applied sciences

MDPI

Review

Overview of Voice Conversion Methods Based on Deep Learning

Tomasz Walczyna and Zbigniew Piotrowski *

Institute of Communication Systems, Faculty of Electronics, Military University of Technology, 00-908 Warsaw, Poland
* Correspondence: zbigniew.piotrowski@wat.edu.pl

Abstract: Voice conversion is a process where the essence of a speaker's identity is seamlessly transferred to another speaker, all while preserving the content of their speech. This usage is accomplished using algorithms that blend speech processing techniques, such as speech analysis, speaker classification, and vocoding. The cutting-edge voice conversion technology is characterized by deep neural networks that effectively separate a speaker's voice from their linguistic content. This article offers a comprehensive overview of the development status of this area of science based on the current state-of-the-art voice conversion methods.

Keywords: voice conversion; voice disentangling; voice synthesis

check for updates

Citation: Walczyna, T.; Piotrowski, Z. Overview of Voice Conversion Methods Based on Deep Learning. *Appl. Sci.* **2023**, *13*, 3100. https://doi.org/10.3390/app13053100

Academic Editors: Yoshinobu Kajikawa and Cheng-Yuan Chang

Received: 30 January 2023
Revised: 23 February 2023
Accepted: 24 February 2023
Published: 28 February 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Voice conversion using artificial intelligence is an essential field of science. It is the science of transforming one voice to sound like another person's voice without changing the linguistic content [1]. Voice conversion belongs to the general technical field known as speech synthesis, which converts text into speech or changes speech properties, such as voice identity, emotion, or accent [2]. Neural network approaches that were recently considerably affected the development of numerous voice converter applications [3]. Nowadays, most synthesis techniques and algorithms include a deep learning component [4]. Voice conversion using neural networks is a rapidly expanding discipline with significant breakthroughs. This review aimed to quickly bring readers up to date on the most recent developments in this technology field. This article summarizes the state-of-the-art neural-network-based voice converter techniques focusing on recent advancements. The article discusses the most significant technological developments and explains how they have enhanced the effectiveness and quality of voice conversion. The article also discusses particular issues that still need to be resolved and offer predictions for the direction of voice conversion research. Whether the reader is a researcher, a practitioner, or simply someone interested in this topic, this article provides a comprehensive introduction to the latest advancements in voice conversion using neural networks.

Currently, a typical voice conversion procedure consists of a part analyzing and decomposing the voice to extract individual components/characteristics and a piece involving the mapping/combining of the extracted elements via reconstructions using a vocoder [5].

The workflow of analysis–mapping–reconstruction also changes as a result of deep learning approaches. The mapping must successfully obtain an appropriate intermediate representation of the speech. Embedding in deep understanding provides a new way of deriving indirect expression, for example, latent code for linguistic content and speaker embedding for speaker identity. It also makes it easier to separate the speaker from the scope of the speech [6].

This study aimed to break down the conversion into its components and present the current testing status in each area. These additional components are as follows (also shown in Figure 1):

- Speaker identity extraction—this involves extracting information about the speaker's identity from the speech.
- Linguistic content extraction—this involves extracting from statements or appropriately processing other data (e.g., text) to obtain the most time-dependent information in the output. These include information about the content of speech, rhythm, and intonation.
- Encoder—this is responsible for the integration and appropriate representation of the above extractions. As the information fed into the encoder and the information obtained from the extraction of linguistic content latent embeddings are time-dependent, these two tasks are often combined.
- Decoder/vocoder—these are responsible for processing the data obtained from the encoder to produce an appropriately manipulated soundtrack in the output. The input is frequently a spectrogram. However, sometimes, to reduce the number of models or unnecessary intermediate representations [7], the encoder is combined with the vocoder, and there is no intermediate representation between them.

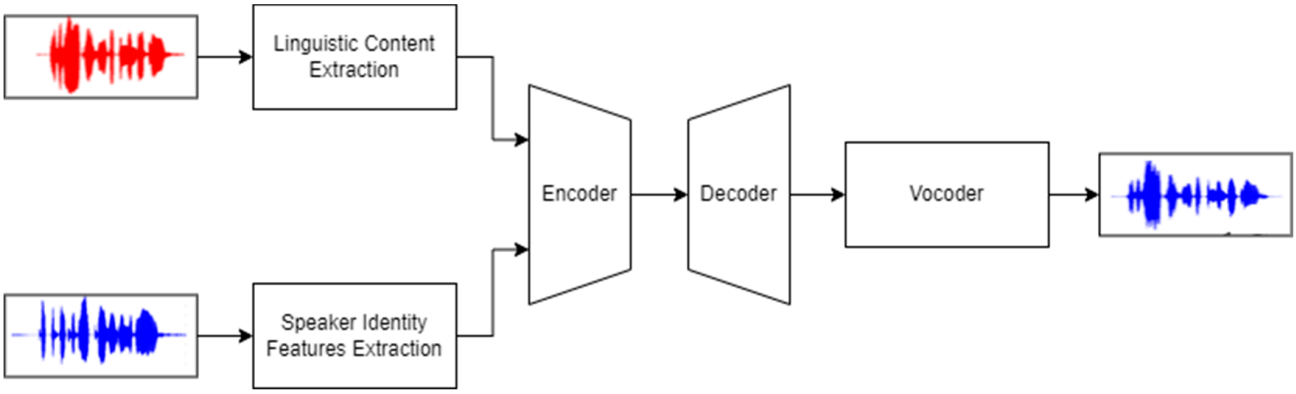

Figure 1. Typical voice conversion pipeline. The input is most often a waveform or spectrogram. A distortion-free new soundtrack is produced by extracting features from this data and combining them in the encoder before further processing.

