A TEACHER STUDENT NETWORK FOR FASTER VIDEO CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Over the past few years, various tasks involving videos such as classification, description, summarization and question answering have received a lot of attention. Current models for these tasks compute an encoding of the video by treating it as a sequence of images and going over every image in the sequence, which becomes computationally expensive for longer videos. In this paper, we focus on the task of video classification and aim to reduce the computational cost by using the idea of distillation. Specifically, we propose a *Teacher-Student* network wherein the teacher looks at all the frames in the video but the student looks at only a small fraction of the frames in the video. The idea is to then train the student to minimize (i) the difference between the final representation computed by the student and the teacher and/or (ii) the difference between the distributions predicted by the teacher and the student. This smaller student network which involves fewer computations but still learns to mimic the teacher can then be employed at inference time for video classification. We experiment with the YouTube-8M dataset and show that the proposed student network can reduce the inference time by upto 30% with a negligent drop in the performance.

1 INTRODUCTION

Today video content has become extremely prevalent on the internet influencing all aspects of our life such as education, entertainment, communication *etc.* This has led to an increasing interest in automatic video processing with the aim of identifying activities (Simonyan & Zisserman, 2014; Yue-Hei Ng et al., 2015), generating textual descriptions (Donahue et al., 2015), generating summaries (Zhang et al., 2016; Pan et al., 2016), answering questions (Jang et al., 2017) and so on. On one hand, with the availability of large scale datasets (Soomro et al., 2012; Wang et al., 2017a; Kuehne et al., 2011; Abu-El-Haija et al., 2016; Xiao et al., 2016) for various video processing tasks, it has now become possible to train increasingly complex models which have high memory and computational needs but on the other hand there is a demand for running these models on low power devices such as mobile phones and tablets with stringent constraints on latency and computational cost. It is important to balance the two and design models which can learn from large amounts of data but still be computationally cheap at inference time.

With the above motivation, we focus on the task of video classification (Abu-El-Haija et al., 2016) and aim to reduce the computational cost at inference time. Current state of the art models for video classification (Yue-Hei Ng et al., 2015; Li et al., 2017; Wu et al., 2016; Skalic et al., 2017) treat the video as a sequence of images (or frames) and compute a representation of the video by using a Recurrent Neural Network (RNN). The input to the RNN at every time step is an encoding of the corresponding image (frame) at that time step as obtained from a Convolutional Neural Network. Computing such a representation for longer videos can be computationally very expensive as it requires running the RNN for many time steps. Further, for every time step the corresponding frame from the video needs to pass through a convolutional neural network to get its representation. Such computations are still feasible on a GPU but become infeasible on low end devices which have power, memory and computational constraints.

Typically, one can afford more computational resources at training time but a less expensive model is desired at test time. We propose to achieve this by using the idea of distillation wherein we train a computationally expensive *teacher* network which computes a representation for the video

by processing all frames in the video. We then train a relatively inexpensive *student* network whose objective is to process only a few frames of the video and produce a representation which is very similar to the representation computed by the teacher. This is achieved by minimizing (i) the squared error loss between the representations of the student network and the teacher network and/or (ii) by minimizing the difference between the output distributions (class probabilities) predicted by the two networks. We refer to this as the *matching loss*. Figure 1 illustrates this idea where the teacher sees every frame of the video but the student sees fewer frames, *i.e.*, every j-th frame of the video. At inference time, we then use the student network for classification thereby reducing the time required for processing the video.

We experiment with two different methods of training the *Teacher-Student* network. In the first method (which we call *Serial* Training), the teacher is trained independently and then the student is trained to match the teacher with or without an appropriate regularizer to account for the classification loss. In the second method (which we call *Parallel* Training), the teacher and student are trained jointly using the classification loss as well as the matching loss. We experiment with the YouTube-8M dataset and show that the smaller student network reduces the inference time by upto 30% while still achieving a classification performance which is very close to that of the expensive teacher network.

