Real-time Detection of RVC-based DeepFaked Audio

Visal Dam[a]

[a]School of Information Technology, Deakin University, Burwood, VIC, Australia

Abstract

Audio impersonation has become more accessible due to the rising popularity of machine and deep learning. This report explores how machine learning can also be used to detect them in real-time. A range of statistical models are evaluated on a custom generated dataset, considering both detection accuracies and inference times. XGBoost was found to be the best overall model, with a high accuracy of 0.952 and low inference time of 0.016 ± 0.070 ms. Considering as well the time needed for feature extraction, a theoretical rate of 5 to 7.7 1-second blocks processed per second is determined, indicating the possibility of integrated real-time detection of continuous audio data. Some results of the RVC process are provided in the Appendix, namely comparing the amplitudes, Mel spectrograms, and Fast Fourier Transforms (FFT) of the original and synthesized audios. This report is an independent HD task for SIT332 Task 5.2, and has been approved by the Unit Chair, Dr. Duc Thanh Nguyen.

Keywords: **DeepFake, RVC, Machine Learning.**

1. Introduction

The rise and intertwining of deep learning and computer vision has led to the possibility of modifying what is deemed reality. Though a useful novelty, as shown in [1, 2, 3, 4], concerns arise in the possibility of malicious use by threat actors, such as scam-oriented impersonations and defamation [5]. Scam phone calls are one such case, wherein a threat actor uses techniques, such as Retrieval-based Voice Conversion (RVC), to clone someone else's voice in real-time using their own. This is a major security and privacy concern, especially given how such models and their audio training data are becoming increasingly publicly accessible.

Motivated by this, Bird & Lotfi [5] propose detection of RVC-generated audio using statistical classifiers, noting both their high detection accuracies and low inference times. The dataset generated by their study, DEEP-VOICE[4], is also provided. As an independent HD task for SIT332, this report aims to explore, analyze, and even extend their approach. The work is structured as follows:

Section 2 provides a background of RVC, a summary of [5] (as required), and discusses related work; Section 3 presents the methodology and considerations applied in data collection, processing, and analysis; Section 4 discusses and analyzes observed results; Section 5 postulates future work; and, finally, we conclude our work in Section 6.

2. Background and Literature Review

2.1. Retrieval-based Voice Conversation (RVC)

Voice conversion is a popular area in speech synthesis and is aimed at separating the *content features* (what is spoken) from the *speaker features* (how it is spoken). A voice model first disentangles the linguistic content from acoustic characteristics such as timbre, pitch, and tone [6, 7]. Deep learning methods, such as HuBERT [8] and ContentVec [9], are commonly used to extract these high-level feature representations. An acoustic model then recreates the target speaker by applying the extracted features on given content.

Retrieval-based Voice Conversion (RVC) extends this process by using a retrieval mechanism to enhance the converted voice quality. During the training process, RVC stores the target speaker's acoustic feature representations, and the highly-relevant ones are *retrieved* during runtime to guide the inference process. Essentially, the similarity between the speech features of the target and the given speaker allows the model to more accurately reconstruct the target speaker's timbre and vocal characteristics, while preserving the linguistic content, prosody, and style of the given speaker. RVC is designed to be deployed in real-time, such as via the RVC Web GUI[1]. An overview of the RVC process is presented in Figure 1.

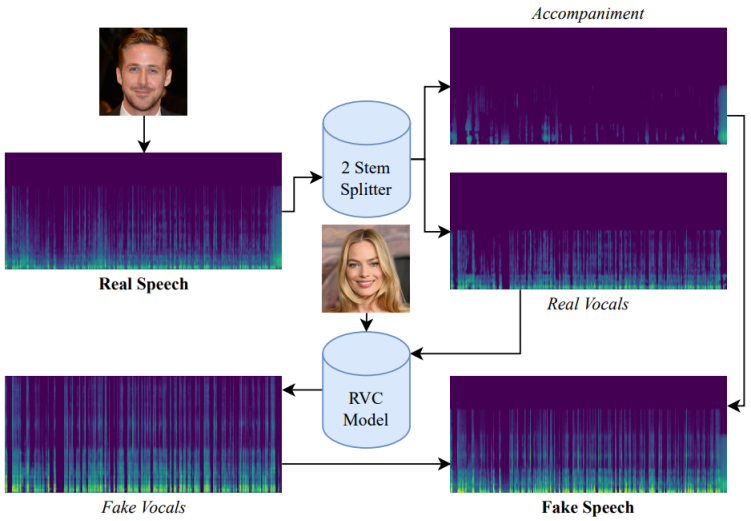


Figure 1: Overview of the RVC pipeline wherein Ryan Gosling's speech is converted to Margot Robbie's voice (taken from [5]).

2.2. Main Paper

The main objective in [5] is to use Machine Learning (ML) to detect DeepFaked audio generated through RVC. The trained model then predicts incoming audio data via 1-second blocks, alerting the user accordingly. They are motivated by RVC's ability to

convert short speech samples in real-time, thereby facilitating malicious usage such as in scam calls and misinformation through impersonation.

To combat this, the authors model their detection strategy in the same manner. They consider both efficacy and computational complexity to enable real-time detection, with their proposed system demonstrated in Figure 2. For example, they note that while techniques such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTMs), and their variants, are strong for audio data, their complexity limits their inference times. Thus, they instead focus on classical statistical models.

Audio data is continuous, and transmitted in steams in real-time contexts, such as phone calls. Thus, in training and inferring, the authors perform feature extraction on 1-second audio blocks. They begin by collecting audio data of 8 famous individuals, such as politicians and celebrities, and are each cropped to a maximum of 10 minutes. In total, they collected 62 minutes and 22 seconds of real speech data, mostly from YouTube.

Next, they use an implementation of RVC[1] to convert each individual's speech into one another. Given the fame of each target, their voice models (RVC version 2) were retrieved from public sources such as HuggingFace[2] and the AI Hub Discord server[3]. This creates a total 56 fake speech data. This obviously skews the dataset, hence they undersample fake data by random selection to achieve a 1:1 ratio.

Treating entire 1-second blocks as processed frames, they utilize Librosa [10] to extract the Chromagram, Spectral Centroid (SC), Spectral Bandwidth (SB), Spectral Rolloff (SRf), Zero Crossing Rate (ZCR), Root Mean Square (RMS), and the first 20 Mel-Frequency Cepstral Coefficients (MFCCs) for a total of 26 features.

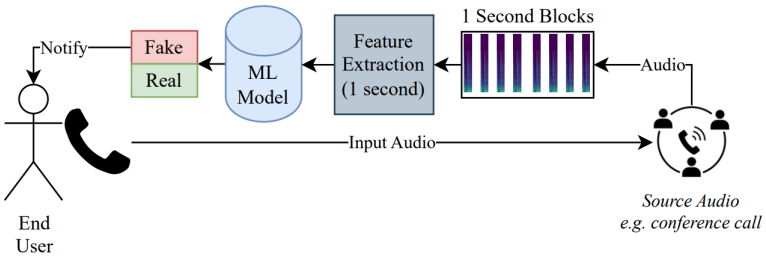


Figure 2: Proposed detection and alerting pipeline (taken from [5]).

While the authors provide the resulting dataset[4], their data processing methodology is unclear. For example, the author's dataset contains only one Chromagram column, whereas by default it is normalized into 12 bins, each representing the main notes in the Chromatic scale of Western music. Likewise, there are 5889 feature rows for both REAL and FAKE labels, each representing a 1-second block, despite the total length of audio data being $(62 * 60) + 22 = 3742$ seconds long. It can therefore be determined that the hop length of a given audio, $hop(y)$, between blocks, for a given sample rate $sr(y)$, is determined by

$$hop(y) = \frac{3742}{5889} \times sr(y) = 0.6354 \times sr(y) \quad (1)$$

Different statistical classifiers from the sk-learn library [11] are evaluated and include: Extreme Gradient Boost (XGBoost), Random Forest, classic Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), and K-Nearest Neighbors (KNN). Hyperparameterization is used to optimize the models, trained over 10-fold cross validation based on accuracy score.

XGBoost, boosted at 330 rounds, was found to be the best performing model, with the highest accuracy of 0.994. The model's precision, recall, and F scores were found to be 0.995, 0.991, and 0.991, respectively. While the authors boast a low average inference time of 0.004 ms per given block, they did not include the time taken to extract its features.

2.3. Related Work

Recent works surveying the rise in DeepFakes stress its increasing availability and detection challenges. Nguyen et al. [12] explore the efficacy of state-of-the-art tools for the creation and detection of video DeepFakes, noting that a major challenge in the latter is the lack of generalization. Likewise, they urge for the adoption of detection capabilities in social media platforms to mitigate targeted misinformation. Yi et al. [13] explore the merits and demerits of dedicated deep-learning and end-to-end detection approaches against audio DeepFakes, noting too that a lack of generalization contributes to the overall challenge.

Noting the lack of transparency and explainability of traditional ML detection, Yu et al. [14] propose fusing raw waveform signals and spectrograms, using their distributions to determine the likelihood of synthetic speech, highlighting their model's higher detection capabilities with lower performance costs. Zong et al. [15] propose embedding audio watermarks to protect real speech by stifling DeepFake voice models into learning watermarked patterns. While their results show how easier it is to detect watermarked synthetic speech, their method cannot protect currently-available nor unseen voices.

