MLVU information sheet

Please include this page in your report either at the start or at the end, before the appendix. Do not change the formatting.

Group number
46

Authors

name	student number
Sofija Kali Kirn	2769090
Sławomir Meczyński	2783154
Kamil Pulchny	2770523
Andreea Stănescu	2757026
Stanisław Wasilewski	2763732

Software used *For the entire project we used Python while using different libraries. The main library is Keras, which has been needed since we built a neural network. The rest of the used libraries are: opencv-python, tensorflow, wave, numpy, scipy, matplotlib, IPython, pydub, scikit-learn, pandas.*

Use of AI tools *AI tools, like ChatGPT, have been used only once, for informative purposes. When the dataset containing the audio data was found, ChatGPT was used in order to explore the possible machine-learning-based solutions for the deep-fake detection problem. After we got the idea that we wanted to create a convolutional neural network, AI tools were not used anymore.*

Link to code (optional) *https://github.com/Stasieniec/VU-ML-AI-voice.git*

How Effective Are Convolutional Neural Networks with Various Hyperparameters in Differentiating AI-generated Speech from Real Human Speech?

March 29, 2024

Abstract

This paper investigates the effectiveness of Convolutional Neural Networks (CNNs) with varying hyperparameters in discerning AI-generated speech from authentic human speech. We construct a baseline CNN model, employing convolutional, max pooling, flattening, and dense layers to extract and analyze spectrogram features. We investigate the impact of L1 and L2 regularization techniques on model performance, alongside the influence of integrating RGB color channels into the neural network architecture. Despite achieving high training accuracy on validation data (98%), testing on unseen voices showcases a lower accuracy, nevertheless, still indicating its ability to generalize to novel data. Our findings underscore the potential of CNN-based models in detecting deepfake audio and highlight areas for further research in detection capabilities for AI-generated audio.

1 Introduction

1.1 Motivation

AI-generated content, particularly in the form of deepfake audio poses significant challenges in many aspects of our society such as security, privacy, and trust. As the use of AI-generated voices continues to rise, the need for robust detection algorithms becomes increasingly important.

Considering the context, our research focuses exactly on this, with our research question being: How effective are Convolutional Neural Networks (CNNs) with various hyperparameters in distinguishing AI-generated speech from real human speech? The motivation behind this inquiry stems from the need to advance the detection of deepfake audio from real human speech as people's privacy and personal safety can be greatly affected by such technology. By exploring the efficacy of CNNs across different hyperparameter configurations, in particular by changing color, which modifies the input dimension, and adding L1 and L2 regulators, we aim to search for and find valuable insights into the ongoing discourse on combating the rapid increase of AI-generated audio content.

1.2 Literature review

Since the use of AI-generated voices has increased in recent years, the need to detect them has also emerged more than ever. Despite these advancements, the development of effective detection algorithms is still in its early stages, presenting challenges in tackling this threat [1]. Therefore, there were attempts to create the best machine learning that could detect DEEP FAKE voices, which rests as a foundation not only for this paper but also for future research. At the moment, the main focus of the available models is on video and image deepfakes, but not plenty on audio deepfakes, which is an area that didn't have much success, despite the research efforts [3].

In order to create a model that can detect fake audio recordings, multiple techniques can be used. Some of the most frequent approaches that can be selected for such a task are Generative Adversarial

Networks (GANs), Convolutional Neural Networks (CNNs), and Deep Neural Networks (DNNs), even though such techniques are also used for generating the audio [3]. There are also some other innovative ways of detecting AI-generated speech, such as DeepSonar. This technology applies layer-wise activation of neurons, which can identify the subtle differences between real voices and AI-generated ones [1]. Another approach to this issue could be the one taking into consideration the Explainable AI (XAI) methods, which were used originally for image classification, but recently, a human-perception level interpretability approach has been introduced [2].

In the previous literature, the importance of generalization in detecting AI-generated speech is emphasized. One of the characteristics that could ensure the model generalization is the biometric characteristics of the speaker, which could give an insight into how different hyperparameters work in the case of speaker-based detection [5]. Another feature that was found to be viable for audio spoofing detection is the emotional feature of the audio, which could be used to detect emotional inconsistencies in DEEP FAKE audio [6].

When comparing existing detection methods of fake audio, it highlighted the fact that the method choice has an impact on performance, identifying trade-offs between accuracy and scalability [4]. Important characteristics for effective detection and the generalizability of detection models are revealed by the evaluation of preprocessing steps [9].

1.3 Approach

Our approach to tackling the issue of deepfake audio detection begins with a structured analysis of the available data. Firstly the dataset is inspected, addressing any imbalances through preprocessing to ensure its suitability for training our models. Subsequently, we partition the data into distinct subsets, namely training, validation, and test sets, to ensure robust model evaluation. This is performed twice, for male and female voice, as the test set consists of the voice of a person that the model has not seen at all during training. Then the construction of our baseline model starts, designed to provide a baseline upon which further enhancements can be implemented. Leveraging convolutional layers for feature extraction, max pooling for dimensionality reduction, flattening to prepare the data for dense layers, and dense layers for classification, we iteratively refined our model by experimenting with various hyperparameters. This repetition aims to identify the optimal configuration that yields the best output; an effective deepfake audio detection system capable of reliably distinguishing between AI-generated and authentic human speech.

