FEW-SHOT SOUND EVENT DETECTION

Yu Wang[1*] *Justin Salamon*[2] *Nicholas J. Bryan*[2] *Juan Pablo Bello*[1]

[1]Music and Audio Research Laboratory, New York University, NY, USA
[2]Adobe Research, San Francisco, CA, USA

ABSTRACT

Locating perceptually similar sound events within a continuous recording is a common task for various audio applications. However, current tools require users to manually listen to and label all the locations of the sound events of interest, which is tedious and time-consuming. In this work, we (1) adapt state-of-the-art metric-based few-shot learning methods to automate the detection of similar-sounding events, requiring only one or few examples of the target event, (2) develop a method to automatically construct a partial set of labeled examples (negative samples) to reduce user labeling effort, and (3) develop an inference-time data augmentation method to increase detection accuracy. To validate our approach, we perform extensive comparative analysis of few-shot learning methods for the task of keyword detection in speech. We show that our approach successfully adapts *closed-set* few-shot learning approaches to an *open-set* sound event detection problem.

Index Terms— Few-shot learning, sound event detection, keyword detection, keyword spotting, speech

1. INTRODUCTION

Locating perceptually similar sound events within a continuous recording is a basic, but important task for many audio applications. For example, animators need to locate particular sounds in music and SFX tracks to synchronize motion graphics, and podcasters have to edit out filler words (e.g. "ahhs" and "umms") to improve the flow of speech. Noise monitoring solutions require identifying specific sound events [1] and, more generally, labeling large audio datasets for training machine learning models often requires identifying all time locations where specific sound events occur [2–4]. However, current audio processing tools require users to listen through the entire recording and manually identify and label all the locations of the sound events of interest, which is both hard and tedious. A method to automate this process would save a significant amount of time and human effort.

Modern deep learning-based sound event recognition and detection methods typically require large amounts of data for training or fine-tuning models for specific applications [5–9]. As such, the application of deep learning models to detect unseen and/or rare sound classes with only few labels has been very limited. Interactive user-in-the-loop sound event detection was proposed in [10], but this work focused on reducing annotation time rather than improving machine accuracy. Different strategies for training audio classifiers with few data for the task of acoustic event recognition were investigated in [11], however, it focused on coping with limited data during *training*, whereas our goal is to train models that can generalize to unseen classes for which we have very few examples at *inference time*.

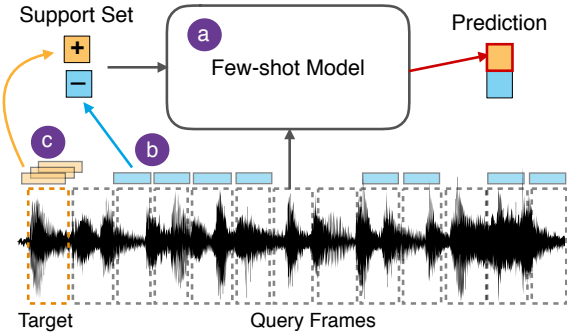

Fig. 1. Proposed few-shot sound event detection method. To detect a target sound event in a recording, we (a) apply metric-based few-shot learning, (b) automatically construct a set of negative (blue) examples needed for inference, and (c) propose an inference-time data augmentation method to generate more positive examples (orange).

Recently, studies have proposed to tackle this latter problem using few-shot learning, where a classifier must learn to recognize novel classes given only few examples from each [12, 13]. Traditional few-shot learning methods consider a *C-way K-shot* classification task as a *closed-set* classification problem of labeling an audio query with one of C unique class labels, given K labeled examples per class, where C is *fixed*. To the best of our knowledge, however, few-shot learning has not been applied to an *open-set* problem, such as sound event detection, where a previously unseen target sound needs to be detected in a sequence of unknown, previously unseen sounds from an unbounded number of sound classes.

