VIRTUAL ADVERSARIAL TRAINING FOR DS-CNN BASED SMALL-FOOTPRINT KEYWORD SPOTTING

Xiong Wang[1*], *Sining Sun*[1,2*], *Lei Xie*[1†]

Audio, Speech and Language Processing Group,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China[1]
Tencent, China[2]

ABSTRACT

Serving as the tigger of a voice-enabled user interface, on-device keyword spotting model has to be extremely compact, efficient and accurate. In this paper, we adopt a depth-wise separable convolutional neural network (DS-CNN) as our small-footprint KWS model, which is highly competitive to these ends. However, recent study has shown that a compact KWS system is very vulnerable to small adversarial perturbations while augmenting the training data with specifically-generated adversarial examples can improve performance. In this paper, we further improve KWS performance through a virtual adversarial training (VAT) solution. Instead of using adversarial examples for data augmentation, we propose to train a DS-CNN KWS model using adversarial regularization, which aims to smooth model's distribution and thus to improve robustness, by explicitly introducing a distribution smoothness measure into the loss function. Experiments on a collected KWS corpus using a circular microphone array in far-field scenario show that the VAT approach brings 31.9% relative false rejection rate (FRR) reduction compared to the normal training approach with cross entropy loss, and it also surpasses the adversarial example based data augmentation approach with 10.3% relative FRR reduction.

Index Terms: depthwise separable convolutional neural network, DS-CNN, KWS, virtual adversarial training

1. INTRODUCTION

With the exponential growth of mobile and intelligent devices, such as smart speakers, voice-enabled user interfaces play an increasingly crucial role in achieving natural user experiences. Such voice interfaces are usually triggered by an on-device *keywords spotting* (KWS) module that always stands by and listens for the wake/trigger word(s). With limited on-device memory and computational capabilities, the KWS module has to be deployed with a small-footprint algorithm with real-time response. Meanwhile, as the first step before speech interactions, accurate on-device detection, with low false reject rate (FRR) and false alarm rate (FAR), is crucially important for customer experiences, especially for those always-on devices deployed in real-world complicated acoustic environments with noises and reverberations.

There has been a rich literature on the topic of keyword spotting from audio. Previous heavy KWS approaches, which rely on a large vocabulary continuous speech recognizer (LVCSR) [1, 2, 3] while latency and computation issues are not their concerns, are apparently not suitable for on-device deployment. In the past decade, hidden Markov model (HMM) based keyword/filler approaches [4, 5, 6, 7] have been very popular for online low-latency and computation-constrained KWS. Under such an HMM framework, with the recent renaissance of neural networks, Gaussian mixture model (GMM) based acoustic model has been replaced by deep neural network (DNNs). The HMM based approaches, though very compact and competitive, still need Viterbi search on the HMM graph. Alternatively, some recent systems were solely based on a single DNN without the use of HMM topology and Viterbi decoding. In this small-footprint system, a compact DNN is trained to predict the posteriors of (sub-)keyword and filler units and a simple post-processing module produces a confidence score for keyword/non-keyword decision [8]. Following this line, various types of neural networks, including recurrent and convolutional structures with better contextual modeling ability, have been intensively explored recently [9, 10, 11, 12, 13]. Among them, a depth-wise separable convolutional neural network (DS-CNN) [14] approach has become highly competitive, outperforming other models in all aspects of accuracy, model size and operation time.

With a small-footprint efficient solution, deploying a highly accurate KWS system to real applications is still very tricky. False alarms and false rejections are unavoidable, although feeding more positive and negative examples in the model training is quite useful to suppress these errors. More frustratingly, in a real system, such as the voice trigger on a home smart speaker, the false-alarmed and false-rejected queries are extremely non-reproducible. This is mainly due to 1) the complicated time-varying acoustic environments and 2) the subtle change on speech timbre when the same speaker uttering the same trigger word at different time. Recent study [15] has treated these unpredictable false alarm

*The first two authors contributed equally to this work. This research work is supported by the National Natural Science Foundation of China (No.61571363).

