data

MDPI

Data Descriptor

Spectrogram Dataset of Korean Smartphone Audio Files Forged Using the “Mix Paste” Command

Yeongmin Son [1], Won Jun Kwak [2] and Jae Wan Park [3,*]

[1] Department of Digital Media, Soongsil University, 50 Sadang-ro, Dongjak-gu, Seoul 07027, Republic of Korea
[2] School of Business Administration, Soongsil University, 369 Sangdo-ro, Dongjak-gu, Seoul 06978, Republic of Korea
[3] Global School of Media, Soongsil University, 50 Sadang-ro, Dongjak-gu, Seoul 07027, Republic of Korea
* Correspondence: jaewan.park@ssu.ac.kr

Abstract: This study focuses on the field of voice forgery detection, which is increasing in importance owing to the introduction of advanced voice editing technologies and the proliferation of smartphones. This study introduces a unique dataset that was built specifically to identify forgeries created using the “Mix Paste” technique. This editing technique can overlay audio segments from similar or different environments without creating a new timeframe, making it nearly infeasible to detect forgeries using traditional methods. The dataset consists of 4665 and 45,672 spectrogram images from 1555 original audio files and 15,224 forged audio files, respectively. The original audio was recorded using iPhone and Samsung Galaxy smartphones to ensure a realistic sampling environment. The forged files were created from these recordings and subsequently converted into spectrograms. The dataset also provided the metadata of the original voice files, offering additional context and information that could be used for analysis and detection. This dataset not only fills a gap in existing research but also provides valuable support for developing more efficient deep learning models for voice forgery detection. By addressing the “Mix Paste” technique, the dataset caters to a critical need in voice authentication and forensics, potentially contributing to enhancing security in society.

Dataset: https://drive.google.com/drive/folders/10cBCvQTF-XqCfdQuUU4y_ssrbi3hUJkw (accessed on 19 November 2023).

Dataset License: CC BY-NC-ND

Keywords: forged Korean audio file; spectrogram dataset; smartphone audio; Mix Paste; audio forensic

check for updates

Citation: Son, Y.; Kwak, W.J.; Park, J.W. Spectrogram Dataset of Korean Smartphone Audio Files Forged Using the “Mix Paste” Command. *Data* **2023**, *8*, 183. https://doi.org/10.3390/data8120183

Academic Editor: Giuseppe Ciaburro

Received: 19 October 2023
Revised: 17 November 2023
Accepted: 29 November 2023
Published: 1 December 2023

1. Summary

With the rapid advancement and penetration of smartphones, voice recording using these devices has become common. Additionally, the importance of audio authentication and voice recording forensics is increasing owing to developments in voice file-editing software [1]. Recently, voice processing using Deep Voice 3 software and deepfakes has introduced significant challenges related to the integrity and authenticity of digital evidence [2,3]. These forged audio files cannot be detected using the human ear, necessitating the development of powerful voice forgery detection technology [4].

Editing techniques for voice files can vary from audio enhancement to pitch manipulation; however, the basic editing functions used in voice file forgery and forgery techniques, which are deletion, insertion, and copy–move, are standard. Among these, copy–moves are difficult to detect because the forged segments originate from the same audio file [5]. Recently, with the popularity of audio editing software such as Adobe Audition CC, audio file content can be easily edited in various ways [4,6]. In addition to basic editing functions, this software provides the “Mix Paste” command, which selects the desired part of a current audio file, copies it, and combines it by pasting. Using this command, it is possible to overlap or compose an empty space without creating a new timeframe [7], making it difficult for existing detection methods to detect forged content. Moreover, using this command, voice clips created with Deep Voice software can be easily synthesized into voice files recorded in a physical environment. Recently, research on detecting voice-forged files created through editing techniques such as splicing and copy–move [5,8–10] and voice-forged files created through Deep Voice software and audio deepfakes [11–13] has been actively conducted using deep learning models, which mainly convert voice signals into spectrograms and use them as a dataset.

