A MULTI-MODAL FUSION APPROACH FOR AUDIO-VISUAL SCENE CLASSIFICATION ENHANCED BY CLIP VARIANTS

Soichiro Okazaki, Quan Kong, Tomoaki Yoshinaga

Lumada Data Science Lab., Hitachi, Ltd.
{soichiro.okazaki.xs, quan.kong.xz, tomoaki.yoshinaga.xc}@hitachi.com

ABSTRACT

In this paper, we propose a system for audio-visual scene classification with a multi-modal ensemble way consisting of three features: (1) Log-mel spectrogram audio features extracted by CNN variants from audio modality. (2) Frame-wise image features extracted by CNN variants from video modality. (3) Another frame-wise image features extracted by OpenAI CLIP models which are trained with a large-scale web crawling text and paired image dataset under contrastive learning framework. We trained the above three models respectively and made an ensemble weighted by class-wise confidences of each model's semantic outputs. As a result, our ensemble system reached 0.149 log-loss (official baseline: 0.658 log-loss) and 96.1% accuracy (official baseline: 77.0% accuracy) on TAU Audio-Visual Urban Scenes 2021 dataset which are used in DCASE2021 Challenge Task1B.

Index Terms— Audio-visual Scene Classification, Multi-modal, CLIP, Convolutional Neural Network, Vision Transformer, Log-mel Spectrogram, SpecAugment, Random Erasing

1. INTRODUCTION

Audio-visual scene classification is one of the classification problems which uses both audio and video modalities for classifying the defined scene. Like human perception, we can expect to create a better model by exploiting complementary information from different modalities.

As recent research, several works tackle audio-visual joint learning. In [1, 2, 3, 4], the recognition performance of the audio-visual models is enhanced with a self-supervised manner using multi-modal information in each way. From action recognition perspectives, [5] proposed the Audiovisual SlowFast Networks, which utilize the SlowFast Networks [6] mainly used in video recognition tasks. In this research, audio features are concatenated to the image features through the model's internal pathway for multi-modal fusion. Other general multi-modal architectures are also used for audio-visual recognition tasks. [7] proposed the general perception architecture called Perceiver, which can treat image and audio features in the same way through concatenating the over 50 thousand dimension inputs. For leveraging audio-visual recognition performance using multi-modality information, various research has been proposed vigorously.

This year, Detection and Classification of Acoustic Scenes and Events (DCASE) challenge 2021 [8] holds the audio-visual scene classification task as Task1B [9] with a large-scale dataset called TAU Audio-Visual Urban Scenes 2021. This dataset provided by the organizer contains synchronized audio and video recordings from 12 European cities in 10 different scenes [10].

This paper describes the details of our team's (team name: LD-SLVision) solution for Task1B of DCASE2021. For this task, we developed various audio classification models and video classification models, and created final submissions by fusing those models using an ensemble method and a post-processing technique.

The features of our system can be concluded as three folds:

1) Instead of learning raw audio waves directly, we only used log-mel spectrogram features extracted from audio files as inputs, and leveraged those features with strong CNN variants which are used vigorously in the recent computer vision community.

2) We developed CLIP Late Fusion Network, which uses extracted features from various CLIP image encoders [11] as inputs for a multi-branch network. As far as we know, this is the first approach which uses CLIP models for audio-visual scene classification task.

3) We applied a post-processing technique to suppress the value of log-loss, which is defined as the competition's metric.

2. PROPOSED SOLUTION

In this section, we describe the details of our solution for DCASE2021 challenge Task1B. For tackling the audio-visual scene classification task, we created various audio classification models and video classification models respectively in each modality. After created various models, we integrated these models with ensemble method and applied post-processing technique to suppress the log-loss value for final submissions. The overview of our multi-modal fusion approach is shown in Fig. 1.

2.1. Audio Classification Models by Log-mel CNN Variants

For utilizing the audio modality of the provided dataset, we created various audio classification models with 1 second split audio files. The test dataset is provided as 1 second audio files in this competition. Therefore, we divided each 10 seconds audio files provided as development dataset for DCASE2021 Task1B into ten 1 second audio files.

