Audio Style Conversion Based on AutoML and Big Data Analysis

Dan Chi
School of Liberal Arts Education and Art Media, Xiamen Institute of Technology, Xiamen, 361000, China

***Abstract*—In the field of audio style conversion research, the application of AutoML and big data analysis has shown great potential. The study used AutoML and big data analysis methods to conduct deep learning on audio styles, especially in style transitions between flutes and violins. The results show that using iterative learning for audio style conversion training, the training curve tends to stabilize after 100 iterations, while the validation curve reaches stability after 175 iterations. In terms of efficiency analysis, the efficiency of the yellow curve and the green curve reached 1.05 and 1.34, respectively, with the latter being significantly more efficient. This study achieved significant results in audio style conversion through the application of AutoML and big data analysis, successfully improving conversion accuracy. This progress has practical application value in multiple fields, including music production and sound effect design.**

Keywords*—*AutoML; audio style conversion; machine learning; big data analysis; adain module

I. Introduction

Audio style conversion, as an important branch in the field of audio processing, has always been the focus of researchers. The transformation of audio style aims to make subtle adjustments to the characteristics of audio without loss, such as time domain, frequency domain, timbre, pitch, etc., while retaining the essential information of audio [1-2]. The implementation of this transformation has a profound impact on many fields such as music production, speech synthesis, and oral teaching [3-4]. However, traditional audio style conversion methods often require a large amount of manual feature extraction and complex algorithm design. This limits the research process of audio style conversion. At present, to solve the above problems, most scholars advocate the introduction of data analysis to achieve various audio processing. But in this way, automatic search and optimization of audio conversion models and parameters can be achieved [5-6]. Meanwhile, this study also extracts valuable style information from massive audio data through big data analysis, further improving the accuracy and naturalness of audio style conversion. Integrating AutoML into audio style conversion research directly addresses the inefficiencies of current methodologies. This provides a systematic approach to model selection and parameter tuning, which is critical for enhancing the practicality and accessibility of audio style transformations. The innovation of research is mainly manifested in two aspects. AutoML is introduced into the research of audio style conversion to achieve automated search and optimization of audio conversion models, with the aim of improving the efficiency of audio style conversion. The second is to use big data analysis methods to extract style information from massive audio data, making audio style conversion more accurate and natural. The research contributions consist of developing an AutoML-based framework for optimizing audio style conversion models efficiently, introducing big data analytics for extracting precise style features, establishing a benchmark dataset for comparative analysis that demonstrates enhanced conversion accuracy and naturalness, and validating real-world applications across music production, speech synthesis, and language learning. This study also provides new research methods and ideas for other related fields, with broad application prospects and important academic value. The research will be conducted in four parts. The first part is an overview of audio style conversion on the grounds of AutoML and big data analysis. The second part is the research on audio style conversion on the grounds of AutoML and big data analysis. The third is the experimental verification of the second. The fourth is a summary of the research content and points out the demerits.

II. Related Works

Audio style conversion has always been an essential research topic in the audio processing, with the goal of achieving lossless conversion of audio styles while maintaining the original audio information. Li J et al. presented a novel ALRW method. The research results indicate that this method could markedly decrease compensation information. And it exhibits strong robustness to common operations. In the absence of an attack, it is possible to recover the covered audio signal without loss [7]. Lin F et al. presented a new text audio sentiment analysis framework called StyleBERT, which enhances unimodal sentiment information representation by learning different modal styles and reduces dependence on fusion. The research results indicate that StyleBERT performs excellently on multiple benchmark datasets, markedly superior to state-of-the-art multimodal baselines, and is an effective multimodal sentiment analysis framework [8]. Chen B and other scholars proposed a non-parallel data to speech conversion technology on the grounds of data augmentation - ParaGen. The experiment showcases that ParaGen can effectively convert the speaker identity of the source speech to the target speaker while preserving the local speaking style. And the converted speech possesses more excellent speech naturalness and speaker similarity than the statistical parameter speech synthesis system [9]. Xu D et al. proposed a bipolar phase shift modulation single-stage inverter for efficient and low distortion audio amplification. The research results were validated through a prototype with an output power of 200kHz and 250W. It demonstrates the effectiveness of the proposed

BPSM • FBAC-SSI method in improving the efficiency of audio amplifiers and reducing distortion [10]. Chandrakar R et al. proposed an enhanced system for automatic motion object detection and tracking using RBF-FDLNN and CFR algorithms. It can effectively handle the problem of motion target detection and tracking in traffic monitoring. The research results indicate that the proposed RBF-FDLNN classifier performs better than other existing methods in video frame object detection, proving the effectiveness of this method [11].

However, traditional audio style conversion methods rely on complex algorithm design and a large amount of manual feature extraction. This to some extent limits the development of audio style conversion technology. Zhang J presented a music feature extraction and classification model on the grounds of convolutional neural networks. The research results indicate that this method outperforms traditional manual models and machine learning based methods in music feature extraction and classification. This effectively addresses the shortcomings of traditional methods in feature selection and multi classification [12]. Singh P K et al. proposed new feature descriptor-binary image symbolization, for recognizing handwritten digits of different texts. The research results indicate that the symbolic feature descriptors of binary images have high script invariance, and can maintain high recognition rates even in mixed use of text [13]. Jiang ZG et al. proposed a segmentation and keyframe extraction method for video behavior recognition, and further proposed an improved vehicle detection algorithm on the grounds of fast R-CNN. The research results indicate that the application of keyframe extraction technology and optimized fast R-CNN model significantly improves the accuracy of vehicle detection, reduces missed detections, and demonstrates satisfactory detection rates [14]. Jia Z et al. proposed domain invariant feature extraction and fusion. The research results indicate that domain invariant feature extraction and fusion methods have achieved significant performance improvements on multiple datasets, effectively addressing the challenge of cross domain character re recognition [15]. Grzegorowski et al. proposed a supply management solution that considers individual delivery plans for each location. The research outcomes demonstrate that the method could markedly handle high uncertainty in data and effectively solve the cold start problem of vending machine networks [16].

