2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Self-Supervised Feature Learning of 1D Convolutional Neural Networks with Contrastive Loss for Eating Detection Using an In-Ear Microphone

Vasileios Papapanagiotou[1] and Christos Diou[2] and Anastasios Delopoulos[1]

Abstract— **The importance of automated and objective monitoring of dietary behavior is becoming increasingly accepted. The advancements in sensor technology along with recent achievements in machine-learning–based signal-processing algorithms have enabled the development of dietary monitoring solutions that yield highly accurate results. A common bottleneck for developing and training machine learning algorithms is obtaining labeled data for training supervised algorithms, and in particular ground truth annotations. Manual ground truth annotation is laborious, cumbersome, can sometimes introduce errors, and is sometimes impossible in free-living data collection. As a result, there is a need to decrease the labeled data required for training. Additionally, unlabeled data, gathered in-the-wild from existing wearables (such as Bluetooth earbuds) can be used to train and fine-tune eating-detection models. In this work, we focus on training a feature extractor for audio signals captured by an in-ear microphone for the task of eating detection in a self-supervised way. We base our approach on the SimCLR method for image classification, proposed by Chen et al. from the domain of computer vision. Results are promising as our self-supervised method achieves similar results to supervised training alternatives, and its overall effectiveness is comparable to current state-of-the-art methods. Code is available at `https://github.com/mug-auth/ssl-chewing`.**

I. Introduction

While obesity and eating-related diseases are affecting ever-growing portions of the population, awareness of our eating habits and behavior can play a very important role in both prevention and treatment. The under-reporting of eating in questionnaire-based studies is a well known fact [1]; as a result, technology-assisted monitoring using wearable sensors is gaining more and more attention.

Meaningful information is usually extracted from signals captured by wearable sensors by means of signal-processing algorithms; these algorithms are often based on supervised machine learning and thus require labeled training data to achieve satisfactory effectiveness. This is more important in deep-learning—based approaches where larger volumes of (annotated) data are required. Creating such large datasets, however, is challenging. Generating ground truth annotations requires a lot of manual work, where experts or specially trained personnel process each part of the dataset in detail in order to derive the annotations. Besides being laborious and time-consuming, this process is sometimes prone to errors (which is sometimes reduced by using multiple annotators for the same data) and can often introduce limitations in data collection. For example, in the case of video-based annotation, subjects are limited to the room/area covered by the cameras and data collection cannot take place in free-living conditions.

One way to overcome this is semi-supervised or unsupervised training methods. Such methods have already been used with great success in fields such as speech processing [2] as well as in applications with wearable sensors [3], [4], [5]. Self-supervised methods for image classification have received a lot research attention recently [6]. These methods use multiple augmentations on unlabeled images in order to learn effective image representations.

In this work, we adapt ideas from self-supervised image classification to 1D convolutional neural networks (CNN) with the goal to train a chewing-detection model on audio from an in-ear microphone, and focus on training the feature extractor using only unlabeled data. The model is a deep neural network (DNN) that includes convolutional and max-pooling layers for feature extraction and fully-connected (FC) layers for classification. We follow the approach of [7], [8] and train the convolutional and max-pooling layers in a self-supervised way, and then use them as a fixed feature extraction mechanism to train the FC classification layers. We evaluate on a large and challenging dataset and compare with our previous work that uses only supervised training, as well as other algorithms from the literature, and obtain highly encouraging results.

II. Network training

To study if we can successfully train a self-supervised feature extractor we use the architecture of our previous work on chewing detection [9]. In particular, we focus on the architecture of 5 s input window due to its effectiveness in the supervised learning setting. The network is split in two: the feature extraction sub-network $\mathbf{f}$, and the classification sub-network, $\mathbf{h}$. Sub-network $\mathbf{f}$ maps an audio window to a feature vector, i.e. $\mathbf{f} : \mathbb{R}^{10,000} \rightarrow \mathbb{R}^{512}$ (where $10,000\,\text{samples} = 5\,\text{s} \cdot 2\,\text{kHz}$) and consists of five pairs of convolutional layers followed by max-pooling layers. The convolutional layers have progressively more filters (8, 16, 32, 64, and 64 respectively) and constant length (16 samples for all layers except for the last one that has 39); the activation function is ReLU. The max-pooling ratio is always $2:1$. Sub-network $\mathbf{g}$ classifies a feature vector to a binary chewing vs. non-chewing decision, i.e. $\mathbf{h} : \mathbb{R}^{512} \rightarrow [0,1]$ and consists of two FC layers of 200 neurons with ReLU

