Efficient Video Classification Using Fewer Frames



Shweta Bhardwaj
Indian Institute of Technology Madras

Mukundhan Srinivasan
NVIDIA Bangalore

Mitesh M. Khapra
Indian Institute of Technology Madras

Introduction

- Amazing growth in online video content

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Introduction

Example: YouTube-8 Million[1]Video Dataset - 2 TB

- Amazing growth in online video content
- Availability of large scale datasets

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Introduction

- Amazing growth in online video content
- Availability of large scale datasets $\rightarrow$ Complex models

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Introduction

- Amazing growth in online video content
- Availability of large scale datasets $\rightarrow$ Complex models
- More demand for high memory and computational requirements

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Introduction

- Amazing growth in online video content
- Availability of large scale datasets $\rightarrow$ Complex models
- More demand for high memory and computational requirements
- End goal?

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Introduction

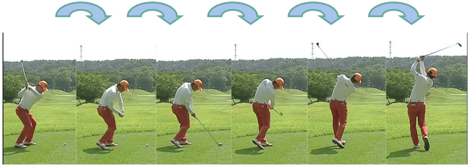
- Amazing growth in online video content
- Availability of large scale datasets $\rightarrow$ Complex models
- More demand for high memory and computational requirements
- End goal? Need to run models on low-power devices

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Motivation in Videos

- Existing models process almost all the frames in videos

Motivation in Videos



- Existing models process almost all the frames in videos
- Longer sequence

Motivation in Videos

- Existing models process almost all the frames in videos
- Longer sequence → Slow and costly video processing

Motivation in Videos

- Existing models process almost all the frames in videos
- Longer sequence $\rightarrow$ Slow and costly video processing
- Redundancy in consecutive frames

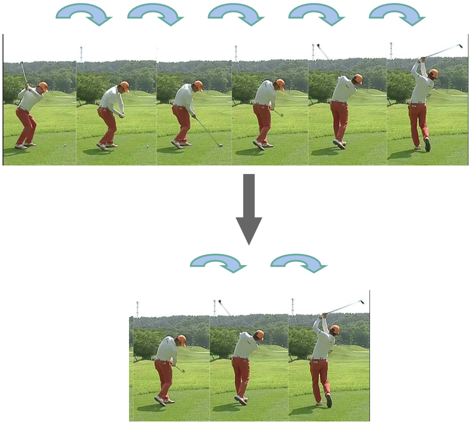
Motivation in Videos

- Existing models process almost all the frames in videos
- Longer sequence $\rightarrow$ Slow and costly video processing
- Redundancy in consecutive frames
- High demand for compute-efficient models

Motivation in Videos

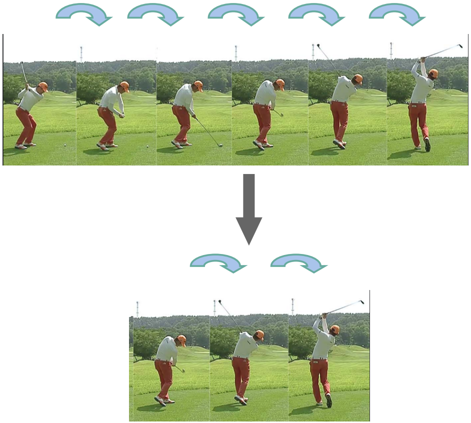
- Existing models process almost all the frames in videos
- Longer sequence → Slow and costly video processing
- Redundancy in consecutive frames
- High demand for compute-efficient models
- Any scope to reduce extra computations ? Yes

Motivation for Videos



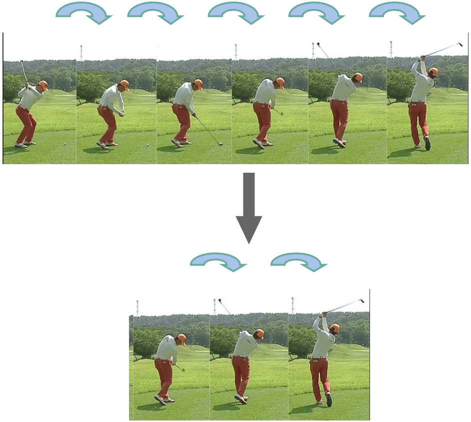
- Directions of work ?

Motivation for Videos



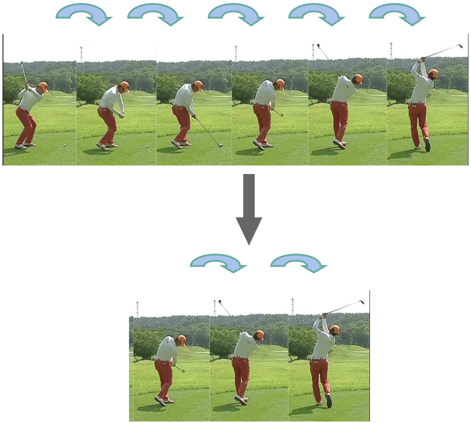
- Directions of work :
  1. Use a fraction of frames only

Motivation for Videos



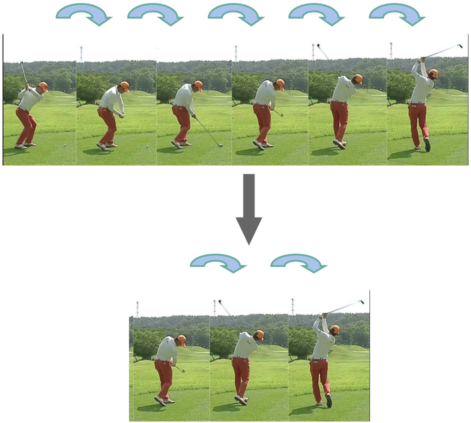
- Directions of work :
  1. Use a fraction of frames only
  2. Reduce memory requirement

