP) is a regularization technique introduced in the Wasserstein GAN with Gradient

Penalty (WGAN-GP) to enforce the Lipschitz constraint, which is crucial for the optimal performance of the Wasserstein GANs.

HiFiFace combines the information obtained above using residual blocks with adaptive instance normalization [24,46]. The Semantic Facial Fusion Model has also introduced an additional component to obtain a higher resolution of the synthesized image and increase its realism. In this module, a face mask is predicted from the decoder output using HRNet [50]. This mask is then fused with the information obtained from the encoder and decoder using residual blocks. Further fusion is thus obtained, and the previously obtained mask generates a lower-resolution image. In addition, the fusion, as mentioned earlier, is also generated using upsampling—this creates a mask and an image with a higher resolution. Different resolutions are intended to affect the attentiveness of the encoder and decoder, which helps separate attributes from identities.

The cost functions used in the algorithm fall into two main subsets: 3D-shape-aware identity loss and realism loss.

The 3D-shape-aware identity loss cost function consists of:

- The cost L1 obtained from 3D facial landmarks of the face concatenated in a 3D-shape-aware identity extractor and output for both the lower resolution and higher resolution face.
- The cosine similarity obtained from an identity vector extractor [38] for the source face and the output face of both resolutions.

Realism loss consists of the following:

- A segmentation cost, which is the L1 distance of the obtained masks using HRNet for the input image containing the attributes and the output of both resolutions.
- The L1 reconstruction cost, which occurs when the people shown in the input images represent the same person. It is calculated between the input image containing the attributes and the outputs of both resolutions.
- Cycle loss—obtained by reusing the model on the output image once it is obtained, only with a change of the person containing the identity information. In the second iteration, the output image must match the image given in the first iteration as the one containing the attribute information.
- Learned perceptual image patch similarity [51] is used to increase realism and capture detail. LPIPS works by learning a perceptual similarity measure from human judgments of image similarity. It operates by comparing patches of images rather than entire images, thereby capturing local information that can be important for perceptual similarity. The image patches are processed by a deep neural network, which is specifically a variant of the VGG network that is widely used in image recognition tasks. The network is trained to predict the similarity judgments made by humans.
- Adversarial loss [52]—derived from GANs used to increase realism and capture detail.

The total cost is the sum of the above cost functions.

As mentioned earlier, SberSwap's AEI-Net base architecture draws from the FaceShifter [6]. Unlike FaceShifter, the developers used only one architecture. In addition, they tested the optimal number of AAD (Adaptive Attentional Denormalization—mentioned when describing FaceShifter) blocks and ended up using two.

The exact solutions known from FaceShifter were used within the cost function with some modifications. Within the cost function responsible for the reconstruction, suggesting the solution contained in SimSwap [7], the developers, in order to activate this cost function, did not require the input images to be the same, and it was sufficient that the person represented in the images be the same. In addition, they introduced an additional cost function based on comparing the L2 features of the eye areas between the image containing the attributes and the output image using a pre-trained detector [53].

DiffFace is a diffusion model involving iterative image reconstruction based on noise. The manipulation of this noise and conditional components during the synthesis process leads to the generation of realistic samples belonging to the data group on which the model

was trained but with different parameters. The authors directed the generation process using ID Conditional DDPM (diffusion model), Facial Guidance, and Target-Preserving Blending.

In terms of the diffusion model, by definition, U-Net [40] was used; in this case, it was based on Wide ResNet [54]. Naturally, only noise is approximated in diffusion models, and training is conducted based on it. However, in this case, a vector obtained from the identity extractor based on the source face was introduced to add a condition such as identity to the model. In order to retain the source's identity, in each training step, a sample of x_0 is approximated, then fed to the identity feature extractor, and the resulting vector is compared with the feature vector of the face fed under the condition, which adds another component to the cost function, which is cosine similarity.

In a further step, to guide the model to preserve the features of the person on whom the face is projected, Facial Guidance was introduced. In this regard, based on the x_0 approximation of the diffusion model, gradients for each component are calculated in each step: Identity Guidance, Semantic Guidance, and Gaze Guidance. The cosine distance of the Identity Guidance vector also prevents the loss of identity features during conditional sampling, thereby balancing the other two components that preserve the attributes of the face being manipulated.

