Exploring the Consistency of Segment-level and Video-level Predictions for Improved Temporal Concept Localization in Videos

Zejia Weng, Rui Wang and Yu-Gang Jiang

{*zjweng16, ruiwang16, ygj*}*@fudan.edu.cn*

Fudan University, Shanghai, China

Abstract

Compared with the previous video-level classification, the YouTube-8M video understanding challenge of 2019 mainly focuses on temporally localizing the entities from videos. Specifically, human-verified segment-level annotations are provided for learning temporal localization models. This paper mainly introduces our system designed for the challenge. Specifically, we consider utilizing the consistency between segment&video-level predictions and ensembling different feature aggregation methods, such as variants of NetVLAD, Soft-Bag-of-Feature, Gated-Bag-of-Feature, Fisher Vector and Average Pooling. Experimental results demonstrate the effectiveness of our system on this task. Equipped with the proposed system, we achieve 0.82620 in terms of MAP@100,000, ranking 2-nd among all submissions in the challenge.

1. Introduction

Video understanding has been greatly progressed by large-scale video challenges in the past three years. As a representative, YouTube-8M challenge progresses video understanding in many different aspects. Specifically, in YouTube-8M 2017 challenge (the 1st challenge year), technologies such as context gating [10], multi-stage training [17], feature aggregation [4] were proposed for multi-label video classification. And in YouTube-8M 2018 challenge, participates adopt model compression techniques such as parameter quantization under limited model-size conditions (i.e., 1GB). Different from previous challenges, the objective of the challenge held in this year is to temporally localize the entities in the videos, aiming to achieve a finer-level understanding of the videos.

Temporal concept localization in videos, which aims to localize the video segments that contain specific entities in the video, is a challenging task that introduced in YouTube-8M Large-Scale Video Understanding Challenge of 2019. Compared to video-level classification, temporal localization brings unique challenges as it requires to understand the videos at the segment level. First, due to the higher cost of manual labeling, the segment-level labels are scarce. As shown in Figure 1, only a small set of segments in the video have been labeled. The problem of insufficient training data brings difficulties in estimating the real data distribution and avoiding model overfitting. Second, as the challenge provides two parts of labels including a large number of automatic generated noisy video-level labels and a small number of human-verified segment-level labels, how to leverage the sufficient noisy video labels and the small number of segment labels to train models for temporal localization is a difficult problem that worth exploring. Third, temporal localization is formulated as a segment classification problem and how to infer the temporal location of entities from the results of segment classification is also a challenging problem that needs to explore.

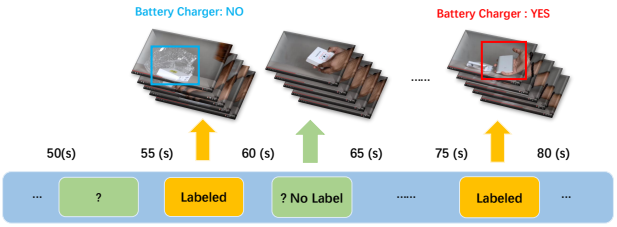

Figure 1. Examples of video segment annotations in YouTube-8M Segments dataset. For each video, a small set of segments with a fixed length of 5 seconds are randomly sampled for human annotation.

To address the aforementioned challenges, we developed a system that utilizes the consistency between segment&video-level predictions for temporal concept lo-

calization. The systems ensembles different feature aggregation methods and more importantly, a new and efficient inference strategy is proposed to transfer existing video-classification techniques in solving temporal localization problems. The proposed refinement inference strategy considers the fact that if an entity has very low confidence in appearing in a video, then it is also unlikely to appear in any of the segments in this video. We summarize important components in our system as follows:

- A mixture of different feature quantization methods (e.g., GatedDBOF, SoftD-BOF, NetVLAD, NetFV, ResNetLike, etc) is proposed to generate video/segment feature representation.

- Weighted binary cross-entropy loss is adopted to increase the influence of positive samples in segment classification.

- Global video-level prediction is employed to filter out false-positive segment predictions, boosting the performance of temporal localization.

2. Our Approach

In this section, the proposed pipeline will be elaborated. We firstly pre-train base models using all the videos, then those models are fine-tuned on all the segments using a segment-level loss. Finally, a refinement inference strategy takes in both the video-level predictions and the segment-level predictions to obtain the refined segment-level predictions. The overall solution is illustrated in Figure 2.