In this paper, we propose to leverage few-shot learning for open-set sound event detection in order to identify perceptually similar sound events within a recording. In doing so, we (1) adapt prior metric-based few-shot learning approaches to the *open-set* sound event detection task, (2) propose a method to automatically construct a set of labeled negative examples required at inference time, and (3) propose an inference-time data augmentation method to increase detection accuracy, while reducing user-labeling effort. In Figure 1, we depict the proposed method with our key contributions mentioned above, which is applicable to a variety audio domains such as speech, music, and environmental sound. We evaluate our approach on speech and (4) provide extensive comparative analysis of few-shot learning for the application of sound-based keyword detection in speech. We show that our method achieves an average area under the precision-recall curve (AUPRC) of 75.42% for detecting unseen target keywords with only five labeled examples provided. Finally, we (5) show that our approach generalizes to unseen languages without requiring any retraining or fine tuning.

* This work was performed during an internship at Adobe Research. This work was partially supported by National Science Foundation award 1544753.

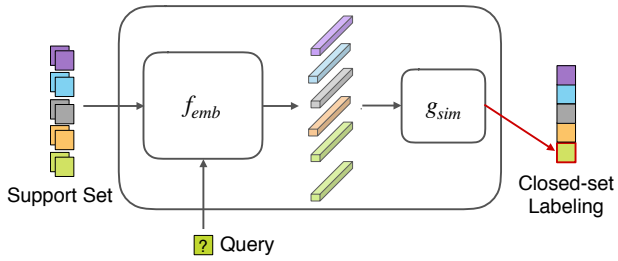

Fig. 2. Metric-based few-shot learning model (*5-way 2-shot*).

2. METHOD

2.1. Metric-based few-shot learning

In contrast to other types of few-shot learning methods [14–19], metric-based methods provide a simple framework that allows feed-forward inference, which is ideal for our purpose [20–23]. Few-shot learning models are often trained to solve the aforementioned *C-way K-shot* classification task, where C (ways) is the fixed number of classes to discriminate between, and K (shots) is the number of examples provided per-class at testing time. Figure 2 is an example of a *5-way 2-shot* classification task, where each color represents a class. Given a support set of $C \times K$ labeled examples, the goal of a metric-based model is to embed the support and query sets into a discriminative embedding space using a neural network f_{emb}, and correctly classify each query by measuring the similarity between the support and query embeddings with g_{sim}, which can be a fixed or learned metric function.

Training such a model using only $C \times K$ labeled examples would not generalize well to unseen classes at inference time, since K is often a small number in the range of one to five. To address this, *episodic training* has been proposed to exploit a large training set and extract transferrable knowledge by mimicking the few-shot inference problem during training [21]. In each training iteration, a training episode is formed by randomly selecting C classes from the training set. For each selected class, K samples are first selected to build a support set $\mathcal{S}$ of size of $C \times K$, while a disjoint set of q samples are selected to form a query set $\mathcal{Q}$ of size $C \times q$. The training objective is to minimize prediction loss of the samples in $\mathcal{Q}$ conditioned on $\mathcal{S}$. Therefore, in each training episode, the model is learning to solve the *C-way K-shot* classification task. By training with a large collection of episodes, each consisting of a different set of C classes, the model learns *how to learn* from limited labeled data and a class-agnostic discriminative ability.

In this work, we adapt four metric-based few-shot learning methods for open-set sound event detection and compare their performance. The models we compared include: Siamese Networks [20], Matching Networks [21], Prototypical Networks [22], and Relation Networks [23]. The main difference between these methods is g_{sim}, or the distance metric used to measure the similarity between the support and query embeddings to predict query labels. Siamese Networks, our baseline, do not incorporate few-shot classification or episodic training [24, 25]. Rather, they are trained using paired data from the same or different classes with a triplet loss and L1 distance [20, 25]. Matching Networks compute the cosine distance between the query embedding and each support embedding. Prototypical Networks compute the squared Euclidean distance between the query embedding and the class mean (prototype) of each support embedding. Relation Networks replace the fixed distance metrics with a learnable relation module.

2.2. Open-set sound event detection

Once a few-shot model is trained, we propose applying it to sound event detection as depicted in Figure 1. Given a few labeled examples of a target sound event we want to find within a long recording, we can apply a few-shot model by taking the labeled examples as a support set and the remaining frames as a query set. The similarity predicted by the model can then be used to form a detection curve.

