The 2nd YouTube-8M Large-Scale Video Understanding Challenge

Joonseok Lee, Apostol (Paul) Natsev,
Walter Reade, Rahul Sukthankar, and George Toderici

Google Research, Mountain View, USA
{joonseok,natsev,inversion,sukthankar,gtoderici}@google.com

Abstract. We hosted the 2nd YouTube-8M Large-Scale Video Understanding Kaggle Challenge and Workshop at ECCV'18, with the task of classifying videos from frame-level and video-level audio-visual features. In this year's challenge, we restricted the final model size to 1GB or less, encouraging participants to explore representation learning or better architecture, instead of heavy ensembles of multiple models. In this paper, we briefly introduce the YouTube-8M dataset and challenge task, followed by participants statistics and result analysis. We summarize proposed ideas by participants, including architectures, temporal aggregation methods, ensembling and distillation, data augmentation, and more.

Keywords: YouTube · Video Classification · Video Understanding

1 YouTube-8M Dataset

Many recent breakthroughs in machine learning and machine perception have come from the availability of large labeled datasets, such as ImageNet [8], which has millions of images labeled with thousands of classes. Their availability has significantly accelerated research in image understanding.

Video provides even more information for detecting and recognizing objects, and understanding human actions and interactions with the world. Improving video understanding can lead to better video search, organization, and discovery, for personal memories, enterprise video archives, and public video collections. However, one of the key bottlenecks for further advancements in this area, until recently, has been the lack of labeled video datasets with the same scale and diversity as image datasets.

Recently, Google announced the release of YouTube-8M [1], a dataset of 6.1+ million YouTube video URLs (representing over 350,000 hours of video), along with video-level labels from a diverse set of 3,862 Knowledge Graph entities.[1] This represents a significant increase in scale and diversity compared to existing video datasets. For example, Sports-1M [11], the previous largest labeled video

[1] This statistics is based on the most recent dataset update on May 14, 2018.

dataset we are aware of, has around 1 million YouTube videos and 500 sports-specific classes–YouTube-8M represents nearly an order of magnitude increase in both number of videos and classes.

YouTube-8M represents a cross-section of our society. It was designed with scale and diversity in mind so that whatever lessons we learn on this dataset can transfer to all areas of our lives, from learning, to communication, and entertainment. It covers over 20 broad domains of video content, including entertainment, sports, commerce, hobbies, science, news, jobs & education, health.

The dataset comes with pre-computed state-of-the-art audio-visual features from billions of frames and audio segments, designed to fit on a single hard disk. This makes it possible to get started on this dataset by training a baseline video model in less than a day on a single machine. Considering the fact that this dataset spans over 450 hours of video, training from the raw videos is impractical–it would require 1 petabyte of raw video storage, plus video decoding and training pipelines that run 20,000 times faster than real time video processing in order to do one pass over the data per day. In contrast, by standardizing and pre-extracting the frame-level features, it is possible to fit the dataset on a single commodity hard drive, and train a baseline model to convergence in less than a day on 1 GPU. Also, by standardizing the frame-level vision features, we focus the challenge on video-level temporal modeling and representation learning approaches. The annotations on the training videos are machine-generated from different sources of information and are somewhat noisy and incomplete. A key research angle of the challenge is to design systems that are resilient in presence of noise.

2 Challenge Task

Continuation from the First Google Cloud & YouTube-8M Video Classification Challenge on Kaggle [2] and CVPR'17 Workshop [3], we hosted the challenge on Kaggle [4] with the revised video classification task, described in this section.

Participants are asked to produce up to 20 video-level predictions over the 3,862 classes on the YouTube-8M (blind) test set. The training and validation sets are publicly available, along with 1024-D frame-level and video-level visual features, 128-D audio features, and (on average) 3.0 video-level labels per video. The challenge requires classifying the blind test set of ∼700K videos, labels for which ground truth labels have been withheld. This blind test set is divided into two same-sized partitions, called public and private. For each submission, we evaluate performance on the public portion of the test set and release this score to all participants. Another half is used for final evaluation. Award and final ranking is determined based on this private test set, and this score is visible upon completion of the competition. Participants do not know which examples belong to which set, so they are asked to submit answers to the entire test set.

