A segment-level classification solution to the 3rd YouTube-8M Video Understanding Challenge

Shubin Dai (bestfitting.ai@gmail.com)

Abstract

In the 3rd YouTube-8M Video Understanding Challenge, datasets with video-level and segment-level annotation were provided for classifying video segments. Based on the winning solutions of the previous years, we extended the video-level classification to segment-level by employing techniques such as Knowledge Distillation, Exponential Moving Average to train a set of Deep Bag of Frames models. We achieved 0.795 MAP@100,000 score for our best single model and 0.817 for an ensemble of models, which leads to the 4th place on the leaderboard. In this paper we describe the details about the strategies and techniques used during the competition.

1. Introduction

With the rise of mobile platform and internet, visual understanding has becoming increasingly important for building online services for searching videos and making recommendations. Due to the sheer amount of data and computation requirement, the task remains challenging and unsolved. The YouTube-8M Video Understanding Challenge is one of the most popular challenge aims to tackle this challenge. During the first two years, the challenge has been focusing on solving video-level classification with the YouTube-8M datasets [11], and it has been successfully generated promising solutions. With the newly released YouTube-8M Segments dataset, the 3rd challenge is intended to classify videos with high temporal resolution. The model needs to predict the precise time in the video where the label actually appears. High temporal localization capability powers a whole range of applications including searching and localizing within a video and discovering interesting action moments.

1.1. YouTube-8M and YouTube-8M Segments Dataset

The YouTube-8M is a large-scale labeled video dataset that consists of more than 6.1 million of YouTube videos, with annotations from a vocabulary of 3862 unique tags. It comes with pre-computed frame representation consists of 1024 image and 128 audio features instead of raw videos.

The YouTube-8M Segments dataset is an extension of the YouTube-8M dataset with segment-level annotation. It provides ~237K video segments with human-verified labels. Each segment is 5 second long and the annotation covers 1000 classes from the validation set of the YouTube-8M dataset. About 5 segments were extracted from one video and labeled. Similarly, each segment comes with time-localized frame-level features.

There are 2,062,258 segments of 89,506 videos in the test set.

1.2. Task definition and Evaluation Metric

The YouTube-8M Video Understanding Challenge aims to solve the following task: given a query tag (e.g. one out of the 1000 classes in the dataset), the model needs to predict a list of segments ordered by their relevance to the query.

The model is then evaluated according to the Mean Average Precision (MAP) metric, specifically, the first 100,000 predictions will be evaluated with MAP@100,000 defined as:

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

Where *C* is the number of classes, *P(k)* is the precision at cutoff *k* and *n* is the number of segments predicted per class, *rel(k)* is an indicator function equal to 1 if the item at rank *k* is a relevant (correct) class, or 0 otherwise, and *Nc* is the number of positively labeled segments for each class. In this work, we used 1000 classes (*C*=1000).

2. Related work

The task is previously formulated as a classification problem, i.e., for each segment in the dataset, the model predicts a probability score for each class which will then be used to sort the segments according to the query class. Since one segment consists of a sequence of frames, the

classification model should be able to effectively aggregate the corresponding frame vectors.

During the two previous challenges, successful models were proposed to aggregate frame sequences efficiently. These models mainly fall into two categories: recurrent neural networks (RNNs) and features distribution capturing models. RNNs such as LSTM [17] and GRU [10] process the sequence in temporal fashion. By contrast, features distribution capturing models such as Deep Bag of Frames (DBoF) [11], NetVLAD [8] and FVnet [3] discard the temporal ordering and treat the sequence as a set. Besides that, context gating is often introduced to allow capture dependency between features as well as capturing prior structure of the output space to them.

In the 2nd YouTube-8M Video Understanding Challenge, Lin, Rongcheng et al. designed a model NeXtVLAD [2] which was shown to be both effective and parameter efficient in aggregating temporal information. Moreover, Knowledge Distillation with On-the-fly Naive Ensemble [5] was also shown to be useful for video classification task.

For the 3rd challenge, the competition host provided starter code to help competitors get familiar with the dataset and the competition, besides training code on video level, scripts to evaluate models on segment level labels are also provided.