2 Related Work

Since we focus on task of video classification in the context of the YouTube-8M dataset (Abu-El-Haija et al., 2016), we first review some recent works on video classification. In the latter half of this section, we review some relevant work on model compression in the context of image processing.

On average the videos in the YouTube-8M dataset dataset have a length of 200 seconds. Each video is represented using a sequence of frames where every frame corresponds to one second of the video. These one-second frame representations are pre-computed and provided by the authors. The authors also proposed a simple baseline model which treats the entire video as a sequence of these one-second frames and uses an Long short-term memory networks (LSTM) to encode this sequence. Apart from this, they also propose some simple baseline models like Deep Bag of Frames (DBoF) and Logistic Regression (Abu-El-Haija et al., 2016). Various other classification models (Miech et al., 2017; Wang et al., 2017b; Li et al., 2017; Chen et al., 2017b; Skalic et al., 2017) have been proposed and evaluated on this dataset which explore different methods of: (i) feature aggregation in videos (temporal as well as spatial) (Chen et al., 2017b; Miech et al., 2017), (ii) capturing the interactions between labels (Wang et al., 2017b) and (iii) learning new non-linear units to model the interdependencies among the activations of the network (Miech et al., 2017). We focus on one such state of the art model, *viz.*, a hierarchical model whose performance is comparable to that of a single (non-ensemble) best performing model (*Multi Scale CNN-LSTM* reported in Wang et al. (2017b)) on this dataset. We take this model as the teacher network and train a comparable student network as explained in the next section.

Recently, there has been a lot of work on model compression in the context of image classification. We refer the reader to the survey paper by Cheng et al. (2017) for a thorough review of the field. For brevity, here we refer to only those papers which use the idea of distillation. For example, Ba & Caruana (2014); Hinton et al. (2015); Lopez-Paz et al. (2016); Chen et al. (2017a) use *Knowledge Distillation* to learn a more compact *student* network from a computationally expensive *teacher* network. The key idea is to train a shallow student network to mimic the deeper teacher network, ensuring that the final output representations produced by the student network are very close to those produced by the teacher network. This is achieved by training the student network using soft targets (or class probabilities) generated by the teacher instead of the hard targets present in the training data. There are several other variants of this technique, for example, Romero et al. (2015) extend this idea to train a student model which not only learns from the outputs of the teacher but also uses the intermediate representations learned by the teacher as additional *hints*. This idea of *Knowledge Distillation* has also been tried in the context of pruning networks in various domains (Chen et al., 2017a; Polino et al., 2018). For example, Chen et al. (2017a) exploited the idea of intermediate representation matching along with *Knowledge Distillation* to compress state-of-art models for multiple object detection. On similar lines, Polino et al. (2018) combine the ideas of *Quantization* and *Knowledge Distillation* for better model compression in different image classification models. Stu-

