RARE SOUND EVENT DETECTION USING 1D CONVOLUTIONAL RECURRENT NEURAL NETWORKS

Hyungui Lim[1], *Jeongsoo Park*[1,2], *Kyogu Lee*[2], *Yoonchang Han*[1]

[1] Cochlear.ai, Seoul, Korea
[2] Music and Audio Research Group, Seoul National University, Seoul, Korea
{hglim, jspark, ychan}@cochlear.ai, kglee@snu.ac.kr

ABSTRACT

Rare sound event detection is a newly proposed task in IEEE DCASE 2017 to identify the presence of monophonic sound event that is classified as an emergency and to detect the onset time of the event. In this paper, we introduce a rare sound event detection system using combination of 1D convolutional neural network (1D ConvNet) and recurrent neural network (RNN) with long short-term memory units (LSTM). A log-amplitude mel-spectrogram is used as an input acoustic feature and the 1D ConvNet is applied in each time-frequency frame to convert the spectral feature. Then the RNN-LSTM is utilized to incorporate the temporal dependency of the extracted features. The system is evaluated using DCASE 2017 Challenge Task 2 Dataset. Our best result on the test set of the development dataset shows 0.07 and 96.26 of error rate and F-score on the event-based metric, respectively. The proposed system has achieved the 1st place in the challenge with an error rate of 0.13 and an F-Score of 93.1 on the evaluation dataset.

Index Terms— Rare sound event detection, deep learning, convolutional neural network, recurrent neural network, long short-term memory

1. INTRODUCTION

Auditory information helps people recognize their surroundings. In an emergency situation, auditory information becomes even more important as it allows nearby people to react effectively and quickly. Rare sound event detection (RSED) is a set of algorithms that aim to automatically detect certain emergency sounds with high accuracy. As part of such efforts, task 2 in Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 is organized, which asks to identify the presence of three target events – baby crying, glass breaking, and gunshot – and their corresponding onset time.

According to its necessity, sound event detection has been studied extensively in recent years. Some studies aim to recognize multiple sound events that occur simultaneously (polyphonic) [1], [2], [3], [4] while others detect one prominent event among multiple candidates (monophonic) [1], [5], [6], [7]. In the case of an emergency, monophonic detection is considered as more suitable approach since the emergency-related sounds scarcely occur simultaneously. Hence, detecting single type of sound with high accuracy is more valuable in such cases.

In terms of algorithms, a number of conventional research efforts have applied machine learning algorithms such as hidden Markov model (HMM) [5], non-negative matrix factorization (NMF) [8], [9], support vector machine (SVM) [10], and random forest [7]. Recent approaches use deep learning-based methods using deep neural network (DNN) [2], convolutional neural network (ConvNet) [11], recurrent neural network (RNN) [3], [12], and convolutional recurrent neural network (CRNN) [4].

In this paper, we apply a hybrid neural network of 1D ConvNet and RNN with long short-term memory units (LSTM). Frame-wise log-amplitude mel-spectrogram is fed into our proposed model, and the model returns the output for every incoming sequence. It makes possible to estimate a relatively accurate onset time by maintaining small temporal resolution. This single model is applied to compute event probability for all three target events. We also conduct experiments with different fixed length input (timestep) and different set of data mixtures to find the best hyperparameters. We confirm that our proposed method shows significant improvement in the test set of TUT rare sound events 2017 dataset compared to the baseline.

The rest of the paper is organized as follows. Section 2 describes the proposed method. Section 3 shows the experimental results with TUT rare sound events 2017 dataset. Conclusions are presented in Section 4. Algorithm description for DCASE 2017 submission is presented in Section 5.

2. PROPOSED METHOD

Fig. 1 shows an overall framework of our proposed method which consists of four parts: 1) extracting log-amplitude mel-spectrogram from audio, 2) converting spectral feature with 1D ConvNet, 3) incorporating temporal dependency with RNN-LSTM, and 4) determining the presence and the onset time of audio event with post-processing.

2.1. Log-amplitude mel-spectrogram

Mel-spectrogram is a 2D time-frequency representation extracted from an audio signal. It has been recognized as a useful feature and has been used for various deep learning-based audio analyses. Unlike normal spectrogram, the frequency components are filtered with log-scale filter banks to imitate the function of human ears. It leads compression of high frequency components and helps to concentrate more on low frequency components.

