Interspeech 2022
18-22 September 2022, Incheon, Korea

Analysis of Self-Supervised Learning and Dimensionality Reduction Methods in Clustering-Based Active Learning for Speech Emotion Recognition

Einari Vaaras[1], *Manu Airaksinen*[2], *Okko Räsänen*[1]

[1]Unit of Computing Sciences, Tampere University, Finland
[2]Helsinki University Hospital, Helsinki, Finland

einari.vaaras@tuni.fi, manu.airaksinen@hus.fi, okko.rasanen@tuni.fi

Abstract

When domain experts are needed to perform data annotation for complex machine-learning tasks, reducing annotation effort is crucial in order to cut down time and expenses. For cases when there are no annotations available, one approach is to utilize the structure of the feature space for clustering-based active learning (AL) methods. However, these methods are heavily dependent on how the samples are organized in the feature space and what distance metric is used. Unsupervised methods such as contrastive predictive coding (CPC) can potentially be used to learn organized feature spaces, but these methods typically create high-dimensional features which might be challenging for estimating data density. In this paper, we combine CPC and multiple dimensionality reduction methods in search of functioning practices for clustering-based AL. Our experiments for simulating speech emotion recognition system deployment show that both the local and global topology of the feature space can be successfully used for AL, and that CPC can be used to improve clustering-based AL performance over traditional signal features. Additionally, we observe that compressing data dimensionality does not harm AL performance substantially, and that 2-D feature representations achieved similar AL performance as higher-dimensional representations when the number of annotations is not very low.

Index Terms: active learning, unsupervised learning, contrastive learning, manifold learning, speech emotion recognition

1. Introduction

In many complex real-world machine-learning applications, annotating data can be expensive and time-consuming. This is often the case particularly in situations where domain experts are needed to carry out the annotation process [1, 2]. For such cases, active learning (AL) algorithms can be used to reduce human annotation effort and to produce machine-learning models that perform well with limited labeled data [3]. For example, data scarcity is an ever-present problem in deploying speech emotion recognition (SER) systems to new domains [4].

There are a number of different approaches for AL, of which by far the most popular methods are based on uncertainty or confidence scores of a classifier trained on already-labeled data (e.g. [4–8]). However, when the maximum number of labels that can be manually assigned, also known as the *labeling budget*, adds up to only a small subset of the data, the aforementioned AL approaches cannot be applied. This is because these methods often require a rather large number of annotated samples before they can outperform random sampling [9].

In the absence of any existing annotated data, a potential approach to AL is to utilize the distributional properties of the dataset with clustering-based AL methods (e.g. [10–13]). These methods rely heavily on how the samples are organized in the feature space (i.e. the choice of features) and what distance metric is used, as the methods need to use these two to cluster the data points and to prioritize the order in which cluster samples are provided for human annotators. This puts particular emphasis on how the features behave in the given metric space with respect to the analysis task at hand. In addition, the dimensionality of the used features poses a potential challenge for the AL algorithms, as accurate data density estimation from finite data becomes less accurate in higher-dimensional spaces [14]. Hence, it could be beneficial for AL if there was a systematic way of representing data samples in a low-dimensional feature space. Moreover, a lower dimensionality would heavily reduce the computational complexity of AL algorithms. Taken to the extreme, a 2-D feature space could be used to combine AL with data visualization and efficient human-based data exploration and reorganization, such as in the annotation platform described in [15].

While the use of standard acoustic features such as i-vectors [12] or MFCCs [16] has been applied to previous AL clustering approaches, there are nowadays a number of highly promising unsupervised methods for learning feature spaces (e.g. [17–19]). These methods can learn linearly separable representations for many speech phenomena of interest (e.g. [17]), and could potentially be used to learn organized feature spaces for clustering-based AL algorithms. However, the aforementioned methods typically result in high-dimensional feature spaces which can impose challenges for AL algorithms, as discussed above. Also, these methods have been used primarily for learning features that are relevant for supervised downstream tasks, but not for intermediate tasks such as clustering, where data grouping is equally important to data separability (see Fig. 1 for an example).

