Auxiliary Attention Pooling Network-Based Recording Device Detection System

A project report submitted in partial fulfillment of the requirements for B.Tech. Project

B.Tech.

by

Devansh Chowdhury (2019IMG-016)

विश्वजीवनामृतं ज्ञानम्

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT GWALIOR-474 010

2022

CANDIDATES DECLARATION

I hereby authenticate that the work being presented in this report **Auxiliary attention pooling network-based recording device detection system**, in partisan fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** is an authentic record of my research work conducted during the period *June 2022* to *September 2022* under the supervision of **Dr.Vinal Patel**. All the imported references in the figure, papers and tables have been cited.

Date: Signatures of the Candidates

This is to certify that the above declaration is true to the best of my knowledge.

Date: Signatures of the Research Supervisor

ABSTRACT

All of our electronic devices have some sort of microphone source installed in it. The microphone is used to record and transmit audio signals. Every sound we hear has been recorded on some microphone or the other. Due to an extensive range of different audio recording devices, it has become difficult to identify them. Microphones vary greatly in shape, size, component structure, and most importantly the unique noise that is added to the audio whenever they are used for recording. A solution to this problem can help improve speech-to-text conversion, Audio forensics analysis, check the authenticity of audio files in a court of law, and pinpoint the source of audio leaks. We explore the idea of using an attention-based network combined with Convolution overloaded with an additional auxiliary task.

*Index Terms:-*Audio forensics, mechanical recording devices, Auxiliary Output, Attention, variance vs bias, Noise

ACKNOWLEDGEMENTS

I am grateful to **Dr.Vinal Patel** to have the opportunity to conduct my **Bachelor of Technology project** under their guidance. Their ideas and insights have been a boon throughout my journey. Their constant guidance and constant support were crucial for the whole success of this research project. Their expertise in the area of research made this work enjoyable and interesting. My supervisor was always ready to help me in case of any queries and spared no effort to provide me with appropriate material and guidance. Their contribution is sincerely appreciated and I am indebted to them for all the help.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ReLu	Rectified linear unit
CNN	Convolutional neural network
LSTM	Long short-term memory
MFCC	Mel-frequency cepstral coefficients
K-SVD	K-singular value decomposition
MLP	Multilayer perceptron
GMM-UBM	Gaussian Mixture Model - Universal Background Model
DFT	Discrete Fourier transform
IWSLT	International Conference on Spoken Language Translation

CHAPTER 1

Introduction and Literature Review

This chapter includes the introduction, motivation and literature review of the related work and the objective of this thesis.

1.1 Introduction

Identification of the audio device is the most important piece of information that could be extracted from an audio file [7] . In theory, each audio device is unique based on the number of parts, their quality, and their architecture, and every small detail has a contribution to making that device unique. This makes it possible to classify devices using the audio file recorded on that system. But in practice to identify these small irregularities and inaccuracy is a difficult task. The quality of audio microphones is increasing day by day. With the new advancements in science and technology, it is almost unreal to see how clear and crisp the audio recorder is by microphones. It is impossible for a mere human to even try to differentiate microphones based on their recorded audio file. This advancement in microphone technology makes our task of classifying audio based on their source microphone very difficult [8]. The only way to accurately predict the microphone is to identify what types of noises and imperfections they part to the sound when they are recorded. Also, there is constant research going on the opposite side on improving the quality of Deepfakes which aims at making changes in forging the audio files and still look as authentic as the real ones [17].

1.2 Problem/Motivation

The ability to correctly identify the source microphone based on the audio file can open doors to multiple segments of real-life applications. It could be used in audio forensics [11] to check the credibility of an audio signal to make the system more robust to forged or tampered audio files. They could also be applied to detect audio bots [13]

.Contribute to Improvement in the field of speech-to-text conversion [4]. The quality of the recorded audio signal can also be improved by removing the unique noise. In the future, this work could be expanded to distinguish among phones of the same model but with different serial numbers.

1.3 Literature Review

Several Studies have been conducted throughout the decade to identify a microphone source from the sound it recorded. Past research has hinted that all mechanical devices have certain inaccuracies or slight irregularities and exploitation of this shortcoming is the best path forward. To add to this uphill battle no sizeable audio corpus has yet been officially open-sourced.As discussed in [12] Deep learning approaches have fared far better in this area but they are still not performing at their potential due to limited data bringing us back to the old-age tradeoff between variance and bias [5]. The most common wall here is the problem of a very simple corpus. Our system aims to classify audio samples to the audio devices they were recorded on. The simplest workflow to acquire this information is to process the audio files, extract features from them and then classify them based on feature of their respective class. There have been several attempts in this area. As discussed in [9], proposed that to fully focus on noise imparted by the audio device is to carefully extract particular segments of silence as these would have the least influence on speech signals and help the model to classify audio based solely on the noise signals.As discussed in [18], worked on the problem of identifying if two audio files were from the same device or not. They used MFCC features and K-SVD algorithm for the identification along with the KISS metric.

As discussed in [14], proposed that an audio signal is made up of two signal noise signals and a speech signal. Although the speech signal is generally accepted as the feature to move forward with. So multiple techniques have been used aimed at removing noise from the audio sample to make the speech clearer. But noise signals can be viewed as a fingerprint of an audio device as these are unique to each device.The noise traces if and can be isolated would be incredibly useful for audio device identification. They removed the original speech from the audio and extracted the Fourier coefficient histogram of the signal as the feature vector, which has a powerful descriptive capability for audio signals, Then multiple models were trained upon this data like MLP, CNN, and Softmax Regression model. These were then averaged. Also to avoid the penalty of having one model being a misfit,voting was used.As discussed in [1], they concentrated on the preprocessing aspect of the data. Rather than cleaning the data and removing the noise they used alternately approached and focused on eliminating the speech from the audio. The idea is that this noise that is introduced while recording is unique and attempted to identify this transfer function. Features were extracted using MFCC. Then using Kmeans clustering to segregate all points to the k devices. As discussed in[6], they proposed a system to improve the efficiency of MFCC parameters. As MFCC is the most common and widely used preprocessing algorithm to retrieve features from audio files. More interest was poured into frequency domain features over time domain features, as it has been observed that frequency domain features have performed better. Audio recordings are split into frames of 20-30 ms and MFCC coefficients are used as features. The database used was the same as ours the [10] mobiphone dataset. They were able to achieve a 66% accuracy.They used the GMM-UBM model for the classification part. As discussed in [15], their work proposed using an alternate attention layer for audio recording classifications. They also used filter bank features to extract information from the audio files and then pass it to the classification model. They also experimented using the data of whispered tone speech which was self-recorded and developed by them. These helped in understanding the practicality of real-life situations. They were able to achieve 84% accuracy on the [10] Mobiphone dataset.