As inputs for audio classification models, we extracted log-mel spectrograms with delta/delta-delta features, which are also used in DCASE2020 Task1A winner's solution [12]. We used librosa library [13] for creating log-mel spectrograms. The parameters of log-mel spectrogram transformation are as follows: sampling rate (sr) is 48kHz, the number of mel bins (n_mels) is 256, the length of FFT window (n_fft) is 4096, and the number of samples between successive frames (hop_length) is 512.

About the choice of inputs type (e.g. log-mel spectrogram, raw audio waveform, etc.), we referred to the PANNs paper [14] which proposed a widely used audio classification model. In the

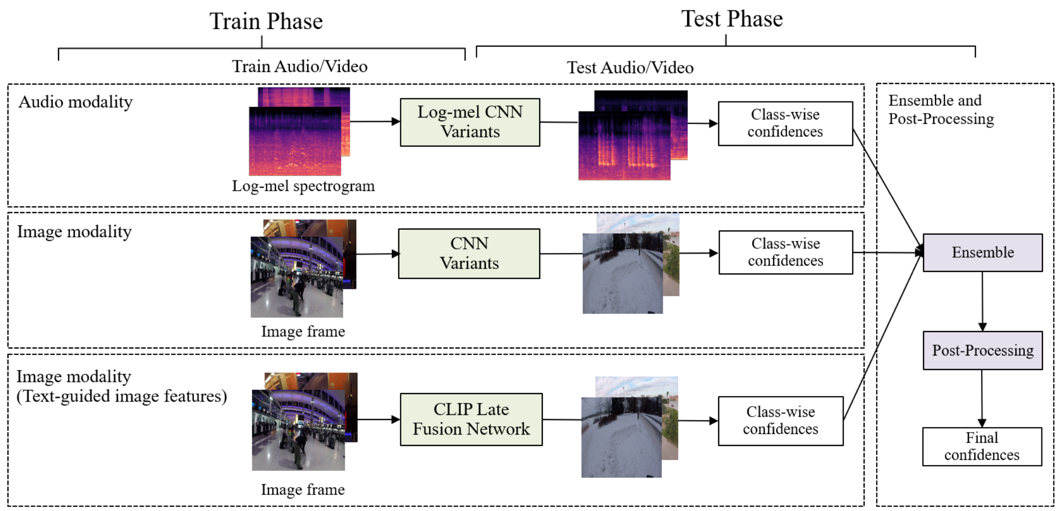

Figure 1: The overview of our system for DCASE2021 challenge Task1B. This picture shows the case of S02 in Table 3.

experiments of PANNs paper, the recognition performance of log-mel spectrogram classification with ResNet-38 [15] is competitive with that of wavegram log-mel CNN which is proposed as state-of-the-art architecture in the paper. Therefore, we only selected log-mel spectrogram features as inputs, and uses these features with strong CNN backbones (EfficientNet [16] with Noisy Student [17], ResNeSt [18], RegNet [19]) which are better than ResNet-38 in the recent computer vision community.

2.2. Video Classification Models by CNN Variants

For creating video classification models, we firstly extracted 12 image frames from 10 seconds video file with equal interval. After extracted all image frames, we created standard image classification models with strong CNN variants. We selected three backbones (ResNeSt [18], RegNet [19], HRNet [20]) for expecting different characteristics. These models have better recognition performances than ResNet in ImageNet classification task, and the combination of these models have achieved top accuracy in multi-label multi-class disaster scene classification task (LADI-only section) [21].

2.3. Video Classification Models by CLIP Late Fusion Network

For leveraging text modality and large amounts of publicly available data, we used CLIP image encoders which are trained with various web image and text caption pairs using contrastive learning method [11] [22]. With CLIP image/text encoders, we first conducted a zero-shot prediction on the provided TAU dataset. As a result, even without training, CLIP models achieved strong recognition performances which are competitive with our trained audio classification models as shown in Table 2.