Considering these advantages of using mel-spectrogram, we also use it as the input feature of our proposed method. To extract this feature, a window is applied to an audio signal with a size of 46 ms, being overlapped with half size of the window. We also apply 128 mel-filter banks on the spectrum of each frame and take logarithm on the amplitude. The mel-spectrogram is divided into a chunk with the size of a timestep (τ), and fed into our network.

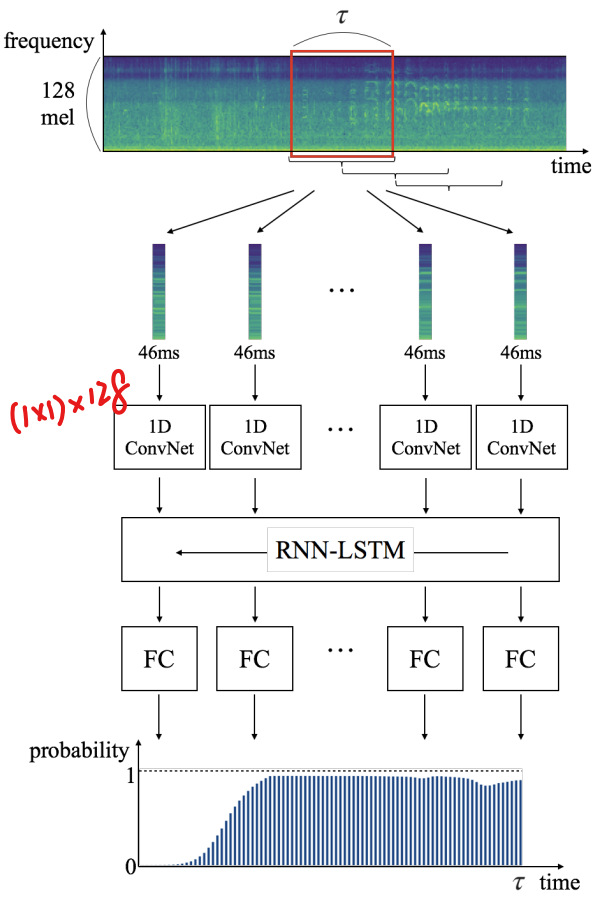

Figure 1: Overall framework of the proposed method.

2.2. 1D ConvNet

Many audio contents analysis studies that use 2D input features such as spectrogram, mel-spectrogram, and mel-frequency cepstral coefficients (MFCC) apply 2D ConvNet [13], [14], which is often used for image content analysis. It focuses on the spectral and temporal locality from the audio features to extract meaningful information. However, 2D ConvNet-based methods analyze the audio in chunk-level rather than frame-level. Since the precise estimation of onset time is necessary for this task, we apply spectral-side 1D ConvNet that enables frame-level investigation.

The 1D ConvNet step consists of 1D convolution layer, Batch Normalization (BN) [15] process, and pooling layer. Fig. 2 shows the concept of the 1D convolution layer and the max-pooling layer. The filter size of the convolution layer is set to 32, and 128 filters are used in total. Therefore, 128 outputs each contains 97 $(128-32+1)$ elements are produced when single frame of the mel-spectrogram (128 frequency bin) is fed into the 1D convolution layer. In the next step, BN is applied on feature map outputs so that they maintain the mean close to 0 and the standard deviation close to 1. After that, rectified linear unit (ReLU) [16] is applied as an activation function. Finally, max-pooling with the size of 97 is applied to each output to extract representative value. Dropout is also applied with the value of 0.3 at the end of ConvNet to prevent overfitting.

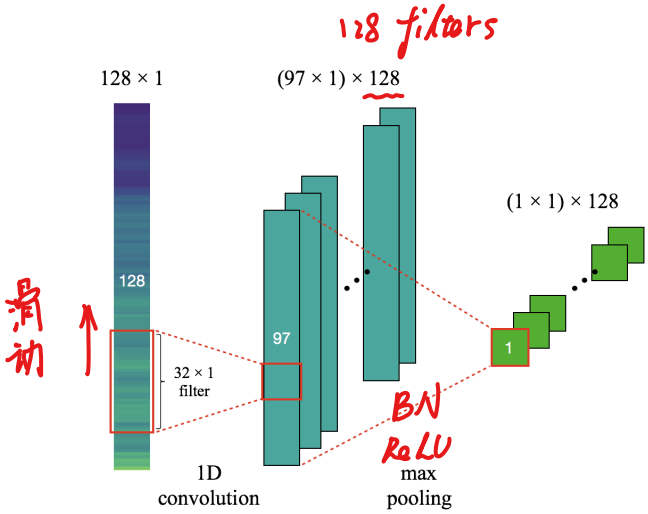

Figure 2: 1D-ConvNet structure for frame-wise feature extraction. The output feature size is same as the input mel-band size at 128.