As discussed in [16], their work was also aimed at using a distinctive specific signature that an audio device gives to an audio signal while recording. Using this distinctiveness along with a CNN classification model which would learn this difference was their proposed plan. They used DFT for frequency domain representation of the audio signal. The model was made using multiple simple Conv 1D layers followed by multiple dense layers to give the final output. As researched in [3], recently attention implementation using transformer in IWSLT 2022 has given recommendable results on speech-to-text conversions,proving the ability of attention to correctly process the sequential information present in the audio signal. As researched in [2],their work was aimed at identifying the phone model used to record a video. They made two different detectors both based on CNN which jointly exploited the audio and visual information present in the frames of the video and analyzed them. The first one applied a voting system to detect the best features of two CNN-based models. The second detector made the combined decision and was given both models' outputs. It is performed by jointly analyzing video and audio data. They successfully showed that using different detectors/models gave a huge advantage over just a single model.

1.4 Objective

Recorded audio can be of any type - mono or stereo, indefinite period, recorded on any microphone. Our objective here is to make a system/pipeline workflow that can accurately identify the identity of the source microphone from the recorded audio sample. The paper focuses more on the modeling and training side of the problem to improve the quality of research. We will also be predicting the gender of the user through the voice in the audio sample.

1.5 Research Gaps

The main limitation that could be found in all the above approaches was that the system was always able to perform well on the training part but was not able to generalize it to new test samples. Also, there was a huge problem with having open-sourced good-quality audio data.Mobiphone [10] dataset only had 24 audio files for each device which was too low to use any model up to its maximum efficiency. Weighing in all the factors two decisions were set in stone, first one was the use of dual implementation of CNN-based and attention-based models,the second wall was how to overcome the overfitting encountered by other researchers. We proposed a supervised auxiliary attention pooling and CNN-based neural network.

CHAPTER 2

Methodology

This section introduces the hypothesis and the analytical validation of the proposed solution.

2.1 Proposed hypothesis

In source microphone detection through recorded audio samples, we will be given a recorded audio sample and we want to identify on which mobile device was it recorded. This problem falls under the supervised audio classification.As show in Figure 2.1, Our model is based on CNN and LSTM-based classifier network. We extract useful features present in the audio file through log-filter bank. The features are passed as input to both networks. Attention-based LSTM computes useful features from the sequential input. The output from both these layers is then multiplied and passed onto a classification network to make accurate predictions about the class of audio device and gender of the user in the audio device.

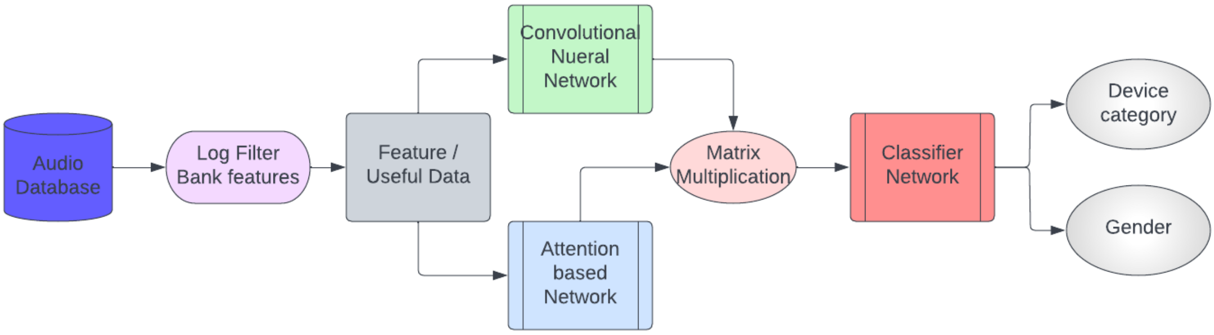


Figure 2.1: Proposed model workflow consists of three blocks - CNN, LSTM and Classifier Network.

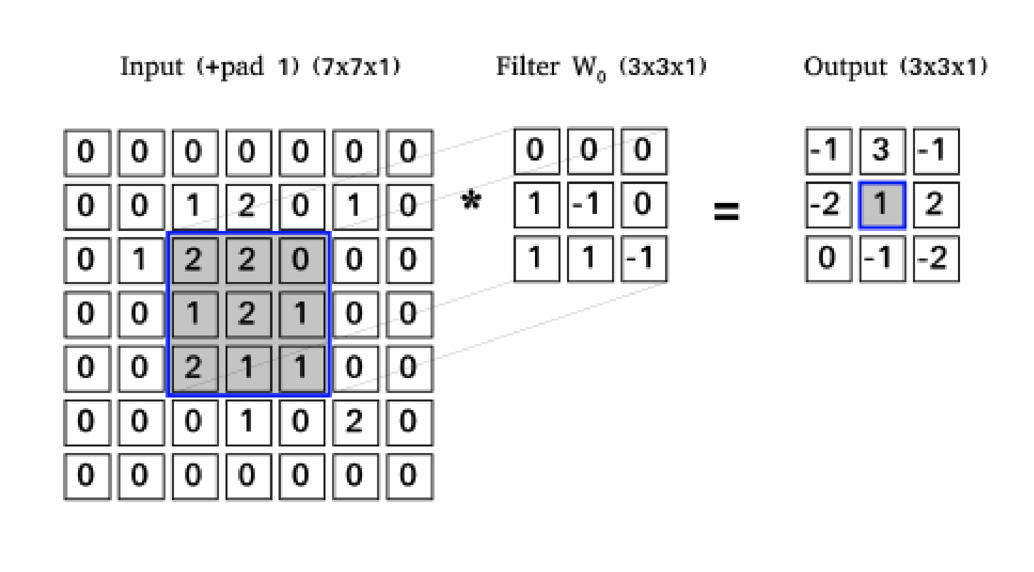
2.2 Novelty

The novel idea here is to avoid overfitting we will be tasking our model to predict the gender of the user also. No other research paper has used this important metadata information to increment the model performance.This way our system would be multitasking which would make it difficult to start overfitting the dataset which was the main problem in all the research papers as the open-sourced dataset for such a problem is very small in size.