3. Methodology

3.1. Voice Conversion

Our audio dataset consists of speeches from 10 individuals and are detailed in Table 1. 5 of them are sourced from the original dataset provided by the authors in [5], while the rest are sourced from YouTube and converted to `.wav` format[5]. Their RVC (version 2) models were sourced from HuggingFace[2].

We use both male and female speakers for diversity. We use voices of recent United States presidents: Barack Obama, Joe Biden, and Donald Trump; celebrities: Ariana Grande, Michael Jackson, Elon Musk, and Taylor Swift; and online personalities: PewDiePie (Felix Kjellberg), Ludwig/MogulMail (Ludwig Ahgren), and Gawr Gura (real name unknown). We believe that this extension of the dataset reflects a higher variety of speaking styles, vocal characteristics, and levels of public exposure. Thus,

[1]The RVC Python implementation: `https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI` (last accessed Sep, 2025)
[2]RVC models on HuggingFace: `https://huggingface.co/models?other=rvc` (last accessed Sep, 2025)
[3]AI Hub Discord server: `https://discord.me/aihub` (last accessed Sep, 2025)
[4]DEEP-VOICE dataset: `https://www.DEEP-VOICE.com/datasets/birdy654/deep-voice-deepfake-voice-recognition` (last accessed Sep, 2025)
[5]WavNinja was used: `https://wav.ninja` (last accessed Sep, 2025)

our study is more applicable to real-world scenarios, such as combating the increase of impersonation of online figures.

This forms our real speech dataset and is 74 mins and 56 seconds long. Using the ratio determined in 1, this results in ≈ 7069 rows, each representing 1 second of audio. We then train 10 RVC (version 2) models, each corresponding to an of the individual, and are fed the audio of the other 9. This results in 90 fake speech files, which, after feature extraction, we then undersample (via random selection) to achieve a balanced 1:1 ratio with our real speech. The pipeline code is also provided, and was run on a T4 GPU using Google Colab (free).

Some results of the RVC process are provided in the Appendix, comparing the amplitudes, Mel spectrograms, and FFT of the original and synthesized audio. We present only the following speakers for succinctness: PewDiePie to Gawr Gura in Appendix A and vice versa in Appendix B, as well as Visal (the author) to both of the aforementioned in Appendix C (but not the other way around). The first and second appendices showcase how the RVC process affects long (6:21 mins), clean (only vocal speech) audio and short (1:03 mins), noisy (background music) audio, respectively.

Individual	Source Audio	Source RVC Model	Length
Ariana Grande	"Ariana Grande Accepts Woman of the Year Award | Women in Music"[6]	Hugging Face[2]	4:21
Barrack Obama	DEEP-VOICE[4]		10:00
Donald Trump	DEEP-VOICE[4]		10:00
Elon Musk	DEEP-VOICE[4]		10:00
Felix Kjellberg (PewDiePie)	"Let's Talk About Money"[7]		6:21
Gawr Gura	"Gura's Excellent Pep Talk"[8]		1:03
Joe Biden	DEEP-VOICE[4]		10:00
Michael Jackson	"Michael Jackson ~ Speech about helping the world"[9]		3:11
Ludwig Ahgren (Ludwig/MogulMail)	"sad news."[10]		10:00
Taylor Swift	DEEP-VOICE[4]		10:00
			74:56

Table 1: Data collected for training and validation

3.2. Feature Extraction

We use Librosa [10] to extract the same features used in [5]. However, we also chose to keep the 12 bands Chroma bands, extending our features to a total of 37. As the Chromagram captures pitch changes, we felt that this would be of significance in classifying whether a given block of audio is real or fake due to the tendency of DeepFaked audio having abnormal pitch ranges. Likewise, since RVC is not a perfect (1-to-1) voice conversion system, the generated audio can contain residual artifacts or inconsistencies. These may include unnatural blending with ambient noise, leading to distortions or 'unnatural sounding' artifacts that could serve as additional cues for detecting synthetic speech.

For a given audio file y, we first preserve its $sr(y)$ (using the `sr=None` param). We then define the block length $bl(y) = sr(y)$ so that all processed blocks are equivalent to 1 second. We also define the $hop(y)$ as in 1, which forms our overlap. Feature extraction is performed treating each block as an entire frame; this was done by setting the frame length $fl > bl(y)$ and window length $wl = fl$. This is demonstrated in Figure 3. This ensures that the feature outputs are 1D arrays representing that feature at that point in time.

The features are concatenated into a single array, representing a row. This is repeated for each block until the entire audio file is processed. Finally, all collected rows are transformed into a dataframe, and labeled either REAL or FAKE. This process is facilitated by the `get_features_row()` function. The time taken to extract features from each block is also averaged and returned. We note also that this function takes in a time parameter (in seconds) which determines how the blocks are treated. For example, if a time of $\frac{1}{3}$ seconds is passed, then for each block the feature extraction is performed over 3 frames (i.e., $fl = \frac{1}{3}bl$) rather than the entire block at once, and the resulting outputs, per frame, are stacked on top another; thus, 3 sequential rows represent features seen in 1 second of audio. After investigating this approach however, the models were found to perform better for entire blocks rather block chunks.

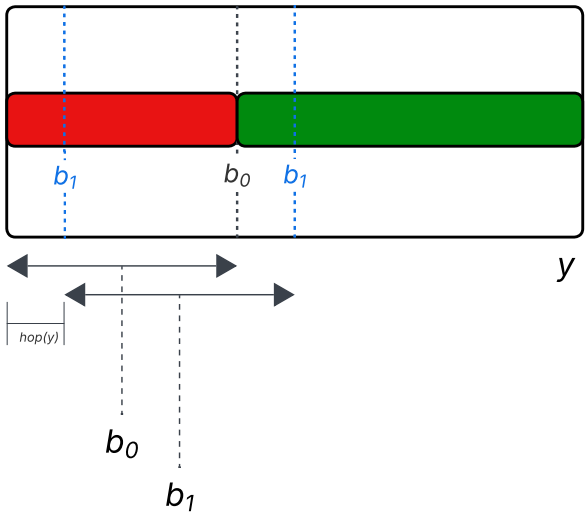


Figure 3: Splitting the audio into 1 second blocks

3.3. Machine Learning

Due to our emphasis on detection efficacy and inference time, selected for this study are the following statistical classifiers: XGBoost, Random Forest, Light Gradient-Boosting Machine (LightBGM), CatBoost, SVM, SGD, Logistic Regression, and Ridge Regression. All of the models were sourced from the SKlearn library. A seed is used to ensure reproducibility of our study. Likewise, a fixed random state of 42 is used whenever necessary, such as the train-test split and model initialization.

We trained the models over 10 folds and also used hyperparameterization, using `KFold` and `GridSearchCV`, respectively, to

[6] "Ariana Grande Accepts Woman of the Year Award | Women in Music": `https://youtu.be/BE9GDcQEIlk?si=cZevVY6aTcwrANxP`(last accessed Sep, 2025)
[7] "Let's Talk About Money: `https://youtu.be/zn0y3Opb8Wk?si=JosJZnJn3a9fcC0I` (last accessed Sep, 2025)
[8] "Gura's Excellent Pep Talk": `https://youtu.be/ogsmJIGb3QM?si=KTAdma4dyq7ZKwdu` (last accessed Sep, 2025)
[9] "Michael Jackson ~Speech about helping the world": `https://youtu.be/VvNdv6sVCPo?si=9WTc82dA5kThVfMl` (last accessed Sep, 2025)
[10] "sad news.": `https://youtu.be/z2Du27CkXM0?si=_YdKdDD_f-DfAcOt` (last accessed Sep, 2025)

ensure fair overall performance. We thus only present the results of the best performing model instances, based on the f1 score due to the binary nature of our study [11]. We note also that, from the train-test split, we only used the training data during the hyperparameter tuning phase, and then validated the models using the unseen testing data. This was to prevent bias in the resulting model's performance.

Alongside traditional metrics, i.e, the accuracy, precision, recall, and f1 scores, the authors in [5] also considered the Matthews Correlation Coefficient (MCC). This metric is useful in considering all potential correct (TP, TN) and incorrect (FP, FN) predictions [16], and is calculated via

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}. \quad (2)$$

Ranged from -1 to +1, the closer to +1 a model's MCC is the better its predictive ability. Additionally, the Receiver Operating Characteristic Area Under the Curve (ROC-AUC) is also considered, though this only applies to models that support probabilistic predictions. Ranged from 0 to 1, the closer to 1 a model's ROC-AUC is the better its predictive ability, whereas an ROC-AUC of 0.5 indicates that its predictions are as good as random guessing.

3.4. Data Analysis

To better differentiate our work from [5], in addition to the Pearson's Correlation Coefficient (PCC), we also used Principal Component Analysis (PCA) to observe the correlation of features and classes of the generated dataset. We present the sorted PCC-based feature correlation in Figure 5, where the points in orange represent the absolute value of the magnitude of correlation, with REAL = 0 and FAKE = 1. We see that the highest correlation magnitudes between feature and class is shown to be the 2^{nd} MFCC, with a PCC of 0.35. The same observation was made in [5], though with a PCC of 0.36.

Interestingly, the 2^{nd} MFCC is the lowest when considering its original negative (blue) values. This suggests that it is the strongest indicator of the REAL class. The highest correlation would therefore be attributed to the Spectral Bandwidth, with a PCC of 0.28, thereby being the strongest indicator of the FAKE class.