As discussed in the introduction, the goal of sound event detection is to discriminate between the target sound and everything else, which is an *open-set* problem. The problem is then how do we model the negative class? A practical solution is to formulate the task as a binary classification problem where the negative class is comprised of all non-target sounds. This, however, would require the user to explicitly provide negative examples (to produce a negative support set), which is undesirable in terms of human effort. To address this, we make the assumption that the target sound event is relatively sparse and propose simply constructing negative examples by randomly selecting frames within the recording as shown in Figure 1(b). In this way, we can automatically generate a (partial) support set for the negative class, required for few-shot learning, without additional human effort.

2.3. Inference-time data augmentation

To further reduce user labeling effort, we also propose a novel strategy for augmenting the positive examples *at inference time* without additional human effort. For each user-provided example of the target sound event, we generate x additional examples by shifting the selection window in time in both directions by 50 ms increments. This is illustrated in Figure 1(c) with $x = 2$. The assumption here is that shifting the context window around the original example creates new examples that are slightly different (time-shifted) from the original one, but still contain the target sound event. In this way, the model obtains several positive examples from every user-provided example, increasing the robustness of the predictions.

3. EXPERIMENTAL DESIGN

3.1. Datasets

For experimentation, we use the Spoken Wikipedia Corpora (SWC) [26]. It contains audio recordings from volunteer readers speaking Wikipedia articles. We filter SWC in English by only keeping recordings with word-level time-alignment annotations, resulting in a subset consisting of 183 readers, approximately 700K aligned words and 9K classes, where a class is defined as a specific word spoken by a specific reader. The readers are partitioned into training, validation, and test sets with a 138:15:30 ratio. Audio recordings are downsampled to a sampling rate of 16 kHz. For each word instance, we take a half-second context window centered on the word and compute a 128 bin log-mel-spectrogram as the input to the model using the `librosa` [27] software package. We use a window length of 25 ms, hop size of 10 ms, and an fast Fourier transform size of 64 ms. To construct a *C-way K-shot* training episode, we randomly sample a reader from the training set, sample C word classes from the reader, and sample K instances per class as the support set. The query set is comprised of 16 separate word instances per each of the C classes [28]. While we evaluate our proposed approach on speech (due to the size of the available data), there is nothing speech-specific about our method and it can be directly applied to other audio domains such as music, bioacoustics, and environmental sound.

3.2. Model architecture and training

For all four metric-based few-shot learning methods, we use a standard convolutional neural network (CNN) for the embedding module f_{emb} [28]. It consists of four CNN blocks, each of which has a convolutional layer with a 3×3 kernel, a batch normalization layer, a ReLU activation layer, and a 2×2 maxpooling layer. We modify the size of the last pooling layer in each few-shot learning model such that the overall number of parameters in each architecture is roughly the same (120k-230k). Models are trained using the `Adam` optimizer in `PyTorch` with a learning rate of 0.001 for 60,000 episodes with early stopping.

3.3. Metrics

Our evaluation is based on the detection of unseen word classes in our test set recordings: we use 96 recordings from test readers in SWC. In each recording, we pick up to 10 target words, resulting in a total of 768 target keywords. For target words, we only consider words that occur at least 10 times in the recording. If there are more than 10 words that satisfy this condition, we sort the words by their number of occurrences, divide the sorted list into 10 equally sized bins, and sample one keyword per bin. In this way, we avoid only selecting words that are either very common or very rare.