The last component of the system is Target-Preserving Blending, which deals with background preservation during the face synthesis process. The developers, within the mask obtained from the face parser, increased its intensity gradually from 0 to 1 from the beginning of the synthesis to a specific step of the diffusion process T. Manipulating the starting point when the mask takes the value one allows for adaptive control and balancing of the behavior of identity features against attributes such as pose, expression, or face shape. Blending is an element-wise product of the synthesized result with an obtained target hard mask.

2.5. Postprocessing

Most authors do not refer to postprocessing as part of the algorithm's operation. However, it will not be wrong to mention the possibilities of its use to increase the sophistication of the created deep fake.

In the case of Faceswap, the creators have provided special tools for previewing the results obtained and possible adjustments to the swapped faces. These include changing the color intensity (balancing in selected palettes, including RGB or HSV), brightness, or contrast. In addition, methods for manipulating the blending of two masks have been added. This includes the box mask, which is a square field containing the face to be swapped, and the face mask, which is a mask that defines the edges of the face. The box blend is used to blend the edges of the resulting frame into the original frame. The mask blend is used to blend the edges of the mask into the original frame. The authors provided a normalization filter and a Gaussian filter. The Gaussian filter, although slower, often obtains the best results. The last setting to adjust is "scaling," or, more precisely, sharpening. Sometimes, it is needed to scale the image to fit into the final frame. Tools also provide options for artificially sharpening the image. Whether sharpening produces good results is a matter of personal preference, and the settings must be changed individually by case.

To Highlight the Most Important Algorithms Used in the methods—They Are Summarized Collectively in Table 1.

Table 1. Algorithm used in the described face swap and deep fake works.

Method	Type of Method	Source Identity Extraction	Target Attributes Extraction	Generation	Extras
[2]	One-to-one	Different Decoder	Same Encoder	AutoEncoder 11 different models	S3FD [14], 2DFAN [16], BiSeNet [17]

Table 1. *Cont.*

Method	Type of Method	Source Identity Extraction	Target Attributes Extraction	Generation	Extras
[3]	One-to-one	Different Decoder intermediate representation	Same Encoder intermediate representation	AutoEncoder 2 different models intermediate representation Optional GAN [4]	S3FD [14], 2DFAN [16], TernausNet [19], Poisson blend [55], segmenatation tool (XSeg)
[6]	Many-to-many	Arcface [36]	U-Net [40] Multi-level loss	Adaptive Attentional Denormalization inspired by AdaIN [46] Multi-scale discriminator [45]	The second stage against occlusions JCFDaA [28] Reconstruction loss for the same photo
[7]	Many-to-many	Arcface [36]	Built into Generator; Weak Feature Matching Loss from Discriminator	AdaIN [46] patchGAN [48]	Gradient Penalty [49] Reconstruction loss for the same identity
[8]	Many-to-many	CurricularFace [38]	3D Face Reconstruction model [37]	AdaIN [46] StarGAN v2 [52]	High Resolution—HRNet [50] LPIPS [51]
[9]	Many-to-many	Arcface [36]	U-Net [6,40]	Modified FaceShifter stage 1 architecture [6]	Blending Eye loss function based on [53] Reconstruction loss for the same identity [7]

The proposed face-swapping scheme in DeepFaceLab involves several steps in the postprocessing phase. The first step is to use Umeyama's reversibility [22] to transform the generated face with its mask from the swapped target decoder to the original position of the target image in the source. The next step involves blending the reconstructed face with the target image to achieve a smooth match along the outer contour. To ensure complexion consistency, DeepFaceLab offers five co-location transfer algorithms that approximate the color of the reconstructed face to the target face. These algorithms are RCT (Reinhard Color Transfer) [56], LCT (Linear Color Transfer), MKL (Monge–Kantorovitch Linear) [57], IDT (Iterative Distribution Transfer) [58], and SOT (Sliced Optimal Transfer) [59].

Poisson blending [55] seamlessly blends different skin tones, face shapes, and lighting conditions, particularly at the interface where the reconstructed face meets the designated area and the target face. This technique has helped ensure that the blended regions were visually appealing and realistic.