3. Approach

Since the labeled video segments dataset are relatively small, we decided to fine-tune the video-level models produced by the winning teams during previous challenges. As illustrated in Fig. 1, models were trained in two stages: 1) DBoF, NetVLAD, NetVLAD_light, NextVLAD, LSTM (moe and logistic) model were trained on the YouTube-8M dataset with the video-level labels; 2) fine-tune them on segment-level labels from YouTube-8M Segments dataset.

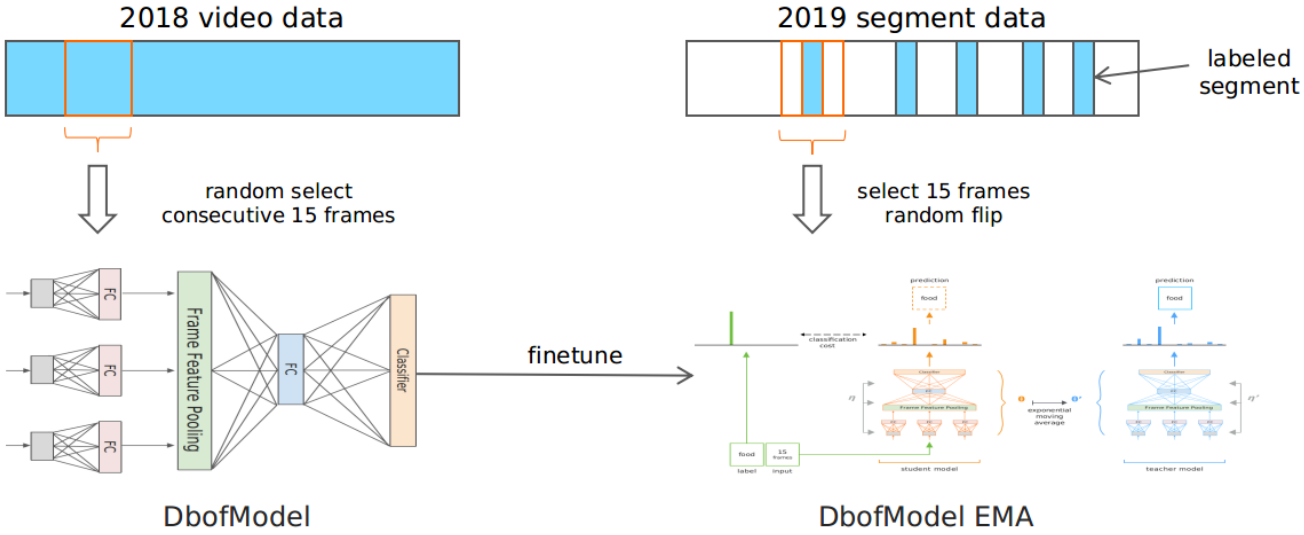

Fig.1. Overview of my network architecture for video temporal localization task, the left part is stage 1 model, it was trained on video-level frames. The right part is the stage2 model, it was fine-tuned using labeled segments and their previous and later segments as context, EMA was used.

In stage 2, we extracted n frames before and after each 5-second segment from the video to feed into the model. This allows the model to capture more context from the video for each segment which shown to be useful in our experiment. The count of context frames n was selected by experiments, and we can obtain an optimal result with n=15 for RNN models and 5 for non-RNN models. After selecting the context frame, DBoF, NetVLAD, NetVLAD_light, NextVLAD stage1 models were retrained by sample same count of frames.

Exponential Moving Average (EMA) was shown to be useful to stabilize the model weights for this type of tasks [1]. In stage 2, we used the EMA of model variables to validate the model and used in the inference phase. It works by making the models more stable and less sensitive to the iterations we chose.

We also adopted the Knowledge Distillation strategy from the winning teams in last year's competition. It was used by the 3rd place team [2], as introduced in [5], distilled knowledge from on-the-fly mixture prediction was used to teach each sub model.

