iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** V **Month of publication:** May 2020

DOI: http://doi.org/10.22214/ijraset.2020.5124

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

Voice Segregation using Fully Supervised RNN

Sebastian Thomas[1], Christie Joseph[2], Deepak Antony[3], Prof. Eldo P Elias[4]
[1, 2, 3, 4,] *Department of Computer Science and Engineering, Mar Athanasius College of Engineering*

Abstract: ***Human beings have a noteworthy behaviour to devote their attention to one person during a noisy atmosphere by mentally muting out all alternative sounds. This can be referred to as the cocktail-party effect and is seen naturally on humans. This project is meant to separate associate audio signals into its individual speaker sources in real-time.The main challenges faced during this problem are paying attention to voices i.e. cancelling out different background noises, modeling every separate speaker and calculating the quantity of speakers. These challenges are tackled by employing a combination of supervised recurrent Neural Network, mainly the unbounded Interleaved-State recurrent Neural Network (UIS-RNN) and VGG-Speaker-recognition. The UIS-RNN solves the matter of segmenting and clustering immense sequential information by learning from examples given. The VGG-Speaker-recognition is employed to spot the independent speakers from the audio input. The RNN is initially trained using a test dataset and test training set for optimum accuracy. Once the RNN is trained,it knows what to look for within the given input signal. The given system is absolutely supervised and is in a position to be told from examples wherever time-stamped speaker labels are annotated. The performance of this project is often considerably improved by d-vectors that are neural network embeddings. This mostly thanks to the very fact that neural networks can be trained with massive datasets, specified the model is sufficiently strong against variable speaker accents and acoustic conditions in several use situations.***

Index Terms: VGG, RNN, d-vectors, Ghost VLAD .

I. INTRODUCTION

The process of identifying the different speakers in a conversation is a critical task, with many underlying complications and this paper proposes a method to address this problem in an efficient manner. The challenges in implementing a speaker identification and separation system are but not limited to (1) A speech segmentation module that detects and removes noise; (2) Segmenting the audio clip into small segments so as to extract speaker discriminative embeddings such as d-vectors; (3) A clustering module that determines the number of speakers and assigns speaker identities to each segment; (4) A final segmentation module to refine the whole voice segregation process. Here we have used a RNN as the base of our model because it is one of the most promising algorithms in the recent year with an internal memory. This property of having an internal memory allows them to exhibit temporal dynamic behaviour for a time sequence which makes it ideal for processing sequences of input. Hence its widespread use in Speech Recognition modules and chosen as the base model for the network. For separating the speech from the non-speech part and to extract the speaker discriminative embeddings (d-vectors) we have employed Visual Geometry Group (VGG) - Speaker recognition. The number of speakers in the audio clip is not known beforehand. To accommodate the unknown number of speakers a probabilistic model called distance dependent Chinese restaurant process is used. A more detailed explanation of ddCRP and d-vectors are coming in the following sections. The model was trained in the VCTK dataset of the University of Edinburgh.

II. BASELINE ARCHITECTURE

A. VGG-Speaker Recognition

To recognize the different speakers in the audio clip, an utterance level aggregation is used. The utterances may be of different length and also contain background noise. We used a powerful speaker recognition deep network, using thin ResNet trunk architecture and a dictionary based GhostVLAD layer to aggregate features across time. The data set used to train this network is VoxCeleb1. This method was developed by Visual Geometry Group, Department of Engineering Science, University of Oxford, UK and we have adopted this speaker recognition model to generate speaker embeddings. See reference [3] for more details.

Previous methods of speaker recognition are pooling methods which have been successful in visual tasks. For instance, average pooling and fully connected layers to condense frame level information into utterance level representations. These methods ,although successful, have failed to identify the important parts of the signal or input which contain the useful information. This method uses a Convolutional Neural Network (CNN) and a dictionary based GhostVlad layer , where the CNN captures the patterns in the signal and the GhostVLAD can be trained for creating a fixed size descriptor for each speaker despite the arbitrary size of the

input signal. The speaker recognition model should have the following properties : (1) It should produce a fixed-length descriptor for each speaker despite the input segments being of different time lengths. (2) The descriptor produced as output should have a smaller size so that the process becomes memory efficient and can be stored in the memory and retrieved from the memory faster. (3) The descriptors of the same speaker, if more than one are produced, should have similar properties and the descriptors of different speakers should have distinct properties. To achieve these properties, a modified ResNet is used in a fully convolutional way to encode the 2D spectrograms that are given as input, which is followed by a GhostVLAD layer for feature aggregation along the temporal axis. The output is a fixed length descriptor. Finally a fully connected layer is added to reduce dimensionality, thus making the similarity computation significantly faster. The initial part of the process is feature extraction from the input spectrograms. The network used is a modified ResNet with 34 layers. The number of channels in each block was cut down thus making it a thin ResNet-34. The core idea of a ResNet is a shortcut connection that skips one or more layers. The layers that do nothing are stacked together and the resulting architecture should perform the same.