[2] https://www.kaggle.com/c/youtube8m
[3] https://research.google.com/youtube8m/workshop2017/index.html
[4] https://www.kaggle.com/c/youtube8m-2018

Formally, for each video v, we have a set of ground-truth labels G_v. Participants produce up to 20 pairs $(e_{v,k}, f_{v,k})$ for each video v, where $e_{v,k} \in E$ is the class label, and $f_{v,k} \in [0, 1]$ is its confidence score. We bucket $f_{v,k}$ with $\tau_j = j/10000$, where $j \in \{0, 1, ...10000\}$, and compute the Global Average Precision (GAP) across all classes as follows:

$$P(\tau) = \frac{\sum_{v \in V} \sum_{k=1}^{20} I(f_{v,k} \geq \tau) I(e_{v,k} \in G_v)}{\sum_{v \in V} \sum_{k=1}^{20} I(f_{v,k} \geq \tau)} \tag{1}$$

$$R(\tau) = \frac{\sum_{v \in V} \sum_{k=1}^{20} I(f_{v,k} \geq \tau) I(e_{v,k} \in G_v)}{\sum_{v \in V} |G_v|} \tag{2}$$

$$GAP = \sum_{j=1}^{10000} P(\tau_j) \left[R(\tau_{j-1}) - R(\tau_j) \right] \tag{3}$$

Performance is measured using this GAP score of all the predictions and the winner and runners-up of the challenge are selected based on this score. Note that this metric is optimized for systems with proper score calibration across videos and across entities.

Rank	Team Name	Best Performance (GAP)		# Models in Ensemble
		Single Model	Ensembled	
1	WILLOW [18]	0.8300	0.8496	25
2	monkeytyping [24]	0.8179	0.8458	74
3	offline [15]	0.8275	0.8454	57
4	FDT [6]	0.8178	0.8419	38
5	You8M [22]	0.8225	0.8418	33
6	Rankyou [25]	0.8246	0.8408	22
7	Yeti [5]	0.8254	0.8396	21
8	SNUVL X SKT [19]	0.8200	0.8389	22
9	Lanzan Ramen	–	0.8372	–
10	Samartian [26]	0.8139	0.8366	36

Table 1. Best performance achieved by top 10 teams from 2017 YouTube-8M Challenge, with number of ensembled models.

In addition, we restrict the model size up to 1 GB without compression. This is to discourage participants to try extremely heavy ensemble models, which we observed at the last year's competition (and other Kaggle challenges as well). Table 1 shows the best GAP scores achieved by the top 10 performers, with and without ensembles. We clearly see that most top performers ensembled tens of models and get consistent improvement about $2 \sim 3\%$ on GAP. In large-scale applications, it is practical to limit the size of vectors representing videos due to CPU, memory, or storage considerations. We focus our challenge on developing models under a fixed feature size budget, encouraging participants to focus on

developing novel single-model architectures, or multi-modal ensembles trained jointly end-to-end, as opposed to training many (in some cases, thousands of) models independently and doing a late-stage model ensemble to squeeze the last 1% of performance. The latter makes the approach infeasible for any real applications, and makes it difficult to compare single-model architectures fairly, as top performance is typically achieved by brute force approaches. This also gives an unfair edge to competitors with large compute resources, as they can afford to train and ensemble the most number of models.

3 Challenge Result

In this section, we review some stats regarding participants, and present the final leaderboard with GAP as well as some other useful metrics to compare performance.

3.1 Participants Overview

This year, total 394 teams participated in the competition, composed of 531 total competitors. For 106 participants among these, the 2nd YouTube-8M competition was their first competition at Kaggle. 61 participants have participated in the First YouTube-8M competition and returned to 2018 competition. Participants come from 40 and more countries, summarized in Table 2. This is based on the IP address where each participant created the account, so this is just an approximate statistics.

