Kernel based Matching and a Novel training approach for CNN-based QbE-STD

Prajyot Naik
Dept. of CSE, NIT Goa
prajyotnaik3@gmail.com

Manisha Naik Gaonkar
Dept. of CSE, NIT Goa
manisha@gec.ac.in

Veena Thenkanidiyoor
Dept. of CSE, NIT Goa
veenat@nitgoa.ac.in

Dileep A. D.
SCEE, IIT Mandi
addileep@iitmandi.ac.in

***Abstract*—Query-by-Example based spoken term detection (QbE-STD) to audio search involves matching an audio query with the reference utterances to find the relevant utterances. QbE-STD involves computing a matching matrix between a query and reference utterance using a suitable metric. In this work we propose to use kernel based matching by considering histogram intersection kernel (HIK) as a matching metric. A CNN-based approach to QbE-STD involves first converting a matching matrix to a corresponding size-normalized image and classifying the image as relevant or not [6]. In this work, we propose to train a CNN-based classifier using size-normalized images instead of splitting them into subimages as in [6]. Training approach proposed in this work is expected to be more effective since there is less chance of a CNN based classifier getting confused. The effectiveness of the proposed kernel based matching and novel training approach is studied using TIMIT dataset.**

I. Introduction

Audio search refers to searching and retrieving relevant utterances from an audio database. This involves matching a query with the reference utterances in the database. Conventionally, a text query is matched with transcriptions of reference utterances to retrieve the relevant ones [2]. This requires a robust automatic speech recognizer (ASR). Building an ASR requires large amount of annotated data for training and may not be suitable for under resourced languages. To address this issue, an audio query is used in Query-by-Example spoken term detection (QbE-STD) approach to audio search. Approaches to QbE-STD involve matching a query utterance with reference utterances, after representing them suitably.

Conventionally, dynamic time warping (DTW) distance based matching approaches are considered for QbE-STD [1], [2], [3]. Here, DTW distance matrices are computed between query utterance and reference utterances. The relevant utterance for a query is found by finding a warping path in the respective DTW distance matrix. Various variants of DTW-based approaches have been proposed like segmental DTW [2], [3], [4], slope constrained DTW [5] to improve the performance of DTW-based approaches. In segmental DTW-based approach, query matching is done by dividing test utterances into segments which are overlapping and are of the same length as the query [2], [3], [4]. In slope constrained DTW-based approach a constraint on the slope of the warping path is considered [5]. In constrained-endpoint DTW-based approach the restrictions on the endpoint of the warping path are enforced [2]. Finding warping path in a DTW distance matrix is a computationally intensive task.

To address this issue, recently Convolutional Neural Network (CNN) based approaches are explored for determining the relevant utterances for a query [6]. These approaches involve computing a suitable matching matrix between a query and reference utterances using metrics like DTW distance, Kullback-Leibler divergence (KLD), etc. [6], [7]. The matching matrix is then visualized as an image. Image corresponding to a matching matrix computed between a query utterance and a reference utterance is expected to exhibit a diagonal pattern if the query matches with that reference utterance. This pattern is used to discriminate a relevant utterance for a query from an irrelevant utterance. In [6], a CNN-based classifier is used to classify an image corresponding to a matching matrix as a relevant or an irrelevant image to a given query. An important issue in this approach is that, the images generated may be of variable sizes depending on the size of the query utterances and the reference utterances. Another issue is that, the query utterance may appear in a small subset of reference utterances. This will result in class imbalance problem. These issues are addressed in [6] by size normalizing and data augmentation. However, size normalized images are divided into non overlapping subimages which are used for training a CNN-based classifier. All subimages of a size normalized image are assigned with the same label (positive or negative) as that of the size normalized image. This may confuse a CNN-based classifier. To avoid this, we propose a novel training approach which directly uses the size normalized images for training. The proposed novel approach to training a CNN-based classifier is expected to perform better when compared to the CNN-based classifier trained using subimages proposed in [6].

