ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446970

ACTIVE LEARNING FOR SOUND EVENT CLASSIFICATION USING BAYESIAN NEURAL NETWORKS WITH GAUSSIAN VARIATIONAL POSTERIOR

Stepan Shishkin[1] *Danilo Hollosi*[1] *Stefan Goetze*[2] *Simon Doclo*[1,3]

[1] Fraunhofer Institute for Digital Media Technology IDMT,
Oldenburg Branch for Hearing, Speech and Audio Technology HSA, Germany
[2] Dept. of Computer Science, Speech and Hearing (SPandH), The University of Sheffield, United Kingdom
[3] Dept. of Medical Physics and Acoustics and Cluster of Excellence Hearing4all,
Carl von Ossietzky Universität Oldenburg, Germany

ABSTRACT

Manual annotation of audio material is cumbersome. Active learning aims at minimizing the annotation effort by iteratively selecting an *acquisition batch* of unlabeled data, asking a human to annotate the selected data and re-training a classifier until an *annotation budget* is depleted. In this paper we propose the Gaussian-dense active learning (GDAL) algorithm to train a sound event classifier. The classifier is a Bayesian neural network where the weights are normally distributed. This is in contrast to conventional neural networks where weights are not distributed, but have assigned values. The Bayesian nature of the classifier empowers GDAL to select acquisition batches from a set of unlabeled audio clips based on their estimated informativeness. Evaluation results on the UrbanSound8k dataset show that GDAL outperforms a state-of-the-art algorithm based on medoid active learning for all considered annotation budgets and an algorithm based on dropout active learning for sufficiently large annotation budgets.

Index Terms— sound event classification, active learning, Bayesian neural networks, PANN embeddings

1. INTRODUCTION

Sound event classification [1] aims at distinguishing events or situations based on properties of audio signals [2]. Some of its applications are wildlife [3] or environmental [4] monitoring and healthcare [5]. Training a sound event classifier requires a sufficiently large collection of annotated audio material. Providing manual annotations for such an audio corpus is often the most time-consuming part in the entire process of generating a sound event classifier.

In active learning (AL) [6, 7], the annotation process is integrated into the training process: the model learns from labels provided by a human annotator *on the fly*. The *AL process* iterates between selecting an *acquisition batch* of unlabeled data, collecting the annotations, and re-training the model on the updated dataset. As a result, the annotator does not need to label the entire dataset, but only the data that was selected by the AL algorithm.

Bayesian AL [8, 9] is a subset of AL where the model trained is Bayesian. A Bayesian model allows the AL algorithm to select acquisition batches based on the estimated informativeness of unlabeled data. A popular type of Bayesian models are Bayesian neural networks (BNNs) [10–14].

Bayesian AL algorithms have been deployed on multiple occasions to solve computer vision problems [8, 15–17]. However, they have not yet been applied to the sound event classification problem. This paper proposes Gaussian-dense active learning, in the following denoted as GDAL, a Bayesian AL algorithm that trains a sound event classifier.

GDAL makes use of transfer learning and semi-supervised learning to train a label-efficient BNN on a corpus of initially unlabeled sound clips. Semi-supervised learning is implemented by tailoring the Bayesian prior around the empirical data distribution - an approach not yet explored in the context of AL with BNNs. Evaluated on the UrbanSound8k dataset [18], GDAL consistently beats the MAL-PANN [19] baseline, while outperforming a modification of the DAL [19] baseline for sufficiently large annotation budgets.

2. ACTIVE LEARNING WITH BAYESIAN NEURAL NETWORKS

Developing an accurate classifier requires a collection of data where a sufficiently large portion of the dataset is annotated with the respective class labels. Collecting manual labels for the data can be tedious. To this end, AL is a machine learning paradigm which exploits the idea that a machine learning algorithm can be more label-efficient if it is allowed to choose which data is to be annotated [6].