The next part of the process is to aggregate frame-level descriptors into a single vector. This part is a GhostVLAD layer. The GhostVLAD model was proposed by Y. Zhong [1]. GhostVLAD works similar to NetVLAD [2]. The NetVLAD is in turn, an extension of VLAD. In a NetVLAD, the hard assignment based clustering is replaced with a soft assignment based clustering. The thin ResNet maps the input spectrogram(R257T1) to frame-level descriptors with size R1T/32512. The NetVLAD layer then takes dense descriptors as input and produces a single KD matrix V , where K refers to the number of chosen clusters, and D refers to the dimensionality of each cluster. Concretely, the matrix of descriptors V is computed using the following equation:

$$V(k,j) = \sum_{t=1}^{T/32} \frac{e^{w_k x_t + b_k}}{\sum_{k'=1}^{K} e^{w'_k x_t + b^{k'}}} (x_t(j) - c_k(j)) \qquad (1)$$

where {wk}, {bk} and {ck} are trainable parameters, with k ∈ [1, 2, ..., K]. The first term corresponds to the soft assignment weight of the input vector xi for cluster k, while the second term computes the residual between the vector and the cluster centre.

GhostVLAD works in the same manner except that it has some ghost clusters along with the NetVLAD clusters. Thus it has K+G clusters, where G is the number of ghost clusters. These are added so that any noise or irrelevant signal is mapped to these ghost clusters as they are not included during feature aggregation. Which means that the matrix V is computed for both the normal K clusters and for the G ghost clusters, but the vectors belonging to the ghost clusters are not included during feature concatenation. The final output is obtained by performing L2 normalisation and concatenation. To keep computational and memory requirements low, dimensionality reduction is performed via a Fully Connected (FC) layer, where we pick the output dimensionality to be 512.ddg

B. RNN Model

As mentioned in the introduction we have used a Recurrent Neural Network, RNN as our base model because of its property of having an internal memory which makes it best suited for Speech Recognition modules. The input to the RNN is speaker discriminative embeddings called d-vectors. They can be considered as voice fingerprints. They are unique to each and every speaker. Also d-vectors prove to be useful for handling very large datasets since they are generated by neural networks. Due to the fact they can be trained with very large datasets, the model is robust against varying speaker accents. Given an utterance from VGG module, we get an observation sequence of embeddings X = (x1, x2, x3, …. xn). Each value in this sequence is a real-valued d-vector corresponding to a segment in the original input utterance. Since the model is supervised we also have a truth label for each segment Y = (y1, y2, y3, …. yn) for each segment.

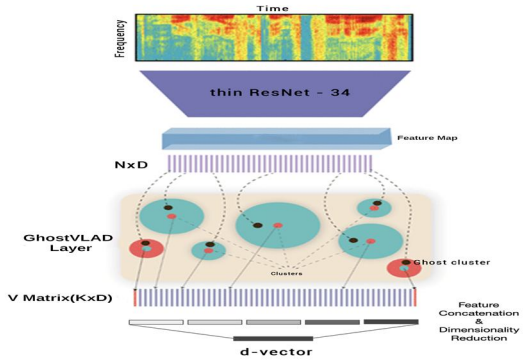


Fig. 1. Architecture of the VGG recognition network

For example, Y = (1, 2, 2, 1, 3, 3) means this utterance has six segments, from three different speakers, where y_n = m means segment n belongs to the speaker *m*. .

We can say that the RNN is an generative process of an entire input utterance (X, Y), where

$$p(\mathbf{X}, \mathbf{Y}) = p(x_1, y_1) \cdot \prod_{n=2}^{T} p(x_n, y_n | x_{[n-1]}, y_{[n-1]}) \quad (2)$$

And to model speaker changes, we use the representation given below

$$p(\mathbf{X},\mathbf{Y},\mathbf{Z}) = p(x_1, y_1) \prod_{n=2}^{T} p(x_n, y_n, z_n | x_{[n-1]}, y_{[n-1]}, z_{[n-1]}) \quad (3)$$

where Z = (z1, z2, z3, …. zn) is a binary indicator of speaker changes. Say, Y = (1, 2, 2, 1, 3, 3), then Z is (1, 0, 1, 1, 0).