To detect a keyword in a test recording, we first draw p instances of the keyword as a positive support set, then randomly draw n half-second windows from the recording as a negative support set. We explore various combinations of p and n in our evaluation. Given the support set, we apply the trained few-shot model to half-second windows centered on every (annotated) word in the test recording. The ground truth labels are defined as value one for windows containing the target keyword and zero for windows containing other words. We repeat this process 10 times for each target keyword and aggregate the results to compute a keyword-level area under precision-recall curve (AUPRC). We compute a recording-level AUPRC for each test recording by averaging all keyword-level AUPRC scores from that recording. Finally, we compute and report the mean and standard deviation across the 96 recording-level AUPRC scores. AUPRC is a common metric for evaluating binary classification models which factors in both precision and recall. It is particularly informative in scenarios where there is significant class imbalance, such as our case, where the negative class dramatically outweighs the positive class. AUPRC scores range from 0 (worst case) to 1 (best case, in our plots represented as 100%).

4. RESULTS

We present the results from four experiments, where we systematically vary (C, K, p, n, x) to see how each parameter affects model performance. First, we compare different few-shot models with different training configurations. Next, at inference time, we experiment with different strategies for modeling negative and positive examples. Lastly, we evaluate the performance of our best model on few-shot keyword detection in unseen languages.

4.1. Model comparison

We start by comparing models trained with different metric-based few-shot learning methods and combinations of (C, K). Note that for Siamese Networks the effective C is always two, since they are trained with paired examples. The performance is evaluated with $p \in \{1, 5\}$, without inference-time data augmentation, using the optimal number of negative examples n, as discussed in Section 4.2.

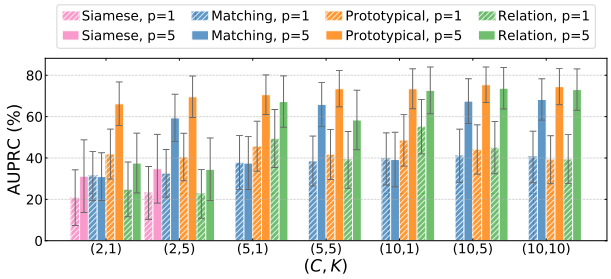

Fig. 3. AUPRC results for different few-shot models trained with different (C, K) combinations. Either one or five labeled examples p were provided at inference time.

The results are presented in Figure 3. We see that with K and p fixed, increasing C produces better model performance for all methods. This matches findings from previous studies in computer vision [22]: a higher number of ways during training makes the classification task harder, forcing the model to learn a more discriminative embedding space. This trend is particularly significant for Relation Networks, which benefit from more ways to improve the learnable relation module. We also experimented with training models with $C = 20$, but did not see further improvement due to the dramatically reduced amount of training data available (each training episode consists of words from the same reader and the number of valid readers drops significantly for large C).

With C and K fixed, $p = 5$ gives better model performance for almost all methods compared to $p = 1$. This is especially critical for Prototypical Networks. A prediction of Prototypical Networks is based on a fixed distance measurement between the embedding of the query and each class prototype, which is averaged over the examples in the class. Therefore, more positive examples result in a more stable representation of the target keyword, leading to more robust inference. Our results also confirm that Prototypical Networks achieve the best audio classification results when $K = p$, which was previously shown in the image domain [22], but never before for audio. Overall, the best performance is obtained by Prototypical Networks trained with $(C, K) = (10, 5)$, i.e. *10-way 5-shot*.

We also experimented with using an open-set formulation during *training*: rather than including various positive classes as ways, we trained models with 2-way 5-shot. The ways represent a positive and a negative class, and the shots for the negative class are drawn from multiple non-target classes, mimicking a 1-vs-all scenario for sound event detection. However, this form of training did not lead to any performance improvements and is not explored further in this work. A possible explanation is that the pressure on the model to learn a space that can discriminate between many different classes is reduced under this binary classification formulation.

4.2. Negative class modeling

In our next experiment, we study the impact of the number of negative support samples n on performance, presented in Figure 4. We see that model performance increases with increasing n and starts plateauing at $n = 50$, when n is high enough to capture the variance of the track. The best performance is achieved when $n =$ All, i.e. using all frames of the recording. This is ideal from an application standpoint, since we can just take the whole track as the negative support set without any sampling (and without overfitting n).