2.1. Base Models

As the given video data is composed of sequence features extracted from video frames, it is important to effectively aggregate the feature frames. We use various pooling methods to achieve this, including variants of NetVLAD, Bag-of-Features, Fisher Vector, Average-pooling, and Max-pooling. The following introduces the architecture of these models, which are further extended by the Mixture structure [9].

2.1.1 Single Model Architecture

Frame-level Models. Figure 3 shows our general architecture for the frame-level models. The Learnable Pooling module can be flexibly replaced with different pooling methods, including NetVLAD, DBOF, NetFV and GRU. After completing the pooling process, we drop out the compact vector and pass it through an FC and a SE context gating. Finally, we use the logistic structure as the classifier.

VLAD [3, 7] is a popular descriptor pooling method for instance-level retrieval and image classification, as it captures the statistic information about the local descriptors aggregated over the image. However, since the VLAD algorithm involves a hard cluster assignment that is non-differentiable, it is hard to apply this algorithm to neural networks. To address this problem, Arandjelovic et al. proposed NetVLAD[2] and achieved good results on a weakly-supervised place recognition task. We utilize the variants of NetVLAD. including NeXtVLAD from [9] and nonlocal-NetVLAD from [16]. Bag-of-Visual Words(BOW) [13, 14] and Fisher vector(FV) [8, 12] are two traditional aggregation methods. Inspired by the cluster soft-assignment idea of NetVLAD, similar operations are done on traditional BOF and FV, constructing the DBOF and NetFV[10]. We finally use the GatedDBOF, SoftDBOF, NetFV descriptors with the codes from [16] and [15]. Gated Recurrent Unit(GRU)[5] naturally fits to aggregate series information into a compact vector, so it is concerned as one of our pooling methods.

Video-level Models. Due to the short segment length and the low sampling rate (1 frame / 1 second), we assume that the temporal information of the segments is not so important and video-level models will be appropriate approaches.

Our video-level model is based on a ResNetLike architecture proposed in [11]. Besides using the original ResNet-Like model, we do some modifications, including tuning fully connected channels and adding extra max-pooling features, to achieve two more variants.

2.1.2 Mixture Architecture

We adopt the mixture structure [9] to strengthen the single model. The design of the mixture architecture is very subtle as it combines the idea of knowledge distillation and also uses the KL divergence to construct an extra regularization term. The on-the-fly distillation is the most attractive thing since it avoids us re-training models to do knowledge distillation learning. We apply the mixture structure to the NeXtVLAD, EarlyNetVLAD, LightNetVLAD, GatedDBOF, SoftDBOF, NetFV, GRU and three versions of ResNetLike.

2.2. Segment-level Model Fine-tuning

The temporary localization task is attributed to a very-short-video classification problem. Since segments in videos somehow can be regarded as short videos, we fine-tune the segment classification model initialized by the video-level model.

For one certain segment extracted from a video, suppose A is the set of 1000 segment-level categories, B is the annotated segment category, and C is the annotated video-level categories of the video. The Cross-Entropy function (CE), which is shown below, only concerns the annotated category of the segment and regardless of other unlabeled categories:

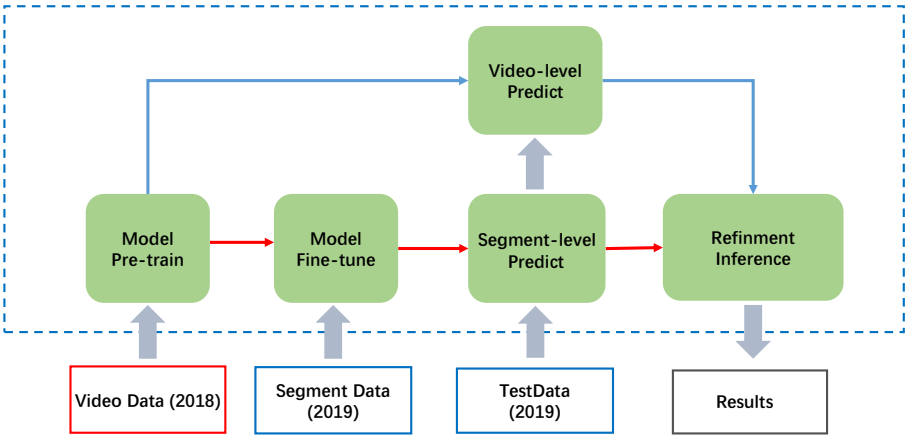

Figure 2. An overview of the proposed solution.