Matching matrix is very important in the proposed approach. Matching matrix is computed using suitable similarity or dissimilarity measures. In [6], KLD, a dissimilarity measure is used for computing the matching matrix. In this work we propose to consider kernel based matching. Kernel functions correspond to inner product between two examples in the kernel induced space. Value of a kernel function can be considered as a measure of a similarity. In this work, we propose to represent an utterance using Gaussian posteriorgrams. Gaus-

978-1-7281-8895-9/20/$31.00

sian posteriorgrams being discrete probability distributions, we propose to use histogram intersection kernel (HIK) [8] for computing matching matrix between a query utterance and a reference utterance. Computation of a matching matrix comprising of HIK values is computationally less expensive than using KLD. Hence, the matching matrix computed using the proposed approach is expected to be computationally efficient in QbE-STD task as we will see in following section.

The rest of the paper is organized as follows: Section II describes the proposed method in detail. Section III summarizes the experiments and results while section IV concludes the paper and mentions its future scope.

II. Proposed Approach

In this section, we present the proposed approach. We first present the representation used. Then, we present the matching matrix computation and visualization, followed by CNN-based classifier training.

A. Feature Extraction and Representation

We propose to represent every frame of speech using a 39-dimensional Mel-Frequency Cepstral Coefficients (MFCC). We propose to build a Gaussian Mixture Model (GMM) of Q components using the MFCC feature vectors from all the utterances in the training set of database. This GMM is used to obtain a Q-dimensional Gaussian posteriorgram representation for every frame. Each element in a Gaussian posteriorgram vector represents the probability of a 39-dimensional MFCC vector coming from one of the Q GMM components.

B. Matching matrix computation

Let N frames of a query utterance be represented by respective posteriorgrams, $\boldsymbol{u_1}, \boldsymbol{u_2}, ..., \boldsymbol{u_n}, ..., \boldsymbol{u_N}$, where $\boldsymbol{u_n}$ ϵ $\mathbb{R}^{\mathbb{Q}}$. Let a reference utterance be represented by M frames, $\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_m}, ..., \boldsymbol{v_M}$, where $\boldsymbol{v_m}$ ϵ $\mathbb{R}^{\mathbb{Q}}$. A matching matrix computed between them, $\mathbf{K}$ is a $N \times M$ dimensional matrix such that,

$$\mathbf{K}(n, m) = mat(\boldsymbol{u_n}, \boldsymbol{v_m}) \tag{1}$$

where $mat(\boldsymbol{u_n}, \boldsymbol{v_m})$ is a suitable matching score between $\boldsymbol{u_n}$ and $\boldsymbol{v_m}$. In [6], Kullback-Leibler divergence (KLD) is used to compute a matching matrix. KLD is a quantitative measure of how one probability distribution is different from a second probability distribution. Since, Gaussian posteriorgrams are discrete probability distributions, KLD can be considered as a matching score. KLD between $\boldsymbol{u_n}$ and $\boldsymbol{v_m}$ is given by:

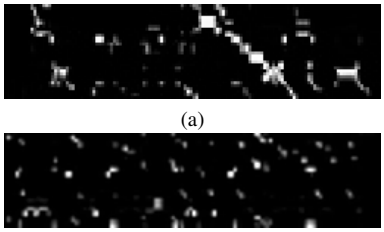
$$KLD(\boldsymbol{u_n}, \boldsymbol{v_m}) = \sum_{l=1}^{Q} u_{nl} \ln \frac{u_{nl}}{v_{ml}} \tag{2}$$

The KLD in (2) is not symmetric in nature. The symmetric KLD is defined as:

$$SKLD(\boldsymbol{u_n}, \boldsymbol{v_m}) = \sum_{l=1}^{Q} u_{nl} \ln \frac{u_{nl}}{v_{ml}} + \sum_{l=1}^{Q} v_{ml} \ln \frac{v_{ml}}{u_{nl}} \tag{3}$$

We propose to consider SKLD as a matching score as:

$$\mathbf{K}(n, m) = SKLD(\boldsymbol{u_n}, \boldsymbol{v_m}) \tag{4}$$



(a)

(b)

Fig. 1: Illustration of images corresponding to HIK-based matching matrix (a) when there is a match between query and reference utterance (b) when there is no such match.

