TWO-STAGE ANOMALOUS SOUND DETECTION SYSTEMS USING DOMAIN GENERALIZATION AND SPECIALIZATION TECHNIQUES

Technical Report

Ibuki Kuroyanagi[1,2], *Tomoki Hayashi*[1,2], *Kazuya Takeda*[1], *Tomoki Toda*[1]

[1] Nagoya University, School of Informatics, Nagoya, Japan
[2] Human Dataware Lab. Co., Ltd., Nagoya, Japan

ABSTRACT

This report proposes anomalous sound detection (ASD) methods using domain generalization and specialization techniques for the DCASE 2022 Challenge Task 2. We propose two-stage ASD systems consisting of an outlier exposure-based feature extractor and an inlier modeling-based anomalous detector in serial. We further employ two approaches to deal with domain shift: a domain generalization approach and a domain specialization approach. Each approach improves performance significantly by adding several techniques to the two-stage ASD systems, such as generating pseudo-target domain data by Mixup and utilizing pseudo-anomalous data from Audioset. Our final systems are obtained by ensembling several systems with several hyperparameters for each approach. The proposed systems achieve 81.15 % in the harmonic mean of all machine types, sections, and domains for the area under the curve (AUC) and partial AUC ($p = 0.1$) on the development set.

***Index Terms*—** anomalous sound detecion, outlier exposure, inlier modeling, domain shift, mixup

1. INTRODUCTION

This report describes unsupervised anomalous sound detection (ASD) methods developed for the DCASE 2022 Challenge Task 2 [1]. This task aims for machine condition monitoring and requires detecting unknown anomalous data using only normal data. We propose two-stage ASD methods for this task. In the first stage, a feature extractor is trained by introducing an outlier exposure (OE) approach that classifies normal and pseudo-anomalous data. In the second stage, to create an anomalous detector, inlier modeling (IM) is employed to model the probability distribution of normal data using the features obtained from the feature extractor created in the first stage. During inference, data that deviates from the probability distribution of normal data is detected as an anomaly. Datasets used in this task [2, 3] are subject to a domain shift in which the surrounding environment and the operating conditions of the machine change between the training and test data. In DCASE 2021 Challenge Task 2 [4], the domain to which the test data belonged is known in advance, but in this task, the domain to which the test data belongs is not known. We employ two approaches to address this issue. The first approach, which we call the domain generalization approach, is to develop systems that generalize domain differences. The systems can be used with this approach without caring about domain shifts. However, there is a trade-off between generalization performance and detection performance. Another approach to address this issue, which we call the domain specialization approach, is to develop systems specialized for each domain and combine the identification results with a domain classifier. Finally, we ensemble these systems tuned with several hyperparameters to maximize the detection performance.

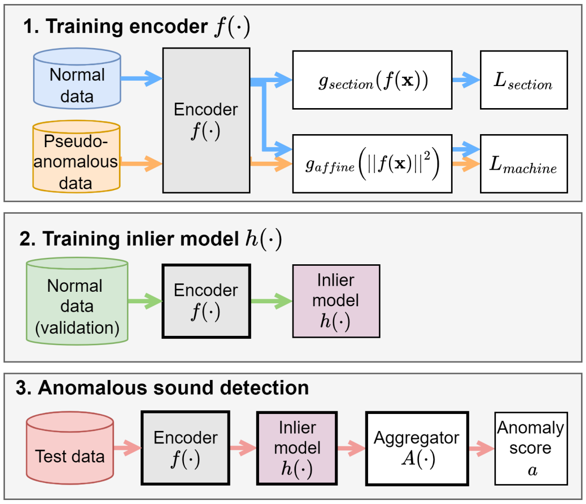

Figure 1: An overview of the two-stage ASD method.

