Text-independent Speaker Verification Using Hybrid Convolutional Neural Networks

M. Selin
Research Scholar, Department of Computer Applications, Cochin University of Science and Technology, Cochin, Kerala, India. E-mail: selin.m.a@gmail.com

Dr.K. Preetha Mathew
Professor, Cochin University College of Engineering, Kuttanad, Pulincunnu, Alappuzha, Kerala, India. E-mail: preetha.mathew.k@gmail.com

Received April 23, 2021; Accepted August 18, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V18I2/WEB18352

Abstract

Automatic speaker verification is an active research area for more than four decades, and the technology has gradually upgraded for real application. In this paper, a hybrid convolutional neural network (CNN) model is proposed where a combination of the 3D CNN & 2D CNN model is used for speaker verification in the text-independent scenario. For speaker verification, this novel convolutional neural network architecture was built to capture and discard speaker and non-speaker information at the same time. In the training process, the network is trained to differentiate between different identities of a speaker to establish the background model. The model development of the speaker is one of the important aspects. Most conventional techniques employed the d-vector system to create speaker models by means of an average of the features collected from the speaker utterance. Here a hybrid of convolutional neural networks model is utilized in the development and registration phases for building a speaker model. The approach suggested exceeds the existing methods of speaker verification.

Keywords

Convolutional Neural Network, Deep Neural Network, Speaker Verification.

Introduction

Speaker verification is the process of verifying the claimed identity of a speaker based on the information from the speech signal (Das et al. (2020)). Speaker verification may be divided into two classes, which are text-independent and text-dependent (Nagrani et al. (2020)) relying on the text to be pronounced. A fixed or a predetermined passphrase is used in all phases of the speaker verification procedure in the text-dependent speaker verification mode. On the other hand, no constraints on utterances are considered in advance while verifying text-independent speakers, which makes the scenario a harder task as compared to the text-dependent scenario (Nidadavolu et al. (2020)). At Google, we are involved in the verification of text-dependent speakers with the global code "Ok Google." The selection of this especially fast, roughly 0.6-second global password applies to the Google Keyword Spotting system and Google Voice Search and facilitates device combination (Li et al. (2020)). In general, the three steps of the speaker evaluation process are development, enrollment, and assessment. A background model to represent information related to the speaker will be built throughout the development stage. During the enrollment or registration process, the background model is used to build speaker models for new people. Finally, the requested identification of the speaker is checked by comparison to existing speaker models during the assessment phase. The basic speaker checking mechanism is shown in Figure 1.