2 Data inspection and preparation

2.1 Data inspection

The data used for the development of this model comes from a dataset called DEEP-VOICE, which provides a moderately large collection of audio samples that are designed to be training material for AI-generated speech detection algorithms. Therefore this dataset serves its purpose of facilitating the development of real-time detection of fake human speech, having a direct impact on solving the issues regarding voice cloning. The dataset comprises two parts: the real audio and the DeepFake versions. In the subset of the real speeches, there are 8 celebrities, each one having around 20 minutes of audio recording. The other subset consists of the fake audio, which is created by sampling one of the 8 celebrities and transforming their voice into the ones of the 7 others. The AI-generated speeches were created using Retrieval-based Voice Conversion (RVC). The dataset also comes with a CSV file that contains extracted features from one-second windows of audio, but it will not be used for this model. Before it was put into the dataset, the data was preprocessed. For the audio to be suited for the conversion using RVC, the background noise has to be removed from each speech sample, keeping just the true voice of the celebrity.

2.2 Data augmentation

The original dataset came with a high class imbalance since each one of the real audio was transformed into 7 alternative versions. Therefore, the subset of the fake data was 7 times bigger than the one of the real data. The data imbalance can cause issues in the training process of the algorithm, such as overfitting. Because of this, more real data had to be gathered.

For each one of the 8 celebrities, audio recordings were searched and processed. Since a large quantity of processed audio was needed, interviews were not a feasible solution, so speeches were the best options for getting a long, uninterrupted recording of someone's voice. After the speeches were found, the background noises had to be removed. In most cases, it was an easy process, since minor background noise, such as claps, is easy to identify and extract. In some cases, some manual processing was needed, since the speeches would be interrupted by other people. The change of voice would have a significant impact on the performance of the model, since the voice characteristics would change, thus, special attention was put into gathering quality audio. After the additional real recordings were gathered, the data imbalance issue was solved, since there were approximately equal parts of real human recordings and AI-generated voices.

2.3 Data preparation

The data has to be handled in a specific way in order to be fed into the model and used to the fullest extent possible for training and testing. As a result, every recording was divided into 10-second segments, which are both sufficiently lengthy to provide the model with adequate information and yet short to provide numerous samples. If the audio windows were longer, there would be fewer data for testing and training, but if the audio windows were too short to find patterns that could determine if an audio is real or fake. Following audio splitting, each 10-second window was transformed into a spectrogram. Spectrograms were used as the input method for the CNN due to their large feature set, compressed encoding of the audio signals, and most significantly, their frequency-time representation—a critical component of the CCN model [10]. This model excels at learning hierarchical features from input, and spectrograms provide a systematic mechanism for the convolutional neural network to extract significant characteristics from signal amplitude and frequency.

2.4 Splitting of training and testing data

The dataset consisting of spectrograms as 128x128 images was split into training, validation, and test data. The ultimate purpose of the model - predicting completely new voices - has been accounted for in the split. One person with a male voice (Barrack Obama) and one with a female voice (Taylor Swift) were selected. Then, individual datasets were created for both of these people. For each person, the training and validation split consists of every spectrogram except for those that are created from real audio of that person or from audio generated to imitate this person. Then, the training split is 80% of this part of the dataset and validation is 20%.

For test data, only spectrograms consisting of the real voice of this person or the voice generated to imitate this person were used. The duplication of the dataset was performed due to the restricted amount of data. If two people were kept only for testing, 25% of the data would be in the test split, which would not be optimal considering the need for train and validation splits, and the amount of data available. Thus, two datasets in total were created: the F dataset where yet unseen data of female voice is used for testing, and the M dataset where yet unseen data of male voice is used for that purpose. Furthermore, each of these two datasets was further duplicated into two versions: one of which is in grayscale and one in RGB. Finally, this creates the following datasets:

- One female voice used for testing, grayscale spectrograms
- One female voice used for testing, RGB spectrograms

- One male voice used for testing, grayscale spectrograms
- One male voice used for testing, RGB spectrograms

3 Methods

3.1 Overview of the model

Our deep learning model distinguishes grayscale spectrograms of real human voices from deepfake voices. It comprises convolutional layers (yellow blocks)[Figure 1.] to extract spatial patterns, followed by Max-Pooling layers (red blocks)[Figure 1.] for dimensionality reduction while retaining crucial information. Flattening (green block)[Figure 1.] transforms pooled feature maps into a vector format. Dense layers (blue blocks)[Figure 1.] learn patterns from previous layers. Dropout (0.5, 0.3, 0.2, 0.2) between dense layers prevents overfitting.