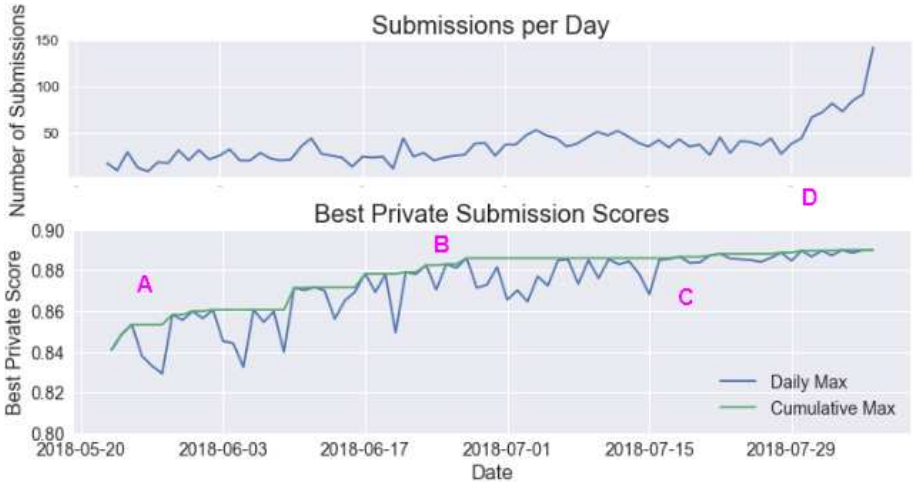

Fig. 1. The number of submissions (top) and best private leaderboard scores on each day and cumulatively (bottom).

Country	# Competitors	Award Winners
United States	136	1st*, 3rd
P. R. China	69	4th
India	56	
Russia	30	2nd
Korea	25	5th
Japan	19	
France	15	
Canada	15	
United Kingdom	14	
Taiwan	10	
Singapore	9	
Hong Kong	9	
Belarus	8	
Ukraine	8	
Germany	7	
Poland	6	
Australia	5	
Greece	4	

Table 2. Number of participants by country. (*This team is multinational, with a Spanish and an American participant.)

In this year's competition, we received total 3,805 submissions. This is about 10 submissions per team on average, which is relatively lower than usual. Median number of competition is 15. Figure 1 shows overall trend of competition progress. We launched the competition on May 22, 2018. Early on in the competition (**A** in Figure 1), we observe a rapid increase in the best score (green). We also see lots of variability in the best daily scores (blue). This suggests participants were trying a wide variety of different ideas. About mid-way through the competition, around point **B**, the best score starts to plateau, but we still see lots of day-to-day variability, indicating continued exploration of techniques. During the last third of the competition (**C**), the day-to-day variability decreases significantly, suggesting competitors were trying to fine-tune submissions. We observe a sharp increase in model submissions towards the end of the competition (**D**), where participants were trying to get final incremental improvements in their submissions.

Another interesting analysis is about how returning participants performed. Figure 2 shows relative rank change (in percentile) of all returning teams. Red arrows mean moved down, while green ones mean moved up. Group **A**, the teams from lowest ranks last year, showed modest progress up to the 50–70 percentile. Group **B** from the middle last year showed significant improvement to within top 20%. Group **C** slipped a bit from top 10% to 20%. Among the top performers in Group **D**, we observe two patterns. Some of them dropped a lot, probably putting little effort on this year's competition. On the other hand, some teams

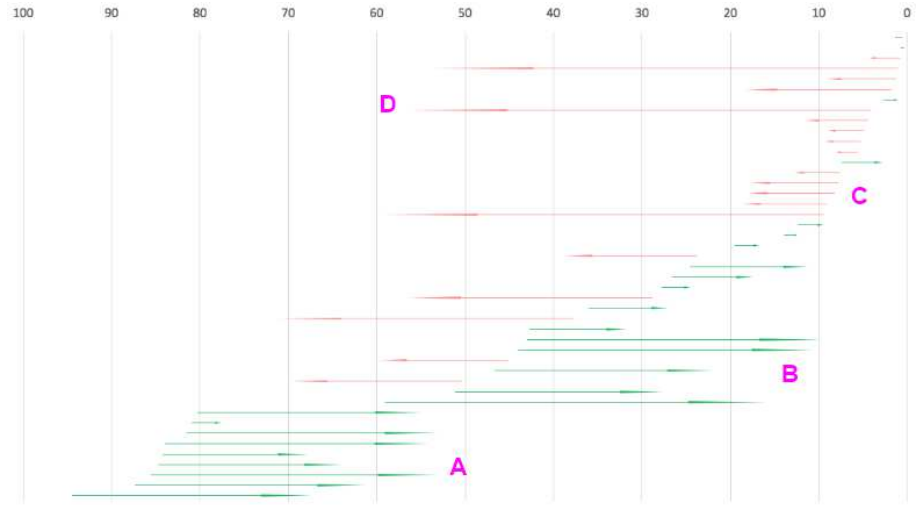

Fig. 2. The number of submissions (top) and best private leaderboard scores on each day and cumulatively (bottom).

achieved the top ranks again, including Team Next top GB model from 5th to 1st and Team YT8M-T staying at 4th again.