We conduct an experimental evaluation of the developed system using the DCASE 2022 Task 2 Challenge development datasets. Here, the datasets [2, 3] contain seven machine types. Each machine type has six section IDs, reflecting a domain shift scenario type. The training data contains domain data for both the source and target domains, but only a few samples of target domain data. Experiments on the datasets show that all of the created systems significantly outperformed the official baseline system in the evaluation metric, the harmonic mean of the area under the curve (AUC), and partial AUC ($p = 0.1$) for all machines types, and section IDs (all / har-mean). The domain generalization approach achieved 80.71 %, and the domain specialization approach achieved 81.15 % on the all / har-mean.

2. METHOD

2.1. Two-stage ASD method

As the base of our system, we use [5] as the ASD method. An overview of the method is shown in Figure 1. The method consists of two stages: a feature extractor using OE and an anomalous detector using IM. When training the feature extractor, this method explicitly divides the training method according to the type of pseudo-

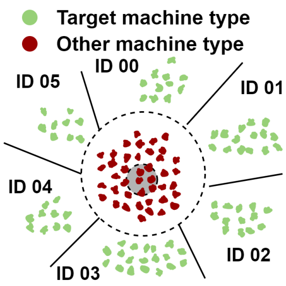

Figure 2: A schematic image of embeddings obtained by the two-stage ASD method. Pseudo-anomalous data are distributed in the center of the hypersphere, and normal data are distributed away from the center of the hypersphere for each ID.

anomalous data. It allows for cases in which normal and pseudo-anomalous data distributions are highly similar or different. This method obtains the feature extractor by multi-task training of two types of binary cross-entropies. The first is a loss function that classifies the sound of the target machine type to which section ID it is emitted from, which deals with the case where the normal data and the pseudo-anomalous data are highly similar. The second is a loss function that identifies whether the sound is emitted from the target machine type or not, which deals with the case where the normal data and the pseudo-anomalous data are highly different. Figure 2 shows a schematic image of an embedding obtained by the method.

If the audio input is represented as x_i $(i = 1, 2, ..., N)$, the machine type is represented as t_i $(i = 1, 2, ..., N)$, where t_i is 1 for the target machine type and 0 for the other machine types. Each machine type has K section IDs, and x_i belongs to one of them. The one-hot vector for the section ID is represented as $y_{\{i,k\}}$ $(i = 1, 2, ..., N, k = 1, 2, ..., K)$, where $y_{\{i,k\}}$ is 1 for the k th element and 0 for the other elements when the section ID is k. The section IDs classification loss is calculated as follows:

$$\begin{aligned}\mathcal{L}_{\text{section}} = &-\frac{1}{K\sum_{i=1}^{N} t_i}\sum_{i=1}^{N}\sum_{k=1}^{K} t_i \left\{y_{\{i,k\}}\log\left(\sigma\left(g_{\text{section}}\left(f(x_i)\right)\right)\right)\right.\\ &\left.+(1-y_{\{i,k\}})\log\left(1-\sigma(g_{\text{section}}\left(f(x_i)\right))\right)\right\},\end{aligned} \tag{1}$$

where f is an encoder, g_{section} is a linear transformation, and σ is a sigmoid function. The machine type classification loss is calculated as follows:

$$\begin{aligned}\mathcal{L}_{\text{machine}} = &-\frac{1}{N}\sum_{i=1}^{N}\left\{t_i\log\left(\sigma\left(g_{\text{affine}}\left(||f\left(x_i\right)||^2\right)\right)\right)\right.\\ &\left.+(1-t_i)\log\left(1-\sigma\left(g_{\text{affine}}\left(||f\left(x_i\right)||^2\right)\right)\right)\right\},\end{aligned} \tag{2}$$

where g_{affine} is an affine transformation. When creating mini-batches, we use a batch sampler so that the value of t is 1:1. The final loss function is calculated by the following equation:

$$\mathcal{L} = \mathcal{L}_{\text{machine}} + \lambda_{\text{section}}\mathcal{L}_{\text{section}}, \tag{3}$$

where λ_{section} is a hyperparameter. During inference, the anomaly score a_i $(i = 1, 2, ..., N)$ is calculated as follows:

$$a_i = \mathcal{A}\left(h\left(f\left(\mathcal{X}_i\right)\right)\right), \tag{4}$$

where $\mathcal{X}_i$ is a set of S segments that divide x_i into T seconds, allowing for overlap, h is an anomalous detector by IM such as Gaussian mixture models (GMM) [6, 7], local outlier factor (LOF) [8], or k-nearest neighbor algorithm (KNN) [9] and $\mathcal{A}$ is an aggregator of the anomaly scores such as average / max / median pooling.

