Soft-Label: A Strategy to Expand Dataset for Large-scale Fine-grained Video Classification

Han Kong, Yubin Wu, Kang Yin, Feng Guo, Huaiqin Dong, Yulu Wang
OPPO
{konghan, wuyubin, yinkang, feng.guo, donghuaiqin, wangyulu}@oppo.com

Abstract

In this paper, the solution for The 3rd YouTube-8M Video Understanding Challenge is introduced. The final submission achieves 0.78687 MAP score in the private leaderboard, which is ranked 10th. This year's challenge is significant different from the previous ones, as the new metrics and fine-grained dataset are introduced. A series of training processes are introduced. Those processes involve proper use of the YouTube-8M coarse-grained frame-level dataset and fine-grained segment validation dataset. The results show the improvements of the performance.

1. Introduction

With the popularity of smartphone and 5G technology, the amount of videos which are watched and shared through the internet has been increased significantly. Automatic video content understanding has become a critical technique in many application scenarios, such as auto pilot, video-based search and intelligent robots etc. However, the problem of video understanding largely remains open and needs more technical breakthrough in the field of computer vision.

In this work, we focus on the video multi-label classification task presented in the 3rd YouTube-8M Challenge. Different from the previous two YouTube-8M Challenges[1, 2], the 3rd challenge focus on fine-grained video understanding by introducing human-verified segment labels. Compared to the previous video-level labels, segment labels are more fine-grained and more precise. Kaggler are encouraged to localize video-level labels to the accurate time in the video where the label actually appears. The main challengis how to use abundant of "dirty" frame-level dataset and the tiny amount of "clean" segment validation dataset to help the model learn more fine-grained features in the time dimension.

In previous challenges, many maturate video modelling methods are applied to improve the performance, such as NetVLAD[3, 4], NetFV[5, 6], None-Local[7, 8] and RNN modules like GRU[9, 10] and LSTM[11, 12, 13]. Besides, ensemble technique is widely adopted for better performance[14, 15, 16, 17]. In this challenge, the NeXtVLAD[18] is used as our basic model. The main contribution in our work is to design a series processes to utilize the datasets with different granularity to refine labels and expand dataset. During the processes, Mixup[19] and knowledge distillation [20, 21, 22, 23, 24] are applied to improve the models. Better results are achieved in the segment test dataset.

2. Dataset and Analysis

2.1. YouTube-8M Dataset

The YouTube-8M Video Understanding Challenge is held every year starting from 2017. This challenge provides a large-scale labelled video dataset containing 6.1M videos and 3862 classes. The raw videos were encoded as a sequence of feature vectors, including visual features and audio features. Both of them are produced by pre-trained convolutional neural networks using the frames extracted from video at the rate of 1Hz. In the first two YouTube-8M Challenges, there are two kinds of dataset (frame-level dataset and video-level dataset). The feature vectors in video-level dataset is produced by averaging the sequence of feature vectors in frame-level dataset. In the dataset, video labels are tagged by both automated and manual curation strategies, which lead to low accuracy of these labels. According to the technical report[1] of YouTube-8M Dataset, label precision and recall of frame-level dataset are only 78.8% and 14.5%. In this challenge,

