Interspeech 2018
2-6 September 2018, Hyderabad

Spoken Keyword Detection using joint DTW-CNN

Ravi Shankar[1], *C.M. Vikram*[2], *and S.R.M. Prasanna*[2,3]

[1]Johns Hopkins University, Baltimore
[2]Indian Institute of Technology Guwahati
[3]Indian Institute of Technology Dharwad

rshanka3@jhu.edu, cmvikram@iitg.ac.in and prasanna@iitdh.ac.in

Abstract

A method to detect spoken keywords in a given speech utterance is proposed, called as joint Dynamic Time Warping (DTW)-Convolution Neural Network (CNN). It is a combination of DTW approach with a strong classifier like CNN. Both these methods have independently shown significant results in solving problems related to optimal sequence alignment and object recognition, respectively. The proposed method modifies the original DTW formulation and converts the warping matrix into a gray scale image. A CNN is trained on these images to classify the presence or absence of keyword by identifying the texture of warping matrix. The TIMIT corpus has been used for conducting experiments and our method shows significant improvement over other existing techniques.
Index Terms:DTW, CNN, keyword detection.

1. Introduction

Spoken keyword detection is a task which aims to detect the occurrence of a particular word or sequence of words in a given speech utterance. A widely used approach for keyword detection is to use an automatic speech recognizer (ASR) for creating word lattices and then indexing them using weighted finite state transducer (WFST) [1, 2, 3]. A textual search is then performed on the indexed word lattices. The problem with ASR lies in the amount of hand labelled data required for training which is very expensive.

Other approaches rely on pattern matching schemes such as dynamic time warping (DTW) [4, 5] to get a similarity score between the keyword template and a given speech utterance [6, 7]. Gaussian posteriorgrams [8] are common feature representation used in these algorithms. However, pattern matching methods are dependent on thresholds chosen for accepting or rejecting a keyword. The optimization of these threshold values is a difficult task as a single value does not fit all the keywords. A solution is to use keyword specific thresholds, but it requires *a priori* knowledge of the keyword being searched. To address these problems we are proposing an innovative approach to combine a variation of DTW with a binary classifier.

The motivation for the current approach comes from the observation that warping matrices exhibit a temporal relationship between two given sequences in its texture (when visualized as an image). This idea was explored in [9] for keyword detection. However, the features used in [9] are derived from an artificial neural network (ANN) model which requires training on phonetically transcribed data. Besides, the method is also not completely threshold independent as it involves segmentation of image and keyword length specific information. Fig. 1 (a) and 1 (b) show the warping matrix generated in [9] when the keyword is present and absent, respectively. The proposed method uses Gaussian posteriorgrams for feature representation

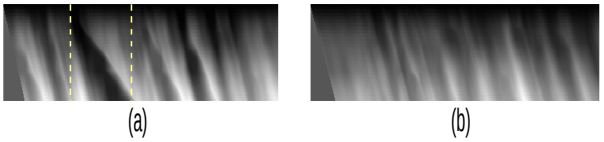

Figure 1: *Warping matrix formed when keyword is (a) present and (b) absent. Highlighted region in (a) corresponds to the region where keyword is present.*

of the speech signal which apart from easily trained, also provide a smoother representation of the signal. The texture information of warping matrix is exploited by using a convolutional neural network (CNN) [10] on top of the DTW matrix. CNNs have been very successful for the object recognition tasks, but their usage for keyword detection is a novel contribution of this paper. We will show that CNNs can learn appropriate filters in an unsupervised manner to extract texture specific information and use it for classification. Further, the algorithm is also shown to generalize over unseen keywords which is very important.

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.

2. Proposed Method

This section describes the steps involved in the devised method from feature extraction to the final training of discriminative classifier.