Some of the presented algorithms represent only a part of the task, e.g., proper separation of sound components; however, these are described because of their potential use in the functions mentioned earlier. Furthermore, to maximize the timeliness of the review, much of the algorithms presented will be from recent work.

The breakdown shown above is for illustrative purposes; it is not as readily apparent in every algorithm.

Developers recently created an increasing number of zero-shot voice conversion algorithms. An unsupervised zero-shot voice conversion (VC) aims to change the speaker's characteristics in an utterance to match an unseen target speaker without using parallel training data.

2. Voice Conversion Process

To attain high performance in voice conversion, the models must undergo pre-training with vast amounts of data, resulting in large models that may be inefficient to use. Mingjie Chen et al. [8] presented a model that is significantly smaller and, therefore, faster to process while still providing the same performance. For this purpose, dynamic GAN-VC (DYGAN-VC) uses a non-autoregressive structure and vector quantized embeddings obtained from a VQWav2vec [9] model. In addition, they introduced dynamic convolution [10] to improve the modeling of speech content while requiring the use of a small number of parameters.

In [7], authors proposed an any-to-any voice conversion pipeline. They offered an approach that uses automated speech recognition (ASR), pitch tracking, and SOTA functions of a modified vocoder.

Two-stage pipelines using low-level indirect speech representations, such as mel spectrograms, dominated recent advances in neural voice processing tests. The authors in [11] noted that a data-driven approach to learning implicit representations could not fully realize its potential due to the limitations of predetermined characteristics. They proposed WavThruVec, which is a two-stage architecture that solves the bottleneck using multidimensional embeddings as an intermediate representation of speech.

A variational autoencoder (VAE) [12] is a neural network that separates speech into two parts: speaker identity and language content. It then uses this information to make a target speaker sound like the source speaker. The VAE concatenates the target speaker's identity embedding and the source speaker's content embedding to accomplish the separation and deliver the desired sentence. The researchers in [13] found that adding a self-observation layer to the VAE decoder could improve the accuracy of the speaker classification in voice transformation. This layer uses non-local information and hides the source speaker's identity to create more accurate transformed speech.

Similarly, in [14], zero-shot voice conversion was performed by inserting any speaker embedding and content obtained from the encoders into the VAE decoder. In addition, they used a learning strategy by augmenting on-the-fly data during training to make the learned representation resistant to interference.

The authors of [15] proposed a voice conversion method using a general adversarial network (GAN) called StarGAN v2. The many-to-many conversion model does not need large data sets. It is not, however, designed for zero-shot VC. The style encoder used is capable of transferring not only the characteristics of the speech read but also the emotions to another voice. The presented model is fully convolutional and, combined with a suitable vocoder, can operate in real-time conversion.

The developers of the FreeVC algorithm [16] presented a compelling approach using GANs, which is an extension of the text-to-speech method VITS [17]. This method also implements an extended conditional VAE and other innovative mechanisms, such as a monotonic alignment search using normalizing flows [16,17].

2.1. Speaker Identity Extraction

The fundamental aspect of voice conversion is how the model is informed about a speaker's voice characteristics. The model uses these voice characteristics during the conversion process to produce the appropriate timbre in the speech. A modern method is the use of the D-vector [4]. The mentioned article presents the application of deep neural networks for verification. During the training stage, the model calibrates itself to classify speakers correctly. The trained network extracts speaker-specific characteristics from the last hidden layer when recording a speaker's speech. The average of these characteristics, or the D-vector, is taken as the speaker model. At the evaluation, the network extracts a D-vector for each voice and compares it with the speaker model to verify the speaker. This algorithm obtained excellent results for the verification task; however, the zero-shot conversion uses speakers whose speech samples are not present in the training set. When this happens, using a more generalized characteristics map is better [18]. The authors proposed a generalized end-to-end loss for speaker verification, which achieved state-of-the-art results regarding speaker verification. The researchers applied the proposed system in several VC and voice synthesis systems, including SV2TTS [19], AutoVC [20], and DYGAN-VC [9]. GE2E employs a cost function that moves speech representation toward the centroid of speech by the same speaker and away from the centroid of speech by a different speaker.

For better generalization and better placement of samples in embedded space, algorithm developers also use a normal distribution. The output of the network produces an average and an expected value. An example is then taken from such a distribution to represent information about the speaker's identity. To increase the similarity to the normal distribution, an additional Kullback–Leibler cost function is then used, which is also used

in the variational auto-encoders described later [12]. A speaker-encoder structure was used in HiFi-VC [7].

Conventional speaker embedding averages frame-level characteristics over all frames in a single speech. Some algorithms, such as WavThruVec [11], use a self-attention mechanism to give different weights to different structures and generate weighted averages and standard deviations. Self-attention, which is sometimes referred to as intra-observation, is an attention mechanism that refers to the different positions of a single sequence to compute its representation [21]. This mechanism captures long-term changes in speaker characteristics more effectively. WavThruVec uses the ECAPA-TDNN architecture as the speaker encoder [22]. Despite the considerable difference in the final analysis of speaker similarity between the use of seen-speech and zero-shot conversions, using a significant amount of training data led to an increased generalization of the speaker feature extractor.