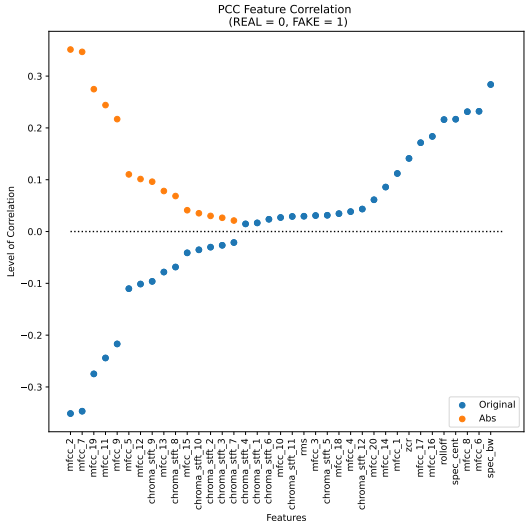


Figure 5: PCC Feature Correlation

While the PCA is strictly a technique to reduce data dimensionality (and, therefore, complexity) and does not directly compute correlation, its reduced components do reveal their distribution among classes. The PCA Feature correlations of the dataset in [5] and ours are presented in Figures 4a and 4b, respectively. We see that in both datasets, the correlation of features corresponding to the FAKE class tend to form close clusters while those corresponding to the REAL class are more spread out, thus revealing the general uniformity and less natural characteristics of RVC-based audio. Given out work's higher variety of speakers, these clusters are larger.

Note also that the features have not been scaled. Their scaled variants, using the Standard, Min-Max, and Robust Scalers, are presented in Figures 4c, 4d, and 4e, respectively. This reveals a greater level of separation between the classes, which should result in more accurate detections, with the Robust and Min-Max Scalers showing the best and worst separations of feature correlations, respectively.

The tree-based classifiers were found to perform better on raw data, while the other classifiers performed best after feature scaling using the Robust Scaler as they rely on distances and margins that are tied to the variance in feature sizes. However, it is important to note that feature scaling adds an extra layer of processing time, thus these approaches may not be as suited for real-time contexts.

4. Discussion of Results

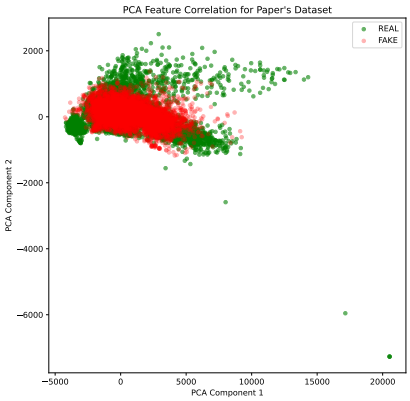
4.1. Validation Metrics

Presented in Table 2 are the averaged validation metrics of the evaluated models. The average inference time refers to the time taken to predict the class of a given 1-second block of speech audio. For tree-based classifiers, feature importance is also provided.

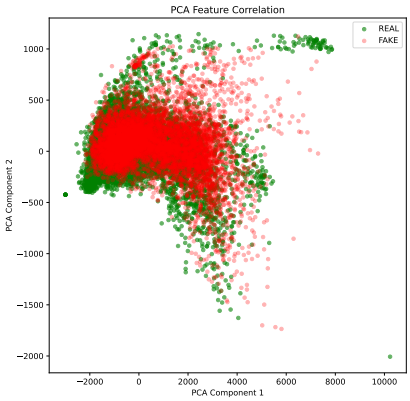
Unlike [5], the most accurate model was found to be the SVC, with an accuracy, and averaged precision, recall, and f1 scores, of 0.957. This is slightly higher than the second best performer, XGBoost, at 0.952, which was the best model in [5]. However, their inference times, 0.351 ms and 0.008 ms, respectively, differ by a factor of $\frac{0.351}{0.008} = 43.875$. Hence, XGBoost performs around 44 times faster than SVC with relatively low accuracy tradeoffs, which is critical for real-time detection. Thus, based on these constraints, we consider XGBoost to be the best model overall.

It should be noted that, while SVC supports probabilistic predictions (via the `probability=True` param) we found that it far longer training times yet performed the same as when this parameter was disabled. Hence, we omit SVC's ROC-AUC.

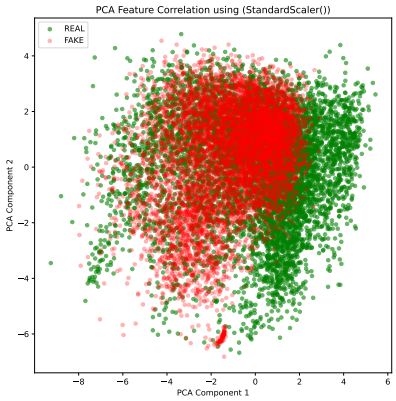
Among the tree-based classifiers, the majority agree that the 7^{th} MFCC is the most important feature, with the exception of XGBoost, which attributed this to the 2^{nd} MFCC. This coincides with both the earlier PCC analysis and the findings in [5]. Additionally, both XGBoost and LightBGM agree that the 10^{th} Chroma STFT is the least important, whereas RandomForest and CatBoost attribute this to the 8^{th} and 11^{th} Chroma STFT, respectively. Thus, results suggest that pitch changes, specifically the musical notes G#, A#, and B, are not as relevant for detection as previously hypothesized.



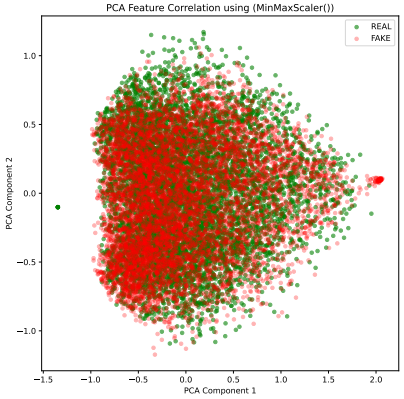
(a) Dataset from [5]



(b) This paper's dataset



(c) This paper's dataset after using the Standard Scaler



(d) This paper's dataset after using the Min-Max Scaler

(e) This paper's dataset after using the Robust Scaler

Figure 4: Comparison of PCA Feature Correlation (dim=2)

4.2. Averaged Processing Times

To better average the processing time, we run each of the best model instances 10 times to infer on randomly chosen unseen data, such as both real and fake audio generated from the authors of this paper. The means μ and standard deviations σ of the inference time I (model-specific), feature extraction time F, and scaling time S (for non-tree classifiers) are presented in Table 3, Figure 6, and Figure 7, respectively.

Model	**Inference Time (ms)**
XGBoost	0.016 ± 0.070
LightBGM	0.072 ± 0.061
RandomForest	0.129 ± 0.034
CatBoost	0.054 ± 0.023
kNN	0.074 ± 0.014
Logistic Regression	0.019 ± 0.006
Ridge Regression	0.013 ± 0.009
SVC	0.322 ± 0.058
SGD	0.014 ± 0.011

Table 3: Averaged inference times (ms) across models (10 iterations)

The averaged total time T to process a 1-second audio block is defined by

$$T = \mu_T \pm \sigma_T \tag{3}$$

where

$$\mu_T = \begin{cases} \mu_F + \mu_I & \text{if tree-based classifier} \\ \mu_F + \mu_I + \mu_S & \text{else} \end{cases} \tag{4}$$

and

$$\sigma_T = \begin{cases} \sqrt{\sigma_F^2 + \sigma_I^2} & \text{if tree-based classifier} \\ \sqrt{\sigma_F^2 + \sigma_I^2 + \sigma_S^2} & \text{else.} \end{cases} \tag{5}$$

Taking XGboost as an example, the total processing time would thus be $(163.96 + 0.016) \pm \sqrt{33.83^2 + 0.07^2} = 163.976 \pm 33.82$ms or 0.164 ± 0.03s per 1-second block, which is equivalent to a range of around 5.06 to 7.68 blocks per second. We believe this rate to be reasonable to process continuous data streams.