FaceShifter and SimSwap do not use any form of postprocessing as part of their work. In HiFiFace and DiffFace, blending is part of the network operation.

Unlike the above, GHOST-A also includes the removal of additional artifacts appearing in the periphery of images generated with the models. This correction is, therefore, significant because the entire image will be submitted for manipulation in practical use. Direct pasting of the output will result in making the mentioned artifacts visible. A segmentation mask was utilized to identify the pixels that belonged to the face and those that do not. This approach helped to isolate the facial region and ensure that the face-swapping process did not affect other parts of the image. In addition, a Gaussian blur was applied to the edges of the generated faces, and manipulation of the mask size was introduced in cases where the shapes of the generated and input faces differed significantly from each other.

2.6. Evaluation Methods

There are two evaluation methods commonly used in face swapping: objective and subjective. Objective evaluation involves quantifying the difference or correlation between the output and the target using some form of metric. The metrics used in these works are typically based on loss functions such as SSIM (structural similarity) [3], Euclidean distance for shape, pose, and expression [3,6–10], or ID retrieval [6–10]. As a subjective evaluation, the developers do not implement additional tests. However, as a review, samples of the performance of the described many-to-many algorithms are shown in Figure 2, based on the comparison provided by the DiffFace authors [10] and our experiments.

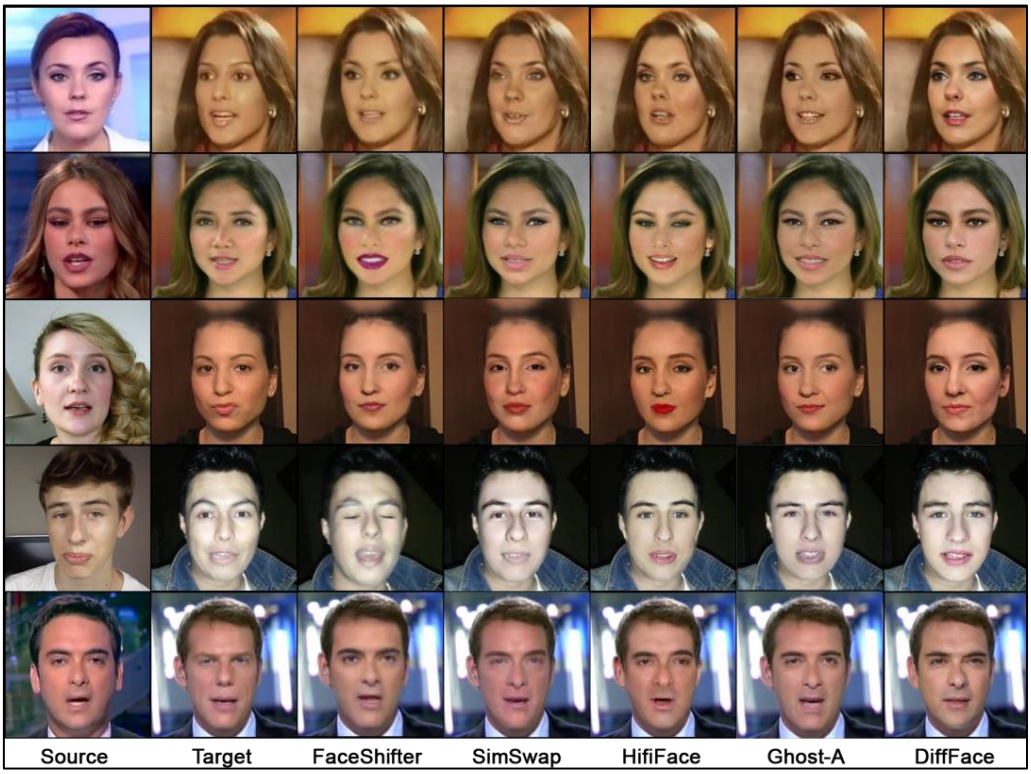

Figure 2. Qualitative comparison of results of face swap models.

Based on the subjective evaluation of the results, it can be concluded that DiffFace achieved the best outcomes regarding the quality of the generated images—this is a feature of diffusion models. However, it should be noted that, currently, these are significantly slower than other models.