In the present study, we combine AL with unsupervised learning and dimensionality reduction methods in order to seek answers to the questions of whether self-supervised representation learning (SSRL) can improve AL performance over classical signal features, and how AL performance depends on input feature dimensionality. While doing so, the work also sheds light on the question of how the local and global topology of a feature space affects clustering-based AL systems. In order to answer these questions, we conduct a set of experiments involving a

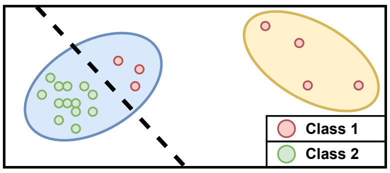

Figure 1: *A toy example for 2-D data, clustered into two clusters (blue and yellow) with k-means clustering using Euclidean distance. Although the classes are linearly separable (dashed line), the data points are not organized into class-specific clusters.*

10.21437/Interspeech.2022-329

simulated annotation procedure and a clustering-based AL algorithm from [16]. We use SER as our test case using four different SER corpora and with two distinct classification tasks. Our ultimate goal is to find functioning practices for developing machine-learning algorithms for applications where access to human expert labels is expensive. Hence, our primary focus is in cases with a low labeling budget, although we also include higher labeling budgets in our present experiments.

2. Methods

Fig. 2 depicts a block diagram of the present experiments. First, a standard acoustic feature (log-mel) representation is obtained from an input signal. Then, an SSRL algorithm called contrastive predictive coding (CPC) [17] is applied to the data. CPC aims to produce linearly separable features that can be used to predict signal evolution over time [20, 21]. CPC has already been successfully used with clustering-based approaches (e.g. [21–23]) and also produces features that separate suprasegmental properties such as speaker identities [17]. However, to the best of our knowledge, [24] is the only study so far using CPC for AL.

Next, an array of alternative dimensionality reduction methods is optionally applied to the log-mel and CPC features. In the present study, we explore the use of t-distributed stochastic neighbor embedding (t-SNE) [25], nonlinear bottleneck autoencoders (AEs), and principal component analysis (PCA) for dimensionality reduction. Among these, t-SNE preserves the local structure of the high-dimensional data while also revealing some global aspects, such as clusters at multiple scales [25]. In contrast, PCA simply maps the data into principal axes of variation without differentially altering the local and global metric structure of the feature space. Bottleneck AEs, on the other hand, attempt to learn a low-dimensional feature embedding from which the original input features can be reconstructed.

Finally, we run a simulated AL-based speech emotion annotation and a SER classifier deployment procedure using a support vector machine (SVM) classifier, similar to [26]. For this, we use a clustering-based AL algorithm together with each possible combination of log-mel or CPC features and the aforementioned dimensionality reduction methods, including not using dimensionality reduction at all, and compare the resulting SER performance among the alternative strategies.