For boosting the CLIP-based approach, we utilized CLIP models by adding a learnable multi-branch network, which we call CLIP Late Fusion Network. The architecture of CLIP Late Fusion Network is shown in Table 1. For this multi-branch network, we extracted image features from three types CLIP image encoders and feed these features as inputs to the network. In the SI-Score paper [23] which compares various CNN/ViT/CLIP models, the authors show that each CNN/ViT/CLIP models have different characteristics. Therefore, we selected to use not a single image encoder but multiple image encoders (i.e. ResNet50x4, ResNet101, Vit-B/32) for creating more diversity in input features.

Table 1: The architecture of CLIP Late Fusion Network.

RN50x4 (dim:640)	RN101 (dim:512)	ViT-B/32 (dim:512)
Linear(640, 512)	Linear(512, 512)	Linear(512, 512)
BatchNorm1d(512)	BatchNorm1d(512)	BatchNorm1d(512)
ReLU()	ReLU()	ReLU()
Dropout(p=0.2)	Dropout(p=0.2)	Dropout(p=0.2)
Linear(512, 256)	Linear(512, 256)	Linear(512, 256)
concatenation of 256*3 dimension		
Linear(256*3, 128)		
Linear(128, 10)		

2.4. Ensemble and Post-Processing

After created audio and video classification models, we used these models to output the confidences for each defined 10 scene classes. For the validation of the official fold1 split, we firstly inferred confidences for each 10 split audio files and 12 image frame files from each 10 seconds synchronized files. For each 10 seconds file, we equally made an ensemble of the output confidences of each split audio file and image frame file. In the ensemble process, audio classification models are generally worse than video classification models (Table 2). Due to the reason, we used the good accuracy's class score (e.g. In fold1 validation, the recognition performance of A04 for tram class is competitive with that of C04/V04.) Therefore, we used only bus/park/tram classes' confidence scores and discard the other classes in ensemble. In addition, for each sample, we replaced the confidences of video models with those of audio models when the maximum confidence of 10 classes from video models is 0.20 lower than that of audio models. This method improves the recognition performance on night scenes, to which video models have low confidences due to visual difficulty, but audio models can correctly classify the class (Fig. 2).

Table 2: Summary of our created models. In the baseline system, the Audio-only model is trained with 10 sec. audio files, but we trained our audio classification models with 1 sec. audio files as test audio files are provided as 1 sec. audio files. When we train our audio classification models with 10 sec. audio files, the recognition performances are more boosted than that of 1 sec. models. About CLIP models indexed as C01-C03, we provided original labels as sentences to the CLIP text encoders and evaluated the models with the CLIP image features.

Index	Architecture	Audio	Video	Notes	Logloss	Accuracy
B01	OpenL3's model	log-mel CNN	-	Baseline model of Audio-only	1.048	65.1
A01	RegNet-6.4F	log-mel CNN	-	Training with 1 sec. audio files	0.711	76.6
A02	ResNeSt-50d	log-mel CNN	-	Training with 1 sec. audio files	0.732	76.9
A03	TF-Efficientnet-B1-NS	log-mel CNN	-	Training with 1 sec. audio files	0.821	77.2
A04	A01-A03's models	log-mel CNN	-	Ensemble of A01-A03	0.721	78.1
B02	OpenL3's model	-	CNN	Baseline model of Visual-only	1.648	64.9
V01	RegNet-6.4F	-	CNN	-	0.328	90.0
V02	ResNeSt-50d	-	CNN	-	0.367	91.7
V03	HRNet-W18	-	CNN	-	0.336	90.9
V04	V01-V03's models	-	CNN	Ensemble of V01-V03	0.316	92.4
C01	ResNet-101	-	CLIP CNN	No Training	0.671	76.7
C02	ResNet-50x4	-	CLIP CNN	No Training	0.668	74.5
C03	ViT-B/32	-	CLIP ViT	No Training	0.725	72.5
C04	C01-C03's models	-	CLIP CNN&ViT	Late Fusion of C01-C03	0.273	90.9
B03	OpenL3's model	log-mel CNN	CNN	Baseline model of Audio-Visual	0.658	77.0
E01	A04/V04/C04's models	log-mel CNN	CNN / CLIP CNN&ViT	Ensemble of A04/V04/C04	0.238	95.8
E02	**A04/V04/C04's models**	**log-mel CNN**	**CNN / CLIP CNN&ViT**	**E01 with Post-Processing**	**0.149**	**96.1**