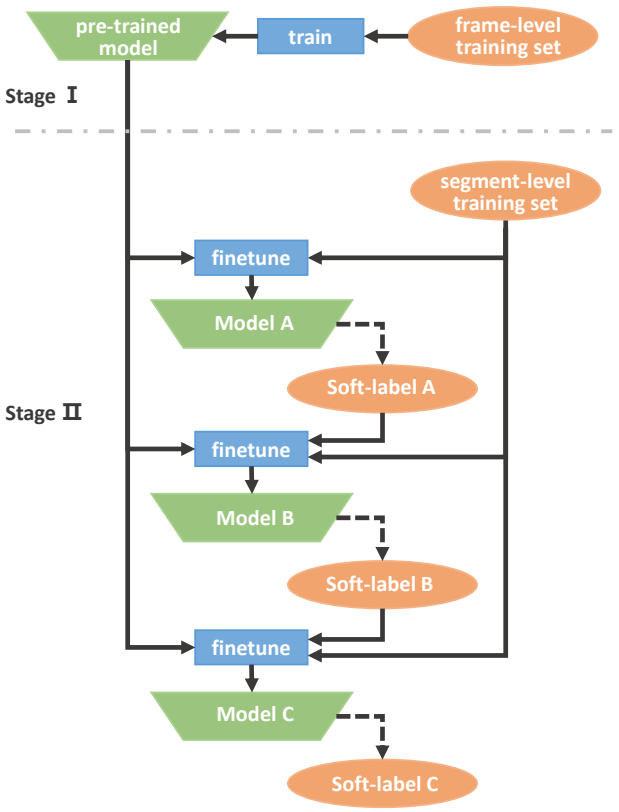

Figure 1. **Overview of model training process in Stage I and II**. Orange ellipses represent datasets, blue rectangles represent training and fine-tuning operation, green trapeziums represent the generated model of preceding operations, solid arrows indicate the inputs and outputs of each operation, dashed arrows represent the generation process of soft-label datasets.

fine-grained features in the time dimension is more valuable, so we only use the frame-level dataset.

2.2. Segment Test Dataset

The YouTube-8M Segments dataset is an extension of the YouTube-8M dataset with human-verified segment annotations. In this year, the dataset were updated to include the segment-level human-labeled ground truth for a subset of videos in the dataset. The granularity of the labeling is therefore increased from one per video, to one per 5 seconds. Each video will again come with time-localized frame-level features so classifier predictions can be made at segment-level granularity. Unlike previous challenge, the competition task will focus on temporal localization within a video.

Thus, the main focus of this year's challenge is how to leverage noisy video-level labels and a small subset of segment-level calibration set jointly in order to better annotate and temporally localize content of interest.

In the 3rd Youtube-8M video understanding challenge, submissions are evaluated according to the Mean Average Precision @ K (MAP@K), where K=100,000

$$\text{MAP@100,000} = \frac{1}{C}\sum_{c=1}^{C}\frac{\sum_{k=1}^{n} P(k)\times rel(k)}{N_c} \qquad (1)$$

where C is the number of classes, P(k) is the precision at cutoff k, n is the number of segments predicted per class, rel(k) is an indicator function. It equals 1 if the item at rank k is a relevant (correct) class, or zero otherwise. And N_c is the number of positively-labeled segments for the each class. All of the MAP scores mentioned following represent the MAP score in the private leaderboard.

In the early stage of our experiments, we used the trained model (only trained on the frame-level dataset) to test on the segment validation set and calculate the respective APs for 1000 categories. Based on the results of the AP values, combined with the analysis of the original video data on the YouTube website, we found the data of some categories has special characteristics. We chose two labels "landing" and "hunting" whose AP values are very low. After browsing a number of original training videos containing such tags, it was found that most videos with the "landing" tag were very similar. With the original training data alone, it is difficult for the model to learn the ability to locate such tags. Based on this, we think that we should fully use the data of the segment validation set.

3. The Solution

As shown in Figure 1, the training processes include three stages. More details will be introduced as following:

3.1. Stage I

In this stage, the frame-level dataset are used to pre-train the model. As we know, the max frame amount of a video in frame-level dataset is 300, but for a segment dataset it is only 5. We develop a down-sampling strategy to narrow the gap between these two dataset and keep the accuracy of the video labels in the processed video at the same time. Each batch of data in the frame-level dataset is down-sampled with a random factor from 5 to 10. After down-sampling, the single NeXtVALD model reaches a MAP score of 0.72770, which is about 0.004 higher than the score without down-sampling.