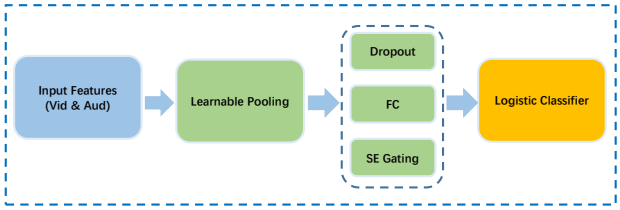

Figure 3. General architecture for the frame-level models.

$$CE(B, y^*) = -(y^* \log(p) + (1 - y^*) \log(1 - p)), \quad (1)$$

where y^* refers to the label, p is the predicted category score.

Our experiments show the above loss function results in a higher average prediction probability for the final 1000 categories, which will have a big impact on the order of the final predictions. To solve this problem, we consider adding weak supervision to constrain the predictions of negative samples. To take the unlabeled categories into consideration, we design a weighted cross-entropy loss for segment-level model fine-tuning.

As the average number of positive categories of a video is around 3, most categories in $A \backslash C$ can be weakly annotated as negative categories. We further regard all categories in $A \backslash C$ as negative categories to approximate $CE(A \backslash C)$:

$$CE(A \backslash C) = \frac{\Sigma_{a_i \in (A \backslash C)} CE(a_i, 0)}{|A \backslash C|} \quad (2)$$

The overall loss is calculate as:

$$Loss = CE(B) + \alpha * CE(A \backslash C), \quad (3)$$

where α is an adjustable parameter and we simply set it to 1.0.

2.3. Inference Strategy

Inferring temporal location of entities from the results of segment classification is an important step in our system. More importantly, as it directly affects the performance of temporal localization, the inference strategy should be carefully designed. Hence, we discuss different inference strategies including the basic inference strategy provided by the challenge as well as several improved strategies in this section.

Basic Inference Strategy. The basic version of the inference method creates 1000 minimum heaps for segment-level predictions. The predictions for each segment are pushed into the heap of respective categories. Once a heap size overflows the maximum threshold, the segment with the least predicted probability in the heap will be popped. Finally, the segment classification predictions can be converted into the final temporal localization results. However, it ignores the powerful instructions of global video information. In the following part, we will utilize video-level predictions to improve the segment-level predictions.

Refinement Inference Strategy. The proposed inference strategy is shown in Figure 4. *The main idea is: If a video contains no food, then neither do the segments within this video.*

Considering the fact that if an entity has very low confidence in appearing in a video, then it is also unlikely to appear in any of the segments in this video. Based on this consistency observation, we build a list of candidate labels for segment classification. The list of candidate labels is obtained from video labels that predicted by pretrained models, and is quite effective in removing false-positive predictions on video segments. For example, if our model

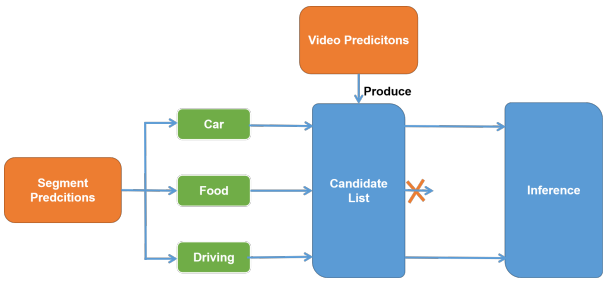

Figure 4. Overview of the proposed refinement inference strategy.

predicts that class 1,2,3,4,5 are more likely to appear in a given video, then these classes will be considered as candidate classes for video segment classification. That means, if the video segment classification model predicts the labels that are not in the candidate classes, the predicted labels will be removed. By incorporating the consistency between segment-level and video-level predictions in this way, false-positive predictions on segments can be effectively reduced.

Top rank k. Through simple statistics, we found that using the top 20 for video label predictions can have a high recall rate. Top 20 can cover 97% categories and top 100 can cover over 99% categories(top selection is done from all the 3862 category predictions instead of top 1000 in our experiments).

So our first thought is to select the top k predicting results on the video data. These selected classes consist of candidate categories, and are used to constrain the probable category scope for each segment.