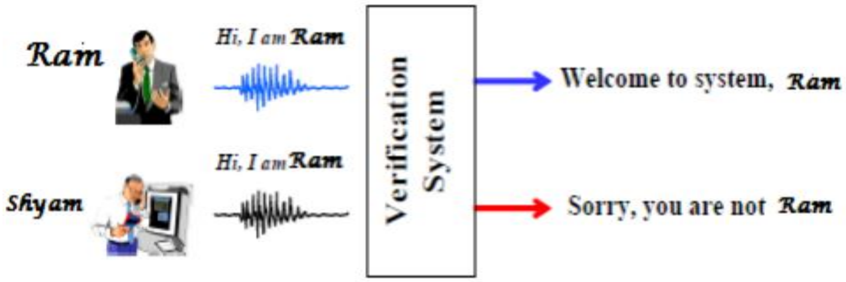

Figure 1 Speaker Verification System

Related Works

Md Raibul et al. (2015) focused on the identification of speakers using cepstral characteristics and Principal Component Analysis (PCA) for classification. Muda. L et al. (2010) performed an improvement analysis on Mel Frequency Cepstral Coefficients (MFCC), as well as advanced time warping methods to achieve better efficiency. Urmila Shrawankar et al. (2013) and MJ Alam et al. (2013) have carried out a detailed study of the extraction techniques of features (Rishi Charan et al. (2017)) such as MFCC, Perceptual Linear Prediction (PLP), Fast Fourier Transform (FFT), Linear Prediction Coefficients (LPC), and Linear Prediction Cepstral Coefficients (LPCC), etc. The Gaussian Mixture Model-Universal Background Model (GMM-UBM) and i-vector (D.A. Reynolds (2000)) are some historically popular speaker verification models developed by Dunn et al. The key drawback of such designs is their unsupervised nature because the setup of speaker verification is not trained properly. Sturm et al. have suggested several approaches for supervising the above training models, including such Support Vector Machine (SVM), based GMM-UBMs and Probabilistic Linear Discriminant Analysis (PLDA) for i-vectors model (W.M. Campbell (2006)).

The advancements of deep learning include various areas such as speech, image processing, and network pruning (Islam et al. (2010)). The effective feature learning process for Automatic Speech Recognition (ASR) (A. Krizhevsky et al. (2012)) and Speaker Recognition (K. Simonyan and A. Zisserman (2014)) was also developed utilizing data-driven approaches employing Deep Neural Networks (DNNs). The deep architecture was also often regarded as black boxes; some methods focused on information theory (G. Hinton et al. (2012)) were proposed for the processing of multimodal features, and positive results were demonstrated (Y. Lei et al. (2014)). For the text-independent setup, DNNs were examined. There have been inquiries for speaker verification in some research projects, such as Convolutional Neural Networks (CNN's) and Locally Connected Networks (LCNs). They consider only the text-dependent system though. To avoid this issue, a method was proposed which uses CNN's intrinsic features to obtain a cohort of different speaker expressions that can be used to construct speaker models.

In this paper, a combination of 3D-CNN & 2D-CNN is used for concurrent extraction of features and construction of speaker models at both development and enrollment stages. For both phases, the proposed method produces equivalent speaker representation structures which have functional and computational benefits. The main objective of this proposed speaker verification technique is to extract, analyze, characterize, and verify information about the speaker identity using a combination of 3D-CNNs & 2D-CNNs algorithms.

Speaker Verification Procedure Using DNN

DNN should be used for the protocol to verify the speaker. In general, in the first section, the approach was discussed. The three stages of development, registration, and assessment are explained in this section.

Development: In this stage, a background model for the representation of the speaker should be developed, derived from the utterances of the speakers. The model gives a representation of the speaker. The display of input data using DNN may be created using speech feature maps from the utterance of the speaker. The model's loss (e.g. Softmax) leads to discriminatory speakers during training in the final representations. Different attempts in research at this level have been studied utilizing state-of-the-art techniques like i-vectors (Das et al. (2020); Nagrani et al. (2020)) and d-vectors (Shrawankar et al. (2013); G. Hinton et al. (2012)). DNN (Hossein Salehghaffari (2018)) would be the essential idea as the speaker feature extractor for the categorization of speakers at the level of framing and utterance.

Enrollment: Here in the enrollment process, a model will be created for every speaker. Each model of the speaker is built on the utterances of the target speaker. At this level, the supervised trained network is given with each utterance (or framing, depending upon the representation level) and the results (output of one of the layers in front of the Softmax layer which offers good representation) are obtained for all expressions. D-vector is the final description of the utterances produced by DNN results. All d-vectors of the target speaker's utterances can be averaged for the development of a speaker model to make a speaker model. Rather than averaging often employed in d-vector methods, a technique was proposed that builds the speaker model at one time by gathering all of the same speaker's utterances.

Evaluation: During the model assessment phase, test statements would be made available to the network and its representations gathered. The most important setting for testing is a one-vs-all configuration, in which the representation of the test utterance is compared to all speaker models and a similarity value is used to pick one. The main metrics of failure in this system are false acceptance and false rejection rates. The predefined threshold is used in erroneous rate rejection/acceptance. If the two rates above are comparable, the Metric Equal Error Rate (EER) displays the error.