Motivation for Videos



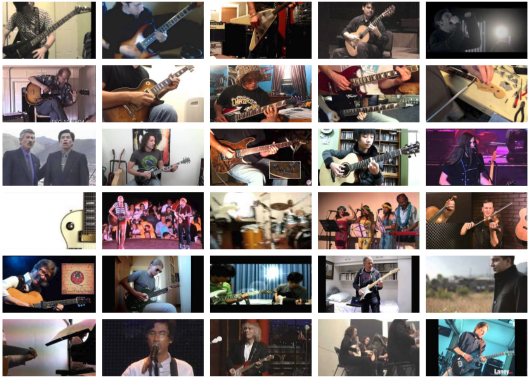
- Directions of work :
  1. Use a fraction of frames only
  2. Reduce memory requirement
- We focus on complementary approach of frame reduction

Motivation for Videos



- Directions of work :
  1. Use a fraction of frames only
  2. Reduce memory requirement
- We focus on complementary approach of frame reduction
- Necessary to balance the trade-off b/w performance on *classification* and efficiency

Dataset : Multi-Label Video Classification



YouTube-8M dataset[1]

- 7 million videos
- 450,000 hours
- 230s avg. video length
- 4,716 classes
- 23 max. labels in a video
- 3.4 avg. labels/video
- 3.2B visual features

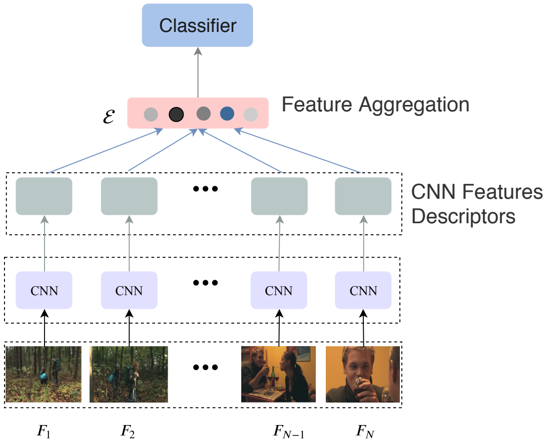
Visual features are extracted from ResNet-50[2]

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

[2] Deep Residual Learning for Image Recognition

Video Processing Pipeline

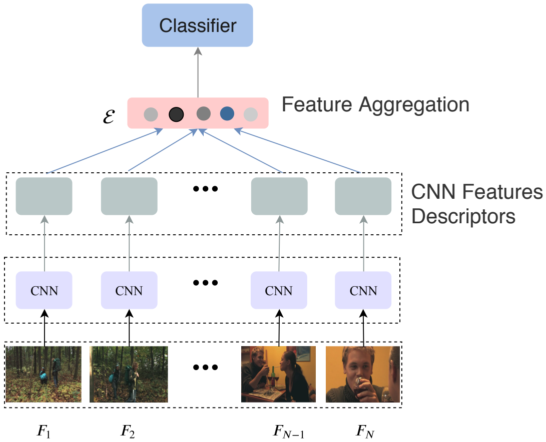
CNN feature extraction of video frames



- Extract features from each raw frame

Video Processing Pipeline

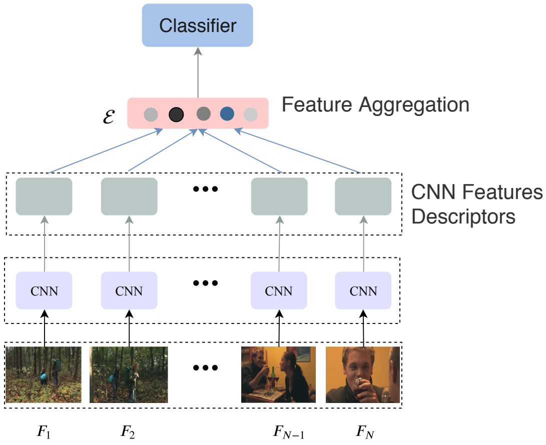
CNN feature extraction of video frames



- Extract features from each raw frame
- Features are aggregated using different methods (Recurrent or Non-Recurrent)

Video Processing Pipeline

CNN feature extraction of video frames



- Extract features from each raw frame
- Features are aggregated using different methods (Recurrent or Non-Recurrent)
- Single video encoding vector $\mathcal{E}$ is fed to 'Classifier' module

Video Classification Models

- Recurrent Network Based Models
- Cluster And Aggregate Based Models
- 3D Convolutional Based Models
  very computationally expensive!!