The purpose of this study was to construct a spectrogram dataset of forged Korean voice files recorded on smartphones to develop a deep learning model for detecting voice files that were forged using the "Mix Paste" command. This dataset consisted of spectrogram images that had been converted from original audio files and spectrogram images that had been converted from forged audio files edited by "Mix Paste." Furthermore, this dataset provides metadata about the original voice file and the location of the section where this command was applied. The original recording file used here was obtained with the consent of the speakers. However, our dataset was constructed using high-resolution spectrogram images to support enhanced privacy protection and provide easier and faster access for developing deep learning models for voice forgery detection. In other words, the training time for developing deep learning models can be reduced, and these models can even be operated on personal computers with low performance.

To build this dataset, four speakers were recorded using iPhone and Samsung Galaxy smartphones, and forged files were created using these recorded files. Additionally, the metadata of the original voice file were extracted, and the forged section of the forged voice file was recorded. After the forged voice file was encoded into the same voice file as that of the original file, the original and forged voice files were converted into a spectrogram and saved as an image file.

Currently, there are datasets for detecting audio deepfakes. However, datasets that can detect audio files that have been forged or altered by editing are rare. Existing audio deepfake detection datasets include Automatic Speaker Verification spoof (ASVspoof) 2021 [14], WaveFake [15], and In-the-Wild Audio Deepfake Data [16]. ASVspoof 2021 is a dataset containing representative audio deepfakes and is entirely composed of "logical access", "physical access", and "speech deepfake" [17]. Additionally, there is a Chinese dataset called the Yuan Ze Mandarin Dataset, which focuses on detecting forgeries produced through editing. This dataset was constructed by manually applying deletion and splicing identical sentences using Audacity, audio editing software [18].

To the best of our knowledge, there are no voice datasets that have been edited using "Mix Paste." However, this dataset has limitations in that it is a Korean dataset, and the raw data were constructed by four speakers. Nevertheless, compared to classical datasets edited based on speech corpus, this dataset is a real forged dataset that was directly recorded and edited by humans, re-encoded in a similar way to the original, and provides information such as the forged sections and the bona fide sections. In these respects, this dataset is valuable.

The proposed dataset can be used to derive insights through data analysis that will be useful in detecting voice forgeries and developing useful detection algorithms. Additionally, this dataset can be used in various machine-learning fields, such as classification, to determine the type of recording device that is used in a scenario. Furthermore, we believe that releasing this dataset will contribute to the advancement of deep learning technologies used for detecting forgeries in voice files.

2. Data Description

The constructed spectrogram dataset contained 1555 original voice files and 15,224 forged voice files that had been edited with "Mix Paste." The original and forged voice files were converted into log, linear, and Mel spectrograms; there were 4665 and 45,672 spectrogram images from the 1555 original voice files and forged voice files, respectively. The spectrogram images were saved at a high resolution of 4608 × 3456 (linear/log) and 4320 × 2880 (Mel).

2.1. Original Audio

The original audio was recorded as 933 and 622 files on iPhone and Samsung Galaxy smartphones, respectively, with a total recording time of 122,378 s. The 1555 original files had minimum, maximum, and average lengths of 80, 100, and 90 s, respectively. Each voice file contained 10–15 utterances, with an average of 13 utterances. Additionally, metadata extracted from the original voice file were provided to increase the usability of this dataset.

2.2. Forged Audio

There were 15,224 voices forged through "Mix Paste" based on the original voice recording file, with minimum, maximum, and average lengths of 66, 93, and 78 s, respectively. Information on the section to which this command had been applied and the section of the source that had been used was also provided. Table 1 shows the specifications of the dataset.

Table 1. Specifications of the dataset.

Specifications	Original Audio	Forged Audio
No. of Spectrogram Images	4665	45,672
Max. Audio Length	93.1	93.1
Min Audio Length	66.3	66.3
Average Audio Length	78.7	78.8
Total Audio Length	122,378	1,199,651
Max. No. of Utterances	16	17
Min. No. of Utterances	7	8
Linear/Log Spectrogram Image Size	4608 × 3456 (W × H)	4608 × 3456 (W × H)
Mel Spectrogram Image Size	4320 × 2880 (W × H)	4320 × 2880 (W × H)

3. Methods

The process that was employed to construct the forged voice dataset can be divided into four steps: (1) voice recording, (2) editing using "Mix Paste", (3) encoding, and (4) data preprocessing. Through these steps, a forged voice file was created, and the original and forged audio were converted into spectrogram images and saved. Figure 1 illustrates the dataset construction process that we used.