Baseline Approach

This section explains the baseline technique. The architecture presented as a model here is a Locally Connected network (LCN). For the removal of low-level features and completely linked layers, the network uses locally connected layers as high-level generators. The PReLU activation is utilized instead of ReLU, showing high repeatability in training and improved performance (Y. Lei et al. (2014)) In the first hidden layer, the locally linked layers are utilized to guarantee sparsity. The network was trained using the loss of cross-entropy as a metric. After the training level, the network parameters will be established. Averaging the output vectors of the previous layer yields utterance d-vectors (before Softmax and without elimination of the PReLU non-linearity). The averaged vectors of the speaker-owned utterances are used to construct the speaker model for implementation. Finally, the similarity score is calculated by evaluating the cosine similarity between the test utterance and the speaker model during the assessment process.

A type of machine learning is deep learning which contains algorithms inspired by brain structure and function. Convolution Neural Network (CNN) is a special type of neural network for processing data in image, text, and sound forms that have worked successfully in their implementation (Kataria et al. (2020)). The term "Convolution Neural Network" developed a statistical operation called convolution, to indicate their network. The convolution operation is the operation of a dot product between processed input matrices. Convolutional Neural Network depends on linking the preceding layer's local area to the next layer. Spatially, CNN establishes local correlation by applying a hierarchical pattern of interaction between neurons of adjacent layers. The preceding layer sub-units are related to the one-layer units. The width of a feature map is determined by the preceding-layer unit number.

The DNN architecture is used in conjunction with the audio stream's stacked frames to perform DNN-based Speaker verification at the utterance level rather than the frame level, and one d-vector is produced for each utterance. A layer connected locally is the baseline design, which is supported at the end by three completely connected layers and a Softmax layer. The output is a Softmax layer, and the cardinality is the number of speakers observed in the development set. Each totally connected layer contains 256 hidden units, and the locally linked layer employs 8 x 8 local patches, rather than the whole visible features as in typical DNNs, for each stimulation of the hidden units.

Hybrid CNN Model

Various difficulties may occur regarding the method used for the baseline. The representation at frame-level does not derive adequate meaning from the information related to speakers. Non-speaker-related information, such as a large number of uttered words in the text-independent setup, can have an impact on the utterance level representation achieved by simple frame stacking. In addition, the Softmax layer requires a large number of samples per speaker, as well as cross-entropy loss, to efficiently construct the speaker-discriminative model. A Hybrid CNN (3D&2D) architecture is meant to collect both temporal and spatial information concurrently to solve the aforementioned concerns (Amirsina Torfi et al (2018)). The network could be able to extract the discriminative characteristics of the speaker, as well as changes within the speaker. The feature maps of various utterances spoken by the same speaker are layered and sent into the proposed network as input. Instead of using a single utterance in the development phase and for creating the speaker model based on the d-vector system, the suggested technique feeds the network the same number of utterances at the same time in both the enrollment and development stages.

The general structure that is used with the utterance level as input for development, enrollment, and evaluation is given in Figure 2, and Table 1 describes the Hybrid CNN architecture. Here the kernel spatial sizes are stated as to $D \times H \times W$ where kernel size H & W are the height (temporary) and width (frequency) measurements, respectively. D is the dimension of the kernel alongside the depth, which defines how much knowledge about utterances collected for the basic operation of the convolution (Amirsina Torfi et al (2018)).

The challenge, in this case, is that Softmax may infer the different words spoken differently, even though the utterance is from the speaker himself. To address these difficulties, the suggested methods collect several within-speech utterances at the same time in order to extract speaker discriminative characteristics. When utilized as the input to the Hybrid CNN for many distinct utterances delivered by the same speaker, the suggested method stacks the function mappings. Rather than using a single utterance (in the development stage) and developing a speaker model based on average descriptive features of various utterances from the same speaker (d-vector system), the proposed Hybrid CNN model uses the same set of utterances for both phases, which are simultaneously integrated.

