APPROVAL SHEET

Title of Thesis: Video Summarization using Unsupervised Methods

Due to the increasing volume of the video data uploaded daily on the web through prime sources including social media, Youtube, and video sharing websites, video summarization has emerged as an important and challenging problem in the industry. Video summarization finds applications in various domains like consumer industry and marketing, generating a trailer for movies, highlights for different sports events. As a result, an efficient mechanism for extracting important video contents is the need to deal with a large amount of videographic repositories. We present a novel unsupervised approach to generate video summaries using simpler networks like VGG and ResNet instead of using complex networks i.e. LSTM and RNN. Video summarization and Image captioning are two completely different and independent tasks, yet we propose an approach that considers generating summaries using a feature space produced as a result of the image captioning of a video. Our main idea is generating short and informative summaries in a completely unsupervised manner using basic and traditional clustering technique modeled jointly with the video captioning framework NeuralTalk2. We conducted experiments in different settings with SumMe and TVSum datasets. Our approach achieved state-of-the-art results for SumMe dataset with an F-score of **35.6%**. For TVSum dataset, we obtained an F-score of **43.5%**.

Name of Candidate: Akanksha Nanaji Bhosale
Masters of Science, 2018

Thesis and Abstract Approved: _____

Dr. Tim Oates
Professor
Department of Computer Science
and Electrical Engineering

Date Approved: 7/23/2018

ABSTRACT

Title of Thesis: Video Summarization using Unsupervised Methods

Akanksha Bhosale, M.S. Computer Science, 2018

Thesis directed by: Dr. Tim Oates, Professor
Department of Computer Science and
Electrical Engineering

Due to the increasing volume of the video data uploaded daily on the web through prime sources including social media, Youtube, and video sharing websites, video summarization has emerged as an important and challenging problem in the industry. Video summarization finds applications in various domains like consumer industry and marketing, generating a trailer for movies, highlights for different sports events. As a result, an efficient mechanism for extracting important video contents is the need to deal with a large amount of videographic repositories. We present a novel unsupervised approach to generate video summaries using simpler networks like VGG and ResNet instead of using complex networks i.e. LSTM and RNN. Video summarization and Image captioning are two completely different and independent tasks, yet we propose an approach that considers generating summaries using a feature space produced as a result of the image captioning of a video. Our main idea is generating short and informative summaries in a completely unsupervised manner using basic and traditional clustering technique modeled jointly with the video captioning framework NeuralTalk2. We conducted experiments in different settings with SumMe and TVSum datasets. Our approach achieved state-of-the-art results for SumMe dataset with an F-score of 35.6

Keywords: Video Summarization, NeuralTalk2, K-means clustering

Video Summarization using Unsupervised Methods

by
Akanksha Bhosale

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
M.S. Computer Science
2018

Dedicated to my friends and family. To my advisor Dr.Tim Oates, who guided me to my destination

ACKNOWLEDGMENTS

I would like to thank my research adviser and mentor Dr. Tim Oates for guiding me through my research, and for encouraging me to pursue and tackle interesting, challenging problems. From the start, he has been a great source of inspiration for me and has guided me through my failures. I would like to thank him for being patient with me and having faith in me. I would also like to thank him for giving me such an interesting topic as my thesis. I would like to thank Dr. Finin and Dr. Hamed for being part of my committee.

I am fortunate to have had so many amazing friends throughout my graduate studies. I would like to thank some of my amazing friends in the US especially Rahul, Aparna, Suraj, Siddhi. I'd like to thank Sunil, Ashwin for all the guidance and discussions.

Special thanks to my parents and my brother for their immense support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Videos and visuals have emerged as a critical source of information in various domains ranging from digital marketing, consumer markets, service and product industries. Nowadays, so much visual information is available on the internet that while making a decision people need an informative and meaningful summary. In case of surveillance systems, videos are often hours in length resulting in the need to find the distinct events that convey meaningful information. So automatic mechanisms generating highly informative summaries provide an efficient way to browse through long length videos. Due to the large volume of videos, video summarization is a key research topic nowadays.

Video summarization is challenging because one must decide which parts of the video are important and how to decide whether they should be included in the synopsis. As a result, a lot of experimentation is in progress dealing with the intricacies involved in extracting the parts of the video and developing the matrics proving the usefulness of those extracted parts of the video. Thus, a novel summary would concisely depict all distinguishable and significant events of the original video as if we went through the whole actual video.

Different learning techniques have produced a quality summary in different ways.

Some of the ways include keyframe selection or frame importance score or keyshots. Keyframe selection includes picking the frames of the original video providing the gist of the whole storyline, while frame importance score give a score to every frame of the video and, as per the criteria, frames satisfying the criteria are choosen constituting the short summary. In the case of keyshots, a set of subshots obtained through temporal segmentation of the video that contain prime events are extracted. We concentrate on the keyframes and keyshots form of summary generation.