2.2. Improvement of a feature extractor

We make two changes to improve the feature extractor.

The first is to use Audioset [10] as pseudo-anomalous data when training the feature extractor by OE. It improves the robustness of anomalous data whose distribution differs significantly from normal data.

The second is to use not only EfficientNet-B0 [11] but also Conformer [12] and Transformer [13] for the models used in the feature extractor. We obtain feature extractors that focus on different features using convolutional neural network-based and self-attention-based models.

As a result, these modifications especially improve the target domain performance.

2.3. Domain generalization approach

The key to domain generalization is treating data from the source and target domains as the same. We employ two techniques for domain generalization.

The first is to sample the normal data in the target domain in creating a mini-batch so that at least one sample in the target domain is in the mini-batch when training the feature extractor by OE. It reduces the problem of data imbalance between the source and target domains.

The second is to use a Mixup [14] of source and target domain data to generate 50 samples of pseudo-normal data when training the anomalous detector by IM. It is effective since it models the intermediate data representation in the source and target domains as normal data.

2.4. Domain specialization approach

The key to domain specialization is treating data from both the source and target domains as different data. We employ two techniques for domain specialization.

The first is to use only data from each domain when training the anomalous detectors for each domain. Although the number of the target domain data is small, the performance of the target domain data is improved compared to modeling it using the source and target domain data.

The second is the development of a high-performance domain classifier. The domain classifier plays an important role in the domain specialization approach. The reason is that if the performance of the domain classifier is poor, the performance degrades due to differences in the distribution of anomaly scores between the source and target domains when weighted averaging. The domain classifier is implemented by adding a new domain classification module to the feature extractor in Section 2.1. The domain classifier is trained by extending the training by OE. We create the mini-batch so that it includes $1/4$ of the source domain's data, $1/4$ of the pseudo-target domain's data, and $1/2$ of preudo-anomalous data. The pseudo-target domain's data is generated by mixing up the target domain and pseudo-anomalous data.The domain classification module then classifies the domain of the normal data. Here, the domain is set to d_i $(i = 1, 2, ..., N)$, where 1 is the target domain, and 0 is the

Table 1: *Evaluation results. The values represent the harmonic mean of AUC* [%] *and pAUC* ($p = 0.1$) [%] *for each section of source domain. The value in the column "all / har-mean" represents the harmonic mean of AUC and pAUC over all machines and sections.*

Method	beraing	fan	gearbox	slider	valve	ToyCar	ToyTrain	all / har-mean
Official Autoencoder-based baseline	54.42	78.59	68.93	77.95	52.01	90.41	76.32	67.29
Official MobileNetV2-based baseline	60.55	70.75	69.19	65.05	67.66	67.66	57.57	63.04
I.Kuroyanagi *et al* [5]	64.76	70.78	78.36	93.08	92.03	63.40	63.70	73.36
Domain generalization approach	69.46	68.90	78.98	94.52	92.33	73.58	72.46	77.46
Domain specialization approach	68.16	73.87	79.44	94.18	92.18	74.82	79.08	79.30
Domain generalization approach per machine	84.99	80.41	86.10	96.09	96.28	82.29	76.59	85.55
Domain specialization approach per machine	**90.38**	**84.21**	**90.82**	**96.66**	**97.42**	**91.03**	**84.86**	**90.52**

Table 2: *Evaluation results. The values represent the harmonic mean of AUC* [%] *and pAUC* ($p = 0.1$) [%] *for each section of target domain. The value in the column "all / har-mean" represents the harmonic mean of AUC and pAUC over all machines and sections.*