2.3 Modelling Layers

2.3.1 Convolutional Neural Network

These are neural networks that are specialized in the Image format data, which is resembled MFCC graphs. These are formed using blocks that consist of Convolutional layers, Pooling layers followed by batch normalization. A convolutional layer does the main processing of using filters/kernels and overlapping it all over the data to generate feature maps as shown in Figure 2.2. Using multiple of these gives us features of depth that are equal to the number of filters we used.Pooling is used to minimize the area of interest for the next step. We have used Average pooling, followed by batch normalization which normalizes the outputs to regulate overfitting.



Input (+pad 1) (7x7x1)

0	0	0	0	0	0	0
0	0	1	2	0	1	0
0	1	2	2	0	0	0
0	0	1	2	1	0	0
0	0	2	1	1	0	0
0	0	0	1	0	2	0
0	0	0	0	0	0	0

*

Filter W_0 (3x3x1)

0	0	0
1	-1	0
1	1	-1

=

Output (3x3x1)

-1	3	-1
-2	1	2
0	-1	-2

Figure 2.2: Overview of Convolutional Operation in a Convolutional layer.

2.3.2 Long short-term memory Network

LSTM cells can handle sequential data better than any other network. This helps us to capture the sequential information hidden in the log-filter bank features. The first input of sequence is passed to the cell with gives an output, this along with the second input of sequence is passed again to the cell. This keeps on repeating till all the elements of the sequence have been processed as shown in Figure 2.3.

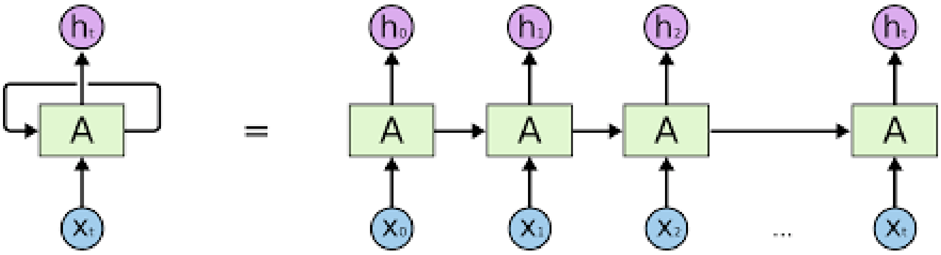


Figure 2.3: Work flow of LSTM layer when Sequential data has been provided.

2.3.3 Classifier Network

A Classifier network is a simple network of dense layers with a ReLu activation function as shown in Figure 2.4. Just the last layer uses the acivation Softmax to predict the final class.

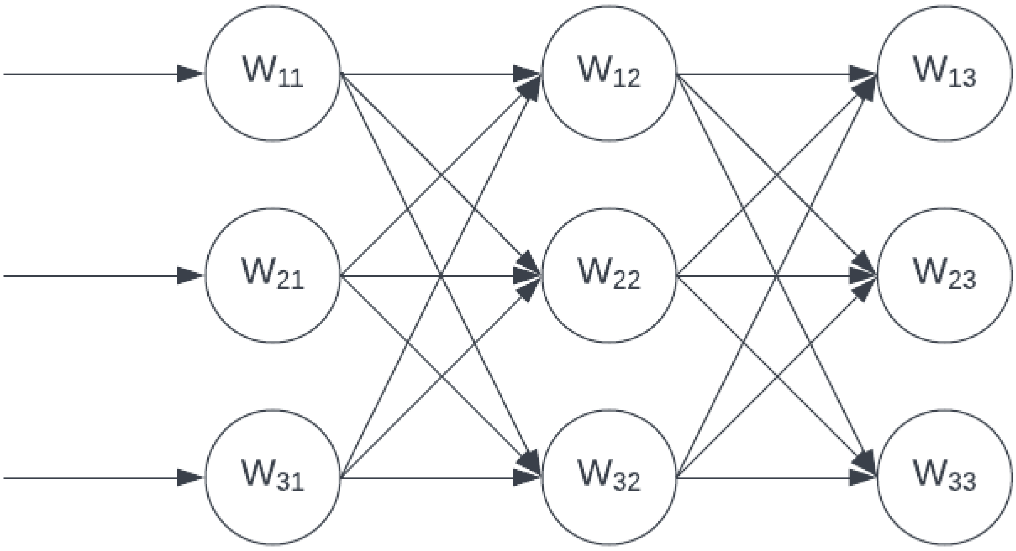


Figure 2.4: Simple Classifier Network made up of multiple Dense layer where each cell output is passed through ReLu.

2.4 Dataset

We have used the public Mobiphone dataset comprises 21 different mobile devices with 24 audio files each comprising 12 males and 12 females [10]. The different mobile device manufacturers are Apple, HTC, LG, Nokia, Samsung, Sony, and Vodafone. This is the most popular open-sourced common data used to set benchmark results. We would be computing 24 log filter bank features. The audio file is filtered first for pre-emphasis which boosts the high-frequency components while ignoring the low-frequency components of the signal. The signal is divided into overlapping frames. The power spectrum is then calculated for each frame. The sum along frames gives the energy of the signal for the frame. We also compute the 24 Mel - filter bank coefficients. And multiply it with the frame energy we calculated earlier. This is our filter bank feature and then we do a simple log over the values hence the replacement of zero in the previous step. These features are then normalized.