[1]Vasileios Papapanagiotou and Anastasios Delopoulos are with the Multimedia Understanding Group, Dpt. of Electrical and Computer Engineering, Faculty of Engineering, Aristotle University of Thessaloniki, Greece `vassilis@mug.ee.auth.gr`, `adelo@eng.auth.gr`

[2]Christos Diou is with Department of Informatics and Telematics, Harokopio University of Athens, Greece `cdiou@hua.gr`

activation followed by a layer of a single neuron with sigmoid activation.

In [9], the entire network[1] $\mathbf{h} \circ \mathbf{f}$ is trained together on labeled data. In this work, our goal is to train $\mathbf{f}$ in a self-supervised way, and only use labeled data for training $\mathbf{h}$.

A. Self-Supervised feature learning

To train $\mathbf{f}$ we follow the paradigm of [7] where each training sample $\mathbf{x}$ is augmented with two different augmentations, $\mathcal{T}_1$ and $\mathcal{T}_2$, in parallel, yielding samples $\mathbf{x}^1$ and $\mathbf{x}^2$ respectively. Given an initial training batch of n samples, $\{\mathbf{x}_i : i = 1, \ldots, n\}$ we create a new batch of double the original size as $\{\mathbf{x}_i^1 : i = 1, \ldots, n\} \cup \{\mathbf{x}_i^2 : i = 1, \ldots, n\}$ using the two augmentations. Given a similarity metric, the network is then trained to maximize the similarity between all pairs that are derived from the same original sample, i.e. $\mathbf{x}_i^1, \mathbf{x}_i^2$, and to minimize the similarity between $\mathbf{x}_i^{k_1}, \mathbf{x}_j^{k_2}$ for $i \neq j$, $k_1 = 1, 2$, and $k_2 = 1, 2$.

Following [7], we use the cosine similarity with temperature [10] as our similarity metric:

$$\operatorname{sim}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|} \tau^{-1}$$

where $\langle \bullet, \bullet \rangle$ is the inner product, $\|\bullet\|$ is the Euclidean norm, and τ is the temperature parameter. Higher τ values leads to "sharper" softmax at the network's output, while lower τ values lead to "smoother" output which can yield more effective representations.

For the contrastive loss function we use the normalized temperature-scaled cross-entropy loss [11] since it has been used on similar applications, mainly in the domain of 2D signals (such as images). Given a training batch of $2n$ samples, i.e. $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$ for $i = 1, \ldots, n$, the loss for the i-th positive pair, i.e. $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$, is defined as:

$$l_a\left(\mathbf{x}_i^1, \mathbf{x}_i^2\right) = -\ln \frac{e^{\operatorname{sim}\left(\mathbf{x}_i^1, \mathbf{x}_i^2\right)}}{\sum_{k=1}^{n} \left(\mathbb{I}_{[i \neq k]} e^{\operatorname{sim}\left(\mathbf{x}_i^1, \mathbf{x}_k^1\right)} + e^{\operatorname{sim}\left(\mathbf{x}_i^1, \mathbf{x}_k^2\right)} \right)}$$

where $\mathbb{I}_{[i \neq k]} \in \{0, 1\}$ is a boolean indicator that is equal to 1 if and only if $i \neq k$. The indicator is necessary because the similarity between the a vector and itself is always the same, i.e. $\operatorname{sim}(\mathbf{x}, \mathbf{x}) = \tau^{-1}$. This, in turn, renders $l_a\left(\mathbf{x}_i^1, \mathbf{x}_i^2\right)$ asymmetric, and the final loss between samples $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$ is simply the average:

$$l(i) = \frac{1}{2}\left(l_a\left(\mathbf{x}_i^1, \mathbf{x}_i^2\right) + l_a\left(\mathbf{x}_i^2, \mathbf{x}_i^1\right) \right)$$

While it is possible to compute the similarity metric directly on the output of $\mathbf{f}$, it is usually better to use a projection head, $\mathbf{g}$, that is applied after $\mathbf{f}$ [7]. We experiment with two projection heads: (a) a linear, $\mathbf{g}^{\mathrm{L}}$, that consists of a single FC layer of 128 neurons with linear activation, and (b) a non-linear, $\mathbf{g}^{\mathrm{NL}}$, that consists of 2 FC layers of 512 neurons with ReLU activations followed by a FC layer of 128 neurons with linear activation. Thus, the resulting network is $\mathbf{g} \circ \mathbf{f}$.