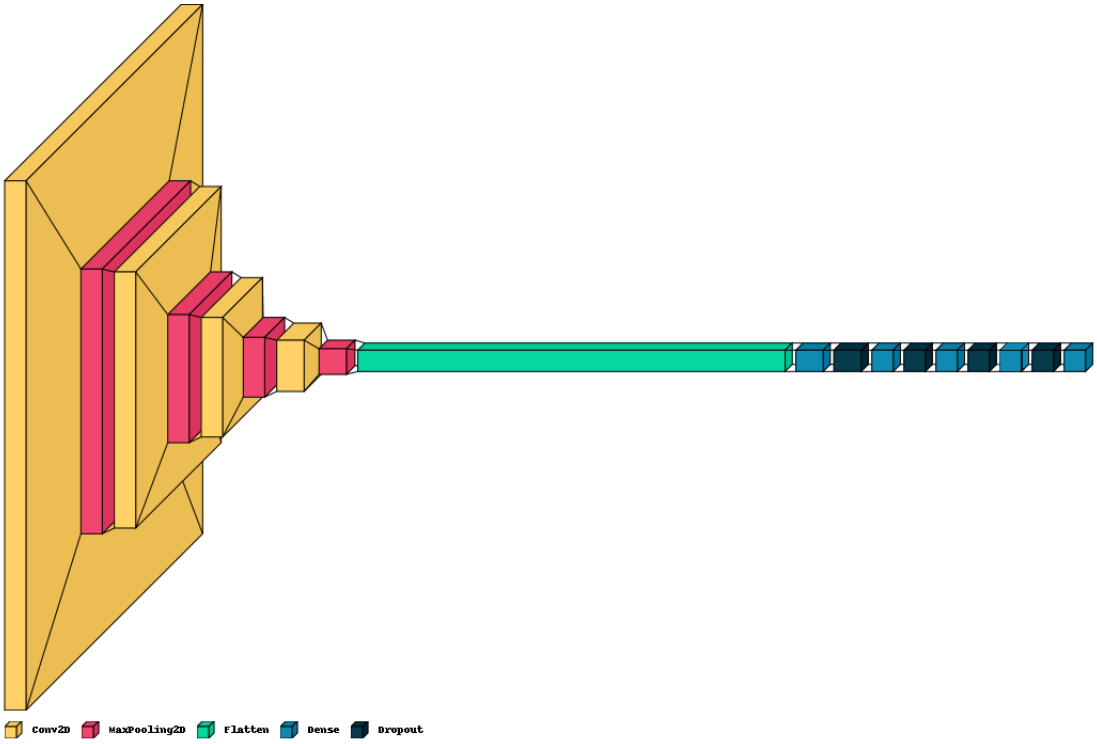


Figure 1: Baseline model visualization

Through trial and error, the number of 13 epochs was discovered to be the optimal number before the model started to overfit [Figure 2.]. To determine this, validation data was used. The batch size for training was set to 16, and the number of steps per epoch was 101. The model utilizes Adam optimizer and binary cross entropy as a loss function. During training, the model achieved 98% accuracy on the validation data.[Figure 2.] It will be, however, evaluated on test data, from yet unseen people further in the paper.

The structure and choice of hyperparameters for the baseline model is as follows:

1. Convolutional Layers and Pooling:
    - Input Conv2D layer employing 32 filters of size (5, 5), stride of 1, padding of 0, with Rectified Linear Unit (ReLU) activation function, operating on input images of dimensions (128, 128, 1)

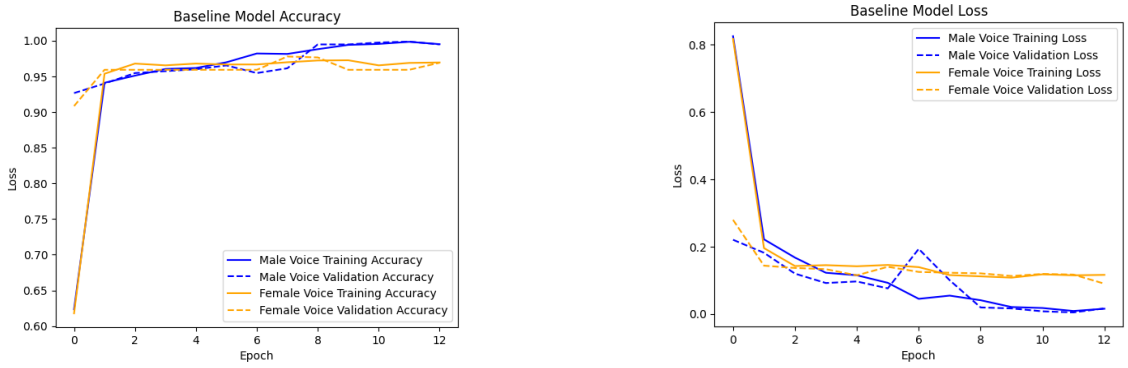
- MaxPooling2D layers subsequent to each convolutional layer, downsampling the feature maps using a pooling window of size (2, 2).
- Three additional Conv2D layers with the same stride and padding as input layer, utilizing 64, 128, and 256 filters of size (3, 3) each, activated by ReLU.

2. Transformational Flatten layer to reformat the output of the convolutional neural network into one-dimensional vector.

3. Fully connected Dense layers:

   - Dense layer with 256 units and ReLU activation, followed by a Dropout layer with a dropout rate of 0.5.
   - Subsequently, incorporated three additional Dense layers, each comprising 128, 64, and 32 units respectively, each followed by ReLU activation and a Dropout layer with dropout rates of 0.3, 0.2, and 0.2 correspondingly.

4. Single-unit Dense layer with Sigmoid activation, facilitating binary classification tasks by outputting a probability score.