Confidence threshold to constrain. The top k strategy is a good way to generate candidate lists, but there exists a little problem. Some videos contain more detailed category information, while some video content is more singular and less scene switching. For content-rich videos, it contains a large number of categories, while for a single-content video, the number of categories appears will be small. Top k ignores the diversity of categories between videos.

So the main idea of our second strategy is to consider the confidence. First, we set a threshold (such as 0.00025 in our experiment). If the prediction score is smaller than the threshold, then the related category won't be considered when predicting the inside segments.

Video number constraint for each category. The last idea is to simply limit the number of predictable videos for each category. We predict each video using a video-level model, and then limits the predicted video number by confidence for each category. Only segments in these videos will be considered as the corresponding categories.

3. Experiments

3.1. Data

The YouTube-8M Segments Dataset adopted in the 3rd YouTube-8M Video Understanding Challenge is an extension of the YouTube-8M dataset [1]. The YouTube-8M dataset contains about 6.1 million videos, 3862 class labels and 3 labels per video on average, and each video is pre-processed to extract audio/visual features at both video-level and frame-level. The YouTube-8M Segments Dataset consists of 237K human-verified segments on 1000 classes from the validation set of the YouTube-8M dataset. 5 segments per video are labeled on average and each segment is 5-seconds long.

3.2. Evaluation

In the competition, the predictions are evaluated by the Mean Average Precision (GAP) at 100,000.

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

where C is the number of classes, $P(k)$ is the precision at cutoff k, n is the number of Segments predicted per class, $r(k)$ 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. While we should predict all segments contained in the test set during inference, only human-rated segments are used in scoring and other segments that were not explicitly rated are removed from the prediction list before scoring.

In our experiments, We use MAP of the private leaderboard to evaluate our approaches.

Model	Video GAP	Segment MAP
Mix-NeXtVLAD	**0.88433**	0.7373
Mix-EarlyNetVLAD	0.88288	0.65238
Mix-LightNetVLAD	0.88142	0.69911
Mix-GatedDBOF	0.8802	0.73679
Mix-SoftDBOF	0.88071	**0.74305**
Mix-NetFV	0.88251	0.73049
Mix-GRU	0.87659	0.68541
Mix-ResNetLike	0.86499	0.71616
Mix-ResNetLike-Imbalance	0.86284	0.71958
Mix-ResNetLike-Concat	0.86288	0.72541

Table 1. Evaluation of base models.

3.3. Results

Base Models. The first step of experiments is to train models using the 2018 large-scale YouTube-8M video-level annotation data. Mixture architecture is applied to several ba-

Model Name	Fine-tune - -	Fine-tune Label Refinement -	Fine-tune Label Refinement All Data
Mix-NeXtVLAD	**0.78638**	**0.81127**	**0.81548**
Mix-EarlyNetVLAD	0.77147	0.80857	0.81212
Mix-LightNetVLAD	0.77579	0.80625	0.8093
Mix-GatedDBOF	**0.79963**	**0.81122**	**0.81327**
Mix-SoftDBOF	**0.80582**	**0.81237**	**0.81421**
Mix-NetFV	0.77949	0.80967	0.81235
Mix-GRU	0.77332	0.80436	0.8058
Mix-ResNetLike	0.7835	0.8061	0.80928
Mix-ResNetLike-Imbalance	0.78614	0.80796	0.81034
Mix-ResNetLike-Concat	0.78558	0.80825	0.81100

Table 2. Segment MAP comparison for evaluating our approaches.

sic models and ten single models of mixture structure were constructed.

The trained models were evaluated on last year's video label prediction task. As shown in the Table 1, most frame-level models with the mixture structure can achieve high scores on the leaderboard, and the simple equivalent weight ensemble of these models can reach 0.88932 on the private GAP which is a high score (although we do not consider the model size limitation).

These single models are regarded as "Base Models" and were evaluated on this year's temporal localization task. Their private MAP scores are shown in the second column. The Mixture-SoftDBOF achieves the best MAP, followed by the Mixture-GatedDBOF and the Mixture NeXtVLAD model.

Fine-tune. We fine-tune our base models on YouTube-8M Segments Dataset. We randomly sample 5/6 of the segments dataset as the training set and the remaining data as the validation set.

As we can see in Table 2, the fine-tuning operation improves our score a lot. Our best model: Mix-SoftDBOF improves nearly 0.06 and some of the other models improve even more.

We believe that the reasons for the improvement are as follows: Firstly, the video annotations contain more noise than the segment manually annotations. Secondly, there exist differences between long video classification and short video classification tasks. Therefore, the fine-tuning process is meaningful, making our models more adaptable to the new task.