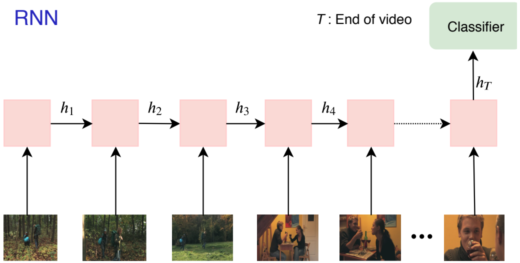
Video Classification Models

- Recurrent Network Based Models
- Cluster And Aggregate Based Models
- 3D Convolutional Based Models

Video Classification Models

Recurrent Neural Network (RNN)

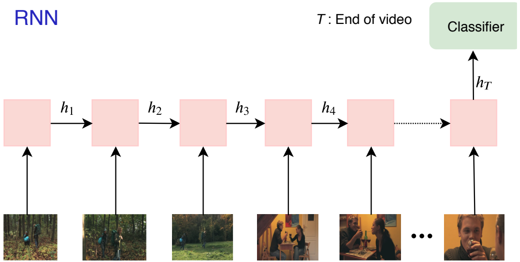
- Process video in a sequential way (frame-by-frame)



Video Classification Models

Recurrent Neural Network (RNN)

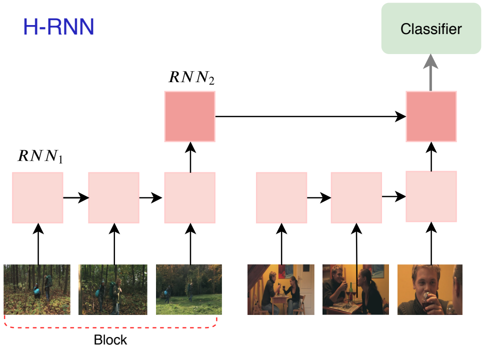
- Process video in a sequential way (frame-by-frame)
- At each step, maintain long-term history h of frames seen so far



Video Classification Models

Recurrent Neural Network (RNN)

- Process video in a sequential way (frame-by-frame)
- At each step, maintain long-term history h of frames seen so far
- Consider Hierarchical Recurrent Neural Network (H-RNN[a]) which:

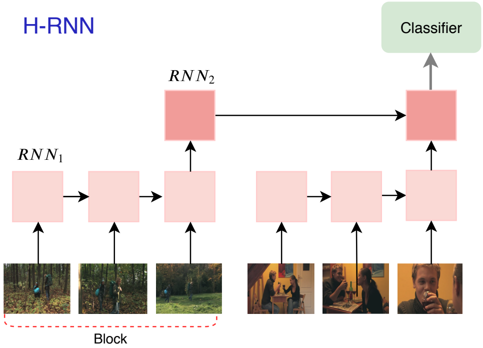


[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Video Classification Models

Recurrent Neural Network (RNN)

- Process video in a sequential way (frame-by-frame)
- At each step, maintain long-term history h of frames seen so far
- Consider Hierarchical Recurrent Neural Network (H-RNN[a]) which:
  - treats video as a sequence of blocks
  - memorize *longer* context

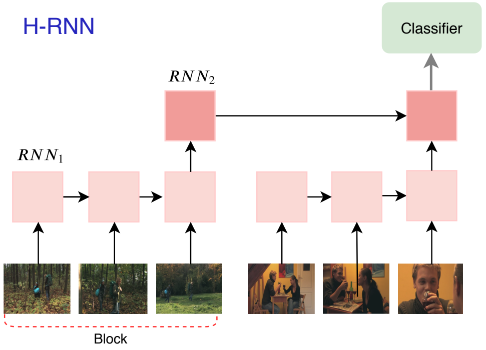


[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Video Classification Models

Recurrent Neural Network (RNN)

- Process video in a sequential way (frame-by-frame)
- At each step, maintain long-term history h of frames seen so far
- Consider Hierarchical Recurrent Neural Network (H-RNN[a]) which:
  - treats video as a sequence of blocks
  - memorize *longer* context



Note: Number of FLOPs $\propto$ length of frames processed

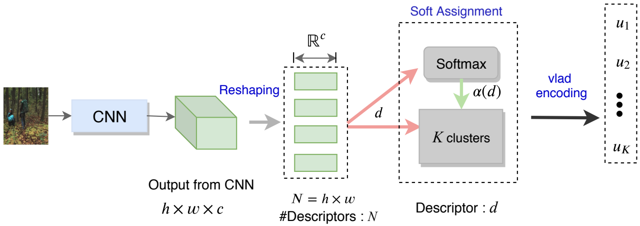
[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Video Classification Models

- Recurrent Network Based Models
- Cluster And Aggregate Based Models
- 3D Convolutional Based Models

Cluster And Aggregate Models

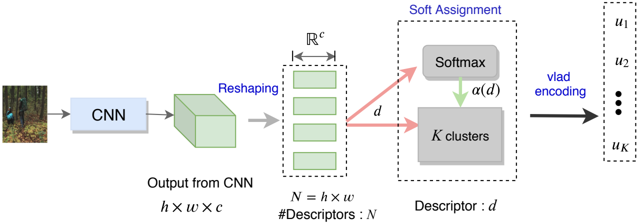
NetVLAD Scheme:



Video Classification Models

Cluster And Aggregate Models

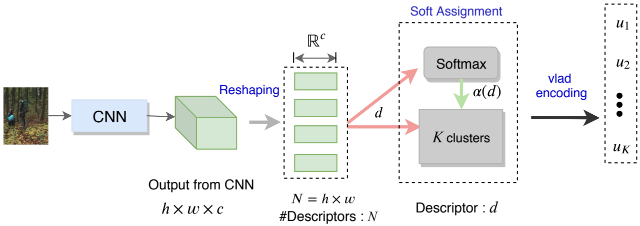
NetVLAD Scheme:



- Reshape *CNN* representation of a frame to obtain a descriptor *d*

Cluster And Aggregate Models

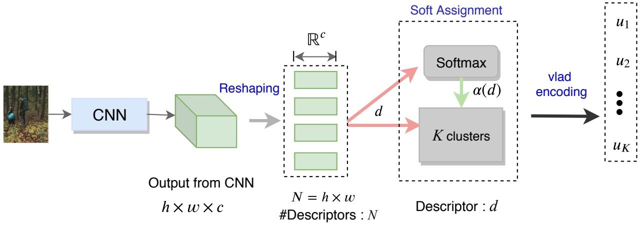
NetVLAD Scheme:



- Reshape *CNN* representation of a frame to obtain a descriptor *d*
- Soft-assignment of each cluster to the descriptor

Cluster And Aggregate Models

NetVLAD Scheme:

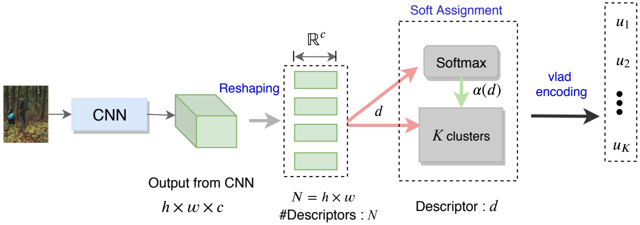


- Reshape *CNN* representation of a frame to obtain a descriptor d
- Soft-assignment of each cluster to the descriptor
- Stack *NetVLAD*[1] encodings u_k of each cluster to obtain output vector $v \in \mathbb{R}^{cK}$

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Cluster And Aggregate Models

NetVLAD Scheme:



- Reshape *CNN* representation of a frame to obtain a descriptor d
- Soft-assignment of each cluster to the descriptor
- Stack *NetVLAD*[1] encodings u_k of each cluster to obtain output vector $v \in \mathbb{R}^{cK}$
- Combine output vectors v from all frames to get a *video representation*

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Video Classification Models

Cluster And Aggregate Models

- Single video representation from NetVLAD is fed to classifier

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Video Classification Models

Cluster And Aggregate Models

- Single video representation from NetVLAD is fed to classifier
- NeXtVLAD[1]: A *memory-efficient* version of NetVLAD

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Video Classification Models

Cluster And Aggregate Models

- Single video representation from NetVLAD is fed to classifier
- NeXtVLAD[1]: A *memory-efficient* version of NetVLAD

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

Video Classification Models

Cluster And Aggregate Models

- Single video representation from NetVLAD is fed to classifier
- NeXtVLAD[1]: A *memory-efficient* version of NetVLAD
- However!

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

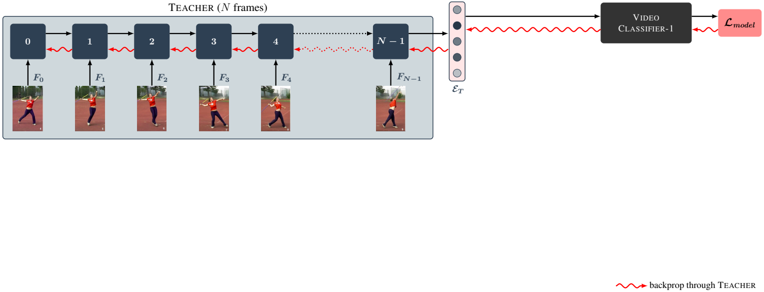
Cluster And Aggregate Models

- Single video representation from NetVLAD is fed to classifier
- NeXtVLAD[1]: A *memory-efficient* version of NetVLAD
- However! both of these models still look at every frame in the video
  $\therefore$ #FLOPs $\approx$ large, even with small memory footprint

[1] A large-scale video classification benchmark, Abu-El-Haija et. al, arXiv

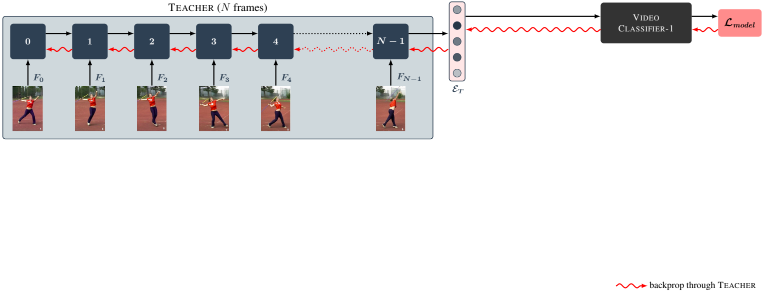
Proposed *Teacher-Student* Framework

- See-it-all *teacher* processes all the N frames in a video



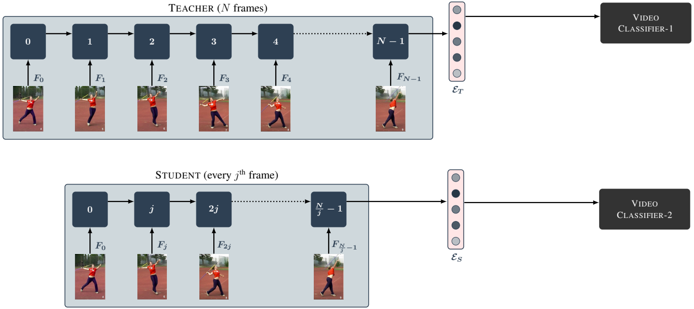
Proposed *Teacher-Student* Framework

- See-it-all *teacher* processes all the N frames in a video
- Trained using a standard multi-label classification loss $\mathcal{L}_{CE}$