In this work, we also propose to use a kernel function, a measure of similarity as a matching score. The Gaussian posteriorgrams being non-negative, histogram representation, we propose to use histogram intersection kernel (HIK) for matching two Gaussian posteriorgrams. HIK between $\boldsymbol{u_n}$ and $\boldsymbol{v_m}$ is computed as:

$$HIK(\boldsymbol{u_n}, \boldsymbol{v_m}) = \frac{1}{Q} \sum_{l=1}^{Q} min(u_{nl}, v_{ml}) \tag{5}$$

In this work, we propose to consider:

$$\mathbf{K}(n, m) = HIK(\boldsymbol{u_n}, \boldsymbol{v_m}) \tag{6}$$

It is seen from equations (3) and (5) that HIK needs less number of computations as compared to SKLD computation.

C. Visualising a matching matrix

A matching matrix $\mathbf{K}$ corresponding to a reference utterance where a query utterance is present exhibits a diagonal pattern having very small matching scores (in case of dissimilarity or distance measures) or very high matching scores (in case of similarity measures). In case of non-matching cases, no such diagonal pattern appears. It is a difficult task to find this pattern in the matching matrix. Hence, we visualize matching matrix as an image. To convert the matching matrix to an image, we normalize the matching scores in a matrix and a grayscale image is created for matching matrix that shows the diagonal pattern in case of a match. This is illustrated in Figure 1, for HIK-based matching metric. It is seen in Figure 1(a), there is a diagonal pattern that indicates a match between a query and that portion of reference utterance. However, in Figure 1(b), there is no such pattern since the query does not match with its reference utterance. M varies for different utterances. We resize the normalized images to 32×128, so that all the generated images have same dimensions [6].

D. CNN-based Classifier

Recently CNNs are found to be effective in image classification. In this work, we propose to use CNNs to classify an image to decide whether the corresponding reference utterance is relevant or irrelevant for a query utterance. Figure 4 shows the proposed CNN architecture that comprises of 3 convolution layers. Each convolution layer is followed by a max-pooling

layer. The hidden layer consists of 300 nodes and is followed by a 2-node output layer. The details of every convolution and max pooling layers are given in Table I. Negative log likelihood is used as an error function. In [6], size normalized images are divided into non-overlapping subimages of size 32×32. Figures 3 and 4 show how the images in Figure 1(a) and 1(b) would be divided into subimages as proposed in [6]. All the subimages in Figure 3 shall be labeled as positive class, but only the third subimage contains the diagonal pattern, that we are interested in. Thus, labeling other three subimages as positive class will increase the confusion for any classifier and might even lead to learning different uninteresting patterns. For example, image in Figure 3(a) is given a positive class label and image in Figure 4(a) is given a negative class label in [6]. However, images in Figures 3(a) and 4(a) are similar. To avoid this confusion, in this work we propose not to divide an image into subimages. Instead, we propose to use 32×128 sized images to train CNN-based classifier as shown in Figure 2. Since chances of confusion to the classifier are less the proposed approach to training is expected to be more effective.

III. Experimental Studies

In this section we present the studies conducted on proposed approach. In following subsections we present the dataset used, the experimental setup and studies conducted.

A. Dataset

TIMIT corpus has been used for conducting studies in this work. It is divided into train and test sets consisting of 4320 and 1680 speech utterances respectively. Both sets contain data from 8 different dialects, and 600 different speakers. A set of 21 keywords of varying length have been chosen from the TIMIT corpus. The chosen keywords are referred to as query utterances. Table II shows the 21 keywords considered in this work. 7 templates have been randomly chosen for each keyword from test set and used for creating matching matrices and the size normalized images [6].

Every utterance is represented by Mel-Frequency Cepstral Coefficients (MFCC) extracted from 25 ms frame, with 15 ms shift. Every frame is represented by 39-dimensional MFCC features. To represent a frame using Gaussian posteriorgram, Gaussian mixture model (GMM) is built using MFCC features extracted from all the training utterances. In this work, we explored building GMMs with components 32, 64 and 128 respectively. These GMMs are used to obtain Gaussian posteriorgrams of dimensions 32, 64 and 128 respectively, for the utterances.

B. Experimental setup

For each of the 7 randomly selected template of 21 keywords, matching matrices are computed with reference utterances containing this keyword and an equal number of reference utterances not containing this keyword. Matching scores are computed using SKLD (equation (3)) and HIK (equation (5)). All these matrices are converted to grayscale, size normalized images of size 32×128. These two sets of images form two classes: positive (representing relevant utterances) and negative (representing not relevant utterances). These two classes of images are further divided into new train and test sets of images, such that images for all the keywords are present in both training and test sets. A part of training set is used as validation set.