	Model	Feature Scaling	Accuracy Score	F1 Score	Precision	Recall	MCC	ROC−AUC	Average Inference Time (ms)	Least Important Feature	Most Important Feature
Trees	XGBoost (800)	No	0.952	0.952	0.952	0.952	0.903	**0.990**	0.008	10^{th} Chroma STFT	2^{nd} MFCC
	LightBGM (800)	No	0.938	0.938	0.938	0.938	0.876	0.986	0.024	10^{th} Chroma STFT	7^{th} MFCC
	RandomForest (50)	No	0.914	0.914	0.914	0.914	0.827	0.977	0.012	8^{th} Chroma STFT	7^{th} MFCC
	CatBoost (800)	No	0.856	0.853	0.885	0.856	0.741	0.983	0.007	11^{th} Chroma STFT	7^{th} MFCC
Distance	kNN	Yes	0.930	0.930	0.930	0.930	0.860	0.980	0.638	N/A	N/A
Regression	Logistic Regression	Yes	0.841	0.841	0.841	0.841	0.682	0.924	0.001	N/A	N/A
	Ridge Regression	Yes	0.839	0.839	0.839	0.839	0.678	N/A	0.001	N/A	N/A
SVM	SVC	Yes	**0.957**	**0.957**	**0.957**	**0.957**	**0.914**	N/A	0.351	N/A	N/A
	SGD	Yes	0.832	0.832	0.832	0.832	0.666	N/A	0.001	N/A	N/A

Table 2: Comparison of averaged validation metrics

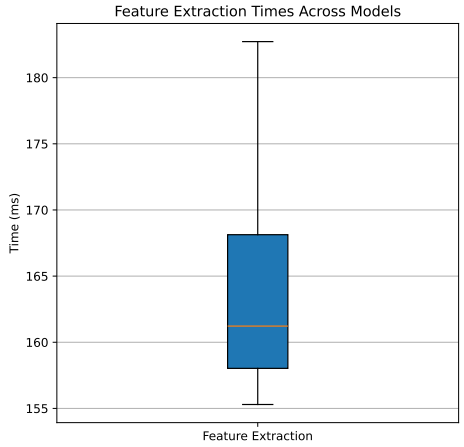


Figure 6: Box plot of averaged feature extraction times (ms) across models (10 iterations)

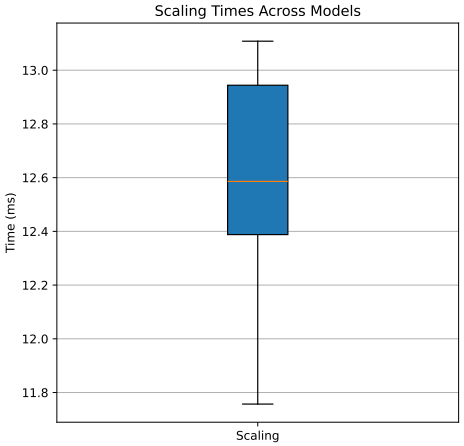


Figure 7: Box plot of averaged scaling times (ms) across non-tree models (10 iterations)

5. Future Work

Although our models slightly underperformed compared to [5], in terms of slightly lower accuracies and slightly higher inference times, we note that this is due to two main reasons:

1. We extended the feature dimensionality, length, and diversity of individuals in our dataset. Likewise, our hyperparameter tuning used training data and was validated using unseen test data. Therfore, while our metrics are lower, they are also more realistic.
2. The authors in [5] trained their models on an Intel Core i7 CPU, whereas we only had access to an Intel Core i5 CPU, thus lowering inference speeds.

Thus, future work should focus on incorporating speech from more public individuals into the dataset, as we only expanded to 10. Likewise, future work can look into improving the processing times by extracting and optimizing only the significant features, as well as explore the integration of such detection models in real-world pipelines.

6. Conclusion

The rise of DeepFake technology, such as the RVC technique, has enabled threat actors to impersonate famous individuals in both online and real-time offline scam operations. Basing our work on [5], this study explores how ML to detect against RVC-generated speech. We employ our own data processing methodology, and have extended the speech dataset to include a greater range of public individuals and features. We found that the XGBoost, at 800 rounds, to be the best overall model, with an accuracy score of 0.952 and average inference time of 0.016 ± 0.070 ms, demonstrating both high detection capabilities and low performance costs. Considering as well the time needed for feature extraction, using the XGBoost model thus results in a theoretical rate of 5 to 7.7 1-second blocks processed per second, thereby indicating the possibility of real-time detection of continuous audio data.

Data availability

The code for data collection and model training, as well as the final generated dataset, can be found at `https://deakin365-my.`

sharepoint.com/:f:/g/personal/s223058093_deakin_edu_au/Et5yr-BQMUdBmI06G9Pd3r0BrfxgNGvL-_M-KBD7pIev8g.

References

[1] L. Fischbach, A. Karimi, C. Kleen, A. Lameli, and L. Flek, "Improving low-resource dialect classification using retrieval-based voice conversion," 07 2025.

[2] T. Szabo, R. Zitný, P. Ildikó, and P. Pšenák, "Using retrieval-based voice conversion in educational video materials," *R&E-SOURCE*, pp. 352–362, 03 2025.

[3] A. Kamble, A. Tathe, S. Kumbharkar, A. Bhandare, and A. C. Mitra, "Custom data augmentation for low resource asr using bark and retrieval-based voice conversion," 2024. [Online]. Available: https://arxiv.org/abs/2311.14836

[4] A. M. Alhumud, M. AL-Qurishi, Y. O. Alomar, A. Alzahrani, and R. Souissi, "Improving automated speech recognition using retrieval-based voice conversion," in *The Second Tiny Papers Track at ICLR 2024*, 2024. [Online]. Available: https://openreview.net/forum?id=OMBFB6pU6c

[5] J. J. Bird and A. Lotfi, "Real-time detection of ai-generated speech for deepfake voice conversion," 2023. [Online]. Available: https://arxiv.org/abs/2308.12734

[6] T. Walczyna and Z. Piotrowski, "Overview of voice conversion methods based on deep learning," *Applied Sciences*, vol. 13, no. 5, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/5/3100

[7] H. H. Tan, "Understanding rvc - retrieval-based voice conversion," https://gudgud96.github.io/2024/09/26/annotated-rvc/#, 2024, [Accessed 15-10-2025].

[8] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," 2021. [Online]. Available: https://arxiv.org/abs/2106.07447

[9] K. Qian, Y. Zhang, H. Gao, J. Ni, C.-I. Lai, D. Cox, M. Hasegawa-Johnson, and S. Chang, "Contentvec: An improved self-supervised speech representation by disentangling speakers," 2022. [Online]. Available: https://arxiv.org/abs/2204.09224

[10] B. McFee, C. Raffel, D. Liang, D. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," 01 2015, pp. 18–24.

[11] scikit learn, scikit-learn User Guide, accessed 8 Sep 2025. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html#scoring-string-names

[12] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen, "Deep learning for deepfakes creation and detection: A survey," *Computer Vision and Image Understanding*, vol. 223, p. 103525, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314222001114

[13] J. Yi, C. Wang, J. Tao, X. Zhang, C. Y. Zhang, and Y. Zhao, "Audio deepfake detection: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2308.14970

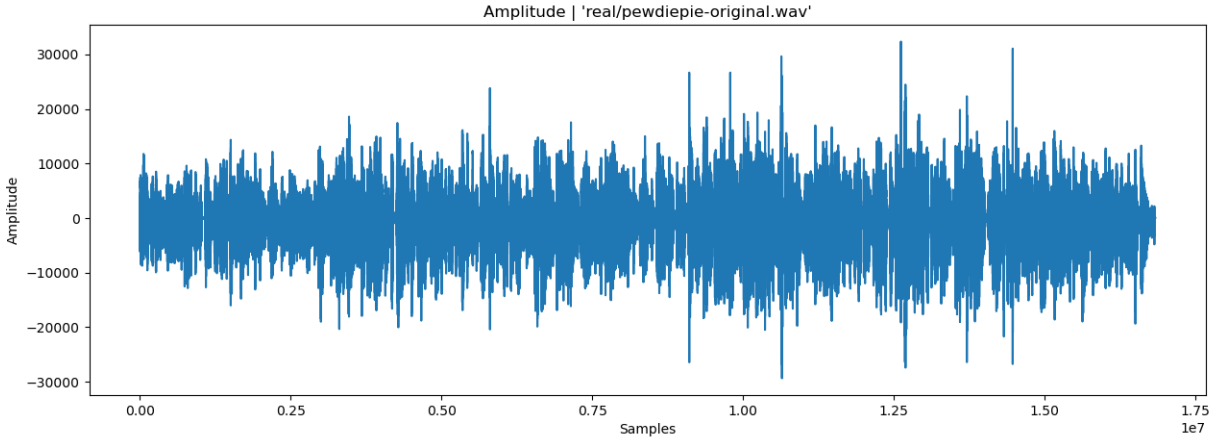
[14] N. Yu, L. Chen, T. Leng, Z. Chen, and X. Yi, "An explainable deepfake of speech detection method with spectrograms and waveforms," *Journal of Information Security and Applications*, vol. 81, p. 103720, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212624000231

[15] W. Zong, Y. Chow, W. Susilo, and J. Baek, "usenix.org," https://www.usenix.org/system/files/usenixsecurity25-zong.pdf, 2025, [Accessed 08-09-2025].