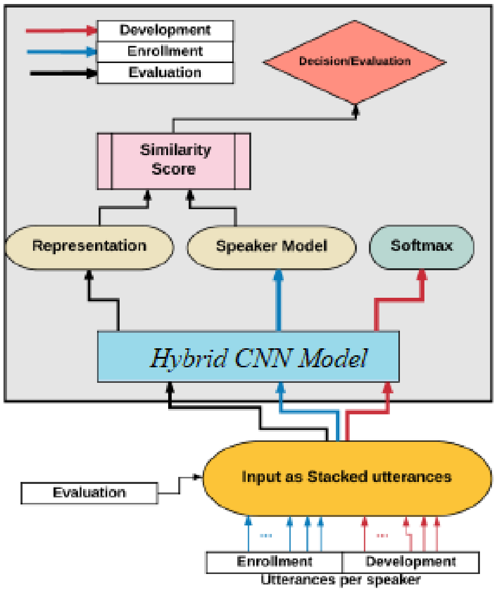

Figure 2 Hybrid CNN Model

To keep the relevant temporal features in the time frame, the pooling procedures are only executed in the frequency axis. The work in (Islam et al. (2015)) inspired our approach since it resists temporal downsampling. Stride 2 is used to conduct a basic reduction on low-level convolution layers to capture features that are substantially overlapped. To develop a more computationally efficient design, successive 2D kernels are utilized instead of cubic kernels (S. Han et al. (2015)).

Table 1 Architecture of Hybrid CNN

Layer	Input Size	Output Size	Kernel	Stride
Conv3d-1	60×80× 40	80× 36 × 16	3× 1 × 5	1× 1 × 1
Conv3d-2	80× 36 × 16	36× 36 × 16	3× 9 × 1	1× 2 × 1
Conv3d-3	36× 36 × 16	36× 17 × 32	3× 1 × 4	1× 1 × 2
Pool1	36× 17 × 32	36× 8 × 32	1× 1 × 2	1× 1 × 2
Conv3d-4	36× 8 × 32	29× 8 × 32	3× 8 × 1	1× 1 × 1
Conv3d-5	29× 8 × 32	29× 6 × 64	3× 1 × 3	1× 1 × 1
Conv3d-6	29× 6 × 64	12× 6 × 64	3× 7 × 1	1× 2 × 1
pool	12× 6 × 64	12× 3 × 64	1× 1 × 2	1× 1 × 2
Reshape	12× 3 × 64	12× 192		
Conv2d-1	12× 192	10× 32	3 × 3	
Conv2d-2	10× 32	8× 64	3× 3	
Conv2d-3	8× 64	6× 128	3×3	
Flatten	6× 128			

Experimental Results

Dataset Description

The LibriSpeech dataset was used in our experiments. The audio portion of LibriSpeech comprises around 1,000 hours of 16 kHz read English Speech. The data is extracted from the LibriVox project read audiobooks. The LibriVox project is a voluntary group responsible for producing around 8,000 public domain audiobooks, most of them in English. The major portion of the recordings is based on Project Gutenberg texts, now in the public sphere. This dataset includes both scripted as well as unscripted data. For the scripted studies, the participants would read a predefined sample of sentences. Interview questions include conversational responses to unscripted samples, which the speakers reply to. Because the recording only comprises the voice of the topic of interest, we exclusively employ written audio recordings.

Evaluation and Verification Metric

The variance scaling initializer, which was built newly for weight initialization (Y. Lei et al. (2014)), is used during the training phase. Batch normalization (E. Variani (2014)) was also used to increase the integration of training and to generalize more effectively. The softmax layer receives the last layer output (FC5), which has a cardinality of N=50, where N is the total number of speakers throughout the development stage. Except for the last layer, PReLU activation follows each layer. During this time, experiments are carried out to check the speakers and assess their performance. The features of Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves are used to assess the findings. The

False Acceptance Rate (FAR) and Validation Rate (VR) are shown on the ROC curve (VR). P_{gen} denotes all match pairs (X_{P1}, X_{P2}), i.e. identity pairings, whereas P_{imp} denotes nonmatch pairs. Assume that DW represents the Euclidean distance (X_{P1}, X_{P2}), between network outputs as an input. As a result, it may be divided into two categories: true positive and false acceptance.