2.1. Feature Extraction

In our current method, Gaussian posteriorgrams have been used as feature vectors. These posteriorgram vectors provide a smoother representation of speech by fitting a multimodal Gaussian distribution on the frequency domain features. The 39 dimensional mel-frequency cepstral coefficients (MFCCs) [11] are extracted from the speech files which are then pooled together to fit a Gaussian mixture model (GMM). While testing, a posterior probability of each Gaussian component from the distribution is computed which makes the dimension of posteriorgrams same as the number of Gaussian components. The posterior probability of each GMM is given by

$$P(C_i|x_j) = \frac{P(x_j|C_i) \times P(C_i)}{\sum_{k=1}^{N} P(x_j|C_k) \times P(C_k)} \quad (1)$$

where, C_i and C_k represent the i^{th} and k^{th} Gaussian in the distribution, respectively. Vector x_j represents the j^{th} mfcc feature extracted from an audio file.

10.21437/Interspeech.2018-1436

2.2. Modified DTW

Several modifications to the original DTW algorithm have been made by researchers to handle different tasks. Segmentation of the DTW matrix into multiple regions was proposed in [6] for phonetic similarity based clustering. Each segment of the matrix is then used to find a minimum distortion warping path which is considered as a patch of local similarity. Non-segmental DTW was suggested in [7] for the task of keyword spotting by modifying the end point constraint.

The proposed method changes the formulation of recursion equation by taking the average of cost over different paths instead of the minimum. This assigns a soft penalty at each cell in the warping matrix. The underlying assumption is that the average value takes into account all possible ways of matching upto a certain point in the matrix. The distance between corresponding features is calculated using KL divergence since the features are treated as posterior probabilities. Assuming that $X_1, X_2, .., X_n$ and $Y_1, Y_2, .., Y_m$ are the feature representations for reference and query, respectively. The warping matrix D is defined as follows: $D(1,\ j) = d(1,\ j)$ for $j = 1, 2, ..n$ and $D(i,\ 1) = D(i-1,\ 1) + d(i,\ 1)$ for $i = 2, 3, ..m$. Rest of the matrix is defined by

$$D(a,\ b) = mean \begin{cases} D(a-1,\ b) + d(a,\ b), \\ D(a-1,\ b-1) + 2 * d(a,\ b), \\ D(a,\ b-1) + d(a,\ b) \end{cases} \tag{2}$$

where, $d(a,\ b)$ is the local dissimilarity between X_a and Y_b. In conventional DTW, the objective is to discover a distinct path of alignment between the two given sequences. The alignment upto $(a,\ b)$ given by equation 2 measures the average penalty of taking all possible routes. Fig. 2 shows the warping matrices in two different cases. The back tracing procedure is difficult

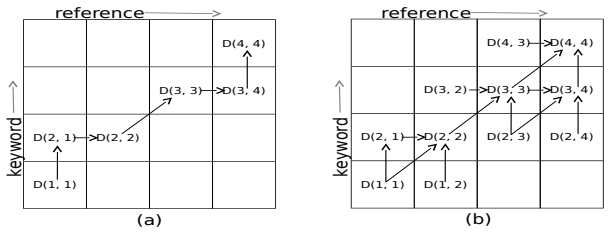

Figure 2: *Penalty at location (4,4) using (a) min and (b) mean cost formulation.*

in the second case because there is no distinct path that can be identified for cost assignment. However, it is also not important for our task since we directly use the structure of warping matrix for classification. Fig. 3 (a) and 3 (b) show the warping matrix formed using the normal recursion and modified one in the presence and absence of a keyword, respectively.