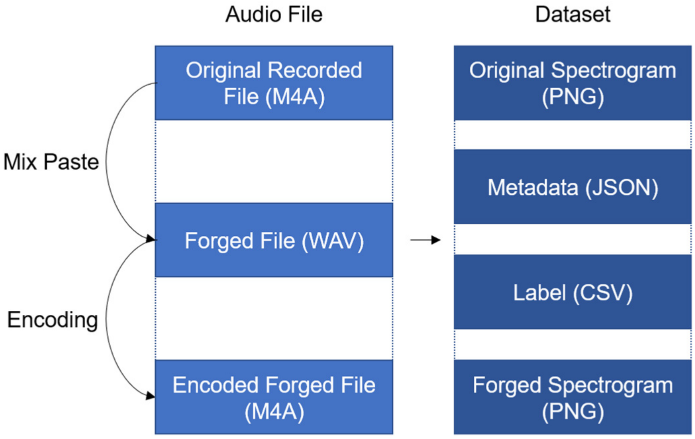


Figure 1. Dataset construction process.

3.1. Voice Recording

To build this dataset, four speakers—two men and two women—were recorded using the following smartphones: Apple iPhone 14 Pro Max, Apple iPhone 13 Mini, Samsung Galaxy Note 20, and Samsung Galaxy S23+; the recording software was the software built into each smartphone: Voice Memo on the iPhone and Voice Recorder on the Galaxy. Files recorded in this software have the same sampling rate of 48,000 Hz. Figure 2 shows the proportion of each recording device. Various feature points were extracted from different Korean pronunciations. Figure 3 shows the distribution of the plain, aspirated, sibilant, and tense Korean consonants in the recording script.

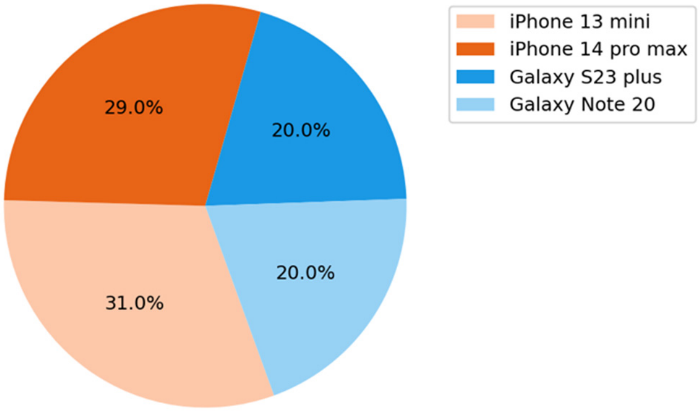


Figure 2. Proportions of the recording devices used in the study.

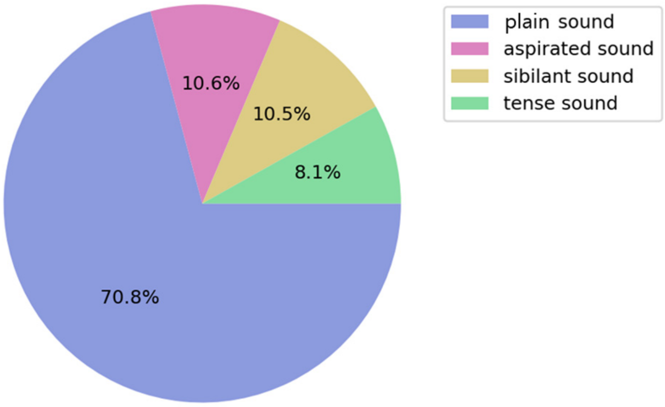
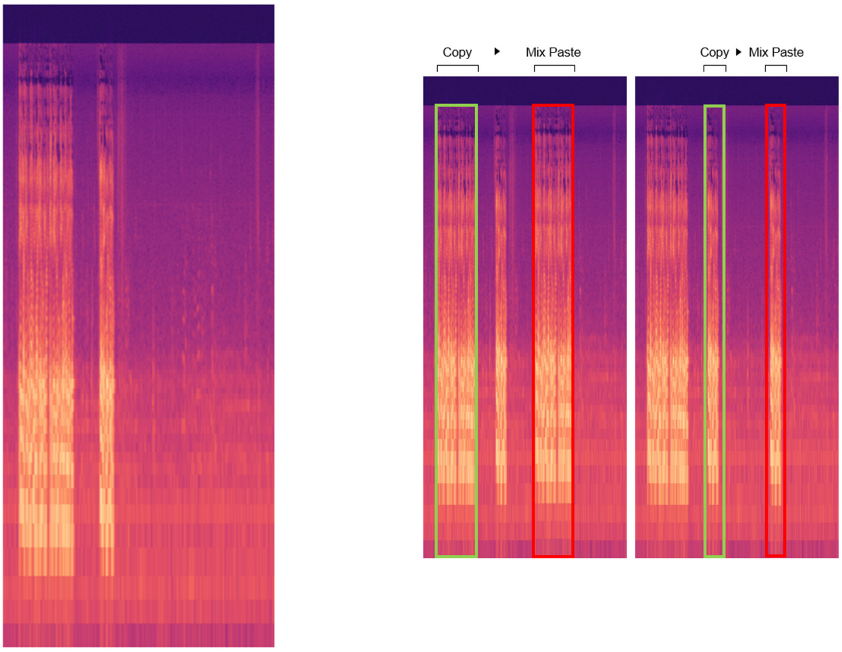