When comparing the rest of the models, which are trained based on GANs, it can be observed that SimSwap provided the best results in preserving attributes. Unfortunately, this comes at the expense of accurately representing identity.

Ghost-A achieved high-quality results due to the application of additional postprocessing actions. In terms of transferring identity, HiFiFace yielded the best results.

In conclusion, each model has its strengths and weaknesses. DiffFace offers superior image quality but at the cost of speed. SimSwap excellently retains attributes but struggles with identity preservation. Ghost-A shows high-quality results with additional postprocessing, and HiFiFace stands out for its ability to transfer identity accurately. Future developments in face swap models could aim to combine these strengths to improve overall performance.

3. Challenges

Although neural networks adapt very well to new datasets, the provided algorithms solve or produce additional difficulties.

Insufficient training data could be a potential issue. In the case of face synthesis, typically, significantly more training data is required for deep neural network algorithms. Developers of algorithms are searching for ways to separate universal features for the person from universal attributes for photo/video to exchange some of these characteristics. This strategy provides access to databases that do not require advanced labeling. Most developers utilize distinct training databases, which is not optimal for a performance comparison. One-to-one and autoencoder methods require one-of-a-kind, individual input, although training is faster than in other applications. As for training, many-to-many GANs train quicker than diffusion models, although they are not always superior in performance quality. The authors also recognize the significance of training data when employing pre-trained external models for identity extraction or face parsing. Instead of training it on the same dataset as the converter, developers utilize an adequately generalized model. Federated learning can help address the challenge of insufficient training data by allowing multiple users to collaborate and train models locally without the need to share their sensitive data, thereby enhancing performance and avoiding data privacy concerns [60].

The intricacy of human faces and their occurrence of unique traits present a second difficulty. Algorithms based on a prepared model for extracting distinguishing features can miss them, producing aberrant deep fakes. One of the most challenging tasks for deep fake generation algorithms is to generate sufficiently realistic visuals to trick viewers. This generation demands sophisticated machine learning techniques and a comprehensive comprehension of human facial expressions, body motions, and behavior. The algorithm must learn how to accurately map a person's facial motions onto another person's face while preserving the same facial expressions, lip movements, and other minor features that make an image or video appear genuine. Realizing this level of realism is complex and demands a significant investment of time, data, and processing resources.

The described methods lack built-in features that increase performance with access to more data during operation. It relies solely on one image sample. Developing these types of algorithms can enhance the quality of the generated samples.

Another difficulty for deep fake creation algorithms is ensuring that the generated photos do not contain artifacts or flaws that could disclose their false character. This issue is crucial when generating images with a high degree of manipulation, as even slight errors can ruin the illusion and make the image or video appear false. In addition, the system must account for various lighting conditions, camera angles, facial expressions, occlusions, and other difficult circumstances.

Faceswap algorithm development is complicated by the nature of deep fake detection technology, which constantly evolves. As new techniques and tools are produced, it becomes increasingly challenging to create a deep fake using traditional methods; therefore, new strategies must be developed to keep up with the most recent advancements.

4. Conclusions

Face swapping is a promising, rapidly expanding research field that has the potential to improve application performance by more accurately reflecting human identity and features. While there are still obstacles to solve, the given models have demonstrated considerable potential for achieving high swapping efficiency and naturalness levels. These techniques have potential uses in various industries, including medicine, the entertainment business [61], education [62], and the military sector. By adopting strategies from complex systems such as rolling bearing fault diagnosis, the conditional weighting transfer Wasserstein autoencoder offers an innovative method for transferring knowledge between multiple source domains, thereby promising advancements in the development of face swap deep fakes [63]. There is, however, another side to the coin. These algorithms may be employed unethically [64,65]. However, we believe that making this information public is preferable, as it will help you to know your opponent. Integrating this technology into various disciplines can yield substantial benefits, such as creating new virtual characters in video games and animated films. Face swapping is a fascinating area of study that

will continue to advance. The second section's presentation of the algorithms enables an examination of the problem of deep fakes from the authors' perspectives on numerous techniques. The juxtaposition of their performance outcomes with the challenges they encounter enables a worldwide perspective on the current state-of-the-art technologies. This article explains the current state-of-the-art technologies for face swapping using deep learning techniques. Therefore, it serves as a helpful starting point for developing additional or comparable strategies.