The averaging of cost over multiple paths in accumulation matrix leads to a robust penalty estimate at each cell of warping matrix. Since, the values in nearby cells of warping matrix are very close to each other, assigning an average cost ensures that there are minimal irregularities in the estimation of warping path. The markers in Fig. 3 (a) and (b) denote the region that most likely contains the keyword being checked against. In Fig. 3 (b) (top) even though the keyword is not present, a dark band can still be identified which can lead to misclassification. However, the dark band gets smeared if we use the modified

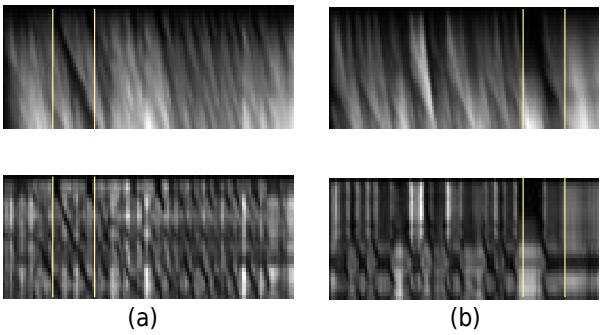

Figure 3: *Warping matrices using original and averaged recursion equations are shown in top and bottom panel, respectively. (a) Keyword present and (b) Keyword absent. Highlighted region corresponds to the portion that most likely contains the keyword.*

recursion relation. The effect of equation 2 is to smooth out any such potential streaks that might cause higher false alarms, but enhance the same region if it is a true hit. One way to think of DTW matrix is that it is merely an approximation of how the perfect matching matrix should look like. The ideal matrix when seen as a grayscale image should exhibit only a single dark band across the image and white elsewhere. Since dynamic programming employed by DTW leads to an optimal solution, we get a matrix which is significantly different from the ideal one. The minimum cost formulation can be thought of as applying a dynamic erosion [12] to the warping matrix/image which gradually thickens the evidence of keyword (dark band) as it moves across it (see Fig. 1). Hence, at an elementary level we are performing a filtering operation on portion of the matrix/image developed using a filter of shape determined by the chosen recursion relation. The motivation behind taking mean of cost comes from the fact that averaging is better for removing high frequency noise from grayscale images compared to erosion. Moreover, as we will see in the next section, the averaging approach also allows us to augment huge amount of data for the classification stage of the algorithm.

2.3. Data Augmentation

For training a complex model we need a large number of examples so that a desirable solution in the manifold can be achieved. In images, this problem has been tackled through augmentation techniques such as mirroring and intensity transformations [13]. We used a similar approach here by randomizing the recursive equation over three different topologies to simulate the effect of intensity variation. A random number p is generated at every cell of the warping matrix and accumulated cost is assigned using

$$D(i,j) = \begin{cases} (D(i-1,j) + D(i-1,j-1) + \\ \quad D(i,j-1) + 4 * d(i,j))/3 \quad \text{if } p \,\epsilon\, [0, 0.33) \\ \\ (D(i-1,j-2) + D(i-1,j-1) + 2 * d(i,j) \\ \quad + D(i-2,j-1) + \frac{1}{2} * d(i-1,j) + \\ \quad \frac{1}{2} * d(i,j-1))/3 \qquad \text{if } p \,\epsilon\, [0.33, 0.67) \\ \\ (D(i-1,j-2) + D(i-2,j-1) \\ \quad + 8 * d(i,j))/3 \qquad \text{if } p \,\epsilon\, [0.67, 1) \end{cases} \tag{3}$$

Further, the images are circularly rotated by a random integer

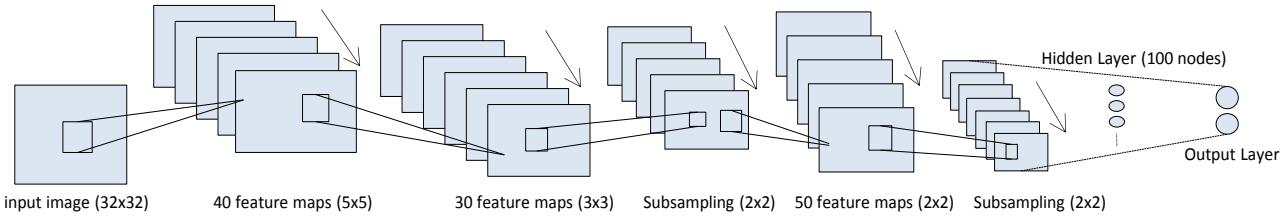

Figure 4: *CNN's architecture*

to generate more variations. This is done for both, the original image and 180° rotated version of it. Fig. 5 shows the different images generated using this technique for same keyword and reference speech input. The inherent structure in the image remains same while the position of dark bands change in these images.