Long et al. [13] improved the efficiency of speaker style transfer to the VAE in voice conversion by adding a self-attention mechanism. They also trained the model using the group lasso division method (RGSM [23]) to capture speaker information's global and interdependent nature over a significant period and mitigate overfitting [23]. However, it is essential to note that the authors did not create a separate representation of the speaker that is added during the model training. Instead, they jointly separated it from the linguistic content in the VAE. The decoder separates the content embedding from the speaker latent space by taking a content embedding from the distribution of a single speech and a speaker embedding from the average distribution of all samples of voices from the same speaker group.

The D-DSVAE developers also separate speaker traits from track and speech traits. [14]. In this case, to correctly separate the information, the authors tested different-sized weights with the Kulbak–Leibner cost function for the speaker features and the speech. In addition, they assumed that the speaker characteristics were not time-dependent, and thus, they used average pooling on them by suggesting a suitably modified 1D InstanceNorm method [24].

In addition to VAE, adversarial networks, namely, GANs, are commonly used in voice conversion. The developers of StarGANv2-VC [15] extracted information about the speaker's speech style and identity from the mel spectrogram. The system removes the speaker's characteristics by adding a cost function from the discriminator, which classifies a sound's authenticity and speaker identity. Furthermore, to induce the generator to create samples with different styles, a cost function calculates the mean absolute error between examples of different styles and maximizes it. The style encoder minimizes the difference between the style obtained from the actual sample and the style generated by the entire pipeline.

FreeVC [17] tested the performance of two speaker feature extraction methods: pre-trained and non-pre-trained. The developers of the first type of method trained them on a large set of speakers, similar to those using GE2E. However, they utilized a different model instead, namely, BNE-Seq2seqMoL [25]. The authors noted that implementing the pre-trained method in this algorithm did not produce significantly better results than training the speaker encoder with the other parts from scratch. These results indicate that if the linguistic representation is extracted correctly, the speaker encoder learns the missing parts describing the identity.

2.2. Linguistic Content Extraction

The second component of conversion is how the linguistic information of the speech, i.e., the content, and the phonemes, is communicated to the encoder. Rhythm, intonation, and emotion can be conveyed through the speaker encoder and other encoders, as well as omitted or added. However, isolating the content of the speech from the track is the most crucial part.

The developers of DYGAN-VC [8] used a VQWav2vec model [9] in their work, the main aim of which was to present a lighter voice conversion model while maintaining

SOTA results for linguistic information extraction. The generator receives the data extracted from the soundtrack and the pre-trained speaker embedding with its help.

Ref. [7] noted that pretreated ASR models extracted little of the speaker's characteristics from the soundtrack. Furthermore, once trained, they can serve as a cost function to minimize the loss of linguistic data. This application's downside is that it does not convey rhythmic or tonal information. The team utilized an additional F0 fundamental frequency extractor to address this problem. The ASR and F0 encoder is similar to that used in TTS Skins [26], except F0 is additionally preprocessed using a network that is almost identical to BNE-Seq2seqMoL [25]. During training, the authors only modified the F0 weights of the encoder and the speaker encoder mentioned above. The developers used a conformer [27] ASR model pre-trained using NVIDIA in the linguistic encoder.

The pre-trained model was also used in WavThruVec [11]. Although the work itself mainly refers to voice synthesis, it also engages with conversion in its operation. Based on transformers, the applied pre-trained speech recognition model Wav2vec 2.0 [28] performs well as a characteristic linguistic extractor. Because these latent activations derived from this model provide high-level linguistic characteristics, they are more resistant to noise. The decoder can be trained on large, non-transcribed audio corpora, as Wav2vec 2.0 embeddings are time-aligned and speaker-independent.

Using VAE, Long et al. [13] and D-DSVAE [14] carried out the separation of linguistic characteristics in the same way they separated speaker characteristics, with the exception that they treated these characteristics as an individual for each audio track. They did not introduce grouping or averaging for them. However, they required suitably adjusted weights for the cost function during training.

In StarGANv2-VC [15], the mel spectrogram containing the speech content is fed directly to the encoder/generator. The only additional attribute extracted from the sound beyond the styles is the fundamental frequency extracted using the pre-trained JDC network [29].

Extracting linguistic information in FreeVC [16] is more complex, as it varies based on whether the model is in use during the training or inference process. The developers used the prior pre-trained WavLM model [30] to extract linguistic features, which were then fed to the bottleneck to reduce speaker information and noise. In addition, the authors implemented a mechanism to change the dimension of the input data to degrade individual speaker features in the spectrogram. This implementation reduces the need for fine-tuning the dimensions of the bottleneck in the linguistic extractor. The authors then projected the latent representation into the mean and variance of the distribution. The normalization flow, conditioned on the speaker embedding, was adapted to improve the complexity of the prior distribution. VITS [17] envisions it as consisting of multiple affine coupling layers [31] designed to preserve volume with a Jacobian determinant of 1.

2.3. Generation

As the importance of the intermediate representation obtained from the encoder processing the audio tracks is negligible in the VC pipeline, these issues are often combined.

StarGANv2-VC [15], which is based on GANs, uses a single discriminator and generator to generate audio tracks with speaker-specific style vectors derived from a style encoder. The encoder processes the soundtrack in the generator, and the JDC network mentioned in the previous section extracts the fundamental frequency. The characteristics obtained at the output are then fed to a decoder, to which style information is also added using AdaIN [32]. The result of the decoder, and thus, the generator, is a mel spectrogram with an altered style. The discriminator, on the other hand, consists of two networks. In addition to the classic one used in GANs (the real/fake classifier), an additional model is responsible for speaker classification. A pre-trained Parallel WaveGAN was used to convert the mel spectrogram to a wave [33].