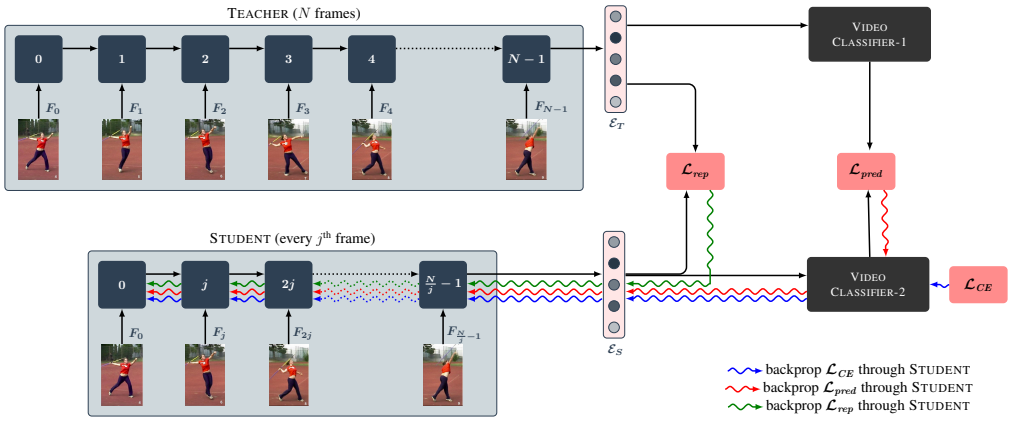


Figure 1: Architecture of TEACHER-STUDENT network for video classification

dent teacher networks have also been proposed for speech recognition (Wong & Gales, 2016) and reading comprehension (Hu et al., 2018). While in these works, the teacher and student differ in the number of layers, in our case, the teacher and student network differ in the number of time steps or frames processed by the two networks. To the best of our knowledge, this is the first work which uses a *Teacher-Student* network for video classification.

3 PROPOSED APPROACH

Given a fixed set of m classes $y_1, y_2, y_3, ..., y_m \in \mathscr{Y}$ and a video $\mathbf{V}$ containing N frames $(F_0, F_1, \ldots, F_{N-1})$, the goal of video classification is to identify all the classes to which the video belongs. In other words, for each of the m classes we are interested in predicting the probability $P(y_i|\mathbf{V})$. This probability can be parameterized using a neural network f which looks at all the frames in the video to predict:

$$P(y_i|\mathbf{V}) = f(F_0, F_1, \ldots, F_{N-1})$$

The focus of this work is to design a simpler network g which looks at only a fraction of the N frames at inference time while still allowing it to leverage the information from all the N frames at training time. To achieve this, we propose a teacher student network as described below wherein the teacher has access to more frames than the student.

TEACHER: The teacher network can be any state of the art video classification model and in this work we consider the hierarchical RNN based model. This model assumes that each video contains a sequence of b equal sized blocks. Each of these blocks in turn is a sequence of l frames thereby making the entire video a sequence of sequences. In the case of the YouTube-8M dataset, these frames are one-second shots of the video and each block b is a collection of l such one-second frames. The model contains a lower level RNN to encode each block (sequence of frames) and a higher level RNN to encode the video (sequence of blocks). As is the case with all state of the art models for video classification, this teacher network looks at all the N frames of video $(F_0, F_1, \ldots, F_{N-1})$ and computes an encoding $\mathcal{E}_T$ of the video, which is then fed to a simple feedforward neural network with a multi-class output layer containing a sigmoid neuron for each of the $\mathscr{Y}$ classes. This teacher network f can be summarized by the following set of equations:

$$\begin{aligned}
\mathcal{C}_t &= CNN(F_t) && \forall t \in \{0 \ldots N-1\} \text{ and } \mathcal{C}_t \in \mathbb{R}^{1024} \\
h_t &= LSTM(\mathcal{C}_t, h_{t-1}) && \forall t \in \{0 \ldots N-1\} \text{ and } h_t \in \mathbb{R}^{4096} \\
\hat{y}_i &= \sigma(W_i^T h_{N-1}) && \forall \hat{y}_i \in \mathscr{Y} \text{ classes}
\end{aligned}$$

The CNN used in the above equations is typically a pre-trained network and its parameters are not fine-tuned. The remaining parameters of the teacher network which includes the parameters of the $LSTM$ as well as the output layer (with W parameters) are learnt using a standard multi-label

classification loss $\mathcal{L}_{CE}$, which is a sum of the cross-entropy loss for each of the $\mathscr{Y}$ classes. We refer to this loss as $\mathcal{L}_{CE}$ where the subscript CE stands for cross entropy between the true labels y and predictions $\hat{y}$.

$$\mathcal{L}_{CE} = -\sum_{i=1}^{|\mathscr{Y}|} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{1}$$

STUDENT: In addition to this teacher network, we introduce a student network which only processes every j^{th} frame ($F_0, F_j, F_{2j}, \ldots, F_{\frac{N}{j}-1}$) of the video (as shown in figure 1) and computes a representation $\mathcal{E}_S$ of the video from these $\frac{N}{j}$ frames. Similar to the teacher, the student also uses a hierarchical recurrent neural network to compute this representation which is again fed to a simple feedforward neural network with a multi-class output layer. The parameters of this output layer are shared between the teacher and the student. The student is trained to minimize the squared error loss between the representation computed by the student network and the representation computed by the teacher. We refer to this loss as $\mathcal{L}_{rep}$ where the subscript *rep* stands for representations.