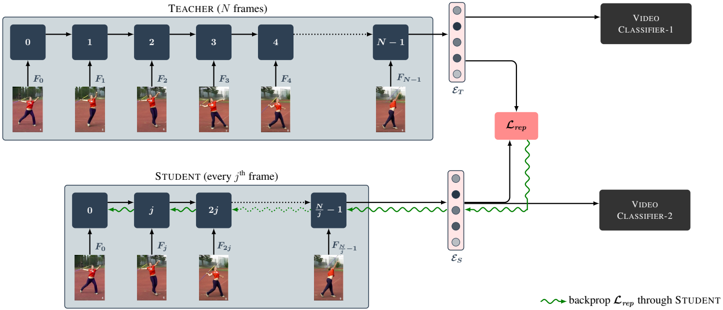
Proposed *Teacher-Student* Framework

- See-very-little *student* looks only at a fraction of frames *i.e.*, uniformly spaced k frames



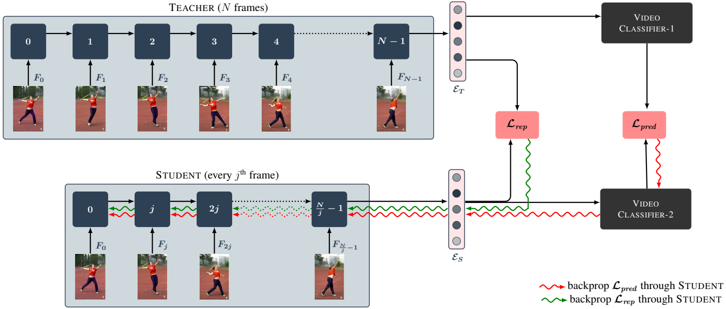
Proposed *Teacher-Student* Framework

- Train student to minimize difference between the video representations of *teacher* $\mathcal{E}_T$ and *student* $\mathcal{E}_S$ using $\mathcal{L}_{rep} = ||\mathcal{E}_T - \mathcal{E}_S||^2$



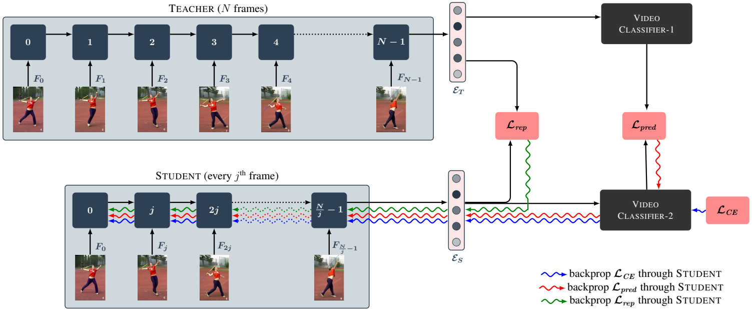
Proposed *Teacher-Student* Framework

- Train *student* to minimize the difference between the class probabilities predicted by the teacher $\mathcal{P}_T$ and the student $\mathcal{P}_S$ using $KL(\mathcal{P}_T, \mathcal{P}_S)$



Proposed *Teacher-Student* Framework

- Keep an eye on final performance with classification loss $\mathcal{L}_{CE}$



Results: Experiments on H-RNN

Hierarchical Recurrent Neural Network H-RNN[1]
Skyline Model with GAP:0.811, mAP: 0.414

Model	k=6		k=10		k=15		k=20		k=30	
	GAP	mAP	GAP	mAP	GAP	mAP	GAP	mAP	GAP	mAP
Model with k frames	**Baseline Methods**									
Uniform-k	0.715	0.266	0.759	0.324	0.777	0.350	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

[1] Hihi et. al., Hierarchical Recurrent Neural Networks for Long-Term Dependencies

Results: Experiments on H-RNN

Hierarchical Recurrent Neural Network H-RNN[1]
Skyline Model with GAP:0.811, mAP: 0.414

Model		k=6		k=10		k=15		k=20		k=30	
		GAP	mAP	GAP	mAP	GAP	mAP	GAP	mAP	GAP	mAP
Model with k frames		**Baseline Methods**									
Uniform-k		0.715	0.266	0.759	0.324	0.777	0.350	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**									
Serial	$\mathcal{L}_{rep}$	0.727	0.288	0.768	0.339	0.786	0.365	0.795	0.381	0.802	0.394
Serial	$\mathcal{L}_{pred}$	0.722	0.287	0.766	0.341	0.784	0.367	0.793	0.383	0.798	0.390
Serial	$\mathcal{L}_{rep}, \mathcal{L}_{CE}$	0.728	0.291	0.769	0.341	0.786	0.368	0.794	0.383	0.803	0.399
Serial	$\mathcal{L}_{pred}, \mathcal{L}_{CE}$	0.724	0.289	0.763	0.341	0.785	0.369	0.795	0.386	0.799	0.391
Serial	$\mathcal{L}_{rep}, \mathcal{L}_{pred}, \mathcal{L}_{CE}$	**0.731**	**0.297**	**0.771**	**0.349**	**0.789**	**0.375**	**0.798**	**0.390**	**0.806**	**0.405**

[1] Hihi et. al., Hierarchical Recurrent Neural Networks for Long-Term Dependencies

Results: Experiments on H-RNN

Hierarchical Recurrent Neural Network H-RNN[1]
Skyline Model with GAP:0.811, mAP: 0.414