Given is a set of class labels $\mathcal{C}$ and a *dataset* $\mathcal{D} = \{\mathbf{a}_i\}_{i=1...|\mathcal{D}|}$ of unlabeled audio clips $\mathbf{a}_i$, i.e. signals, with $|\mathcal{D}|$ denoting the cardinality of $\mathcal{D}$. The goal of an AL algorithm is to train a classifier that can accurately predict the respective class label of any clip $\mathbf{a} \in \mathcal{D}$. To do so, an AL algorithm is allowed to request an annotator to assign labels for up to A clips, where A is the so-called *annotation budget*. An AL algorithm does not have to request all A labels at once, but can do so iteratively, by querying smaller *acquisition batch*es and re-training the classifier on each iteration. Whenever a clip $\mathbf{a} \in \mathcal{D}$ is annotated with a class label $c \in \mathcal{C}$, the tuple $(\mathbf{a}, c)$ is added to the *labeled dataset* $\mathcal{L} = \{(\mathbf{a}_i, c_i)\}_{i=1...|\mathcal{L}|}$. The size of an acquisition batch we denote as $\Delta|\mathcal{L}|$.

Bayesian AL is a subclass of AL algorithms where the classifier does not define a single mapping from input $\mathbf{a}$ to class probabilities,

but is capable of representing a Bayesian distribution over mappings. By defining some prior distribution $p(\mathbf{w})$ over mappings $\mathbf{w}$, and observing human annotations in $\mathcal{L}$, the posterior distribution $p(\mathbf{w}|\mathcal{L})$ over mappings can be constructed via the Bayes rule

$$p(\mathbf{w}|\mathcal{L}) = p(\mathbf{w}) \prod_{(\mathbf{a},c)\in\mathcal{L}} p(c|\mathbf{w},\mathbf{a}) \cdot \text{const.} \quad (1)$$

where $p(c|\mathbf{w},\mathbf{a})$ is the likelihood of observed annotation (class label) c for input $\mathbf{a}$ predicted by a mapping $\mathbf{w}$. A Bayesian AL algorithm selects unlabeled data for annotation based on the Bayesian posterior distribution: a data sample is deemed more informative when the different mappings in the Bayesian posterior predict different classes [8, 9, 15].

A popular family of models capable of representing a Bayesian distribution over mappings are Bayesian neural networks (BNNs) [10, 12, 20]. For these, a mapping from input to class probabilities corresponds to a configuration of network weights. While the Bayesian posterior $p(\mathbf{w}|\mathcal{L})$ in (1) is not tractable, it can be approximated. To this end, a BNN defines a *variational distribution* $q(\mathbf{w}|\boldsymbol{\theta})$ over weights, which depends on some parameter set $\boldsymbol{\theta}$. This variational distribution is fitted towards the Bayesian posterior $p(\mathbf{w}|\mathcal{L})$ by finding a $\boldsymbol{\theta}$ that minimizes an objective that is based on the evidence lower bound (ELBO) [20]

$$L(\boldsymbol{\theta}) = \mathrm{KL}\Big(q(\mathbf{w}|\boldsymbol{\theta}) \Big\| p(\mathbf{w})\Big) - \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})} \left[\sum_{(\mathbf{a},c)\in\mathcal{L}} \log p(c|\mathbf{a},\mathbf{w}) \right], \quad (2)$$

where $\mathrm{KL}(\cdot\|\cdot)$ is the Kullback-Leibler divergence and $\mathbb{E}$ denotes the expectation.

While BNNs have been employed for AL tasks, the algorithm proposed in this paper, GDAL (cf. next Section for more details) is the first application of a BNN to solve the AL task in the domain of sound event classification.

3. GAUSSIAN-DENSE ACTIVE LEARNING (GDAL)

GDAL is comprised of a feature extractor, a BNN classifier and an acquisition mechanism. The feature extractor (cf. Section 3.1) makes use of transfer learning by feeding clips $\mathbf{a}_i$ into a pre-trained neural network and obtaining respective feature vectors $\mathbf{x}_i$.

The classifier (cf. Section 3.2) operates on the feature vectors. It models the Bayesian posterior distribution over mappings from a feature vector to the respective predicted label. The model learns from the empirical clip distribution by explicitly incorporating it into the Bayesian prior. To our knowledge such an approach has not yet been explored in the context of AL with BNNs.