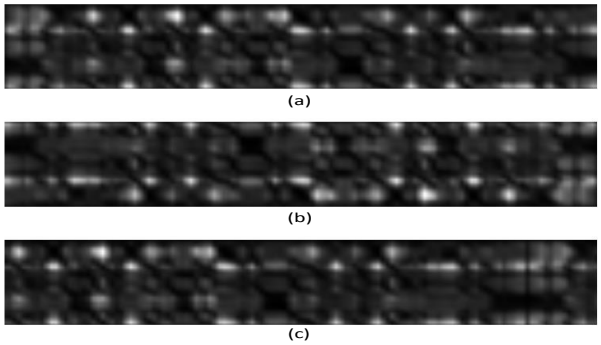

Figure 5: *Variation in (a) original image by (b) flipping and (c) random circular shift.*

2.4. Convolutional Neural Network

The original LeNet-5 [10, 14] has been modified to address our problem of keyword detection. The model has 3 convolutional layers 40, 30 and 50 feature maps, respectively. The size of feature maps vary from 5×5 in the first convolutional layer to 3×3 in the second and 2×2 in the third (see Fig 4). There is no local response normalization [15] in the convolution layers. The hidden layer is a fully connected layer with 100 nodes followed by 2 final output nodes for softmax classification. The max-pooling windows are of size 2×2 with zero overlapping. It reduces the size of input matrix by half along each dimension. The activation function for hidden nodes and output nodes is the tan hyperbolic function which is known to have faster convergence than logistic sigmoid. The initial value of parameters for hidden units in any layer l are chosen by uniformly sampling from the interval $\left(-\sqrt{\frac{6}{z_{in}^{l-1}+z_{out}^{l+1}}}, \sqrt{\frac{6}{z_{in}^{l-1}+z_{out}^{l+1}}}\right)$, where, z_{in}^{l-1} and z_{out}^{l+1} are the number of nodes in layer $l-1$ and $l+1$, respectively [16]. We chose negative log likelihood as the error function for our model with $L2$ regularization of parameters to avoid overfitting. The loss function is defined as

$$L(x, y, \theta) = -log(\prod_{i=1}^{N} P(y_i|x_i, \theta)) + \lambda * ||\theta||^2 \qquad (4)$$

where, θ represents the parameters of the CNN and λ controls the regularization.

3. Experiments and Results

This section summarizes the results of our proposed method. It gives a brief overview of the dataset we used and how the optimization of hyperparameters were done.

3.1. Dataset

The TIMIT corpus [17] is split into the training and testing data. The training set has 4320 speech files belonging to 432 speakers, while the test set has 1680 utterances spoken by 168 speakers (not part of the training set). A set of 21 keywords of varying lengths were chosen from the TIMIT database. The selected keywords list has a good overlap with those used in [6]. 7 templates are randomly picked for each keyword (on average) from the training set. Table 1 contains the list of keywords used for experiment (overlapping keywords with [6] are shown in bold).

Table 1: *List of keywords used for evaluation*

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

3.2. CNN Parameters

Increasing the depth of CNN is the easiest way to reduce the training error. However, to avoid the model from overfitting the training data, the size of training samples must also increase in the same proportion. We fixed the depth of CNN to 3 convolutional layers and 1 fully connected hidden layer with 100 nodes. Further decrease in depth leads to poor generalization by increasing test error. Number of feature maps in each convolution layer and regularization parameter (λ) are decided using cross validation. Fig. 6 shows the cross validation curve of network (Fig 4) for first few epochs.