In this work we propose to use CNN-based classifier to find relevant utterances for a query utterance. Architecture of CNN is chosen as given in Table I. Number of convolutional layers is fixed to 3. Size of feature maps has also been fixed as given in the architecture. Number of feature maps and number of units in hidden layer is decided using cross validation. Negative log likelihood is used as error function. CNN-based classifier is implemented using Keras library with TensorFlow backend. In following subsection we refer to this CNN architecture as our proposed CNN architecture for the task.

C. Studies on proposed approaches

First, we compare proposed approach with the approach considered in [6]. CNN architecture in [6] consists of 3 convolution layers equivalent to our CNN. However, in [6] only two pooling layers are used following the second and third convolution layers. Fully connected layer consists of 100 units in [6] and output layer has 2 units. We have used tanh as activation function for maintaining consistency while comparing the approach with [6]. We consider Gaussian posteriorgram representation of 32, 64 and 128-dimensions obtained using respective GMMs. Table III describes the architecture used, size of Gaussian posteriorgram and matching metric used for different CNNs. Here CNN1, CNN2, CNN3, CNN7, CNN8, and CNN9 correspond to proposed architecture and CNN4, CNN5, and CNN6 correspond to the architecture proposed in [6]. In Table IV, we compare the performance of proposed CNN-based classifiers (CNN1, CNN2 and CNN3) with the ones proposed in [6] (CNN4, CNN5 and CNN6). It is seen from Table IV that proposed approach has lower false alarm rate, lower missing rate and lower total error. This shows the effectiveness of the proposed approach. Figure 5 shows the training and validation loss plots during training of CNNs. Training loss for proposed approach is much less and stabilises even before 40 epochs of training. Validation loss curves show that 64-component GMM attains lower loss very quickly during initial epochs, indicating Gaussian posteriorgrams from 64-component GMM work better for the task. This shows the effectiveness of the proposed training approach.

Next we compare kernel based matching with the SKLD for the computation of matching matrix. In this study we propose to use histogram intersection kernel (HIK) for computation of matching matrix. In Figure 6 we compare the performance of proposed CNN-based classifier using HIK with CNN-based classifier that uses SKLD. It is seen from Figure 6 that HIK has lower false alarm rates but higher missing rate than SKLD. However, total error rate for HIK is found to be less than that for SKLD. Also, computation of HIK has less number of computations when compared to that required for computing

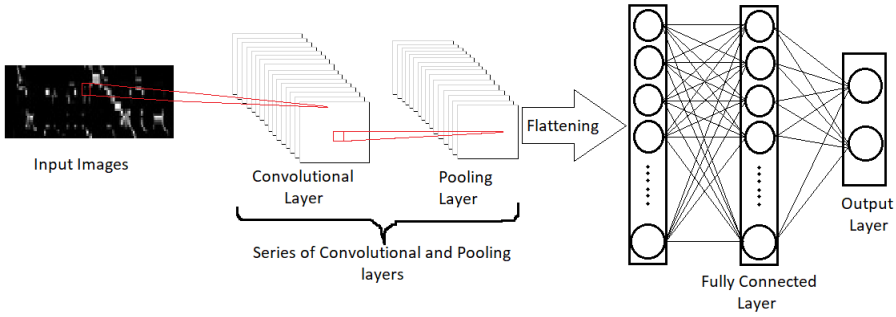


Fig. 2: The proposed CNN Architecture.

Layer Name	Details	Activation	Output shape	Parameters
Convolution Layer 1	40 feature maps, 5×5, no padding	tanh	(None, 28, 124, 40)	3040
Maxpooling layer 1	2×2 window, no overlapping	-	(None, 14, 62, 40)	0
Convolution Layer 2	30 feature maps, 3×3, no padding	tanh	(None, 12, 60, 30)	10830
Maxpooling layer 2	2×2 window, no overlapping	-	(None, 6, 30, 30)	0
Convolution Layer 3	50 feature maps, 2×2, no padding	tanh	(None, 5, 29, 50)	6050
Maxpooling layer 3	2×2 window, no overlapping	-	(None, 2, 14, 50)	0
Flattening Layer	Flattens output from previous layer into a vector	-	(None, 1400)	0
Fully Connected Layer	300 units	tanh	(None, 300)	420300
Output Layer	2 units	softmax	(None, 2)	602

TABLE I: Details of proposed CNN Architecture.