For each iteration of the AL process, the acquisition mechanism (cf. Section 3.3) is invoked to select a batch of unlabeled clips for annotation, and the classifier is re-trained on the updated dataset. The acquisition mechanism incorporates two different approaches, a geometric approach and an information-theoretic approach, to find most informative unlabeled clips for annotation.

3.1. Feature extractor

GDAL uses a pre-trained audio neural network (PANN) [21] to generate a 2048-dimensional feature vector $\mathbf{x}$ for each clip $\mathbf{a}$ in the dataset $\mathcal{D}$. As a result, all signals $\mathbf{a}_i$ are transformed into feature vectors $\mathbf{x}_i$, upon which the classifier (Section 3.2) and the acquisition mechanism (Section 3.3) operate.

3.2. Classifier

GDAL employs a Bayesian classifier which models different mappings from input clip $\mathbf{a}$ to class probabilities. This allows GDAL to quantify the informativeness of unlabeled clips and incorporate this quantity into the acquisition mechanism (Section 3.3).

The classifier is shown in Figure 1. An audio clip $\mathbf{a}$ is transformed into the respective feature vector $\mathbf{x}$. The feature vector passes through a dense layer followed by a softmax layer to produce a class probability distribution.

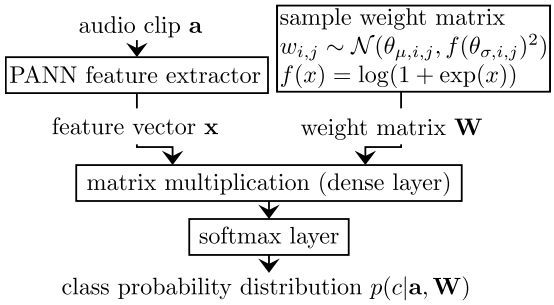

Fig. 1. Computation graph of the classifier in GDAL.

The dense layer in Figure 1 multiplies a weight matrix $\mathbf{W}$ with a feature vector $\mathbf{x}$. Crucially, instead of assigning deterministic values to the entries of $\mathbf{W}$, the weight matrix is sampled from a probability distribution. Throughout the AL process, this probability distribution is fitted towards the Bayesian posterior distribution over $\mathbf{W}$. In other words, GDAL trains a BNN (cf. Section 2) where the weights, denoted as $\mathbf{w}$ in (1) and (2), are given by the weight matrix $\mathbf{W}$.

Training a BNN requires defining a Bayesian prior $p(\mathbf{W})$ (cf. Section 3.2.1) and a variational Bayesian posterior parameterization $q(\mathbf{W}|\boldsymbol{\theta})$ (cf. Section 3.2.2) before optimizing the resulting objective function (2) (cf. Section 3.2.3).

3.2.1. Bayesian prior & data-driven regularization

To infer the Bayesian distribution of the network weights $\mathbf{W}$ in the dense layer in Figure 1, a prior distribution $p(\mathbf{W})$ over those weights needs to be defined. To make the model learn from unlabeled data typically present abundantly in AL settings, we propose a prior which explicitly incorporates the empirical data distribution:

$$p(\mathbf{W}) = \prod_{i,j} \mathcal{N}\Big(w_{i,j}\Big|0,\sigma_p^2\Big) \cdot \exp\left(\lambda \sum\nolimits_{\mathbf{a}\in\mathcal{D}} \max_c \log p(c|\mathbf{W},\mathbf{a})\right) \cdot \text{const.} \quad (3)$$

where i and j denote the rows and columns of $\mathbf{W}$.

The first factor in (3) is the Gaussian prior over network weights $\mathbf{W}$, widely used in the literature [12, 20] to apply weight regularization, i.e. enforce small weights. It is regulated by the standard deviation σ_p of the Gaussian term: smaller σ_p leads to more weight decay.

The second factor is a product over all clips in the dataset $\mathcal{D}$ and makes the prior favor weight matrices $\mathbf{W}$ that result in confident

class predictions for all labeled and unlabeled clips. This can be seen as a generalization of pseudo-label training [22] to BNNs. The strength of this *data-driven regularization* is determined by the non-negative parameter λ: higher λ incurs higher prior preference for weight matrices that give confident class predictions. Although the general idea to incorporate data into the prior is not new [20], to our knowledge, a data-driven regularization as in (3) has not yet been explored for BNNs in AL tasks.