†Lei Xie is the corresponding author.

and rejections as *adversarial examples* [16, 17, 18, 19]. In other words, the model's output is unsmooth with respect to the input, where a small perturbation in the input space can lead to a big change in the output space. Specifically, the KWS models are extremely vulnerable to such small perturbations or more formally *adversarial perturbations*. Based on this phenomenon, recent study [15] has further proposed to augment the training data with specifically-generated adversarial examples and then retrain the model, which leads to significant performance improvement.

In this paper, we continue to explore the adversarial example idea and further improve the KWS performance through a *virtual adversarial training* (VAT) [20] solution. Instead of using adversarial examples for explicit data augmentation, we propose to train a DS-CNN KWS model using adversarial regularization. In the proposed VAT approach, the original loss function is amended by the local distributional smoothness (LDS) loss on the adversarial example. Explicitly introducing the impact from adversarial examples into the loss function can make model more robust to minor deviations from the original training data. In other words, increasing the model smoothness by this way, we expect that different outcomes of the same keyword should get closer neural network outputs, hopefully leading to decreased false rejections. We validate our approach on a KWS corpus collected from a circular microphone array in far-field scenario. Results have consistently confirmed our expectations: the proposed VAT approach brings 31.9% relative FRR reduction, and it also outperforms the previous adversarial example based data augmentation approach [15] with 10.3% relative FRR reduction.

2. DS-CNN KWS MODEL

A standard deep KWS system usually consists of three modules [8], i.e., feature extraction, acoustic model and post-processing. The neural network based acoustic model accepts frame-level speech features and outputs frame-level posterior probability of modeling units, i.e., keyword or sub-keyword units and filters. In this paper, we adopt a DS-CNN as our acoustic model given its advances of high accuracy, compact model size and competitive operation time. The DS-CNN structure has been widely used in computer vision [21] because it adopts a compact network structure which can effectively replace the standard computation-intensive 3-D convolution operation.

Figure 1 depicts the DS-CNN based KWS acoustic model that takes acoustic features as input and outputs the prediction to the target labels. The neural network structure mainly includes three components, a standard convolution layer, N DS-convolution layers and an average pooling layer. Specifically, a depthwise separable convolution is made up of a depthwise convolution layer and a pointwise convolution layer, where the ReLu nonlinearities are followed. Depthwise convolutions are used to apply a single filter per input channel (input depth), which are extremely efficient as compared with standard convolution. But it only filters input channels without combining them to create new features. Hence pointwise convolution, or a simple 1×1 convolution is used to make a linear combination of the output of the depthwise layer. Please refer to [21] for more details.

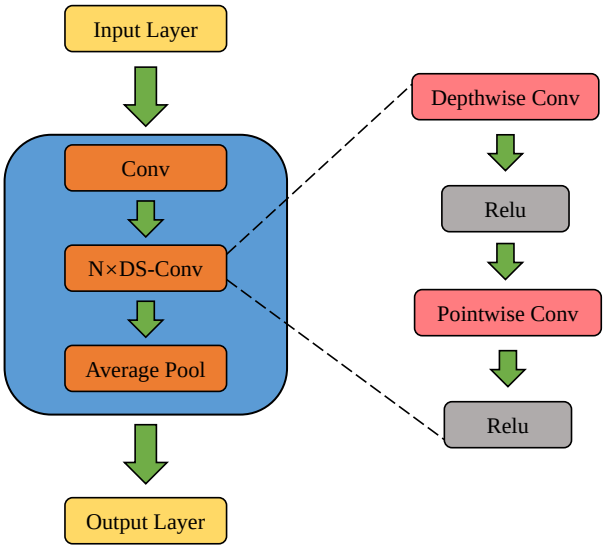

Fig. 1. DS-CNN based KWS system [14].