DYGAN-VC [8], similar to StarGANv2-VC based on its assumptions on GAN, uses the generator and discriminator models. In addition, the authors compared two ways

of cross-adapting data obtained from VQWav2vec and the speaker encoder. These were AdaIN [32] (also used in StarGANv2-VC) and WadaIN [34]. Ultimately, they chose to use the latter because it indicated the creation of better-quality sound. As mentioned earlier, the generator structure includes dynamic convolution and the WadaIN. The characteristics obtained from the VQWav2vec [9] and speaker characteristics obtained from the speaker encoder are applied to the generator, while they are only used to the WAdaIN block. In turn, a spectrogram is received at the output. This speech is then fed to a discriminator, which decides whether it is false or true. The discriminator uses the same architecture as StarGANv2-VC [15], and Parallel WaveGAN [33] was used as the vocoder.

Authors of HiFi-VC modified the commonly used HiFi-GAN vocoder [35]. As HiFi-GAN is a separate model responsible for converting a mel spectrogram into a waveform, the developers changed it to receive characteristics obtained from speaker characteristics extraction and linguistic characteristics as input. The authors omitted the intermediate representation and only presented the mel spectrogram at the encoder inputs. In addition, the speaker characteristics are fed directly into the residual blocks of the modified HiFi-GAN generator. The VC pipeline is considerably simplified by this approach, and it does not require fine-tuning the vocoder once the decoder has been trained.

Authors of WavThruVec [11] used a GAN model and cost functions based on HiFi-GAN as the decoder responsible for combining and processing linguistic and speaker information.

The authors of D-DSVAE [14] used a modified model from AutoVC for the encoder responsible for data separation and the decoder. During training, they operated with one encoder and a single audio track. Still, during conversion, they used two encoders with two audio tracks: one for extracting linguistic data and the other for identity. Then, they combined these characteristics and fed them to the decoder, which had the task of reconstructing the spectrogram at the output. The vocoder used in this work, which was responsible for converting the spectrogram into a wave, was WaveNet [36], but the authors noted that it only served for inference and was not used during training. The results of the modified sound were strongly dependent on the weights used during training.

In the case of Long et al. [13], although the training principle was very similar to that of the D-DSVAE, as one encoder was used for training and two for conversion, the models used were different, and the mel frequency cepstral coefficients were used as input and output data. In addition, the decoder structure implemented an attention mechanism responsible for catching long-range dependency. A WaveNet was used as a vocoder as in D-DSVAE.

FreeVC [16] includes a posterior encoder, decoder, and discriminator in its architecture in addition to the speaker encoder and prior encoder described previously. The posterior encoder is used only during training to train the latent space. The prior distribution mentioned earlier in the linguistic feature extraction framework must be close to the posterior distribution conditioned by the linear spectrogram. This work was done to correctly estimate the match between the content of the utterance and the target whose voice is synthesized. The decoder receives the output of the posterior encoder or prior encoder, depending on whether the model is trained or not, respectively. The decoder and discriminator used models of the HiFi-GAN algorithm [35].

2.4. Vocoders

Vocoders are tools used to convert a speech spectrogram into sound waves. They are crucial to the voice conversion process, as they enable the generation of the appropriate sound based on the spectrogram. Table 1 compares the described methods in terms of the solutions used. The described methods use Parallel WaveGAN, WaveNet [36], and HiFi-GAN [35].

Table 1. Models used in the described VC works. Abbreviations: LN—layer normalization, RC—residual connections, and LSTM—long short-term memory.

Paper	Speaker Identity Extraction	Linguistic Content Extraction	Generation	Vocoder
[8]	GE2E [18]	VQWav2vec features [9]	Dynamic convolutions [10] WadaIN [34] LN + RC as in [21]	Parallel WaveGAN [33]
[7]	5-layer residual FC similar to [37]	Linguistic encoder: TTS Skins [26] Conformer [27] F0 encoder: BNE-Seq2seqMoL [25]	Modified HiFi-GAN [35]	Modified HiFi-GAN [35]
[11]	ECAPA-TDNN [22]	Wav2vec [28]	Vec2wav model based on HiFi-GAN [35]	Vec2wav model based on HiFi-GAN [35]
[13]	β-VAE [38]—average distribution	β-VAE [38]—individual distribution for each audio	β-VAE [38] RGSM [23] Attention [21] Post-Net as in [39]	WaveNet [36]
[14]	Modified DSVAE [40]—time-invariant disentanglement	Modified DSVAE [40]—time-variant disentanglement	Modified AutoVC Decoder [20]	WaveNet [36] HiFi-GAN [35]
[15]	Mapping network/style encoder	Encoder + F0 Encoder: JDC network [29]	Encoder output + F0 output + style injected by AdaIN [32]	Parallel WaveGAN [33]
[16]	LSTM based on [25]	Prior encoder: WavLM [30] bottleneck extractor posterior encoder based on flow used only during training	HiFi-GAN [35]	HiFi-GAN [35]

WaveNet is a neural network architecture that DeepMind proposed in 2016. WaveNet is a generative model that can generate audio data based on previous predictions. The authors proposed a model architecture based on PixelCNN [41], which is applied to images. WaveNet is very effective at generating natural voice sounds because it relies on dilated convolutions, which allow for modeling a wide range of information in the input data. Dilated convolutions previously used in signal processing were used in various contexts, e.g., signal processing [42,43] and image segmentation [44,45]. In addition, the authors noted that using gated activation units in PixelCNN works better in modeling audio signals than rectified linear activation functions [46]. Authors also used residual connections [47] to accelerate training for deeper models.