Refinement Inference Strategy. In this part, we will demonstrate the efficiency of the new inference method through experiments. We used the confidence constraint method to generate a list of candidate labels, and Mix-NeXtVLAD Model was chosen to generate candidate label prediction here. Those labels whose predicted confidence is larger than 0.00025 will be added to the list of candidate labels. In the segment prediction process, we will remove the predicted labels that are not in the candidate classes.

The experiment results are shown in the Table 2. The huge improvement of the MAP illustrates the effectiveness of this approach.

Model Selection and Data Size Effects. More segments to train models can improve the final scores. However, more training data means fewer validation data for selecting models. In our work, firstly we randomly sampled 5/6 segments dataset as the training set, and the remaining 1/6 data was used as our validation set for model selection. By the MAP on the validation set, a good interval of the training steps can be found. Then we used all data for model training and use the previously recorded interval to estimate the training steps we need. Finally, we used the Stochastic Weight Averaging(SWA) [6] technique to combine those models into a single one. On the one hand, training step interval estimation improves the tolerance of the model selection. On the other hand, SWA operation improves the robustness of the models and can gain higher scores.

As shown in the Table 2, by increasing the number of trainable labels, we can improve the performance of our models a lot.

Ensemble. We finally choose Mix-NeXtVLAD, Mix-GatedDBOF, Mix-SoftDBOF, Mix-EarlyNetVLAD, and three kinds of Mix-ResNetLike Models for the final ensemble. According to [10], the ensemble does not bring much improvement when combining best but similar models, so the weight of every Mix-ResNetLike Model was reduced to 1/3. More models for ensemble and other ensemble ratios may bring some improvement to the final MAP scores.

On account of the effectiveness of the refinement inference strategy, we studied several inference strategies and found that the refinement method using the constraint of video number obtained the biggest advancement. More specifically, we picked the top 2000 videos for each category when generating the video-level predictions for refine-

ment. From results in Table 3, we observe that the new refinement approach achieves more promotion than the previous confidence constrained refinement.

Inference Strategy	Segment MAP
Origin	0.80419
Confidence Constrain: 25e-5	0.82326
Video Num Constrain: 2000	**0.8262**

Table 3. Final Ensemble Results.

4. Conclusion

In this paper, we have introduced a deep learning based system for temporally localizing the entities from videos. In our system, we consider utilizing the consistency between segment&video-level predictions and give the solving solution for this task. The proposed system firstly pretrains models on the noisy video labels, and then fine-tunes those models on the segment dataset. Finally, by combining the video&segment-level predictions, our system will give an outstanding result. Important work we do includes, applying mixture structure to different models for better robustness, using new loss function to prevent overfitting caused by the insufficient segment annotations, and exploring proper strategies for the segment predicting process. Adding all these designed strategies together, our system ranked 2nd out of 283 teams worldwide in the competition.

Acknowledgement

The authors would like to thank Jingjing Chen, Zheng Wang, Xing Zhang, Yuqian Fu, Linxi Jiang and Shaoxiang Chen for their help and suggestions.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[3] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[4] Shaoxiang Chen, Xi Wang, Yongyi Tang, Xinpeng Chen, Zuxuan Wu, and Yu-Gang Jiang. Aggregating frame-level features for large-scale video classification. *arXiv preprint arXiv:1707.00803*, 2017.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[6] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[7] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311. IEEE Computer Society, 2010.

[8] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2011.

[9] Rongcheng Lin, Jing Xiao, and Jianping Fan. Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[10] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017.

[11] Pavel Ostyakov, Elizaveta Logacheva, Roman Suvorov, Vladimir Aliev, Gleb Sterkin, Oleg Khomenko, and Sergey I Nikolenko. Label denoising with large ensembles of heterogeneous neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[12] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3384–3391. IEEE, 2010.

[13] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[14] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[15] Miha Skalic and David Austin. Building a size constrained predictive model for video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[16] Yongyi Tang, Xing Zhang, Lin Ma, Jingwen Wang, Shaoxiang Chen, and Yu-Gang Jiang. Non-local netvlad encoding for video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[17] He-Da Wang, Teng Zhang, and Ji Wu. The monkeytyping solution to the youtube-8m video understanding challenge. *arXiv preprint arXiv:1706.05150*, 2017.