3.2 Final Leaderboard

Table 3 shows the final leaderboard, sorted by our official evaluation metric, GAP score. We also evaluate submitted final models in terms of the following additional metrics:

– **Mean Average Precision (MAP)**: In practice, examples are not uniformly distributed over labels. For some labels, we have a plethora of training examples, while for some other labels, we have just a few of them. This is the case for YouTube-8M as well. Thus, we are also interested in a metric that deals with each label equally. Instead of computing AUC of overall precision-recall curve, MAP computes mean per-class AUC of precision-recall curves. Formally,

$$P_e(\tau) = \frac{\sum_{v \in V} \sum_{k=1}^{20} I(f_{v,k} \geq \tau) I(e_{v,k} \in G_v) I(e_{v,k} = e)}{\sum_{v \in V} \sum_{k=1}^{20} I(f_{v,k} \geq \tau) I(e_{v,k} = e)} \tag{4}$$

$$R(\tau) = \frac{\sum_{v \in V} \sum_{k=1}^{20} I(f_{v,k} \geq \tau) I(e_{v,k} \in G_v) I(e_{v,k} = e)}{|v : e \in G_v|} \tag{5}$$

$$AP_e = \sum_{j=1}^{10000} P_e(\tau_j) \left[R_e(\tau_{j-1}) - R_e(\tau_j) \right] \tag{6}$$

$$MAP = \frac{1}{|E|} \sum_{e \in E} AP_e \tag{7}$$

- **Hit@k** is the fraction of test samples that contain at least one of the ground truth labels in the top k predictions. We measure and report Hit@1 of the top performers.
- **Precision at Equal Recall Rate (PERR)** is similar to MAP, but instead of using a fixed $k = 20$, we compute the mean precision up to the number of ground truth labels in each class. Formally,

$$PERR = \frac{1}{|V|} \sum_{v \in V} \left[\frac{1}{|G_v|} \sum_{k \in G_v} I(rank_{v,k} \leq |G_v|) \right], \tag{8}$$

  where G_v is the set of ground truth labels for video v and $I(rank_{v,k} < |G_v|)$ counts the number of correct predictions made within the top $|G_v|$.

For all metrics, higher values indicate better performance. We measure these metrics based on the final submission, although other intermediate models might have achieved higher scores in other metrics.

Rank	Team Name	GAP	MAP	Hit@1	PERR	Model Size
1	Next top GB model [21]	**0.88987**	0.59637	0.9074	**0.8311**	1,010MB
2	Samsung AI Center Moscow [2]	0.88729	0.58436	**0.9075**	0.8297	943MB
3	PhoenixLin [16]	0.88722	**0.59682**	0.9074	0.8310	901MB
4	YT8M-T [23]	0.88704	0.58794	0.9059	0.8283	923MB
5	KANU [12]	0.88527	0.58300	0.9039	0.8260	964MB
6	[ods.ai] Evgeny Semyonov	0.88506	0.58476	0.9057	0.8274	982MB
7	Liu [17]	0.88324	0.58194	0.9030	0.8242	1,020MB
8	Sergey Zhitansky	0.88113	0.50362	0.8861	0.7977	844MB
9	404 not found	0.88067	0.49868	0.8842	0.7947	**682MB**
10	Licio.JL	0.88027	–	–	–	817MB
11	Weimin Wang	0.88012	0.56076	0.9006	0.8197	1,021MB
12	IIAI	0.87912	–	–	–	–
13	NPhard	0.87796	0.55979	0.8992	0.8178	753MB
14	CV_Group	0.87662	0.53946	0.8925	0.8067	964MB
15	NII	0.87465	0.54857	0.8968	0.8143	938MB
16	DeepCats	0.87342	0.55275	0.8955	0.8122	970MB
17	Axon AI [7]	0.87287	0.46735	0.8614	0.7608	971MB
18	Steeve Huang	0.87216	0.55425	0.8962	0.8133	880MB
19	running out of time	0.87190	–	–	–	992MB
20	Newers	0.87186	–	–	–	–

Table 3. Final leaderboard with 20 top performers in GAP, listed with other metrics (MAP, Hit@1, and PERR). The top performers in each metric are marked **bold**.