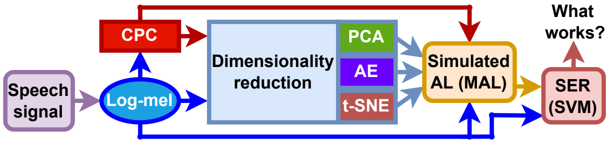

Figure 2: *A block diagram of the present experimental setup.*

2.1. Contrastive predictive coding

CPC [17] is an unsupervised method for extracting representations from data that encode underlying shared information between different parts of the input signal. This is achieved by predicting $k \in \{1, ..., K\}$ future latent representations of the input signal where typically $K > 1$ (e.g. $K = 12$ for speech in [17]). A CPC model consists of two separate models, a nonlinear encoder, g_{enc}, and an autoregressive model, g_{ar}. First, g_{enc} maps the input observations, $\boldsymbol{x}_t$, into latent representations $\boldsymbol{z}_t = g_{enc}(\boldsymbol{x}_t)$. Then, g_{ar} maps $\boldsymbol{z}_{\leq t}$ into a context latent representation $\boldsymbol{c}_t = g_{ar}(\boldsymbol{z}_{\leq t})$. Instead of directly modeling $\boldsymbol{x}_{t+k}$, CPC models a density ratio that aims to preserve the mutual information between $\boldsymbol{x}_{t+k}$ and $\boldsymbol{c}_t$. For this, a log-bilinear model $f_k(\boldsymbol{x}_{t+k}, \boldsymbol{c}_t) = exp\left(\boldsymbol{z}_{t+k}^T W_k \boldsymbol{c}_t\right)$ is used, where W_k are linear transformations. Both g_{enc} and g_{ar} are trained using the loss

$$L_{\text{CPC}} = -\frac{1}{K}\sum_{k=1}^{K} log\left[\frac{f_k(\boldsymbol{x}_{t+k}, \boldsymbol{c}_t)}{\sum\limits_{\boldsymbol{x}_j \in X} f_k(\boldsymbol{x}_j, \boldsymbol{c}_t)}\right], \qquad (1)$$

where $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ is a set of N random samples containing one positive sample and $N - 1$ negative samples. As shown in [17], minimizing the loss in Eq. 1 maximizes the mutual information between $\boldsymbol{x}_{t+k}$ and $\boldsymbol{c}_t$.

2.2. Medoid-based active learning

For AL, we use medoid-based active learning (MAL) [16], which is an AL method developed for scarce labeling budgets. The algorithm consists of three subsequent stages: 1) compute an affinity matrix containing the pairwise distances between each sample in a dataset, 2) perform k-medoids clustering using this affinity matrix, and 3) in a descending cluster size order, query for human annotations for the medoids.

In the first stage, the pairwise distances between each sample in a dataset are computed using a distance metric, d, and are stored to an affinity matrix, A. In the second stage, k-medoids clustering (see e.g. [27]) is performed for the data using A. First, one sample is randomly selected as a member of a set, S. Then, $k - 1$ additional samples are added to S one at a time using the *farthest-first traversal* algorithm [28], each added sample being farthest from the current set S. Here, the distance from a sample, $\boldsymbol{a}$, to the set S is defined as $d(\boldsymbol{a}, S) = \min_{\boldsymbol{b} \in S} d(\boldsymbol{a}, \boldsymbol{b})$. Next, the samples in S are used as initial medoids for a k-medoids clustering algorithm in order to assign each sample into one of the k clusters. Based on a previous study [29], k was set to be $\frac{N}{3}$, where N is the number of samples in a corpus.

In the third stage, the clusters are first sorted in a descending-size order, after which the cluster medoids are presented to human annotators for labeling. In the present experiments, the obtained labels were used in two different ways: i) using only the medoid labels as labeled data (referred to as “medoid labels”), or ii) propagating the medoid label to all elements in the cluster (referred to as “cluster labels”).

3. Experiments

3.1. Emotional Speech Corpora

We conducted our experiments using four SER corpora:

1) *The Berlin Emotional Speech Database* (EMO-DB) [30] is perhaps the most widely used SER corpus. It contains 535 spoken utterances in German from 10 actors in seven emotions: anger, boredom, disgust, fear, joy, neutral, and sadness.

2) *eNTERFACE* [31] contains 1,287 videos in English (42 test subjects, 14 nationalities), of which only the audio tracks were used in the present study. The corpus contains emotions in six categories: anger, disgust, fear, joy, sadness, and surprise.

3) *The Finnish Emotional Speech Corpus* (FESC) [32] consists of 450 spoken passages from nine Finnish professional actors portraying five emotions: neutral, sadness, joy, anger, and tenderness. Based on long silences defined by an energy threshold [26], the passages were further split into 4,254 utterances.