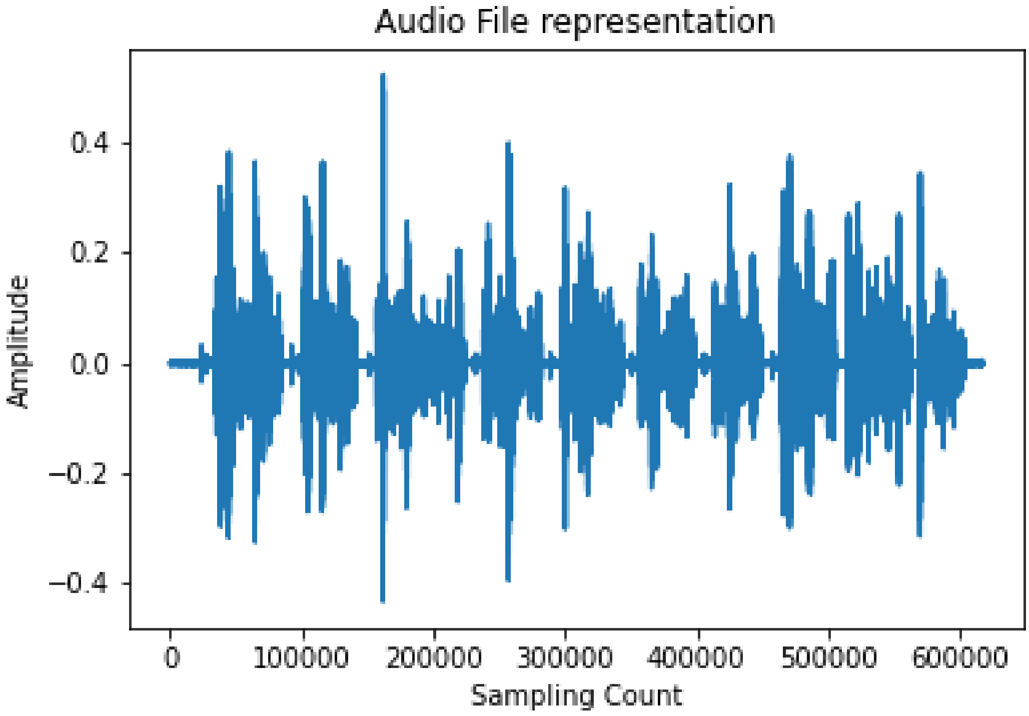


Figure 2.5: Amplitude Representation of a randomly selected audio file having 16Khz sampling rate.

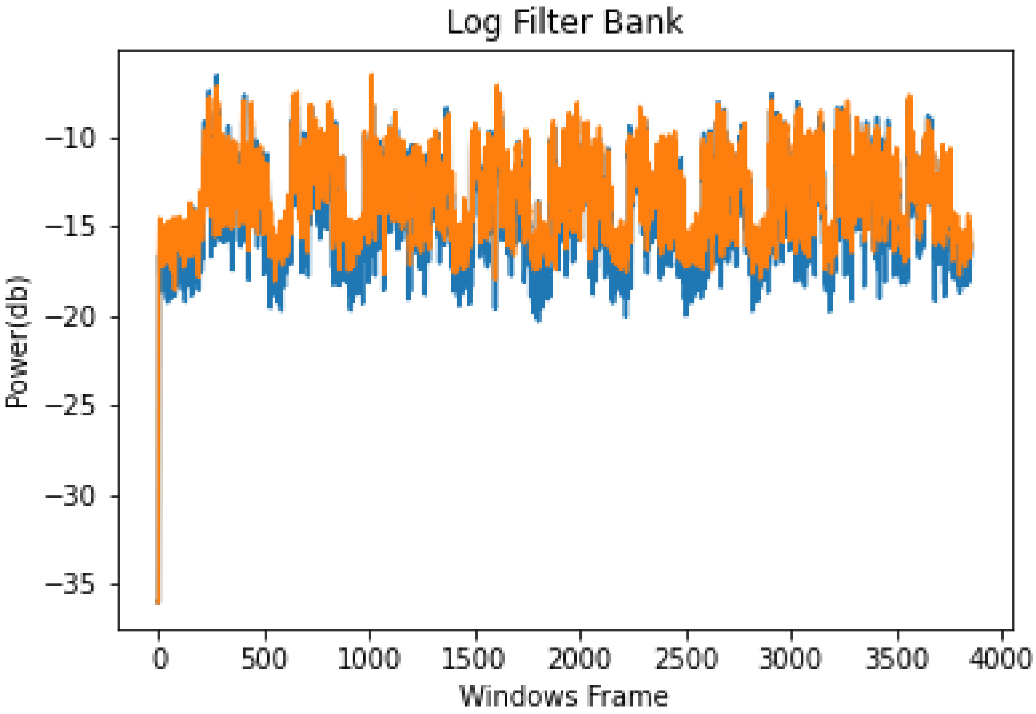


Figure 2.6: Log Filter bank representation of a randomly selected audio file.

