Youtube Video Classification: Video-Level vs. Frame-Level

Erika Fox Marlyne Hakizimana Shufan Xia

***Abstract*—This report describes an effort to contribute to progress in the video classification space by using the YouTube-8M dataset. We trained three different models with both detailed frame-level data and more summarized video-level data, and then tested them on three different styles of feature inputs (visual only, audio only, and combined audio/visual) in order to determine the optimal way to accurately assign multiple "tag" labels to videos. We determined our video-level logistic regression approach to be optimal, achieving a gAP score of 0.37 for the audio/visual combined input and an 8% improvement in the RGB only input's Hit@1 score as compared to our benchmark paper: YouTube-8M: A Large-Scale Video Classification Benchmark [1]. Our results provide useful insights as to what features are more important for video classification, and this work could be used to improve the experiences for both the content creators and audiences of YouTube.**

1. Introduction

YouTube "tags" are descriptive keywords to help viewers find video content. Content creators choose tags to highlight when they upload their videos [2,3]. YouTube uses these tags to organize millions of videos for downstream tasks including sorting genre, grouping linked content and establishing search relevancy in YouTube's search algorithm and recommendation system [4]. However, currently, the tags that content creators provide are not necessarily of good quality [5]. For example, a case study focusing on videos of entrepreneurial beauty vloggers highlights creators using tags such as "make-up" or "fashion" for videos unrelated to beauty to increase their visibility in the more lucrative video themes [4]. Tagging video falls under video classification.

For this report, we sought to explore the video classification space by reviewing the literature on the subject and then working to build an optimized model of our own, experimenting with different types of inputs to see how they differed in performance.

2. Background

Video classification approaches typically fall into the categories in terms of the features they use: text-based, audio-based, visual-based, or some combination thereof. The text-based approach uses Optical character recognition (OCR) to create a transcript of the dialog to analyze [6]. This approach is less common due to high error rate, often due to misspellings and omissions in transcripts. The visual-based approach for video classification typically treats video frames as still images [7]. An alternative visual-based approach focuses on motions, finding generalized patterns in optical flow and motion density, however this approach is often limited to classifying videos into binary categories such as sport and non sports [8,9]. Audio-based models use less computational power than visual-based models and utilize features such as volume, pitch, and proportion of silence [1,10]. Our work includes the audio and visual-based image approaches, and attempts to compare the performances of using these two inputs or their combination.

We also will be comparing how to aggregate video frames. Large labeled image datasets such as ImageNet [11] have encouraged extensive research in frame-level models, which have demonstrated state-of-art results in image recognition with Convolutional Neural Networks (CNN) such as AlexNet [12] and InceptionNet [11]. Similar CNN models have shown promising ability in audio classification [13]. However, the challenges for video classification models remain in the need for a large dataset with quality labels and the amount of required computational power that comes with that [14]. Various frame visual models are usually different in how frame features or frame predictions are aggregated to video-level prediction. This is sometimes done naively by averaging predictions over frames, but several state-of-art models leverage LSTM to generate compact features for video representation and classification [7,15]. Currently, the best performing model in this space is the Deep Convolution Graph Network (DCGN), which applies a variant of CNNs to gradually abstract information by convolution. The goal is to capture the sequential relations from frame to frame, and the hierarchical structure [16]. Bag-of-words inspired approaches are shown to have comparable performance [1]. Our work experiments with this approach and the naive averaging method. We hypothesize aggregating abstract frame features before classification or hidden representations during the classification results in different model performances.

3. Data

Youtube-8M is a large-scale, categorically labeled, video dataset released in 2018 by Google. Rather than retrieving the labels selected by the video creators, Youtube-8M annotated each video by deriving labels from Knowledge Graph entities based on video metadata, context, and content signals. Each video is decoded at 1 frame-per-second up to