3.2.2. Variational Bayesian posterior

To approximate the Bayesian posterior over weights $\mathbf{W}$ in Figure 1 the variational distribution $q(\mathbf{W}|\boldsymbol{\theta})$ needs to be parameterized. To this end, GDAL employs a fully factorized Gaussian variational posterior [12, 20, 23]. For that, the variational parameters $\boldsymbol{\theta}$ in (2) are defined as $\boldsymbol{\theta}=(\boldsymbol{\Theta}_\mu,\boldsymbol{\Theta}_\sigma)$, i.e. a tuple of two matrices, each with the same dimensions as $\mathbf{W}$. The variational distribution is

$$q(\mathbf{W}|\boldsymbol{\Theta}_\mu,\boldsymbol{\Theta}_\sigma)=\prod\nolimits_{i,j}\mathcal{N}\left(w_{i,j}\middle|\theta_{\mu,i,j},f(\theta_{\sigma,i,j})^2\right) \quad (4)$$

$$f(\theta_{\sigma,i,j})=\log(1+\exp(\theta_{\sigma,i,j})), \quad (5)$$

where the standard deviation in (4) is defined by applying the softplus function (5) to $\boldsymbol{\Theta}_\sigma$ as in [20]. In other words, the weights in the dense layer are distributed independently and normally, whereby the mean is defined by the parameter $\boldsymbol{\Theta}_\mu$, and the variance by $\boldsymbol{\Theta}_\sigma$.

3.2.3. Optimization

BNNs are trained by minimizing the objective (2). The prior (3) in conjunction with the variational parameterization (4) amount to the objective

$$\begin{aligned}L(\boldsymbol{\Theta}_\mu,\boldsymbol{\Theta}_\sigma)=&\sum_{i,j}\left(\log\sigma_p-\log f(\boldsymbol{\Theta}_{\sigma,i,j})+\frac{f(\boldsymbol{\Theta}_{\sigma,i,j})^2+\boldsymbol{\Theta}_{\mu,i,j}^2}{2\sigma_p^2}\right)\\&-\lambda\mathbb{E}_{q(\mathbf{W}|\boldsymbol{\Theta}_\mu,\boldsymbol{\Theta}_\sigma)}\left[\sum\nolimits_{\mathbf{a}\in\mathcal{D}}\max_{c\in\mathcal{C}}\log p(c|\mathbf{a},\mathbf{W})\right]\\&-\mathbb{E}_{q(\mathbf{W}|\boldsymbol{\Theta}_\mu,\boldsymbol{\Theta}_\sigma)}\left[\sum\nolimits_{(\mathbf{a},c)\in\mathcal{L}}\log p(c|\mathbf{a},\mathbf{W})\right]+\text{const.},\end{aligned} \quad (6)$$

with the set of clips $\mathcal{D}$ and the labeled set $\mathcal{L}$.

The first term in (6) is the KL divergence between the Gaussian variational posterior (4) and the Gaussian term in the prior (3). The second term results from the KL divergence between the variational posterior and the data-driven regularization term in (3). The third term is the likelihood contribution to the objective.

Unlike the first term, the second and the third cannot be computed analytically, and are estimated via Monte Carlo sampling. First, a minibatch of B clips is sampled, where one half is drawn from $\mathcal{D}$ and used to compute the second term, and the other half is drawn from $\mathcal{L}$ to compute the third term. For each sampled clip, the weight matrix $\mathbf{W}$ is sampled n_{train} number of times. The loss computed from a minibatch is backpropagated to $(\boldsymbol{\Theta}_\mu,\boldsymbol{\Theta}_\sigma)$ and the parameters of the variational distribution are updated via Adam. On each iteration of the AL process, parameter update is repeated several times.