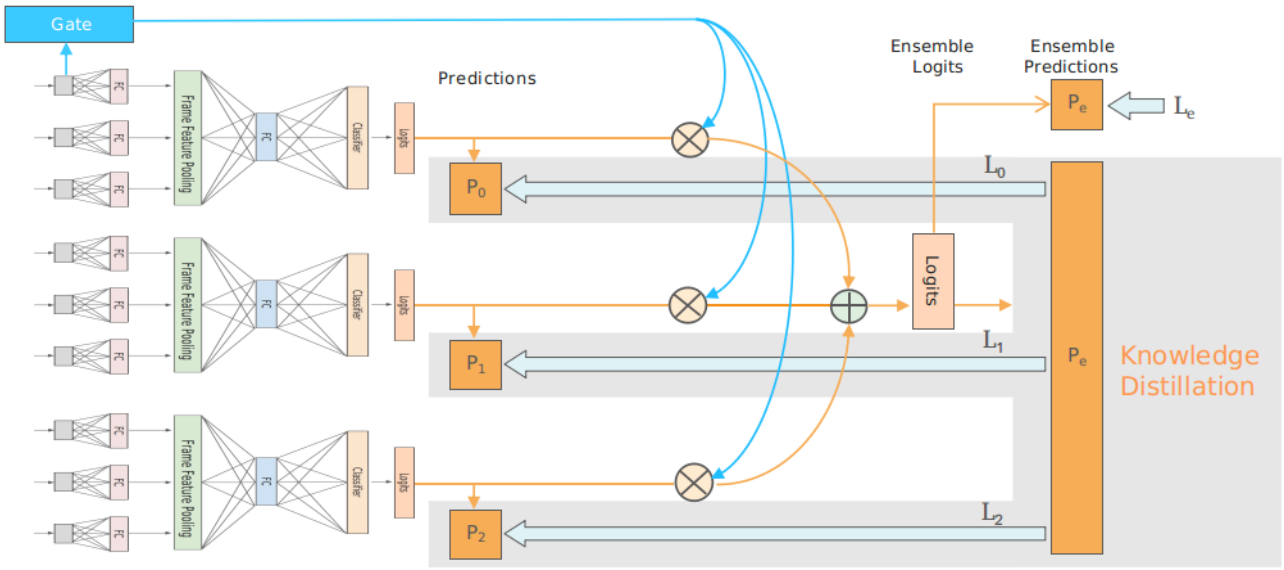

Fig.2. Overview of knowledge distillation architecture, a mixture of 3 DoF models with on-the-fly knowledge distillation.

For this dataset, semi-supervised learning techniques such as mean-teacher introduced in [23] was supposed to be promising. We therefore used the predictions of the test set to calculate a consistency cost during training. Specifically, the probabilities predicted by using current model weights was treated as student output, and the predictions by using EMA weights were treated as teacher output, the MSE loss was calculated on these two outputs (Fig. 3). Meanwhile, the flipped frames were fed into the student model to increase diversity for the inputs.