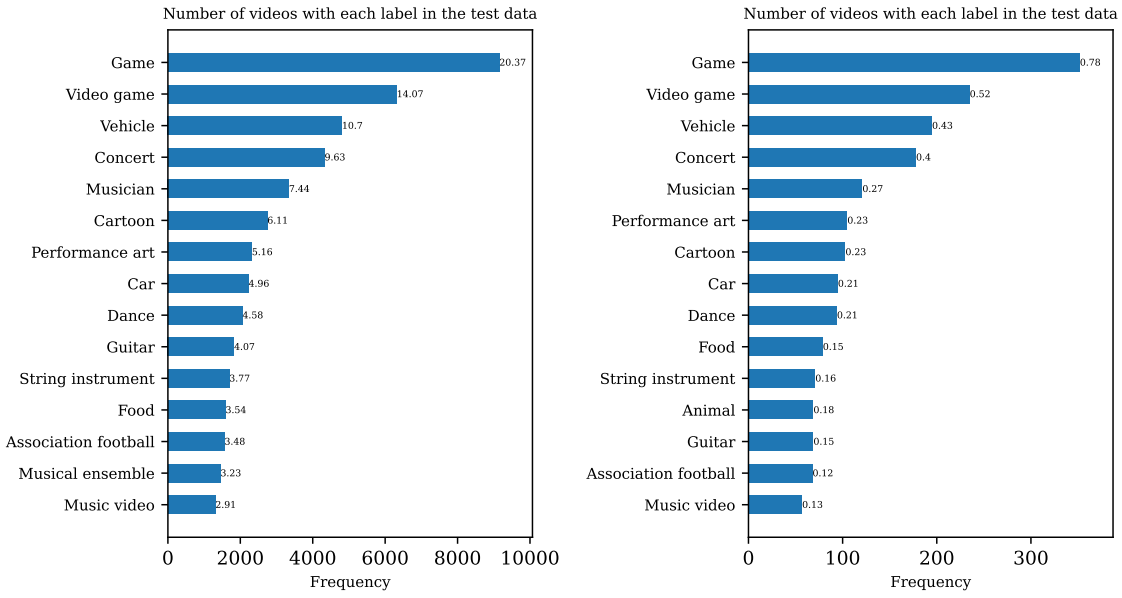


Figure 1: How many videos contains each label in the a) train and validation set, b) test set.

the first 6 minutes. At each frame, the RGB features have been pre-processed and extracted by a state-of-art inceptionV3 model in Tensorflow into 1024 float numbers [1,17]. The audio features are pre-processed using a VGG-inspired CNN audio classification model pre-trained on previous version of the data [1,13]. These abstracted audio features are represented as 128 float numbers. Due to computational power, our project uses 36k (0.6%) of the data for training and validation, 1785 videos for test performances.

We focused on examining the distribution of the tag labels when it came to checking for class imbalance. On average, each video has 3 labels, and the median number is 5. The distribution of tagged labels in training + validation is highly skewed, with "Game", "Game Video","Concert", "Vehicle" and "Musician" attributed to more than 60% of the videos (Fig. 1). This imbalance is more apparent in the test data. Therefore, evaluating model performance by each class is considered when interpreting the results. We searched for a correlation between certain labels and view count but we did not find one in this subset data.

4. Methods and Experiments

4.1. Data Preprocessing

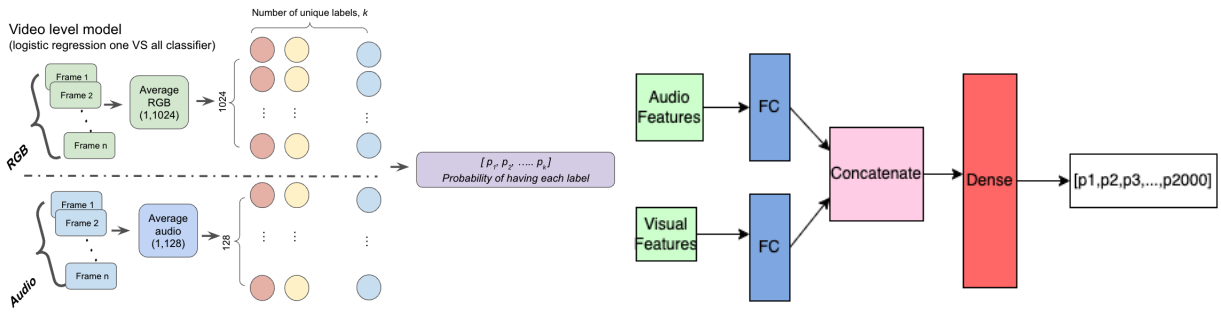
We get our data from the official YouTube 8M website where the train, validation and test data had already been separated into training and validation folders. Since the test folders do not contain the ground truth labels (they were set aside for Kaggle competition), we chose a few videos from the validation folder with available labels. Out of the 36K videos, the train+validation+test split is 85%+10%+5%. Our dataset came in tf records, so we began by extracting each video's ids, RGB features, audio features and their corresponding labels. We used the video ids for inference to extract metadata (such as view and like counts) on 25% of the videos used for training and validation. With the limited training size, we also decided to only keep videos classified with labels within the 1000 most commonly used labels as opposed to the original 3812 from the original data. As we plan to use a one vs all classification method, our ground truth labels are transformed accordingly which turns our output for each video to be (1,1000) dimensions.