(a) Baseline model accuracy

(b) Baseline Model Loss

Figure 2: Baseline Model Training performance

3.2 Other models

Based on the baseline model, five additional models were developed. Each model was trained in two versions, one with a male voice in the test split, and one for a female voice in the test split:

Baseline + L1: Utilizing the architecture of the baseline model, this model applies L1 regularization to weights of every layer, with a regularization strength of 0.01.[Figure 5]

Baseline + L2: Likewise, this model applies the L2 regularization to the weights of every convolutional and dense layer. The regularization strength is 0.01.[Figure 7]

Color model: This model was trained on spectrograms utilizing RGB colors. The shape of the input of the baseline model was extended to (128, 128, 3), to account for RGB values for each pixel.[Figure 9]

Color + L1: This model, based on the color model, applies L1 regularization to every weight of convolutional and dense layers, with regularization strength of 0.01.[Figure 11]

Color + L2: In parallel to the Color + L1 model, this model is based on the Color model and applies L2 regularization to every weight of convolutional and dense layers, with a regularization strength of 0.01.[Figure 13]

3.3 Convolutional layers

In image recognition or computer vision we are often encountering convolutional layers. These layers apply convolution operations to input data, allowing the network to learn difficult paterns and spatial hierarchies of features.

Given an input tensor X with dimensions H, W, C_{in}, where H and W represent the height and width of the input, and C_{in} represents the number of channels, and a set of learnable filters W of dimensions F, F, C_{in}, C_{out}, where F is the filter size and C_{out} is the number of output channels, the output tensor Y of the convolutional layer is given by[11, 12]:

$$Y_{i,j,k} = \sum_{l=0}^{C_{in}-1} \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} X_{(i+m),(j+n),l} \cdot W_{m,n,l,k} + b$$

Where:

- (i, j) represents the spatial location in the output of features.
- k indexes the output channels.
- $X_{(i+m),(j+n),l}$ represents the input values affected by the filter at position (i, j).
- $W_{m,n,l,k}$ denotes the weights of the filter applied to the input at position (m, n) and channel l to produce the output at channel k.
- b is the bias term.

The output Y is then passed through an activation function, commonly ReLU, to achive non-linearity. During training, the weights W and biases b are optimized through backpropagation, minimizing loss function.

3.4 Max Pooling

Max Pooling is used to reduce the dimensions of the input. It improves the detection of dominant features with the spectrograms of audio files and improves computational efficiency. Max Pooling adds a small amount of translation invariance, which means that translating the image of the spectrogram by a small amount does not significantly affect the values of most pooled outputs. We are using a 2x2 pool size which is equivalent to reducing the size by 2. The mathematical function is:

$$MaxPooling(X)_{i,j,k} = \max_{m,n} X_{i \cdot s_x + m, j \cdot s_y + n, k}$$

where X is the input, (i, j) are the indices of the output, k is the channel index, s_x and s_y are the stride values in the horizontal and vertical directions, respectively, and the pooling window is defined by the filter size f_x and f_y centered at the output index (i,j).

By emphasizing the highest values within the pooling window we make our model resistant to small variations in the input data.

3.5 Flattening

Flattening is the process of converting multi-dimensional matrices into 1-dimensional linear vectors. We do that because we need to insert this data into subsequent fully connected Dense layers. In our model, we apply flattening after three sets of convolution and max pooling. It results in converting high-level features into a single vector correctly. We can describe this process symbolically as: given a pooled feature map P of dimensions p_x, p_y, where p_x and p_y are the width and height of the map, flattening is the process of reshaping P into a one-dimensional vector F of length $p_x \cdot p_y$.

Flattening ensures that the hierarchy of the features is preserved, which is crucial for the classification problem.

3.6 Dense layers

In neural networks Dense layer is called a fully connected layer as it is connected to all neurons of the subsequent layer.

Given an input vector x of size n, a weight matrix W of size $n \times m$, where m is the number of neurons in the dense layer, and a bias vector b of size m, the output vector y of the dense layer is computed as[11, 12]:

$$y = f(Wx + b)$$

- Wx represents the matrix multiplication of the input vector x and the weight matrix W.
- b is the bias vector added to the result of the matrix multiplication.
- f denotes the activation function applied element-wise to the result, introducing non-linearity to the layer.

Commonly used activation functions are ReLu and Sigmoid. During the training phase a layer learns weight matrix W and bias b through a gradient descent resulting in minimalization of the loss function. It is capable of recognizing complex relationships, making it useful for image processing.

3.7 L1 and L2 regularization

L1 and L2 regularizations are techniques used to prevent overfitting. They provide a balance between fitting the training data and ensuring good generalization to unseen data. L1, also called lasso regression, adds the sum of the absolute values of our model's coefficients to the loss function, encouraging sparsity and feature selection[8]. It adds the *absolute value of magnittude* of the coefficient as a penalty term to the loss function. It helps to distinguish essential features between artificially generated speech and the real one. The L1 cost function is the following:

$$Cost = \sum_{i=0}^{N} \left(y_i - \sum_{j=0}^{M} x_{ij} w_j \right)^2 + \lambda \sum_{j=0}^{M} |w_j|$$

Here, the first sum is the loss function and the last is the regularization term. x represents features, n number of data points, y target values (dependent variables we want to predict), w weights, and λ predicted values.