Table 3 indicates that Next top GB model team [21] achieved the best GAP as well as PERR scores. Samsung AI Center Moscow team [2] achieved the best

Hit@1 score, and PhoenixLin [16] did for MAP score. The smallest model we see from the top 20 is achieved by the team 404 not found, which is 682MB. Note that the GAP metrics in Table 3 are not compatible to those in Table 1 from 2017 leaderboard, as the a different version of YouTube-8M dataset was used each year.

4 Approaches

In this section, we briefly review commonly used techniques for this year's competition, as well as some interesting ideas proposed by participants.

4.1 Architecture

Many participants [16, 13, 17, 4, 20] built their models based on the WILLOW architecture illustrated in Figure 3, designed by the WILLOW team, the 2017 competition winner [18]. In this work, team WILLOW explored combinations of learnable pooling techniques such as Soft Bag-of-words, Fisher Vectors, NetVLAD, GRU, and LSTM to aggregate video features over time. Also, they introduced a learnable non-linear network unit, called Context Gating, aiming at modeling inter-dependencies between features. For other architectures, Samsung AI Team [2] uses ResNet-based model.

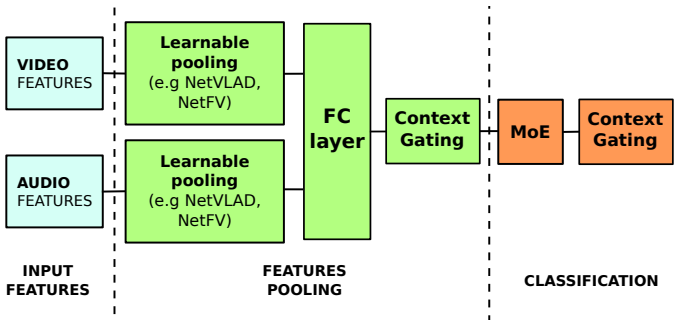

Fig. 3. The WILLOW architecture [18]

4.2 Temporal Aggregation

As the dataset provides frame-level features while the task requires video-level label estimation, a lot of participants propose novel ideas of how to aggregate frame-level features into video-level.

The most widely-used approach within this year's competition is NetVLAD [3] and its variants. Team PhoenixLin [16] substitutes the NetVLAD part with its

variant NeXtVLAD. Team Deep Topology [13] applies modified NetVLAD to consider cluster similarity as well. They also try attention-enhanced NetVLAD, with transformer block inserted before and after NetVLAD module. Team YT8M-T [23] proposes non-local NetVLAD, modeling the relations between different local cluster centers. They try variants with early and late fusions of NetVLAD and its variants.

Another popular approach is recurrent neural networks, LSTM and GRU. Samsung AI Moscow Team [2] and Shivam Garg [9] apply uni-directional and bi-directional LSTM to aggregate frame-level features. Team Axon AI [7] uses both LSTM and GRU.

Convolution on temporal axis is another popular way to substitute recurrent neural nets. Shivam Garg [9] proposes ResidualCNN-X, where X is the output size, composed of a fully connected layer and a deep CNN network with several residual modules. Samsung AI team [2] tries time-distributed convolutional layers, containing several layers of convolutions followed by max-pooling for video and audio separately, then concatenating the resulting features.

Team KANU [12] selects informative frames using spatio-temporal attention model; temporal attention on audio guided by image, and temporal attention on image guided by audio.

Lastly, Deep Topology [13] proposes Multi-modal Factorized Bi-linear (MFB) pooling approach. For a video feature $\mathbf{v}$ and an audio feature $\mathbf{a}$, the MFB vector f_i is defined as weighted sum of elements in outer products $\mathbf{v}\mathbf{W}_i\mathbf{a}^\top$, where the weight matrix $\mathbf{W}_i$ is a low-rank bi-linear model. They combine MFB with different video-level features and explore its effectiveness in video classification.