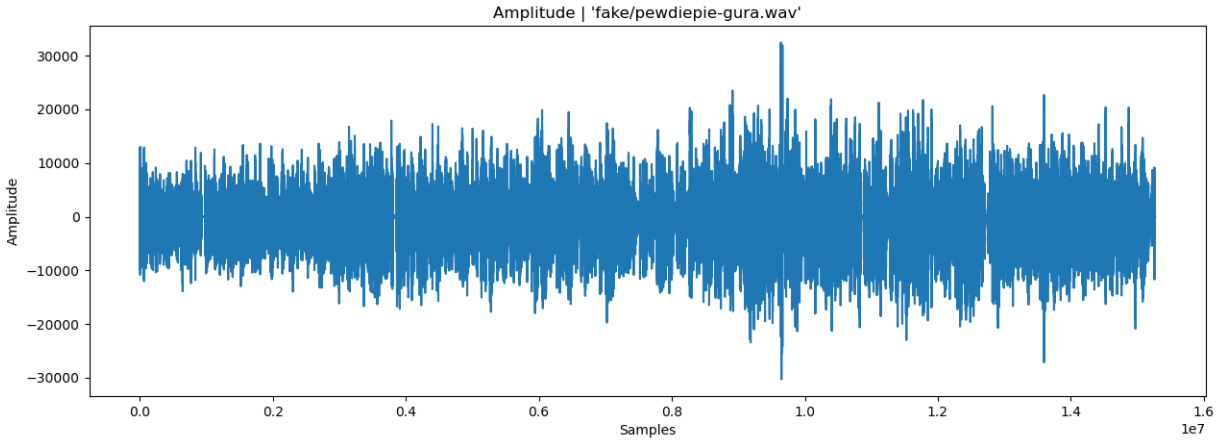
[16] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Jan 2020. [Online]. Available: https://doi.org/10.1186/s12864-019-6413-7

Appendix A. PewDiePie (Long - 6:21 mins, Clean - only vocal speech) to Gawr Gura

Appendix A.1. Amplitude



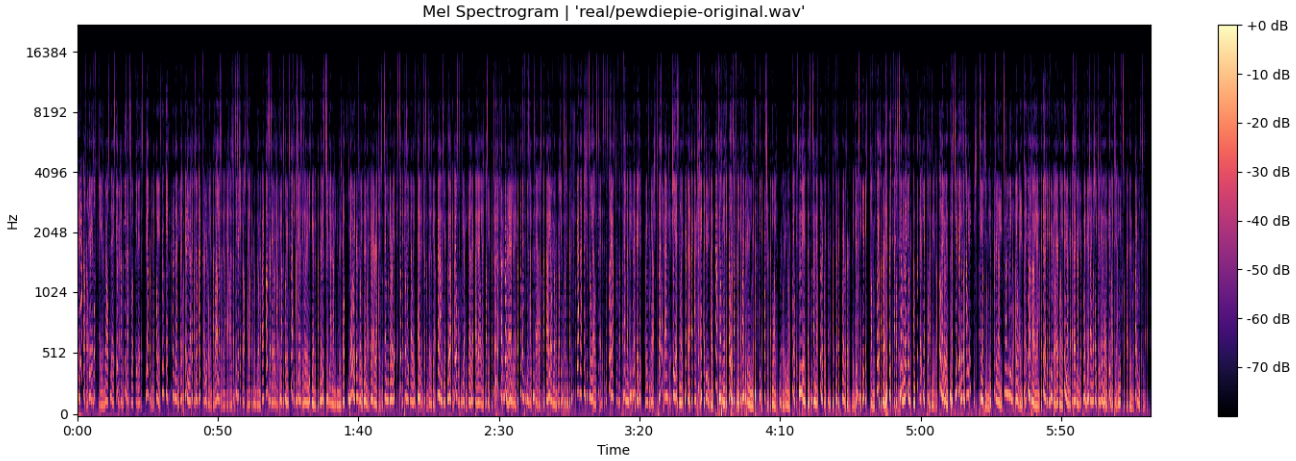
(a) Amplitude of original audio



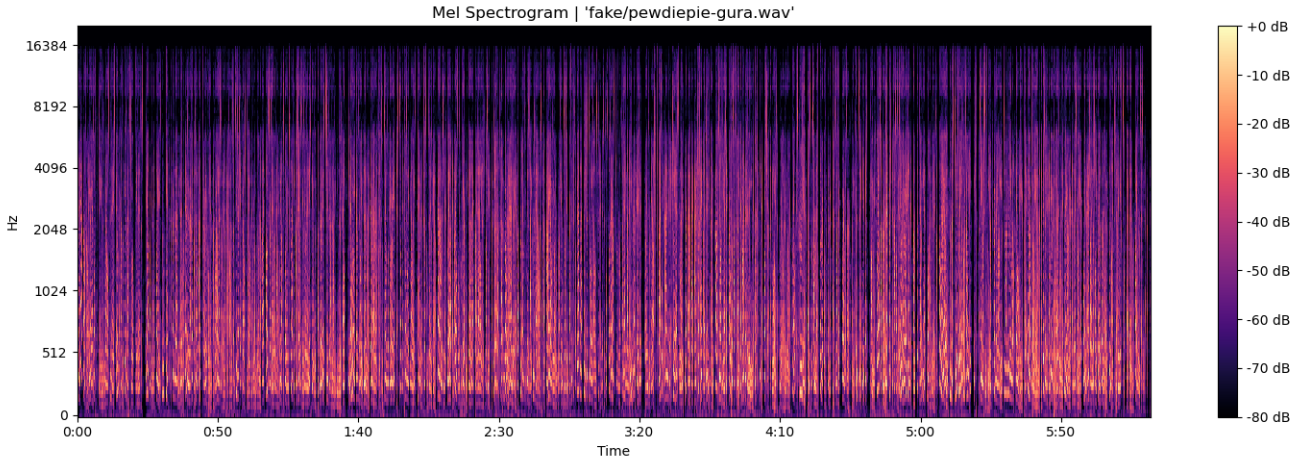
(b) Amplitude of RVC audio

Figure A.8: Amplitude

Appendix A.2. Mel Spectrogram (sr = original sr, hop_length = 512, n_fft = 2048)



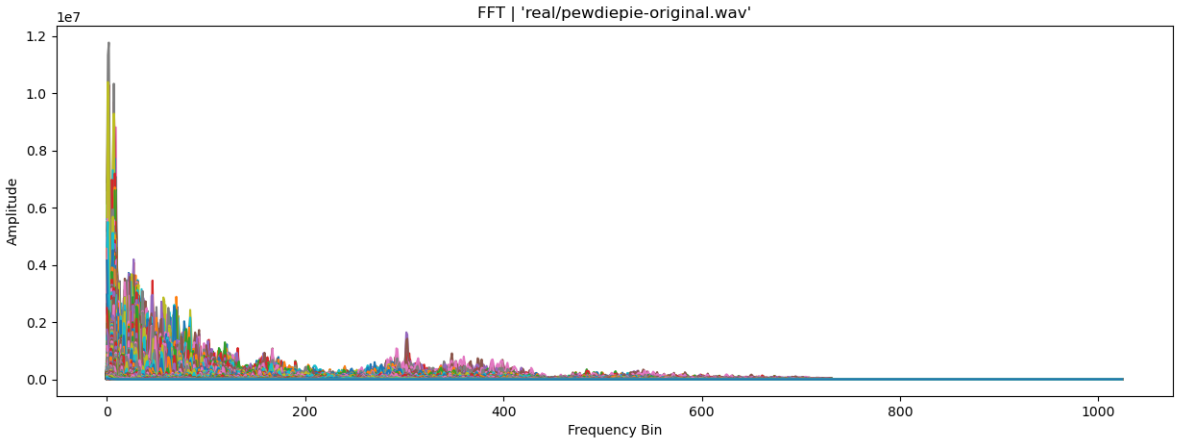
(a) Mel Spectrogram of original audio



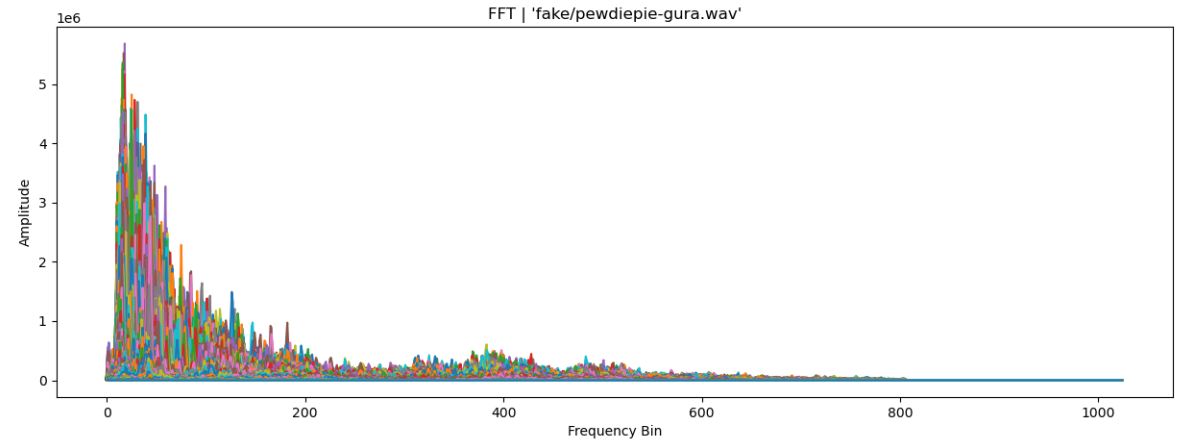
(b) Mel Spectrogram of RVC audio

Figure A.9: Mel Spectrogram

Appendix A.3. Fast Fourier Transform (FFT) (n_fft = 2048, window = 'hann')