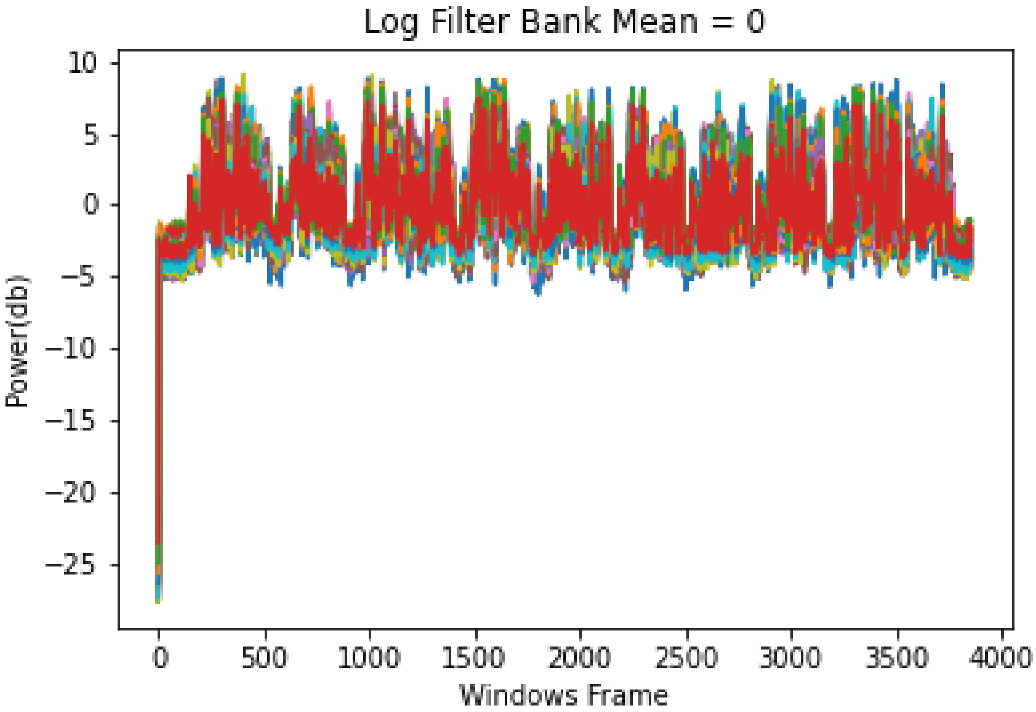


Figure 2.7: The Signal mean is shifted to the value zero.

2.5 Model Architecture

The model architecture comprises the feature being extracted from the audio file and are passed as inputs to both CNN and LSTM-based Networks. The CNN network as shown in Table 2.1 is made up of 3 blocks consisting of a 2D convolutional layer, batch normalization followed by average pooling. The LSTM network as shown in Table 2.2 is made up of two simple LSTM layers followed by average pooling. These two outputs are multiplied with each other and applying average pooling again. The result as shown in Table 2.3 is passed onto the Classifier network to predict the mobile device category and gender of the user.

Table 2.1: Implementation details of proposed Convolutional Architecture

No.	Layer	Filters/Pooling
1	Conv 2D 3x3, ReLu	20
2	Batch Normalizaton	–
3	Average Pooling 2D ReLu	(2,1)
4	Conv 2D 3x3, ReLu	20
5	Batch Normalizaton	–
6	Average Pooling 2D ReLu	(1,2)
7	Conv 2D 3x3, ReLu	20
8	Batch Normalizaton	–
9	Average Pooling 2D ReLu	(2,1)

Table 2.2: Implementation details of proposed Attention Architecture

Layer	Filters/Pooling
LSTM, ReLu	24
LSTM	1
Activation, Sigmoid	–
Average Pooling 1D ReLu	–

Table 2.3: Implementation details of proposed Classifier Architecture

Layer	**Filters/Pooling**
Global Average Pooling 1D	24
Dense, ReLu	30
Dense, Softmax, (Device Classification)	21
Dense, Sigmoid, (Gender Classification)	1

.

CHAPTER 3

Experiments and results

This section discusses the various experiments conducted on the proposed hypothesis and their findings.

3.1 Experimental Setup

We have divided it into three folds. For each of the threefold, we would be performing fit using the first fold for training, the second for validation, and the last for test accuracy. This would give us a total of 6 different combinations. The metric was logged on wandb for better visualization. All folds ran for the 30 epochs and only the best validation model was saved. The loss weightage was 60% for our main task of prediction of mobile device and 40% was for the prediction of gender of the user. All experiments were run for 30 Epochs and the model with the best validation accuracy was saved.KFOLD(0,1,2) indicates that the model was trained on the 0th fold, validated at the 1st fold and test accuracy is predicted from the 2nd fold. The graphs are made by merging the values for all the six folds and taking mean. This helps us to understand the overall performance and convergence of our model better.

3.1.1 Experiment 1

This is our Baseline model made up of Convolutional and Attention network. We would be comparing this experiment with other and seeking out to find any type of improvement.Three-fold Cross validation was made on the category of Audio device.This acted as a baseline model to our novel approach in order to measure how much can we improve upon. The model was trained to just focus on the device category.The dataset was used without any prior modifications.

3.1.1.1 Results

We can see in Figure 3.1 that baseline model achieved 73% accuracy on the validation folds.It took 26-27 epoch to achieve the result.The averaged accuracy was 70% on the test fold.

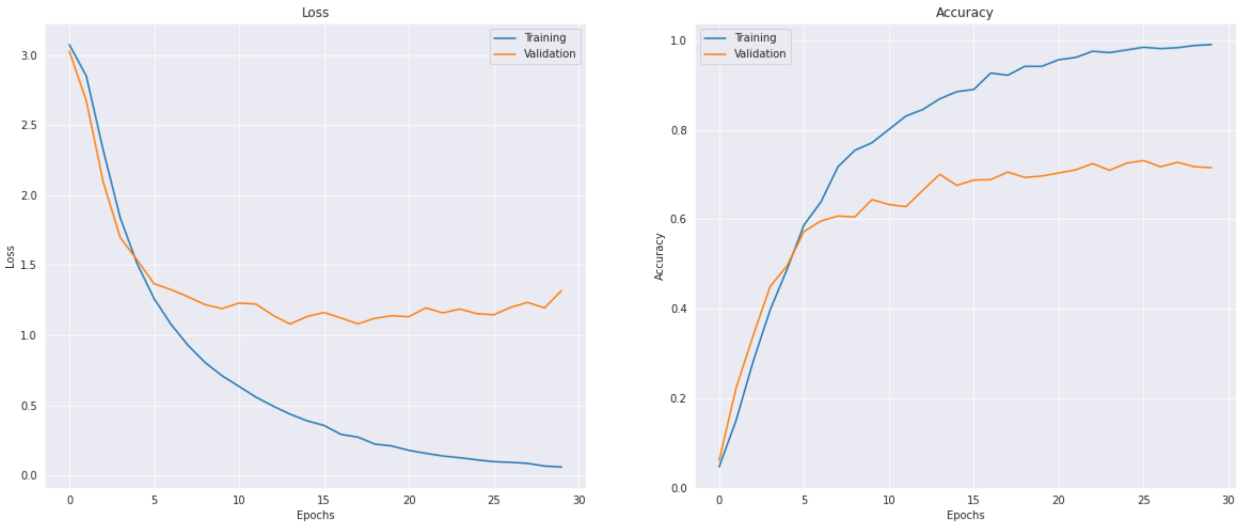


Figure 3.1: Training & Validation Loss & Accuracy Curve

3.1.2 Experiment 2

Making cross-validation based on user class which showed a very poor performance in validation accuracy as well as test accuracy. It was obvious that the Cross-validation strategy had to be applied to the device category.