Artists	Beautiful	Carry	Breakdown
Greasy	Development	Wash	Hostages
Children	Like-that	Darksuit	Lunch
Money	Oilyrag	Popularity	Problem
Organizations	Review	Water	Warm
Woolen			

TABLE II: List of Keywords used.

Abbreviation	Size of Gaussian posteriorgram	Matching Metric	Architecture
CNN1	32	SKLD	Proposed CNN
CNN2	64		
CNN3	128		
CNN4	32	KLD as in [6]	As in [6]
CNN5	64		
CNN6	128		
CNN7	32	HIK	Proposed CNN
CNN8	64		
CNN9	128		

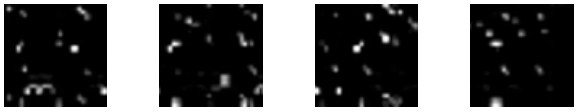
TABLE III: Descriptions for various CNN architectures.

	False Alarm Rate	Miss Rate	Total Error Rate
CNN1	0.079	0.026	0.053
CNN2	0.070	0.040	0.055
CNN3	0.082	0.025	0.054
CNN4	0.211	0.122	0.167
CNN5	0.151	0.140	0.146
CNN6	0.171	0.122	0.147

TABLE IV: Performance comparison of proposed approach with the approach in [6].

(a) (b) (c) (d)

Fig. 3: Illustration of dividing image in Figure 1(a) into four non-overlapping sub-images[6].



(a) (b) (c) (d)

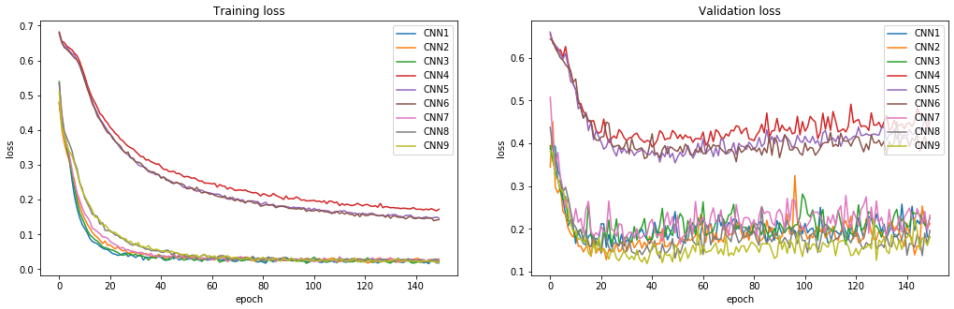
Fig. 4: Illustration of dividing image in Figure 1(b) into four non-overlapping sub-images[6].

SKLD. This shows the effectiveness of the proposed kernel based matching for QbE-STD.

From Table IV and Figure 6, it can be inferred that overall performance of SKLD doesn't vary much with variation in number of GMM components used. However, for KLD as in [6] and HIK metrics, 64 component GMM worked best, followed by 128 component GMM. This implies that, 39-dimensional MFCCs are better learnt by 64 component GMM then 32 or 128 component GMM.

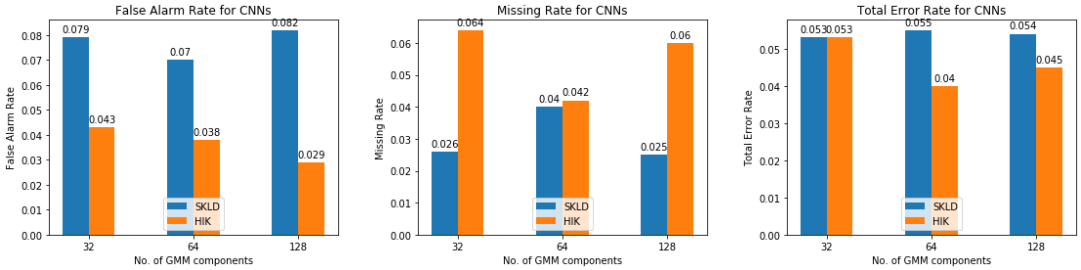
IV. Summary and Conclusions

In this paper, a novel approach to training a CNN-based classifier for audio search using matching-based QbE-STD is proposed. QbE-STD involves matching an audio query with the reference utterances. In matching based QbE-STD, a matching matrix corresponding to a query and a reference