Parallel WaveGAN is a GAN-based generative model that was proposed in 2020. WaveGAN is specifically designed to generate audio data, such as nature sounds or music. The model uses GAN to develop an artificial audio signal. WaveGAN models are very effective at generating audio data that is highly realistic and natural. The developers used two models: a generator and a discriminator typically used for GANs [48]. As part of the generator, they used a modified WaveNet, but the authors used non-causal convolutions instead of causal convolutions; the input is random noise drawn from a Gaussian distribution, and the model is non-autoregressive at both the training and inference steps. In addition, to increase the stability, they introduced multi-resolution STFT auxiliary loss [49,50]. Combining multiple STFT losses with different analysis parameters helps the generator to obtain the time-frequency characteristics of speech [51].

As research showed [52], the best performance in vocoder tasks among the presented three was achieved by the third model: the HiFi-GAN. The authors of the generator developed a proprietary model using multi-receptive field fusion to work in parallel on

patterns of different lengths. They increased the receptive field to counter problems such as identifying long-term dependencies, according to [53]. In addition, they focused on the issue of identifying diverse periodic patterns. For this purpose, they used two discriminator models: the author's multi-period discriminator and the multi-scale discriminator proposed in MelGAN [54].

2.5. Datasets

It is necessary to use the best possible datasets to achieve good results. For attention-based algorithms, a large dataset is essential.

The most commonly selected [7,11,13,14,16] dataset for training and testing in the described works was the VCTK [55]. Some results [11,14] supplemented it with datasets such as Hi-Fi TTS [56], LibriSpeech [57], CommonVoice [58], AVSpeech [59], or TIMIT [60]. Only [8] used a different dataset, the VCC2020 [61], partly because the work was a comparison with another model that used this dataset. In [15], to demonstrate the conversion ability of stylized speech, a model for additional datasets—JVS [62] and ESD [63]—was trained separately.

2.6. Model Inputs

Most VC studies [7,8,14,16] used logarithmic mel spectrograms as the soundtrack input. The same was true for [11] as an input to the speaker encoder. However, for the pre-trained Wav2vec, the authors gave a wave as the input. In [7], the authors gave the F0 fundamental frequency to extract information such as tonality. In [13], the MFCCs (mel frequency cepstral coefficients) were used instead of the mel spectrogram.

2.7. Evaluation Methods

Two evaluation methods are used in voice conversion: objective and subjective. In general, objective evaluation involves calculating some measure of difference or correlation between the outcome and the target. The works used the WER [7,8,16] (word error rate) and CER [7,8,15,16] (character error rate) to evaluate linguistic consistency. The authors of [8] also used mel-cepstral distortion (MCD) [8] to measure spectral changes during conversion. In [7,16], the PCC (Pearson correlation coefficient) was used to calculate prosody consistency. In [13], the effectiveness of speaker ratings shows the influence of attention mechanisms on this factor. A classification accuracy (CLS) metric was also introduced in [15]. The authors of [14] used the EER (equal error rate) as a metric for the quality of data separation. In addition, [7,8,11,14–16] provided a mean opinion score (MOS) for naturalness and similarity. Some works [8,13,15] used MOSnet, which simulated MOS feedback. Table 2 shows the resulting MOS values of the described algorithms.

Table 2. Information about the described VC works. Abbreviations: CER—character error rate, CLS—classification accuracy, EER—equal error rate, MCD—mel-cepstral distortion, MOS—mean opinion score, PCC—Pearson correlation coefficient, WER—word error rate., M-M—many-to-many, and A-A—any-to-any, ✓—exist, ✗—no exist.

Paper	Evaluations Methods	MOS M-M Quality	MOS M-M Similarity	MOS A-A Quality	MOS A-A Similarity	Dataset	Public Code/Demo
[8]	MCD/MOS/CER/ WER/MOSNet [64]	3.81	3.87	-	-	VCC2020 [61]	✓ [1]/✓ [2]
[7]	MOS/WER/CER/PCC	4.08	4.08	4.03	3.02	VCTK [55]	✓ [3]
[11]	MOS	4.09	-	-	-	VCTK [55], Hi-Fi TTS [56] LibriSpeech [57], CommonVoice [58], AVSpeech [59]	✗/✓ [4]

Table 2. *Cont.*

Paper	Evaluations Methods	MOS M-M Quality	MOS M-M Similarity	MOS A-A Quality	MOS A-A Similarity	Dataset	Public Code/Demo
[13]	MOSNet [64], average speaker classification accuracy	3.74	-	3.58	-	VCTK [55]	✗/✗
WaveNet [14]	EER/MOS	3.40	3.56	3.22	3.54	VCTK [55], TIMIT [60]	✗/✓ [5]
HiFi-GAN [65]		3.76	3.83	3.65	3.89		
[15]	MOS/MOSNet [64]/CLS/CER	4.09	3.86	-	-	VCTK [55], JVS [62], ESD [63],	✓ [6]/✓ [7]
[16]	MOS/WER/CER/PCC	4.01	3.80	4.06	2.83	VCTK [55], LibriTTS [66]	✓ [8]/✓ [9]

[1] https://github.com/mingjiechen/dyganvc (accessed on 29 January 2023), [2] https://mingjiechen.github.io/dygan-vc/ (accessed on 29 January 2023), [3] https://github.com/tinkoff-ai/hifi_vc (accessed on 29 January 2023), [4] https://charactr-platform.github.io/WavThruVec/, [5] https://jlian2.github.io/Robust-Voice-Style-Transfer/ (accessed on 29 January 2023), [6] https://github.com/yl4579/StarGANv2-VC (accessed on 29 January 2023), [7] https://starganv2-vc.github.io/ (accessed on 29 January 2023), [8] https://github.com/OlaWod/FreeVC (accessed on 29 January 2023), [9] https://olawod.github.io/FreeVC-demo/ (accessed on 29 January 2023).