Wu SL et al. utilized the advantages of Transformer and VAE to propose MuseMorphose for music generation, which is characterized by the user's ability to control style attributes. The results showed that MuseMorphose exceeded the RNN baseline in style transfer metrics [17]. Rashid A B et al. proposed an automatic detection model for student learning style in a learning management system based on online learning activities. The research shows that this model can assist educators in optimizing teaching content and recommending suitable learning materials based on student characteristics [18]. Chen et al. proposed reinforcement learning based audiovisual speech recognition framework MSRL, which focuses on stable supplementary information of visual modalities. The research results show that MSRL achieves the best performance on the LRS3 dataset, especially demonstrating better universality in unknown noise testing [19].

In summary, existing research results indicate that AutoML can achieve automatic recognition and conversion of audio styles, which helps to solve the efficiency and accuracy problems of traditional methods in large-scale data processing. However, the complexity and diversity of audio data processing, such as feature extraction and model selection, remain challenges that limit the comprehensive application of AutoML. These technologies also need to be further optimized in practical scenarios such as music creation and speech synthesis. In view of this, this study aims to develop stronger audio feature extraction algorithms and establish effective model evaluation methods. And it is necessary to study how to better integrate these technologies into practical applications to maximize the potential of AutoML in audio processing. AutoML's advancement in audio style conversion heralds new creative horizons in music production, elevates speech synthesis realism, and promises tailored, immersive language learning experiences that are revolutionizing multiple industries.

III. AUDIO STYLE CONVERSION METHODS ON THE GROUNDS OF AUTOML AND BIG DATA ANALYSIS

This study combines an improved VGG and EfficientNet feature extraction network to deeply extract audio data features. It utilizes Adain based normalization modules and feature decoding networks to achieve lossless audio style conversion. It combines AutoML and big data analysis to construct an automatically optimized audio style conversion model to improve conversion efficiency and accuracy. This study integrates the latest machine learning techniques into audio processing, providing a new research perspective for the development of audio style conversion technology.

A. *Based on Improved VGG and EfficientNet Feature Extraction Network*

Audio style conversion relies on deep learning to automatically extract features, obtaining abstract and robust features. The VGG network has fewer parameters and requires less computing resources. The new EfficientNet breaks the convention of improving network performance in a single dimension by adjusting input resolution, depth, and width, achieving a balance between accuracy and efficiency [20-21]. The VGG-16 network uses small convolutional kernels instead of large ones to enhance model nonlinearity, reduce computational complexity, and remove fully connected layers (FCL). Then it changes the pooling layer to a convolutional layer with a stride of 2, and uses a swish activation function to improve model performance. EfficientNet extracts useful features in audio style conversion, compares feature representations, and predicts the effect of style conversion. The optimization process relies on the loss function, and the smaller the loss function, the higher the accuracy of the model. Therefore, EfficientNet and VGG networks have important application value in the study of audio style conversion [22-23]. The mean square error is shown in Eq. (1).

$$MSE = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - f\left(x_i\right)\right)^2 \tag{1}$$

In Eq. (1), y_i serves as the actual value, $f(x_i)$ serves as the predicted value, and $f(x_i)$ serves as the number of samples. The backpropagation of gradient information is crucial for neural network algorithms to self-learn and update. The optimized EfficientNet algorithm is showcased in Eq. (2).

$$\theta_k = \theta_{k-1} - \alpha \cdot g \tag{2}$$

In Eq. (2), θ_k is the parameter value at the current time, α is the learning rate, and g is the gradient. It increases the number of audio processing channels and adds feature layers to obtain more audio features. It increases network depth, utilizes deep neural networks to improve performance, and enhances audio feature extraction. It improves the input audio sampling rate, enhances network accuracy, enriches audio features, and reduces information loss. From this, it can be concluded that the tensor of the network output is shown in Eq. (3).

$$Y_i = F_i(X_i) \tag{3}$$

In Eq. (3), X_i is the tensor of a specific convolutional layer. A deep neural network composed of k convolutional layers is shown in (4).

$$N = F_k \square \ldots \square F_2 \square F_1(X_1) = \underset{j=1\ldots k}{\square} F_j(X_1) \tag{4}$$

In Eq. (4), $\square$ is the multiplication operation, i is the stage number, and i is a single operation. Scaling the model could enhance the accuracy of the network within the limits of memory and computational complexity, as shown in Eq. (5).

$$\begin{cases} \max\limits_{d,w,r} Accuracy\left(N(d,w,r)\right) \\ s,t, N(d,w,r) = \underset{i=1\ldots S}{\square} F_i^{d,\hat{L}}\left(X_{r\cdot\hat{H}_i^r\hat{W}_I^R\hat{C}_i}\right) \end{cases} \tag{5}$$

In Eq. (5), d, w, and r represent the scaled depth, width, and resolution, respectively. In EfficientNet, achieving composite optimization by simultaneously scaling three dimensions at appropriate proportions could enhance the performance and classification accuracy of the network. This could also decrease the computational complexity of the model and enhance the performance of the network. The MBConv module used internally is the core structure, which is a unique feature extraction structure of EfficientNet. The two-dimensional view is efficiently extracted during the continuous stacking process in the Block layer. The MBConv module is shown in Fig. 1.

In Fig. 1, EfficientNet first performs pointwise convolution on the input feature map and adjusts the expansion ratio by changing the output channel dimension. Then it performs deep convolution, reducing the dimension to the original number of channels, and then performs point by point convolution again. This network module integrates compression and arousal of network attention to focus on channel features. The feature map is processed by stacking 32 MBConvs, and then sequentially passes through one-dimensional convolutional layers, global average pooling 2D, and FCL to generate feature vectors with a dimension of 2640. EfficientNet reduces the computational complexity of the network through deep convolution and point by point convolution, compared to conventional convolution operations. The schematic diagram of EfficientNet network structure is showcased in Fig. 2.