L2 regularization, also called ridge regression, adds the sum of the squared values of the model's coefficients. It enables smaller but non-zero coefficients. The regression adds the *squared magnitude* of the coefficient as the penalty term to the loss function.[8] The cost function is:

$$Cost = \sum_{i=0}^{N} \left(y_i - \sum_{j=0}^{M} x_{ij} w_j \right)^2 + \lambda \sum_{j=0}^{M} w_j^2$$

Here, similarly to L1, the first sum is the loss function and the last is the regularization term. The variables are explained in the L1 cost formula.

In summary, the advantage of L1 regularization is producing sparse models, which is beneficial when we are selecting features from the audio. L2 regularization is more optimal when we analyze strong correlations between features and small but non-zero coefficients. L1 uses the absolute value and L2 squares. By squaring values, we are putting more emphasis on large values and less influence on small values.

3.8 Output

The output consists of a single neuron with a Sigmoid activation function. It computes an output, between 0 and 1, which is a probability score. A value close to 0 suggests that the input is a real voice, and, a value close to 1 indicated that it is probably an AI-generated speech.

4 Results

4.1 Presentation of Results

Baseline

Model/Metric	Mean Accuracy	Mean Precision	Mean Recall	Mean AUC	Mean F1
Baseline	0.6223	0.2957	0.2339	0.5112	0.2593
Baseline + L1	0.7087	0.0000	0.0000	0.5000	0.0000
Baseline + L2	0.6172	0.2869	0.2273	0.5123	0.2516
Color	0.6233	0.2978	0.2364	0.5064	0.2617
Color + L1	0.7087	0.0000	0.0000	0.5000	0.0000
Color + L2	0.5722	0.2853	0.3048	0.4929	0.2871

Table 1: The average results between the female and male evaluations

To effectively distinguish AI-generated sounds, the assessment method investigates the performance of various settings within CNN models, each targeting both female and male voice identification. The conclusions drawn highlight important differences in performance between various model configurations. The baseline accuracy obtained in the case of both male and female voices is fairly moderate, at roughly 54.55%, and respectively 70.15%. In both of the cases, the scores obtained for precision, recall and F1 show potential for advancement in both cases. When looking at the outcome of applying L1 regularization, a significant enhancement of the accuracy can be observed, mainly noticeable in the male model, which achieves an accuracy of 80.61%, but with the cost of transforming the precision and recall null. Upon manual inspection, this phenomenon occurs because the Baseline + L1 model classifies every instance as fake. The relatively high accuracy is due to the imbalance in the test set. On the other hand, L2 is generating more balanced improvements, with a lower accuracy gain in the case of females and a slight decrease in the case of males.

The addition of the color on the spectrograms produced intriguing results, since accuracy was expected to increase significantly, but, similar to the case of the L2 regularization applied to the baseline, the increase was low in the case of females and a slight decrease happened in the case of males. When looking at the precision, recall, AUC, and F1 scores, both in the cases of females and males, the majority of them displayed a minor decrease. Another unexpected finding is the fact that the integration of L1 and L2 regularizations into the color model doesn't produce substantial improvements in the performance of the model, mirroring the behavior observed when the same regularizations were applied to the baseline model.

Model/Metric	Female					Male				
	Accuracy	Precision	Recall	AUC	F1	Accuracy	Precision	Recall	AUC	F1
Baseline	0.5455	0.3963	0.3234	0.4993	0.3562	0.7015	0.2405	0.2500	0.5398	0.2452
Baseline + L1	0.6112	0.0000	0.0000	0.5000	0.0000	0.8061	0.0000	0.0000	0.5000	0.0000
Baseline + L2	0.5609	0.4071	0.2836	0.5246	0.3343	0.6735	0.1667	0.1711	0.4999	0.1688
Color	0.5629	0.4113	0.2886	0.5075	0.3392	0.6837	0.1842	0.1842	0.5054	0.1842
Color + L1	0.6112	0.0000	0.0000	0.5000	0.0000	0.8061	0.0000	0.0000	0.5000	0.0000
Color + L2	0.5474	0.4012	0.3333	0.5170	0.3641	0.5969	0.1694	0.2763	0.4689	0.2100

Table 2: The results divided into the cases of the male and female

4.2 Interpretation of Results

The above-described observations regarding the performance of the model, depending on various hyper-parameters, shed light on the factors influencing the performance of the AI-generated voice detection algorithm.

The most notable change that can be seen in Table 2 is the significant improvement of the accuracy when the L1 regularization is applied to the male model. As stated in the presentation of results, this is due to the model classifying every instance as fake, which was discovered by manual inspection and can be seen in Figure 5 and Figure 8. This suggests that reducing the model's complexity by the regularizer is not a feasible solution. When looking at the results generated by the application of L2 regularization on the baseline, it can be seen as more balanced enhancements. This behavior might be due to the fact that L2 is capable of penalizing large weight magnitudes, without suppressing non-zero weights.

The color model shows somewhat consistent performance, both by itself and with L2 regularization, suggesting that the color analysis, by itself, is improving the baseline, therefore not generating major differences between the colored model and the regularized one. The improvement in the accuracy of the colored model can indicate the fact that a color analysis might capture the subtle variations of the spectrograms' characteristics, which can be missing in AI-generated speeches. The slim improvements observed with L2 regularization applied to the colored model hint at the fact that more nuanced models, that can catch the subtle details, can be an alternative for future enhancements.