Model		k=6		k=10		k=15		k=20		k=30	
		GAP	mAP	GAP	mAP	GAP	mAP	GAP	mAP	GAP	mAP
Model with k frames		**Baseline Methods**									
Uniform-k		0.715	0.266	0.759	0.324	0.777	0.350	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**									
Serial	$\mathcal{L}_{rep}$	0.727	0.288	0.768	0.339	0.786	0.365	0.795	0.381	0.802	0.394
Serial	$\mathcal{L}_{pred}$	0.722	0.287	0.766	0.341	0.784	0.367	0.793	0.383	0.798	0.390
Serial	$\mathcal{L}_{rep}, \mathcal{L}_{CE}$	0.728	0.291	0.769	0.341	0.786	0.368	0.794	0.383	0.803	0.399
Serial	$\mathcal{L}_{pred}, \mathcal{L}_{CE}$	0.724	0.289	0.763	0.341	0.785	0.369	0.795	0.386	0.799	0.391
Serial	$\mathcal{L}_{rep}, \mathcal{L}_{pred}, \mathcal{L}_{CE}$	**0.731**	**0.297**	**0.771**	**0.349**	**0.789**	**0.375**	**0.798**	**0.390**	**0.806**	**0.405**
Parallel	$\mathcal{L}_{rep}$	0.724	0.280	0.762	0.331	0.785	0.365	0.794	0.380	0.803	0.392
Parallel	$\mathcal{L}_{rep}, \mathcal{L}_{CE}$	0.726	0.285	0.766	0.334	0.785	0.362	0.795	0.381	0.804	0.396
Parallel	$\mathcal{L}_{rep}, \mathcal{L}_{pred}, \mathcal{L}_{CE}$	0.729	0.292	0.770	0.337	**0.789**	0.371	0.796	0.388	**0.806**	0.404

[1] Hihi et. al., Hierarchical Recurrent Neural Networks for Long-Term Dependencies

Results: Serial v/s Parallel

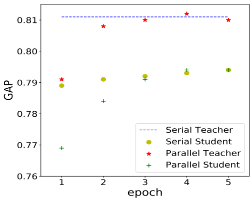


Figure: Training with $\mathcal{L}_{rep}$ only

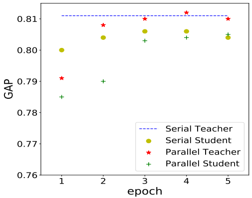


Figure: Training with $\mathcal{L}_{rep}$ and $\mathcal{L}_{CE}$

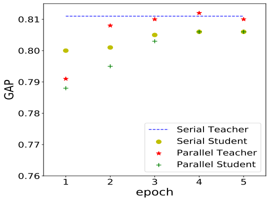


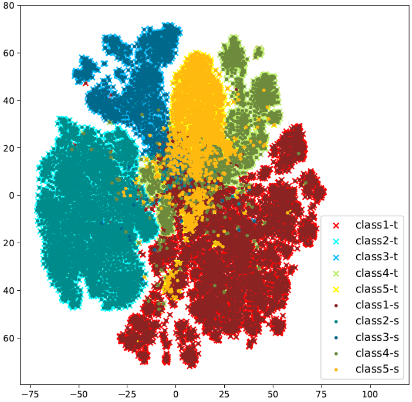
Figure: Training: $\mathcal{L}_{rep}$, $\mathcal{L}_{CE}$ and $\mathcal{L}_{pred}$

Performance comparison (GAP score) of different variants of *Serial* and *Parallel*

methods in *Teacher Student* training

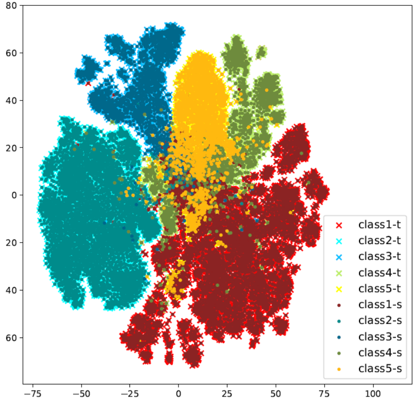
Results: Analysis

- TSNE-Plot of *student* $\mathcal{E}_S$ and *teacher* $\mathcal{E}_T$ encodings for top-5 most uncorrelated classes



Results: Analysis

- TSNE-Plot of *student* $\mathcal{E}_S$ and *teacher* $\mathcal{E}_T$ encodings for top-5 most uncorrelated classes



- Computation Cost v/s Frames

Model	Time (hrs.)	FLOPS (Billion)
Teacher-Skyline	13.00	5.058
$k = 30$	9.11	0.520
$k = 20$	8.20	0.268
$k = 10$	7.61	0.167

Inference: 89% FLOPs reduction with only 0.5-0.9% drop in performance

Results: Experiments on Non-Recurrent Models

NetVLAD[1]

Model: *NetVLAD*	k=10		k=30	
	mAP	GAP	mAP	GAP
Skyline			**0.462**	**0.823**
Uniform	0.364	0.773	0.421	0.803
Student	0.383	0.784	0.436	0.812

1

[1] Learnable pooling with Context Gating for video classification

Results: Experiments on Non-Recurrent Models

NetVLAD[1]

Model: *NetVLAD*	k=10		k=30	
	mAP	GAP	mAP	GAP
Skyline			**0.462**	**0.823**
Uniform	0.364	0.773	0.421	0.803
Student	0.383	0.784	0.436	0.812

NeXtVLAD[2]: compact version of NetVLAD

Model: *NeXtVLAD*	k=30 mAP	GAP	FLOPs (in Billion)
Skyline	0.464	0.831	1.337
Uniform	0.424	0.812	0.134
Student	0.439	0.818	0.134