Figure 3. Distribution of the plain, aspirated, sibilant, and tense consonants in the recording script used in the study.

3.2. Voice File Editing

The "Mix Paste" editing function is available under Adobe Audition, which is the most widely used voice editing tool in Korea. One of the speech sections was selected and copied from the original voice file. Then, it was pasted by setting the "Overlab" option in an empty space without speech (Figure 4). We marked the original and pasted regions with "markers" and then saved them in WAV format. At this time, the sample positions of the marker's start time and duration were stored in a WAV file by Adobe's Extensible Metadata Platform (XMP). The sample positions of the marker's start time and duration were extracted from the metadata of these WAV files, and the start time and duration of each of the original and pasted areas were saved in a CSV file. A forged voice file was created by pasting one of the speech sections of the original voice file, and several edited files were created from the original voice file, resulting in the number of forged voice files exceeding that of the original voice files.

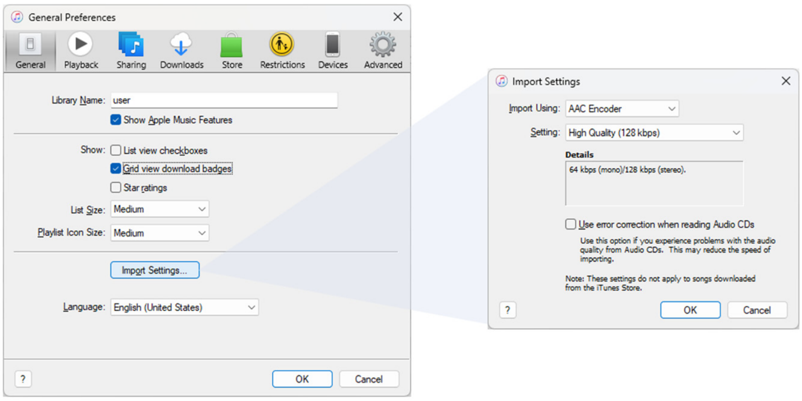


(**a**) (**b**)

Figure 4. Log spectrogram images: (**a**) original and (**b**) forged.

3.3. Encoding Edited Voice Files

The third step was encoding the edited data. The file format, metadata, and structure of the edited data needed to match those of the original [19]. Therefore, the WAV format needed to be encoded in M4A format. The encoder needed to encode the data of the file such that it would contain a file structure and metadata that closely resembled those of the original file. Consider a case where a file recorded on an iPhone was encoded by selecting "Good Quality" in the Advanced Audio Coding encoder provided by iTunes; it had a sample rate of 48,000 Hz, which was the same as that of the original. With the exception of some metadata, the encoded file could be made to almost resemble the original. Moreover, because the metadata and file structure could be changed to resemble those of the original file using the Hex editor, forged content could not be detected simply on the basis of metadata or file structure. Figure 5 illustrates the settings employed for encoding using iTunes.