4) *The Ryerson Audio-Visual Database of Emotional Speech and Song* (RAVDESS) [33] is a multimodal database containing 7,356 recordings in English from 24 professional actors. Only the recordings including speech (1,440 utterances) were used in

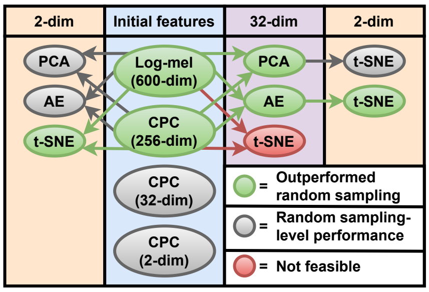

Figure 3: *A visualization of the feature, dimensionality reduction method, and data dimensionality combinations that were experimented with in the present study.*

the present experiments, including eight emotions: neutral, calm, happy, sad, angry, fearful, surprise, and disgust.

In order to harmonize the emotional labels of each SER corpus, the labels were mapped into the quarters of the valence-arousal plane following the mapping of [34], which has been popularly used in SER studies (e.g. [35–37]). The mapping also simplifies the SER classification task into two binary classification tasks: valence (positive/negative) and arousal (high/low).

3.2. CPC and AE model training

A separate CPC model was trained for each SER corpus as a data-driven feature extractor. Log-mel frames (40 mel filters, 30-ms Hann window, 10-ms shifts) were used as input features for the encoder g_{enc}. To get constant-length inputs, 5-second segments were extracted from the acoustic signal by zero-padding utterances shorter than 5 s and randomly selecting 5-second segments of utterances longer than 5 s. The encoder g_{enc} consisted of three fully-connected ELU [38] layers of 256 units, each followed by a dropout of 20%. The autoregressive model g_{ar} was a one-layer GRU [39] with a 256-dimensional hidden state. Prediction was carried up to 12 steps (120 ms) ahead.

Each corpus was randomly split into training and validation sets in a ratio of 80:20 utterances. The models were trained using the loss L_{CPC} in Eq. 1, a batch size of 8 utterances, and Adam [40] optimizer. An initial learning rate of 10^{-4} was used with a reduction factor of 0.7 based on the validation loss with a patience of 20 epochs. Early stopping with a patience of 100 based on validation loss was used to select the model with the lowest validation loss. For each sample in a minibatch, the rest of the minibatch samples acted as the negative samples.

Using a similar 80:20 split as above, an AE model was trained for dimensionality reduction individually for each corpus and for two input features (utterance-level log-mel and CPC features, see Sec. 3.3). The network consisted of six fully-connected ELU layers of 512 units each, except for the third layer (32 units) and the last layer (600 and 256 units for log-mel and CPC features, respectively). The 512-unit layers had a dropout of 10%. The AE models were trained using MSE loss, Adam optimizer, batch size of 1024, and a learning rate of 10^{-4}. Early stopping with a patience of 300 based on validation loss was used, and the best encoder (the first three network layers) was selected from the AE model with the lowest validation loss.

3.3. Features

For log-mel features of each utterance, seven functionals (the first four moments, min, max, and range) were applied to the time dimension of log-mel frames (Sec. 3.2), and the first four moments were applied to the first and second order delta features to get a 600-dimensional utterance-level feature vector. For CPC features, the mean over the time dimension of the encoder outputs $\boldsymbol{z}_t$ was computed to get 256-dimensional utterance features (ignoring potential zero padding). The context latent representations $\boldsymbol{c}_t$ were also tested but resulted in slightly worse results than using $\boldsymbol{z}_t$, and hence are not separately reported. In addition to the mean, we also tested computing other functionals from the CPC features. However, since the mean is the only moment that preserves the metric properties of the original CPC feature space, the inclusion of these additional functionals turned out to systematically worsen the SER performance of the CPC features, and hence are also not reported.