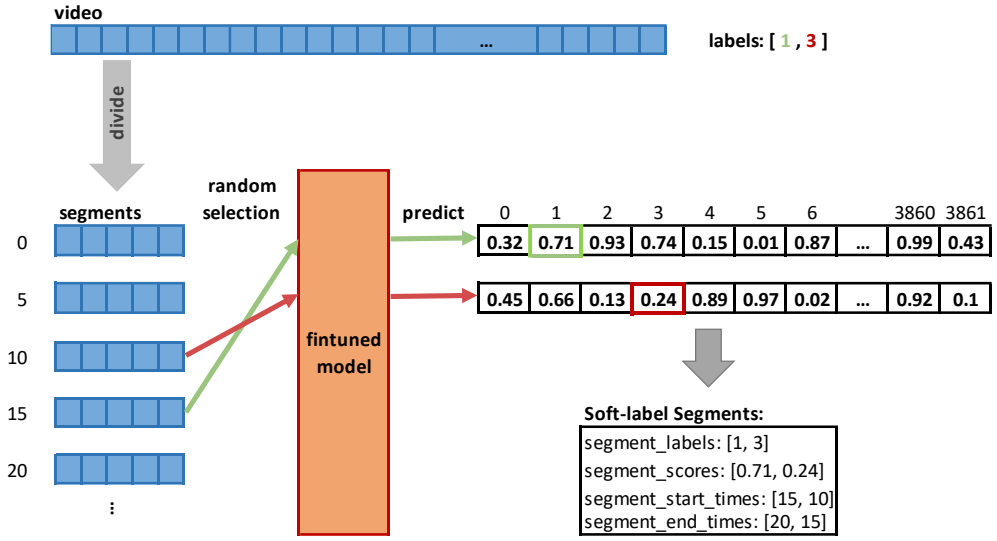

Figure 2. **Illustration of the process of generating soft-label segments (n = 1).** Each video in frame-level dataset is divided into 5-frame segments by continuous sampling, in which n * j segments are randomly selected to feed into fine-tuned model for generating two probability vectors. These predicted probabilities corresponding to the video-level labels are preserved as soft-labels.

3.2. Stage II

In this stage, the 237K segments in validation set are used to finetune the model in the previous stage. Compared with the amount of videos in frame-level dataset, the number of segments in validation dataset is much smaller and the homogeneity of these segments is obvious. There are 5 segments per video on average. Based on our observation, many segments belonging to the same video have the same segment labels and segment scores, which will increase the correlation between segments and decrease the diversity of samples in the segment dataset. In order to train the model using the segments dataset effectively and to avoid overfitting, we design the following steps.

Firstly, we directly use the segment dataset to finetune the model pre-trained by the frame-level dataset. The labeled 5-second segments are extracted from dataset as the independent videos, and the labels of these videos are set based on the segment labels and segment scores. To calculate the model loss, the loss of unrelated labels is set to zero because only the categories appeared in the segment labels of a segment are verified by human raters. Also, we use Mixup to prevent overfitting. After this step, single NeXtVLAD model gets MAP = 0.75723 in the test dataset.

Secondly, the finetuned model is used to create soft-label segments that are extracted from the frame-level dataset. In this step, each video in frame-level dataset is divided into many 5-second segments and the model makes prediction for all of these segments. To save the storage space and to make the distance between the distributions of segment validation dataset and soft-label segments in frame-level dataset closer, n segments are randomly selected for each video label in a video (in our solution, n is 2) and the probability predicted by model corresponding to the video-level labels of the entire video is preserved. So, the number of soft-label segments in a video is n * j, where j is the number of video labels in this video. For convenience, we keep the soft-label information in the form of segment dataset, and use these preserved predictions as segment scores.

An example is illustrated in Figure 2. The segments created in the frame-level dataset are called Original Soft-Label Segments dataset. Then, in the Original Soft-Label Segments, m segments are drew equally for each class in the 1000 segment classes (we use m = 1000, because in the 1k segment classes, the fewest class contains about 500 videos in the frame-level dataset, and m = 500 * n) to generate the Equal Soft-Label Segments dataset. In our experiment, the Equal Soft-Label Segments dataset is better for the following training steps. We will discuss it in details in section 4.2. The Equal Soft-Label Segments dataset and segment validation dataset generate a new model with the better performance. This process will be run twice to further improve the model accuracy. In the end, the single NeXtVLAD reachs a MAP score of 0.77457 in the private leaderboard.