3.1.2.1 Results

The model achieved 68% accuracy on the validation folds as seen in Figure 3.2. It took about all 30 epoch to achieve the result. The accuracy was 64% on the test fold. The model performed very poorly and showed a significant drop in performance by 5%.

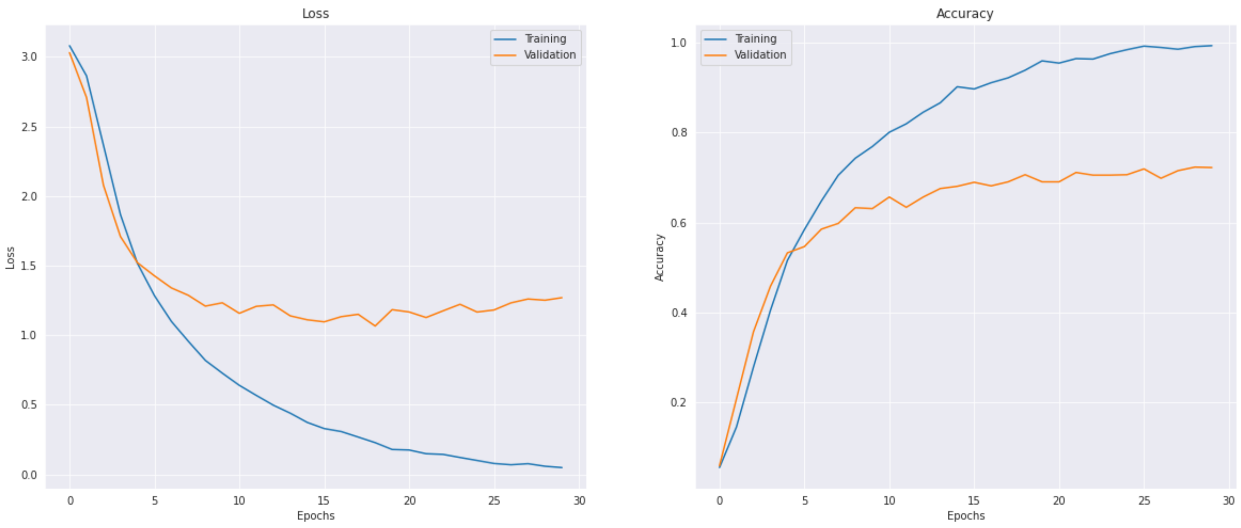


Figure 3.2: Training & Validation Loss & Accuracy Curve

3.1.3 Experiment 3

Changing the Model architecture to Auxiliary output. We also added novelty of multi-tasking to our model.

3.1.3.1 Results

Figure 3.3 show us that the model achieved 76% accuracy on the validation folds.It took just 18 epoch to reach the optimal performance. The system also achieved 77% on the test fold. We can see that change in architecture helped our model to converge faster and better.

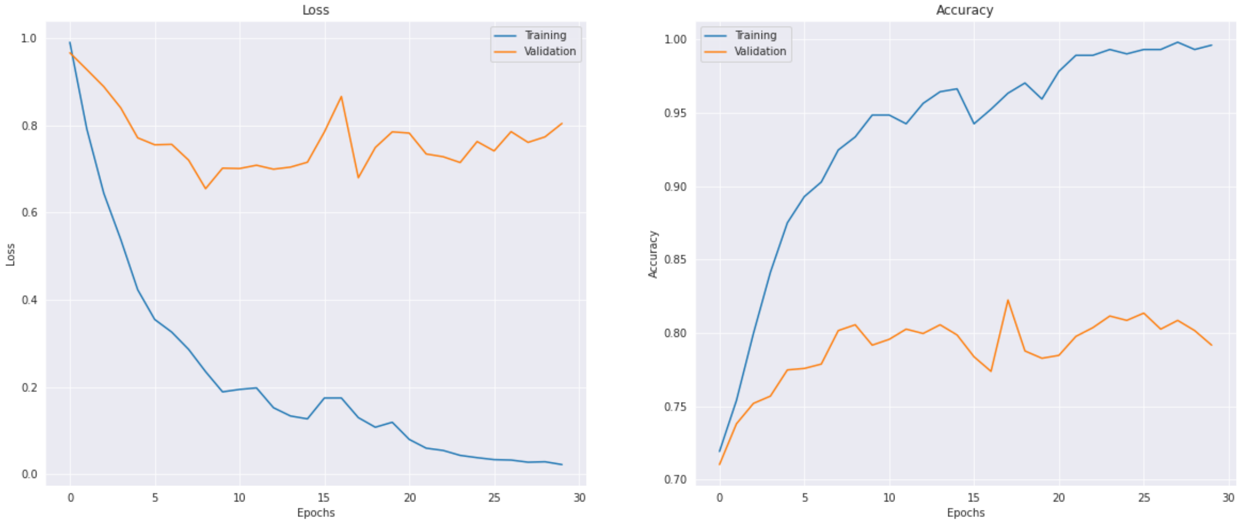


Figure 3.3: Training & Validation Loss & Accuracy Curve

3.1.4 Experiment 4

Introducing the concept of white noise to the audio sample to make the dataset more complex and model more robust.

3.1.4.1 Results

Augmenting the audio signal with white noise made the signal a bit more complex and difficult to understand.As seen in the Figure 3.4 we achieved 77% accuracy on Validation fold and gave 4% boost to our best test fold score until now. It took whole 30 epoch to reach the optimal accuracy.