(a) FFT of original audio

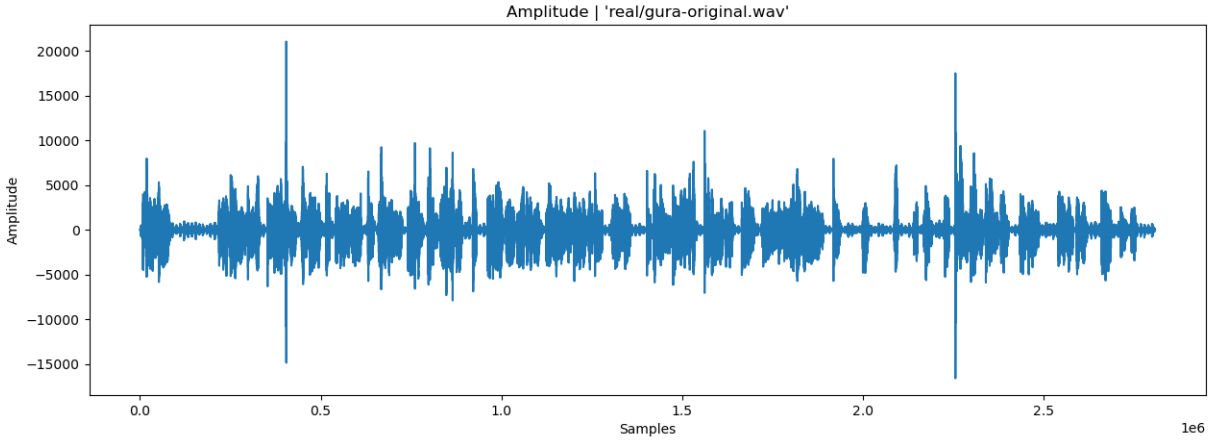


(b) FFT of RVC audio

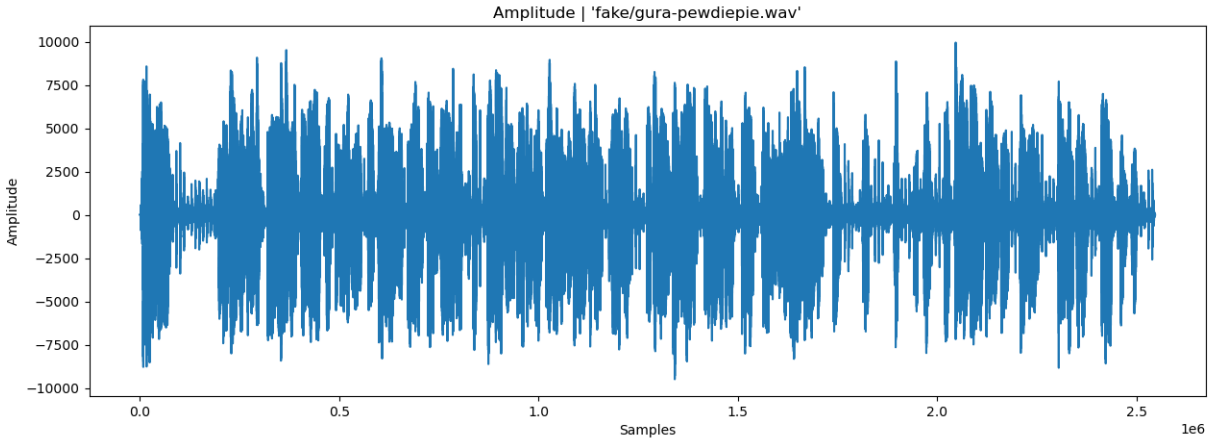
Figure A.10: FFT

Appendix B. Gawr Gura (Short - 1:03 mins, Noisy - background music) to PewDiePie

Appendix B.1. Amplitude



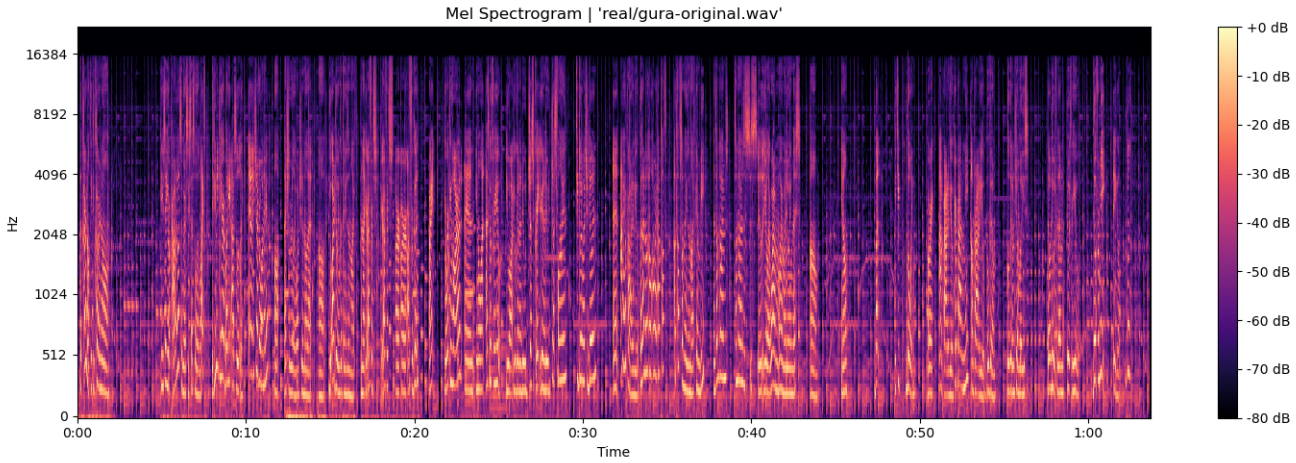
(a) Amplitude of original audio



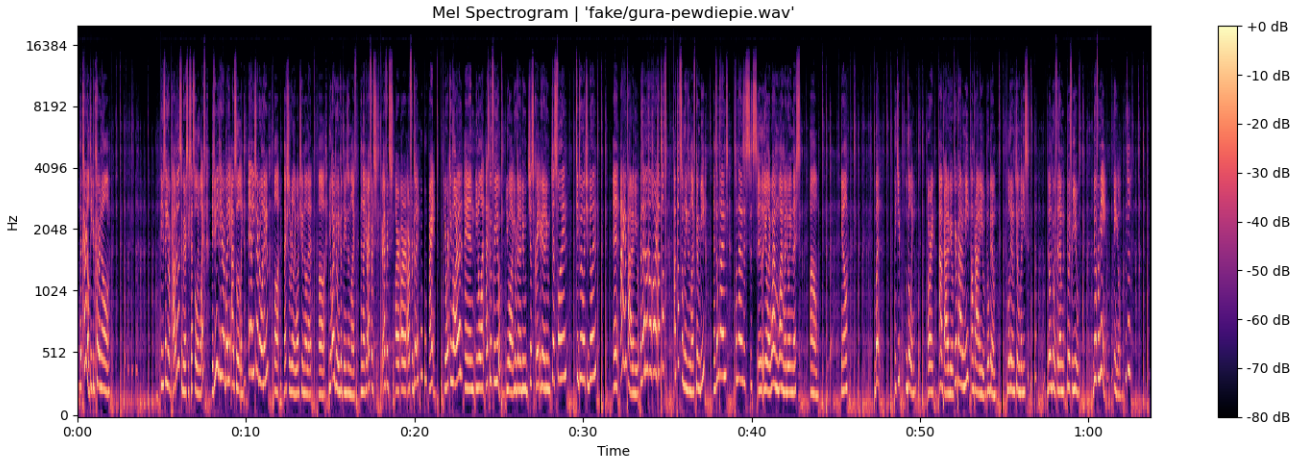
(b) Amplitude of RVC audio

Figure B.11: Amplitude

Appendix B.2. Mel Spectrogram (sr = original sr, hop_length = 512, n_fft = 2048)