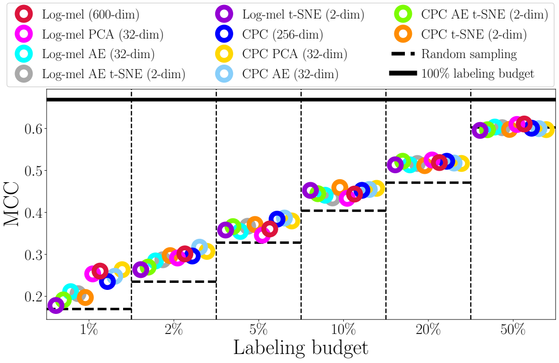

Figure 4: *The MCC performance scores for different experimented features using the MAL algorithm and varying labeling budget, and with random sampling as a baseline reference. Each reported number is the mean of 200 feature-specific experiments (valence and arousal classification tasks, four SER corpora, 5-fold CV, five different random initializations for each data split). The features are ordered based on mean performance, with the best-performing feature being the rightmost.*

Fig. 3 illustrates the tested combinations of features and dimensionality reduction methods that were included in the experiments. For both the 600-dimensional log-mel and 256-dimensional CPC features, 32- and 2-dimensional PCA and AE representations were computed. The AE representations were obtained using the encoder networks of Sec. 3.2. Additionally, 2-D t-SNE features were computed from the high-dimensional log-mel and CPC features as well as their 32-dimensional representations. The Scikit-learn [41] implementation of t-SNE was used with default values, with the exception of using PCA initialization for better overall stability. We also tested training CPC models directly into lower-dimensional feature spaces but with poor results. In order to save space, only the features that outperformed the random sampling baseline in AL-based classification are included in the present results. Computing 32-dimensional t-SNE features was not considered due to the poor scalability of the algorithm for higher than 2-dimensional projections.

3.4. AL simulation setup for SER

For each corpus and for each feature described in Sec. 3.3, SER experiments were carried out with the MAL algorithm using a simulated annotation procedure. In the simulation, samples selected for annotation by MAL were replaced by the ground-truth labels of the corpus as available based on earlier human annotations, and were then used to train a classifier for the valence and arousal classification tasks (see also [26]). The data was randomly split into training and test sets using 5-fold cross-validation (CV), after which MAL was applied to the training set. Labeling budgets of 1%, 2%, 5%, 10%, 20%, 50%, and 100% of the total training set samples were used in the experiments. Although MAL was originally intended for scarce labeling budgets,

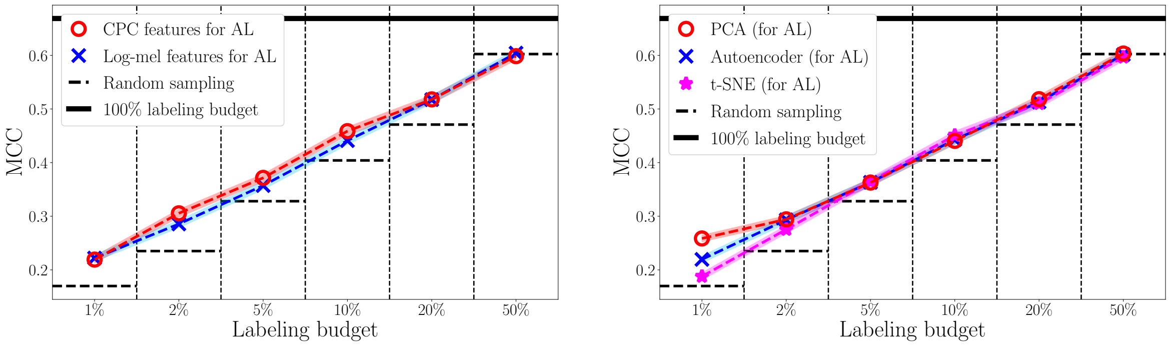

Figure 5: *Comparison of CPC and log-mel features (left) and PCA, AE, and t-SNE dimensionality reduction methods (right) in MAL-based SER simulations, averaged across all experimental conditions (standard error as shaded area). For the latter, the cases where only one dimensionality reduction method is being used are considered. Random sampling is shown as a baseline reference. Note that a SER classifier is always trained using 600-dim utterance-level log-mel features independently of the features and dimensionality used for AL.*

higher labeling budgets are also included in the analyses. Each cross-fold experiment was repeated five times to account for the minor variability in the results due to t-SNE and MAL random initialization. We use Matthew's correlation coefficient (MCC) [42] as our primary evaluation metric. As argued by e.g. [43], MCC can be considered as one of the most informative quality measures for binary classifiers since it requires good results from all four confusion matrix categories to output a high score.