2.3. RNN-LSTM

RNN has proven to be a powerful model for identifying sequential information such as speech recognition [17] and hand writing recognition [18]. In particular, RNN-LSTM is a well-known deep learning model that prevents vanishing gradient that disturbs long-term sequence learning [19]. Thus, we use RNN-LSTM here to incorporate the temporal dependency of the extracted features.

Here, we use two RNN layers each contains 128 LSTM units. Unlike general studies using forward or bidirectional RNN-LSTM, we apply unidirectional backward RNN-LSTM. This is because the information after the onset of an event is appeared to be more important for the precise onset detection compared to the information before the onset. According to our experiment, this unidirectional backward analysis has shown better performance than the other methods.

Fig. 3 shows the processing structure inside the RNN-LSTM step. The features extracted from the ConvNet $(x_t, x_{t+1}, ..., x_{t+\tau-1})$ are fed into the networks that passes it through the layers. Note that the 128-dimensional output vectors $(z_t, z_{t+1}, ..., z_{t+\tau-1})$ are obtained for each frame. We use hyperbolic tangent (tanh) as an activation function and apply a dropout rate of 0.3 for all RNN-LSTM layers.

2.4. Fully connected layer and post-processing

The returned features from the RNN-LSTM layer are fed into a fully connected layer (FC) that contains 128 hidden units. Similar to the previous 1D ConvNet step, BN and ReLU are applied as a normalization function and activation function, respectively. The updated features are then forwarded to a time-distributed output layer with one sigmoid unit, of which output represents the probability of presence of the target sound event. As a result, the probability values are calculated for each frame of the mel-spectrogram during the timestep.

In order to obtain the probability sequence of an entire audio clip at the test stage, sliding ensemble method is utilized. As the probability values are calculated in each chunk with our trained model, this method combines the entire probabilities by sliding the prediction chunk with a hop size of one frame (23 ms) and aver-

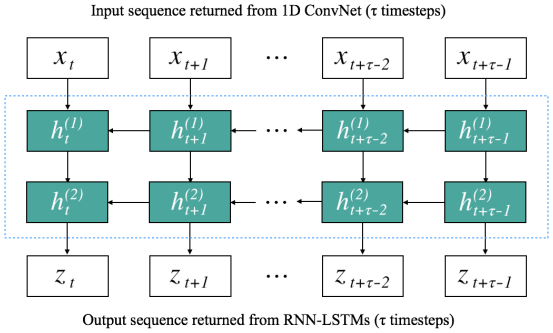

Figure 3: RNN-LSTM structure for sequential learning. Two hidden RNN-LSTM layers (h) are applied in a backward direction. They return the output (z) for all inputs (x) during the timestep (τ).

aging the probabilities of the indices where the value exists. An illustration of this method is shown in Fig. 4.

Fig. 5 shows an example of the determination of event presence and prediction of its corresponding onset from probability sequence. In order to determine the presence of a sound event, hard thresholding scheme with empirical assumptions is used. If the maximum value in the probability sequence is greater than 0.8 (0.5 for 'gunshot'), the audio clip is considered to include the target event. To find the onset time of the sound event, we select the first index of the value greater than 0.5 among the 50 (200 for 'baby crying') preceding frames from the maximum value.

3. PERFORMANCE EVALUATION

3.1. Dataset

For the task 2 of DCASE 2017, 'TUT Rare Sound Events 2017' dataset is provided, which consists of isolated sound events for each target class and recordings of everyday acoustic scenes to serve as background. In the dataset, three target sound events are considered: 'baby crying', 'glass breaking', and 'gunshot'. The background audio set contains recordings from 15 different audio scenes, which are a part of 'TUT Acoustic Scenes 2016' dataset.

The source code for creating a combination of different event-to-background is also given along with the audio recordings. Using the code, we can generate training data with different parameters such as number of mixtures, event-to-background ratio (EBR) and event occurrence probability. Annotations for the mixtures including the name of the target event and its temporal position are also produced automatically. We have created 4 sets of mixtures (S_1, S_2, S_3, S_4). Each set consists of 15,000 audio clips (5,000 per event class), generated with EBRs of -6, 0, 6dB and an event occurrence probability of 0.5. All mixtures are created as a 30-seconds monaural audio with 44,100 Hz and 24 bits. For the training, these mixtures are randomly divided into a train set and validation set at 8 to 2 ratios. Pre-combined test set which contains 1,500 audio clips (500 per event class) is used at the test process.