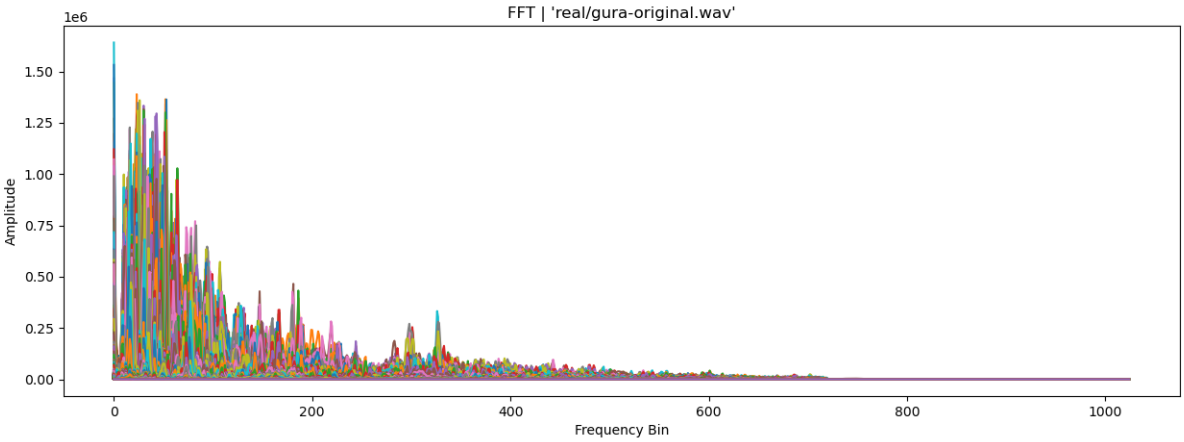
(a) Mel Spectrogram of original audio



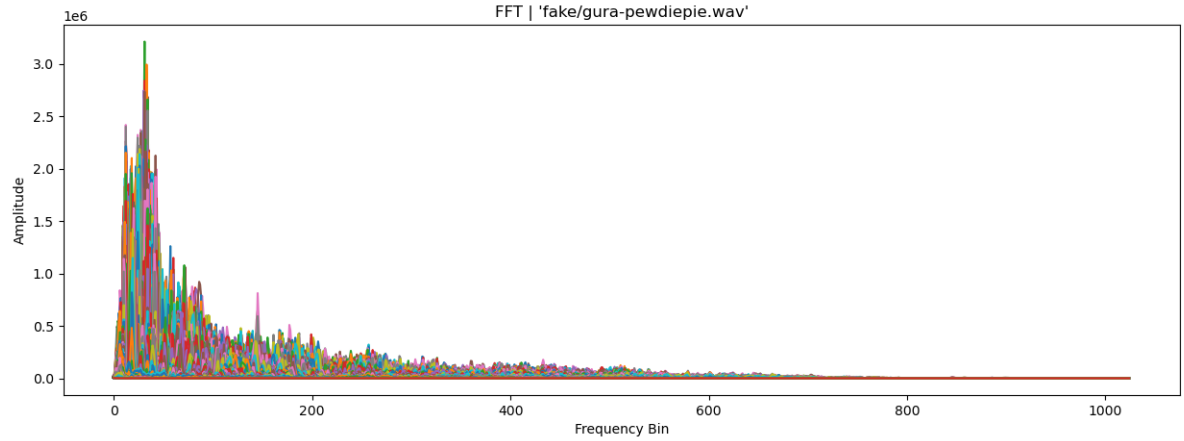
(b) Mel Spectrogram of RVC audio

Figure B.12: Mel Spectrogram

Appendix B.3. Fast Fourier Transform (FFT) (n_fft = 2048, window = 'hann')



(a) FFT of original audio

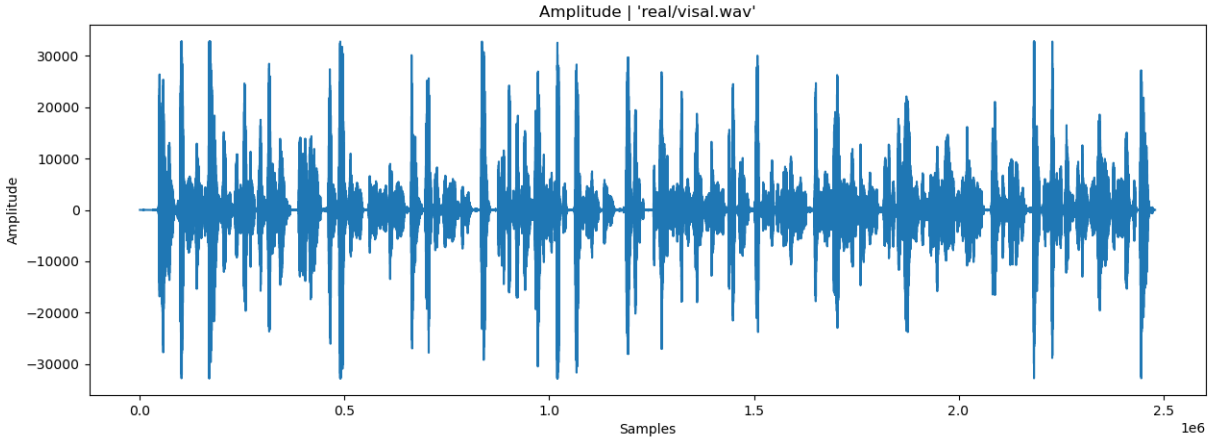


(b) FFT of RVC audio

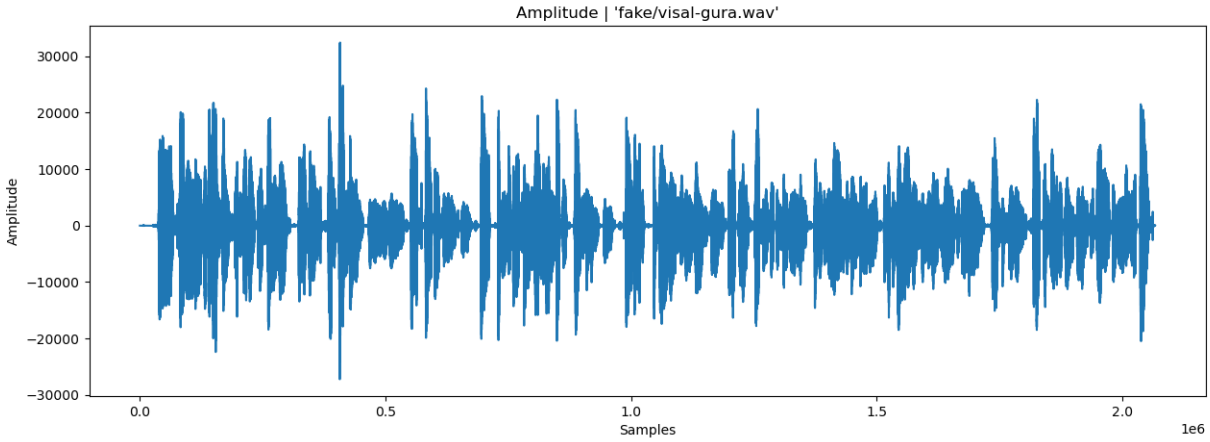
Figure B.13: FFT

Appendix C. My Voice (Short - 0:51 mins, Clean - only vocal speech) to Gawr Gura and PewDiePie

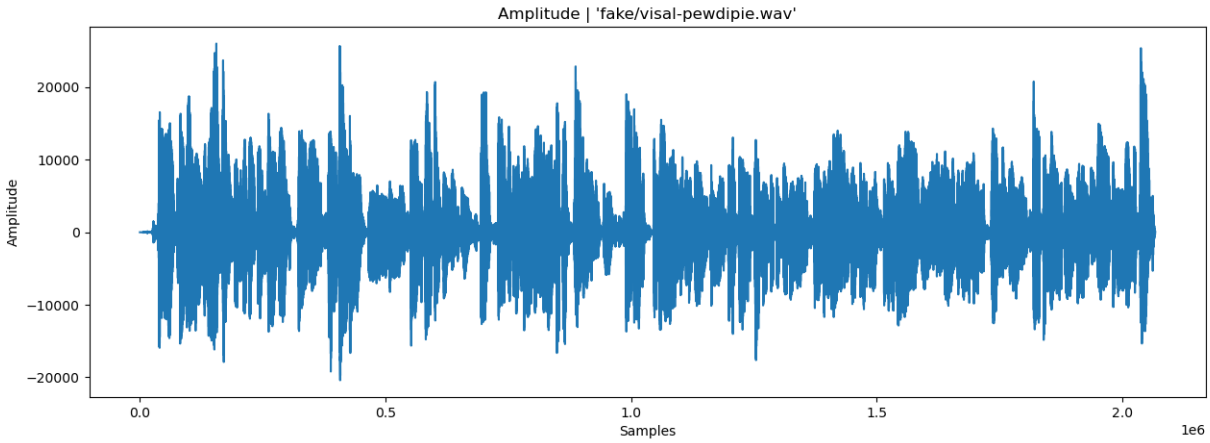
Appendix C.1. Amplitude



(a) Amplitude of original audio



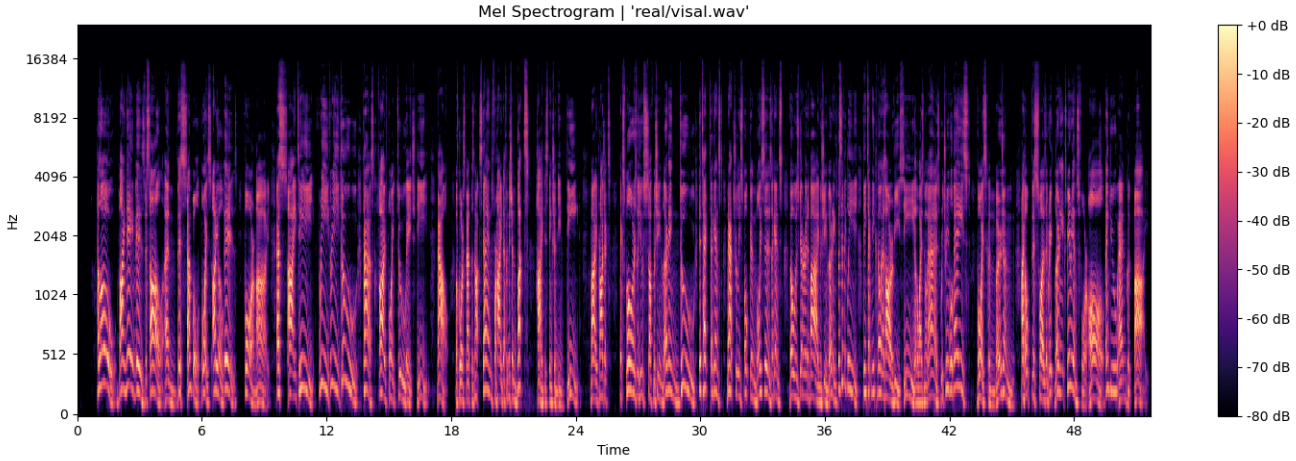
(b) Amplitude of RVC audio (to Gawr Gura)