The frame level posteriori probability obtained by the acoustic model cannot be used directly to make decision because it is too noisy and unstable. Therefore, a post-processing module is necessarily followed to calculate the confidence score and make a decision. In this paper, we use the similar post-processing strategy in Deep KWS [8]. Since the output of the neural network is not smooth, we first process the outputs with a window of size w_{smooth} using

$$p'_{ij} = \frac{1}{j - h_{smooth} + 1} \sum_{k=h_{smooth}}^{j} p_{ik} \tag{1}$$

where $h_{smooth} = \max\{1, j - w_{smooth} + 1\}$ and p_{ij} means the neural network posterior probability for the i^{th} label and the j^{th} frame. Confidence score will be calculated with sliding window of size w_{max} by

$$p = \sqrt[n-1]{\prod_{i=1}^{n-1} \max_{h_{\max} \le k \le j} p'_{ik}}. \tag{2}$$

where $h_{smooth} = \max\{1, j - w_{max} + 1\}$ and n represents the output probability dimention. Besides, p represents the confidence of the keyword which needs to be detected.

3. VIRTUAL ADVERSARIAL TRAINING

In KWS, our aim is to suppress false alarms and false rejections. But in real-world, a small perturbation such as background noise may incur such errors. From the *adversarial*

perturbation point of view [22], the model's output is *unsmooth* with respect to the input, where a small perturbation in the input space can lead to a big change in the output decision space. Hence our goal is to achieve a smooth acoustic model that is insensitive to the small perturbations to the input. From this view, Miyato et al. [20] have recently proposed *local distribution smoothness* (LDS) to measure a model's smoothness. Given a well-trained model distribution $\boldsymbol{P}(y|x,\theta)$ parameterized by θ, where x is input and y is output, we define

$$\Delta(\delta, x, \theta) = \mathbf{KL}[\boldsymbol{P}(x,\theta)||\boldsymbol{P}(x+\delta,\theta) \tag{3}$$

$$\delta_V = \arg\max_{\delta} \{\Delta(\delta, x, \theta) \; where \; \|\delta\|_2 \leq \varepsilon\} \tag{4}$$

where $\Delta(\delta, x, \theta)$ is the KL divergence between the model distributions before and after input perturbation with δ. ε is a small positive constant. The small perturbation δ_V is referred as *virtual adversarial perturbation* for input data x, because it is the direction to which the model distribution is most sensitive in the sense of KL divergence. The LDS at data point x can be defined as

$$\text{LDS}(x, \theta) = -\Delta(\delta_V, x, \theta) \tag{5}$$

Because the δ_V breaks the model distribution $\boldsymbol{P}(y|x,\theta)$ at data point x in a most extreme way, the bigger the value of LDS, the smoother the model distribution $\boldsymbol{P}(y|x,\theta)$ at x. Our goal is to improve LDS in the neighborhood of all the training data, therefore the regularized objective function becomes:

$$J_{\text{VAT}}(x, y; \theta) = \frac{1}{N}\sum_{n=1}^{n=N} J(x^{(n)}, y; \theta) - \alpha \frac{1}{N}\sum_{n=1}^{n=N} \text{LDS}(x^{(n)}, \theta) \tag{6}$$

where $\alpha > 0$ is used to balance the weight of LDS and N means the total number of samples. Training the model using Eq (6) is referred as *virtual adversarial training (VAT)*.

Calculating analytical solution of δ_V involves heavy computation. In [20], the authors have proposed an iterative estimation algorithm that can obtain δ_V efficiently. According to the principle of Taylor expansion, we can use the second-order Taylor expansion to get $\Delta(\delta, x, \theta)$ as shown in Eq (7).

$$\Delta(\delta, x, \theta) \approx \frac{1}{2}\delta^T H(x,\theta)\delta \tag{7}$$

Where $H(x,\theta)$ is a Hessian matrix define by $H(x,\theta) = \nabla\nabla_\delta\Delta\ (\delta, x, \theta)|_{\delta=0}$. Under the assumption of Eq (7), δ_V can be replaced by the first dominant eigenvector $u(x,\theta)$ of Hessian matrix $H(x,\theta)$ described by Eq (8), while $\overline{x}$ operation means to get the unitary vector of x.

$$\begin{aligned} \delta_V &= \arg\max_{\delta} \left\{\delta^T H(x,\theta)\delta \text{ where } \|\delta\|_2 \leq \varepsilon\right\} \\ &\approx \varepsilon\overline{u(x,\theta)} \end{aligned} \tag{8}$$