$$TP(T) = \{(X_{P1}, X_{P2}) \in P_{gen}; D_W \leq T\} \quad (1)$$
$$FA(T) = \{(X_{P1}, X_{P2}) \in P_{imp}; D_W \leq T\} \quad (2)$$

Here, the test samples identified as match pairs are shown by TP(T), and FA(T) are non-matching pairs that were mistakenly categorized as positive pairs. This calculates the False Acceptance Rate (FAR) and True Positive Rate (TPR) as:

$$TPR = \frac{TP(T)}{P_{gen}}, FAR = \frac{FA(T)}{P_{imp}} \qquad (3)$$

Data Representation

The MFCC features may be utilized as the data format for describing spoken utterances at the frame level. Because of the non-local features of the final DCT operation used to generate the MFCCs, the locality property is disturbed, which contrasts with the convolutional process's local characteristics. The log-energies, also known as the MFEC, are employed in this case, obviating the need for the DCT procedure.

For the generation of spectrum features, the temporal features are overlapping 20ms windows, with a stride of 10ms. 80 temporal feature sets, each form 40 MFEC features can be obtained from a 0.8 second sound sample, which forms the input speech feature map. Each input feature map has the dimensionality of $\tau \times 80 \times 40$ which is formed from 80 input frames and their corresponding spectral features, where τ is the number of utterances used in modeling the speaker during the development and enrollment stages. By default, we set $\tau = 20$.

The temporal features overlap 20 mm windows, with a 10 m stride, for generating spectrum features. A 0.8 second sound sample, which serves as the input speech feature map, yields 80 temporal feature sets, each of which contains 40 MFEC characteristics. Each input feature map has a dimensionality of $\tau \times 80 \times 40$, and it is made up of 80 input frames and associated spectral characteristics, where τ is the number of utterances utilized to represent the speaker during the development and enrolment stages. We set it to 20 by default.

The measure used in performance measurement is Equal Error Rate (EER) and is defined as the area where the False Rejection Rate and False Acceptance Rate are equivalent. Also, Area Under the Curve (AUC) was used for an example of precision that is the area under the ROC curve. The AUC for the proposed model is 97.91 %; AUC offers an aggregate performance measurement across all possible thresholds for classification. Table 2 demonstrates the comparison of the proposed work with the existing model. From the comparison, the proposed models produce the best results when compared to other existing methods.

Table 2 Comparison of proposed work with the existing model

Model	**EER**	**AUC**
3 D CNN Model	5%	91.6%
Hybrid CNN Model	2%	97.9%

Conclusion

In this paper, a hybrid CNN model for text-independent speaker verification with an utterance representative model is proposed. The proposed CNN model was trained to build up a feature extractor to capture the inter-speaker and intra-speaker variations. Here the background model for the speaker is built while learning the speaker characteristics in a one-shot representation technique. The proposed network significantly performs better than the existing 3D CNN model. With the proposed hybrid CNN model we have achieved the EER of 2%.

References

Das, R.K., Tian, X., Kinnunen, T., & Li, H. (2020). The attacker's perspective on automatic speaker verification: An overview. *arXiv preprint arXiv:2004.08849.*

Nagrani, A., Chung, J.S., Xie, W., & Zisserman, A. (2020). Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language, 60,* 101027.

Nidadavolu, P.S., Kataria, S., Villalba, J., Garcia-Perera, P., & Dehak, N. (2020). Unsupervised feature enhancement for speaker verification. *In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 7599-7603.

Li, X., Zhong, J., Wu, X., Yu, J., Liu, X., & Meng, H. (2020). Adversarial attacks on GMM i-vector based speaker verification systems. *In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 6579-6583.

Islam, M.R., & Rahman, M.F. (2015). An Approach of Multi-modal Biometric Iris and Speech based Person Recognition System with Decision Fusion Technique. *Journal of Multimedia Technology & Recent Advancements, 2*(2), 5-10.