[1]Operator $\circ$ denotes function composition, i.e. $f \circ g(x) = f(g(x))$

To train $\mathbf{g} \circ \mathbf{f}$ we use the LARS optimizer which has been proposed in [12]. We set the batch size to 256 (thus obtaining 512 samples after the dual augmentation process) and train for 100 epochs, based on some initial experimentation with the dataset. We apply a warm-up schedule on the learning rate for 10% of the total epochs (i.e. for 5 epochs) and reach a maximum learning rate of 0.3, after which we apply cosine decay to the learning rate [13].

B. Augmentations

An important part of this method is selecting augmentations that can help the training process to learn features that can be effectively used in the final classification task. Time-stretching (speeding up or slowing down) of audio has been used both in the speech recognition domain [14] as well as in more general applications (e.g. environmental sounds classification [15]) with relatively small changes, i.e. 80% to 120%. Other augmentations include pitch shifting, dynamic range compression, and adding background noise [15].

Based on the above as well as the nature of audio chewing signals, we focus on two augmentations: global amplification level and background noise. In particular, global amplification is $\mathcal{T}_1(\mathbf{x}) = \alpha \cdot \mathbf{x}$ where α is the global amplification level and is drawn from a uniform distribution in the range $[0.5, 2.0]$ (a different α is drawn for each $\mathbf{x}$). This choice for this augmentation is based on our experience with in-ear microphone signals [16], where placement, ear and sensor shape compatibility, and even movement can change the overall amplification level of the captured audio.

The second augmentation is the addition of noise and implemented as: $\mathcal{T}_2(\mathbf{x}) = \mathbf{x} + \mathbf{v}$ where $\mathbf{v}$ is a vector of IIR "noise" samples, drawn from uniform distributions in the range of $[-0.005, 0.005]$. Value 0.005 has been chosen as it roughly equal to 0.1 of the average standard deviation of audio signal across our entire dataset, yielding 20 dB SNR. Adding noise to the audio signal simulates noisy environments (such as noisy city streets, restaurants, etc) and helps our feature extractor learn to "ignore" its influence.

C. Supervised classifier training

Given the trained feature extractor $\mathbf{f}$ we can now train a chewing detection classifier based on label data. In this stage, the projection head $\mathbf{g}$ can be discarded and thus the final network is $\mathbf{h} \circ \mathbf{f}$, where only the weights of $\mathbf{h}$ are trained. It is possible, however, to retain a part of the projection head in the final model [8]. We do this for the case of the non-linear projection head, $\mathbf{g}^{\mathrm{NL}}$: let $\mathbf{g}^{\mathrm{NL}} = \mathbf{g}_2^{\mathrm{NL}} \circ \mathbf{g}_1^{\mathrm{NL}}$ where $\mathbf{g}_1^{\mathrm{NL}}$ corresponds to the first layer of $\mathbf{g}^{\mathrm{NL}}$ and $\mathbf{g}_2^{\mathrm{NL}}$ to the remaining (second and third) layers of $\mathbf{g}^{\mathrm{NL}}$; thus, the final network is $\mathbf{h} \circ \mathbf{g}_1^{\mathrm{NL}} \circ \mathbf{f}$ (again, only weights of $\mathbf{h}$ are trained here).

We use the ADAM optimizer [17] with a learning rate of 10^{-3} and minimize binary cross-entropy based on ground truth values:

$$H(\hat{y}_i; y_i) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

where $y_i \in \{0, 1\}$ is the ground truth value for the i-th sample and $\hat{y} \in [0, 1]$ is the output of $\mathbf{h}$.