3. Challenges

Although neural networks themselves adapt very well to new datasets, there are other problems that the presented algorithms solve or, in contrast, generate.

One problem may be the lack of sufficient training data. Algorithms of deep neural networks need a large amount of training data to train neural networks correctly, and in the case of voice conversion, even more data is usually needed. Algorithm developers are looking for solutions to disentangle time-variant features from these time-invariant ones to swap some of these features. This approach opens access to databases that do not require advanced labeling. Thus, developers mostly use a single training database (VCTK [55]), which allows for a good performance comparison in a constrained environment. Concerning the presented applications, algorithms using VAE require fewer training data than those based on GANs or attentions. The authors also see the importance of the training data in using an external pre-trained vocoder. Developers do not train it from scratch for the same dataset as the converter but instead use an already appropriately generalized model.

Another problem is the complexity of human speech and its appropriate resolution. VAE-based algorithms often require careful adjustment of the bottleneck, which will prevent redundant information from being transferred to the generated signal, and thus, degrading it. GANs themselves, on the other hand, are not always able to build a sufficiently generalized distribution to allow for the generation of samples outside the training data area. This problem is evident, for example, in StarGANv2-VC [15], which does not even include any-to-any covariance in its coverage, or even in FreeVC [16], where one can see the poor performance of the similarity MOS for out-of-sample data. The developers of FreeVC [16] and HiFi-VC [7] implemented a combination of these models, obtaining excellent results in many-to-many conversion but worse results in SMOS for any-to-any conversion.

One standard procedure for developing a conversion pipeline is for developers to use pre-trained speaker encoders. This action can generate problems related to the inadequate quality of a specific component contributing to a decrease in the entire system's performance. However, algorithms are emerging, such as the input-degrading FreeVC spectrogram [16] or the fully VAE-based [11], which integrate speaker feature extractor training into a single training loop jointly with linguistic feature extraction, which solves this problem to some extent. In addition, using advanced models, which are most often trained for other purposes

in the model, can slow down the algorithm's performance during use and training. Among others, one such algorithm, Wav2vec, was used in [11].

Except for [11], the algorithms presented do not have built-in mechanisms that improve performance with access to more data during use. Working on algorithms of this type can increase the quality of the generated samples.

4. Conclusions

In summary, voice conversion is a promising, fast-growing research area that has the potential to improve the performance of voice conversion applications by better representing human speech. While some challenges still need to be overcome, the presented models have shown great potential in achieving high conversion efficiency levels and naturalness in converted voices. These methods have potential applications in various fields, such as entertainment, medicine, teaching, and the military industry. The integration of voice conversion technology into these fields can bring significant benefits, such as the creation of new voices for virtual characters in video games and animated movies and the ability to synthesize speech for people with speech impairments. Voice conversion is an exciting area of research that will continue to evolve and improve. The description of the algorithms in the second part makes it possible to look at the problem of voice conversion from the authors' perspectives of various algorithms. The juxtaposition of their performance results and the problems they face allows for a global view of the current state of the art. The presented article certainly provides a better understanding of the current state of the art in voice conversion using deep learning techniques, and thus, be a good starting point for developing other or similar techniques.

5. Future Directions

The analysis of the presented methods revealed several solutions that are superior to others in specific ways. This report highlights some favorable sub-solutions and suggests selecting the most suitable ones to develop a technique that achieves a high conversion efficiency. Future work includes plans for implementing such a method. Moreover, the information in this report about the datasets used and the evaluation methods will establish an appropriate benchmark for testing the effectiveness of future algorithms.

Although the problem of voice conversion has been faced by algorithm developers for a long time, some issues still require research. Here are some of them:

- The complex individual nature of human speech voice conversion is a task of great complexity, as it requires an understanding of various aspects of sound, such as tone, timbre, intonation, and tempo.
- Real-time performance requirements—in some cases, voice conversion must be done in real time, meaning that the algorithm must run fast enough for the user to hear the result in real time.
- Satisfactory results—the resulting quality of voice conversion can be crucial, especially for commercial applications. Algorithm developers face the challenge of ensuring that the results are good enough to be helpful to users.
- The flexibility of operation—the algorithm's performance should adapt to our data. The final product should also adapt in the case of higher-quality data. Furthermore, if users have different lengths of statements, the algorithm should work smoothly.
- Developing appropriate metrics for evaluating performance—in order to put voice conversion algorithms into practice, it is necessary to determine how well they work. Therefore, algorithm developers must create appropriate quality assessment metrics that consider various aspects of voice conversion, such as speech fluency, naturalness, and intelligibility.