Library Name: user

☑ Show Apple Music Features

Show: □ List view checkboxes

☑ Grid view download badges

□ Star ratings

List Size: Medium

Playlist Icon Size: Medium

Language: English (United States)

Import Settings

Import Using: AAC Encoder

Setting: High Quality (128 kbps)

Details

64 kbps (mono)/128 kbps (stereo).

□ Use error correction when reading Audio CDs

Use this option if you experience problems with the audio quality from Audio CDs. This may reduce the speed of importing.

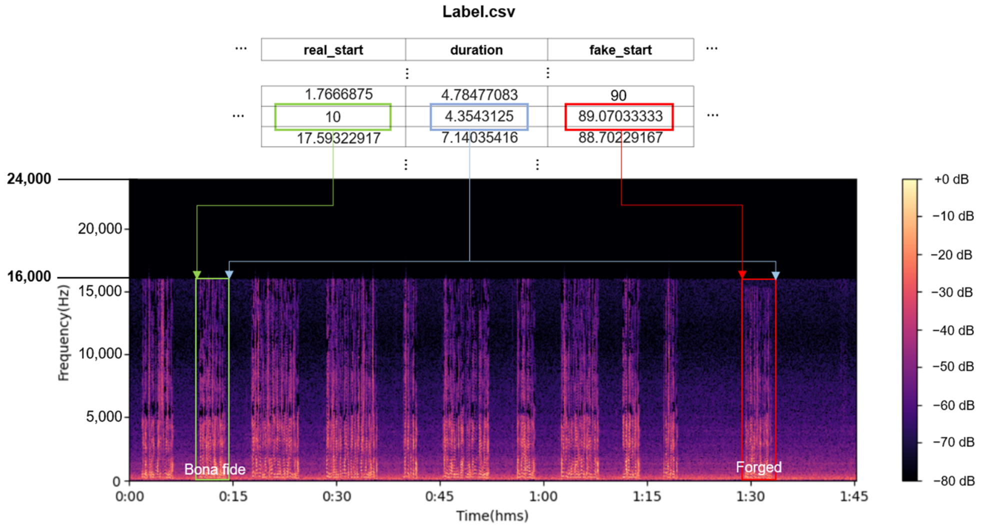
Note: These settings do not apply to songs downloaded from the iTunes Store.

Figure 5. Settings for encoding in iTunes.

Samsung Galaxy smartphones do not require separate encoders. Using the convolutional neural network model, we confirmed that irrespective of the encoder used for the original file, the spectrogram would be unaffected. Therefore, in the case of recorded files on Galaxy smartphones, encoding was performed at the same sample rate as that of the original, namely 44,100 Hz for Galaxy Note 20 and 48,000 Hz for Galaxy S23+, using an online encoding site [20].

3.4. Data Preprocessing

Finally, the original and forged voice files were converted to a spectrogram image using the "librosa" module [21] in Python and saved as a PNG file. The iPhone utilized for this data collection has a frequency bandwidth of approximately 24,000 Hz and a cutoff frequency of approximately 16,000 Hz. On the other hand, Samsung Galaxy smartphones have a frequency bandwidth of approximately 23,000 Hz and a cutoff frequency of approximately 20,000 Hz. By calculating the ratio of this frequency information to the height value of the spectrogram image and the ratio of the time information of the bona fide and forged sections to the width value of the spectrogram image, the bona fide and forged sections could be accurately labeled as a bounding box (Figure 6). As the forged segments were obtained from the same audio file, similar to what is carried out with the copy–move technique, the bona fide and forged segments coexist in a single file.



Label.csv

…	real_start	duration	fake_start	…
	⋮		⋮	
	1.7666875	4.78477083	90	
…	10	4.3543125	89.07033333	…
	17.59322917	7.14035416	88.70229167	
	⋮		⋮	

Figure 6. Bona fide bounding box and forged bounding box on linear spectrogram.

4. Dataset Verification

To prove the applicability of this proposed dataset, we performed validation by building a deep learning model for speaker identification during speech recognition. Speaker identification is commonly performed based on a convolutional neural network (CNN) based on a Mel spectrogram [22]. To do this effectively, transfer learning to use the VGG19 pre-trained CNN was performed [23]. Figure 7 shows the VGG19-based transfer learning model.