1 2

[1] Learnable pooling with Context Gating for video classification

[2] NeXtVLAD: An Efficient Neural Network to Aggregate Frame-level Features for Large-scale Video Classification

Summary So Far

- Leverage **knowledge distillation** for **efficient** video classification with:

Summary So Far

- Leverage **knowledge distillation** for **efficient** video classification with:
  - recurrent models (*HRNN*)

Summary So Far

- Leverage **knowledge distillation** for **efficient** video classification with:
  - recurrent models (*HRNN*)
  - cluster-and-aggregate models (*NetVLAD*)

Summary So Far

- Leverage **knowledge distillation** for **efficient** video classification with:
  - recurrent models (*HRNN*)
  - cluster-and-aggregate models (*NetVLAD*)
- **Complementary** approach to **memory-efficient** clustering models (*NeXtVLAD*)

Summary So Far

- Leverage **knowledge distillation** for **efficient** video classification with:
  - recurrent models (*HRNN*)
  - cluster-and-aggregate models (*NetVLAD*)
- **Complementary** approach to **memory-efficient** clustering models (*NeXtVLAD*)
- Reduce FLOPs by $\sim$ **90%**, which are $\propto$ number of processed frames

Summary So Far

- Leverage **knowledge distillation** for **efficient** video classification with:
  - recurrent models (*HRNN*)
  - cluster-and-aggregate models (*NetVLAD*)
- **Complementary** approach to **memory-efficient** clustering models (*NeXtVLAD*)
- Reduce FLOPs by $\sim$ **90%**, which are $\propto$ number of processed frames
- Manages to use $\frac{1}{10}$ of frames with **0.5-0.9%** i.e., minimal drop in performance

Dynamic Selection of Frames

Question:

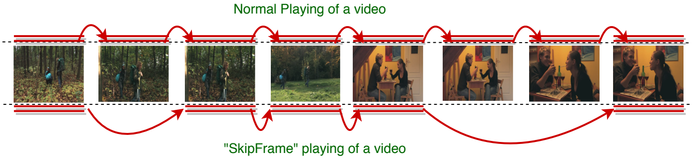
Dynamic Selection of Frames

Question: *"Does there exist a computationally efficient way in which we can dynamically select the frames through a video, which are different from uniformly sampled frames, and as a result of which, only relevant frames are presented to the classification network?"*

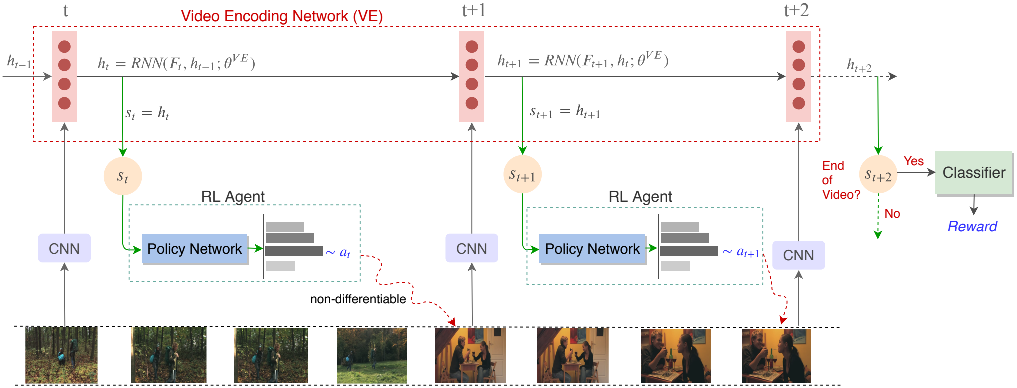
Dynamic Selection of Frames

Question: *"Does there exist a computationally efficient way in which we can dynamically select the frames through a video, which are different from uniformly sampled frames, and as a result of which, only relevant frames are presented to the classification network?"*

Yes ! *SkipFrame* comes to rescue



SkipFrame Architecture



Experiments: Rewards

Model	Reward-Design	Actions	GAP	mAP
Skyline	-	-	0.812	0.414
Uniform-10	-	-	**0.759**	**0.324**
Random-10	-	-	0.675	0.251
First-10	-	-	0.539	0.163
Middle-10	-	-	0.600	0.198
Last-10	-	-	0.267	0.067
SkipFrame	Delay-Reward	5-25	0.755	0.322
	Imm-Reward	5-25	0.738	0.286
SkipFrame	Delay-Reward	alt-5-25	0.742	0.291
	Imm-Reward	alt-5-25	0.739	0.288
SkipFrame	Delay-Reward: GAP	5-25	**0.764**	**0.341**

Table: Performance comparison of different variants of the *SkipFrame* models and the baselines. For all the variants of *SkipFrame*, we fix a budget of k=10 frames.

Experiments: Exploration in Frame Selection

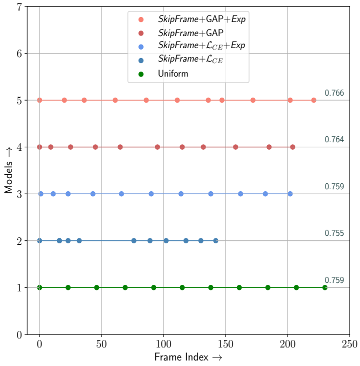


Figure: Comparison of frame-indices picked by different models.
Note: GAP score performance of each model is shown at the end of its series in the graph. The average number of frames in a video is 230.