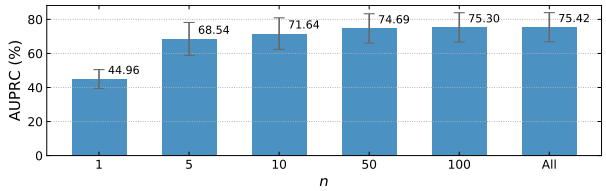

Fig. 4. AUPRC results as a function of the size of the negative support set, n. Results are from the best model: Prototypical Networks with $(C, K, p, x) = (10, 5, 5, 0)$.

4.3. Positive class modeling and inference-time augmentation

Next, we take a closer look at the positive support set by experimenting with different p and x. For each p from one to five, we try three inference-time data augmentation settings: $x = 0$, i.e. no augmentation; $x = 2$, i.e. for each labeled positive example we add two neighboring windows into the positive support set, one shifted 50 ms to its left and one shifted 50 ms to its right; and $x = 4$, where we add two neighbors on each side using 50 ms and 100 ms shifts.

Starting with the top plot of Figure 5, we see that model performance increases with increasing p for all x. This matches the intuition that the model should be able to make more accurate predictions when more examples of the target keyword are provided. Notably, the relative increase in performance diminishes as p gets larger. The model performs dramatically better when p goes from one to two or three, but the performance increase is less significant when p goes up from four to five. Having more than one example of the target sound event is definitely beneficial, but for scenarios where the annotation cost is high, we can safely limit ourselves to five or less examples without a significant compromise in performance.

While the increase in performance is significant when increasing p from one to two or three, when we increase x (augmentation), we see little improvement. A possible explanation is our use of a fixed half-second context window. For words shorter than half a second, the context window may include noise (parts of other words), while words longer than half a second are not fully captured by the context window. In both cases, our proposed augmentation may not result in a better representation of the target keyword. To explore this, in the bottom plot of Figure 5, we present the results when we limit the target keywords to those with a duration that closely matches our context-window, between 0.49 and 0.51 seconds. Now, we see a significant improvement with augmentation: $x = 2$ gives an 11.4 percentage point boost when $p = 1$. The improvement remains notable when $p = 2$. The effect of augmentation become less significant for larger p and x, as the variation between additional positive examples generated by augmentation become less significant. But, when there is only one example of the target sound event available, our inference-time augmentation approach provides a significant performance improvement with no additional human effort.

In future work, we plan to replace the fixed half-second context window with a dynamic context window duration, where the window length can adapt to the duration of the target sound event. Under this formulation, we expect our proposed inference-time data augmentation approach will be more effective.

4.4. Cross-language evaluation

It is important to note that even though we use recorded speech in our evaluation, our approach is completely language agnostic and does not rely at all on speech recognition at inference time. To demonstrate this, we take our best performing model, trained on English recordings only, and evaluate it on recordings spoken in other languages, namely Dutch and German. We use the Dutch and German versions of the SWC corpus for these experiments, following the same evaluation setup used for English as discussed in Section 3.3. Results for all three languages are presented in Table 1. Note, again, that the model is only trained on English recordings. We see that the model performs equally well on keyword spotting in Dutch, and even better in German, despite having never been trained on audio recordings in these languages. These results confirm that our model is able to recognize similar sounding events in a way which is robust to cross-language differences, and can easily generalize to unseen languages. It also shows potential for application in audio domains other than speech, such as music or environmental sound recordings.

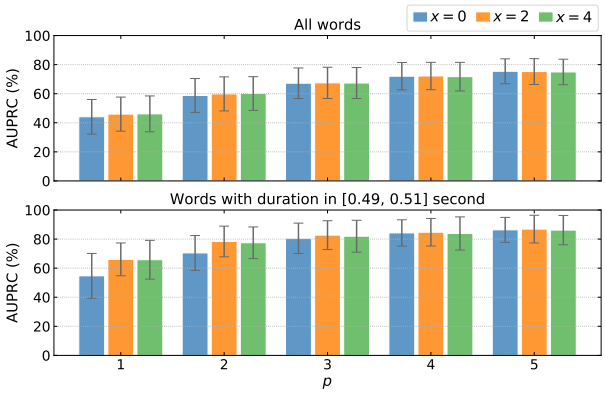

Fig. 5. AUPRC results for different p and x. Results are from the best model: Prototypical Networks with $(C, K, n) = (10, 5, \text{All})$. (Top) All target keywords are tested. (Bottom) Only keywords with duration close to half-second context window are considered.