3.2. Deep learning setup

In the training stage, after the input chunks are fed into the model and converted to probability values, errors between the predicted values and correct values (0 or 1) are calculated with a binary cross

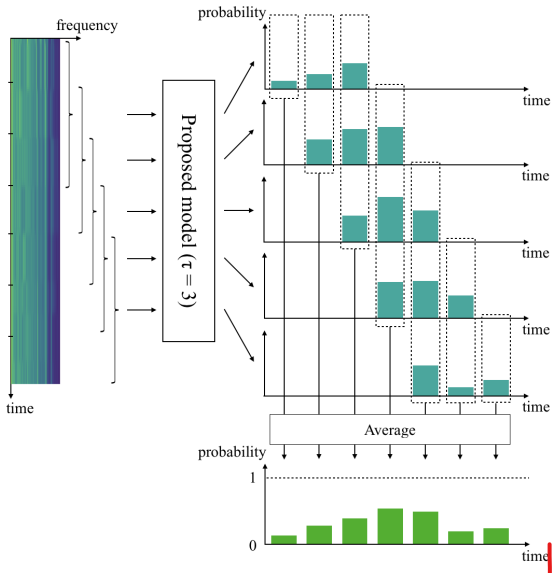

Figure 4: An example of a sliding ensemble method using a model with a timestep of 3. The predicted probability sequences from each sliding window are combined into a single probability sequence in this way.

entropy as a loss function. To optimize the loss, we apply adaptive momentum (Adam) as an optimizer and the size of a mini-batch is set to 256. The learning rate is initially set to be 0.001 and decayed over each epoch with decaying factor 0.01 of learning rate. Learning is stopped early when a validation loss has stopped improving for 10 epochs.

3.3. Evaluation metric

We evaluate our method using event-based metric [20], which requires calculation of true positives (TP), false positives (FP), and false negatives (FN). If the system's output accurately predicts the presence of an event and its onset, it is computed as TP. The onset time detection is considered true only when it is predicted within the range of 500 ms of the actual onset time. Meanwhile, FP indicates that the system incorrectly detects the presence of an event when there is no event. If the system output misses the event, it is considered an FN. These metrics are used to calculate error rate and F-score in the final step, which are mathematically defined as

$$ER = \frac{FN + FP}{N} \tag{1}$$

$$F = \frac{2PR}{P + R}, \tag{2}$$

where N denotes the total number of samples in the evaluation dataset, and P and R denote precision and recall, defined as below.

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{TP + FN} \tag{4}$$

These evaluation metrics were computed using sed_eval toolbox [20] which is given in the task.

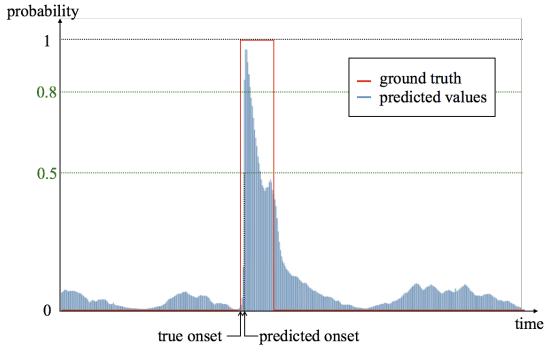

Figure 5: An example of applying a threshold to detect the presence and the onset time of an event.

3.4. Result and discussion

We have compared the experimental results by changing the set of mixtures and timestep. Then we have selected models that show relatively high-performance, followed by the ensemble method to combine them. Table 1 shows the types of models combined for an ensemble method and their mixing weights. p_a^b denotes the probability value calculated by the trained model using a mixture set of S_a and a timestep size of b. Table 2 shows the event based error rate and F-score results on the test set of the development dataset. Both results of our proposed method show better performance than the DCASE 2017 baseline system.

The result shows that our method achieves the best performance on 'glass breaking' followed by 'baby crying' and 'gunshot'. In the case of 'glass breaking', the frequency component at the moment when the glass breaks is clear compared to the background sound. Therefore, the model with short timestep was effective for this class. In the case of 'baby crying', since the length of the sound event is longer than the others, it was better to apply a relatively longer timestep. For the same reason, a long frame range to find the onset time worked better as mentioned in Section 2.4. Still, there existed misclassified events such as bird sound which has similar tonality to the baby crying. In the case of 'gunshot', relatively short timestep was used because similar to 'glass breaking', the moment of the gunshot is obvious as it sounds like an impulse. However, since the gunshot sound has a lot of reverberations, it seemed to require slightly longer timestep than 'glass breaking'. The result shows that the performance of 'gunshot' detection is worse than the others because the sounds vary according to the gun type. For that reason, several misclassifications are observed on impulse-like sound events such as footstep and metal door-closing sounds. Regardless of the event class, the onset time was relatively accurate even if the model estimated the presence of event incorrectly.