5. Future Directions

The investigation of the offered methods revealed numerous solutions that were superior to others. This paper identifies several advantageous subsolutions and recommends selecting the most applicable ones to construct a high-efficiency method. Plans for implementing these methods are included in future work. In addition, the information in this paper on the datasets used, as well as the assessment methodologies, will offer an appropriate standard for measuring the effectiveness of future algorithms.

Although algorithm developers have been confronted with face swapping for a long time, several aspects still demand investigation. The following are some:

- Real-time performance requirements: In some instances, face swapping must be performed in real-time, which means the algorithm must execute quickly enough for the user to sees the outcome in real-time. Consequently, diffusion models still require development.
- The complicated individual nature of human identity is a task of considerable difficulty, as it necessitates an awareness of multiple facets of human beings. While the described methods are limited to images, moving to the time domain could be a potential future approach.
- The operation's adaptability to our data: The algorithm's performance must be adaptable. For better quality data, the end output must likewise adapt. Moreover, the system must adapt if various users have varying data quality.
- To implement face-swapping algorithms, it is required to determine their performance. Therefore, algorithm designers must establish quality evaluation measures that consider multiple identification characteristics.
- Satisfactory outcomes: Especially for commercial purposes, the quality of face-swapping results might be vital. Developers of algorithms are faced with the difficulty of ensuring that the algorithm's outcomes are helpful to consumers.

Author Contributions: Investigation, T.W.; methodology, T.W.; supervision, Z.P.; validation, Z.P.; resources T.W.; writing—original draft, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Centre for Research and Development, grant number CYBERSECIDENT/381319/II/NCBR/2018 on "The federal cyberspace threat detection and response system" (acronym DET-RES) as part of the second competition of the CyberSecIdent Research and Development Program—Cybersecurity and e-Identity.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Swathi, P.; Saritha, S.K. DeepFake Creation and Detection: A Survey. In Proceedings of the 3rd International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2–4 September 2021; pp. 584–588.
2. Deepfakes Deepfakes_Faceswap. Available online: https://github.com/deepfakes/faceswap (accessed on 3 May 2023).

