Voice Gender Recognition Using 2D CNN

DOAA Hamed
dept of Computer Science
Nile University
Cairo, Egypt
d.hamed@nu.edu.eg

Rawan Hisham
dept of Computer Science
Nile University
Cairo, Egypt
R.Hisham@nu.edu.eg

Abstract—**In this paper, a convolutional neural network (CNN) deep learning model has been described to recognize voice gender. The data set has 3,168 recorded samples of male and female voices. The samples are produced using acoustic analysis. A CNN A deep learning algorithm has been applied to detect gender-specific traits. Our model achieves 96.74% accuracy on the test data set. Also, an interactive API has been built for the recognition of the gender of the voice.**

Keywords—deep learning, CNN, dataset

I. INTRODUCTION

The goal of this project is to create a phonetic gender detection system that can efficiently and accurately determine a speaker's gender based on auditory features. This study aims to tackle the difficulty of correctly detecting gender from auditory data. This is crucial in a variety of fields, including human-computer interaction, speech recognition, and security. The inability of deep learning-based linguistic gender recognition models to be interpreted is one of the issues. Although deep learning models have excelled in many language processing tasks, it can be challenging to grasp the models' reasoning because of their opaque nature. This hampers the creation of more precise and effective gender classification models and restricts the capacity of researchers and specialists to recognize characteristics and patterns employed in those models linguistic gender detection systems. A proposed solution to solve this problem is to develop a deep learning-based linguistic gender detection approach using convolutional neural networks (CNNs). CNNs have achieved remarkable success in various language processing tasks such as speech recognition, speaker recognition, and emotion recognition. The proposed approach uses a CNN architecture to extract relevant features from speech signals and use them for gender classification.

II. RELATED WORK

A. The Power Of CNN Architecture

The use of convolutional neural networks (CNNs) in voice gender recognition has been a topic of recent research in the field of speech processing. CNNs have shown promise for improving the accuracy and efficiency of gender recognition systems by automatically extracting relevant features from speech signals. The architecture of CNN typically includes convolutional layers, pooling layers, and fully connected layers that can learn the relevant features from the input data and classify them into male or female gender classes. which increases the evaluation of the model.

TABLE I. Accuracy of CNN model

Accuracy (%)		
Model	*Train*	*Test*
CNN	0.9707	0.9503

B. Data Set

This database was created to identify a voice as male or female based on the acoustic properties of the voice and speech. The dataset consists of 3,168 recorded voice samples collected from male and female speakers. The voice samples are pre-processed by acoustic analysis in R using the see wave and tuner packages, with an analyzed frequency range of 0 Hz–280 Hz. Some of the acoustic properties used in implementation In Table II shown below.

TABLE II. Measured Acoustic Properties

Acoustic Properties	
Properties	*Description*
meanfreq	mean frequency (in kHz)
sd	standard deviation of frequency
median	median frequency (in kHz)
Q25	first quantile (in kHz)
Q75	third quantile (in kHz)
IQR	interquartile range (in kHz)
skew	skewness
kurt	kurtosis
sp.ent	spectral entropy
sfm	spectral flatness
mode	mode frequency
centroid	frequency centroid
peakf	peak frequency
modindx	modulation index

C. *Software Libraries*

Python is an open-source, interpreted programming language that is interactive, dynamic, object-oriented, and simple to learn. The syntax of Python is incredibly clear while still having amazing power.

Keras is a Python-based high-level neural network library that can operate on top of TensorFlow or Spark. Keras offers a high-level API for constructing and training neural networks.

TensorFlow is an open-source software package that uses data flow graphs to do numerical calculations. The graph's nodes represent mathematical processes, and the graph's edges represent the multidimensional data arrays (tensors) that communicate between them. TensorFlow's flexible architecture allows you to conduct machine learning and deep neural network research on a GPU or CPU, but new fields can be simply adapted.

NumPy is an open-source fundamental package that can quickly and easily interact with a wide range of databases. Numpy is used by Keras for input data types.

Gradio is an open-source Python library that allows you to quickly create custom web interfaces for your machine learning models or any other Python function. Gradio provides a simple and intuitive way to create web-based interfaces that users can interact with to input data, visualize results, and explore model predictions.

III. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks (CNNs) are a type of neural network that has revolutionised the field of computer vision and led to significant improvements in image, voice, and video recognition tasks. Convolutional neural networks (CNNs) typically consist of several types of layers, each performing a specific operation on the input data.

main types of layers used in CNNs:

- Convolutional Layer
- Pooling Layer
- Flatten Layer
- Fully Connected layer

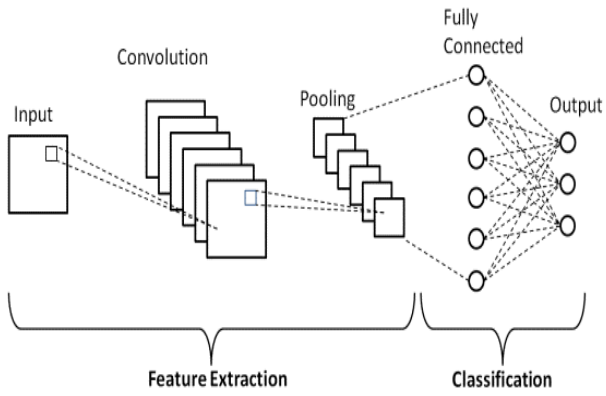

Fig.1: CNN Architecture

- Convolutional Layer: The convolutional layer performs feature extraction by convolving the input data with a set of learnable filters. Each filter is a small matrix of weights that slides over the input data and performs a dot product at each location. The output of the convolutional layer is a feature map of the learned features. Suppose that we have some N×NN×N square neuron layer, which is followed by our convolutional layer. If we use an m×mm×m filter ωω, our convolutional layer output will be of size (N−m+1)×(N−m+1)(N−m+1)×(N−m+1). In order to compute the pre-nonlinearity input to some unit xℓijxijℓ in our layer, we need to sum up the contributions (weighted by the filter components) from the previous layer cells:

$$x_{ij}^{\ell} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{(i+a)(j+b)}^{\ell-1}.$$

- Pooling Layer: The pooling layer reduces the dimensionality of the feature maps by down sampling them. The most common type of pooling is max pooling, where the maximum value in each region of the feature map is retained.

- The flatten layer simply takes the output feature maps of the previous layer and flattens them into a one-dimensional vector by concatenating all the feature maps along the spatial dimensions.

- Fully Connected Layer: The fully connected layer performs classification or regression by connecting every neuron in the previous layer to every neuron in the current layer. The output of the fully connected layer is a vector of class scores or regression values.

IV. METHOD

The Python library is used for all the training, testing, and prediction codes. With the help of the built-in Python library, data was transferred from a CSV file into a NumPy array. The dataset was imported into a two-dimensional Python array from a CSV file. There are 20 parameters and 1 label on each line. label has been converted to an integer (1 for males and 0 for females) and added to the Python array. Also, the data is split into 3 sets: training, testing, and validation.

The model was built using a single input layer, four hidden layers, and a single output layer. The first layer consists of 16 input channels, which are connected to the input layer. The second consists of 32 input channels. The first two hidden layers automatically enter the flatten layer to reshape the output to be a 1D vector. then the fully connected layer, which takes 64 input channels and the positive

activation function relu., then adding to the output layer with a sigmoid activation function.

For compiling the model, we used the Adam optimizer technique to train the model and cross-entry loss for calculating the loss. metric was accuracy and AUC "area under ROC". model with a trained number of epochs equal to 100 and batch size equal to 32. Several loss functions were evaluated using our model algorithm, and the one offering the best performance and accuracy was chosen. On the test data set, the model had an accuracy of 0.95.01%.

For testing the model, we get some voices from websites. to test our model on external data. and we get amazing results by using the librosa library, which extracts the features we entered into the model and processes them into the CNN model to predict the label. The anticipation function is done, as is the deployment of the project.

V. CONCLUSION

The model obtained in this paper shows us that we can use The acoustic properties of voices and speech to detect the voice gender. CNN has been used to obtain the model for classification from data. sets that have the features of voice samples. A larger data set of voice samples can minimize incorrect classifications from intonation. The web page has been published on Git Hub to develop the model from loaded examples of male and female voice samples.

IV. REFERENCES

[1] Becker, K. (2016) *Gender recognition by voice*, *Kaggle*. Available at: https://www.kaggle.com/datasets/primaryobjects/voicegender (Accessed: 22 May 2023).

[2] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[3] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[4] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[5] K. Elissa, “Title of paper if known,” unpublished.

[6] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[7] A. Gibiansky, "Andrew Gibiansky math[code]. Convolutional Neural Networks - Andrew Gibiansky http andrew.gibinky.com/blog/machine-learning/convolutional-neural-networks (accessed May 22, 2023).

APPENDIX

Name	**Tasks**
Doaa Hamed	All in Sequence
Rawan Hisham	All in Sequence