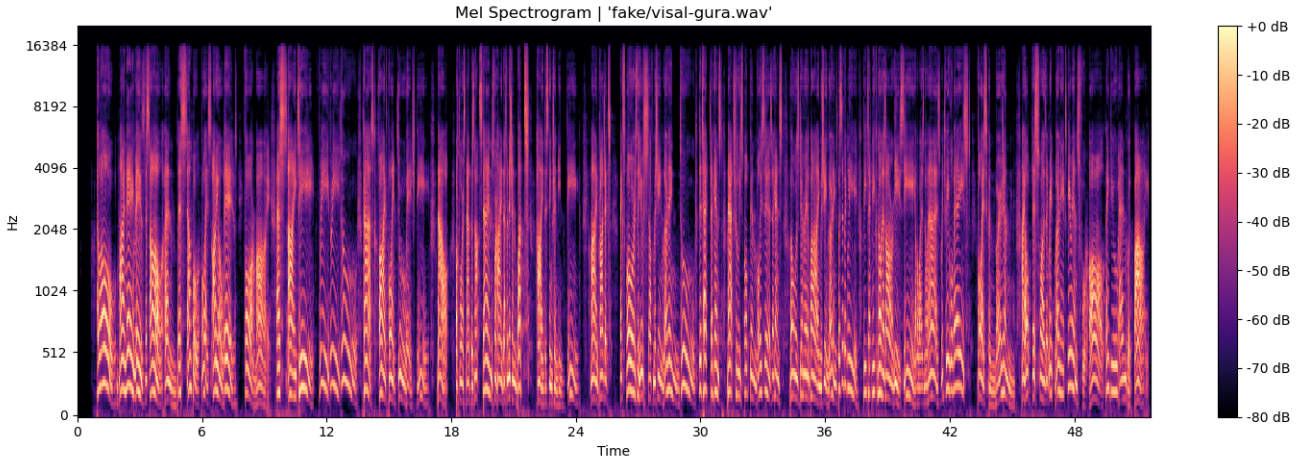
(c) Amplitude of RVC audio (to PewDiePie)

Figure C.14: Amplitude

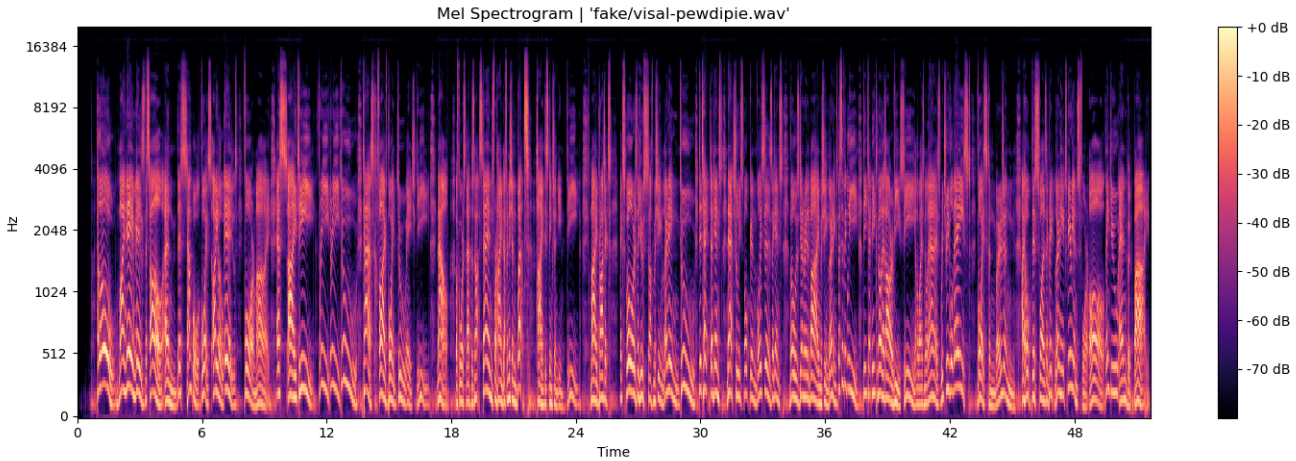
Appendix C.2. Mel Spectrogram (sr = original sr, hop_length = 512, n_fft = 2048)



(a) Mel Spectrogram of original audio



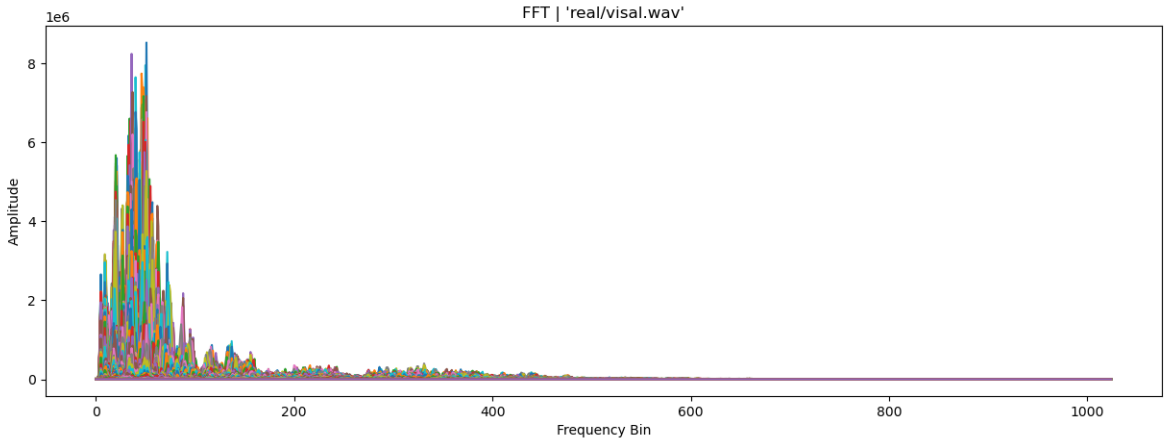
(b) Mel Spectrogram of RVC audio (to Gawr Gura)



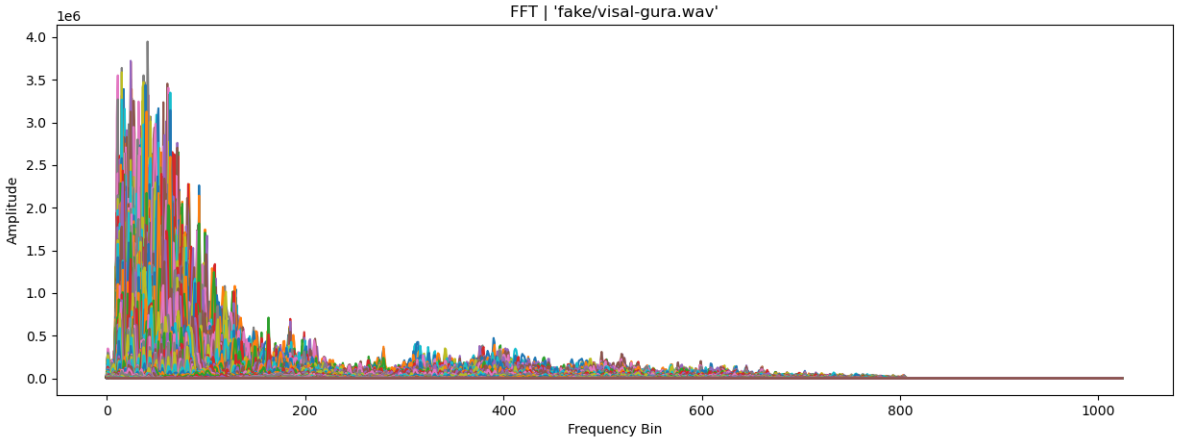
(c) Mel Spectrogram of RVC audio (to PewDiePie)

Figure C.15: Mel Spectrogram

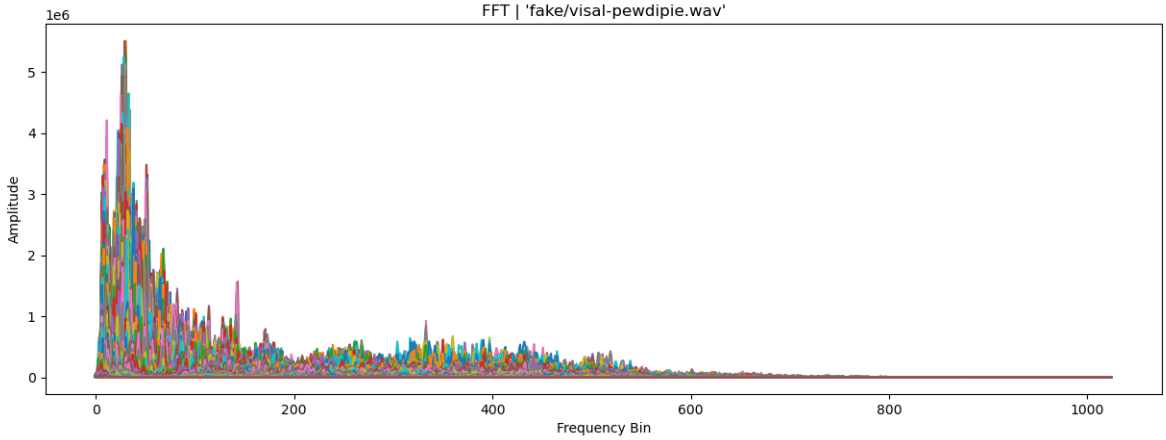
Appendix C.3. Fast Fourier Transform (FFT) (n_fft = 2048, window = 'hann')



(a) FFT of original audio



(b) FFT of RVC audio (to Gawr Gura)



(c) FFT of RVC audio (to PewDiePie)

Figure C.16: FFT