4.3 Ensembles

Top performers are still taking advantage of ensembling, as listed in Table 4. However, the number of combined models has dropped from previous year, with an exception of Samsung team who ensembled 115 (95 video-level and 20 frame-level) models. Most other teams (among those who submitted a paper) ensembled less than 10 models. On average, top performers take advantage of ensembels to improve their final performance by $\sim 2.5\%$. Most teams ensembled different models or same models with different hyper-parameters. Liu et al. [17] proposed ensembling different checkpoints from the same model with same hyper-parameters.

4.4 Techniques for a Compact Model

Due to the model size limit, many teams propose ideas to make the model compact. The most popular approach among participants is knowledge distillation [10], transferring the generalization ability of a huge teacher model (usually ensembles of multiple models in this competition) to a relatively simpler student network by using prediction from the teacher model as an additional soft target during training. Team PhoenixLin [16] distills from 3 NeXtVLAD models. Samsung AI team [2] distills from ensembles of 95 video-level models and 20

Rank	Team Name	Best Performance (GAP)		# Models in Ensemble
		Single Model	Ensembled	
1	Next top GB model [21]	0.87237	0.88987	15
2	Samsung AI Center Moscow [2]	0.87417	0.88729	115
3	PhoenixLin [16]	**0.87846**	0.88722	3
4	YT8M-T [23]	0.87030	0.88704	6
5	KANU [12]	0.86078	0.88527	6
7	Liu [17]	0.87440	0.88324	4
17	Axon AI [7]	0.85750	0.87287	7

Table 4. The number of ensembled models by top performers.

frame-level models. The winner, Next top GB model team, designs two-level distillation on combination of ground truth and predicted labels by teacher models. Axon AI team [7] also applies similar idea to use convex combination of distilling ground truth and teacher model.

Another approach that most teams use is quantization. It is known that full float precision may not be necessary to represent a video [14]. Thus, we can almost preserve the end-to-end accuracy with using less number of bytes to represent values. In other words, increasing the number of dimensions with fewer bytes is more efficient use of space. Most teams use `float16` instead of full precision.

4.5 Other Interesting Ideas

We briefly introduce other interesting approaches proposed by participants. Some of these approaches have not been proved to be superior than the top performing models within the competition, but many of these are indeed novel and worth to explore further.

- **Label Correlation**: Team KANU [12] proposes conditional inference using label dependency for multi-label classification. Assuming $p(y|\mathbf{x})$ can be factorized as $\prod_{i=1}^{q} f_i(\mathbf{x}, y_{<\phi_i})$, they proposed a stage-wise algorithm to find positive labels in a greedy manner. Axon AI [7] proposes an additional regularized term $\mathbf{tr}(\mathbf{W}_{L-1}\Omega^{-1}\mathbf{W}_{L-1}^{\top})$ to guide related labels to have similar estimation, where $\mathbf{W}_{L-1}$ is the last layer's weights. The Ω encodes label relationship, which is driven from the data as well. Team sogang-mm [20] studies imbalance of label distribution, splitting the dataset into two, a fine-grained subset with rare labels and the rest with common labels. They compare training on one and re-training on the other, but conclude that there is no significant difference.
- **Data Augmentation**: Dataset augmentation is an effective way to increase data samples for training, usually by adding noise into existing examples. Team Axon AI [7] proposes generating virtual training data points by interpolating or extrapolating video features from K-nearest neighbors. Training

with oversampled dataset in this way shows consistent improvement on performance. Samsung AI [2] also uses similar idea, creating virtual training examples by convex combination of existing ones.

- **Reverse Whitening**: Team PhoenixLin [16] reports that reversing the whitening process, which is applied after dimension reduction by PCA of frame-level features, is beneficial for NeXtVLAD model to generalize better. They argue that whitening after PCA might distort the feature space by eliminating different contributions between feature dimensions with regard to distance measurements, which could be critical for the encoder to find better anchor points and soft assignments for each input feature.
- **Hierarchical Relationship between Frames**: Deep Topology [13] represents a video as a graph with frames as nodes and relationship between frames as edges, and applies Graph Convolutional Networks on it. Their graph is constructed in a hierarchical manner, starting from frame-level, simplified into shot-level, event-level, and final video-level embedding in a row.
- **Circlant Matrix**: Given success of model distillation and compression approaches (Section 4.4), Team Alexandre Araujo [4] poses a question if it is possible to devise models that are compact by nature while exhibiting the same generalization properties as large ones. They propose replacing unstructured weight matrices with structured circulant matrices $\mathbf{C} \in R^{n \times n}$, which can be defined with a single vector of size n, and demonstrate to build a compact video classification model based on them.