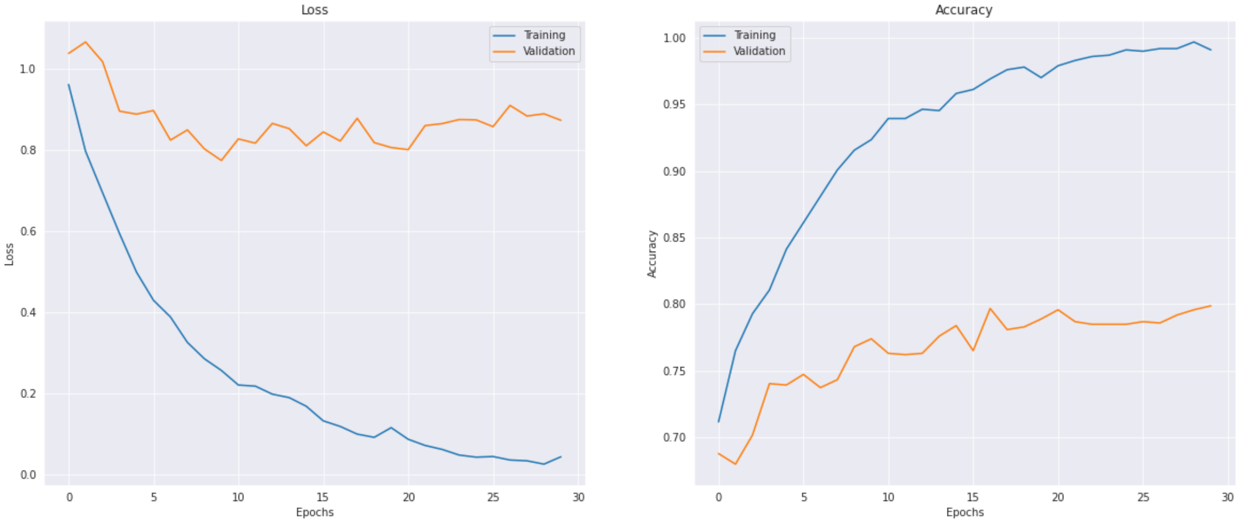


Figure 3.4: Training & Validation Loss & Accuracy Curve

3.1.5 Experiment 5

3.1.5.1 Experiment description

Changing kernel size of the Convolutional layers to a incremental form of (3,3) , (5,5) and (7,7).

3.1.5.2 Results

The model achieved 83% accuracy on validation folds as seen in Figure 3.5. It took 24 epoch to reach the optimal result. We achieved 81% accuracy on the test fold.We can say that this hyperparameter tuning of kernel size didn't yield any improvement.

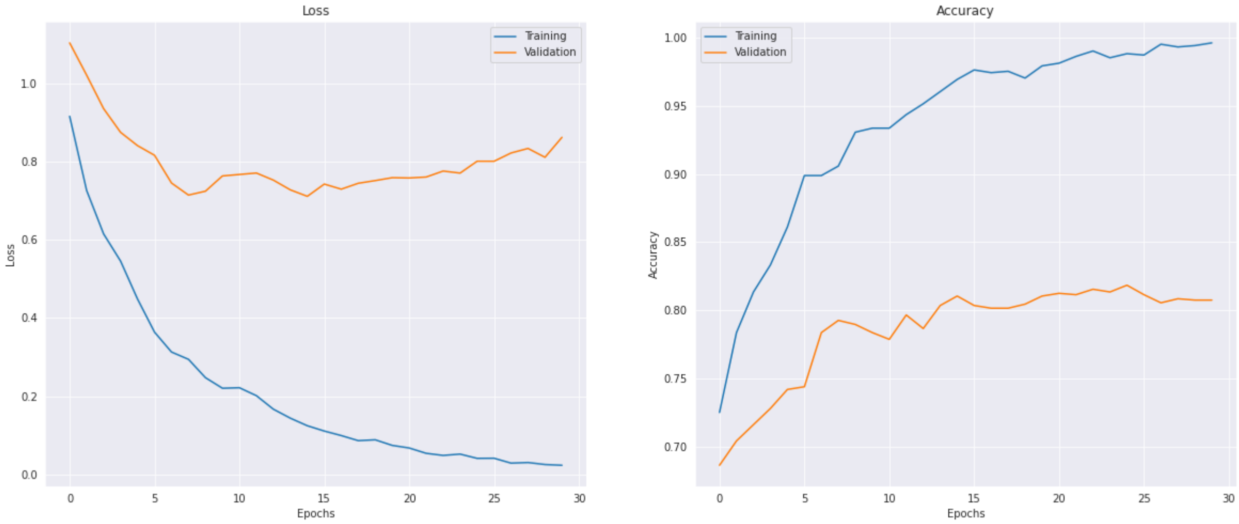


Figure 3.5: Training & Validation Loss & Accuracy Curve

3.1.6 Experiment 6

Tweaking the loss weightage proportion between audio device classification and the gender of the user and setting it to 70:30.

3.1.6.1 Results

From Figure 3.6 we can infer that model achieved 82% accuracy on validation folds. It took all 30 epoch to reach the optimal result. We achieved 80% accuracy on the test fold. We can conclude from this that hyper-parameter tuning didn't yield any significant improvement.