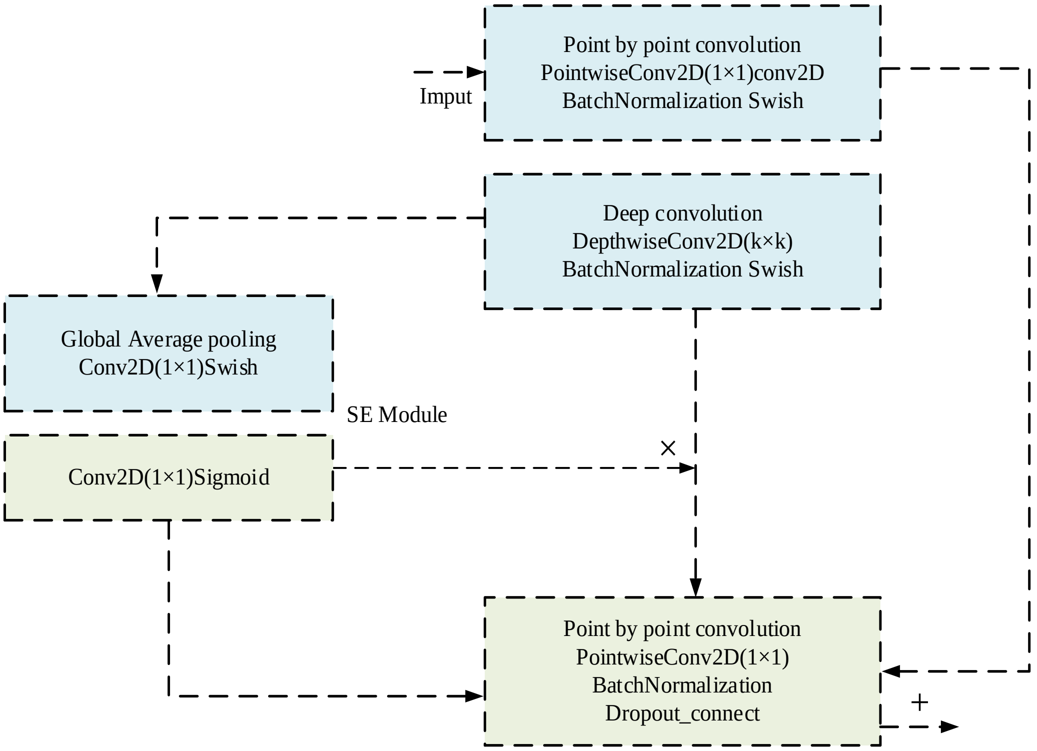


Fig. 1. MBConv module.

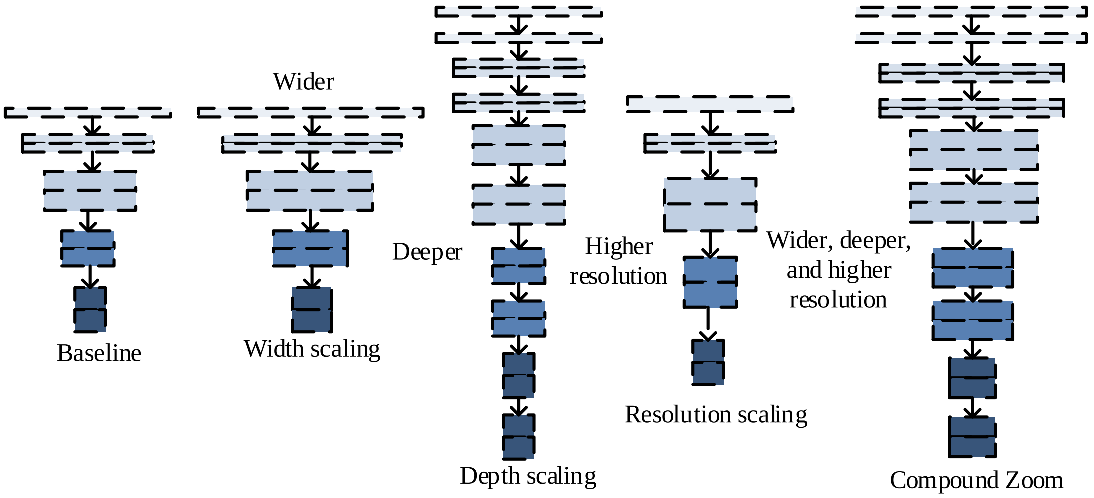


Fig. 2. EfficientNet network structure diagram.

In Fig. 2, EfficientNet utilizes MBConv as the backbone network, which originates from MobileNet V2. MBConv includes a regular convolution, a deep convolution (including BN and Swish), an SE module, another regular convolution (for dimensionality reduction, including BN), and a Droupout layer. The SE module contains a global average pooling and two FCL. The quantity of nodes in the first FCL is equal to the quantity of channels in the feature matrix of the input MBConv, and the activation function is Swish. The quantity of nodes in the second FCL is equal to the number of channels in the output feature matrix of the deep convolutional layer, and the activation function is sigmoid.

B. Audio Style Conversion Normalization Module and Feature Decoding Network on the Grounds of Adain

After completing the feature extraction for audio style conversion, the next step is to use the Adam based normalization module and feature decoding network for audio style conversion. In this process, Adain technology is used to convert audio features into styles, and then a feature decoding network is used to convert the converted features into perceptible audio signals. This can achieve style conversion of audio.

The normalization process can make the data distribution have a mean of 0 and a variance of 1, which helps to avoid gradient vanishing and exploding, thus accelerating the training process. When processing large amounts of data, BatchNorm needs to use mini batch data to estimate mean and variance, but training may become unstable when computing power is limited and the input audio data volume is too large. The Adain method confirms that the Instance Normalization layer can reduce style loss faster than the BN layer, thereby accelerating training. The core of Adain is to fuse the features obtained from content audio and style audio through an Encoder network, and then decode them to obtain style audio. The decoding of style audio is shown in Eq. (6).

$$AdaIN(x,y)=\sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right)+\mu(y) \tag{6}$$

In Eq. (6), the content image is x and the style image is y. The current mainstream normalization methods mainly include Batch Normalization, Layer Normalization, Instance Normalization, Group Normalization, and Switchable Normalization. These methods are all on the grounds of normalization processing of different dimensions of input audio. Specifically, given the dimensions of the input audio as (N, C, H, W), different normalization methods choose different normalization strategies on these four dimensions. An example of centralized normalization is shown in Fig. 3.

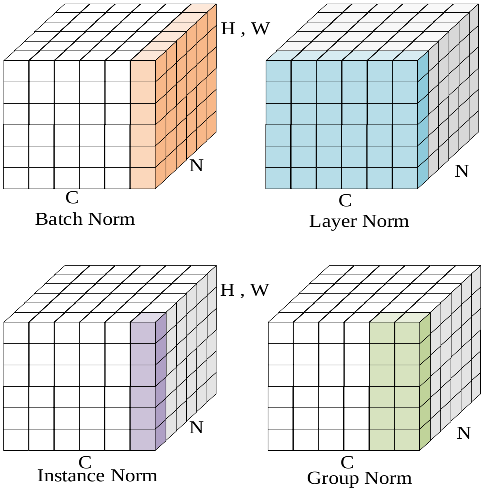


Fig. 3. An example of centralized normalization diagram.