5 Summary

We hosted the First (2017) and Second (2018) YouTube-8M Large-Scale Video Understanding Kaggle Challenge and Workshop at CVPR'17 and ECCV'18, respectively. With two runs of this competition, researchers indeed proposed interesting working ideas on architecture, temporal aggregation, ensembling, label correlation, data augmentation, and more. Most top performers heavily ensembled tens of models to maximize the Global Average Precision, and distilled it into a smaller model that fits into the size limit (1GB). Some participants proposed interesting novel ideas to tackle this problem, although they did not outperform the ensembled models. We will continue to host this challenge and workshop to advance research in video understanding, possibly with updated dataset, new features, on diverse metrics or tasks.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8M: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)

2. Aliev, V., Ostyakov, P., Suvorov, R., Sterkin, G., Logacheva, E., Khomenko, O., Nikolenko, S.: Label denoising with large ensembles of heterogeneous neural networks. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
3. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
4. Araujo, A., Negrevergne, B., Chevaleyre, Y., Atif, J.: Training compact deep learning models for video classification using circulant matrices. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
5. Bober-Irizar, M., Husain, S., Ong, E.J., Bober, M.: Cultivating dnn diversity for large scale video labelling. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)
6. Chen, S., Wang, X., Tang, Y., Chen, X., Wu, Z., Jiang, Y.G.: Aggregating frame-level features for large-scale video classification. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)
7. Cho, C., Antin, B., Arora, S., Ashrafi, S., Duan, P., Huynh, D.T., James, L., Nguyen, H.T., Solgi, M., Than, C.V.: Axon AI's solution to the 2nd Youtube-8M video understanding challenge. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (2009)
9. Garg, S.: Learning video features for multi-label classification. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
11. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proc. of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR) (2014)
12. Kim, E.S., Kim, J., On, K.W., Heo, Y.J., Choi, S.H., Lee, H.D., Zhang, B.T.: Temporal attention mechanism with conditional inference for large-scale multi-label video classification. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
13. Kmiec, S., Bae, J.: Learnable pooling methods for video classification. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
14. Lee, J., Abu-El-Haija, S., Varadarajan, B., Natsev, A.: Collaborative deep metric learning for video understanding. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2018)
15. Li, F., Gan, C., Liu, X., Bian, Y., Long, X., Li, Y., Li, Z., Zhou, J., Wen, S.: Temporal modeling approaches for large-scale Youtube-8M video understanding. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)
16. Lin, R., Xiao, J., Fan, J.: NeXtVLAD: An efficient neural network to aggregate frame-level features for large-scale video classification. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
17. Liu, T., Liu, B.: Constrained-size tensorflow models for Youtube-8M video understanding challenge. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)

18. Miech, A., Laptev, I., Sivic, J.: Learnable pooling with context gating for video classification. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)
19. Na, S., Yu, Y., Lee, S., Kim, J., Kim, G.: Encoding video and label priors for multi-label video classification on Youtube-8M dataset. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)
20. Shin, K., Jeon, J., Lee, S.: Approach for video classification with multi-label on Youtube-8M dataset. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
21. Skalic, M., Austin, D.: Building a size constrained predictive model for video classification. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
22. Skalic, M., Pekalski, M., Pan, X.E.: Deep learning methods for efficient large scale video labeling. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)
23. Tang, Y., Zhang, X., Wang, J., Chen, S., Ma, L., Jiang, Y.G.: Non-local netVLAD encoding for video classification. In: Proc. of the 2nd Workshop on YouTube-8M Large-Scale Video Understanding (2018)
24. Wang, H.D., Zhang, T., Wu, J.: The monkeytyping solution to the Youtube-8M video understanding challenge. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)
25. Zhu, L., Liu, Y., Yang, Y.: Uts submission to google Youtube-8M challenge 2017. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)
26. Zou, H., Xu, K., Li, J., Zhu, J.: The Youtube-8M kaggle competition: Challenges and methods. In: Proc. of the CVPR Workshop on YouTube-8M Large-Scale Video Understanding (2017)