3. Perov, I.; Gao, D.; Chervoniy, N.; Liu, K.; Marangonda, S.; Umé, C.; Dpfks, M.; Facenheim, C.S.; RP, L.; Jiang, J.; et al. DeepFaceLab: Integrated, flexible and extensible face-swapping framework. *arXiv* **2021**, arXiv:2005.05535.
4. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1411.1784. [CrossRef]
5. Mahmud, B.U.; Sharmin, A. Deep Insights of Deepfake Technology: A Review. *arXiv* **2023**, arXiv:2105.00192.
6. Li, L.; Bao, J.; Yang, H.; Chen, D.; Wen, F. FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping. *arXiv* **2020**, arXiv:1912.13457.
7. Chen, R.; Chen, X.; Ni, B.; Ge, Y. SimSwap: An Efficient Framework For High Fidelity Face Swapping. In Proceedings of the 28th ACM International Conference on Multimedia, Virtual Event/Seattle, WA, USA, 12–16 October 2020; pp. 2003–2011.
8. Wang, Y.; Chen, X.; Zhu, J.; Chu, W.; Tai, Y.; Wang, C.; Li, J.; Wu, Y.; Huang, F.; Ji, R. HifiFace: 3D Shape and Semantic Prior Guided High Fidelity Face Swapping. *arXiv* **2021**, arXiv:2106.09965.
9. Groshev, A.; Maltseva, A.; Chesakov, D.; Kuznetsov, A.; Dimitrov, D. GHOST—A New Face Swap Approach for Image and Video Domains. *IEEE Access* **2022**, *10*, 83452–83462. [CrossRef]
10. Kim, K.; Kim, Y.; Cho, S.; Seo, J.; Nam, J.; Lee, K.; Kim, S.; Lee, K. DiffFace: Diffusion-based Face Swapping with Facial Guidance. *arXiv* **2022**, arXiv:2212.13344.
11. Dhariwal, P.; Nichol, A. Diffusion Models Beat GANs on Image Synthesis. *arXiv* **2021**, arXiv:2105.05233.
12. Lu, X.; Kang, X.; Nishide, S.; Ren, F. Object detection based on SSD-ResNet. In Proceedings of the IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), Singapore, 19–21 December 2019; pp. 89–92.
13. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
14. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. S^3FD: Single Shot Scale-invariant Face Detector. *arXiv* **2017**, arXiv:1708.05237.
15. Guobing, Y. Cnn-Facial-Landmark. Available online: https://github.com/yinguobing/cnn-facial-landmark (accessed on 3 May 2023).
16. Bulat, A.; Tzimiropoulos, G. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1021–1030.
17. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. *arXiv* **2018**, arXiv:1808.00897.
18. Nirkin, Y.; Masi, I.; Tran, A.T.; Hassner, T.; Medioni, G. On Face Segmentation, Face Swapping, and Face Perception. *arXiv* **2017**, arXiv:1704.06729.
19. Iglovikov, V.; Shvets, A. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. *arXiv* **2018**, arXiv:1801.05746.
20. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep Face Recognition. In Proceedings of the British Machine Vision Conference 2015, British Machine Vision Association, Swansea, UK, 7–10 September 2015; pp. 41.1–41.12.
21. Feng, Y.; Wu, F.; Shao, X.; Wang, Y.; Zhou, X. Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network. *arXiv* **2018**, arXiv:1803.07835.
22. Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380. [CrossRef]
23. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv* **2018**, arXiv:1710.10196.
24. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv* **2019**, arXiv:1812.04948.
25. Bambach, S.; Lee, S.; Crandall, D.J.; Yu, C. Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1949–1957.
26. Fathi, A.; Ren, X.; Rehg, J.M. Learning to recognize objects in egocentric activities. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 3281–3288.
27. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.
28. Chen, D.; Ren, S.; Wei, Y.; Cao, X.; Sun, J. Joint Cascade Face Detection and Alignment. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 109–122. [CrossRef]
29. Cao, Q.; Shen, L.; Xie, W.; Parkhi, O.M.; Zisserman, A. VGGFace2: A dataset for recognising faces across pose and age. *arXiv* **2018**, arXiv:1710.08092.
30. Liu, N. VGGFace2-HQ. Available online: https://github.com/NNNNAI/VGGFace2-HQ (accessed on 3 May 2023).
31. Wang, X.; Li, Y.; Zhang, H.; Shan, Y. Towards Real-World Blind Face Restoration with Generative Facial Prior. *arXiv* **2021**, arXiv:2101.04061.