Batch Normalization is the normalization of NHW on each batch. Due to its calculation of mean and variance on each batch, if the batch size is too small, the calculated mean and variance may not represent the distribution of the entire data, which may lead to unstable training and poor performance. Layer Normalization is the normalization of CHW for each channel direction, mainly applied in RNN networks. Compared to Batch Normalization, Layer Normalization solves the problem of deep non fixed networks by normalizing all neurons in each layer of the deep network, as shown in Eq. (7).

$$\begin{cases} \mu^l = \frac{1}{H}\sum_{i=1}^{H} a_1^l \\ \sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_i^l - \mu^l\right)^2} \end{cases} \quad (7)$$

In Eq. (7), μ is the mean and σ^l is the variance. Instance Normalization is mainly applied in the field of audio style conversion, which normalizes audio signals at the pixel level. Due to the fact that the generated results mainly rely on specific audio samples, it is very suitable for audio stylization, which can accelerate model convergence and maintain independence between samples. Group normalization is achieved by grouping channels and normalizing them within the group, and its calculated mean is independent of batch size. Therefore, it can solve the impact of batch normalization on training results when the batch is small. In the feature map of each sample, channels are divided into G groups, each containing C/G channels, and the mean and standard deviation of these channels are calculated. Switchable Normalization combines BN, LN, and IN normalization methods, assigning them different weights to enable the network to learn which normalization method to use on its own, thus achieving adaptive selection of normalization methods. The style transition effect is showcased in Fig. 4.

The decoder and encoder have a symmetrical structure. During the audio style conversion, the encoder is responsible for feature extraction of the original audio and the target style audio. The decoder combines the original audio features and style features generated by Adain to generate stylized audio. In audio style conversion systems, only the decoder is usually trained, while the parameters of the feature extraction and loss calculation networks remain unchanged. During the feature extraction, downsampling is usually performed through a series of convolutions. In the feature decoding stage, it is necessary to upsample the features to restore the size of the original audio. Common upsampling methods include linear interpolation, deconvolution, and deconvolution. Deconvolution is a special type of convolution that fills feature audio with zeros and then convolves it by rotating the convolution kernel. In decoding networks for audio style conversion, interpolation algorithms are commonly used for upsampling operations. Deconvolution restores feature audio, which operates in the opposite direction of convolution and is essentially transposed convolution. The relevant schematic diagram is showcased in Fig. 5.

C. Audio Style Conversion Model on the Grounds of AutoML and Big Data Analysis

Considering the characteristics of audio data and the complexity of processing audio data, this study selected an audio style conversion model on the grounds of AutoML and big data analysis for research. The model utilizes big data analysis technology to process massive audio data. And it automatically finds the optimal model parameters through AutoML to achieve more efficient and accurate audio style conversion. The framework structure of the model is set as showcased in Fig. 6.

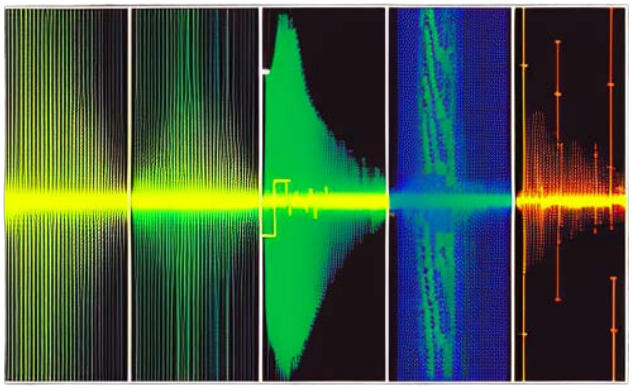


Fig. 4. Style transition effect diagram.

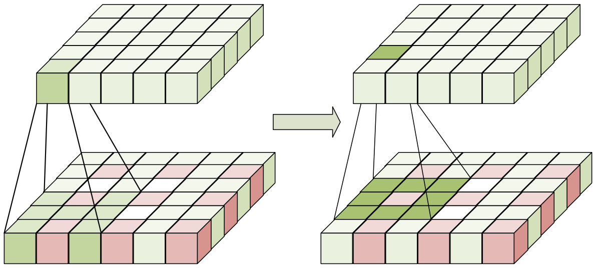


Fig. 5. The relevant schematic diagram.

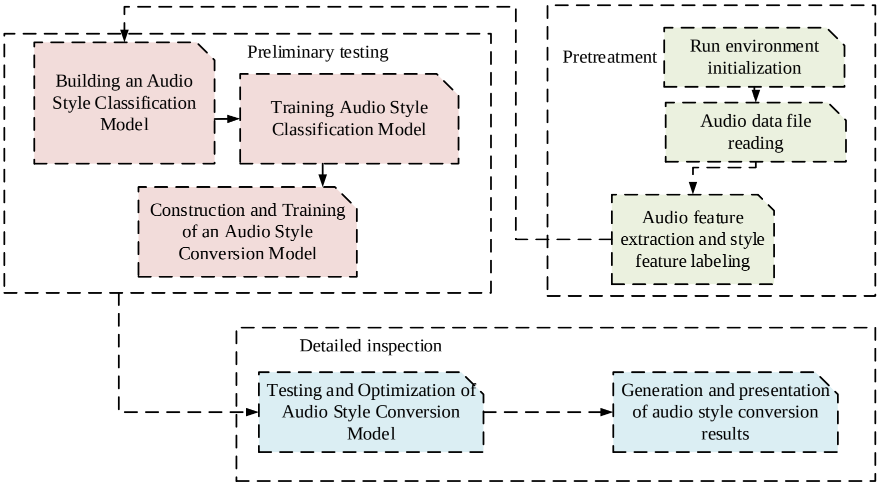

Preliminary testing
Building an Audio Style Classification Model
Training Audio Style Classification Model
Construction and Training of an Audio Style Conversion Model
Pretreatment
Run environment initialization
Audio data file reading
Audio feature extraction and style feature labeling
Detailed inspection
Testing and Optimization of Audio Style Conversion Model
Generation and presentation of audio style conversion results

Fig. 6. Functional module execution process.