$$\mathcal{L}_{rep} = ||\mathcal{E}_T - \mathcal{E}_S||^2 \tag{2}$$

We also try a simple variant of the model, where in addition to ensuring that the final representations $\mathcal{E}_S$ and $\mathcal{E}_T$ are similar, we also ensure that the intermediate representations ($\mathcal{I}_S$ and $\mathcal{I}_T$) of the models are similar. In particular, we ensure that the representation of the frames $j, 2j$ and so on computed by the teacher and student network are very similar by minimizing the squared error distance between the corresponding intermediate representations. We refer to this loss as $\mathcal{L}^{\mathcal{I}}_{rep}$ where the superscript $\mathcal{I}$ stands for intermediate.

$$\mathcal{L}^{\mathcal{I}}_{rep} = \sum_{i=j,2j,..}^{\frac{N}{j}-1} ||\mathcal{I}^i_T - \mathcal{I}^i_S||^2 \tag{3}$$

Alternately, the student can also be trained to minimize the difference between the class probabilities predicted by the teacher and the student. We refer to this loss as $\mathcal{L}_{pred}$ where the subscript $pred$ stands for *predicted probabilities*. More specifically if $\mathcal{P}_T = \{p^1_T, p^2_T, ..., p^m_T\}$ and $\mathcal{P}_S = \{p^1_S, p^2_S, ..., p^m_S\}$ are the probabilities predicted for the m classes by the teacher and the student then

$$\mathcal{L}_{pred} = d(\mathcal{P}_T, \mathcal{P}_S) \tag{4}$$

where d is any suitable distance metric such as *KL* divergence or squared error loss.

TRAINING: Intuitively, it makes sense to train the teacher first and then use this trained teacher to guide the student. We refer to this as the *Serial* mode of training as the student is trained after the teacher as opposed to jointly. For the sake of analysis, we use different combinations of loss function to train the student as described below:

(a) $\mathcal{L}_{rep}$: Here, we operate in two stages. In the first stage, we train the student network to minimize the $\mathcal{L}_{rep}$ as defined above, *i.e.*, we train the RNN parameters of the student network to produce representations which are very similar to the teacher network. The idea is to let the student learn by only mimicking the teacher and not worry about the final classification loss. In the second stage, we then plug-in the classifier trained along with the teacher (see Equation 1) and finetune all the parameters of the student and the classifier using the cross entropy loss, $\mathcal{L}_{CE}$. In practice, we found that the finetuning done in the second stage helps to improve the performance of the student.

(b) $\mathcal{L}_{rep} + \mathcal{L}_{CE}$: Here, we train the student to jointly minimize the representation loss as well as the classification less. The motive behind this was to ensure that while mimicking the teacher, the student also keeps an eye on the final classification loss from the beginning (instead of being finetuned later as in the case above).