D. Post-processing of predicted labels

The predicted labels indicate chewing vs. non-chewing; thus, chewing "pulses" correspond to individual chews. Similarly to our previous works [18], [9], we aggregate chews to chewing bouts and then chewing bouts to meals. In short, (a) chewing bouts are obtained by merging chews that are no more than 2 s apart, (b) chewing bouts of less than 5 s are discarded, (c) meals are obtained by merging chewing bouts that are no more than 60 s apart, (d) meals for which the ratio of "duration of bouts" over "duration of meal" is less than 25% are discarded.

III. Dataset

The dataset we use has been collected in the Wageningen University in 2015 during a pilot study of the EU SPLENDID project [19]. Recordings from 14 individuals (approximately 60 h) are available. Each subject had two meals in the university premises and was free to leave the university, engage in physical activities, and have as many other meals and snacks wished for the rest of the recording time. This dataset has also been used in [18] and [9].

The sensor is a prototype in-ear microphone sensor consisting of Knowles FG-23329-D65 microphone housed in a commercial ear bud. Audio was originally captured at 48 kHz but we have down-sampled it at 2 kHz (as in [18]) to reduce the computational burden; we have also applied a high-pass Butterworth filter with a cut-off frequency of 20 Hz to remove very low spectrum content and the effect of DC drifting that was present in the chewing-sensor prototype.

IV. Evaluation

We split our dataset of 14 subjects into two parts: a "development" set with 10 subjects (selected randomly), S_1, and a "final evaluation" set with the remaining 4 subjects, S_2. Note that S_1 and S_2 are disjoint. In the first part of the evaluation, we explore training hyper-parameters and architecture choices on S_1. In the second part, we apply what we learned and, after training our models on S_1, we evaluate them on S_2.

In the first part of evaluation, our goal is to understand the effect of the temperature and the projection head on classification accuracy. In particular, we first train $\mathbf{g}\circ\mathbf{f}$ on the entire S_1 (all 10 subjects) using self-supervised training (as described in Section II-A). We then train $\mathbf{h}$ on the same 10 subjects but in a supervised way (as described in Section II-C) in typical leave-one-subject-out (LOSO) fashion. During each LOSO iteration, data from 9 subjects are available for training. We select a small part of the 9 subjects (specifically 2 subjects) as a validation set, and train on the remaining (7 subjects). We train for 100 epochs and compute the loss over the validation subjects' data after each epoch; we select the model that minimizes the validation loss at the end of each epoch. We use a batch size of 64. Note that in these experiments self-supervised feature learning takes place in all 10 subjects of S_1 (for computational reasons) and are only used here to obtain an assessment of the effect of temperature and not for evaluating our algorithm. Results of evaluation on the held-out dataset (S_2) are presented in Table IV.

TABLE I: Results for the $\mathbf{h}\circ\mathbf{f}^{\mathrm{L}}$ network on S_1.

τ	**prec.**	**rec.**	**F1-score**	**acc.**	**w. acc.**
0.1	0.82	0.64	0.72	0.94	0.84
0.5	0.75	0.76	0.76	0.94	0.86
1	0.82	0.55	0.66	0.93	0.81
5	0.82	0.62	0.70	0.93	0.83
10	0.80	0.66	0.72	0.94	0.84
50	0.77	0.12	0.21	0.88	0.57
100	0.85	0.63	0.72	0.94	0.84

TABLE II: Results for the $\mathbf{h}\circ\mathbf{f}^{\mathrm{NL}}$ network on S_1.

τ	**prec.**	**rec.**	**F1-score**	**acc.**	**w. acc.**
0.1	0.85	0.67	0.75	0.94	0.86
0.5	0.85	0.63	0.72	0.94	0.85
1	0.85	0.63	0.73	0.94	0.85
5	0.83	0.50	0.62	0.92	0.79
10	0.84	0.58	0.69	0.93	0.82
50	0.76	0.59	0.66	0.92	0.81
100	0.86	0.31	0.45	0.91	0.70

We examine different values of temperature τ; results are presented in Tables I - III. Table I shows results for the $\mathbf{h}\circ\mathbf{f}^{\mathrm{L}}$, where $\mathbf{f}^{\mathrm{L}}$ is $\mathbf{f}$ trained with the linear projection head $\mathbf{g}^{\mathrm{L}}$. Based on F1-score, the best results are obtained for $\tau = 0.5$ while most other values of τ yield F1-score higher then 0.7.

Table II shows similar results; in this case the network is $\mathbf{h}\circ\mathbf{f}^{\mathrm{NL}}$, where $\mathbf{f}^{\mathrm{NL}}$ is $\mathbf{f}$ trained with the non-linear projection head $\mathbf{g}^{\mathrm{NL}}$. Highest F1-score is obtained for $\tau = 0.1$; in general, training seems to benefit more from smaller temperature values. Extremely large temperature values (e.g. 100) seem to not be beneficial (this is also observed in [8]).

Finally, Table III shows similar results for $\mathbf{h}\circ\mathbf{g}_1^{\mathrm{NL}}\circ\mathbf{f}^{\mathrm{NL}}$; this network is the same as before (i.e. $\mathbf{h}\circ\mathbf{f}^{\mathrm{NL}}$) but the first layer of the projection head, $\mathbf{g}_1^{\mathrm{NL}}$, is retained. Here, smaller temperatures seems to benefit the overall effectiveness more, with $\tau = 1$ yielding the highest F1-score, and high temperatures ($\tau \geq 50$) degrade the effectiveness completely.

In the second part of the evaluation, we use evaluate on the 4 subjects of S_2 with models trained on S_1. In particular, we select the best network of the three different approaches (Tables I - III) based on F1-score. We train $\mathbf{f}$ and $\mathbf{g}$ on S_1 in an self-supervised way and then train $\mathbf{h}$ again on S_1 in a supervised way. The three trained models are then evaluated on S_2 and the results are shown in Table IV. To compare, we also train a fourth model by training the entire network

TABLE III: Results for the $\mathbf{h}\circ\mathbf{g}_1^{\mathrm{NL}}\circ\mathbf{f}^{\mathrm{NL}}$ network on S_1.

τ	**prec.**	**rec.**	**F1-score**	**acc.**	**w. acc.**
0.1	0.75	0.72	0.73	0.93	0.85
0.5	0.74	0.59	0.66	0.92	0.81
1	0.87	0.64	0.74	0.94	0.85
5	0.80	0.57	0.67	0.93	0.81
10	0.88	0.02	0.03	0.88	0.48
50	0.73	0.32	0.44	0.90	0.69
100	0.75	0.24	0.36	0.89	0.65

TABLE IV: Final evaluation of effectiveness on S_2, when training on S_1 between self-supervised feature learning and supervised classifier training (lines 1-3), and supervised training of the entire network (line 4).

model	τ	prec.	rec.	F1-score	acc.	w. acc.
$\mathbf{h} \circ \mathbf{f}^{\mathrm{L}}$	0.5	0.83	0.79	0.81	0.95	0.89
$\mathbf{h} \circ \mathbf{f}^{\mathrm{NL}}$	0.1	0.94	0.79	0.86	0.97	0.92
$\mathbf{h} \circ \mathbf{g}_1^{\mathrm{NL}} \circ \mathbf{f}^{\mathrm{NL}}$	1	0.84	0.78	0.81	0.95	0.89
$\mathbf{h} \circ \mathbf{f}$	-	0.93	0.61	0.74	0.95	0.84

TABLE V: Results for comparison with base-line as presented in [9]: three algorithms of [20] and the 5-sec arch. CNN chewing detector of [9] (supervised LOSO on all 14 subjects, batch size of 16, Adam optimizer with learning rate of 0.0016, and $5 \cdot 10^5$ epochs).

approach	prec.	rec.	F1-score	acc.	w. acc.
MSEA [20]	0.29	0.80	0.42	0.72	0.76
MESA [20]	0.30	0.81	0.44	0.74	0.77
LPFSA [20]	0.29	0.81	0.43	0.72	0.76
5-sec arch. [9]	0.89	0.93	0.91	0.98	0.96

(i.e. $\mathbf{h} \circ \mathbf{f}$) on S_1 in a supervised way (similar to how models are trained in [9]). Results are shown in the fourth line of Table IV.

Results are particularly encouraging as the three networks with the self-supervised trained feature-extraction layers (lines 1-3) not only achieve similar effectiveness with the completely supervised-trained network (line 4) but also improve over it. The non-linear projection head (without retainment of the first layer, i.e. the $\mathbf{h} \circ \mathbf{f}^{\mathrm{NL}}$) achieves the highest F1-score and weighed accuracy among all the three self-supervised models and the supervised model.

As a final comparison, we repeat the results of three different algorithms of [20] and for the 5-sec architecture of [9], as presented in [9]. It is important to note that these results are averages across all 14 subjects of the dataset, so they are not directly comparable with all previous results. However, they can give an estimate about the overall effectiveness of our approach. Results indicate that self-supervised training exceeds the effectiveness of several of the methods proposed in the bibliography. Only the 5-sec architecture achieves better results (F1-score of 0.91 versus 0.84 of $\mathbf{h} \circ \mathbf{g}_1^{\mathrm{NL}} \circ \mathbf{f}^{\mathrm{NL}}$).

V. Conclusions

In this work, we have presented an approach for training the feature extraction layers of an audio-based chewing-detection neural network in an self-supervised way. Self-supervised training seems to lead to highly effective models, while at the same time reducing the labor needed for manual annotation as well as takes advantage of large amounts of unlabeled data for representation learning. Our experiments show very promising results, as self-supervised training achieves similar, and sometimes better, effectiveness compared to a similar fully-supervised approach. Additionally, best results (F1-score of 0.86) are comparable to fully-supervised methods on the same dataset (F1-score of 0.91 of [9]). Future work includes further studying the effect of additional augmentations for the self-supervised training part, and examining how much effectiveness is affected when there are less data available for the supervised training part. Additionally, the effectiveness of the trained self-supervised network can be evaluated on other problems such as individual chew detection or food type recognition.

Acknowledgments

The work leading to these results has received funding from the EU Commission under Grant Agreement No. 965231, the REBECCA project (H2020).

References

[1] M. Jessri, W. Y. Lou, and M. R. L'Abbé, "Evaluation of different methods to handle misreporting in obesity research: evidence from the canadian national nutrition survey," *British Journal of Nutrition*, vol. 115, no. 1, p. 147–159, 2016.

[2] G. Trigeorgis *et al.*, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *2016 IEEE ICASSP*, 2016, pp. 5200–5204.

[3] S. Nemes and M. Antal, "Feature learning for accelerometer based gait recognition," *arXiv:2007.15958 [cs]*, 2020.

[4] A. Papadopoulos *et al.*, "Unobtrusive detection of parkinson's disease from multi-modal and in-the-wild sensor data using deep learning techniques," *Scientific Reports*, vol. 10, no. 1, p. 21370, 12 2020.

[5] K. Kyritsis *et al.*, "Assessment of real life eating difficulties in parkinson's disease patients by measuring plate to mouth movement elongation with inertial sensors," *Scientific Reports*, vol. 11, no. 1, p. 1632, 1 2021.

[6] Q. Xie *et al.*, "Unsupervised data augmentation for consistency training," *arXiv:1904.12848 [cs]*, 2020.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[8] T. Chen *et al.*, "Big self-supervised models are strong semi-supervised learners," *arXiv preprint arXiv:2006.10029*, 2020.

[9] V. Papapanagiotou, C. Diou, and A. Delopoulos, "Chewing detection from an in-ear microphone using convolutional neural networks," in *IEEE EMBC 2017)*, 2017, pp. 1258–1261.

[10] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531 [stat.ML]*, 2015.

[11] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748 [cs]*, 2019.

[12] Y. You, I. Gitman, and B. Ginsburg, "Large batch training of convolutional networks," *arXiv:1708.03888 [cs]*, 2017.

[13] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, "A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation," *arXiv:1810.13243 [cs]*, 2018.

[14] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH-2015*, 2015, pp. 3586–3589.

[15] 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.

[16] J. van den Boer *et al.*, "The splendid eating detection sensor: Development and feasibility study," *JMIR Mhealth Uhealth*, vol. 6, no. 9, p. e170, Sep 2018.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980 [cs]*, 2017.

[18] V. Papapanagiotou *et al.*, "A novel chewing detection system based on ppg, audio, and accelerometry," *IEEE JBHI*, vol. 21, no. 3, pp. 607–618, 2017.

[19] C. Maramis *et al.*, "Preventing obesity and eating disorders through behavioural modifications: The splendid vision," in *MOBIHEALTH 2014*, 2014, pp. 7–10.

[20] S. Päßler and W.-J. Fischer, "Evaluation of algorithms for chew event detection," in *7th International Conference on Body Area Networks*, ser. BodyNets '12, 2012, p. 20–26.