Muda, L., Begam, M., & Elamvazuthi, I. (2010). Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv preprint arXiv:1003.4083.*

Shrawankar, U., & Thakare, V.M. (2013). Techniques for feature extraction in speech recognition system: A comparative study. *arXiv preprint arXiv:1305.1145.*

Alam, M.J., Kinnunen, T., Kenny, P., Ouellet, P., & O'Shaughnessy, D. (2013). Multitaper MFCC and PLP features for speaker verification using i-vectors. *Speech communication, 55*(2), 237-251.

Reynolds, D.A., Quatieri, T.F., & Dunn, R.B. (2000). Speaker verification using adapted Gaussian mixture models. *Digital signal processing, 10*(1-3), 19-41.

Campbell, W.M., Sturim, D.E., & Reynolds, D.A. (2006). Support vector machines using GMM supervectors for speaker verification. *IEEE signal processing letters*, *13*(5), 308-311.

Islam, M.R., & Rahman, M.F. (2010). Noise robust speaker identification using PCA based genetic algorithm. *International Journal of Computer Applications, 4*(12), 27-31.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25,* 1097-1105.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine, 29*(6), 82-97.

Lei, Y., Scheffer, N., Ferrer, L., & McLaren, M. (2014). A novel scheme for speaker recognition using a phonetically-aware deep neural network. *In IEEE international conference on acoustics, speech and signal processing (ICASSP),* 1695-1699.

Han, S., Pool, J., Tran, J., & Dally, W.J. (2015). Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626.*

Variani, E., Lei, X., McDermott, E., Moreno, I.L., & Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. *In IEEE international conference on acoustics, speech and signal processing (ICASSP),* 4052-4056.

Shannon, C.E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, *5*(1), 3-55.

Gurban, M., & Thiran, J.P. (2009). Information theoretic feature extraction for audio-visual speech recognition. *IEEE Transactions on signal processing*, *57*(12), 4765-4776.

Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., & Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing, 19*(4), 788-798.

Campbell, W.M., Sturim, D.E., & Reynolds, D.A. (2006). Support vector machines using GMM supervectors for speaker verification. *IEEE signal processing letters*, *13*(5), 308-311.

Garcia-Romero, D., & Espy-Wilson, C.Y. (2011). Analysis of i-vector length normalization in speaker recognition systems. *In Twelfth annual conference of the international speech communication association.*

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence, 35*(1), 221-231.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. *In Proceedings of the IEEE international conference on computer vision*, 4489-4497.

Abdel-Hamid, O., Mohamed, A.R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, *22*(10), 1533-1545.

Kataria, S., Nidadavolu, P.S., Villalba, J., Chen, N., Garcia-Perera, P., & Dehak, N. (2020). Feature enhancement with deep feature losses for speaker verification. *In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 7584-7588.

Xu, J., Wang, X., Feng, B., & Liu, W. (2020). Deep multi-metric learning for text-independent speaker verification. *Neurocomputing, 410,* 394-400.

Charan, R., Manisha, A., Karthik, R., & Kumar, M.R. (2017). A text-independent speaker verification model: A comparative analysis. *In International Conference on Intelligent Computing and Control (I2C2),* 1-6.

Salehghaffari, H. (2018). Speaker verification using convolutional neural networks. *arXiv preprint arXiv:1803.05427.*

Torfi, A., Dawson, J., & Nasrabadi, N.M. (2018). Text-independent speaker verification using 3d convolutional neural networks. *In IEEE International Conference on Multimedia and Expo (ICME),* 1-6.

Torfi, A., Dawson, J., & Nasrabadi, N.M. (2018). Text-independent speaker verification using 3d convolutional neural networks. *In IEEE International Conference on Multimedia and Expo (ICME),* 1-6. http://doi.org/10.21437/Interspeech.2019-1982

Mousa, I.R. (2019). Development of a numerical index to assess the quality of websites design. *Webology*, *16*(2), 72-82.