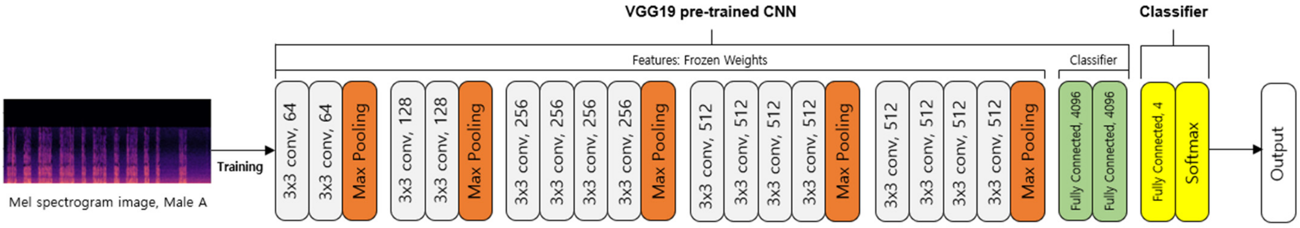


Figure 7. VGG19-based transfer learning model.

The proposed dataset was recorded by four speakers, and this dataset included 1555 Mel spectrogram images from this original audio. Table 2 shows the composition of the dataset according to classes. An experiment was performed by dividing this Mel spectrogram image into a 70% training set, a 10% validation set, and a 20% test set. This transfer learning model was evaluated using accuracy, precision, recall, and F1 scores. Table 3 shows the classification evaluation metrics performed on the test dataset.

Table 2. Composition of the dataset used in the experiment.

Class	Number of Mel Spectrogram Images			Smartphone Model
	Training	Validation	Test	
Male A	304	45	99	iPhone 14 Pro Max
Female A	344	50	91	iPhone 13 Mini
Male B	211	35	65	Galaxy S23+
Female B	229	25	57	Galaxy Note 20
Total	1088	155	312	-

Table 3. Evaluation metrics for speaker identification.

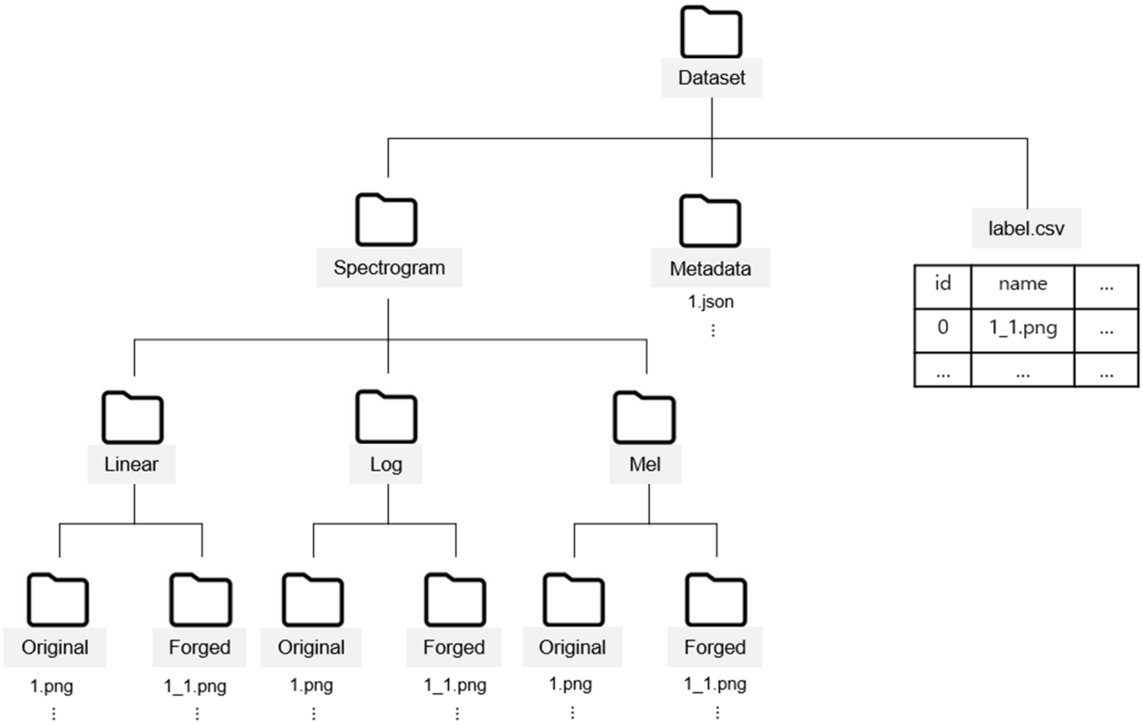
Class	Accuracy	Precision	Recall	F1-Score
Male A	1.00	1.00	1.00	1.00
Female A	0.97	0.96	1.00	0.98
Male B	0.89	0.99	0.92	0.95
Female B	0.91	0.96	0.98	0.97
Overall	0.98	0.98	0.98	0.98