Overall, increasing the amount of training data by synthesizing various mixtures seemed more effective for the performance than adjusting the parameters of the model. The experimental result showed meaningful performance improvement when we boosted the audio clips 10 times more than the given mixture set.

4. CONCLUSION

In this paper, we have presented a rare sound event detection system using 1D convolutional recurrent neural networks. It has shown promising results on IEEE DCASE 2017 Task 2. We believe that three key factors in the proposed method have contributed to the performance improvement. The first factor is frame-wise detection of the model which is effective in finding the precise onset time. The second is the internal/external ensemble method used in Section 2.4 and Section 3.4 which reduces various noises. The last and the biggest contributor to the performance improvement is a large amount of synthesized data consists of various mixtures.

Table 1: Selected models and their weights for an ensemble method.

Event	Ensemble method
Baby crying	$\left(p_1^{(100)} + 2p_2^{(50)} + p_3^{(50)} + p_3^{(100)}\right)/5$
Glass breaking	$\left(p_1^{(5)} + p_3^{(5)}\right)/2$
Gunshot	$\left(2p_1^{(14)} + p_1^{(50)} + p_3^{(10)} + p_4^{(10)} + p_4^{(20)}\right)/6$

Table 2: Performance of baseline and proposed system in the development set.

	ER		F-score	
	baseline	proposed	baseline	proposed
Baby crying	0.67	**0.05**	72.0	**97.6**
Glass breaking	0.22	**0.01**	88.5	**99.6**
Gunshot	0.69	**0.16**	57.4	**91.6**
Overall	0.53	**0.07**	72.7	**96.3**

Table 3: Performance of baseline and proposed system in the evaluation set.

	ER		F-score	
	baseline	proposed	baseline	proposed
Baby crying	0.80	**0.15**	66.8	**92.2**
Glass breaking	0.38	**0.05**	79.1	**97.6**
Gunshot	0.73	**0.19**	46.5	**89.6**
Overall	0.64	**0.13**	64.1	**93.1**

5. DCASE 2017 SUBMISSION

We applied the same model settings of the development set to the evaluation set. For the final submission, we selected four different results by applying four different threshold set of event presence (mentioned in Section 2.4). We used the threshold set with 0.8 / 0.8 / 0.5 ('baby crying' / 'glass break' / 'gunshot') for submission 1, 0.7 / 0.7 / 0.5 for submission 2, 0.6 / 0.6 / 0.5 for submission 3, and 0.5 / 0.5 / 0.5 for submission 4. The error rate and F-score was 0.13 / 93.1 for submission 1, 0.13 / 93.0 for submission 2, 0.15 / 92.2 for submission 3, and 0.17 / 91.4 for submission 4. We achieved the best result with submission 1 and the results of each class from this submission are shown in Table 3.

6. ACKNOWLEDGEMENT

This research was supported by Korean government, MSIP provided financial support in the form of Bio & Medical Technology Development Program (2015M3A9D7066980).

7. REFERENCES

[1] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, p. 1, 2013.

[2] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–7.

[3] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6440–6444.

[4] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *arXiv preprint arXiv:1702.06286*, 2017.

[5] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Signal Processing Conference, 2010 18th European*. IEEE, 2010, pp. 1267–1271.

[6] C. V. Cotton and D. P. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*. IEEE, 2011, pp. 69–72.

[7] H. Phan, M. Maaß, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 20–31, 2015.

[8] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, *et al.*, "An exemplar-based nmf approach to audio event detection," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[9] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 151–155.

[10] A. Temko and C. Nadeu, "Classification of acoustic events using svm-based clustering schemes," *Pattern Recognition*, vol. 39, no. 4, pp. 682–694, 2006.

[11] A. Gorin, N. Makhazhanov, and N. Shmyrev, "Dcase 2016 sound event detection system based on convolutional neural network," *IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events*, 2016.

[12] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1996–2000.

[13] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition." in *Interspeech*, 2013, pp. 3366–3370.

[14] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*. IEEE, 2015, pp. 1–6.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[17] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[18] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[19] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[20] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.