3.3. Training

The speech files from training set are used to create the GMM. The warping matrices are formulated by using each keyword template against all speech files from the test set. The matrices are then converted into gray scale images and are size normalized to a dimension of 32×128. They are further broken down into 4 patches of size 32×32 each (no overlap). Every single patch is labelled same as its parent image (0 when keyword is absent and 1 when present) to avoid manual annotation which

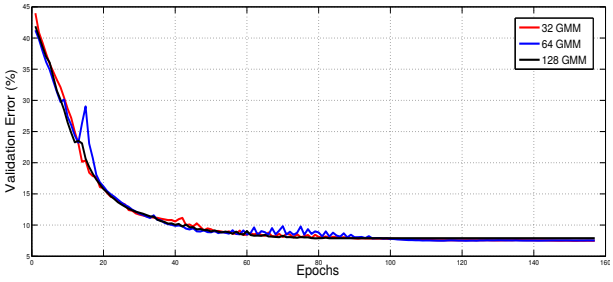

Figure 6: *Cross-validation error for the mentioned architecture.*

is a human intensive task. This approach may lead to a slightly higher false positive rate but the final results noted are within acceptable boundaries. The CNN is trained on these patches with minibatch stochastic gradient of batch-size 1000 and a learning rate of 0.1 (halved after every 50 epochs) in theano [18]. Fig. 7 (a) and 7 (b) show the response of first and second layer of CNN for an input image, respectively.

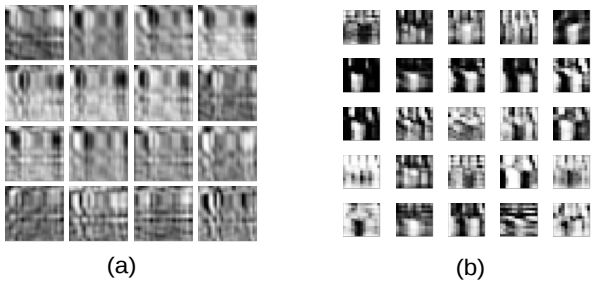

Figure 7: *CNN filter response of (a) first and (b) second layer.*

3.4. DTW recursion

To increase the number of positive samples in the training images, equation 3 is used as it somewhat approximates the effect of intensity variation. Since the classification model has been trained with these samples, the warping matrix in test environment is formulated by taking minimum over all 3 constraints mentioned in equation 3 at each cell. It achieves the lowest error than any original recursion relation (see Table 2).

Table 2: *Comparison among original, averaged and min. of averages of constraints after 200 epochs.*

GMM Model	Constraint	Original	Average	Min. of Averages
32	False alarm rate	0.091	0.077	0.076
	False rejection rate	0.117	0.091	0.081
	Overall error rate	0.104	0.084	0.078
64	False alarm rate	0.103	0.071	0.0752
	False rejection rate	0.102	0.084	0.0758
	Overall error rate	0.103	0.077	0.075
128	False alarm rate	0.092	0.083	0.085
	False rejection rate	0.097	0.077	0.076
	Overall error rate	0.095	0.081	0.0807

Fig. 8 displays the precision at various values of regularization parameter (in log scale) for different mixture models. While the mixture model containing 32 and 64 Gaussians have roughly similar performance, the 128 GMM model is marginally better than its other counterparts. The reason could be due to the better approximation of the distribution of training samples by higher number of Gaussians.

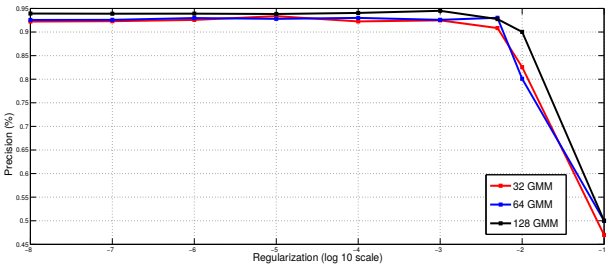

Figure 8: *Precision of model for different values of regularization parameter* (λ).