MODEL		k=6		k=10		k=15		k=20		k=30	
		GAP	MAP	GAP	MAP	GAP	MAP	GAP	MAP	GAP	MAP
Teacher-Skyline										**0.811**	**0.414**
Model with k frames		BASELINE METHODS									
Uniform-k		0.715	0.266	0.759	0.324	0.777	0.35	0.785	0.363	0.795	0.378
Random-k		0.679	0.246	0.681	0.254	0.717	0.268	0.763	0.329	0.774	0.339
First-k		0.478	0.133	0.539	0.163	0.595	0.199	0.632	0.223	0.676	0.258
Middle-k		0.577	0.178	0.600	0.198	0.620	0.214	0.638	0.229	0.665	0.25
Last-k		0.255	0.062	0.267	0.067	0.282	0.077	0.294	0.083	0.317	0.094
First-Middle-Last-k		0.640	0.215	0.671	0.242	0.680	0.249	0.698	0.268	0.721	0.287
Training	*Student-Loss*	TEACHER STUDENT METHODS									
Parallel	L_{rep}	0.724	0.280	0.762	0.331	0.785	0.365	0.794	0.380	0.803	0.392
Parallel	L_{rep}, L_{CE}	0.726	0.285	0.766	0.334	0.785	0.362	0.795	0.381	0.804	0.396
Parallel	$L_{rep}, L_{pred}, L_{CE}$	0.729	0.292	0.770	0.337	**0.789**	0.371	0.796	0.388	**0.806**	0.404
Serial	L_{rep}	0.727	0.288	0.768	0.339	0.786	0.365	0.795	0.381	0.802	0.394
Serial	L_{rep}, L_{CE}	0.728	0.291	0.769	0.341	0.786	0.368	0.794	0.383	0.803	0.399
Serial	$L_{rep}, L_{pred}, L_{CE}$	**0.731**	**0.297**	**0.771**	**0.349**	**0.789**	**0.375**	**0.798**	**0.390**	**0.806**	**0.405**

Table 1: Performance Comparison of proposed *Teacher-Student* models using different ***Student-Loss*** variants, with their corresponding baselines using k frames. Teacher-Skyline refers to the default model which process all the frames in a video.

(c) $\mathcal{L}_{rep} + \mathcal{L}_{CE} + \mathcal{L}_{pred}$: Finally, we also add in $\mathcal{L}_{pred}$ to enable the student to learn from the soft targets obtained from the teacher in addition to hard target labels present in the training data. Figure 1 illustrates the process of training the student with different loss functions.

For the sake of completeness, we also tried an alternate mode in which train the teacher and student in parallel such that the objective of the teacher is to minimize $\mathcal{L}_{CE}$ and the objective of the student is to minimize one of the 3 combinations of loss functions described above.

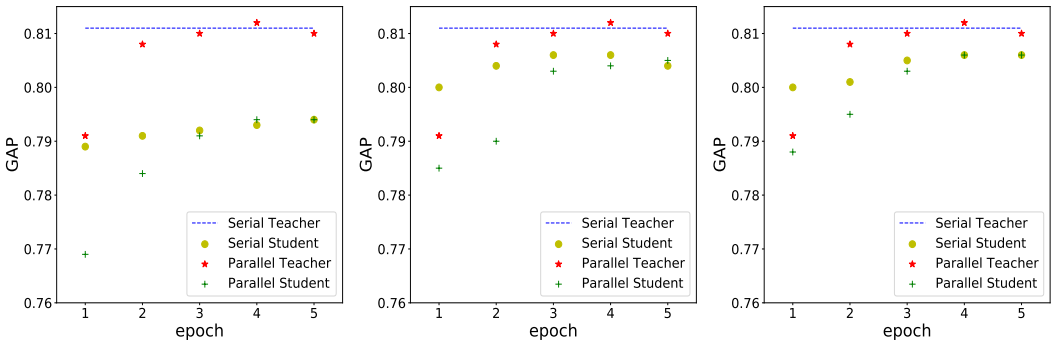
4 EXPERIMENTAL SETUP

In this section, we describe the dataset used for our experiments, the hyperparameters that we considered, the baseline models that we compared with and the effect of different loss functions and training methods.

1. Dataset: The YouTube-8M dataset (Abu-El-Haija et al., 2016) contains 8 million videos with multiple classes associated with each video. The average length of a video is $200s$ and the maximum length of a video is $300s$. The authors of the dataset have provided pre-extracted audio and visual features for every video such that every second of the video is encoded as a single frame. The original dataset consists of 5,786,881 training (70%), 1,652,167 validation (20%) and 825,602 test examples (10%). Since the authors did not release the test set, we used the original validation set as test set and report results on it. In turn, we randomly sampled 48,163 examples from the training data and used these as validation data for tuning the hyperparameters of the model. We trained our models using the remaining 5,738,718 training examples.