After obtaining a set of labeled data from MAL, an SVM with an RBF kernel was trained with the labeled training data and tested on the test data. This was applied to both classification tasks and for both medoid and cluster labels. Since the primary focus of the present experiments was to find features that enhance the performance of clustering-based AL algorithms, the training and testing features for the SVMs were standardized for each experiment. For this, we selected the 600-dimensional utterance-level log-mel feature statistics (z-score normalized at the corpus level), as they were found generally well-performing in valence and arousal classification tasks in [26]. SVM hyperparameters (box constraint and kernel scale parameter) were optimized separately for each corpus and classification task based on a grid search using 5-fold CV on all labeled data for the given corpus. As a baseline reference, random sampling results with corresponding labeling budgets are also reported, with experiments conducted in a similar manner as with MAL.

For the sake of brevity, the results with MAL cluster labels are omitted since they provided an advantage over medoid labels only for 1% labeling budget. For the same reason, only the results from the best-performing distance metrics for MAL are reported, corresponding to the Euclidean distance for 2-D features, and the cosine distance for higher-dimensional features.

4. Results

Fig. 4 presents the results of the experimental setup, averaged over the corpora and both classification tasks (valence and arousal). As the labeling budget increases, the differences between the tested features become smaller. Also, with a 50% labeling budget MAL does not provide a benefit over random sampling. For labeling budgets of 1%–20%, the 32-dimensional "CPC PCA" features had the highest average performance (approx. 0.38 MCC). For the four SER corpora, a labeling budget of 1% corresponds to approx. 15 samples on average.

Fig. 5 (left) shows the performance comparison between CPC and log-mel features. For labeling budgets of 2%–10%, CPC features provided statistically significant improvements over log-mel features (paired t-test, $p < 0.05$, $df = 999$, $t = 3.06$–6.12 across the tests), showcasing better overall clusterability and/or more efficient feature space exploration for the CPC features in the SER task. Fig. 5 (right) shows the comparison between dimensionality reduction methods, demonstrating a similar performance for all three methods from a 5% labeling budget onward. With a 2% labeling budget, PCA and AE obtain a similar performance and outperform t-SNE, and with a 1% labeling budget, PCA outperforms the AE method which, in turn, outperforms t-SNE (paired t-test, $p < 0.05$, $df = 399$, $t = 1.17$–3.92 across the tests). Comparing different dimensionalities in Fig. 4, we can observe that heavily reducing the dimensionality of the 600/256-dimensional features does not provide a major drop in average performance, with "CPC PCA" and "CPC AE" even outperforming the 256-dimensional CPC features. The 600/256-dimensional and 32-dimensional features obtained similar performance on average, and the 2-D t-SNE features also obtained similar performance for labeling budgets of 5% and larger. Overall, these results indicate that for clustering-based AL approaches, both the global (600/256/32-dimensional features) and local (t-SNE features) properties of the data can be used to provide an advantage over random sampling. However, it seems that t-SNE works best if the labeling budget is not below 5%, possibly since t-SNE distorts the global distance metrics.