The main two approaches to solve the video summarization problem are categorized into supervised and unsupervised methods. Supervised techniques have become popular as they directly learn from prior knowledge from manually designed summaries fed to the model in terms of importance score or frame level importance and they output the frames matching with the video content. Thus they take in some form of annotation of either key-frames or keyshots based on features like motion, aesthetics, or attention mechanism. But modeling the problem with the supervised approach is not scalable. Videos belonging to different categories are diverse and their increasing complexity over a period of time poses limitations like it is difficult to have annotations for all videos.

Unsupervised learning approaches [2] do not require any prior knowledge of the videos[3]. The key challenge with this approach is, without any predefined rules or knowledge, how to decide whether a particular frame should be selected. Also, it is essential to understand the visual and semantic similarity of the video frames. Two visually similar frames appearing at more temporal distance in a video could convey different meanings. There exists techniques and ways to deal with the problem in an unsupervised manner but they typically involve the use of complex Long Short Term Memory (LSTM) networks or a kind of auto encoder-decoder architecture [4]. We concentrate on achieving state-of-the art results without any complex networks in an

unsupervised manner using clustering based techniques.

Our main contribution is jointly modeling two completely independent tasks i.e., Image captioning and video summarization. We perform summarization using a model that was learned for an image captioning task. The key point is image captioning task makes video summarization easier.

In this work, we investigate how efficiently we can summarize videos in an unsupervised manner by utilizing a model trained on image captioning. We use NeuralTalk2 [1] for the image captioning task trained on the MSCOCO dataset. Then we experiment on the penultimate layer of NeuralTalk2. In this, we analyze the impact of resnet101, VGG11 and VGG19 as the penultimate layer of the image captioning framework. We further generate synopses using traditional and simple k-means clustering. To deal with redundancy and maximize diversity, we experiment with cluster density using a binning approach. We further explore the relation between cluster size and evaluation metric which is explained in detail below.

The rest of this thesis report work is organized as follows. Chapter 2 reviews background and related work and techniques of video summarization. Chapter 3 describes the approach and implementation details of our work. Chapter 4 discusses the datasets used for experimentation. Chapter 5 describes the experiments and the results obtained. We examine the image captioning task in different settings and analyze the impact of different number of VGG layers. Then, we conclude in Chapter 6.

Chapter 2

BACKGROUND AND RELATED WORK

In this section, we review work on video summarization methods and approaches.

The two main techniques for video summarization are 1. Supervised learning where knowledge in the form of frame importance scores or keyframes is used during training and, 2. Unsupervised learning which does not take into account any prior knowledge to select frames or subshots.

Ke Zhang [4] presented a supervised technique for summarization that uses an LSTM and an MLP. Long short term memory (LSTM) is widely used for summarization due to its efficiency in modeling long range dependencies or inter-dependencies between past and the future. LSTMs are also flexible in modeling sequential structures. Thus, the summary is built by predicting the likelihood that a particular frame should be included in the summary or how important the current frame is from a summary point of view. They enhance the performance of the proposed system by using determinantal point process (DPP) that deals with redundancy in the generated summary. DPP calculates pairwise frame-level repulsiveness. As a result, generated summary contains diverse content. The algorithm takes annotations in the form of either frame importance scores or binary indicators telling whether a frame is selected in the summary. They experimented with canonical, augmented and transfer

variants.

Yingbo Li [5] proposed a technique for multi-video summarization called Video-MMR. They extended the classical algorithm 'Maximal Marginal Relevance' for summarizing in the text to video domain. The selection criterion for a frame to be a keyframe is that it's visual content should be similar to that of the original video frames but not matching with any of the frames those are already a part of the generated summary. By maximizing the marginal relevance between the visual content of the summary S and the video V, a summary is generated. They presented two strategies of Video MMR summarization as follows:

1. Global Summarization involves summarizing all individual videos in a set at a time. It models both inter and intra relevance of independent video simultaneously.
2. Individual Summarization involves summarizing a single video at a time and concatenating the summaries of individual videos of the set. During concatenation, duplicates are retained in the final summary.The similarity of two video frames is derived from the similarity of visual word histograms. They compare global and individual summarization strategies.

Bin Zhao [6] worked on summarizing consumer videos. The videos of surveillance cameras and monitoring devices contains billions of hours. Such video footage is unstructured as it is not manually edited or structured, in contrast to movies or sports videos. They proposed LiveLight - online video highlighting, a way of generating concise and meaningful summaries. Given an input video, it is segmented temporally, then LiveLight scans through the stream of an input video. After the first few segments of the video are processed, a dictionary is built. For further segments of the video, the dictionary is updated. For any new video, LiveLight tries to sparsely rebuild previously unseen video segments. It performs this step using group sparse coding. During this reconstruction process, if LiveLight encounters a segment

which is already present in the dictionary, it does not update the current dictionary. Thus, reconstruction error indicates the content of a current segment already exists in the summary is minimized. This method is proposed to work on videos of hours in length or even streaming videos and summary generation time is almost linear in video length.