In Fig. 6, considering the characteristics of audio style conversion, the audio style conversion process on the grounds of AutoML and big data analysis can be mainly divided into four stages: preprocessing stage, feature extraction stage, model training stage, and style application stage. The preprocessing stage is mainly used for cleaning and formatting audio data for subsequent processing. The feature extraction stage converts audio data into feature vectors that can be processed by machine learning models. During the model training phase, AutoML is used to automatically search for the optimal model parameters, to achieve more efficient and accurate audio style conversion. The final style application stage applies the trained model to new audio data to complete style conversion. The evaluation indicators for audio style conversion are shown in Eq. (8).

$$D_i = \frac{m_i}{N_i} \tag{8}$$

In Eq. (8), m_i represents the degree of style distortion of the audio sample, and N_i represents the total number of style transitions performed. It enhances the accuracy of the model by connecting all the encoded features obtained, and the vector construction is shown in Eq. (9).

$$\begin{cases} G(q_t, a_t) = q + (\max(q) + 1) \cdot a_t \\ v_t = Q(C(q_t, a_t)) \cap Q(C(f_t, a_t)) \cap Q(dc_t) \end{cases} \tag{9}$$

In Eq. (9), q_t represents the feature number, and dc_t represents the difficulty coefficient of coherent instructions. $G(\ )$ represents instruction execution combination, $Q(\ )$ represents encoding format, and $\cap$ represents connection. It uses ReLU activation function to train the autoencoder and removes the output layer after completion. Then it uses the output of the hidden layer as input in the AutoML model, as shown in Eq. (10).

$$v_t' = FAKT(W \cdot v_t + b) \tag{10}$$

In Eq. (10), W, b represent the weight matrix and bias vector for audio execution, respectively. The current state is obtained by combining the information processed by the forget gate with the input information obtained by the input gate. The process is shown in Eq. (11).

$$\begin{cases} f_t = \sigma\left(W_f\left[v_t', h_{t-1}\right] + b_f\right) \\ c_t = f_t \square\ c_{t-1} + i_t \square\ \overset{\square}{c}_t \end{cases} \tag{11}$$

In Eq. (11), the output gate o_t determines what information to extract from c_t on the grounds of h_{t-1}, v_t', and the ReLU function, forming a hidden state h_t. Then, combined with historical encoding, the attention score is calculated using weight factors, as shown in Eq. (12).

$$s(v_i', v_t') = \overset{\square}{v}^T \tanh\left(W_1 v_i' + W_2 v_t'\right) \tag{12}$$

In Eq. (12), v_i' represents the encoding of historical sequence information on the grounds of data compression, and v_t' represents the dimensionality reduction encoding of input information at time t. $W, \overset{\square}{v}$ are both network parameters for the Department of Science. After weighting, cross entropy can be used in machine learning for measuring the difference between actual labels and predicted results. Therefore, the study uses it as the loss function of the AutoML model, as shown in Eq. (13).

$$L = -\sum_t \left(a_{t+1} \log y_t^T \delta(q_{t+1}) + (1 - a_{t+1}) \log\left(1 - y_t^T \delta(q_{t+1})\right)\right) \tag{13}$$

In Eq. (13), $y_t^T \delta(q_{t+1})$ and a_{t+1} represent the predicted and true probability distributions, respectively. It assumes that X is one knowledge unit, including n execution nodes. And if the probability of using the execution node x_i is $P(x_i), i = 1, 2, \ldots, n$, then the audio modeling's mastery of the execution node is shown in Eq. (14).

$$Z(X) = -\sum\nolimits_{i=1}^{n} P(x_i) \log_2 \frac{A(x_i)}{2} \tag{14}$$

In Eq. (14), $A(x_i)$ represents the execution efficiency of the audio at the execution node. $P(x_i)$ represents the probability of the execution node appearing.

IV. Modeling and Analysis of Audio Style Conversion on the GroundS of AutoML and Big Data Analysis

It conducts research on audio style transformation modeling on the grounds of AutoML and big data analysis, and extracts deep features from audio data. Then it utilizes EfficientNet and VGG networks to construct an audio style classification and transformation model. By comparing different audio features, it predicts the performance of audio after different style transitions. The experimental environment and dataset parameters are showcased in Table I.

The accuracy changes of the audio style conversion model on the conversion of training set audio and validation set audio under different iterations are shown in Fig. 7.

TABLE I. EXPERIMENTAL ENVIRONMENT AND DATASET PARAMETERS

Parameter	Description	Parameter	Description
Operating System	Ubuntu 20.04	CUDA Version	11.2
CuDNN Version	8.1.0	Programming Language	Python 3.8
Parameter	Description	Parameter	Description
Operating System	Ubuntu 20.04	CUDA Version	11.2
CuDNN Version	8.1.0	Programming Language	Python 3.8
Data Augmentation	Add noise, Time stretch	Audio Sample Rate	44100 Hz
Audio Resolution	16 bits	Style Audios Styles	Classical, Rock, Jazz, Pop
Total Samples in Training Data	1.2 million samples	Training Data Size after Processing	1024*1024
Processed Sample Count	16 samples per audio	Scales during Training	400400, 300300, 256*256
Style Audio Size	512*512	Big Data Analysis Tool	Apache Hadoop, Apache Spark

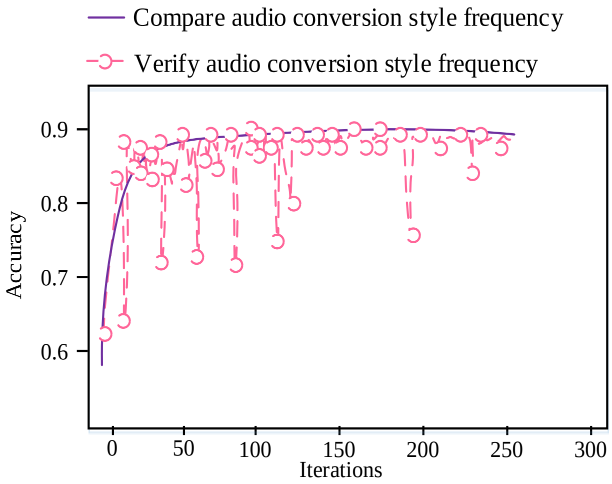


Fig. 7. Accuracy of training and validation sets in audio style transformation.