Language	AUPRC(%)
English	75.42 ± 8.57
Dutch	75.14 ± 8.59
German	77.98 ± 6.94

Table 1. AUPRC results for detecting unseen keywords in SWC in English, Dutch, and German. Results are from the best model: Prototypical Networks with $(C, K, p, n, x) = (10, 5, 5, \text{All}, 0)$ trained on English only.

5. CONCLUSION

In this work, we propose a new method for locating perceptually similar sound events within a single continuous recording. We adapt metric-based few-shot learning methods by (1) modifying them to fit an *open-set* sound event detection problem formulation, (2) proposing a method to automatically construct a set of labeled negative samples without any additional human effort, (3) proposing an inference-time data augmentation method to increase detection accuracy, (4) performing an extensive evaluation for the task of few-shot keyword detection for speech, and (5) validating that our approach generalizes across languages without any retraining or fine tuning. In future work, we plan to explore replacing our fixed context-window with a dynamic context-window, compare our few-shot keyword detection method to state-of-the-art keyword spotting techniques, and apply and evaluate our model on other audio domains such as music and environmental sound recordings.

6. REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for monitoring, analyzing, and mitigating urban noise pollution," *Commun. ACM*, vol. 62, no. 2, pp. 68–77, Jan. 2019.

[2] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, New York, NY, USA, 2014, MM '14, pp. 1041–1044, ACM.

[3] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Reliable detection of audio events in highly noisy environments," *Pattern Recogn. Lett.*, vol. 65, no. C, pp. 22–28, Nov. 2015.

[4] A. Diment, A. Mesaros, T. Heittola, and T. Virtanen, "Tut rare sound events, development dataset," Mar. 2017.

[5] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *2014 22nd European Signal Processing Conference (EUSIPCO)*, Sep. 2014, pp. 506–510.

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

[7] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 559–563.

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

[9] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 25, no. 6, pp. 1291–1303, June 2017.

[10] B. Kim and . Pardo, "A human-in-the-loop system for sound event detection and annotation," *ACM Trans. Interact. Intell. Syst.*, vol. 8, no. 2, pp. 13:1–13:23, June 2018.

[11] J. Pons, J. Serr, and X. Serra, "Training neural audio classifiers with few data," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 16–20.

[12] S. Chou, K. Cheng, J. R. Jang, and Y. Yang, "Learning to match transient sound events using attentional similarity for few-shot sound recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 26–30.

[13] S. Zhang, Y. Qin, K. Sun, and Y. Lin, "Few-Shot Audio Classification with Attentional Graph Neural Networks," in *Proc. Interspeech 2019*, 2019, pp. 3649–3653.

[14] Li F., Rob F., and Pietro P., "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[15] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, ICML'16, pp. 1842–1850.

[16] T. Munkhdalai and H. Yu, "Meta networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2017, ICML'17, pp. 2554–2563.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2017, ICML'17, pp. 1126–1135.

[18] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[19] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *International Conference on Learning Representations*, 2018.

[20] G. R. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Workshop*, 2015.

[21] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems 29*, pp. 3630–3638. 2016.

[22] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems 30*, pp. 4077–4087. 2017.

[23] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 1199–1208.

[24] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proceedings of the 6th International Conference on Neural Information Processing Systems*, San Francisco, CA, USA, 1993, NIPS'93, pp. 737–744.

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1701–1708.

[26] A. Köhn, F. Stegen, and T. Baumann, "Mining the spoken wikipedia for speech data and beyond," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, May 2016.

[27] B. McFee, V. Lostanlen, M. McVicar, A. Metsai, S. Balke, C. Thomé, C. Raffel, et al., "librosa/librosa: 0.7.0," July 2019.

[28] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2019.