Yale [7] summarized videos using titles. The thought process behind their work is that, the most descriptive indicator of all different and interesting events occurring in a video is the title. The suggested framework selects shots of the video containing the result of the title based on image search. Thus, the title serves as a strong prior in generating the summary. They present co-archetypal analysis to deal with the noise or variance present in the summarized short video. Co-archetypal analysis concentrates on patterns similar to both the video and images. Co-archetypes are found patterns. The main contribution of the work is the co-archetypal analysis technique that learns joint-factorial representation of a video and images where archetypes are provided.

Khosla [8] developed a summarization technique for user-generated videos taking into consideration the fact that the natural tendency of humans while capturing an event is to focus on the objects of interest. So, if we use web images as a prior for the process of summarization, it could result in generalized summarization mechanism. Thus, selecting a part of the video containing similar sets of objects outputs maximally informative short descriptions. In order to use web images as a prior, they take the unlabelled corpus of the images and classify them into subclasses by clustering. Each cluster represents a canonical viewpoint. Also, the classifier is learned for each subclass. Subclass models are improved with the help of unlabelled images' assigning every video frame to one of the sub-classes, and it is optimized by repeatedly processing both video frames and web images. For a test video, frames of the video are assigned to one of the clusters. The subclasses are ranked and all represen-

tative frames out of the top k centroids constitute the final summary. To evaluate performance, they have proposed a framework based on crowd-sourcing provided by Amazon Mechanical Turk.In this, 5% of the images are sampled, and with cropping and rescaling a few variants are generated. These variants are included in the summary. So if the workers select both original and perturbed frames then all previous annotations of that worker are excluded. Thus, they provide solid evaluation and comparison of different frameworks.

All reviewed approaches use either web images or titles or any other kind of prior knowledge. Our work simplifies the video summarization task and works directly on the input videos and clusters every video resulting in the generated summary. The proposed approach saves efforts in dealing with the complex networks like LSTMs and RNNs. Only clustering results in the state-of-the-art results for SumMe dataset and for TVSum dataset, the results are comparable to published scores.

Chapter 3

APPROACH

In this section, we introduce the methods and the overall algorithm used for summarizing videos. First, we state the problem then review the Neuraltalk2 image captioning module and the basic framework of summarization methods. Then we describe our overall algorithm. Further, we elaborate on the impact of the VGG layer used as the penultimate layer of the NeuralTalk2 framework. Next, we discuss the generation of shot-based summaries for videos.

3.1 Problem Statement

A sequence of frames in input video is denoted by x = x1,x2,....,xt where xt is the frame level feature extracted at time t. The input to the algorithm is the sequence of frames and the output is keyframes selected as a part of the summary. The keyframes are a subset of the individual frames. We then convert keyframes to temporal keyshots for evaluation. This conversion is required because testing ground truth for our test datasets, SumMe and TVSUM, is in the form of interval-based important subshots, i.e., users scored the video frames on a scale of 1 to 5 that indicates the frame level importance score. The basic unit of representation throughout the implementation is a frame.

3.2 NeuralTalk2

Karpathy [1] presented NeuralTalk2, a deep neural network model that generates dense descriptions of images. The strength of the proposed model lies in its design that simultaneously understands the contents of the images and presents its description in sentences. The framework learns from training data consisting of a set of images and their annotation in form sentence description. NeuralTalk2 consists of two models described below.

Alignment Model

The basic intuition behind this is alignment between image regions and sentence descriptions. When people describe an image, every word they provide points to some part of the image. In this model, objects in every image are detected using Region Convolutional Neural Network where the CNN is pretrained on ImageNet. Then, considering the top detected locations within an image its representation vector of 4096h dimensions is formed where h indicates the size of the multimodal embedding space. Thus, every image is represented in the form of a h-dimensional vector. Next, words in the sentences are represented in the same space. Word representations are computed using a Bidirectional RNN that considers ordering and understands context information of the words. Thus, a correspondence between sentences and image regions is inferred. Bidirectional RNNs consist of two layers of processing that independently compute word representation from left-to-right and right-to- left. The function of both words at that location gives the final representation of the word. In this way, every image and sentence is transformed into common h-dimensional vector space.