Method	beraing	fan	gearbox	slider	valve	ToyCar	ToyTrain	all / har-mean
Official Autoencoder-based baseline	58.38	47.18	62.64	47.67	49.46	34.81	23.35	43.45
Official MobileNetV2-based baseline	60.09	48.22	56.23	38.40	57.75	57.75	45.79	50.90
I.Kuroyanagi *et al* [5]	76.01	61.07	65.41	68.58	84.84	54.42	45.99	62.89
Domain generalization approach	88.43	63.52	66.61	69.16	83.16	64.32	50.19	67.29
Domain specialization approach	85.91	63.17	68.48	76.46	88.52	67.22	53.27	69.93
Domain generalization approach per machine	**91.66**	75.28	81.21	86.90	91.30	75.58	61.55	79.14
Domain specialization approach per machine	91.38	**78.49**	**82.13**	**90.38**	**93.52**	**80.24**	**66.66**	**82.28**

source domain. For pseudo-target domain data, d_i is the mixing ratio; for pseudo-anomalous data, d_i is 0. The domain classification loss is calculated as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{domain}} = -\frac{1}{\sum_{i=1}^{N} t_i} \sum_{i=1}^{N} t_i \left\{ d_i \log\left(\sigma\left(g_{\mathrm{domain}}\left(f(x_i)\right)\right)\right) \right. \\ \left. + (1-d_i) \log\left(1 - \sigma\left(g_{\mathrm{domain}}\left(f(x_i)\right)\right)\right) \right\},\end{aligned} \tag{5}$$

where g_{domain} is a linear transformation. The final loss function for the domain classifier is obtained by the following equation:

$$\mathcal{L} = \mathcal{L}_{\mathrm{machine}} + \lambda_{\mathrm{section}} \mathcal{L}_{\mathrm{section}} + \lambda_{\mathrm{domain}} \mathcal{L}_{\mathrm{domain}}, \tag{6}$$

where $\lambda_{\mathrm{domain}}$ is a hyperparameter. The probability of the target domain p_i $(i = 1, 2, ..., N)$ is calculated as follows:

$$p_i = \sigma(g_{\mathrm{domain}}(f(x_i))/\kappa), \tag{7}$$

where $\kappa = 0.15$ is a hyperparameter for varying the scale of the output of the domain classifier. The anomaly score a_i $(i = 1, 2, ..., N)$ for the domain specialization approach is calculated as follows:

$$a_i = (1 - p_i)\hat{a}_{\mathrm{source}_i} + p_i \hat{a}_{\mathrm{target}_i}, \tag{8}$$

where $\hat{a}_{\mathrm{source}_i}$ and $\hat{a}_{\mathrm{target}_i}$ are anomaly scores standardized by each section ID for the source and the target domain, respectively.

2.5. Ensemble

For both approaches, ensembles are effective in improving performance. When ensembling, the anomaly scores are standardized by each section ID before being used since the output scales differently depending on the anomalous detectors h. The domain generalization approach obtains anomaly scores by selecting multiple models and averaging them. The domain specialization approach selects multiple models from each domain and averages to obtain anomaly scores for the source and target domains. Finally, the anomaly score is obtained by Eq. 8.

3. EXPERIMENTAL EVALUATION

3.1. Experimental conditions

We evaluated the performance of the proposed systems using the DCASE 2022 Task 2 Challenge development sets (MIMII DG [3], ToyADMOS2 [2]). The datasets included seven machine types: bearing, fan, gearbox, valve, slider, ToyCar, and ToyTrain. Each machine type had six-section IDs, and each sample belonged to one. Each section ID was given 990 samples of normal data in the source domain and ten samples in the target domain. Each domain was given 50 samples of normal and anomalous data for the test data. Each recording was a single-channel, 10 sec. segment of audio sampled at 16 kHz. For training of the feature extractor, $r_{\mathrm{train}} \in \{80, 85, 90\}$ % of the source domain data and six samples of the target domain data were used. The remaining samples were used to validate the training of the feature extractor and to train the anomalous detector. The feature extractors were trained for $N_{\mathrm{epoch}} \in \{150, 200, 250, 300\}$ epochs, where one epoch means that all normal source domain data was updated once. We used the 1.8 million samples of Audioset [10] that were available for download as pseudo-anomalous data.