Author Contributions: Investigation, T.W.; methodology, T.W. and Z.P.; resources, T.W.; supervision, Z.P.; validation, Z.P.; 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. Childers, D.G.; Wu, K.; Hicks, D.M.; Yegnanarayana, B. Voice conversion. *Speech Commun.* **1989**, *8*, 147–158. [CrossRef]
2. Mohammadi, S.H.; Kain, A. An Overview of Voice Conversion Systems. *Speech Commun.* **2017**, *88*, 65–82. [CrossRef]
3. Sisman, B.; Yamagishi, J.; King, S.; Li, H. An Overview of Voice Conversion and its Challenges: From Statistical Modeling to Deep Learning. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *29*, 132–157. [CrossRef]
4. Variani, E.; Lei, X.; McDermott, E.; Moreno, I.L.; Gonzalez-Dominguez, J. Deep Neural Networks for Small Footprint Text-dependent Speaker Verification. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014.
5. Wu, Z.; Li, H. Voice conversion versus speaker verification: An overview. *APSIPA Trans. Signal Inf. Process.* **2014**, *3*, e17. [CrossRef]
6. Chorowski, J.; Weiss, R.J.; Bengio, S.; Oord, A. van den Unsupervised speech representation learning using WaveNet autoencoders. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 2041–2053. [CrossRef]
7. Kashkin, A.; Karpukhin, I.; Shishkin, S. HiFi-VC: High Quality ASR-Based Voice Conversion. *arXiv* **2022**, arXiv:2203.16937.
8. Chen, M.; Zhou, Y.; Huang, H.; Hain, T. Efficient Non-Autoregressive GAN Voice Conversion using VQWav2vec Features and Dynamic Convolution. *arXiv* **2022**, arXiv:2203.17172.
9. Baevski, A.; Schneider, S.; Auli, M. vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations. *arXiv* **2019**, arXiv:1910.05453.
10. Wu, F.; Fan, A.; Baevski, A.; Dauphin, Y.N.; Auli, M. Pay Less Attention with Lightweight and Dynamic Convolutions. *arXiv* **2019**, arXiv:1901.10430.
11. Siuzdak, H.; Dura, P.; van Rijn, P.; Jacoby, N. WavThruVec: Latent speech representation as intermediate features for neural speech synthesis. *arXiv* **2022**, arXiv:arXiv:2203.16930.
12. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:1312.6114.
13. Long, Z.; Zheng, Y.; Yu, M.; Xin, J. Enhancing Zero-Shot Many to Many Voice Conversion with Self-Attention VAE. *arXiv* **2022**, arXiv:2203.16037.
14. Lian, J.; Zhang, C.; Yu, D. Robust Disentangled Variational Speech Representation Learning for Zero-shot Voice Conversion. *arXiv* **2022**, arXiv:2203.16705.
15. Li, Y.A.; Zare, A.; Mesgarani, N. StarGANv2-VC: A Diverse, Unsupervised, Non-parallel Framework for Natural-Sounding Voice Conversion. *arXiv* **2021**, arXiv:2107.10394.
16. Li, J.; Tu, W.; Xiao, L. FreeVC: Towards High-Quality Text-Free One-Shot Voice Conversion. *arXiv* **2022**, arXiv:2210.15418.
17. Kim, J.; Kong, J.; Son, J. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. *Proc. Mach. Learn. Res.* **2021**, *139*, 5530–5540.
18. Wan, L.; Wang, Q.; Papir, A.; Moreno, I.L. Generalized End-to-End Loss for Speaker Verification. *arXiv* **2017**, arXiv:1710.10467.
19. Jia, Y.; Zhang, Y.; Weiss, R.J.; Wang, Q.; Shen, J.; Ren, F.; Chen, Z.; Nguyen, P.; Pang, R.; Moreno, I.L.; et al. Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. *arXiv* **2018**, arXiv:1806.04558.
20. Qian, K.; Zhang, Y.; Chang, S.; Yang, X.; Hasegawa-Johnson, M. AUTOVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss. *arXiv* **2019**, arXiv:1905.05879.
21. Vaswani, A.; Brain, G.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
22. Desplanques, B.; Thienpondt, J.; Demuynck, K. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. *arXiv* **2020**, arXiv:2005.07143.
23. Yang, B.; Lyu, J.; Zhang, S.; Qi, Y.; Xin, J. Channel pruning for deep neural networks via a relaxed groupwise splitting method. In Proceedings of the 2019 2nd International Conference on Artificial Intelligence for Industries, AI4I 2019, Lagana Hills, CA, USA, 25–27 September 2019; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2019; pp. 97–98. [CrossRef]
24. Chou, J.; Yeh, C.; Lee, H. One-shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization. *arXiv* **2019**, arXiv:1904.05742.
25. Liu, S.; Cao, Y.; Wang, D.; Wu, X.; Liu, X.; Meng, H. Any-to-Many Voice Conversion with Location-Relative Sequence-to-Sequence Modeling. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 1717–1728. [CrossRef]
26. Polyak, A.; Wolf, L.; Taigman, Y. TTS Skins: Speaker Conversion via ASR. *arXiv* **2019**, arXiv:1904.08983.