About post-processing, we applied the below (1) for replacing each models' output confidences. The idea behind this equation is as follows: For example, in the log-loss metric, when a sample belongs to class "tram" and the output confidence of the sample for class "tram" is a too small value (e.g. 0.00001), the log-loss value for this sample becomes large (i.e. -log(0.00001) = 11.51) and it will have a large negative impact on the calculation of whole log-loss value even with a few misrecognition. Therefore, to mitigate the whole log-loss error, we avoided the extreme confidence value (e.g. x = 0 $\sim$ 0.001 and 0.99 $\sim$ 1.0) by clamping with the small offset. Also, the samples which have low or high confidence scores (e.g. x = 0.001 $\sim$ 0.06, 0.06 $\sim$ 0.20, and 0.70 $\sim$ 0.99) almost always belong to the correct class as the confidence shows, however, the complete correction of the confidence values is difficult by only model learning processes. Therefore, to suppress the whole log-loss error, we introduced this confidence calibration approach to our system as post-processing. This approach is heuristic, but it significantly improved the log-loss results on the validation dataset and test dataset provided in DCASE2021 Task1B (Table 2 and 3).

$$f(x) = \begin{cases} 0.001, & when\ \ 0 < x \leq 0.06 \\ 0.06, & when\ \ 0.06 < x \leq 0.20 \\ x, & when\ \ 0.20 < x \leq 0.70 \\ 0.99, & when\ \ 0.70 < x \leq 1.0 \end{cases} \tag{1}$$

3. EXPERIMENTS

In this section, we present our experimental setting and results for both audio and video classification models.

Experimental setting for 2.1: We created audio classification models by log-mel CNN variants under the following setting: (1) Data augmentation: Resized to 256 × 100 × 3, Random Gain, Frequency Masking [24]. We did not use Mixup [25] and Time Warping/Masking [24] for our final submissions, as these augmentations did not work in our experimental setting. (2) Train batch size: 24 (3) Epoch: 20 (Best models' epoch are around 3-5 epoch.)

Experimental setting for 2.2: We created video classification models by CNN variants under the following setting: (1) Data augmentation: Resized to 448 × 448 × 3, RandomAffine, ColorJitter, GaussianBlur, Random Erasing [26]. In RandomAffine augmentation, we set degrees as [-10, 10], translate as (0.1, 0.1), and scale as (0.5, 1.5). In GaussianBlur augmentation, we set the kernel size as (11, 11). Other augmentations' parameters are the default ones of PyTorch [27]. (2) Train batch size: 20 (3) Epoch: 20 (Best models' epoch are around 15-20 epoch.)

Experimental setting for 2.3: We created video classification models by CLIP Late Fusion Network under the following setting: (1) Data augmentation: We used extracted features from CLIP image encoders and applied no data augmentation to these features in the late fusion network. (2) Train batch size: 48 (3) Epoch: 20 (Best models' epoch are around 3-5 epoch.)

Overall setting: In the above experiments, we used SGD with Momentum method [28] as the optimizer. The learning rate is divided by 10 when the training model reached 5 epoch, 10 epoch and 15 epoch. Other hyper-parameters are the same as used in IBN-Net [29] GitHub repository*1. We trained these models with Focal Loss [30] which γ parameter is 2.0. About pre-trained models, we used ImageNet pre-trained models from timm GitHub repository*2 and CLIP pre-trained models from official CLIP GitHub repository*3.

For ensemble as noted in Table 3, we used best epoch models in the validation accuracy of each model. Instead of using Test-Time Augmentation, we extracted five images from each test video by ffmpeg, and ensembled the confidences of the five images equally for each test video. In addition, all our models are trained and tested on 1 GPU (GeForce RTX 2080Ti).