Multimodal RNN

Novel descriptions are generated using a multimodal RNN. It takes a set of input vectors (x1,x2...xt) corresponding to texual description and image pixels. The output of this setting is the probabilities of words in the dictionary. At the first iteration, we also provide the image context vector. RNN then predicts the next word by combining a word and previous context. The working of multimodal RNN for generating descriptions can be depicted as per figure 1.

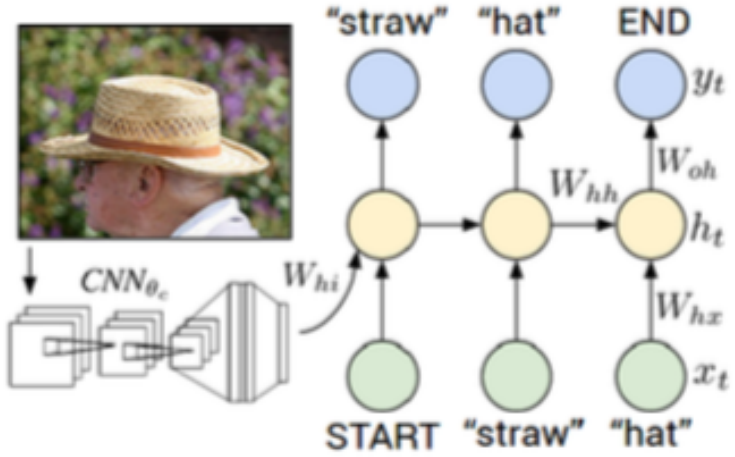

FIG. 3.1. Diagram of our multimodal Recurrent Neural Network generative model. The RNN take a word, and the context from previous time steps, and defines a distribution over the next word in the sentence. The RNN is conditioned on the image information at the first step. START and END are special tokens [1]

3.3 Overall Algorithm

In this section, we describe our video summarization algorithm.

Input:

$$X = \{X_j \quad | \quad j \in \{1, 2, ..., L\}\}$$

be a set of feature vector extracted from the input video where L is the length of the video, number of clusters K

Output: a subset of video segments

$$S = \{S_k \quad | \quad k \in \{1, 2, ..., K\}\}$$

1. Process video frames through NeuralTalk2 with resnet101 to obtain feature vector of every image of the video.

2. Write penultimate layer's output i.e. image feature vector of 1*1024 dimension

3. Apply k-means clustering to feature vector space. 4. We are minimizing the sum of the Euclidean distance of all frames to their closest cluster center

$$x \in S$$

5.The objective function minimized is:

$$F(S) = \sum_{x \epsilon X} \min_{s \epsilon S} ||x - s||_2^2 \tag{3.1}$$

6. Convert obtained keyframes to keyshots to get summary result as described in section 3.5.

3.4 Learning/Implementation Details

First, we train NeuralTalk2 ImageCaptioning on the MSCOCO dataset containing 123,287 images. We extract features from ResNet101 and save them to an HDF5 file for faster processing. HDF5 files are efficient in storing numerical data for large datasets. HDF5 files has two objects groups and datasets where groups are like folders and datasets correspond to actual data in array format. Then, we generate frames of videos using FFmpeg with fps of 25. After that, SumMe and TVSum video frames are passed through the learned model trained on ImageCaptioning. The feature vectors obtained through this are further processed by K-means as described in Algorithm 1.

3.5 Shot-based Summaries

Our implementation predicts keyframes i.e. a subset of the isolated frames indicating important and interesting events that provide a highly informative short summary of the video.But for evaluation we convert them into keyshots using the procedure described below [9]

1. First, segment a video temporally using kernel temporal segmentation (KTS) [7]. The formed intervals are symantically disjoint.

2. If a segment contains one of the keyframes then we encode all frames in that segment as 1 else 0. This could result in producing too many keyshots.

3. To deal with too many generated keyshots, before performing step 2, we assign ranks to every segment in decreasing order of the number of keyframes in that subshot divided by the video length.

4. The step 2 is performed only on those subshots such that the total duration of keyshot-based summary is less than 15% of original video duration.

Chapter 4

DATASETS

This chapter describes the datasets we used for the video summarization task.

4.1 MSCOCO

Microsoft Common Objects in Context[10] is a richly-annotated dataset containing images depicting everyday events of common objects. It contains almost 123,000 images that could be categorized into 91 object categories. The key feature of this dataset is annotation of instance level segmentation. Though COCO has relatively few categories, the number of instances per category is large. Also, labelling is done through Amazon's Mechanical Turk where humans are asked to classify every image they are given into one of the categories. For this task, 91 categories are divided into 11 super-categories. Each image is annotated with 5 sentences using AMT.