4.2. Experiments

We ran three experiments in search of the best performing video classifier: 1) comparing models trained on video-level vs. frame-level data, and 2) testing our two trained models on three different kinds of inputs: a data frame containing only the pre-trained visual (RGB) features, only the pre-trained audio features, and a combination of both. Fig2 shows the architecture of each models.

4.3. Models

4.3.1. Video-level Model: Average + Logistic . Our baseline (Fig.2a) takes the naive aggregating approach by averaging the input features before feeding into a logistic regression classifier [1]. Using RGB input as an example, for each video, we start with (200, 1024) with 200 being the number of frames and 1024 the number of RGB features. After averaging over all frames, we are left with (1,1024) for each video. We then run a standard logistic classifier



(a) (b)

Figure 2: Model architecture: a) Average pooling and logistic regression b) Deep Bag of Frame

TABLE 1: Summary metrics of our models

Input level	Model	Input feature	gAP	PERR	Hit@1	F1 (threshold = 0.5)
Video-level	average-pooling + logistic	RGB	0.33	0.50	0.65	0.48
		audio	0.16	0.32	0.38	0.31
		RGB+audio	0.37	0.53	0.70	0.53
	average-pooling + MLP	RGB	0.13	0.35	0.50	0.08
		audio	0.22	0.33	0.46	0.08
		RGB+audio	0.20	0.35	0.49	0.08
Frame-level	DBoF	RGB	0.40	0.46	0.59	0.14
		audio	0.29	0.39	0.49	0.17
		RGB+audio	0.12	0.30	0.58	0.21

with binary cross entropy loss. For a one-vs-all classification method for each label, we use one-hot-encoding on the response variable to turn it into a binary ground-truth value with (1,1000) dimensions. An SGD optimizer with a 0.01 learning rate and batch size of 500 is used.

4.3.2. Video-level Model: Average+MLP . We also decide on running an MLP classifier after averaging over all frames. The goal is to see with the same aggregation method and inputs, if a neural network classifier would assign labels differently from the baseline described above. A dense layer with 4096 units is introduced as a projection layer followed by a batch normalization along with a Leaky Relu activation and dropout layers. 4096 is our final pick among 2048 and 8192 as suggested [1]. The final layer is a dense layer with 1000 classes with a sigmoid activation. For training, the same set-up of SGD is used with a categorical cross entropy loss.