3.3. Acquisition mechanism

GDAL employs a two-stage approach to select unlabeled clips for annotation. To select the first K clips, K-medoid clustering [24] is done in the feature space, and the medoid of each of the K clusters is annotated. The distance metric s between two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is based on the cosine similarity:

$$s(\mathbf{x}_i,\mathbf{x}_j)=1-\frac{\mathbf{x}_i^T\mathbf{x}_j}{\|\mathbf{x}_i\|_2\cdot\|\mathbf{x}_j\|_2} \quad (7)$$

After the initial acquisition of K annotations, each further acquisition is done via the BatchBALD algorithm [15], which employs the Bayesian model described in Section 3.2 to select an acquisition batch of $\Delta|\mathcal{L}|$ clips that maximizes the mutual information between the weight matrix $\mathbf{W}$ and the class label c, thus seeking the most informative clips.

4. BASELINE ALGORITHMS

In Section 5.5, GDAL's performance is compared against two state-of-the-art AL algorithms for sound classification: one based on medoid active learning (MAL) and another based on dropout active learning (DAL).

MAL [25] splits sound clips into clusters by performing K-medoid clustering [24] with mean cluster size κ. Given the annotation budget A, the medoids of the A largest clusters are annotated, and the medoid labels are propagated to the other clips in the respective clusters. A support vector machine classifier is then trained on ground truth and propagated labels. While the original MAL algorithm operates on features that are based on mel frequency cepstral coefficients, a PANN-embedding-based modification, denoted as MAL-PANN [19], has been shown to achieve significantly better performance. MAL-PANN is evaluated for the mean cluster size $\kappa=4$, same as in [25].

DAL [19] uses a classifier architecture similar to the one in Figure 1. Instead of defining a Gaussian distribution over weights, DAL achieves randomized predictions by applying random dropout masks to the PANN feature vector. Unlabeled clips are selected by making each prediction cast a vote in favor of one class, and picking the clip with the highest vote entropy, one clip per iteration of the AL process. Unlike GDAL, DAL incorporates unlabeled data into the training by assigning pseudo-labels, and training against those. For a fair comparison, DAL is modified to use the same acquisition batch size as in GDAL, and the initial acquisition is identical to GDAL's (i.e. based on K-medoid clustering).

5. EXPERIMENTS

The performance of GDAL and the baseline methods is evaluated by simulating an AL process and measuring the classification accuracy at each iteration. Section 5.1 describes the dataset and the performance metric. The default parameter values of GDAL are listed in Section 5.2. Finally, experimental results are presented in Sections 5.3 to 5.5.

5.1. Dataset and performance metric

The AL process is simulated on the UrbanSound8k [18] dataset, which consists of 8732 short, weakly labeled clips, each belonging to one of 10 classes.

First, the clips in the *training* split (without the class labels) are presented to the AL algorithm; this is the dataset $\mathcal{D}$ in (3) and (6). Manual annotations are simulated by revealing the ground truth labels to the algorithm. In each iteration of the simulated AL process, the performance is evaluated as the macro-recall [26] on the *test* split, i.e. the percentage of correctly predicted class labels, averaged over all classes.

AL processes are simulated for annotation budgets A up to 500 for baseline comparisons, and up to 100 for other experiments. Each experiment is conducted on 10 trials, measuring the mean macro-recall via 10-fold cross-validation. 80% confidence intervals for the mean macro-recall are computed via bootstrapping and shown as shaded areas in all following plots.

5.2. Default parameter values in GDAL

In the following, we define GDAL's default parameter values used for experiments. The prior $p(\mathbf{W})$ in (3) over network weights $\mathbf{W}$ is defined by the standard deviation σ_p of the Gaussian term and the data regularization strength λ, which are set to $\sigma_p = 30$, $\lambda = 10$. The optimization-related parameters (cf. Section 3.2.3) are the minibatch size $B = 4096$, the number of weight samples for each input sample $n_{\text{train}} = 2048$ and the learning rate of the Adam optimizer which is set to 0.1. For the first acquisition, GDAL acquires $K = 30$ annotations via K-medoid-clustering, and applies 6400 backpropagations to train the classifier. For each further iteration, $\Delta|\mathcal{L}| = 3$ labels are acquired via BatchBALD, and 2048 backpropagations are done.