In NeuralTalk2, we use 5000 images for testing and validation. The 11 main categories for videos are:

1. Person And Accessory
2. Animal
3. Vehicle
4. Outdoor Object
5. Sports
6. Kitchenware
7. Food
8. Furniture
9. Appliance
10. Electronics
11. Indoor Objects

4.2 SumMe Dataset

The SumMe dataset benchmark proposed by [11] contains 25 videos. The key feature of this dataset is that it is the first dataset which is annotated with human score for different segments of a video instead of keyframes. The dataset is comprised of videos covering events, sports and holidays. The videos are of different interesting events like base jumping, bike polo, scuba, cockpit landing, cooking, playing on water slide, statue of liberty, air force one, fire Domino, car over camera, paintball, etc. All of the videos are completely raw and unedited. The duration of the videos ranges from 1 to 6 minutes.

4.3 TVSum Dataset

The TVSum dataset [7] consists of 50 Youtube videos of 10 different categories. This dataset is the result of the work done by [7] for summarization using titles. Actually, the categories of the TVSum videos are from TRECVid Multimedia Event Detection (MED). So using each category of MED as a search query, 5 videos per category satisfying following criteria are selected :

1. Videos of duration 2 to 10 minutes.
2. Videos with creative commons license
3. Videos with more than one shot
4. Videos whose title depicts interesting visual topics in the videos.

Also via crowdsourcing, every video's shot-level importance scores were obtained. The different genres of videos includes how-to's, documentories, news, and user generated events. The 10 main categories considered are shown in figure [7]:

FIG. 4.1. TVSum Videos are grouped into query categories. For each video, we show its title, thumbnail image, YouTube unique video identifier, duration, and genre.

FIG. 4.2. Videos are grouped into query categories. For each video, we show its title, thumbnail image, YouTube unique video identifier, duration, and genre.

Chapter 5

EXPERIMENTS AND RESULTS

In this chapter, we describe our experiments that show the utility of our proposed idea on two different datasets, i.e., SumMe and TVSum. First we describe baselines, and then elaborate on features and evaluation. Then we present the results along with the datasets used in different experimental settings.

We concentrate on keyframe as our summary output. Basically it is a binary indicator. So if a frame is important enough to be a part of the summary then it is represented as 1 else 0.

5.1 Baselines:

We use following baselines in our evaluation.

5.1.1 Random Sampling

This involves randomly selecting frames of both the SumMe and TVSum datasets. Ideally, the results for random sampling would be poor. This baseline is a good indicator to show how more, complex approaches perform.

5.1.2 Threshold Binning

This baseline is introduced where we convert keyframes to keyshots. We first temporally segment the video into semantically disjoint segments using kernel temporal segmentation explained in [12]. Then, in every segment if we find one of the keyframes, we make all frames of that subshot a keyframe. This typically results in generation of too many keyshots. To deal with this, we use threshold binning. In every subshot, if a certain number of original keyframes are present, only then do we include the subshot in the summary.

5.2 Evaluation Metric

For quantitative analysis, we use F-score as a measure of performance. In order to compare with state-of-the- art results, the metric used in [4] is also a part of the evaluation. Let A be our keyshot- based summary generated through clustering and B be the user summary available. We compute precision and recall defining the temporal overlap between A and B as below:

$$Precision = \frac{\text{duration of overlap between A and B}}{\text{total duration of A}} \tag{5.1}$$

and,

$$Recall = \frac{\text{duration of overlap between A and B}}{\text{total duration of B}} \tag{5.2}$$

Finally, the harmonic mean F-score is used as a metric of evaluation, which is calculated as :

$$F - score = \frac{2 * \text{precision} * \text{recall}}{precision + recall} \tag{5.3}$$

5.3 Results and Analysis

5.3.1 Random Sampling

As described earlier, in random sampling we pick frames randomly from input video and without applying our clustering based technique. The following table presents results of random sampling:

Dataset	**size of keyshot**	**F-score**
SumMe	30	0.025
	50	0.036
	70	0.048
	100	0.061

Table 5.1. F-score of the **random baseline** for the SumMe dataset. We see that an increase in cluster size results in F-score increase with a maximum of 6.1 % obtained for K = 100.

.

Dataset	**size of keyshot**	**F-score**
TVSum	100	0.016
	200	0.026
	300	0.04
	400	0.052
	500	0.062
	600	0.068

Table 5.2. F-score of the **random baseline** for the TVSum dataset. We see that an increase in cluster size results in F-score increase with a maximum of 6.8 % obtained for K = 100.

.

Thus, through random sampling in case of both datasets the maximum F-score we get is 6.8.

5.3.2 Binning Approach

Threshold Binning approach does not convert keyframes to keyshots.