As a generalization of the results, the hypothesis is that a colored model would perform better than the baseline model since the CNN can pick the more subtle details from a colored model, which could impact the performance of the detection. Another hypothesis that can be drawn from the evaluation conducted is that this type of detection algorithm needs more advanced techniques than L1 and L2 regularization, in order to increase the performance.

5 Discussion

Our study delved into the effectiveness of Convolutional Neural Networks (CNNs), a deep learning architecture, for deepfake audio detection using spectrogram images of audio clips from a dataset of celebrities. The implementation of CNNs proved effective to a limited extent. Upon testing on audio from people the model has never been exposed to, it has achieved 62% mean accuracy at best (not accounting for models using L1 regularization, which classifies every instance as fake). This indicates a potential for generalization, although it is clear that more research should be completed to achieve feasible solutions.

It is also essential to acknowledge the presence of background noise in the spectrograms used for training and testing. While efforts were made to minimize noise interference, future projects could benefit from recordings made with high-quality equipment to further enhance accuracy and applicability across diverse environments.

Furthermore, it is crucial to recognize the dynamic nature of AI development and the potential advancements in deepfake audio generation. Continuous model training on new data may be necessary to adapt to evolving deepfake techniques. Alternatively, the development of more robust models capable

of effectively deciphering increasingly sophisticated DeepFake audio is essential. As AI technology progresses, maintaining vigilance and agility in DeepFake detection methodologies is crucial in mitigating potential threats that come with this technology.

References

[1] Run Wang, Felix Juefei-Xu, Yihao Huang, Qing Guo, Xiaofei Xie, Lei Ma, and Yang Liu. 2020. DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices. In Proceedings of the 28th ACM International Conference on Multimedia (MM '20). Association for Computing Machinery, New York, NY, USA, 1207–1216. https://doi.org/10.1145/3394171.3413716

[2] Lim, S.-Y., Chae, D.-K., & Lee, S.-C. (2022, April 13). Detecting deepfake voice using explainable deep learning techniques. MDPI. https://www.mdpi.com/2076-3417/12/8/3926

[3] Khanjani, Z., Watson, G., & Janeja, V. P. (2021, November 28). How deep are the fakes? focusing on audio deepfake: A survey. arXiv.org. https://arxiv.org/abs/2111.14203

[4] Almutairi, Z.; Elgibreen, H. A Review of Modern Audio Deepfake Detection Methods: Challenges and Future Directions. Algorithms 2022, 15, 155. https://doi.org/10.3390/a1505015

[5] Pianese, A., Cozzolino, D., Poggi, G., & Verdoliva, L. (n.d.). Deepfake audio detection by speaker verification — IEEE conference ... https://ieeexplore.ieee.org/abstract/document/9975428

[6] E. Conti et al., "Deepfake Speech Detection Through Emotion Recognition: A Semantic Approach," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 8962-8966, doi: 10.1109/ICASSP43922.2022.9747186. keywords: Voice activity detection;Emotion recognition;Semantics;Transfer learning;Signal processing algorithms;Speech recognition;Streaming media;deepfake;audio forensics;deep learning,

[7] Schmidt, M., Fung, G., Rosales, R. (2007). Fast Optimization Methods for L1 Regularization: A Comparative Study and Two New Approaches. In: Kok, J.N., Koronacki, J., Mantaras, R.L.d., Matwin, S., Mladenič, D., Skowron, A. (eds) Machine Learning: ECML 2007. ECML 2007. Lecture Notes in Computer Science(), vol 4701. Springer, Berlin, Heidelberg. https://doi.org/

[8] Andrew Y. Ng. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. In Proceedings of the twenty-first international conference on Machine learning (ICML '04). Association for Computing Machinery, New York, NY, USA, 78. https://doi.org/10.1145/1015330.1015435

[9] Müller, N. M., Czempin, P., Dieckmann, F., Froghyar, A., & Böttinger, K. (2022). Does audio deepfake detection generalize?. arXiv preprint arXiv:2203.16263.

[10] Wyse, L. (2017). Audio spectrogram representations for processing with convolutional neural networks. arXiv preprint arXiv:1706.09559.

[11] LeCun et al. (1998). Gradient-Based Learning Applied to Document Recognition.

[12] Nair Hinton, (2010). Rectified linear units improve restricted boltzmann machines