32. InsightFace: 2D and 3D Face Analysis Project. Available online: https://github.com/deepinsight/insightface (accessed on 3 May 2023).
33. Trillionpairs. Available online: http://trillionpairs.deepglint.com/overview (accessed on 3 May 2023).
34. Li, X.; Chen, C.; Zhou, S.; Lin, X.; Zuo, W.; Zhang, L. Blind Face Restoration via Deep Multi-scale Component Dictionaries. *arXiv* **2020**, arXiv:2008.00418.
35. Rössler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; Nießner, M. FaceForensics++: Learning to Detect Manipulated Facial Images. *arXiv* **2019**, arXiv:1901.08971.
36. Deng, J.; Guo, J.; Yang, J.; Xue, N.; Kotsia, I.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5962–5979. [CrossRef]
37. Deng, Y.; Yang, J.; Xu, S.; Chen, D.; Jia, Y.; Tong, X. Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set. *arXiv* **2020**, arXiv:1903.08527.
38. Huang, Y.; Wang, Y.; Tai, Y.; Liu, X.; Shen, P.; Li, S.; Li, J.; Huang, F. CurricularFace: Adaptive Curriculum Learning Loss for Deep Face Recognition. *arXiv* **2020**, arXiv:2004.00288.
39. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. CosFace: Large Margin Cosine Loss for Deep Face Recognition. *arXiv* **2018**, arXiv:1801.09414.
40. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
42. Barron, J.T. A General and Adaptive Robust Loss Function. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4331–4339.
43. Xue, W.; Zhang, L.; Mou, X.; Bovik, A.C. Gradient Magnitude Similarity Deviation: A Highly Efficient Perceptual Image Quality Index. *arXiv* **2013**, arXiv:1308.3052. [CrossRef]
44. Lu, C.; Huang, H. TV + TV2 Regularization with Nonconvex Sparseness-Inducing Penalty for Image Restoration. *Math. Probl. Eng.* **2014**, *2014*, 790547. [CrossRef]
45. Park, T.; Liu, M.-Y.; Wang, T.-C.; Zhu, J.-Y. Semantic Image Synthesis with Spatially-Adaptive Normalization. *arXiv* **2019**, arXiv:1903.07291.
46. Huang, X.; Belongie, S. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. *arXiv* **2017**, arXiv:1703.06868.
47. Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. *arXiv* **2018**, arXiv:1711.11585.
48. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv* **2018**, arXiv:1611.07004.
49. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning, PMLR, Sydney, NSW, Australia, 6–11 August 2017; pp. 214–223.
50. Sun, K.; Zhao, Y.; Jiang, B.; Cheng, T.; Xiao, B.; Liu, D.; Mu, Y.; Wang, X.; Liu, W.; Wang, J. High-Resolution Representations for Labeling Pixels and Regions. *arXiv* **2019**, arXiv:1904.04514.
51. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *arXiv* **2018**, arXiv:1801.03924.
52. Choi, Y.; Uh, Y.; Yoo, J.; Ha, J.-W. StarGAN v2: Diverse Image Synthesis for Multiple Domains. *arXiv* **2020**, arXiv:1912.01865.
53. Wang, X.; Bo, L.; Fuxin, L. Adaptive Wing Loss for Robust Face Alignment via Heatmap Regression. *arXiv* **2020**, arXiv:1904.07399.
54. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. *arXiv* **2017**, arXiv:1605.07146.
55. Pérez, P.; Gangnet, M.; Blake, A. Poisson image editing. In *ACM SIGGRAPH 2003 Papers, Proceedings of the SIGGRAPH03: Special Interest Group on Computer Graphics and Interactive Techniques, San Diego, CA, USA, 27–31 July 2003*; Association for Computing Machinery: New York, NY, USA, 2003; pp. 313–318. [CrossRef]
56. Reinhard, E.; Adhikhmin, M.; Gooch, B.; Shirley, P. Color transfer between images. *IEEE Comput. Graph. Appl.* **2001**, *21*, 34–41. [CrossRef]
57. Pitie, F.; Kokaram, A. The linear Monge-Kantorovitch linear colour mapping for example-based colour transfer. In Proceedings of the 4th European Conference on Visual Media Production, London, UK, 27–28 November 2007; pp. 1–9.
58. Pitié, F.; Kokaram, A.C.; Dahyot, R. Automated colour grading using colour distribution transfer. *Comput. Vis. Image Underst.* **2007**, *107*, 123–137. [CrossRef]
59. Coeurjolly, D. Color Transfer via Sliced Optimal Transport. Available online: https://github.com/dcoeurjo/OTColorTransfer (accessed on 3 May 2023).
60. Zhao, K.; Hu, J.; Shao, H.; Hu, J. Federated multi-source domain adversarial adaptation framework for machinery fault diagnosis with data privacy. *Reliab. Eng. Syst. Saf.* **2023**, *236*, 109246. [CrossRef]
61. Usukhbayar, B. *Deepfake Videos: The Future of Entertainment*; Research Gate: Berlin, Germany, 2020.
62. Westerlund, M. The Emergence of Deepfake Technology: A Review. *Technol. Innov. Manag. Rev.* **2019**, *9*, 40–53. [CrossRef]
63. Zhao, K.; Jia, F.; Shao, H. A novel conditional weighting transfer Wasserstein auto-encoder for rolling bearing fault diagnosis with multi-source domains. *Knowl.-Based Syst.* **2023**, *262*, 110203. [CrossRef]

64. Karasavva, V.; Noorbhai, A. The Real Threat of Deepfake Pornography: A Review of Canadian Policy. *Cyberpsychol. Behav. Soc. Netw.* **2021**, *24*, 203–209. [CrossRef] [PubMed]
65. Wojewidka, J. The deepfake threat to face biometrics. *Biom. Technol. Today* **2020**, *2020*, 5–7. [CrossRef]