1) *Speaker Assignment Process:* One of the main challenges in separating the speakers in the voice segment is to determine the total number of speakers in each utterance. For this we use a probabilistic model called distance dependent Chinese Restaurant Process (ddCRP). ddCRP is commonly used in data clustering. It is a Bayesian non-parametric model that can be used to model unbounded number of speakers Another reason this model is used to accommodate speakers is that it is highly scalable and can be used in distributed environments. In the above section we have introduced a tuple that indicates speaker changes, specifically when zn = 0 the speaker remains unchanged. When zn = 1, we have two probabilities as mentioned below

$$p(y_n = k | z_n = 1, y_{[n-1]}) \propto N_{k,n-1},$$

$$p(y_n = K_{n-1} + 1 | z_n = 1, y_{[n-1]}) \propto \alpha. \quad (4)$$

Here we can see the probability of the switching back to a previously appearing speaker which is directly proportional to the number of continuous speeches he/she has spoken and the probability that a new speaker was introduced which is directly proportional to a constant **α**..

2) *Sequence Generation:* In this paper we use the Gated Recurrent Unit(GRU) as our RNN model. GRU is used to memorize the long dependencies and to solve the vanishing gradient problem seen in standard RNN. The accuracy of RNN or any other neural network lies in the number of the hidden layers they have. As the number of hidden layers increase, the accuracy of the network increases and at the same time becomes increasingly complex. An important step in the training of RNN is back propagation. But having many hidden layers can have an adverse effect on back propagation. When moving backwards in the network to calculate the gradients of loss with respect to the weights, the gradients tend to get smaller as we keep on moving backwards on the network. Because of this the earlier layers in the network learn very slowly as compared to the later layers in the network. Since the earlier layers of the network are actually the building blocks of the network, this can cause the training process to take too long and obviously lead to inaccurate results. This is the vanishing gradient problem. What GRU does is that it uses a series of logical gates to regulate the flow of information through the network. It only transfers the relevant information from one layer to another, making the training in the earlier networks much more efficient and fast. When GRU is coupled with an activation function such as RELU(the activation function used in this paper), the effect of vanishing gradient problem is negligible. In this paper we assume that the observation sequence of the speaker embeddings, X is generated by distributions that are parameterized by the output of the network. The network has multiple instantiations, wrt to each speaker and they all share the same parameters **θ**.

The output of the entire network at time t can be said as

$$\boldsymbol{m}_{\boldsymbol{n}} = f(n|\theta) \quad (5)$$

where ht is the state of GRU corresponding to a specific speaker y_t.

3) *MLE Estimation:* Maximum likelihood estimation(MLE) is used to determine the values of parameters of a model i.e. it is a method to update the parameters in each iteration of the model. The parameter values calculated by this approach are such that they maximize the likelihood that the process described by the model produced the data that were actually observed.

The updations of the parameters **θ** and **α** using stochastic gradient ascent by randomly selecting a subset B(τ) are shown below

$$\theta^{(\tau)} = \theta^{(\tau-1)} + \frac{N\rho^{(\tau)}}{b} \sum_{n \in B^{(\tau)}} \nabla_{\theta} \ln p(X_n | Y_n, Z_n, \theta, -), \quad (6)$$

For , we update

$$\alpha^{(\tau)} = \alpha^{(\tau-1)} + \frac{N\rho^{(\tau)}}{b} \sum_{n \in B^{(\tau)}} \nabla_{\alpha} \ln p(X_n | Y_n, Z_n, \alpha, -), \quad (7)$$

4) *Decoding:* The decoding layer is the final layer of the network. This layer decodes the probability distribution of the next possible speaker. In this model beam search decoding is used. Beam search decoding works by selecting one of the multiple alternatives possible using conditional probability. The error rate of the decoding can be regulated using the width property of beam search. To optimize the results, the width is set to 10.

III. CONCLUSION

In this paper, we try to introduce a voice segregation system that can automate the process of transcribing. The model we described here can be used to get the final transcription of a meeting and other modes of group communication. With further optimizations, it could lead the way to live captioning in videos highlighting the multiple speakers involved.

Here we introduce a fully supervised RNN model for speaker segregation. Since it is fully supervised it can be used instead of unsupervised methods and can provide better results. The output embeddings can be easily coupled to an audio to speech converter to get the final text transcript and makes our voice segregation model end-to-end application.

IV. ACKNOWLEDGEMENT

We express our sincere thanks to our Department and our Institute for fostering a superb academic atmosphere Which made this endeavor fruitful.

We tend to express sincere gratitude to prof. Eldo P Elias. We want to acknowledge his help in making our task simple by giving us his valuable advice and encouragement. We would be very pleased to express our heart full Thanks to the teaching and non-teaching staff of the department of Computer Science Engineering, Mar Athanasius College of Engineering for their motivation and support.