Apendix

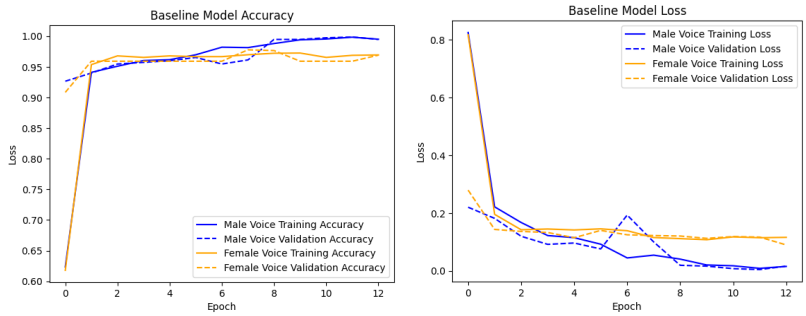


Figure 3: Baseline model accuracy and loss

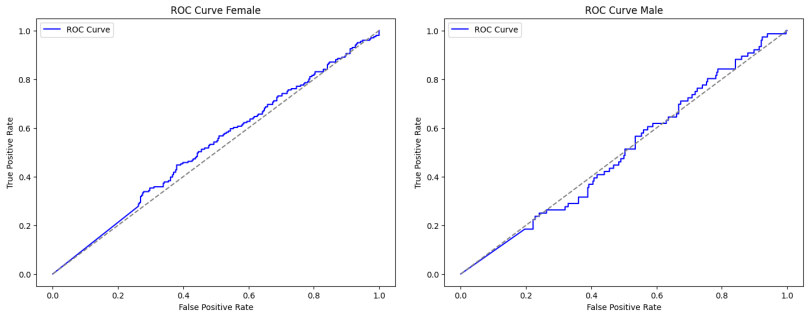


Figure 4: Baseline+L1 model ROC plots for male and female classes

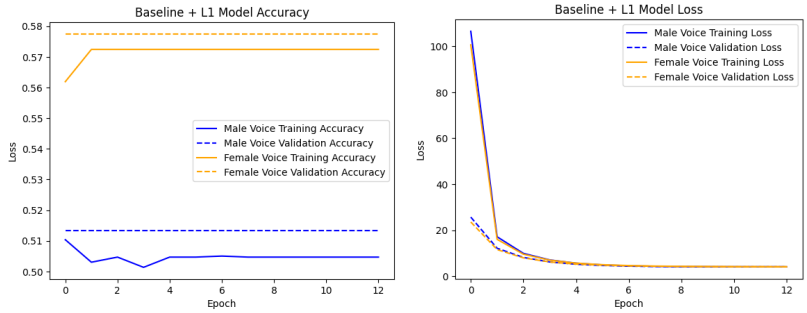


Figure 5: Baseline+L1 model accuracy and loss

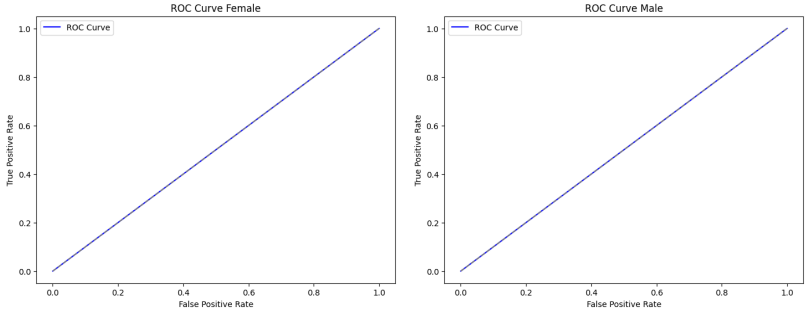


Figure 6: Baseline+L1 model ROC plots for male and female classes

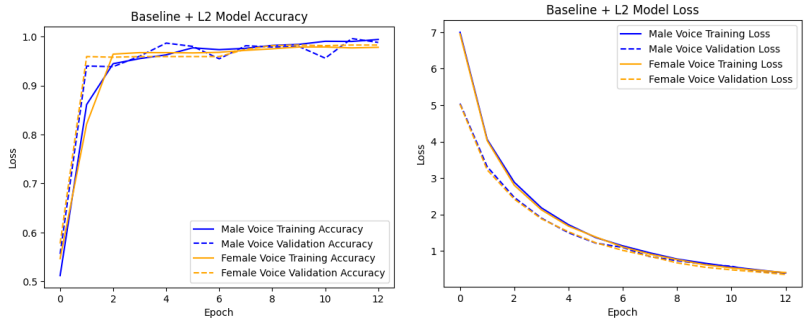


Figure 7: Baseline+L2 model accuracy and loss

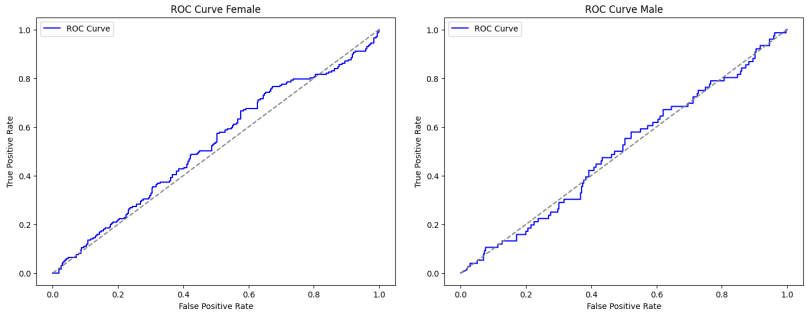


Figure 8: Baseline+L1 model ROC plots for male and female classes