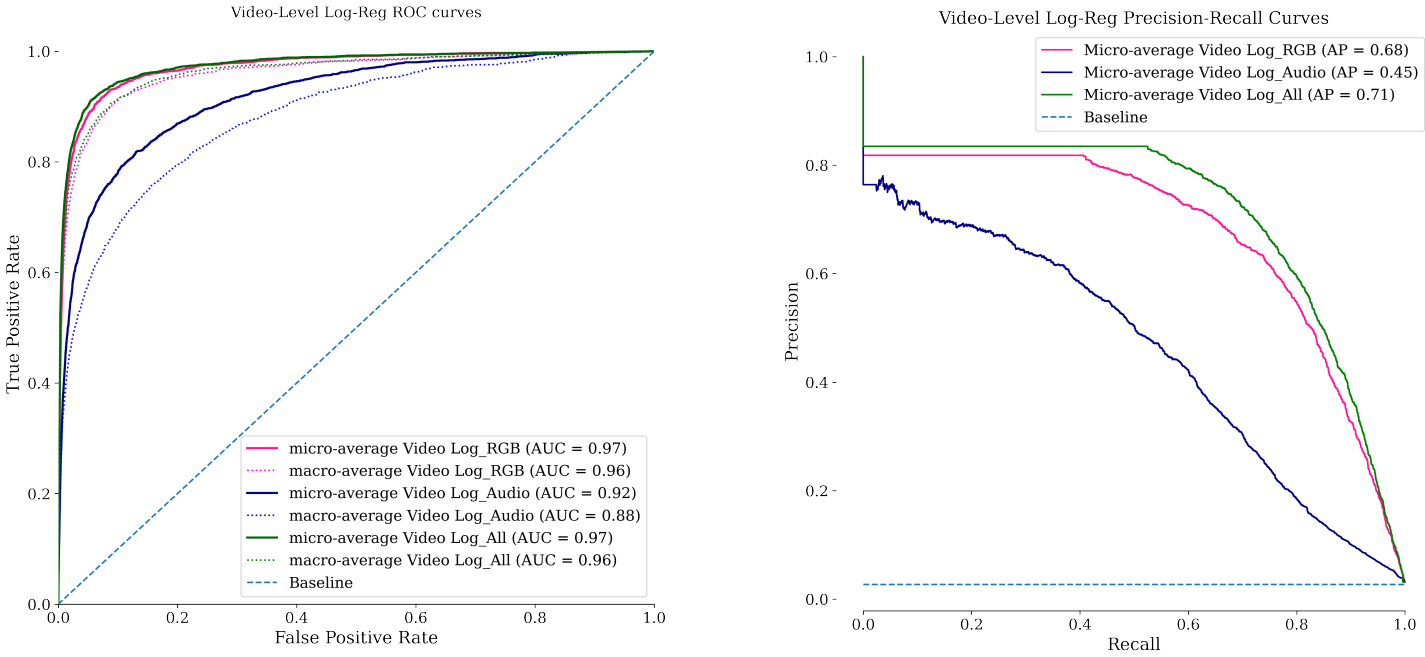
4.3.3. Frame-level Model: Deep Bag of Frame (DBoF) Pooling. This approach is inspired by using bag of words representations to aggregate frame-level features after projecting them into a higher dimension [1]. A sample of 50 random frames are picked for each video, and then are fed into a projection layer of 2048 units with leaky Relu activations that leads to a sparse coding, a technique similar Fisher Vectors [18] and VLAD [19]. After batch normalizing to help with stability, global average pooling aggregates the frames to an overall video level representation. Instead of adding a hidden layer before the output layer [1], we experiment with using logistic regression to classify these DBoF representations for shorter training time and to compare how logistic regression performs with features in even higher dimensions. An SGD optimizer with a 0.001 learning rate and batch size of 500 is used. Figure 5 displays the architecture for both audio and RGB inputs.

4.4. Metrics

For model evaluation, we chose to use A) Global Average Precision (gAP), Hit@1 and PERR as well as B) multi-class ROC and precision-recall curves. We borrow group (A) from many of our literatures, enabling us to directly compare our models to past work [1,15,16].

gAP measures weighted average precision score over all labels. Identifying the most salient label is often useful. Hit@1 represents the fraction of test samples that have the predicted most likely label in one of the one of the ground truth label. Finally, PERR gives the strictest evaluation, as it measures how many times the model gives a exact match to ground truth labels if we retrieve the same number of labels as the ground truth [1]. Evaluating our models on these metrics will allow us to make direct comparisons between our work and what has been accomplished in the field. In addition, we also report weighted averaged F1-score because it is sensitive to uneven class distribution.

Group (B) will allow us to visualize how our models are predicting each of the labels individually, and also to generate macro and micro scores in order to get an overall picture of model performance. The macro-average comes



(a) (b)

Figure 3: a) ROC and b) PR for the average pooling + logistic regression model. The results of using visual, audio and both inputs are shown in both plots.