The amplitude of the audio input sequence was standardized to have a mean of 0 and a variance of 1. The audio input sequence was extracted as Mel-spectrogram with a window size of 128 ms, a hop size of 16 ms, and 224 Mel-spaced frequency bins in the range of 50–7800 Hz in 2.0 sec. The feature was passed to the encoder f using EfficientNet-B0 [11], Conformer [12], or Transformer [13]. The scheduler was OneCycleLR [15], and the optimizer was AdamW [16] with the learning rate of 0.001. The batch size was set to 128. $\lambda_{\mathrm{section}}$ in Eq. 3 was set to 10 and $\lambda_{\mathrm{domain}}$ in Eq. 6 was set to 1. It was a hyperparameter that whether or not using Mixup to obtain intermediate features between normal and pseudo-anomalous data during training for the feature extractor. GMM, LOF, or KNN were used for the anomalous detector h. The hyperparameter of the anomalous detector h was the number of components for GMM or the number of neighbors for LOF or KNN, where it was one of $\{1, 2, 4, 16, 32\}$. During inference, we divided 10.0 sec. clips into $S = 10$ segments with overlapping. As

Table 3: *Evaluation results. The values represent the harmonic mean of AUC* [%] *and pAUC* ($p = 0.1$) [%] *for each section of target domain. The value in the column "all / har-mean" represents the harmonic mean of AUC and pAUC over all machines, sections and domains.*

Method	beraing	fan	gearbox	slider	valve	ToyCar	ToyTrain	all / har-mean
Official Autoencoder-based baseline	54.07	58.23	61.86	57.42	50.53	51.47	58.59	56.10
Official MobileNetV2-based baseline	58.59	57.15	58.90	51.31	62.48	62.48	51.26	56.31
I.Kuroyanagi *et al* [5]	67.40	65.06	71.34	73.96	91.05	59.60	51.64	66.75
Domain generalization approach	73.25	65.33	72.50	75.72	91.21	67.12	54.18	69.81
Domain specialization approach	71.38	66.32	73.34	79.06	92.43	70.71	57.18	71.56
Domain generalization approach per machine	86.87	75.38	82.46	**90.45**	94.94	77.23	**65.40**	80.71
Domain specialization approach per machine	**87.81**	**78.15**	**84.35**	89.34	**96.33**	**79.03**	62.53	**81.15**

Table 4: *Domain classification results. The values represent the mean of AUC* [%] *for each section. The value in the column "all / mean" represents the mean of AUC over all machines and sections.*

	beraing	fan	gearbox	slider	valve	ToyCar	ToyTrain	all / mean
domain classification model	77.45	92.90	65.30	75.15	70.38	98.68	99.82	82.81

a result, each segment was $T = 2.0$ sec. The GMM used the negative log-likelihood as the anomaly score, while the LOF and KNN used the outlier score. The aggregator $\mathcal{A}$ was used one of the mean, maximum, median, and mean above the median.

3.2. Experimental results

Tables 1, 2, 3, and 4 show the performance of the source domain, the performance of the target domain, the performance of both domains evaluated together, and the performance of the domain classifier, respectively. We compared the performance of following seven systems.

Official Autoencoder-based baseline It was an IM-based anomalous detector that used an autoencoder. It was trained to minimize the reconstruction error of the normal training data to obtain small anomaly scores for normal sounds. The anomaly score was calculated as the reconstruction error of the observed sound.

Official MobileNetV2-based baseline It was an OE-based anomalous detector that used MobileNetV2 [17]. It identified from which section ID the observed signal was generated. The anomaly score was calculated as the averaged negative logit of the predicted probabilities for the correct section.

I. Kuroyanagi *et al.* [5] It was the two-stage ASD method described in Section 2.1.

Domain generalization approach It was the average anomaly score of the top 20 performing systems obtained by the domain generalization approach. It used the same hyperparameters for all machine types. It was used as submission 1.