In Fig. 7, with the quantity of iterations grows, the accuracy of training audio style conversion gradually increases. This indicates that the audio style conversion model has a faster convergence ability for training audio and verifying audio. However, the accuracy of verifying audio style conversion fluctuates greatly. Sometimes the conversion accuracy of the verified audio is higher than that of the training audio, but lower than that of the training audio at other iterations. The fluctuation amplitude gradually decreases with the increase of iterations. After 50 iterations of training, the style conversion accuracy of both the training audio and validation audio exceeded 90%, and the effect was significant. The training curve tends to stabilize after 100 iterations, while the validation curve reaches stability after 175 iterations. These two curves indicate that the audio style conversion model achieved good training performance after 175 iterations, and reached its optimal performance after 200 iterations. For the audio samples of bird singing, vehicle horn sound, wave crashing sound, and piano performance, the style conversion ratio is compared with the original style conversion framework, as shown in Fig. 8.

In Fig. 8, the curves of style conversion ratios are higher than those of the original framework, demonstrating that the style conversion model could improve the conversion quality of audio. This is because the model fully takes into account the characteristics of audio style transition, that is, in the audio, certain parts of the style transition are more pronounced than others. It calculates the weighted value of each audio segment on a unit basis. Then it adaptively compensates for the weighted values of each segment, making the style transformation of each segment's weighted values more detailed. This is to achieve the goal of improving the quality of audio style conversion and enhancing audio performance. It compares the improvement in style conversion efficiency for three audio sample sets: FreeSound, Looperman, and SampleSwap, as shown in Fig. 9.

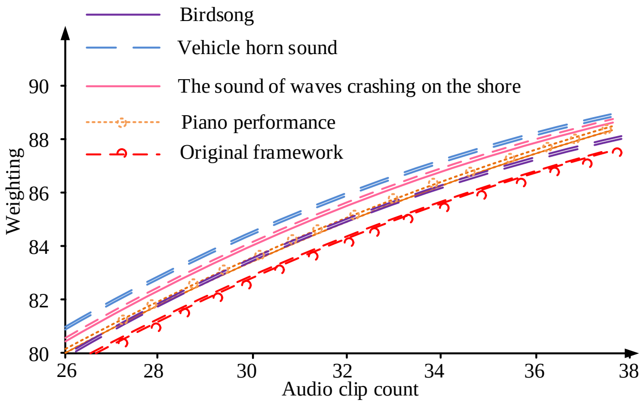
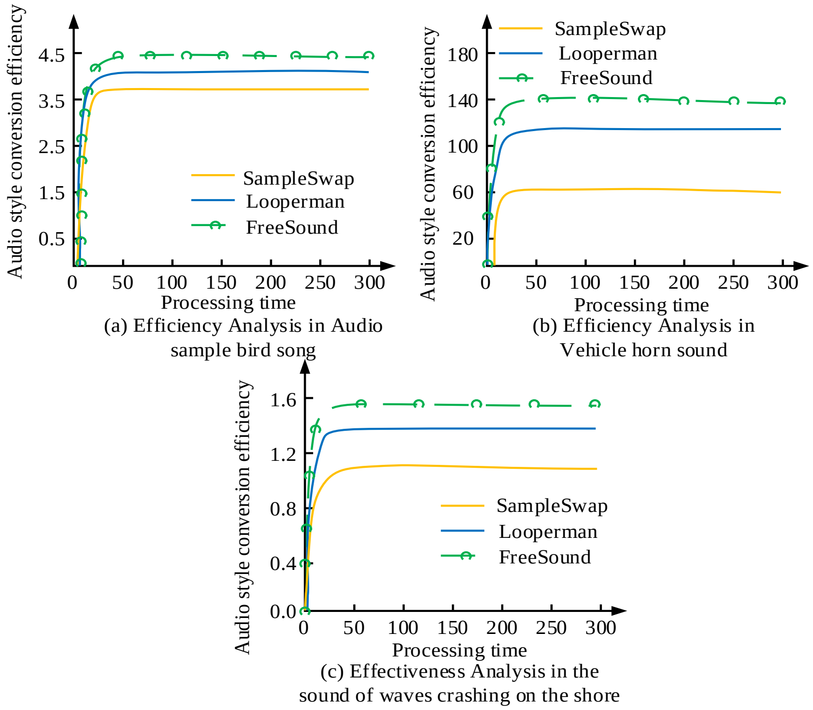


Fig. 8. Comparison of style transformation ratio with original framework.

In Fig. 9, the yellow curve represents the efficiency of SampleSwap style conversion. The blue dashed line represents the efficiency of Loopperman style conversion. The green curve represents the efficiency of FreeSound style conversion. In the efficiency analysis curve of Fig. 11 (a), the green curve has the best effect and is also the most stable, with time ranging from 0 to 300. The green curve is the fastest to reach 4.48 and has been operating at this efficiency. Compared to the yellow lines, the efficiency of the blue dashed lines is much lower. In the efficiency analysis curve of Fig. 11 (b), the yellow curve has the worst effect, only less than 60, while the green curve and blue dashed line are 140 and 116, respectively. In Fig. 11 (c), the efficiency of the yellow curve is around 1.05, while the efficiency of the green curve is still as high as 1.34. The spectrograms of the flute version of "Butterfly Lovers", the violin version of "My Heart Forever", and the flute version of "Titanic" output by the STFT model are shown in Fig. 10.