Various experiments may be necessary, but as shown in Table 3, overall performance meets expectations. Therefore, it is judged that this proposed dataset satisfies the qualitative aspects.

5. User Notes

The first level of the dataset consisted of a spectrogram folder, a metadata folder, and a label.csv file, with the spectrogram folder comprising three folders: linear, log, and Mel. Each of these three folders contained original and forged folders, where each original folder contained a spectrogram image of the original audio file, and each forged folder contained a spectrogram image of the forged audio file. The file names in the original and forged folders were related. For example, 80.png in the original folder indicated the original file and the filenames 80_1.png and 80_2.png indicated that the corresponding files had been created by editing the original 80.png file. Figure 8 shows the hierarchical structure of the dataset.

The "Metadata" folder contained the metadata of the audio file and the audio file structure in JSON format, as well as the gender of the speaker, type of recording device, and operating system information collected during the recording process. The file name of each JSON file was the same as that of the original spectrogram image in the spectrogram folder. The label.csv file contained information regarding the forged section of the forged audio file. Regarding the forged section, the "fake_start" and "duration" columns indicated the start time and length of this section. The bona fide section that was used to be forged was marked with a "real_start" column. Because "duration" is the same, only one "duration" column is indicated. Furthermore, the forged section was converted into text using Whisper, an application programming interface developed by OpenAI to handle speech-to-text (STT) tasks [24]; the "text" column indicated the STT results.



id	name	...
0	1_1.png	...
...	...	...

Figure 8. Structure of the proposed dataset.

Author Contributions: Conceptualization, J.W.P.; methodology, J.W.P.; software, Y.S.; formal analysis, Y.S.; investigation, J.W.P.; resources, W.J.K.; data curation, Y.S.; writing—original draft preparation, J.W.P. and Y.S.; writing—review and editing, J.W.P.; visualization, Y.S.; supervision, J.W.P.; project administration, W.J.K.; funding acquisition, J.W.P. and W.J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Seoul R&BD Program (IC220010) through the Seoul Business Agency (SBA), funded by The Seoul Metropolitan Government.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study and future updates are available at https://drive.google.com/drive/folders/10cBCvQTF-XqCfdQuUU4y_ssrbi3hUJkw (accessed on 19 November 2023).

Acknowledgments: We would like to thank SSMM Co., Ltd. for their assistance with the data collection and labelling.

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

References

1. Imran, M.; Ali, Z.; Bakhsh, S.T.; Akram, S. Blind Detection of Copy-Move Forgery in Digital Audio Forensics. *IEEE Access* **2017**, *5*, 12843–12855. [CrossRef]
2. Mcuba, M.; Singh, A.; Ikuesan, R.A.; Venter, H. The Effect of Deep Learning Methods on Deepfake Audio Detection for Digital Investigation. *Procedia Comput. Sci.* **2023**, *219*, 211–219. [CrossRef]
3. Ramos-Castro, D.; Gonzalez-Rodriguez, J.J.; Ortega-Garcia, J. Likelihood Ratio Calibration in a Transparent and Testable Fo-rensic Speaker Recognition Framework. In Proceedings of the IEEE Odyssey—The Speaker and Language Recognition Workshop, San Juan, PR, USA, 28–30 June 2006. [CrossRef]
4. Bevinamarad, P.R.; Shirldonkar, M.S. Audio Forgery Detection Techniques: Present and Past Review. In Proceedings of the Fourth International Conference on Trends in Electronics and Informatics, Tirunelveli, India, 15–17 June 2020. [CrossRef]
5. Ustubioglu, A.; Ustubioglu, B.; Ulutas, G. Mel Spectrogram-Based Audio Forgery Detection Using CNN. *Signal Image Video Process.* **2023**, *17*, 2211–2219. [CrossRef]