However, using Eq (8) to calculate the virtual adversarial perturbation still requires a large amount of computation. Golub et al. [23] proposed an iterative and finite difference method to approximate the solution of δ_V. It was assumed that d was a random unit vector, and as long as d was not perpendicular to the eigenvector $u(x,\theta)$, so d could be calculated by an iterative method as shown in Eq (9).

$$d \leftarrow \overline{H(x,\theta)d} \tag{9}$$

After several iterations, d will converge to $u(x,\theta)$. In order to avoid directly calculating $H(x,\theta)$, $H(x,\theta)d$ can be calculated by finite difference method. Since the first derivative of $\Delta(\delta, x, \theta)$ is equal to zero when δ is 0, it can be obtained

$$\begin{aligned} H(x,\theta)d &\approx \frac{\nabla_\delta\Delta\ (\delta, x, \theta)|_{\delta=\xi d} - \nabla_\delta\Delta\ (\delta, x, \theta)|_{\delta=0}}{\xi} \\ &= \frac{\nabla_\delta\Delta\ (\delta, x, \theta)|_{\delta=\xi d}}{\xi} \end{aligned} \tag{10}$$

so d can be approximated with the repeated application of the following update rule as shown in Eq (11).

$$d \leftarrow \overline{\nabla_\delta\Delta(\delta, x, \theta)\ |_{\delta=\xi d}} \tag{11}$$

and δ_V could be obtained after using the estimated d

$$\delta_V = \varepsilon\ d \tag{12}$$

Algorithm 1 depicts this iterative approach and model training procedure using VAT.

Algorithm 1 Training KWS acoustic model using VAT

1: Initialize model parameters θ and set step=0
2: Given ε for Eq 4
3: Given α for Eq 6
4: Prepare training dataset $\{\mathbf{X}, \mathbf{Y}\}$
5: **while** model not converge **do**
6: Get a mini-batch $\{\mathbf{x}_m, \mathbf{y}_m|\mathbf{x}_\text{m} \subset \mathbf{X}, \mathbf{y}_\text{m} \subset \mathbf{Y}\}$
7: Initialize a random unit vector d in input space
8: **for** $(i = 0; i <$ Iters; i++) **do**
9: $d \leftarrow \overline{\nabla_\delta\Delta(\delta, x, \theta)\ |_{\delta=\xi d}}$
10: **end for**
11: $\delta_V = \varepsilon\ d$
12: Update parameters θ using Eq 6 as loss function
13: step = step +1
14: **end while**
15: **return** θ

As shown in Algorithm 1, we define the hyperparameters ε and α before training. While the model not converge, we first select a mini-batch from the training dataset and initialize a random unit vector d. Then we calculate δ_V introduced in Eq (3) and (4) by taking the derivative of $\Delta(\delta, x, \theta)$ with respect to δ. Line 8-10 in Algorithm 1 show the iterative δ_V estimation process. In practice, when $Iters = 1$, this algorithm performs well enough to generate adversarial perturbations. Finally, we use Eq (6) as the loss function and update the parameters of our model.

4. EXPERIMENTS

4.1. Corpus

We used wake-up data collected in far-field scenario to verify our KWS approach. The wake-up term is the combination of an English word and two Chinese syllables ("hello xiao gua"). Our dataset covers 206 different speakers (143 males and 63 females), and each speaker's collection includes positive utterances (with wake-up word) and negative utterances. In order to simulate various far-filed scenarios, we replayed all these utterances using a Hi-Fi loudspeaker with different speaker-to-microphone distances ranging from 1m to 5m in typical home environments. A 6-mic uniform circular array (UCA) with 7.5cm diameter was used to collect the multi-channel data. In total, there are ∼6.9 hour positive examples and ∼59 hours negative examples used as the training data. Some of the data were recorded with typical home noises like background sound from TV. The validation set includes ∼0.7h positive examples and ∼6.7h negative examples while the test set includes ∼0.9h positive examples and ∼7.8h negative examples. The speakers involved in the training and test sets were not overlapped. All these data were processed by our front-end processing techniques including automatic echo cancellation, beamforming, dereverberation and automatic gain control. Part of the raw data is from a public dataset[1] and the rest will be disclosed in the future.