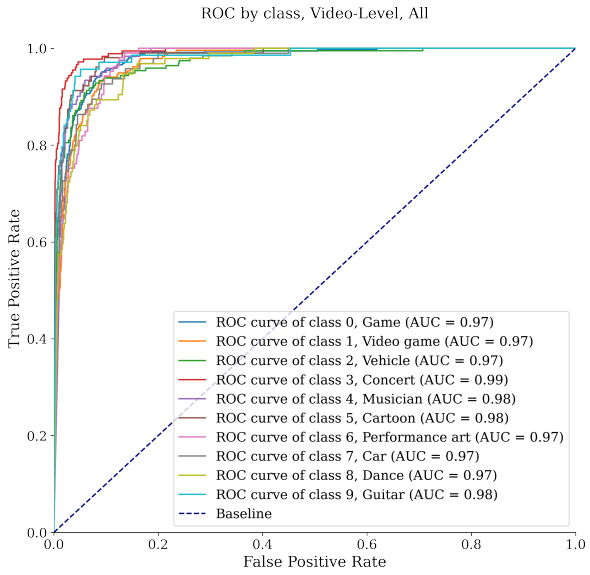


Figure 4: ROC curve for predicting each label for the top 10 most frequently tagged labels. The AUC scores vary only slightly for these 10 labels.

from getting the ROC for each label individually first, and then taking an average (all label classes weighted the same) whereas the micro-average is computed by aggregating contributions of all classes.

5. Results

5.1. gAP, PERR, Hit@1 and F1-scores: Comparisons

Table 1 shows the results for our video and frame level approaches. These metrics reveal that between our video-level models, our logistic approach yields higher scores than our MLP approach. For the logistic model, audio only input yields significantly worse results as compared to the RGB only input, implying that RGB features are more important for video classification in this case. However, the combined RGB and audio input did the best for this approach. For our MLP approach, each style of input yielded similar values across all metrics which shows that this model has no clear preference for features. One hypothesis of why the MLP approach has worse performance is the limited training size, and since our MLP model projects input features into a higher dimension, the output layer cannot predict as well. Another factor impacting our results is the sparse use of the less common labels in our dataset. These rare classes impact significantly our gAP metric, which is a problem the main benchmark paper also encounters [1].

On the frame level, our DBoF approach performs only slightly better gAP scores than the video-level logistic model for the RGB only and audio only inputs, showing that the learning is similar despite the different data types. However, for the RGB and audio combined input, gAP is worse.

Hit@1 values are pretty similar for the RGB only and audio only inputs. It improves minimally for the combined input.

The MLP video-level and DBoF frame-level approaches converged with a higher loss than our video-level logistic approach for different learning rates (experimented with between 0.01 to 0.00001) indicating that our training size and label sparsity is our main limitations.

Overall, these results show that the video-level average pooling and logistic layer approach has the best performance. Compared to benchmark results for RGB only inputs [1], we achieved an 8% improvement for Hit@1. However, we note that this improvement might be thanks to our choice to limit our analysis to the top 1000 labels in our dataset.

5.2. ROC and Precision-Recall Curves: Video-Level Logistic Regression

Fig 3a, 3b and 4 provide further understanding of how our best model performed for our different input types. Each of these plots were made considering only the top 50 labels in our dataset, despite using the top 1000 labels to make predictions with our models. Using less labels for the plots allows to get a better idea of how our models perform on the most popular labels, without getting weighed down by the sparsity in the rarer labels like in our other set of metrics. The ROC plot (Fig 3a) shows a tie for first place in AUC scores between our combined input of both RGB and audio features and the RGB only input. The Precision-Recall Plot (Fig 3b) breaks that tie, showing a higher AP score for the combined RGB and audio input. The difference in AP scores between the combined input and the RGB only input reveals that false negatives are more frequent for the RGB input, implying that audio features helped the model reduce its false negative rate, although it is still lower than we might desire at 0.71.

Fig.4 shows the individual label AUC curves for the top 10 labels of our best performing model. This is meant to further show how all labels are not predicted equally well, but they average out to a high performing 0.96.

6. Conclusion

We experimented with three models for multi-label video classification using two ways to aggregate features from frames to videos: average pooling and Deep Bag-of-Frame, as well as different input features: visual, audio and both. We did this to see if adding different inputs will drastically change results compared to our benchmark paper (Abu-El-Haija, 2016) that only uses RGB features. For our small subset of the YouTube 8M dataset, averaging input features and using a logistic regression classifier outperforms the MLP classifier and Deep Bag-of-Frame aggregation methods. Compared to the benchmark results, a decrease in prediction performance is observed when there is a relative mis-match between feature dimension and complexity of the classifier. Using extracted abstract representation from pre-trained state-of-art image networks helps with downstream classification tasks. For logistic regression classifiers, using audio features in addition to RGB features gives a marginal improvement in predictability. A major problem with multi-label classification is label distribution being highly skewed. Data augmentation should be considered before training in future work. It would be interesting to see if we could get better metric results from a similar project to ours after accounting for our current limitations in label sparsity and training size.

For application of the logistic classifier, even though ROCs seem to suggest good predictability, we can't make such inference because the data is poorly represented for the majority of the labels. It is recommended to use this model on inputs including visual features since it has a low performance on audio inputs. YouTube is a platform with very diverse types of videos. It would be interesting to look at how our model generalizes to other smaller datasets like Sports-1M [20] with more restricted classes.

Roles

- Erika Fox
    1) Report Contribution:
    Abstract, Introduction, Metrics, Results, Overall Editing.
    2) Project Management Contribution:
    Video/Slides Lead, Plotting/Metrics Lead.
- Marlyne Hakizimana
    1) Report Contribution:
    Models and Experiments, Metrics, Data, Conclusion.
    2) Project Management Contribution:
    Report Coordination Lead, Modeling Lead.
- Shufan Xia
    1) Report Contribution: Background, Data, Conclusion.
    2) Project Management Contribution: GitHub Lead, Data Lead.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A Large-Scale Video Classification Benchmark," *arXiv e-prints*, p. arXiv:1609.08675, Sep. 2016.

[2] S. Golder and B. A. Huberman, "The Structure of Collaborative Tagging Systems," *arXiv e-prints*, p. cs/0508082, Aug. 2005.

[3] "Add tags to videos - youtube help." [Online]. Available: https://support.google.com/youtube/answer/146402?hl=en&visit_id=637117734247589915-262413007&rd=1

[4] S. Bishop, "Anxiety, panic and self-optimization: Inequalities and the youtube algorithm," *CONVERGENCE (LONDON)*, vol. 24, no. 1, pp. 69–84, Feb. 2018.

[5] S. Choudhury, J. G. Breslin, and A. Passant, "Enrichment and Ranking of the YouTube Tag Space and Integration with the Linked Data Cloud," in *Lecture Notes in Computer Science*, 2009, vol. 5823, p. 747.

[6] D. Brezeale and D. J. Cook, "Automatic video classification: A survey of the literature," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 3, pp. 416–430, 2008.

[7] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," *arXiv e-prints*, p. arXiv:1503.08909, Mar. 2015.

[8] V. Kobla, D. DeMenthon, and D. S. Doermann, "Identifying sports videos using replay, text, and camera motion features," vol. 3972, pp. 332–343, Dec. 1999.

[9] M. Roach, J. Mason, and M. Pawlewski, "Motion-based classification of cartoons," in *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No.01EX489)*, 2001, pp. 146–149.

[10] P. Q. Dinh, C. Dorai, and S. Venkatesh, "Video genre categorization using audio wavelet coefficients," 2002.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, may 2017. [Online]. Available: https://doi.org/10.1145/3065386

[13] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," 2017. [Online]. Available: https://arxiv.org/abs/1609.09430

[14] B. Varadarajan, G. Toderici, S. Vijayanarasimhan, and A. Natsev, "Efficient Large Scale Video Classification," *arXiv e-prints*, p. arXiv:1505.06250, May 2015.

[15] S. Bhardwaj, M. Srinivasan, and M. M. Khapra, "Efficient Video Classification Using Fewer Frames," *arXiv e-prints*, p. arXiv:1902.10640, Feb. 2019.

[16] F. Mao, X. Wu, H. Xue, and R. Zhang, "Hierarchical Video Frame Sequence Representation with Deep Convolutional Graph Network," *arXiv e-prints*, p. arXiv:1906.00377, Jun. 2019.

[17] "Advanced guide to inception v3 nbsp;—nbsp; cloud tpu nbsp;—nbsp; google cloud." [Online]. Available: https://cloud.google.com/tpu/docs/inception-v3-advanced

[18] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[19] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.