The choice of σ_p and λ was motivated by preliminary experiments in which the classifier's performance on the *training* split was analysed, i.e. all clips in the UrbanSound8k dataset that are not in the *test* split. From all tested combinations, we chose the one with the strongest regularization (i.e. highest λ, lowest σ_p) just before the performance starts dropping. The other parameters were chosen to result in a reasonable computation time.

The impact of the prior parameters λ and σ_p on GDAL's performance is studied in Sections 5.3 and 5.4, respectively. For that, multiple experiments are conducted for different values for one parameter, while the other is fixed at the default value defined above.

Finally, a comparison with baseline methods is performed in Section 5.5, while all of GDAL's parameters are at their default values.

5.3. Varying the strength λ of the data-driven regularization

The parameter λ in (3) regulates the impact of unlabeled audio clips on the classifier.

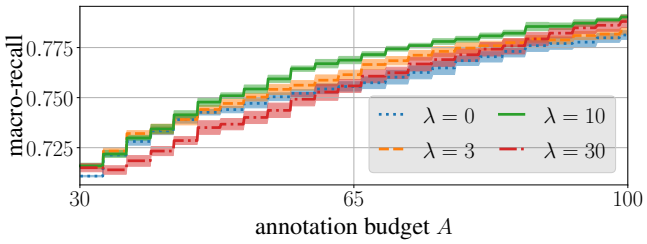

Fig. 2. Macro-recall over annotation budget A up to 100 for varying data-driven regularization strength λ.

As λ is varied, best performance is observed for $\lambda = 10$ as shown in Figure 2. The extreme case of $\lambda = 0$ makes the model ignore unlabeled data, whereas too high λ presumably leads to a poor variational posterior approximation, especially for low number of annotations, ultimately resulting in low classification accuracy. Most importantly, Figure 2 shows that data-driven regularization with moderate strength has a beneficial effect on GDAL's performance.

5.4. Varying the width σ_p of the Gaussian term in the prior

The standard deviation σ_p of the Gaussian term in (3) defines the strength of the weight decay in the dense layer shown in Figure 1.

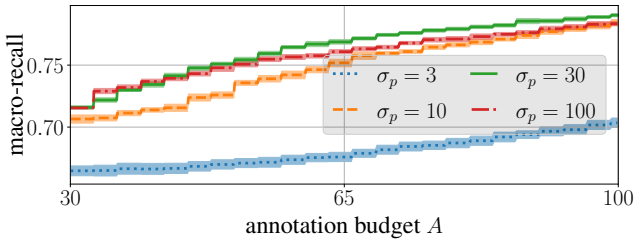

Fig. 3. Macro-recall over annotation budget A up to 100 for different widths σ_p of the Gaussian factor in the prior.

Simulations of the AL process with different values of σ_p are shown in Figure 3. Best performance is observed for intermediate levels of σ_p around 30. Too weak ($\sigma_p = 100$) or too strong ($\sigma_p = 3$) weight decay worsens the performance, whereby too strong weight decay has a more detrimental effect. This is in analogy to conventional neural networks: a network's performance can be boosted by a small weight decay, but degrades if the decay is too strong [27].

5.5. Comparison with baselines

GDAL's performance is compared with baseline methods (cf. Section 4) in Figure 4.

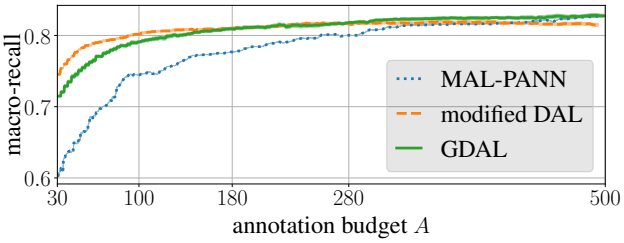

Fig. 4. Macro-recall over annotation budget A up to 500 for GDAL and baseline methods.

The proposed GDAL clearly outperforms the MAL-PANN approach, while performing similarly to the DAL approach. For annotation budgets above ca. 280, GDAL outperforms all other baselines.