Model	MAP
NeXtVLAD (Pre-trained)	0.72770
NeXtVLAD (Model A)	0.75723
NeXtVLAD (Model B)	0.76901
NeXtVLAD (Model C)	**0.77455**
MixNeXtVLAD (Pre-trained)	0.73589
MixNeXtVLAD (Model A)	0.78425
MixNeXtVLAD (Model B)	0.78585
MixNeXtVLAD (Model C)	**0.78595**
Final ensemble	**0.78687**

Table 1. **Model results**. MAP in the table represents the MAP score in the Private Leaderboard. Three MixNeXtVLADs (Model A, B and C) are used in the Final ensemble.

Method	MAP
Original Soft-Label	0.72439
Original Soft-Label + Pre-training	0.74414
Equal Soft-Label	0.76239
Equal Soft-Label + Pre-training	**0.76764**

Table 2. **Model results on different training methods**. A single NeXtVLAD are used. Pre-training means the model are pretrained by frame-level dataset firstly.

3.3. Stage III

For the final submission, MixNeXtVLAD[18] is used as the basic model. MixNeXtVLAD can be seen as the ensemble version of NeXtVLAD, which is designed to distill knowledge from a on-the-fly mixture prediction to each sub-model. It achieved the 3th place in the last YouTube-8M Challenge. The MixNeXtVLAD is used following the procedures mentioned in Stage I and II. The only difference is that both the segment validation dataset and the equal soft-label Segments dataset created by the best single NeXtVLAD are used for the first round MixNeXtVLAD fine-tune. After this Stage, the single MixNeXtVLAD model reaches 0.78595 MAP score.

Finally, we average predicted results of the three fine-tuned MixNeXtVLAD (Model A, B and C as shown in Figure 1) and achieves a MAP score of 0.78687, which reaches top10 in the final private leaderboard.

3.4. Mixup

Mixup is one kind of data augmentation techniques. It has been shown to improve the performance in many image datasets, like CIFAR-10, CIFAR100[25] and ImageNet-2012[26]. Mixup assumes that a new training sample will be obtained by combining any two training samples and their labels linearly, as shown in equation (2):

$$x = \lambda x_i + (1-\lambda)x_j \qquad (2)$$
$$y = \lambda y_i + (1-\lambda)y_j$$

in which (x_i, y_i) and (x_j, y_j) are a pair of samples extracted randomly from training set. x_i and x_j are features and y_i and y_j indicate the corresponding labels. λ is a scale factor under Beta distribution with hyperparameter α.

$$f(\lambda;\alpha,\alpha) = \frac{\lambda^{\alpha-1}(1-\lambda)^{\alpha-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\alpha-1}\mathrm{d}u} \qquad (3)$$

These "virtual" training samples expand the dataset and help avoid overfitting when the segment datasets are used to finetune the model in the Stage II and Stage III.

4. Experiments

4.1. Implementation Details

Our implementation is based on TensorFlow[27] youtube-8m Starter code[1] and the recommended hyper parameters of NeXtVLAD are used. Our NeXtVLAD models are trained on a NVIDIA V100-32GB GPU with the batch size 160. We use the Adam[28] optimizer with the initial learning rate of 2e-4 and exponentially decrease it by a factor of 0.2 every 2 million samples. In the Stage I, we set the range of random down-sampling ratio from 5 to 10. The models are trained on the training partition of frame-level datasetset for 5 epochs (about 120k steps). In the Stage II, Mixup is used in every fine-tune process. The α in Beta distribution is set as 0.4 to avoid overfitting. The initial learning rate in Stage II is changed to 2e-5 and its decrease factor is still 0.2 for every 2 million samples. Each process runs for 7 epochs using segment datasets.

For the final submission, the MixNeXtVLAD is used. Its initial learning rate is 2e-4 in Stage I and 4e-5 in Stage II. The learning rate is exponentially decreased by a factor of 0.8 every 2.5 million samples. Other configurations are the same as mentioned above.

4.2. Soft-label

Soft-label is referred from [29]. In that paper, soft-label, as a kind of recently developed knowledge distillation approach, is used to reduce the noisy video-level labels errors in the 2nd YouTube-8M Challenge. In our approach, however, soft-label is created to expand segment dataset used for fine-tune. A comparison trial shows that model

[1] https://github.com/google/youtube-8m

trained by the equal soft-label dataset improves more than the model trained by original soft-label dataset. As we known, each class in segment validation dataset has roughly the same number of videos, but in original soft-label dataset, numbers of segments contained by each class vary dramatically, which is not appropriate for the results. Also, we have tried to train the single NeXtVLAD directly by segmenting datasets (including segment validation dataset and soft-label segment dataset) without pre-trained by frame-level dataset. But its MAP is only 0.76239, which is much lower than our two stage training MAP 0.77457. The details are shown in Table 2.