2. Hyperparameters: For all our experiments, we used Adam Optimizer with the initial learning rate set to 0.001 and then decreased it exponentially with 0.95 decay rate. We used a batch size of 256. For both the student and teacher networks we used a 2-layered MultiLSTM Cell with cell size of 1024 for both the layers of the hierarchical model. For regularization, we used dropout (0.5) and $\mathbf{L}_2$ regularization penalty of 2 for all the parameters. We trained all the models for 5 epochs and then picked the best model based on validation performance. We did not see any benefit of training beyond 5 epochs. For the teacher network we chose the value of l (number of frames per block) to be 20 and for the student network we set the value of l to 5 or 3 depending on the reduced number of frames considered by the student.

3. Evaluation Metrics: We used the following metrics as proposed in (Abu-El-Haija et al., 2016) for evaluating the performance of different models :



(a) Training with L_{rep} (b) Training with L_{rep} and L_{CE} (c) Training: L_{rep},L_{CE},L_{pred}

Figure 2: Performance Comparison (GAP score) of different variants of *Serial* and *Parallel* methods in *Teacher-Student* training.

- GAP (Global Average Precision): is defined as

$$GAP = \sum_{i=1}^{P} p(i) \nabla r(i)$$

  where $p(i)$ is the precision at prediction i, $\nabla r(i)$ is the change in recall at prediction i and P is the number of top predictions that we consider. Following the original YouTube-8M Kaggle competition we use the value of P as 20.
- mAP (Mean Average Precision) : The mean average precision is computed as the unweighted mean of all the per-class average precisions.

4. Baseline Models: As mentioned earlier the student network only processes k frames in the video. We have considered different baselines to explore the possible selections of frames in a video sequence. We report results with different values of k : 6, 10, 15, 20, 30 and compare the performance of our student networks with the following models:

a) Teacher-Skyline: The original hierarchical model which processes all the frames of the video. This, in some sense, acts as the upper bound on the performance.

b) Uniform-k : A hierarchical model trained from scratch which only processes k frames of the video. These frames are separated by a constant interval and are thus equally spaced. However, unlike the student model this model does not try to match the representations produced by the full teacher network.

c) Random-k: A hierarchical model which only processes k frames of the video. These frames are sampled randomly from the video and may not be equally spaced.

d) First-k: A hierarchical model which processes the starting k frames of the video.

e) Middle-k: A hierarchical model which processes the middle k frames of the video.

f) Last-k: A hierarchical model which processes the last k frames of the video.

g) First-Middle-Last-k: A hierarchical model which processes k frames by selecting the starting $\frac{k}{3}$, middle $\frac{k}{3}$ and ending $\frac{k}{3}$ frames of the video.

5 Discussion And Results

The results of our experiments are summarized mainly in Tables 1 (performance) and 4 (computation time). We also report some additional results in Table 2 and 3. Below, we discuss the main observations from our experiments:

1. Comparisons of different baselines: First, we simply compare the performance of different baselines listed in the top half of Table 1. As is evident, the Uniform-k baseline which looks at

MODEL		Intermediate		Final	
		GAP	mAP	GAP	mAP
Parallel	L_{rep}	0.803	0.393	0.803	0.392
Parallel	$L_{rep} + L_{CE}$	0.803	0.396	0.804	0.396
Parallel	$L_{rep} + L_{pred}$	0.804	0.400	0.806	0.404
Serial	L_{rep}	0.804	0.395	0.802	0.394
Serial	$L_{rep} + L_{CE}$	0.803	0.397	0.803	0.399
Serial	$L_{rep} + L_{pred}$	0.806	0.405	0.806	0.405

Table 2: Comparison of *Final* and *Intermediate* representation matching by *Student* network using k:30 frames.

equally spaced k frames performs better than all the other baselines. The performance gap between Uniform-k and the other baselines is even more significant when the value of k is small. The main purpose of this experiment was to decide the right way of selecting frames for the student network. Based on these results, we ensured that for all our experiments, we fed equally spaced k frames to the student. Also, these experiments suggest that Uniform-k is a strong baseline to compare against.