Threshold	**3**	
Dataset	**size of cluster**	**F-score**
SumMe	30	0.176
	50	0.293
	70	0.316
	100	**0.343**

Table 5.3. F-score of the **binning** baseline for the SumMe dataset with **Threshold = 3**

Threshold	**5**	
Dataset	**size of cluster**	**F-score**
SumMe	30	0.046
	50	0.233
	70	0.315
	100	**0.316**

Table 5.4. F-score of the **binning** baseline for the SumMe dataset with **Threshold = 5**

Threshold	**2**	
Dataset	**size of cluster**	**F-score**
TVSum	100	0.051
	200	0.16
	300	0.25
	400	0.349
	500	0.396
	600	**0.444**

Table 5.5. F-score of the **binning** baseline for the TVSum dataset with **Threshold = 2**

Threshold	**3**	
Dataset	**size of cluster**	**F-score**
TVSum	100	0.042
	200	0.082
	300	0.166
	400	0.238
	500	0.31
	600	**0.372**

Table 5.6. F-score of the **binning** baseline for the TVSum dataset with **Threshold = 3**

Threshold acts as a constraint and increasing the value of the threshold results in decreasing the F-score value as fewer shots are picked to compute the quality of the summary.

5.3.3 K-means clustering based F-score and published scores

This section explores the performance of our video summarization approach performed using K-means clustering and modeled jointly with the image captioning task using NeuralTalk2.

Dataset	**size of cluster**	**F-score**	**Published**
SumMe	30	0.319	
	50	0.334	
	70	0.34	
	100	**0.356**	
			supervised - 0.418±0.5[4]
			unsupervised - **0.266[5]**

Table 5.7. F-score of **K-means** clustering for the SumMe dataset

Dataset	size of cluster	F-score	Published
TVSum	100	0.397	
	200	0.4	
	300	0.412	
	400	0.427	
	500	0.431	
	600	**0.435**	
			supervised - 0.587±0.4[4]
			unsupervised - **0.46**[**6**]
			unsupervised - **0.36**[**8**]
			unsupervised - **0.50**[**7**]

Table 5.8. F-score of **K-means** clustering for TVSum dataset

For the SumMe dataset, we achieved state-of-the art results in all unsupervised techniques with a maximum F-score of 35.6 %. For the TVSum dataset, our results are comparable to the best published unsupervised approaches, but we use far less side information with an overall much simpler method.

5.3.4 Penultimate layer of NeuralTalk2: VGG

We experimented with the ResNet101 layer used by default in NeuralTalk2 and replaced it with VGG11 and VGG19. The purpose of this experiment is to analyze the impact of changing network architecture and also to identify patterns between number of layers used and the resulting F-score.

Dataset	**size of cluster**	**F-score**
SumMe	30	0.359
	50	**0.366**
	70	0.353
	100	0.337

Table 5.9. F-score of the clustering-based technique with **VGG11** for the SumMe dataset

Dataset	**size of cluster**	**F-score**
TVSum	100	0.383
	200	0.401
	300	0.435
	400	0.426
	500	**0.436**
	600	0.431

Table 5.10. F-score of the clustering-based technique with **VGG11** for the TVSum dataset

Dataset	size of cluster	F-score
SumMe	30	0.361
	50	**0.374**
	70	0.353
	100	0.361

Table 5.11. F-score of the clustering-based technique with **VGG19** for the SumMe dataset

Dataset	size of cluster	F-score
TVSum	100	0.396
	200	0.403
	300	0.415
	400	0.425
	500	0.43
	600	**0.432**

Table 5.12. F-score of the clustering-based technique with **VGG19** for the TVSum dataset

5.3.5 Comparison of ResNet and VGG configurations

SumMe Dataset

We get the maximum F-score of 35.6 % with $K = 100$ in case of ResNet101 while with VGG11, 36.6 % is the maximum F-score when $K = 50$. Also, for VGG19 we

get the F-score of 37.4 % when $K = 50$. The above comparison indicates that as we decrease the number of layers of the internal network configuration of NeuralTalk2, we get increased performance even with fewer clusters. This is evident through the F-score of 36.6% with $K = 50$ in case of VGG11. If we compare the two variants of VGG, it is clear that VGG19 outperforms VGG11 with same cluster size $K = 50$. Overall, VGG results in better performance than ResNet. Also, VGG19 outperforms all configurations.

TVSum Dataset

We get the maximum F-score of 43.5 % with $K = 600$ in case of ResNet101 while with VGG11, 43.6 % is the maximum F-score when $K = 500$. For VGG19, the maximum F-score is 43.2 % with $K = 600$. The overall maximum performance is with VGG11.

Thus, for both datasets we get better results with the VGG network than ResNet.