3.5. Generalization

Generalization error is the most important aspect of our keywords detection method. Table 3 shows the generalization result over unseen keywords (chosen randomly). We followed one-of-k experimental approach in which training was done by removing samples of the keyword to be tested.

Table 3: *False rejection and false alarm rate on unseen keywords.*

Artists (0.25/0.005)	Beautiful (0.37/0.002)	Greasy (0.12/0.007)
Organizations (0.061/0.001)	Likethat (0.02/0.004)	Money (0.2/0.003)
Oilyrag (0.08/0.002)	Popularity (0.21/0.001)	Washwater (0.14/0.00)
Problem (0.093/0.00)	Artists (0.05/0.002)	

Maximum term weighted value (MTWV) [19] quantifies the results of keyword detection model appropriately by taking both false alarms and miss rate of each keyword weighted by their prior probability of occurrence. Table 4 shows the MTWV and equal error rate (EER) obtained by proposed method along with segmental DTW (seg DTW), non-segmental DTW (Non Seg DTW) and HMM-ANN posteriorgram method ([6, 7, 9]). However, the results are not directly comparable due to the slight variation in keywords used for these experiments.

Table 4: *Performance of joint DTW-CNN against other models*

Model	Seg DTW	Non Seg DTW	HMM-ANN	CNN1	CNN2	CNN3
MTWV	-	0.399	0.816	0.832	0.841	0.84
EER	0.225	-	-	0.2453	0.223	0.1667

CNN1, CNN2 and CNN3 stand for the joint DTW-CNN framework for 32, 64 and 128 GMMs, respectively.

4. Summary and Conclusion

In this paper a modified DTW coupled with a CNN for the task of spoken keyword detection is proposed. We used a set of keywords from TIMIT corpus and demonstrated that the current method works better than other existing algorithms. It has lower generalization error in comparison to other algorithms that use the similar feature. This method can also be extended to out of vocabulary keywords because it requires no prior knowledge of keywords being used for training and depends on intrinsic property of time warping matrix. The method shows promising results for extension to language independent scenario too and can be explored in detail in future.

5. References

[1] I. Bufyko, O. Kimball, M. H. Siu, J. Herrero, and D. Blum, "Detection of unseen words in conversational mandarin," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 5181–5184.

[2] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient wfst-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1352–1365, May 2007.

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

[4] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, Feb 1978.

[5] *Dynamic Time Warping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 69–84.

[6] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on*, Nov 2009, pp. 398–403.

[7] G. Mantena, S. Achanta, and K. Prahallad, "Query-by-example spoken term detection using frequency domain linear prediction and non-segmental dynamic time warping," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 5, pp. 946–955, May 2014.

[8] Y. Zhang and J. R. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 4366–4369.

[9] R. Shankar, A. Jain, D. K. T., V. C. M., and S. R. M. Prasanna, "Spoken term detection from continuous speech using ann posteriors and image processing techniques," in *2016 Twenty Second National Conference on Communication (NCC)*, March 2016, pp. 1–6.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[11] S. Y. et al., *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, Cambridge, 2009.

[12] R. W. R.C. Gonzalez, *Digital image processing*. Prentice-Hall, Inc., 2002.

[13] D. C. L. C. Zhe Gan, Ricardo Henao, "Learning deep sigmoid belief networks with data augmentation," *Journal For Machine Learning*, vol. 38, p. 268276, 2015.

[14] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009, also published as a book. Now Publishers, 2009.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[16] Y. B. Xavier Glorot, "Understanding the difficulty of training deep feedforward neural networks," *Journal For Machine Learning*, vol. 9, p. 249256, 2010.

[17] e. a. Garofolo, John, "Timit acoustic-phonetic continuous speech corpus ldc93s1," 1993.

[18] "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[19] S. Wegmann, A. Faria, A. Janin, K. Riedhammer, and N. Morgan, "The tao of atwv: Probing the mysteries of keyword search performance," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 192–197.