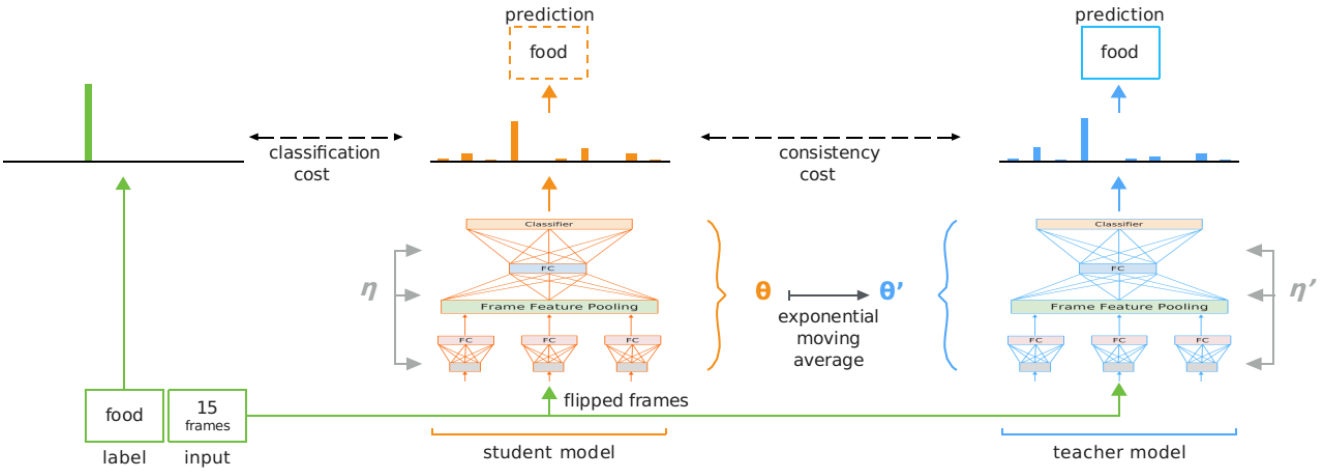

Fig.3. Overview of mean-teacher architecture, the consistency cost was calculated on test segments.

For some reason, the inference part of the starter code is inefficient on both space and computation. It takes 4-5 hours to generate a submission and hard to ensemble different models.

To improve it, we modified the code to use GPU and save the probabilities as 16-bit integer data to a file. Then the segments were sorted by their probabilities, and the top 100k segments from 2M segments was selected for each class in test set. By doing so, the inference time reduced from ~250 minutes to less than 20 minutes. This was shown to be effective for generating results for the model ensemble when evaluating the models on the leaderboard.

4. Experiments

4.1. Training Details

The segments were split into 5 folds by tfrecord files, models were trained on 4 folds and validated on 1 folds.

The implementation was built based on the TensorFlow [9] starter code and merged the 1st and the 3rd position solution to it. All the models were trained using the Adam optimizer [12], the first stage models were train on all train and validate Youbute-8M videos and their labels using the parameters from winner solutions on two Nvidia 2080TI GPUs, but when training non-RNN models, I sampled 15 frames randomly with replacement.

For stage 2, models were trained using 0.1x learning rate of the stage1, and selected the checkpoints by the MAP@100k score on validate set after 10 epochs.

4.2. Results

4.2.1 Compare different model settings

Most experiments were performed with DBoF model, as shown in Table 1. While the EMA, context-segments feeding and knowledge distillation improved the score significantly, mean-teacher model did not improve the score.

Model	Video Model Frames	Segment Model Frames	EMA	Private LB	Public LB
DBoF	30	5	False	0.750	0.758
DBoF	30	5	True	0.756	0.763
DBoF	5	5	True	0.771	0.780
DBoF	15	15	True	0.783	0.792
MixDBoF	**15**	**15**	**True**	**0.795**	**0.805**
MixDBoFMT	15	15	True	0.795	0.805

Table 1. The model performance under different settings on DBoF Model. MixDBoF model means knowledge distillation model.

4.2.2 Compare different model structures

For the model structures, we tested DBoF, NextVLAD, NetVLAD, NetVLAD_light, LSTM_moe, LSTM_logistic models from winner solutions of last 2 competitions.

As shown in Table 2, DBoF model has the best MAP score, after applying knowledge distillation on the model, average ensemble of 5 folds improved score slightly by 0.002. Presumably because it used 3 sub-models' predictions on the fly. Contrary to what we believe, adding more context frames to the LSTM model did not improve the score.

Model	Folds	Frames	Private LB	Public LB
DBoF	single fold	15	0.777	0.787
MixDBoF	single fold	15	0.791	0.803
MixDBoF	**5folds**	**15**	**0.795**	**0.805**
MixNextVLAD	5folds	15	0.795	0.801
NetVLAD_light	5folds	15	0.776	0.784
NetVLAD	5folds	15	0.771	0.781
LSTM_logistic	5folds	35	0.771	0.781
LSTM_moe	5folds	35	0.762	0.771

Table 2. The performance of different models.

5. Final Ensemble

As a common practice in a competition, ensemble a set of models can often improve the performance further. However, since more models are involved in every prediction, it is not suitable to deploy the ensemble in product environment. We therefore suggest using the knowledge distillation strategy to transfer the knowledge from the trained models to small and efficient networks.

