# LDSLVISION SUBMISSIONS TO DCASE'21: A MULTI-MODAL FUSION APPROACH FOR AUDIO-VISUAL SCENE CLASSIFICATION ENHANCED BY CLIP VARIANTS

Technical Report

*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 report, we describe our solution for audio-visual scene classification task of DCASE2021 challenge Task1B. Our solution is based on a multi-modal fusion approach consisting of three different domain features: (1) Log-mel spectrogram audio features extracted by CNN variants from audio files. (2) Frame-wise image features extracted by CNN variants from video files. (3) Text-guided frame-wise image features extracted by CLIP variants from video files. We trained three domain models respectively and created final submissions by ensembling the class-wise confidences of three domain models' outputs. With ensembling and post-processing for the confidences, our model reached 0.149 log-loss (official baseline: 0.658 log-loss) and 96.1% accuracy (official baseline: 77.0% accuracy) on the officially provided fold1 evaluation dataset of Task1B.

***Index Terms—*** Audio-visual Scene Classification, Multi-modal, CLIP, Convolutional Neural Network, Vision Transformer, Log-mel Spectrogram, Focal Loss, 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. This year, Detection and Classification of Acoustic Scenes and Events (DCASE) challenge 2021 [1] holds the audio-visual scene classification task as Task1B [2] 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 [3].

This report describes the details of our team's (team name: LDSLVision) 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 [4] 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 domain. After created various models, we integrated these models with ensemble methods and applied post-processing techniques 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 sec. split audio files. The test dataset is provided as 1 sec. audio files, so we didn't use original 10 sec. audio files and divided each 10 sec. audio files into ten 1 sec. 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 [5]. We used librosa library [6] 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 [7] which proposing a widely used audio classification model. In the experiments of PANNs paper, the recognition performance of log-mel spectrogram classification with ResNet-38 [8] 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 [9] with Noisy Student [10], ResNeSt [11], RegNet [12]) 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 first extracted 12 image frames from 10 sec. 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 [11], RegNet [12], HRNet [13]) which achieved 1st place recognition performance in multi-label multi-class disaster scene classification task [14].

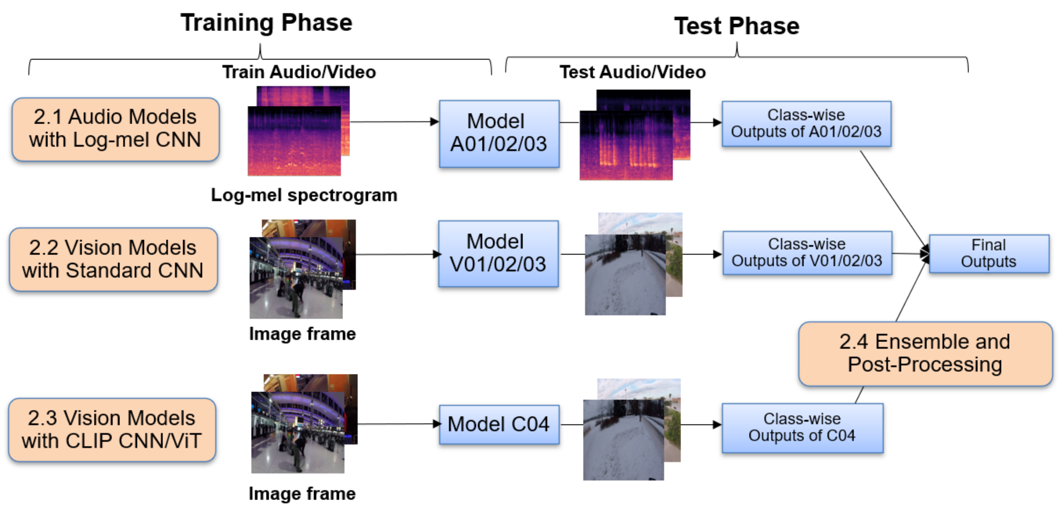
Training Phase
Test Phase
Train Audio/Video
Test Audio/Video
2.1 Audio Models with Log-mel CNN
Log-mel spectrogram
Model A01/02/03
Class-wise Outputs of A01/02/03
2.2 Vision Models with Standard CNN
Image frame
Model V01/02/03
Class-wise Outputs of V01/02/03
Final Outputs
2.4 Ensemble and Post-Processing
2.3 Vision Models with CLIP CNN/ViT
Image frame
Model C04
Class-wise Outputs of C04

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

## 2.3. Video Classification Models by CLIP Late Fusion Network

For leveraging text modality, we used CLIP image encoders which are trained with various web image and text caption pairs using contrastive learning method [4] [15]. 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 fine-tuned 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 [16] 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 class. For the evaluation of the fold1 validation dataset, we first inferred confidences for each 10 split audio files and 12 image frame files from each 10 sec. synchronized files. For each 10 sec. file, we equally ensembled 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 (see Table 2.), so we used the good accuracy's class score (e.g. In fold1 evaluation, 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.

About post-processing, we applied the below equation (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 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 introduced this confidence replacement approach. With a similar idea, we applied other three replacements described in equation (1). This approach is heuristic, but it significantly improved the log-loss results on the evaluation dataset provided as fold1_evaluate.csv (Table 2).

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

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** |

Table 3: Summary of our final submissions. p.p. in the description means post-processing which method is explained in 2.4 section.

| Index | Description | Subsystem |
|---|---|---|
| S01 | video models with 1fold | 4 |
| S02 | audio/video models with 1fold + p.p. | 7 |
| S03 | audio/video models with 5folds | 35 |
| S04 | audio/video models with 5folds + p.p. | 35 |

## 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 [17]. We didn't use Mixup [18] and Time Warping/Masking [17] for our final submissions, as these augmentations didn't 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 [19]. 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 [20]. (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.)

In the above experiments, we used SGD+Momentum 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 [21] GitHub repository*1. We trained these models with Focal Loss [22] which $\gamma$ 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, 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 the evaluation dataset provided as fold1_evaluate.csv. In this task, we found that CLIP models (C01-03) are competitive with the baseline models (B01-03) with no 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 descriptions of our final submissions. 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, so we counted C04's subsystem as 1. S02 is the same as E02. S01-02 are trained and tested with fold1_train.csv and fold1_evaluate.csv respectively. S03 consists of five E01 models. S04 consists of five E02 models. We created five label list files like fold1_train.csv and fold1_evaluate.csv, and used those label list files for creating S03-04 submissions. In the process of creating those five label list files, we splitted the whole dataset into train and validation with keeping no overlapping about the location id.

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

## 4. CONCLUSION

In this technical report, 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 evaluation dataset of Task1B.

## 5. REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.