Domain specialization approach It was the average anomaly score of the top 20 performing systems obtained by the domain specialization approach. It used the same hyperparameters for all machine types. It was used as submission 2.

Domain generalization approach per machine It was the average anomaly score of the top 20 performing systems obtained by the domain generalization approach. It used the best hyperparameters for each machine type. It was used as submission 3.

Domain specialization approach per machine It was the average anomaly score of the top 20 performing systems obtained by the domain specialization approach. It used the best hyperparameters for each machine type and domain. It was used as submission 4.

As a result, the proposed systems outperformed the performance of the official baselines and the conventional method. We compared the domain generalization approach with the domain specialization approach. The domain specialization approach had better all / har-mean performance. The performance differences of each approach were significant, especially when using hyperparameters optimized for each machine type, as shown in Table 1 and Table 2. However, the performance of both domains evaluated together was small between each approach, as shown in Table 3. It could be due to the inadequate performance of the domain classifiers when mixing their respective scores. The domain classifier scores in Table 4 shows that fan, ToyCar, and ToyTrain were well classified, but the other machine types had poor classification performance. The poor performance of the domain classifier reflected anomaly scores from different domains. Another possibility was that the distribution of anomaly scores for different domains differ; ToyTrain achieved adequate domain classifier performance, but the performance of both domains evaluated together was lower than that of source and target domains. We believed it was because even with standardization in each section ID, the anomaly scores still scale differently, and the thresholds used for anomaly detection were different. The results indicated that both approaches have their advantages and disadvantages.

4. CONCLUSION

We proposed ASD methods using domain generalization and specialization techniques for the DCASE 2022 Challenge Task 2. The proposed systems were two-stage ASD systems that used an OE-based feature extractor and an IM-based anomalous detector in series. We proposed two approaches to deal with domain shifts: a domain generalization approach and a domain specialization approach. Each approach made several techniques to the two-stage ASD system and significantly improved their performance by ensembling several systems. Future work will focus on developing systems that can achieve higher performance with a single model, even in a domain-shifted environment.

5. ACKNOWLEDGMENT

This paper was partly supported by a project, JPNP20006, commissioned by NEDO.

6. REFERENCES

[1] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," *In arXiv e-prints: 2206.05876*, 2022.

[2] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 1–5.

[3] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," *In arXiv e-prints: 2205.13879*, 2022.

[4] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 186–190.

[5] I. Kuroyanagi, T. Hayashi, K. Takeda, and T. Toda, "Improvement of serial approach to anomalous sound detection by incorporating two binary cross-entropies for outlier exposure," 2022. [Online]. Available: https://arxiv.org/abs/2206.05929

[6] D. W. Scott, "Outlier Detection and Clustering by Partial Mixture Modeling," in *COMPSTAT 2004 — Proceedings in Computational Statistics*, Springer. Physica-Verlag HD, 2004, pp. 453–464.

[7] W. Liu, D. Cui, Z. Peng, and J. Zhong, "Outlier Detection Algorithm Based on Gaussian Mixture Model," in *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 2019, pp. 488–492.

[8] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," *SIGMOD Rec.*, vol. 29, no. 2, p. 93–104, may 2000. [Online]. Available: https://doi.org/10.1145/335191.335388

[9] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient Algorithms for Mining Outliers from Large Data Sets," *SIGMOD Rec.*, vol. 29, no. 2, p. 427–438, may 2000. [Online]. Available: https://doi.org/10.1145/335191.335437

[10] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[11] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 687–10 698.

[12] A. Gulati, C.-C. Chiu, J. Qin, J. Yu, N. Parmar, R. Pang, S. Wang, W. Han, Y. Wu, Y. Zhang, and Z. Zhang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-3015

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[14] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[15] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, T. Pham, Ed., vol. 11006, International Society for Optics and Photonics. SPIE, 2019, pp. 369 – 386. [Online]. Available: https://doi.org/10.1117/12.2520589

[16] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.