Results: Table 2 shows the results for all of our models on

*1: https://github.com/XingangPan/IBN-Net
*2: https://github.com/rwightman/pytorch-image-models
*3: https://github.com/openai/CLIP

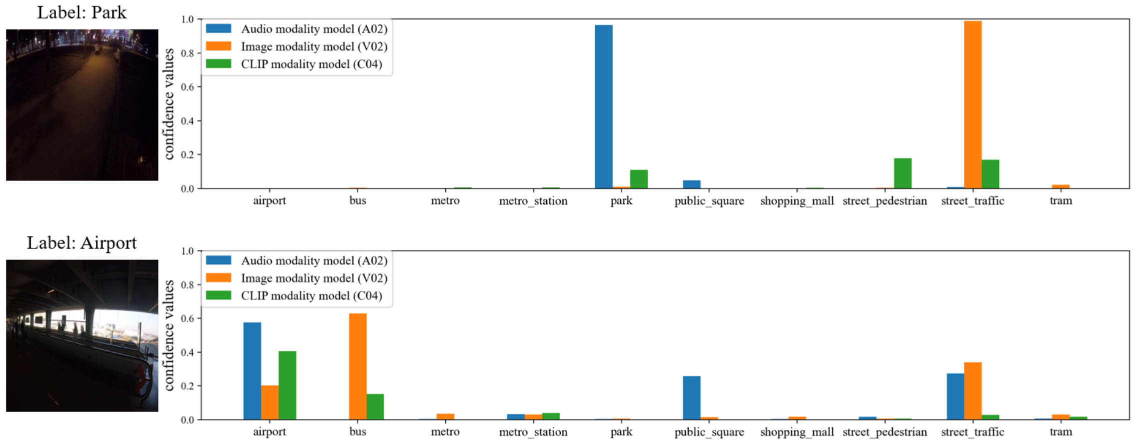

Figure 2: The sample inference results for dim scenes from validation dataset. For each scene, audio modality model (A02) can correctly classify the two dim scenes, however, image modality model (V02) failed to predict the true class. In addition, we can see that image modality model (V02) and CLIP modality model (C04) have different characteristics even trained with same image dataset of DCASE2021 Task1B.

the validation dataset. In this task, we found that CLIP models (C01-03) are competitive with the baseline models (B01-03) without training. In addition, our video classification models are much stronger than our audio classification models and we can classify 10-class scenes well only with our video classification models. In Table 3, we present the submission results for test dataset. The details are as follows: S01 consists of V04 and C04. Though C04 uses extracted features from C01-03, C04 model is constructed from one multi-branch network. Therefore, we counted the number of models in C04 as 1. S02 is the same as E02. S01-02 are trained and tested with official train/val split respectively. S03 consists of five E01 models. S04 consists of five E02 models. In Table 3, the number of models for each submission is denoted as ”Models”.

Table 3: Summary of our final submissions and the logloss scores for test dataset. p.p. in the description means post-processing which method is explained in 2.4 section. We created five train/val label files for creating S03-04 submissions. In the process of creating those five label files, we splitted the whole dataset into train and validation with keeping no overlapping about the location id.

Index	Description	Models	Logloss (Test)
S01	only-visual, 1fold	4	0.312
S02	audio-visual, 1fold, p.p.	7	0.320
S03	audio-visual, 5folds	35	0.303
S04	audio-visual, 5folds, p.p.	35	0.257

The effect of CLIP Late Fusion Network: Table 4 shows the effectiveness of our developed CLIP Late Fusion Network. Comparing the results for the case with and without CLIP, we can see that adding CLIP models can greatly improve the recognition performance of our audio-visual scene classification system, as CLIP models are created from different approach like leveraging text modality and large dataset. From model's complexity perspective, Table 5 compares the space/time complexity of CLIP Late fusion Network with other models. CLIP models have highly discriminative features, and the extracted image features from CLIP image encoders can be used directly as inputs for training shallow networks.

Table 4: The effect of CLIP Late Fusion Network (C04). With adding CLIP Late Fusion Network, the recognition performance are boosted in both logloss and accuracy metric for validation dataset.