27. Gulati, A.; Qin, J.; Chiu, C.-C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. Conformer: Convolution-augmented Transformer for Speech Recognition. *arXiv* **2019**, arXiv:2005.08100.
28. Schneider, S.; Baevski, A.; Collobert, R.; Auli, M. wav2vec: Unsupervised Pre-training for Speech Recognition. *arXiv* **2019**, arXiv:1904.05862.
29. Kum, S.; Nam, J. Joint detection and classification of singing voice melody using convolutional recurrent neural networks. *Appl. Sci.* **2019**, *9*, 1324. [CrossRef]
30. Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 1505–1518. [CrossRef]
31. Dinh, L.; Sohl-Dickstein, J.; Bengio, S. Density estimation using Real NVP. *arXiv* **2016**, arXiv:1605.08803.
32. Huang, X.; Belongie, S. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. *arXiv* **2017**, arXiv:1703.06868.
33. Yamamoto, R.; Song, E.; Kim, J.-M. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. *arXiv* **2019**, arXiv:1910.11480.
34. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. *arXiv* **2019**, arXiv:1912.04958.
35. Kong, J.; Kim, J.; Bae, J. HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17022–17033.
36. van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. *arXiv* **2016**, arXiv:1609.03499.
37. Nguyen, B.; Cardinaux, F. NVC-Net: End-to-End Adversarial Voice Conversion. *arXiv* **2021**, arXiv:2106.00992.
38. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A.; Deepmind, G. β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In Proceedings of the 2017 International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
39. Shen, J.; Pang, R.; Weiss, R.J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerry-Ryan, R.; et al. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018.
40. Li, Y.; Mandt, S. Disentangled Sequential Autoencoder. *arXiv* **2018**, arXiv:1803.02991.
41. van den Oord, A.; Kalchbrenner, N.; Kavukcuoglu, K. Pixel Recurrent Neural Networks. *arXiv* **2016**, arXiv:1601.06759.
42. Holschneider, M.; Kronland-Martinet, R.; Morlet, J.; Tchamitchian, P. A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform. In *Wavelets: Time-Frequency Methods and Phase Space, Proceedings of the International Conference, Marseille, France, 14–18 December 1987*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 286–297. [CrossRef]
43. Dutilleux, P. An implementation of the "algorithme a trous" to compute the wavelet transform. In *Wavelets: Time-Frequency Methods and Phase Space, Proceedings of the International Conference, Marseille, France, 14–18 December 1987*; Springer: Berlin/Heidelberg, Germany, 1989.
44. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *arXiv* **2014**, arXiv:1412.7062.
45. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2015**, arXiv:1511.07122.
46. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
48. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65.
49. Yamamoto, R.; Song, E.; Kim, J.M. Probability density distillation with generative adversarial networks for high-quality parallel waveform generation. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Graz, Austria, 15–19 September 2019; International Speech Communication Association: Baixas, France, 2019; pp. 699–703. [CrossRef]
50. Arik, S.O.; Jun, H.; Diamos, G. Fast Spectrogram Inversion using Multi-head Convolutional Neural Networks. *IEEE Signal Process. Lett.* **2018**, *26*, 94–98. [CrossRef]
51. Wang, X.; Takaki, S.; Yamagishi, J. Neural source-filter-based waveform model for statistical parametric speech synthesis. *arXiv* **2018**, arXiv:1810.11946.
52. Wang, C.; Zeng, C.; He, X. HiFi-WaveGAN: Generative Adversarial Network with Auxiliary Spectrogram-Phase Loss for High-Fidelity Singing Voice Generation. *arXiv* **2022**, arXiv:2210.12740.
53. Donahue, C.; McAuley, J.; Puckette, M. Adversarial Audio Synthesis. *arXiv* **2018**, arXiv:1802.04208.
54. Kumar, K.; Kumar, R.; de Boissiere, T.; Gestin, L.; Teoh, W.Z.; Sotelo, J.; de Brebisson, A.; Bengio, Y.; Courville, A. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. *arXiv* **2019**, arXiv:1910.06711.
55. Liu, Z.; Mak, B. Cross-lingual Multi-speaker Text-to-speech Synthesis for Voice Cloning without Using Parallel Corpus for Unseen Speakers. *arXiv* **2019**, arXiv:1911.11601.
56. Bakhturina, E.; Lavrukhin, V.; Ginsburg, B.; Zhang, Y. Hi-Fi Multi-Speaker English TTS Dataset. *arXiv* **2021**, arXiv:2104.01497.

57. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing, Brisbane, QLD, Australia, 19–24 April 2015; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2015; pp. 5206–5210. [CrossRef]
58. Ardila, R.; Branson, M.; Davis, K.; Henretty, M.; Kohler, M.; Meyer, J.; Morais, R.; Saunders, L.; Tyers, F.M.; Weber, G. Common Voice: A Massively-Multilingual Speech Corpus. *arXiv* **2019**, arXiv:1912.06670.
59. Ephrat, A.; Mosseri, I.; Lang, O.; Dekel, T.; Wilson, K.; Hassidim, A.; Freeman, W.T.; Rubinstein, M. Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation. *arXiv* **2018**, arXiv:1804.03619. [CrossRef]
60. Garofolo, J.S. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. *Linguist. Data Consort.* **1993**. [CrossRef]
61. Zhao, Y.; Huang, W.-C.; Tian, X.; Yamagishi, J.; Das, R.K.; Kinnunen, T.; Ling, Z.; Toda, T. Voice Conversion Challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion. *arXiv* **2020**, arXiv:2008.12527.
62. Takamichi, S.; Mitsui, K.; Saito, Y.; Koriyama, T.; Tanji, N.; Saruwatari, H. JVS corpus: Free Japanese multi-speaker voice corpus. *arXiv* **2019**, arXiv:1908.06248.
63. Zhou, K.; Sisman, B.; Liu, R.; Li, H. Seen and Unseen emotional style transfer for voice conversion with a new emotional speech dataset. *arXiv* **2020**, arXiv:2010.14794.
64. Lo, C.-C.; Fu, S.-W.; Huang, W.-C.; Wang, X.; Yamagishi, J.; Tsao, Y.; Wang, H.-M. MOSNet: Deep Learning based Objective Assessment for Voice Conversion. *arXiv* **2019**, arXiv:1904.08352.
65. Lian, J.; Zhang, C.; Anumanchipalli, G.K.; Yu, D. Towards Improved Zero-shot Voice Conversion with Conditional DSVAE. *arXiv* **2022**, arXiv:2205.05227.
66. Zen, H.; Dang, V.; Clark, R.; Zhang, Y.; Weiss, R.J.; Jia, Y.; Chen, Z.; Wu, Y. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. *arXiv* **2019**, arXiv:1904.02882.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.