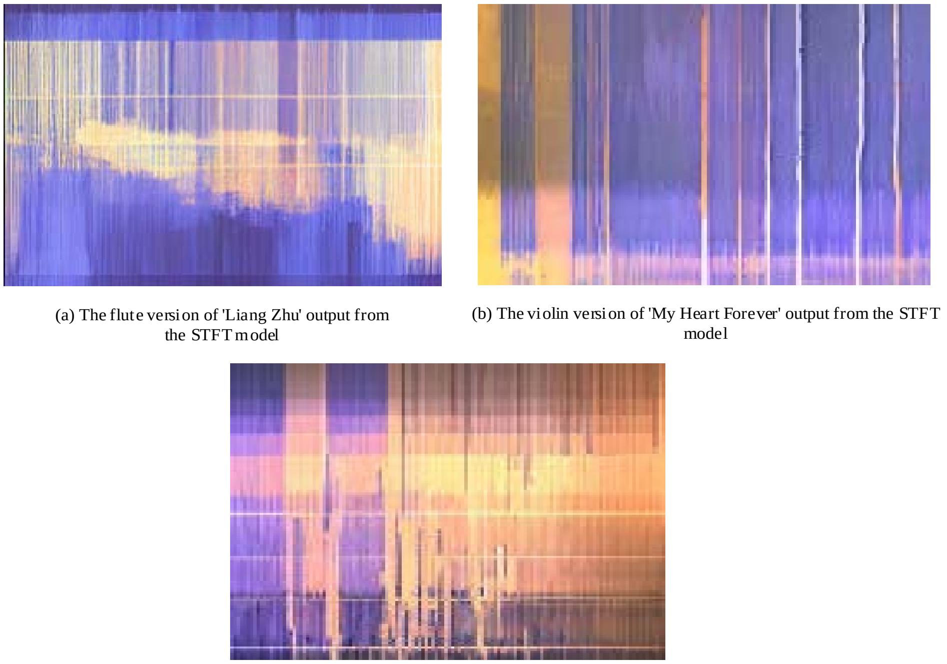


(a) Efficiency Analysis in Audio sample bird song

(b) Efficiency Analysis in Vehicle horn sound

(c) Effectiveness Analysis in the sound of waves crashing on the shore

Fig. 9. Efficiency analysis of audio style transformation improvement in three datasets.



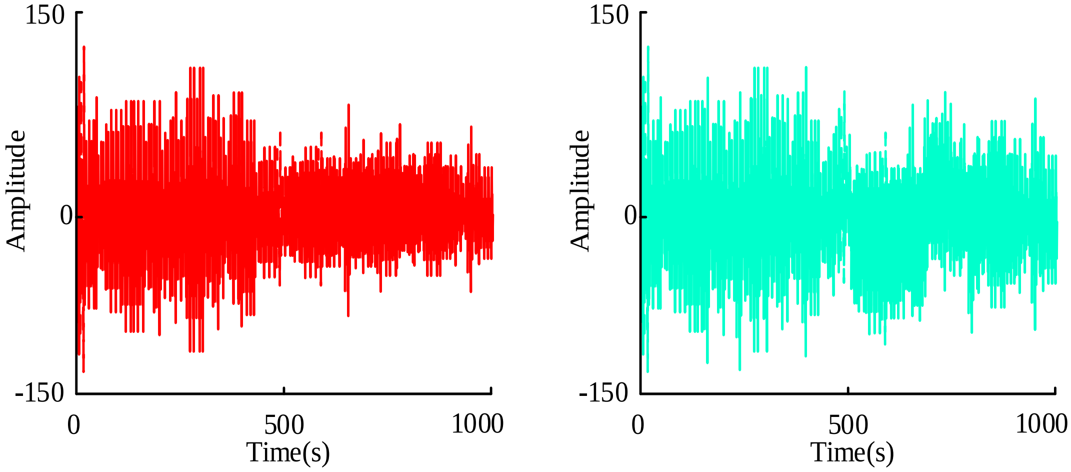
(a) The flute version of 'Liang Zhu' output from the STFT model

(b) The violin version of 'My Heart Forever' output from the STFT model

(c) The violin version of Titanic output from the STFT model

Fig. 10. The spectrograms of the flute version of "Butterfly Lovers", the violin version of "My Heart Forever", and the flute version of "Titanic" output from the STFT model.

In Fig. 10, whether it is the flute to violin or the violin to flute, the audio quality of the spectrograms output by the two models is not very good, resulting in poor sound quality. The audio obtained by the STFT model hardly shows any changes in timbre, and the sound is relatively noisy. The audio obtained by the CQT model can vaguely distinguish the timbre of the instrument. The time domain diagram is drawn on the grounds of the first eight seconds of Beethoven's First String Trio in e-flat major (hereinafter referred to as Beethoven. wav) and the first nine seconds of Telemann's Flute Fantasy. The time-domain diagram drawn by Beethoven. wav and Telemann in the first nine seconds are shown in Fig. 11.



(a) Beethoven. wav Time Domain DiagramStress response curve of nodes at the point of maximum stress

(b) Telemann time-domain diagram

Fig. 11. Time-domain plots plotted from the first 9s fragments of beethoven.wav and Telemann.

In Fig. 11, the horizontal axis serves as time and the vertical axis serves as amplitude, reflecting the temporal variation of the audio signal. In this figure, the amplitude variation of Beethoven. wav is relatively stable, indicating the smooth and harmonious nature of Beethoven's trio. Telemann's amplitude changes significantly, showcasing the dynamic changes and rhythmic sense of flute fantasies.

V. Results and Discussion

The results of applying AutoML and big data analysis to audio style conversion are presented. The results show that AutoML can recognize and transform audio styles more efficiently than traditional methods. By automating the process of model selection and feature extraction, the method has greatly improved the efficiency and accuracy of processing large-scale data sets. The analysis of a significant amount of audio data shows that the feature extraction algorithm developed in this study is robust and capable of capturing the fine features required for high-fidelity audio style conversion. In addition, the established evaluation method has proven effective in determining the optimal model across different data sets and transformation tasks. Challenges remain, especially with regard to the complexity and diversity of processing audio data. Despite the progress made, the deployment of these technologies in real-world scenarios such as music creation and speech synthesis needs to be further optimized. It is evident that while AutoML simplifies workflows, the complex nature of audio data requires a sophisticated understanding of domain-specific functionality, which is not fully implemented by the current AutoML framework. The discussion highlighted the importance of improving these technologies for practical applications. AutoML's potential for audio processing is enormous, but its full application is limited by current technology. Future research should therefore focus on improving AutoML's adaptability to the specific needs of audio data, ensuring that the benefits of automation can be fully utilized in both practical and creative contexts. The integration of these advanced technologies will have a major impact on areas such as music production, speech synthesis, and more, as long as subtle adaptations are taken into account.

VI. Conclusion and Future Work

Audio style conversion is an important research field in digital audio processing, with the goal of changing the style characteristics of audio content without altering it. Audio style conversion on the grounds of AutoML and big data analysis can automatically learn and convert audio styles, thereby improving the efficiency and quality of audio processing. The research results show that using iterative learning for audio style conversion training, the training curve tends to stabilize after 100 iterations, while the validation curve reaches stability after 175 iterations. In efficiency analysis, the efficiency of the yellow curve and the green curve reached 1.05 and 1.34, respectively, with the latter having significantly higher efficiency. In the audio analysis section, some parts had more obvious style transitions than others, such as Telemann's significant amplitude changes, showcasing the dynamic changes and rhythm of flute fantasies. The main contribution of the research lies in utilizing AutoML and big data analysis methods for enhancing the accuracy and efficiency of audio style conversion, offering new tools and methods for music production and sound effect design. However, this study also has some shortcomings, such as poor style switching effects in certain parts of the audio, and the need to improve sound quality. There is still a lot of room for advancement in the study of audio style conversion in future research. It is necessary to further optimize and improve the methods of AutoML and big data analysis to enhance the accuracy and efficiency of audio style conversion. It also brings new possibilities for exploring the application of audio style conversion in more fields, such as speech synthesis, music generation, entertainment industry, etc.