- Exploration helps to better span a video

Experiments: Exploration in Frame Selection

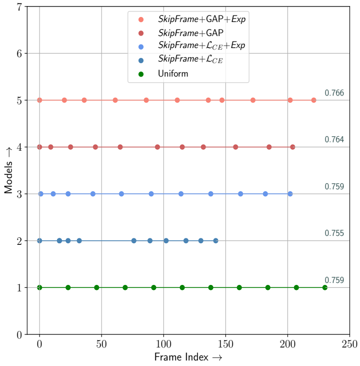


Figure: Comparison of frame-indices picked by different models.

Note: GAP score performance of each model is shown at the end of its series in the graph. The average number of frames in a video is 230.

- Exploration helps to better span a video
- GAP is a better reward signal

Experiments: Exploration in Frame Selection

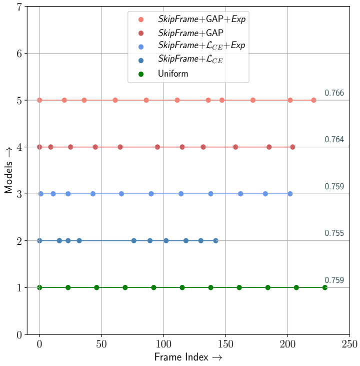


Figure: Comparison of frame-indices picked by different models.

Note: GAP score performance of each model is shown at the end of its series in the graph. The average number of frames in a video is 230.

- Exploration helps to better span a video
- GAP is a better reward signal
- Still, frames lie in close neighborhood of uniformally spaces

Experiments: Exploration in Frame Selection

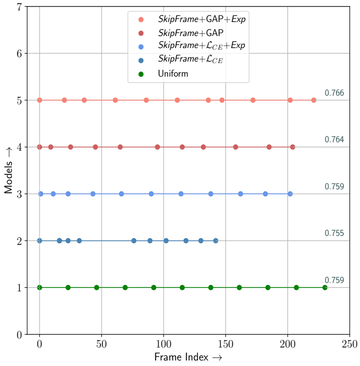


Figure: Comparison of frame-indices picked by different models.
Note: GAP score performance of each model is shown at the end of its series in the graph. The average number of frames in a video is 230.

- Exploration helps to better span a video
- GAP is a better reward signal
- Still, frames lie in close neighborhood of uniformally spaces
- *GAP* + Exp beats Uniform by slight margin of 0.6%

Experiments: Exploration in Frame Selection

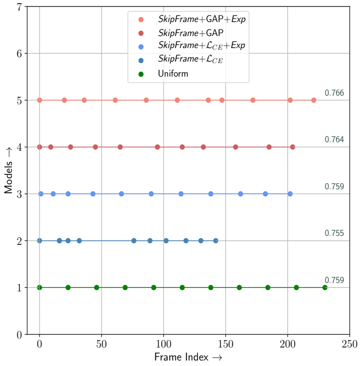


Figure: Comparison of frame-indices picked by different models.

Note: GAP score performance of each model is shown at the end of its series in the graph. The average number of frames in a video is 230.

- Exploration helps to better span a video
- GAP is a better reward signal
- Still, frames lie in close neighborhood of uniformally spaces
- *GAP* + Exp beats Uniform by slight margin of 0.6%
- *How exactly are labels spanned in a video?*

Experiments: Exploration in Frame Selection

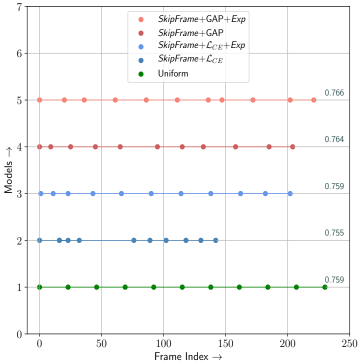


Figure: Comparison of frame-indices picked by different models.

Note: GAP score performance of each model is shown at the end of its series in the graph. The average number of frames in a video is 230.

Label Distribution ?

Figure: Sketch of a sample video with labels: *Travel*, *Nature*, *Train*

Experiments: Computation Cost

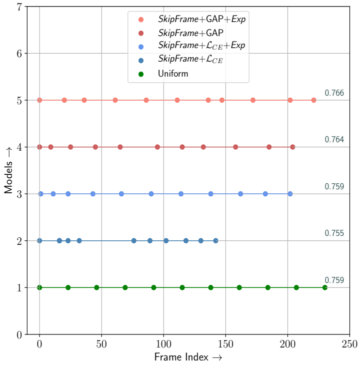


Figure: Comparison of frame-indices picked by different models.

Note: GAP score performance of each model is shown at the end of its series in the graph. The average number of frames in a video is 230.

Model	#Frames	#FLOPs
Skyline	230	5.058 B
Uniform	10	0.167 B
SkipFrame	10	0.167 B + 81.92 K

Table: Comparison of FLOPs of different models. Here, B: Billion and K: Thousand are the order of #FLOPs

Takeway

- Propose a method to reduce the computation time for video classification using the idea of distillation.
- Introduce a *student* network which only processes k frames of the video
- Train the *student* by matching:
  1. final representation produced by the *student* and the *teacher*
  2. output probability distributions produced by the *student* and *teacher*
- Student outperforms the baseline by a significant margin
- Reduce the computation time by 30% while giving an approximately similar performance as the teacher network
- Further analysis on *dynamic* selection of frames, unlike *uniform sampling*
- Establish picking *uniformly spaced* frames as easier and efficient strategy than *dynamic* selection

Thanks

Any questions?