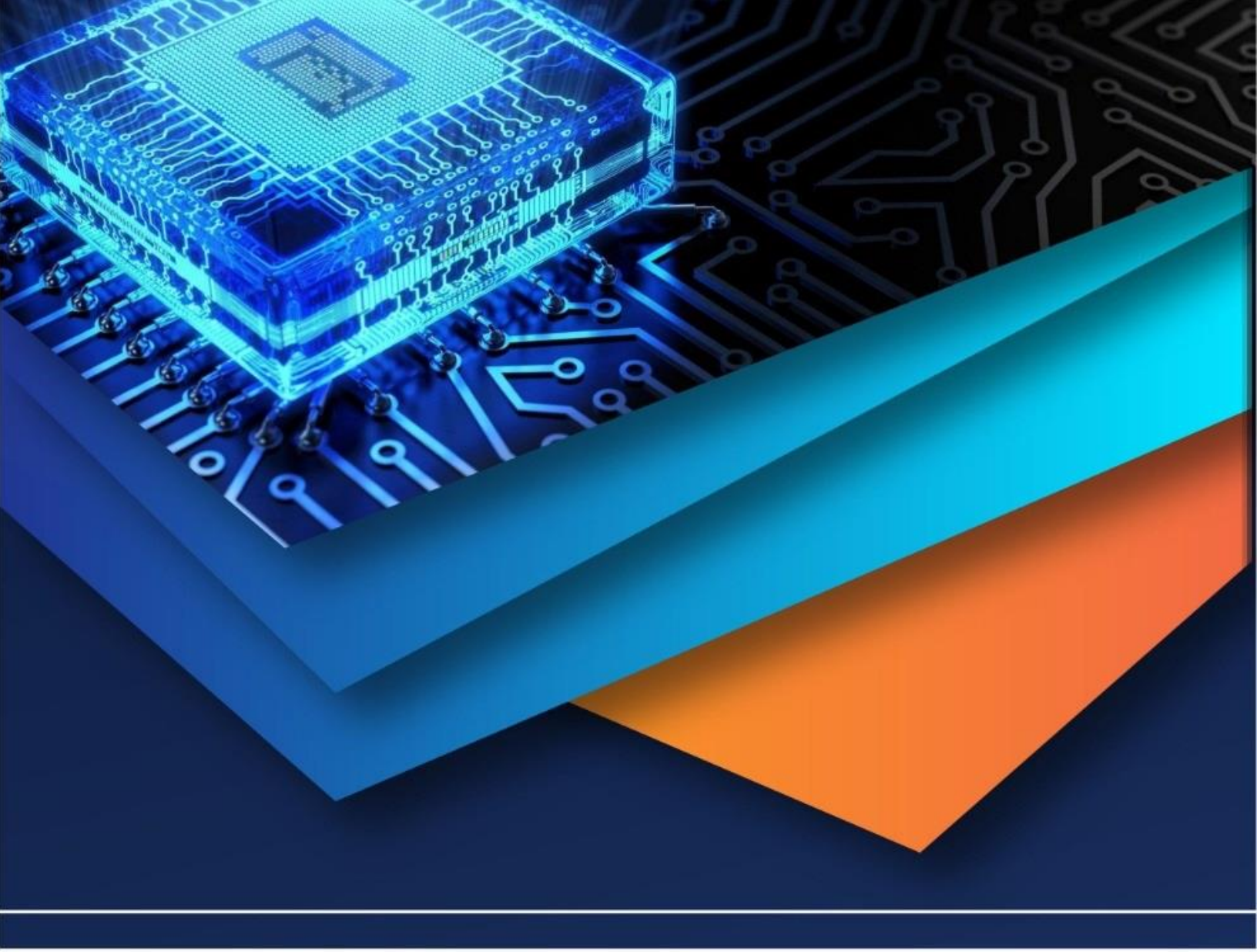
REFERENCES

[1] Y. Zhong, R. Arandjelovic, and A. Zisserman,"GhostVLAD for set-based face recognition," in AsianConference on Computer Vision, ACCV, 2018.

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition" in Proc. CVPR, 2016

[3] Weidi Xie, Arsha Nagrani, Joon Son Chung, Andrew Zisserman, Utterance-level Aggregation For Speaker Recognition In The Wild [4] Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, Chong Wang, Fully Supervised Speaker Diarization arXiv:1810.04719v7, 2019

[4] Miao, Xiaodong, Shunming Li, and Huan Shen. "ON-BOARD LANE DETECTION SYSTEM FOR INTELLIGENT VEHICLE BASED ON MONOCULAR VISION." International Journal on Smart Sensing In-telligent Systems 5.4 (2012).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY