Deformable CNN and Imbalance-Aware Feature Learning for Singing Technique Classification

Yuya Yamamoto[1]*, Juhan Nam*[2]*, Hiroko Terasawa*[1]

[1]Doctoral Program in Informatics, University of Tsukuba, Japan
[2]Graduate School of Culture Technology, KAIST, South Korea

s2130507@s.tsukuba.ac.jp, juhan.nam@kaist.ac.kr, terasawa@slis.tsukuba.ac.jp

Abstract

Singing techniques are used for expressive vocal performances by employing temporal fluctuations of the timbre, the pitch, and other components of the voice. Their classification is a challenging task, because of mainly two factors: 1) the fluctuations in singing techniques have a wide variety and are affected by many factors and 2) existing datasets are imbalanced. To deal with these problems, we developed a novel audio feature learning method based on deformable convolution with decoupled training of the feature extractor and the classifier using a class-weighted loss function. The experimental results show the following: 1) the deformable convolution improves the classification results, particularly when it is applied to the last two convolutional layers, and 2) both re-training the classifier and weighting the cross-entropy loss function by a smoothed inverse frequency enhance the classification performance.

1. Introduction

Professional singers express their characteristics and emotions by various singing techniques such as vibrato and breathy voice effects. At the signal level, singing techniques are observed as time–frequency textures, e.g., strong temporal modulation related to harmonics ("vibrato") and highly noisy components over broad frequency bands ("breathy voice"). Automatic classification of singing techniques is an emerging research topic in singing voice analysis [1].

One of the main problems in this task is extracting features from highly dynamic time–frequency textures of singing techniques. Convolutional neural networks (CNNs) have been recently used as effective methods to capture audio features for singing technique classification [2, 3, 4] as well as similar objectives such as musical playing technique recognition [5]. Although square-shaped kernels, e.g., 3×3 and 5×5, are commonly used in CNNs, it has been shown that customizing the kernel shape improves the classification performance. For example, in a study, oblong-shaped kernels outperformed square-shaped ones for singing technique classification [3]. Similar results have been found in music auto-tagging [6] and musical instrument classification [7]. The above findings suggest that more customized kernels may further improve the performance. However, a brute-force search toward the best kernel shape will be burdensome, and thus, a systemic approach is required.

Another critical problem in singing technique classification is data imbalance, which is mainly attributed to the nature of voice production and musical usage. For example, "vocal fry" and "trillo" are difficult to produce for a long time, and thus, the lengths of such audio samples tend to be relatively short. In addition, "belting" is obtained in only certain musical contexts. Thus, collecting well-balanced samples is problematic.

In this study, we deal with the above two problems by deformable convolution and classifier re-training (cRT) using a class-weighted loss, respectively. Deformable convolution allows the convolution kernel to have a flexible shape [8]. It extends the capability of a CNN by modeling geometric transformation, which can be beneficial in capturing dynamic time–frequency features in singing techniques. cRT decouples the feature extractor and the classifier in training a deep neural network model. It was reported as a simple yet powerful method when the class distribution of the training data has a long tail [9].

The contributions of this study are as follows: 1) We investigate different setups of deformable convolution and show that it improves the singing technique classification performance. 2) We show the effectiveness of cRT for an imbalanced dataset, comparing with joint training of the feature extractor and the classifier. 3) Finally, we present that smoothed weighting the loss function further enhances the effect of decoupled training.

2. Related Work

2.1. Deformable Convolution

Deformable convolution was introduced for image processing to enhance the transformation modeling capability of a CNN [8, 10]. It allows CNN models to only focus on what they are interested in and makes the output feature maps more representative. It has been effective in several tasks involving variations in the temporal context, such as action recognition [11], sign language recognition [12], and video captioning [13]. In the audio domain, deformable convolution has been employed in speaker verification [14] and speech recognition [15], to deal with the variable temporal dynamics of speech. In this study, we apply deformable convolution to singing technique classification.

2.2. Data Imbalance

Data imbalance is a common issue in classification tasks. There are two well-known approaches for solving this problem: sampling and cost-sensitive learning [16]. Sampling manipulates the class representations in an original dataset by either over-sampling the minority classes (over-sampling) or under-sampling the majority classes (under-sampling). In the context of deep learning, neither over-sampling nor under-sampling is efficient; over-sampling decelerates the training and may cause overfitting, whereas under-sampling may discard informative majority examples [17]. Cost-sensitive learning is a type of learning that considers the misclassification costs. A simple approach of cost-sensitive learning is reweighting the loss function using inverse class frequency values [18]. However, this strategy may perform poorly when applied to real-world

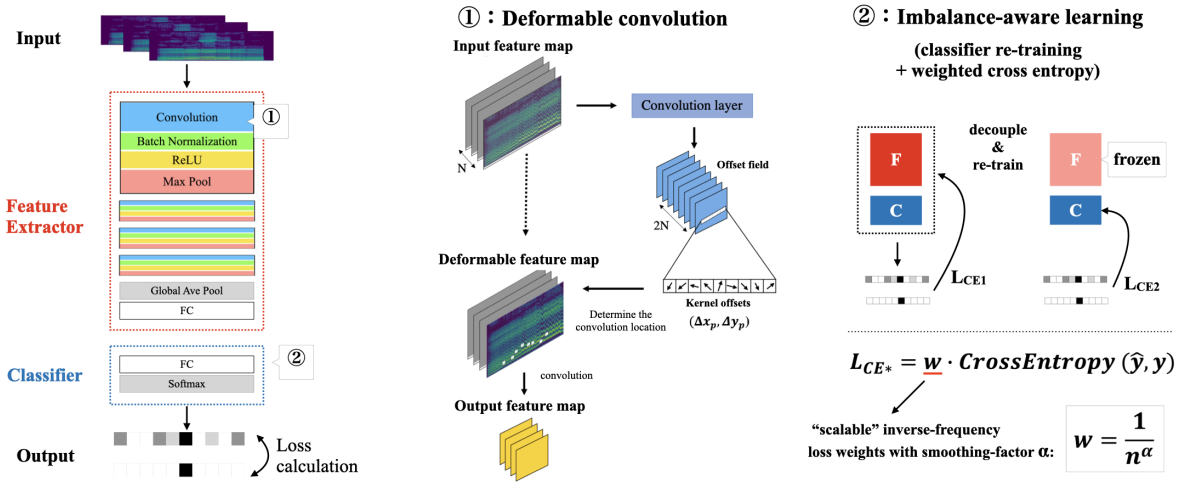

Figure 1: *Overview of the proposed method for singing technique classification*

and large-scale datasets. Comparatively, "smoothed" weighting (e.g., square root of the class frequency values [19] and heuristically determined exponent values [20]) is known to be more effective. Data imbalance was recently addressed in a study by decoupling the feature extractor and the classifier during training of deep neural networks [9]. Empirical experiments showed that the data imbalance problem affects learning classifier decision boundaries, instead of learning feature representations. In this study, we investigate the smoothed weighting and decoupling of the feature extractor and the classifier.

3. Method

Figure 1 shows an overview of our proposed method, and this section describes the details of each of its parts.

3.1. Deformable Convolution

Deformable convolution (DC) facilitates trainable offset parameters of each kernel to deform the convolutional kernel grid. Deformable convolution consists of the following steps: 1) obtain the offset field, 2) output deformable feature maps by the offsets, 3) perform regular convolution on the deformable feature maps. The middle of Figure 1 illustrates the operation of deformable convolution.

1. The offset field is obtained by applying a convolutional layer over the input feature map with channel dimension N. The offset field has the same spatial resolution as the input feature map, and the channel dimension is $2N$. Horizontal offset Δx and vertical offset Δy correspond to each point of the input feature map.
2. Because the offset parameters (Δx and Δy) are typically fractional, the values of offset location are interpolated around the value of closest four points by bilinear interpolation [8].
3. The output feature maps are obtained by operating a regular convolution using the deformable feature maps. For each location p_0 on the input feature map x, and the output feature map y,

$$y(p_0) = \sum_{m \in R}^{M} w(p_0) \cdot x(p_0 + p_m + \Delta p_m)$$

$$p_m = (\Delta x_{p_m}, \Delta y_{p_m})$$

where w, R, and p_m denote the weight of the sampled values, kernel grid with size M, and interpolated offset value, respectively.

We choose a four-oblong-shaped convolution layer CNN with a multi-resolution spectrogram input, which was the best performing model in [3], for the base architecture. The model consists of four convolutional blocks, a global average pooling layer [21] [1], and two fully connected layers.

Note that although its kernel shapes are unidirectional (i.e., $(Vertical, Horizontal) = \{(4 \times 1), (16 \times 1), (1 \times 4), (1 \times 16)\}$), we consider both vertical and horizontal offsets as same, similar to conventional studies [8, 10, 14], to preserve flexibility.

3.2. Weighting Loss Function

We apply a smoothed weighting to the cross-entropy loss function during training, to deal with the data imbalance problem.

$$\mathrm{L}(x, y) = -W \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})} \quad (1)$$

where x is the input, y is the target, and W is the weight of the loss function. We determine the loss weight of each class w_c by the power of the inverse frequency of the training sample as follows:

$$w_c = \frac{1}{(n_c)^{\alpha}} \quad (2)$$

where n_c is the number of training samples in class c, and α is the smoothing factor, controlling smoothing of the loss weights. Note that $\alpha = 0$ corresponds to the value of 1 (i.e., no weighting) and $\alpha = 1$ corresponds to a reciprocal number (i.e., weighting by the inverse class frequency).

3.3. Decoupling Feature Extractor and Classifier

We also investigate decoupling the feature extractor and the classifier from the CNN model, following the method of Kang

[1] In the original study, a flatten layer was used in the top part of the feature extractor. However, in singing technique classification, we confirmed that the global average pooling layer generally outperforms the flatten layer.

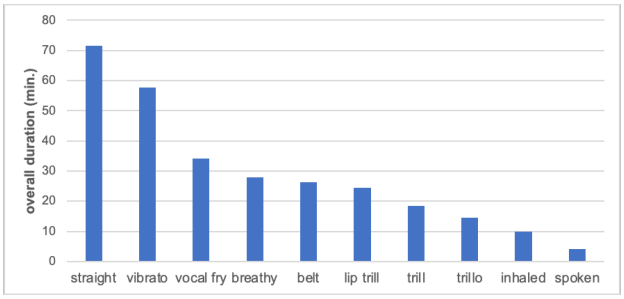

Figure 2: *Audio length distribution of singing techniques in VocalSet [2].*

Table 1: *Configuration of the model. The Four DC conditions differ in the arrangement of DC application. The check mark represents DC application to the corresponding layer.*

Layer Configuration	Ch	Deformable Conv			
		All	Early	Late	Last
Conv(4×1), MP(4×4)	32	✓	✓		
Conv(16×1), MP(4×4)	64	✓	✓		
Conv(1×4), MP(3×3)	128	✓		✓	
Conv(1×16), MP(2×2)	128	✓		✓	✓
Global AP	128		–		
FC (Feature)	30		–		
FC (Softmax)	10		–		

et al. [9]. They proposed two different approaches of the decoupled training method: cRT and normalizing the weights of the classifier by its own norms scaled by a hyperparameter (τ-normalized classifier), called learnable weight scaling, and showed that both outperformed joint training of the classification model. We choose to employ cRT, which was reported as a simple but effective training strategy for an imbalanced dataset. First, the layers of the model are divided into two parts—the feature extractor and the classifier—between the first and second fully connected layers. In the training stage, first the model is trained regularly and subsequently the classifier is re-trained after fixing the weights of the feature extractor part. The right panel of Figure 1 illustrates the training strategy.

4. Experiments

4.1. Dataset

We use VocalSet [2], which is the only publicly available dataset for studies on singing techniques. The dataset contains singing voices of 20 different professional singers (9 female and 11 male) performing 17 different singing techniques in various contexts, such as arpeggio, scale, and long tones. For the classification experiments, we select the samples corresponding to ten different singing techniques ("belt," "breathy," "inhaled singing," "lip trill," "spoken excerpt," "straight tone," "trill," "trillo," "vibrato," and "vocal fry"). Figure 2 shows the total length of each singing technique. The distribution of the dataset has a long-tail shape, i.e., it is imbalanced.

During the learning process, we split the dataset into a training set of 15 singers and a test set of 5 singers [2]. Subsequently, we segment the audio signals in each file into 3-second audio clips and nonoverlapping parts at a sample rate of 44.1 kHz. We evaluate each model using five metrics: macro-F1 score (F1), balanced accuracy (B-Acc.), accuracy (Acc.), top-2 accuracy, and top-3 accuracy.

4.2. Model

We set up four types of deformable convolution model (DC) with weighting and two models without deformbale convolution (w/o DC) with or without weighting. As a result, we compare six conditions in total. For all of these six conditions, the model input and structure are common as follows. The model input is multi-resolution spectrogram (i.e., stacking three spectrograms with different time-frequency resolutions along the channel dimension.) We obtain them by short-time Fourier Transform (STFT), and each spectrogram is obtained by the three window sizes of (2048, 1024, 512 samples) with the same hop length 512 samples and the STFT length 2048 samples with zero-padding. We employed a four-oblong-shaped convolution layer CNN [3] for the model structure. Each convolution block consists of a convolution layer (Conv), a batch normalization layer, a Rectified Linear Unit (ReLU), a max pooling (MP) layer, and a dropout of 0.3. They are followed by a global average pooling (Global AP) layer and two fully-connected layers (FC). We trained our model using the Adam optimizer with a learning rate of 1e-4 and a batch size of 64.

The four DC conditions are denoted as *All*, *Early*, *Late*, and *Last* and their components are listed in Table 1. DC is applied to different layers. All DC models are trained with the weighted loss-function. For the non-DC models, we considered two w/o DC conditions with or without weighting, they are referred to as *w/o DC weighted* and *w/o DC plain*.

4.3. Experiment 1: Effect of Deformable Convolution

We investigate the effect of deformable convolution by replacing standard convolution layers of the model with deformable convolution layers. We tested the six conditions as described in Section 4.2. As baselines, we use one-dimensional CNN (1DCNN) [2] and oblong-CNN feature learning with a random forest classifier (Oblong) [3]. We re-implement the models to investigate the effect of weighting the loss function. For both 1DCNN and Oblong, we tested both weighted and plain (without weighting) conditions. The number of parameter of each conditions are as follows; w/o DC: 337.5k, All: 463.3k, Early: 362.2k, Late: 438.7k, and Last: 435.7k, respectively.

4.4. Experiment 2: Comparative analysis of training strategy and α

We compare three training strategies with a set of smoothing factors α (0, 0.2, 0.5, and 1) in Eq. 2 seeking the best DC setup.

- **Joint training**: without classifier retraining.
- **cRT-WFC**: weights are applied during **both feature representation** training and **cRT** phases.
- **cRT-WC**: weights are applied **only** during the **cRT** phase. (i.e., weights are not applied in the feature representation training phase)

These training strategies were tested upon the *Late* model because it was the best model in experiment 1 as described in Section 5.1. For reasonable comparison, the sum of the number of training epochs is set equal in all conditions. We set 200

[2]The train and test split is officially provided. Refer to the file, "train_singers_technique.txt" in Version 1.2 https://zenodo.org/record/1442513#.YjjqlJrP3a4

Table 2: *The results of experiment 1.*

Models	F1	Acc.	B-Acc.	Top-2	Top-3
1DCNN [2] plain	0.488	0.584	0.484	0.764	0.863
1DCNN [2] weighted	0.306	0.439	0.352	0.643	0.753
Oblong [3] plain	0.540	0.600	0.597	0.757	0.838
Oblong [3] weighted	0.548	0.590	0.613	0.759	0.852
w/o DC plain	0.404	0.492	0.472	0.686	0.805
w/o DC weighted	0.513	0.554	0.575	0.743	0.858
All (1,2,3,4)	0.553	0.604	0.59	0.799	**0.896**
Early (1,2)	0.554	0.593	0.598	0.776	0.862
Late (3,4)	**0.582**	**0.623**	**0.641**	**0.806**	0.894
Last (4)	0.517	0.572	0.607	0.764	0.846

Table 3: *The results of comparison between joint-training, cRT-WC and cRT-WFC, under* $\alpha = 0.2$.

Methods	F1	Acc.	B-Acc.	Top-2	Top-3
Joint-training	0.559	0.610	0.635	0.774	0.874
cRT-WFC	0.582	0.623	0.641	0.806	**0.894**
cRT-WC	**0.620**	**0.656**	**0.655**	**0.815**	0.887

epochs for the entire training time. For all cRT-based methods, we assign 100 epochs for the joint training of the feature extractor and the classifier, and the remaining 100 epochs for the cRT.

5. Results and Discussions

5.1. Effect of Deformable Convolution

The results of experiment 1 are listed in Table 2. They show that DC models significantly improve the classification performance compared to w/o DC models. Among the four DC setups, the *Late* model achieves the best. This agrees with the results from previous works that applying DC to several late convolution layers is effective [10, 15]. Compared to the *Last* model where DC is applied only to the last convolution layer, the accuracy of the *Late* model becomes much higher. Class-wise accuracy may explain this gap: With the *Late* model we observed large accuracy increments on the discrimination of "lip trill" and "vocal fry," which have fine temporal modulation in amplitude, frequency, and breathiness.

This indicates that the small kernel size of the 3rd DC layer plays an important role when the dynamic offset adapts the fine modulation of singing voice. The baseline model with Oblong kernel-shapes achieves higher accuracy than the model without DC, as it uses a random forest classifier on a similar configuration of CNN feature extractor. However, the *Late* model extracts the features more effectively with DC and outperforms the baseline model.

5.2. Effect of cRT

Table 3 shows the results for the training strategies comparison, summarizing the output with the smoothing factor $\alpha = 0.2$ (as discussed in Section 5.3.) Both cRT methods outperform the joint-training method. Between two cRT methods, cRT-WC significantly improves the classification performance. This suggests that the weighting loss-function is only effective in cRT and so it is better to apply the weighting only during the re-training phase. A similar result was also reported in [9].

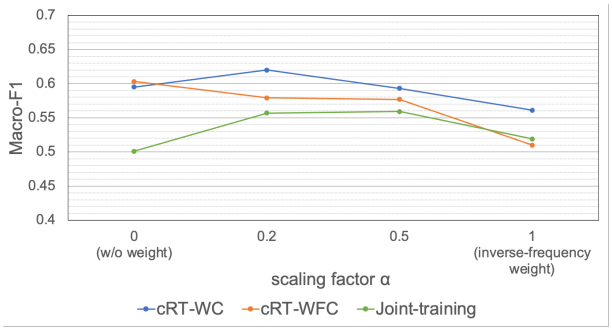

Figure 3: *Macro F1 score for cRT-WC, cRT-WFC, and joint-training respectively with four different* α *values.*

5.3. Effect of Smoothing Factor α

We conducted experiment 2 with four different values of the smoothing factor α; 0, 0.2, 0.5, and 1. Figure 3 plots Macro-F1 over the smoothing factor. The best performing condition is cRT-WC with an α value of 0.2. As α increases, the performance keeps decreasing in all three conditions and reaches the worst accuracy at an α value of 1 (i.e., inverse-frequency weight).

Increasing α has the expected effect of improving performance of minority classes while hurting majority classes. However, when we vary α from 0.2 to 1. the class-wise F1 scores decreased for both minority (e.g., "inhaled" 0.293 $\rightarrow$ 0.268, "trill" 0.544 $\rightarrow$ 0.495) and majority (e.g., "straight" 0.69 $\rightarrow$ 0.645, "vibrato" 0.648 $\rightarrow$ 0.623.) It corresponds to the result of conventional works [19, 17] that inverse frequency weight decreased the performance in large-scale long-tail classification problems.

This indicates that classification difficulty comes from not only data imbalance but also similarity between class samples, e.g., "vibrato" (majority class) and "trill", "trillo" (minority classes). These techniques exhibit both frequency and amplitude modulations, while vibrato and trill mainly rely on frequency modulation and trillo on amplitude modulation. However, close observation of trillo spectrogram also shows some frequency modulation [2]. Detecting these subtle balance of amplitude and frequency modulations was the difficulty in this task.

6. Conclusion

In this paper, we proposed audio feature learning by deformable convolution and imbalance-aware learning based on classifier decoupling and a weighted inverse frequency loss, for singing technique classification. The experiments showed that applying deformable convolution in the last two layers and cRT with smoothed inverse frequency weights improve the classification performance. Future study can explore more complex weighting-based loss functions (e.g., [17]) and evaluating our concept on real-world singing performances, in which the problems of this study (i.e., feature learning and label sparseness [22]) are more serious.

7. Acknowledgements

This work was supported by JST SPRING, Grant Number JPMJSP2124.

8. References

[1] E. J. Humphrey, S. Reddy, P. Seetharaman, A. Kumar, R. M. Bittner, A. Demetriou, S. Gulati, A. Jansson, T. Jehan, B. Lehner *et al.*, "An introduction to signal processing for singing-voice analysis: High notes in the effort to automate the understanding of vocals in music," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 82–94, 2018.

[2] J. Wilkins, P. Seetharaman, A. Wahl, and B. A. Pardo, "Vocalset: A singing voice dataset," in *19th International Society for Music Information Retrieval Conference, (ISMIR)*. International Society for Music Information Retrieval, 2018, pp. 468–474.

[3] Y. Yamamoto, J. Nam, H. Terasawa, and Y. Hiraga, "Investigating time-frequency representations for audio feature extraction in singing technique classification," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2021, pp. 890–896.

[4] B. O'Connor, S. Dixon, and G. Fazekas, "Zero-shot singing technique conversion," in *Proceedings of 15th In-ternational Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2021, pp. 235–244.

[5] J. Abeßer and M. Müller, "Fundamental frequency contour classification: A comparison between hand-crafted and cnn-based features," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2019, pp. 486–490.

[6] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2016, pp. 1–6.

[7] T. Takahashi, S. Fukayama, and M. Goto, "Instrudive: A music visualization system based on automatically recognized instrumentation." in *19th International Society for Music Information Retrieval Conference, (ISMIR)*, 2018, pp. 561–568.

[8] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017, pp. 764–773.

[9] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=r1gRTCVFvB

[10] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9308–9316.

[11] P. Lei and S. Todorovic, "Temporal deformable residual networks for action segmentation in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018, pp. 6742–6751.

[12] K. Papadimitriou and G. Potamianos, "Multimodal sign language recognition via temporal deformable convolutional sequence learning." in *Proceedings of the Interspeech*, 2020, pp. 2752–2756.

[13] J. Chen, Y. Pan, Y. Li, T. Yao, H. Chao, and T. Mei, "Temporal deformable convolutional encoder-decoder networks for video captioning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8167–8174.

[14] Y. Zhang, H. Yu, and Z. Ma, "Speaker verification system based on deformable cnn and time-frequency attention," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 1689–1692.

[15] K. An, Y. Zhang, and Z. Ou, "Deformable TDNN with Adaptive Receptive Fields for Speech Recognition," in *Proceedings of Interspeech*, 2021, pp. 2067–2071.

[16] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International journal of pattern recognition and artificial intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[17] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2019, pp. 9268–9277.

[18] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 5375–5384.

[19] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten, "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 181–196.

[20] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 483–498.

[21] M. Lin, Q. Chen, and S. Yan, "Network in network," in *International Conference on Learning Representations*, 2014. [Online]. Available: https://openreview.net/forum?id=ylE6yojDR5yqX

[22] Y. Yamamoto, D. Moriyama, J. Nam, and H. Terasawa, "Towards computational analysis of singing technique for music information retrieval : A progress report of building dataset and statistical analysis," *Proceedings of the auditory research meeting*, vol. 51, no. 8, pp. 569–572, 2021. [Online]. Available: https://ci.nii.ac.jp/naid/40022767415/