Description	CLIP	Logloss	Accuracy
A04/V04 Fusion	no	0.293	92.4
A04/V04/C04 Fusion	**yes**	**0.238**	**95.8**
A04/V04 Fusion with p.p.	no	0.205	93.0
A04/V04/C04 Fusion with p.p.	**yes**	**0.149**	**96.1**

Table 5: The space/time complexity of CLIP Late Fusion Network (C04). Epoch means the each model's convergence epoch and Parameters means the number of learned parameters. Comparing with other models, CLIP Late Fusion Network is lightweight and can be quickly trained. The feature extraction time is not included in C04's epoch, as it can be ignored compared with training time.

Description	Epoch	Parameters	Logloss	Accuracy
V01	20	24.60M	0.328	90.0
V02	20	25.45M	0.367	91.7
V03	20	19.27M	0.336	90.9
V04	60	69.32M	0.316	92.4
C04	**5**	**1.35M**	**0.273**	**90.9**

4. CONCLUSION

In this paper, we described our approach for tackling the Task1B of the DCASE2021 challenge. We showed that by utilizing the features of CLIP variants with each audio classification models and video classification models, we can improve the recognition performance of the audio-visual scene classification task. In addition, we applied the post-processing method to the ensembled confidences, and our model achieved 0.149 log-loss (official baseline: 0.658 log-loss) and 96.1% accuracy (official baseline: 77.0% accuracy) on the officially provided fold1 validation dataset of Task1B.

5. REFERENCES

[1] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *IEEE European Conference on Computer Vision (ECCV)*, 2018.

[2] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *Neural Information Processing Systems (NeurIPS)*, 2018.

[3] M. Patrick, Y. M. Asano, R. Fong, J. F. Henriques, G. Zweig, and A. Vedaldi, "Multi-modal self-supervision from generalized data transformations," in *arXiv preprint arXiv:2003.04298*, 2020.

[4] P. Morgado, N. Vasconcelos, and I. Misra, "Audio-visual instance discrimination with cross-modal agreement," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[5] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer, "Audiovisual slowfast networks for video recognition," in *arXiv preprint arXiv:2001.08740*, 2020.

[6] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[7] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," in *International Conference on Machine Learning (ICML)*, 2021.

[8] http://dcase.community/challenge2021/.

[9] S. Wang, T. Heittola, A. Mesaros, and T. Virtanen, "Audio-visual scene classification: analysis of dcase 2021 challenge submissions," in *arXiv preprint arXiv:2105.13675*, 2021.

[10] S. Wang, A. Mesaros, T. Heittola, and T. Virtanen, "A curated dataset of urban scenes for audio-visual scene analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *arXiv preprint arXiv:2103.00020*, 2021.

[12] S. Suh, S. Park, Y. Jeong, and T. Lee, "Designing acoustic scene classification models with CNN variants," in *DCASE2020 Challenge Technical Paper*, 2020.

[13] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[14] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28. IEEE, 2020, pp. 2880–2894.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2020.

[17] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Muller, R. Manmatha, M. Li, and A. Smola, "Resnest: Split-attention networks," in *arXiv preprint arXiv:2004.08955*, 2020.

[19] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," in *arXiv preprint arXiv:1904.04514*, 2019.

[21] S. Okazaki, Q. Kong, M. Klinkigt, and T. Yoshinaga, "Hitachi at trecvid dsdi 2020," in *TRECVID Notebook Paper*, 2020.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[23] J. Yung, R. Romijnders, A. Kolesnikov, L. Beyer, J. Djolonga, N. Houlsby, S. Gelly, M. Lucic, and X. Zhai, "Si-score: An image dataset for fine-grained analysis of robustness to object location, rotation and size," in *arXiv preprint arXiv:2104.04191*, 2021.

[24] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019.

[25] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations (ICLR)*, 2018.

[26] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[28] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning (ICML)*, 2013.

[29] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," in *IEEE European Conference on Computer Vision (ECCV)*, 2018.

[30] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.