5. Conclusions

In this study, we combined CPC and various dimensionality reduction methods in search of functioning practices for clustering-based AL. Our experiments revealed that SSRL can be utilized to improve clustering-based AL performance over traditional signal features. In addition, we found that both the local and global topologies of feature spaces can be successfully used for AL. Furthermore, we observed that compressing the dimensionality of high-dimensional features does not provide a major drop in AL performance. We also found that 2-D t-SNE features achieved similar AL performance as higher-dimensional features when the labeling budget is not very low. This finding could be utilized to reduce the computational complexity of AL algorithms, and to combine AL and data visualization to create interactive AL algorithms involving data exploration and visualization, in a similar manner as in the annotation platform of [15].

6. Acknowledgements

This research was funded by Academy of Finland grants no. 314602, 335872, and 343498.

7. References

[1] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng, "Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks," in *Proc. EMNLP*, 2008, pp. 254–263.

[2] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, "A Survey of Human-in-the-loop for Machine Learning," *arXiv preprint arXiv: 2108.00941*, 2021.

[3] B. Settles, "From Theories to Queries: Active Learning in Practice," in *Proc. PMLR*, vol. 16, 2011, pp. 1–18.

[4] Z. Zhang and B. Schuller, "Active Learning by Sparse Instance Tracking and Classifier Confidence in Acoustic Emotion Recognition," in *Proc. INTERSPEECH*, 2012, pp. 362–365.

[5] W. Han, H. Li, H. Ruan, L. Ma, J. Sun, and B. W. Schuller, "Active learning for dimensional speech emotion recognition," in *Proc. INTERSPEECH*, 2013, pp. 2841–2845.

[6] T. Drugman, J. Pylkkönen, and R. Kneser, "Active and Semi-Supervised Learning in ASR: Benefits on the Acoustic and Language Models," in *Proc. INTERSPEECH*, 2016, pp. 2318–2322.

[7] W. Han, E. Coutinho, H. Ruan, H. Li, B. Schuller, X. Yu, and X. Zhu, "Semi-Supervised Active Learning for Sound Classification in Hybrid Learning Environments," *PLOS ONE*, vol. 11, no. 9, pp. 1–23, 2016.

[8] C. Mayer and R. Timofte, "Adversarial Sampling for Active Learning," in *Proc. WACV*, 2020, pp. 3071–3079.

[9] J. Attenberg and F. Provost, "Inactive learning? difficulties employing active learning in practice," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 36–41, 2010.

[10] H. T. Nguyen and A. Smeulders, "Active Learning Using Pre-clustering," in *Proc. ICML*, 2004.

[11] Z. Bodó, Z. Minier, and L. Csató, "Active Learning with Clustering," in *Proc. PMLR*, vol. 16, 2011, pp. 127–139.

[12] K. Malhotra, S. Bansal, and S. Ganapathy, "Active Learning Methods for Low Resource End-To-End Speech Recognition," in *Proc. INTERSPEECH*, 2019, pp. 2215–2219.

[13] H. Peng and N. G. Pavlidis, "Subspace Clustering with Active Learning," in *Proc. IEEE BigData*, 2019, pp. 135–144.

[14] H. Liu, J. Lafferty, and L. Wasserman, "Sparse Nonparametric Density Estimation in High Dimensions Using the Rodeo," in *Proc. PMLR*, vol. 2, 2007, pp. 283–290.

[15] P. Fallgren, Z. Malisz, and J. Edlund, "How to Annotate 100 Hours in 45 Minutes," in *Proc. INTERSPEECH*, 2019, pp. 341–345.

[16] S. Zhao, T. Heittola, and T. Virtanen, "Active learning for sound event classification by clustering unlabeled data," in *Proc. ICASSP*, 2017, pp. 751–755.

[17] A. van den Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv preprint arXiv: 1807.03748*, 2018.

[18] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. de Chaumont Quitry, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Haviv, "Towards Learning a Universal Non-Semantic Representation of Speech," in *Proc. INTERSPEECH*, 2020, pp. 140–144.

[19] A. Baevski, W. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language," *arXiv preprint arXiv: 2202.03555*, 2022.

[20] O. J. Henaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, "Data-Efficient Image Recognition with Contrastive Predictive Coding," in *Proc. ICLR*, 2020.