5.3.6 Precision-Recall Curve

The precision-recall curve validates the quality of our results. The value of precision is the fraction of the relevant results obtained over retrieved results and recall is the fraction of the relevant results obtained over total number of relevant results.[13]

We have taken cluster size as the parameter to generate a precision-recall curve and considered area under mean precision-recall of total no.of videos for the cluster size. Precision tells how many frames selected as a part of the summary match with the ground truth frames, i.e, the user summary. Recall tells how many frames of the ground truth match with the frames selected as a part of the synopsis using our clustering-based technique.

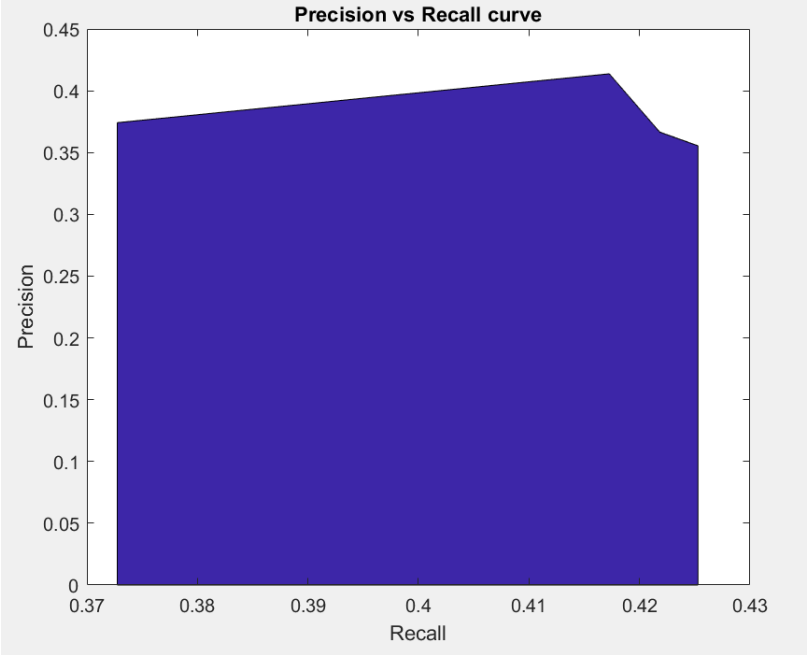

FIG. 5.1. Precision-Recall curve for the SumMe Dataset

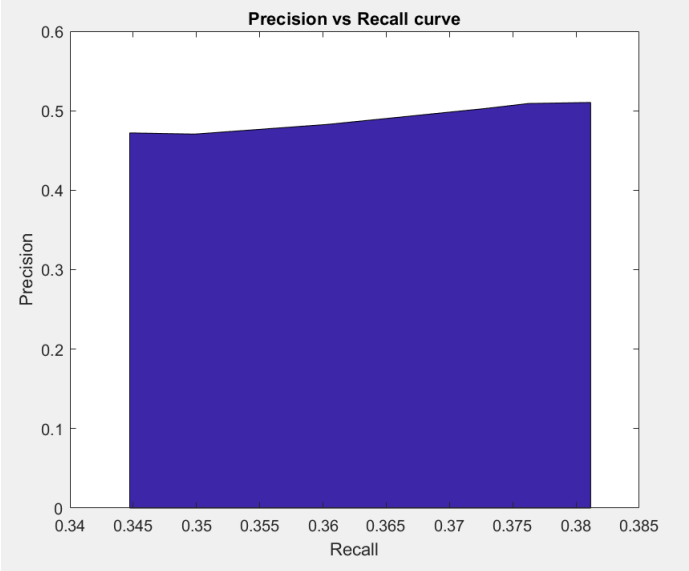

FIG. 5.2. Precision-Recall curve for the TVSum Dataset

Analysis

As shown in above figure for the SumMe dataset, the value of precision increases first and then decreases as cluster size increases. This means that the match between selected frames and ground truth falls over a period of time while the increasing value of recall indicates that we are consistently getting a summary rich in quality. In case of the TVSum dataset, both precision and recall increase as the number of clusters increases, clearly showing that the match between selected frames of the summary and the user summary is of high quality.

5.4 Our Visual Results :

This section shows results obtained through our proposed clustering technique for a few Youtube videos and their respective user summaries. These videos are of different genres including news, commercials, TV-shows, sports and cartoons, and the total video durations range from 1 to 10 minutes.