REFERENCES

[1] C. A. Hallin, M. Koren, A. A. Issa, O. Koren, "AutoML classifier clustering procedure," International Journal of Intelligent Systems, vol. 37, pp. 4214-4232, 2022.

[2] H. Cai, J. Lin, Y. Lin, Z. Liu, H. Tang, H. Wang, L. Zhu, S. Han, "Enable Deep Learning on Mobile Devices: Methods, Systems, and Applications," ACM Transactions on Design Automation of Electronic Systems, vol. 27, pp. 20-50, 2022.

[3] H. S. Yang, K. R. Kim, S. Kim, J. Y. Park, "Deep Learning Application in Spinal Implant Identification," Spine, vol. 46, pp. 318-324, 2021.

[4] O. Owoyele, P. Pal, A. V. Torreira, D. Probst, M. Shaxted, M. Wilde, P. K, "Senecal. Application of an automated machine learning-genetic algorithm (AutoML-GA) coupled with computational fluid dynamics simulations for rapid engine design optimization," International Journal of Engine Research, vol. 23, pp. 1586-1601, 2021.

[5] M. K. Shende, A. E. Feijoo-Lorenzo, N. D. Bokde, "cleanTS: Automated (AutoML) Tool to Clean Univariate Time Series at Microscales," Neurocomputing, vol. 500, pp. 155-176, 2021.

[6] X. He, K. Zhao, X. Chu, "AutoML: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, pp. 1-27, 2021.

[7] J. Li, S. Xiang, "Audio-lossless robust watermarking against desynchronization attacks," Signal Processing, vol. 198, pp. 108561-108573, 2022.

[8] F. Lin, S. Liu, C. Zhang, J. Fan, Z. Wu, "StyleBERT: Text-audio sentiment analysis with Bi-directional Style Enhancement," Information systems, vol. 114, pp. 1-11, 2023.

[9] B. Chen, Z. Xu, K. Yu, "Data augmentation based non-parallel voice conversion with frame-level speaker disentangler," Speech Communication, vol. 136, pp. 14-22, 2022.

[10] D. Xu, S. Zhong, J. Xu, "Bipolar Phase Shift Modulation Single-Stage Audio Amplifier Employing a Full Bridge Active Clamp for High Efficiency Low Distortion," IEEE Transactions on Industrial Electronics, vol. 68, pp. 1118-1129, 2021.

[11] R. Chandrakar, R. Raja, R. Miri, U. Sinha, A. K. S. Kushwaha, H. Raja, "Enhanced the moving object detection and object tracking for traffic surveillance using RBF-FDLNN and CBF algorithm," Expert Systems with Applications, vol. 191, pp. 1-15, 2022.

[12] J. Zhang, "Music Feature Extraction and Classification Algorithm Based on Deep Learning," Scientific Programming, ,vol. 2, pp. 1-9, 2021.

[13] P. K. Singh, I. Chatterjee, R. Sarkar, E. B. Smith, M. Nasipuri, "A new feature extraction approach for script invariant handwritten numeral recognition," Expert Systems, vol. 38, pp. 1-22, 2021.

[14] Z. G. Jiang, X. T. Shi, "Application Research of Key Frames Extraction Technology Combined with Optimized Faster R-CNN Algorithm in Traffic Video Analysis," Complexity, vol. 4, pp. 1-11, 2021.

[15] Z. Jia, Y. Li, Z. Tan, W. Wang, Z. Wang, G. Yin, "Domain-invariant feature extraction and fusion for cross-domain person re-identification," The visual computer, vol. 39, pp. 1205-1216, 2023.

[16] M. Grzegorowski, J. Litwin, M. Wnuk, M. Pabi, U. Marcinowski, "Survival-Based Feature Extraction—Application in Supply Management for Dispersed Vending Machines," IEEE transactions on industrial informatics, vol. 19, pp. 3331-3340, 2023.

[17] S. L. Wu, Y. H. Yang, "MuseMorphose: Full-song and fine-grained piano music style transfer with one transformer VAE," IEEE/ACM Trans. on Audio, Speech, and Language Processing, vol. 31, no. 5, pp. 1953-1967, 2023.

[18] A. B. Rashid, R. R. R. Ikram, Y. Thamilarasan, L. Salahuddin, N. F. Abd Yusof, Z. B. Rashid, "A Student Learning Style Auto-Detection Model in a Learning Management System," Eng. Tech. & Appl. Sci. Res., vol. 13, no. 3, pp. 11000-11005, 2023.

[19] C. Chen, Y. Hu, Q. Zhang, H. Zou, B. Zhu, E. S. Chng, "Leveraging modality-specific representations for audio-visual speech recognition via reinforcement learning," Proc. of the AAAI Conf. on Artificial Intelligence, vol. 37, no. 11, pp. 12607-12615, 2023.

[20] A. K. Nair, J. Sahoo, E. D. Raj, "Privacy preserving Federated Learning framework for IoMT based big data analysis using edge computing," Computer Standards and Interfaces, vol. 86, pp. 1-20, 2023.

[21] B. Wang, J. Wan, Y. Zhu, Y. Chen, "Institutional capability, cooperation level and irrigation water order: Empirical analysis based on survey data from the Yellow River area," Irrigation and Drainage, vol. 72, pp. 716-728, 2023.

[22] Y. Fang, B. Luo, T. Zhao, D. He, B. Jiang, Q. Liu, "ST-SIGMA: Spatio-temporal semantics and interaction graph aggregation for multi-agent perception and trajectory forecasting," CAAI Transactions on Intelligence Technology, vol. 7, pp. 744-757, 2022.

[23] J. Purohit, R. Dave, "Leveraging Deep Learning Techniques to Obtain Efficacious Segmentation Results," Archives of Advanced Engineering Science, vol. 1, pp. 1-16, 2023.