[21] B. van Niekerk, L. Nortje, M. Baas, and H. Kamper, "Analyzing Speaker Information in Self-Supervised Models to Improve Zero-Resource Speech Processing," in *Proc. INTERSPEECH*, 2021, pp. 1554–1558.

[22] S. Takahashi, S. Sakti, and S. Nakamura, "Unsupervised Neural-Based Graph Clustering for Variable-Length Speech Representation Discovery of Zero-Resource Languages," in *Proc. INTERSPEECH*, 2021, pp. 1559–1563.

[23] T. Maekaku, X. Chang, Y. Fujita, L.-W. Chen, S. Watanabe, and A. Rudnicky, "Speech Representation Learning Combining Conformer CPC with Deep Cluster for the ZeroSpeech Challenge 2021," in *Proc. INTERSPEECH*, 2021, pp. 1564–1568.

[24] S. Ma, Z. Zeng, D. J. McDuff, and Y. Song, "Active Contrastive Learning of Audio-Visual Video Representations," in *Proc. ICLR*, 2021.

[25] L. van der Maaten and G. Hinton, "Visualizing High-Dimensional Data Using t-SNE," *JMLR*, vol. 9, pp. 2579–2605, 2008.

[26] E. Vaaras, "Automatic Emotional Speech Analysis from Daylong Child-Centered Recordings from a Neonatal Intensive Care Unit," Master's thesis, Tampere University, 2021.

[27] H.-S. Park and C.-H. Jun, "A Simple and Fast Algorithm for K-Medoids Clustering," *Expert Systems with Applications*, vol. 36, no. 2, p. 3336–3341, 2009.

[28] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.

[29] E. Vaaras, S. Ahlqvist-Björkroth, K. Drossos, and O. Räsänen, "Automatic Analysis of the Emotional Content of Speech in Daylong Child-Centered Recordings from a Neonatal Intensive Care Unit," in *Proc. INTERSPEECH*, 2021, pp. 3380–3384.

[30] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *9th European Conference on Speech Communication and Technology*, vol. 5, 2005, pp. 1517–1520.

[31] O. Martin, I. Kotsia, B. Macq, and I. Pitas, "The eNTERFACE' 05 Audio-Visual Emotion Database," in *Proc. ICDEW*, 2006, pp. 1–8.

[32] M. Airas and P. Alku, "Emotions in Vowel Segments of Continuous Speech: Analysis of the Glottal Flow Using the Normalised Amplitude Quotient," *Phonetica*, vol. 63, pp. 26–46, 2006.

[33] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PLoS ONE*, vol. 13, no. 5, pp. 1–35, 2018.

[34] B. Schuller, B. Vlasenko, F. Eyben, M. Wöllmer, A. Stuhlsatz, A. Wendemuth, and G. Rigoll, "Cross-Corpus Acoustic Emotion Recognition: Variances and Strategies," *IEEE Transactions on Affective Computing*, vol. 1, no. 2, pp. 119–131, 2010.

[35] B. Schuller, Z. Zhang, F. Weninger, and G. Rigoll, "Using Multiple Databases for Training in Emotion Recognition: To Unite or to Vote?" in *Proc. INTERSPEECH*, 2011, pp. 1553–1556.

[36] S. Latif, J. Qadir, and M. Bilal, "Unsupervised Adversarial Domain Adaptation for Cross-Lingual Speech Emotion Recognition," in *Proc. ACII*, 2019, pp. 732–737.

[37] Q. Mao, G. Xu, W. Xue, J. Gou, and Y. Zhan, "Learning emotion-discriminative and domain-invariant features for domain adaptation in speech emotion recognition," *Speech Communication*, vol. 93, pp. 1 – 10, 2017.

[38] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *Proc. ICLR*, 2016.

[39] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.

[40] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, 2015.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *JMLR*, vol. 12, pp. 2825–2830, 2011.

[42] B. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[43] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 6, 2020.