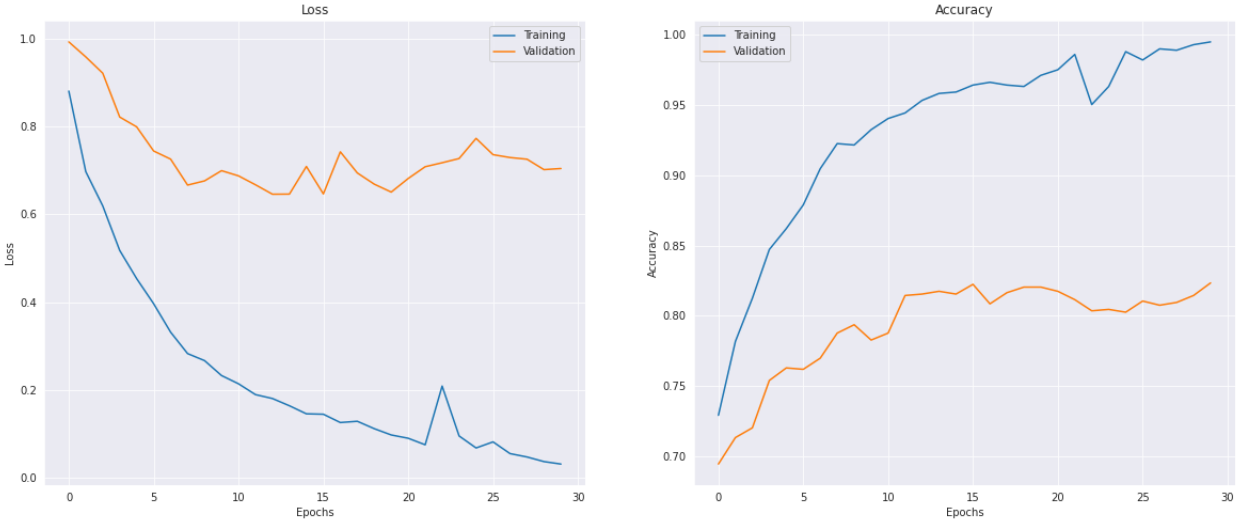


Figure 3.6: Training & Validation Loss & Accuracy Curve

3.2 Experiment Comparison

Table 3.1: All Experiments Result Comparison on basis of Validation & Test Accuracy

Model	**Validation**	**Test**
Simple CNN + LSTM based network	73.4%	70%
Speaker based Cross Validation	68.6%	64.8%
Auxiliary Model Architecture	75.9%	77.7%
White Noise Augmentation	77.38%	81.1%
Kernel size optimization	82.89%	81.01%
Loss Weightage Optimization	82.6%	80.2%
Best 3 out of 6 Model	87.5%	86.9%

Cross-validation performed better on the device category. Auxilary model architecture also gave a boost to our accuracy score. White noise addition also helps us push the score to 80%+. Tweaking kernel size and loss weightage proportion didn't yield any improvement. We then used the best three models out of the six-fold models which gave us 86.9% accuracy.

CHAPTER 4

Conclusion

We have successfully implemented a Multitasking based Auxiliary Attention Pooling Network for recording device classification. We used the top three models out of six that had the highest validation accuracy and were able to successfully train a system that gave an 86.9% accuracy on test data. We achieved a better accuracy score higher than any other published results by 2.9%. In the future, this could be used to improve the efficiency of speech-to-text conversion, individual smartphone device identification, etc.

REFERENCES

[1] Rachit Aggarwal, Shivam Singh, Amulya Kumar Roul, and Nitin Khanna. Cellphone identification using noise estimates from recorded audio. In *2014 International Conference on Communication and Signal Processing*, pages 1218–1222, 2014.

[2] Davide Cortivo, Sara Mandelli, Paolo Bestagini, and Stefano Tubaro. Cnn-based multi-modal camera model identification on video sequences. *Journal of Imaging*, 7:135, 08 2021.

[3] Ryo Fukuda, Yuka Ko, Yasumasa Kano, Kosuke Doi, Hirotaka Tokuyama, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. NAIST simultaneous speech-to-text translation system for IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 286–292, Dublin, Ireland (in-person and online), May 2022. Association for Computational Linguistics.

[4] S. Furui, T. Kikuchi, Y. Shinnaka, and C. Hori. Speech-to-text and speech-to-speech summarization of spontaneous speech. *IEEE Transactions on Speech and Audio Processing*, 12(4):401–408, 2004.

[5] Stuart Geman, Elie Bienenstock, and René Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, 01 1992.

[6] Vishal A. Hadoltikar, Varsha R. Ratnaparkhe, and Rajesh Kumar. Optimization of mfcc parameters for mobile phone recognition from audio recordings. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 777–780, 2019.

[7] Cemal Hanilci, Figen Ertas, Tuncay Ertas, and Ömer Eskidere. Recognition of brand and models of cell-phones from recorded speech signals. volume 7, pages 625–634, 2012.

[8] Cemal Hanilçi and Tomi Kinnunen. Source cell-phone recognition from recorded speech using non-speech segments. *Digital Signal Processing*, 35:75–85, 2014.

[9] Chao Jin, Rangding Wang, Diqun Yan, Biaoli Tao, Yanan Chen, and Anshan Pei. Source cell-phone identification using spectral features of device self-noise. volume 10082, pages 29–45, 02 2017.

[10] Constantine Kotropoulos and Stamatios Samaras. Mobile phone identification using recorded speech signals. In *2014 19th International Conference on Digital Signal Processing*, pages 586–591, 2014.

[11] Christian Kraetzer, Andrea Oermann, Jana Dittmann, and Andreas Lang. Digital audio forensics: A first practical evaluation on microphone and environment classification. In *Proceedings of the 9th Workshop on Multimedia Security*, MM-Sec '07, page 63–74, New York, NY, USA, 2007. Association for Computing Machinery.

[12] Honglak Lee, Peter Pham, Yan Largman, and Andrew Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

[13] A. Lieto, D. Moro, F. Devoti, C. Parera, V. Lipari, P. Bestagini, and S. Tubaro. "hello? who am i talking to?" a shallow cnn approach for human vs. bot speech classification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2577–2581, 2019.

[14] Simeng Qi and Zheng Huang. Identification of audio recording devices from background noise. volume abs/1602.05682, 2016.

[15] Bhavuk Singhal, Abinay Reddy Naini, and Prasanta Kumar Ghosh. wspire: A parallel multi-device corpus in neutral and whispered speech. In *2021 24th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 146–151, 2021.

[16] Vinay Verma and Nitin Khanna. Cnn-based system for speaker independent cell-phone identification from recorded audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[17] Yipin Zhou and Ser-Nam Lim. Joint audio-visual deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14800–14809, October 2021.

[18] Ling Zou, Qianhua He, Jichen Yang, and Yanxiong Li. Source cell phone matching from speech recordings by sparse representation and kiss metric. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2079–2083, 2016.