2. Comparing Teacher-Student Network with Uniform-k Baseline: As mentioned above, the Uniform-k is a simple but effective way of reducing the number of frames to be processed. We observe that all the teacher-student models outperform this strong baseline. Further, in a separate experiment as reported in Table 3 we observe that when we reduce the number of training examples seen by the teacher and the student, then the performance of the Uniform-k baseline drops and is much lower than that of the corresponding teacher student network. This suggests that the teacher student network can be even more useful when the amount of training data is limited.

3. Serial Versus Parallel Training of Teacher-Student: While the best results in Table 1 are obtained using *Serial* training, if we compare the corresponding rows of *Serial* and *Parallel* training we observe that there is not much difference between the two. We found this to be surprising and investigated this further. In particular, we compared the performance of the teacher after different epochs in the *Parallel* training setup with the performance of the a static teacher trained independently (Serial). We plotted this performance in Figure 2 and observed that after 3-4 epochs of training, the *Parallel* teacher is able to perform at par with *Serial* teacher (the constant blue line). As a result, the *Parallel* student now learns from this trained teacher for a few more epochs and is almost able to match the performance of the *Serial* student. This trend is same across the different combinations of loss functions that we used.

4. Visualization of Teacher and Student Representations: Apart from evaluating the final performance of the model in terms of MAP and GAP, we also wanted to check if the representations learned by the teacher and student are indeed similar. To do this, we chose top-5 classes (*class1:Vehicle, class2: Concert, class3: Association football, class4: Animal, class5: Food*) in the Youtube-8M dataset and visualized the TSNE-embeddings of the representations computed by the student and the teacher for the same video (see Figure 3). We use the darker shade of a color to represent teacher embeddings of a video and a lighter shade of the same color to represent the students embeddings of the same video. We observe that the dark shades and the light shades of the same color almost completely overlap showing that the student and teacher representations are indeed very close to each other.

5. Matching Intermediate v/s Final representations: Intuitively, it seemed that the student should benefit more if we train it to match the intermediate representations of the teacher at different timesteps as opposed to only the final representation at the last time step. However, as reported in Table 2, we did not see any benefit of matching intermediate representations.

6. Computation time of different models: Lastly, the main aim of this work was to ensure that the computational cost and time is minimized at inference time. The computational cost can be measured in terms of the number of FLOPs. The main result from the table 4 is that when k=30, the inference time drops by 30% where the number of FLOPs reduces by 90%, but the performance of the model is not affected. In particular, as seen in Table 1, when $k = 30$, the GAP and mAP drop by

Model	Metric	%age of training data		
		10%	25%	50%
Serial	GAP	0.774	0.788	0.796
	MAP	0.345	0.369	0.373
Uniform	GAP	0.718	0.756	0.776
	MAP	0.220	0.301	0.349

Table 3: Effect of amount of training data on performance of *Serial* and *Uniform* models using 30 frames

Model	Time (hrs.)	FLOPs (Billion)
Teacher-Skyline	13.00	5.042
k: 10	7.61	0.151
k: 20	8.20	0.252
k: 30	**9.11**	0.504

Table 4: Comparison of FLOPs and evaluation time of models using k frames with *Skyline* model on original validation set using Tesla k80s GPU

0.5-0.9% and 0.9-2% respectively as compared to the teacher skyline. This shows that the proposed teacher student model is an effective way of reducing the computational cost and time.

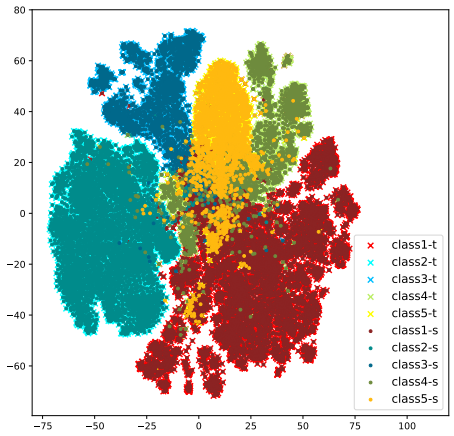


Figure 3: TSNE-Embedding of teacher and student representations. Here, class c refers to the cluster representation obtained corresponding to c^{th} class, whereas t and s denote teacher and student embedding respectively.