6. CONCLUSION

In this paper we propose and analyse GDAL, a Bayesian AL algorithm for solving the sound event classification task. GDAL is label efficient due to its application of pre-trained feature extractors, incorporation of unlabeled data into the training process, and a smart acquisition mechanism that is based on estimated informativeness of unlabeled clips. For sufficiently large annotation budgets, GDAL was shown to beat state-of-the-art baselines.

7. REFERENCES

[1] W. Wang, *Machine Audition: Principles, Algorithms, and Systems*, Information Science Reference, Hershey, USA, 2011.

[2] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*, Springer, 2018.

[3] S. Kahl, M. Clapp, W. Hopping, H. Goëau, H. Glotin, R. Planqué, W.-P. Vellinga, and A. Joly, "Overview of BirdCLEF 2020: Bird Sound Recognition in Complex Acoustic Environments," in *Proc. Conf. and Labs of the Evaluation Forum (CLEF)*, Thessaloniki, Greece, 2020.

[4] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[5] J. Laguarta, F. Hueto, and B. Subirana, "COVID-19 Artificial Intelligence Diagnosis Using Only Cough Recordings," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 1, pp. 275–281, 2020.

[6] B. Settles, "Active Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[7] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical Image Analysis*, vol. 71, 2021.

[8] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian Active Learning with Image Data," in *Proc. Int. Conf. on Machine Learning (ICML)*, Sydney, Australia, 2017, pp. 1183–1192.

[9] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian Active Learning for Classification and Preference Learning," *ArXiv*, vol. abs/1112.5745, 2011.

[10] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. Int. Conf. on Machine Learning (ICML)*, New York, USA, 2016, pp. 1050–1059.

[11] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," in *Proc. Int. Conf. on Machine Learning (ICML)*, Lille, France, 2015, pp. 1613–1622.

[12] D. P. Kingma, T. Salimans, and M. Welling, "Variational Dropout and the Local Reparameterization Trick," in *Proc. Int. Conf. on Neural Information Processing Systems (NeurIPS)*, Cambridge, USA, 2015, pp. 2575–2583.

[13] D. Molchanov, A. Ashukha, and D. Vetrov, "Variational Dropout Sparsifies Deep Neural Networks," in *Proc. Int. Conf. on Machine Learning (ICML)*, Sydney, Australia, 2017, pp. 2498–2507.

[14] C. Louizos, K. Ullrich, and M. Welling, "Bayesian Compression for Deep Learning," in *Proc. Int. Conf. on Neural Information Processing Systems (NeurIPS)*, New York, USA, 2017, pp. 3290–3300.

[15] A. Kirsch, J. van Amersfoort, and Y. Gal, "BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning," in *Proc. Int. Conf. on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2019, pp. 7026–7037.

[16] W. H. Beluch, T. Genewein, A. Nurnberger, and J. M. Kohler, "The Power of Ensembles for Active Learning in Image Classification," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, 2018, pp. 9368–9377.

[17] V. Rakesh and S. Jain, "Efficacy of Bayesian Neural Networks in Active Learning," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2601–2609.

[18] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research," in *Proc. ACM Int. Conf. on Multimedia (ACMMM)*, Orlando, USA, 2014, pp. 1041–1044.

[19] S. Shishkin, D. Hollosi, S. Doclo, and S. Goetze, "Active learning for sound event classification using monte-carlo dropout and PANN embeddings," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, Nov. 2021, pp. 150–154.

[20] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, "Hands-on bayesian neural Networks—A tutorial for deep learning users," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 29–48, May 2022.

[21] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[22] D.-H. Lee, "Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks," *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 2013.

[23] S. Wang and C. Manning, "Fast dropout training," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, June 2013, vol. 28, pp. 118–126.

[24] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3336–3341, 2009.

[25] Z. Shuyang, T. Heittola, and T. Virtanen, "Active learning for sound event classification by clustering unlabeled data," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017, pp. 751–755.

[26] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: An Overview," *arXiv:2008.05756*, Aug. 2020.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts, 2016.