(a) (b)

Fig. 5: (a) Training loss plot, (b) Validation loss plot, for proposed approach and approach in [6]



(a) (b) (c)

Fig. 6: Performance comparison between SKLD (CNN1, CNN2, CNN3) and HIK (CNN7, CNN8, CNN9).

utterance is computed using a suitable metric. Recently kernel based matching is found to be effective. Kernel based matching involves using value of a suitable kernel function as a matching score. In this work we proposed to represent an utterance using Gaussian posteriorgram. We also proposed to consider histogram intersection kernel (HIK) as a metric for obtaining a matching matrix. The matching matrix is converted to a size-normalized image and a CNN-based classifier is used to decide whether the corresponding reference is a relevant utterance for the query. In [6], a size normalized image is split into subimages on which a CNN-based classifier is trained. This approach may confuse the classifier. To address this issue, in this work, we propose to train the CNN-based classifier using size normalized images. This will help the classifier to better discriminate a relevant utterance from an irrelevant utterance. The studies conducted on standard dataset shows the effectiveness of the proposed approach.

REFERENCES

[1] M. Müller, "Dynamic time warping," in *Information retrieval for music and motion*, Springer, Berlin: Heidelberg, pp. 69-84, 2007.

[2] L. Mary and G. Deekshitha, *Searching Speech Databases: Features, Techniques and Evaluation Measures*, Springer, 2018.

[3] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in Proceedings of *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pp. 398–403, 2009.

[4] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[5] T. J. Hazen, W. Shen and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in Proceedings of *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pp. 421–426, 2009.

[6] R. Shankar, C.M. Vikram and S.R.M. Prasanna, "Spoken Keyword Detection using joint DTW-CNN," in Proceedings of *INTERSPEECH 2018*, pp. 117-121, 2018.

[7] D. Ram, L. M. Werlen and H. Bourlard, "CNN based Query by Example Spoken Term Detection," in Proceedings of *INTERSPEECH*, pp. 92-96, 2018.

[8] A. Sharma, A. Kumar, S. Allappa, V. Thenkanidiyoor, D. A. Dinesh and S. Gupta, "Modified Time Flexible Kernel for Video Activity Recognition using Support Vector Machines," In Proceedings of *7th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pp. 133-140, 2018.

[9] S. Myer and V. S. Tomar, "Efficient keyword spotting using time delay neural networks," in Proceedings of *INTERSPEECH 2018*, pp. 1264-1268, 2018.

[10] H. Benisty, I. Katz, K. Crammer and D. Malah, "Discriminative Keyword Spotting for limited-data applications," *Speech Communication*, Volume 99, pp. 1-11, 2018.

[11] W. Shen, C. M. White and T. J. Hazen, "A Comparison of Query-by-Example Methods for Spoken Term Detection," in Proceedings of *INTERSPEECH 2009*, pp. 2143-2146, 2009.

[12] Z. Zhu, Z. Wu, R. Li, H. Meng and L. Cai, "Siamese Recurrent Auto-Encoder Representation for Query-by-Example Spoken Term Detection," in Proceedings of *INTERSPEECH 2018*, pp. 102-106, 2018.

[13] C. Parada, A. Sethy and B. Ramabhadran, "Query-by-Example Spoken Term Detection For OOV Terms," in Proceedings of *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pp. 404-409, 2009.

[14] Y. Yuan, C. C. Leung, L. Xie, H. Chen, B. Ma and H. Li, "Learning Acoustic Word Embeddings with Temporal Context for Query-by-Example Speech Search," in Proceedings of *INTERSPEECH 2018*, pp. 97-101, 2018.

[15] S. Settle, K. Levin, H. Kamper and K. Livescu, "Query-by-Example Search with Discriminative Neural Acoustic Word Embeddings," in Proceedings of *INTERSPEECH 2017*, pp. 2874-2878, 2017.

[16] P. Raghavendra Reddy, K. Sri Rama Murty and B. Yegnanarayana, "Representation Learning for Spoken Term Detection," *Pattern Recognition and Big Data*, World Scientific Publishing Co. Pte. Ltd, Chapter 19, 2016.