6 Conclusion and Future Work

We proposed a method to reduce the computation time for video classification using the idea of distillation. Specifically, we first train a teacher network which computes a representation of the video using all the frames in the video. We then train a student network which only processes k frames of the video. We use different combinations of loss functions which ensures that (i) the final representation produced by the student is the same as that produced by the teacher and (ii) the output probability distributions produced by the student are similar to those produced by the teacher. We compare the proposed models with a strong baseline and skyline and show that the proposed model outperforms the baseline and gives a significant reduction in terms of computational time and cost when compared to the skyline. In particular, We evaluate our model on the YouTube-8M dataset and show that the computationally less expensive student network can reduce the computation time by upto 30% while giving similar performance as the teacher network.

As future work, we would like to evaluate our model on other video processing tasks such as summarization, question answering and captioning. We would also like to experiment with more complex and different teacher networks other than the hierarchical RNN considered in this work. We would also like to independently train an agent which learns to select the most favorable k frames of the video as opposed to simply using equally spaced k frames.

REFERENCES

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

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems 27*, pp. 2654–2662, 2014.

Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 742–751. Curran Associates, Inc., 2017a.

Shaoxiang Chen, Xi Wang, Yongyi Tang, Xinpeng Chen, Zuxuan Wu, and Yu-Gang Jiang. Aggregating frame-level features for large-scale video classification. *CoRR*, abs/1707.00803, 2017b.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *CoRR*, abs/1710.09282, 2017.

Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL `http://arxiv.org/abs/1503.02531`.

Minghao Hu, Yuxing Peng, Furu Wei, Zhen Huang, Dongsheng Li, Nan Yang, and Ming Zhou. Attention-guided answer distillation for machine reading comprehension. *CoRR*, abs/1808.07644, 2018.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pp. 2556–2563, Nov 2011. doi: 10.1109/ICCV.2011.6126543.

Fu Li, Chuang Gan, Xiao Liu, Yunlong Bian, Xiang Long, Yandong Li, Zhichao Li, Jie Zhou, and Shilei Wen. Temporal modeling approaches for large-scale youtube-8m video understanding. *CoRR*, abs/1707.04555, 2017.

D. Lopez-Paz, B. Schölkopf, L. Bottou, and V. Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016.

Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *CoRR*, abs/1706.06905, 2017.

P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1029–1038, June 2016. doi: 10.1109/CVPR.2016.117.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=S1XolQbRW`.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *In Proceedings of ICLR*, 2015.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 568–576. Curran Associates, Inc., 2014.

Miha Skalic, Marcin Pekalski, and Xingguo E. Pan. Deep learning methods for efficient large scale video labeling. *CoRR*, abs/1706.04572, 2017.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

Haiqiang Wang, Ioannis Katsavounidis, Jiantong Zhou, Jeonghoon Park, Shawmin Lei, Xin Zhou, Man-On Pun, Xin Jin, Ronggang Wang, Xu Wang, Yun Zhang, Jiwu Huang, Sam Kwong, and C.-C. Jay Kuo. Videoset: A large-scale compressed video quality dataset based on jnd measurement. *Journal of Visual Communication and Image Representation*, 46, 2017a.

He-Da Wang, Teng Zhang, and Ji Wu. The monkeytyping solution to the youtube-8m video understanding challenge. *CoRR*, abs/1706.05150, 2017b.

Jeremy H. M. Wong and Mark J. F. Gales. Sequence student-teacher training of deep neural networks. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pp. 2761–2765, 2016.

Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. Deep learning for video classification and captioning. *CoRR*, abs/1609.06782, 2016.

Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *Int. J. Comput. Vision*, 119(1):3–22, August 2016. ISSN 0920-5691. doi: 10.1007/s11263-014-0748-y.

Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*. Springer, 2016.