4.2. Experimental setup

In our work, the frame-wise input acoustic features to the DS-DNN consist of 32 stacked frames of 40-dimensional log Mel filterbank energies features (23 left frames, current frame and 8 right frames). The DS-CNN has 3 output targets that corresponds to"hello", "xiao-gua" and a filler. As for the post-processing, w_{smooth} is 15 and w_{max} is 100. The KWS system will be activated when the confidence score for the current frame is greater than a pre-set threshold. Table 1 shows the architecture and parameter details of our DS-CNN model. Our experiments were conducted using Tensorflow and ADAM [24] was adopted as the optimizer.

Table 1. The DS-CNN architecture used for KWS, where Par. means memory footprint of model parameters, and Mul. represents the number of multiplication operations required by the model inference.

layer	channel	kenerl size	stride	Par.	Mul.
conv	128	16×8	2×4	65.6K	1.05M
DS conv ×4	128	3×3	1×1	5.12K	204.8K
avg_pool	128	16×10	1×1	-	-
dnn	128	-	-	20.5K	20.5K
softmax	4	-	-	0.4K	0.4K
Total	**4**	-	-	**107.3K**	**1.89M**

[1]http://www.speechocean.com/member/details/183.htmlc

For the VAT training, there are several hyper parameters need to be determined. For ξ and Iters in Algorithm 1, we adopted the values recommended from [20], where $\xi = 10$ and $Iters = 1$. As for α, for simplicity, we just set it to 1.

Table 2. FRR of different VAT and FGSM training strategies at 1.0 FAR per hour.

Type	Origin	Random	Neg-VAT	Pos-VAT	All-VAT
FRR (%)	5.64	5.35	4.65	4.71	**3.84**
Gain (%)	-	5.14	17.6	16.5	**31.9**
Type	-	-	Neg-FGSM	Pos-FGSM	All-FGSM
FRR (%)	-	-	**4.42**	4.60	13.0
Gain (%)	-	-	**21.6**	18.4	-130

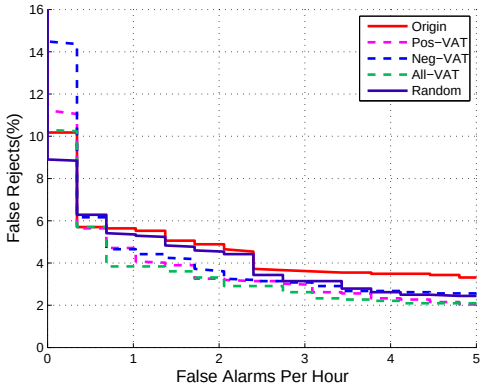

Fig. 2. ROC curves of different VAT training strategies ($\varepsilon = 0.1$).

4.3. Experimental results

Figure 2 shows ROC curves for models with different perturbation strategies. The "Origin" model is trained using the original training data with cross entropy loss only. Neg-VAT and Pos-VAT mean that VAT is applied to negative and positive data, respectively, while All-VAT denotes applying VAT to all the training data. For "Random", a random unit vector d is used to calculate LDS in Eq (6), indicating random regularization instead of adversarial regularization. Table 2 reports FRR at 1.0 FAR when $\varepsilon = 0.1$ for all the above models. Particularly for comparison, we also report the FRR results from adversarial example based data augmentation proposed in [15]. Here we still use FGSM-based adversarial example generation method. We see that all the three VAT based models can achieve significant improvements compared with the Random and Origin models. Specifically, Neg-VAT and Pos-VAT can significantly reduce the FRR, with 17.6% and 16.5% relative reduction, respectively, while All-VAT is the most effective one, which can bring 31.9% relative FRR reduction and outperform the Neg-FGSM data augmentation approach by 10.3% relative. It is worth mentioning that All-FGSM even makes the model performance worse. This is because unlike the RNN model used in [15], deep KWS uses stacked frames

to get context information. That will lead to some similar input acoustic features corresponding to different labels during the training process, thus resulting in a negative gain. On the contrary, the reason why an All-VAT perform better is that this method does not require label information. Considering the extraordinary performance from All-VAT, the remaining results are based on the All-VAT strategy.