6. Huang, X.; Liu, Z.; Lu, W.; Liu, H.; Xiang, S. Fast and Effective Copy-Move Detection of Digital Audio Based on Auto Segment. *Int. J. Digit. Crime Forensics* **2019**, *11*, 127–142. [CrossRef]
7. Jago, M. *Adobe Audition CC Classroom in a Book*, 2nd ed.; Adobe Press: San Jose, CA, USA, 2013; pp. 75–76.
8. Chuchra, A.; Kaur, M.; Gupta, S. A Deep Learning Approach for Splicing Detection in Digital Audios. In Proceedings of the 2nd Congress on Intelligent Systems, New Delhi, India, 4–5 September 2021. [CrossRef]
9. Jadhav, S.; Patole, R.; Rege, P. Audio Splicing Detection using Convolutional Neural Network. In Proceedings of the International Conference on Computing, Communication and Networking Technologies, Kanpur, India, 6–8 July 2019. [CrossRef]
10. Ustubioglu, B.; Tahaoglu, G.; Ulutas, G. Detection of Audio Copy-Move-Forgery with Novel Feature Matching on Mel Spectrogram. *Expert Syst. Appl.* **2023**, *213*, 118963. [CrossRef]
11. Kang, Y.; Kim, W.; Lim, S.; Kim, H.; Seo, H. DeepDetection: Privacy-Enhanced Deep Voice Detection and User Authentication for Preventing Voice Phishing. *Appl. Sci.* **2022**, *12*, 11109. [CrossRef]
12. Khochare, J.; Joshi, C.; Yenarkar, B.; Suratkar, S.; Kazi, F. A Deep Learning Framework for Audio Deepfake Detection. *Arab. J. Sci. Eng.* **2021**, *47*, 3447–3458. [CrossRef]
13. Zhang, Z.; Yi, X.; Zhao, X. Fake Speech Detection Using Residual Network with Transformer Encoder. In Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security, Bruxelles, Belgium, 22–25 June 2021. [CrossRef]
14. ASVspoof. Available online: https://www.asvspoof.org/ (accessed on 16 October 2023).
15. WaveFake. Available online: https://paperswithcode.com/dataset/wavefake/ (accessed on 16 October 2023).
16. 'In-the-Wild' Audio Deepfake Data. Available online: https://deepfake-demo.aisec.fraunhofer.de/in_the_wild/ (accessed on 16 October 2023).
17. Liu, X.; Wang, X.; Sahidullah, M.; Patino, J.; Delgado, H.; Kinnunen, T.; Todisco, M.; Yamagishi, J.; Evans, N.; Nautsch, A.; et al. Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2023**, *31*, 2507–2522. [CrossRef]
18. Hsu, H.P.; Chang, S.C.; Hung, C.H.; Wang, S.S.; Fang, S.H. Performance Comparison of Audio Tampering Detection Using Different Datasets. In Proceedings of the 24th IEEE International Conference on Mobile Data Management, Singapore, 3–6 July 2023. [CrossRef]
19. Park, J.W.; Kawk, W.J.; Lee, S. A Study on Forgery Techniques of Smartphone Voice Recording File Structure and Metadata. *J. Converg. Cult. Technol.* **2022**, *8*, 807–812.
20. Audio Tool Set. Available online: https://audiotoolset.com/ko/wav-to-m4a/ (accessed on 17 October 2023).
21. Librosa. Available online: https://librosa.org/ (accessed on 17 October 2023).
22. Ye, F.; Yang, J. A Deep Neural Network Model for Speaker Identification. *Appl. Sci.* **2021**, *11*, 3603. [CrossRef]
23. Suppakitjanusant, P.; Sungkanuparph, S.; Wongsinin, T.; Virapongsiri, S.; Kasemkosin, N.; Chailurkit, L.; Ongphiphadhanakul, B. Identifying individuals with recent COVID-19 through voice classification using deep learning. *Sci. Rep.* **2021**, *11*, 19149. [CrossRef]
24. Whisper. Available online: https://openai.com/research/whisper/ (accessed on 17 October 2023).

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.