5. Conclusion

In this work, we have presented our solution for the problem of large-scale and fine-grained video content understanding in the 3rd YouTube-8M Video Understanding Challenge. Our approach focuses on using large amount of coarse-grained data and small amount of fine-grained data to improve the performance of the model on fine-grained video content perception. We have shown the validity of our approach. It achieves 10th place in this challenge.

The direction of the further research includes exploring more effective and efficient methods to fuse these two stages of training by modifying model architectures and training targets. Also, the alternate iterative method between finetuning and creating soft-label should be implemented more automatically and time-saving. We have found that the AP scores of many classes which have few positive segment_scores still have poor performance after fintuning. More attention will be paid to look for root cause for this problem.

References

[1] Abu-El-Haija, Sami, et al. "Youtube-8m: A large-scale video classification benchmark." arXiv preprint arXiv:1609.08675, 2016.

[2] Lee, Joonseok, et al. "The 2nd YouTube-8M Large-Scale Video Understanding Challenge." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[3] Arandjelovic, Relja, et al. "NetVLAD: CNN architecture for weakly supervised place recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[4] Shin, Kwangsoo, et al. "Approach for video classification with multi-label on Youtube-8M dataset." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[5] Chen, Jinkun, et al. "End-to-end Language Identification using NetFV and NetVLAD." 2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP). IEEE, 2018.

[6] Araujo, Alexandre, et al. "Training compact deep learning models for video classification using circulant matrices." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[7] Wang, Xiaolong, et al. "Non-local neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[8] Tang, Yongyi, et al. "Non-local netVLAD encoding for video classification." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[9] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555, 2014.

[10] Li, Fu, et al. "Temporal modeling approaches for large-scale youtube-8m video understanding." arXiv preprint arXiv:1707.04555, 2017.

[11] Sepp, Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[12] Zou, Haosheng, et al. "The Youtube-8M kaggle competition: challenges and methods." arXiv preprint arXiv:1706.09274, 2017.

[13] Yoo, Hyeon. "Large-scale video classification guided by batch normalized LSTM translator." arXiv preprint arXiv:1707.04045, 2017.

[14] Skalic, Miha, et al. "Building a size constrained predictive model for video classification." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[15] Kim, Eun-Sol, et al. "Temporal attention mechanism with conditional inference for large-scale multi-label video classification." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[16] Miech, Antoine, et al. "Learnable pooling with context gating for video classification." arXiv preprint arXiv:1706.06905, 2017.

[17] Chen, Shaoxiang, et al. "Aggregating frame-level features for large-scale video classification." arXiv preprint arXiv:1707.00803, 2017.

[18] Lin, Rongcheng, et al. "Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[19] Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." arXiv preprint arXiv:1710.09412 (2017).

[20] Hinton, Geoffrey, et al. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531, 2015.

[21] Fukuda, Takashi, et al. "Efficient Knowledge Distillation from an Ensemble of Teachers." Interspeech. 2017.

[22] Zagoruyko, Sergey, et al. "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer." arXiv preprint arXiv:1612.03928, 2016.

[23] Heo, Byeongho, et al. "Knowledge distillation with adversarial samples supporting decision boundary." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. 2019.

[24] Yim, Junho, et al. "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[25] Torralba, Antonio, et al. "80 million tiny images: A large data set for nonparametric object and scene recognition." IEEE transactions on pattern analysis and machine intelligence, pages 1958-1970, 2008.

[26] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision, pages 211-252, 2015.
[27] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 2016
[28] Kingma, Diederik, et al. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980, 2014.
[29] Ostyakov, Pavel, et al. "Label Denoising with Large Ensembles of Heterogeneous Neural Networks." Proceedings of the European Conference on Computer Vision (ECCV). 2018.