4.4. Hyperparameter

Compared with other hyper parameters, we find that VAT is more sensitive to ε, which affects the upper bound of the allowed perturbation. For the other hyperparameters, we simply take the empirical values in [20], such as ξ is 10 and $Iters$ is 1. We examined various values of ε for the All-VAT model and the results are shown in Figure 3 and Table 3. It can be seen that the best result is achieved when ε is increased to 0.1, but after that the performance degrades substantially and even faces a negative gain when ε reaches 0.20.

Table 3. FRR of different ε for All-VAT training strategy at 1.0 FAR per hour.

ε	Origin	0.05	0.10	0.15	0.20
FRR (%)	5.64	4.58	**3.84**	4.25	6.11
Gain (%)	0	18.8	**31.9**	24.7	-8.33

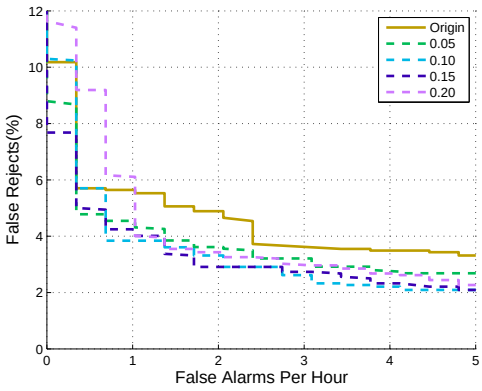

Fig. 3. ROC curves of different ε for All-VAT training strategy.

4.5. Robustness to perturbations

As we discussed in Section 3, VAT is able to smooth the model's output distribution with respect to the input by introducing LDS into the loss function during model training. Therefore, we believe that the model trained by VAT could be more robust to perturbations such as noises. In order to verify this, we separated our test set into two subsets – a clean one and a noisy one, with roughly equal size. The renewed FRR scores are shown in Table 4. Not surprisingly, VAT can obtain more significant improvement on the noisy subset than the clean subset.

Table 4. Performance of model on clean and noisy test dataset. FRR is at 1.0 FAR per hour.

Method	Clean subset		Nosiy subset	
	FRR (%)	Gain (%)	FRR (%)	Gain (%)
Origin	2.90	-	7.23	-
Pos-VAT	2.71	6.55	5.68	21.4
Neg-VAT	2.64	8.97	5.79	19.9
All-VAT	**2.60**	**10.3**	**4.69**	**35.1**
Random	2.80	3.44	6.86	7.61

4.6. Model convergence

In our experiments, we also find that, besides its superior performance, VAT can significantly accelerate model's convergence speed. Figure 4 compares the loss change trajectories for the Origin, Random and All-VAT models on the development set. It's obvious that All-VAT can converge better and faster than the other two.

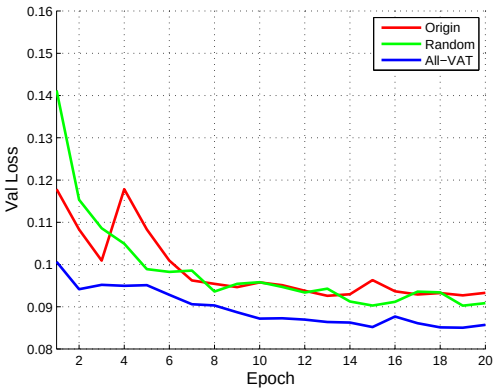

Fig. 4. Loss on validation dataset changes with the number of epochs

5. CONCLUSIONS

In this paper, we have proposed to improve a DS-CNN based small-footprint KWS system using the virtual adversarial training strategy. VAT aims to smooth model's distribution and thus to improve robustness, by explicitly introducing a distribution smoothness measure into the loss function. Compared with the cross entropy based loss function, the proposed VAT based loss function can significantly improve KWS performance. We have also experimentally compared different VAT training strategies and find that applying VAT to both positive and negative training examples is the most effective way. In summary, on our collected KWS corpus from far-field scenario, the proposed VAT training can achieve up to 31.9% relative FRR reduction at 1.0 FAR. The currrent work focuses on the robustness to adversarial perturbations on acoustic features. In the future, we will study the adversarial perturbations directly on raw waveforms.