Models with and without using context segments were trained and the predictions were weighted averaged. As time was limited, knowledge distillation strategy was used only on NextVLAD and DBoF model with context frames feeding. The weights were decided by the score of the validation set and the public test set, the score of the ensemble model on private leaderboard is 0.814.

More complex ensemble strategy which uses second-level model was designed and implemented, it's a LightGBM model which use probabilities of each model and some statistics on them as features, the target is the target score (0 or 1) of the label on every labeled segments, this means that we have 237k samples for LightGBM model, this model increased the score by 0.003, which was 0.817 on the private leaderboard.

6. Conclusion

In this paper, we presented a solution to the challenge of temporal localization on the YouTube-8M dataset. By combining context segments feeding inputs, Knowledge Distillation and EMA, the solution achieved 4th place (score: 0.817) in the 3rd YouTube-8M video understanding challenge. Detailed evaluations for different model architectures and data handling strategies were provided for balancing the speed and accuracy.

References

[1] Škalič, Miha and David Austin. “Building A Size Constrained Predictive Models for Video Classification.” ECCV Workshops (2018).

[2] Lin, Rongcheng et al. “NeXtVLAD: An Efficient Neural Network to Aggregate Frame-Level Features for Large-Scale Video Classification.” ECCV Workshops (2018).

[3] Miech, Antoine et al. “Learnable pooling with Context Gating for video classification.” ArXiv abs/1706.06905 (2017): n. pag.

[4] Hershey, Shawn et al. “CNN architectures for large-scale audio classification.” 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)(2016): 131-135.

[5] Lan, Xu et al. “Knowledge Distillation by On-the-Fly Native Ensemble.” NeurIPS (2018).

[6] Girdhar, Rohit et al. “ActionVLAD: Learning Spatio-Temporal Aggregation for Action Classification.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 3165-3174.

[7] Feichtenhofer, Christoph et al. “Convolutional Two-Stream Network Fusion for Video Action Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 1933-1941.

[8] Arandjelovic, Relja et al. “NetVLAD: CNN Architecture for Weakly Supervised Place Recognition.” CVPR (2016).

[9] Abadi, Martín et al. “TensorFlow: A System for Large-Scale Machine Learning.” OSDI(2016).

[10] Cho, Kyunghyun et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.” ArXiv abs/1406.1078 (2014): n. pag.

[11] Abu-El-Haija, Sami et al. “YouTube-8M: A Large-Scale Video Classification Benchmark.” ArXiv abs/1609.08675 (2016): n. pag.

[12] Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” CoRR abs/1412.6980 (2014): n. pag.

[13] Arandjelovic, Relja and Andrew Zisserman. “All About VLAD.” 2013 IEEE Conference on Computer Vision and Pattern Recognition (2013): 1578-1585.

[14] Deng, Jia et al. “ImageNet: A large-scale hierarchical image database.” 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009): 248-255.

[15] Szegedy, Christian et al. “Rethinking the Inception Architecture for Computer Vision.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 2818-2826.

[16] Jégou, Hervé et al. “Aggregating local descriptors into a compact image representation.” 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010): 3304-3311.

[17] Hochreiter, Sepp and Jürgen Schmidhuber. “Long Short-Term Memory.” Neural Computation 9 (1997): 1735-1780.

[18] Wang, He-Da et al. “The Monkeytyping Solution to the YouTube-8M Video Understanding Challenge.” ArXiv abs/1706.05150 (2017): n. pag.

[19] Chen, S., Wang, X., Tang, Y., Chen, X., Wu, Z., & Jiang, Y. (2017). Aggregating Frame-level Features for Large-Scale Video Classification. ArXiv, abs/1707.00803.

[20] Škalič, Miha et al. “Deep Learning Methods for Efficient Large Scale Video Labeling.” ArXiv abs/1706.04572 (2017): n. pag.

[21] Li, Fu et al. “Temporal Modeling Approaches for Large-scale Youtube-8M Video Understanding.” ArXiv abs/1707.04555 (2017): n. pag.

[22] Fernando, Basura et al. “Modeling video evolution for action recognition.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 5378-5387.

[23] Tarvainen, Antti and Harri Valpola. “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results.” ICLR (2017).