FIG. 5.3. K-means video summary

FIG. 5.4. User video summary

FIG. 5.5. K-means video summary

FIG. 5.6. User video summary

FIG. 5.7. K-means video summary

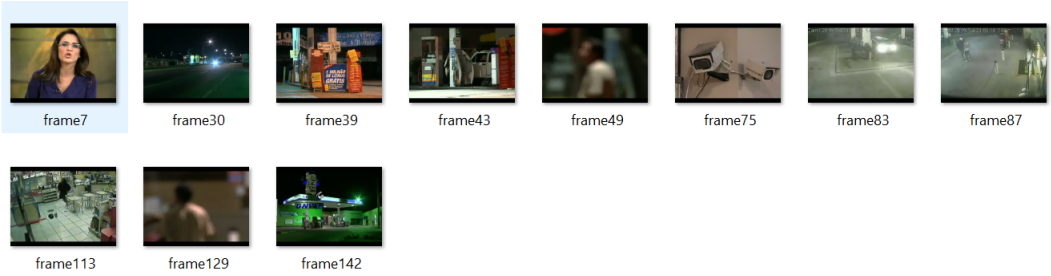

FIG. 5.8. User video summary

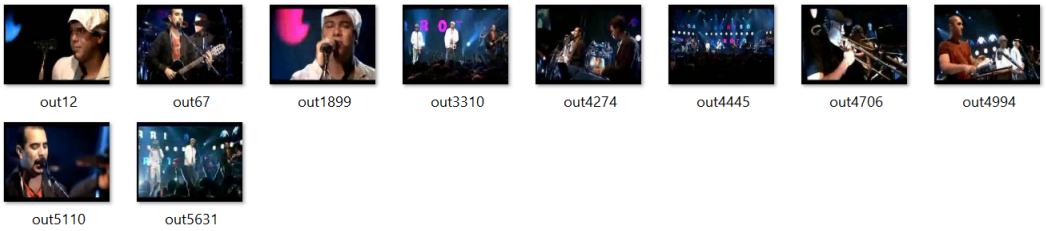

FIG. 5.9. K-means video summary

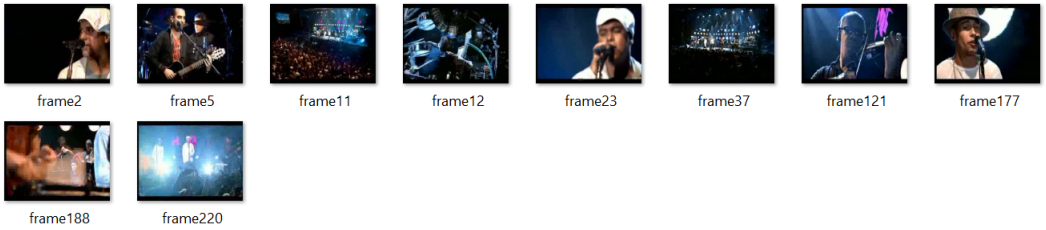

FIG. 5.10. User video summary

FIG. 5.11. K-means video summary

FIG. 5.12. User video summary

Analysis :

For the first video of a cartoon, all user selected frames except the two are extracted by K-means clustering-based approach. Actually, two missing frames are present in our summary with similar visual but slightly different expressions, i.e., frame721 and frame 871 of user summary are depicted by out757 and out947 of our summary wherein frame721 the cartoon is surprised and its corresponding out757 expresses less

surprising expression. The key point is only k-means clustering generates a summary almost matching with the user summary. In case of the second video of a football game, an important thing is the exact frames that user selected are getting selected in our summary. The position of a player running on the ground is accurately captured by our results. Captures covering the ground from different angles and audience are also exactly extracted in our summary though few captures featuring an important highlight when a player scored a goal are missing. For the third video, the notable thing is even blurred frames selected by the user are selected accurately in our results. For a music contest video, our summary captures a different participant than the one user selected and the audience is captured in our synopsis from a different angle. For the last video of advertisement, except a first frame, all remaining frames of user summary are selected by the clustering-based approach. Overall, we achieve qualitative and almost similar results and few frames are captured from different angles for a sports and music video. Also, the number of completely missing frames as compared with user summary is very less. So, only with clustering, we are generating a highly informative summary.

Chapter 6

CONCLUSION

We presented our intuitions for training an unsupervised model first using captioning information and then applying k-means clustering on the features. We show that by using such a simple network architecture, we achieve state-of-the-art in unsupervised video summarization. To show the generalization of our ideas, we tested our network architecture on different CNN network architectures like ResNet and VGG, showing improvement in the task of video summarization.

Bibliography

[1] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[2] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Video summarization using deep semantic features. In *Asian Conference on Computer Vision*, pages 361–377. Springer, 2016.

[3] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2017.

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

[5] Yingbo Li and Bernard Merialdo. Multi-video summarization based on video-mmr. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pages 1–4. IEEE, 2010.

[6] Bin Zhao and Eric P Xing. Quasi real-time summarization for consumer videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2513–2520, 2014.

[7] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015.

[8] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2698–2705, 2013.

[9] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Supplementary material: Video summarization with long short-term memory.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[11] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014.

[12] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014.

[13] Wikipedia contributors. Precision and recall — Wikipedia, the free encyclopedia, 2018. [Online; accessed 17-July-2018].