6. REFERENCES

[1] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan, "Vocabulary independent spoken term detection," in *ACM SIGIR*, 2007, pp. 615–622.

[2] Petr Motlicek, Fabio Valente, and Igor Szoke, "Improving acoustic based keyword spotting using LVCSR lattices," in *ICASSP*, 2012, pp. 4413–4416.

[3] I Fan Chen, Chongjia Ni, Boon Pang Lim, Nancy F. Chen, and Chin Hui Lee, "A novel keyword+LVCSR-filler based grammar network representation for spoken keyword search," in *ISCSLP*, 2014, pp. 192–196.

[4] Minhua Wu, Sankaran Panchapagesan, Ming Sun, Jiacheng Gu, Ryan Thomas, Shiv Naga Prasad Vitaladevuni, Bjorn Hoffmeister, and Arindam Mandal, "Monophone-based background modeling for two-stage on-device wake word detection," in *ICASSP*, 2018, pp. 5494–5498.

[5] Binfeng Yan, Rui Guo, Xiaoyan Zhu, and Bo Zhang, "An approach of keyword spotting based on HMM," in *WCICA*, 2000, pp. 2757–2759.

[6] J Robin Rohlicek, William Russell, Salim Roukos, and Herbert Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *ICASSP*, 1989, pp. 627–630.

[7] Ch Choisy, "Dynamic handwritten keyword spotting based on the NSHP-HMM," in *ICDAR*, 2007, pp. 242–246.

[8] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in *ICASSP*, 2014, pp. 4087–4091.

[9] Zhehuai Chen, Yanmin Qian, and Kai Yu, "Sequence discriminative training for deep learning based acoustic keyword spotting," *Speech Communication*, 2018.

[10] George Retsinas, Giorgos Sfikas, Nikolaos Stamatopoulos, Georgios Louloudis, and Basilis Gatos, "Exploring critical aspects of CNN-based keyword spotting. a PHOCNet study," in *IAPR*, 2018, pp. 13–18.

[11] Santiago Ndez, Alex Graves, J Schmidhuber, and rgen, "An application of recurrent neural networks to discriminative keyword spotting," in *ICANN*, 2009, pp. 220–229.

[12] Changhao Shan, Junbo Zhang, Yujun Wang, and Lei Xie, "Attention-based end-to-end models for small-footprint keyword spotting," in *INTERSPEECH*, 2018, pp. 2037–2041.

[13] Yanzhang He, Rohit Prabhavalkar, Kanishka Rao, Wei Li, Anton Bakhtin, and Ian McGraw, "Streaming small-footprint keyword spotting using sequence-to-sequence models," in *ASRU*, 2017, pp. 474–481.

[14] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra, "Hello edge: Keyword spotting on microcontrollers," *arXiv preprint arXiv:1711.07128*, 2017.

[15] Xiong Wang, Sining Sun, Changhao Shan, Jingyong Hou, Lei Xie, Shen Li, and Xin Lei, "Adversarial examples for improving end-to-end attention-based small-footprint keyword spotting," in *ICASSP*, 2019.

[16] Sining Sun, Pengcheng Guo, Lei Xie, and Mei-Yuh Hwang, "Adversarial regularization for attention based end-to-end robust speech recognition," *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 27, no. 11, 2019.

[17] Sining Sun, Ching-Feng Yeh, Mei-Yuh Hwang, Mari Ostendorf, and Lei Xie, "Domain adversarial training for accented speech recognition," in *ICASSP*, 2018, pp. 4854–4858.

[18] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[19] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[20] Takeru Miyato, Shin Ichi Maeda, Masanori Koyama, and Shin Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *TPAMI*, vol. PP, no. 99, pp. 1–1, 2017.

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[22] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii, "Distributional smoothing with virtual adversarial training," *arXiv preprint arXiv:1507.00677*, 2015.

[23] Gene H. Golub and Henk A. Van Der Vorst, "Eigenvalue computation in the 20th century," *Journal of Computational & Applied Mathematics*, vol. 123, no. 1, pp. 35–65, 2000.

[24] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.