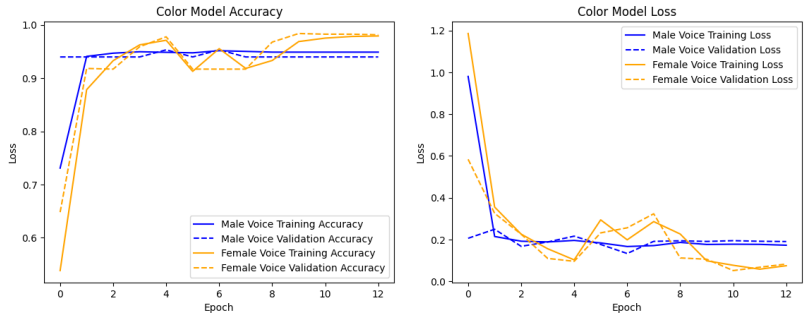


Figure 9: Coloured model accuracy and loss

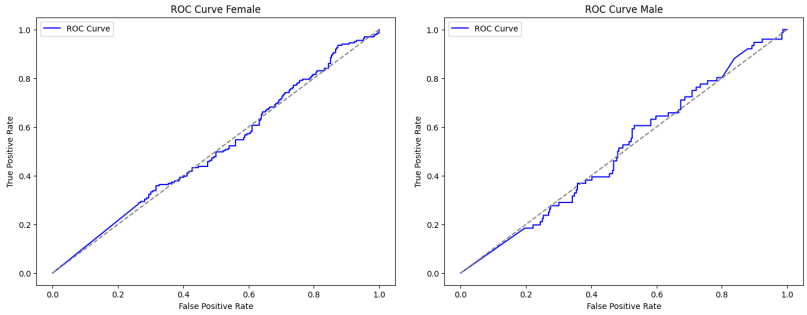


Figure 10: Coloured model ROC plots for male and female classes

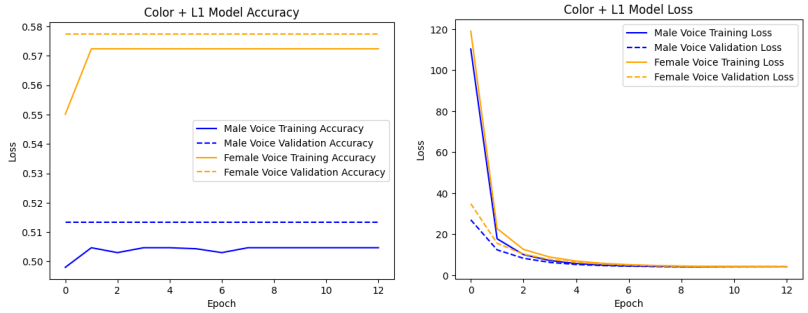


Figure 11: Coloured+L1 model accuracy and loss

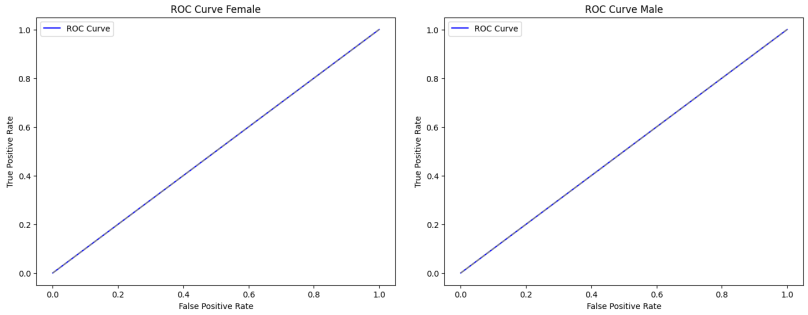


Figure 12: Coloured+L1 model ROC plots for male and female classes

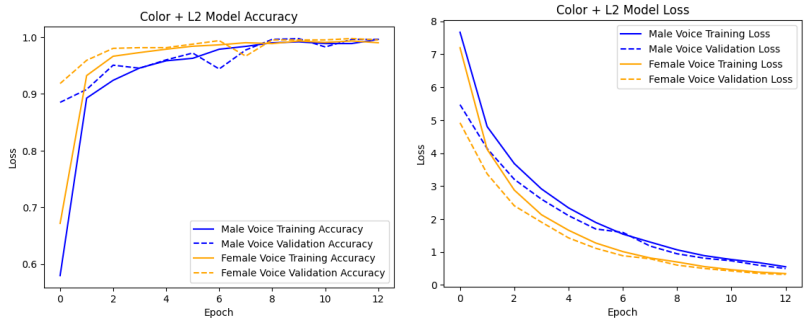


Figure 13: Coloured+L2 model accuracy and loss

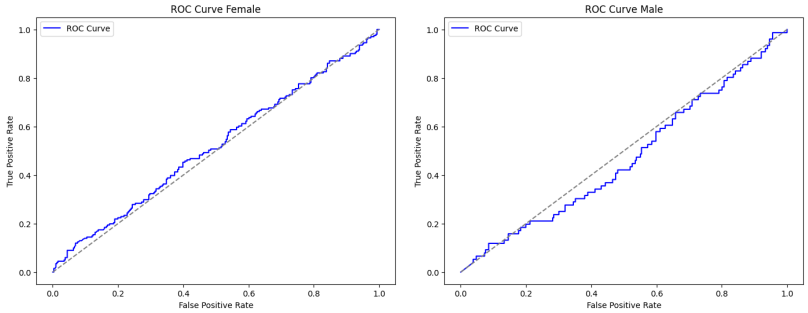


Figure 14: Coloured+L2 model ROC plots for male and female classes