Video Summarization: An Overview of Techniques

Animesh Ramesh
IIT Kanpur
14511

Archit Sharma
IIT Kanpur
14129

Kanishk Gandhi
IIT Kanpur
14235

Nikhil Vanjani
IIT Kanpur
14429

Shibhansh Dohare
IIT Kanpur
14644

1 Introduction

With an explosion in multimedia content production due to ever increasing reach of internet, content in the form of videos is becoming increasingly commonplace, becoming the preferred method for the purpose of delivering content. Advent of social media and growth of video sharing websites, in particular Youtube, has only contributed to the increasing importance of videographic content. More content is uploaded to Youtube a day, than a person is capable of watching in his/her whole lifetime. With the emergence of video content as an effective mode of information propagation, automating the process of summarization a video has become paramount. Video Summarization, in recent times, has emerged as a challenging problem in the field of machine learning, which aims at automatically evaluating the content of a video, and generating a summary with the most relevant content of the video. Video summarization finds applications in generating highlights for sports events, trailers for movies and in general shortening video to the most relevant subsequences, allowing humans to browse large repository of videos efficiently.

Video summarization is a challenging problem in multiple facets. There is no natural ordering of video summaries. Among a given set of video summaries, the best representation of the original video is highly subjective. The general objective in modern literature for video summarization aims at producing a summary, typically 5%-15% of the whole video, consisting of the most informative content from the original video [1, 2]. The content is usually represented in the form of "key frames", or more appropriately as video skims. A good video summary depicts the synopsis of the original video, in a compact way depicting all important and relevant scenes/shots. Through this project, we review the major techniques in video summarization, and look at their performance on some recent datasets. Most of the work pertaining to this project can be found on github here [3].

2 Relevant Work

Video Summarization has gained a lot of attention in recent times. The task of video summarization can be accomplished in primarily two ways: *Key Frame Extraction* and *Video Skimming*. A lot of the initial work in video summarization has been done in the domain of unsupervised learning, but the recent trend has shifted towards learning from how humans generate video summaries, leading it into the supervised domains. A brief review of the work done in the video summarization is done according to the broad categorization above.

2.1 Key Frame Extraction

Key frame extraction saw most of the initial work done in this domain. Key frame extraction, as the name suggests, is to choose the most informative frames from the video. These indexed frames are supposed to be the best ones that summarize the video. The key frame extraction is primarily used

to obtain static summaries.
One of the staple algorithms for video summarization, VSUMM, posed the key frame extraction problem in a clustering setup [4]. Some of the other popular approaches in key frame extraction are [5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. Some of these are based on using web image priors [8, 7] while others use clustering based algorithms on various low-level features and change detection [4, 5, 9]. The essential necessity in the clustering based algorithms is to extract the features of interest that might make a frame worth being displayed in a summary. Some methods suggest choosing video skims around these key frames to make the summary more watchable [1]. Other variations in clustering based algorithms include a different clustering model, like [14], where the authors use hierarchical clustering to summarize a video.

2.2 Video Skims

Using key frames to summarize a video might be useful for automatically analyzing the content of the video, but it produces a discontinuous and rather unpleasant summary for human viewing. This calls for summarizing a video in the form of skims of the video. This however is a complex task and often is more difficult to achieve for user videos which lack structure. The semantic meaning is frequently required in such cases. Some work in this region, like [15], tries to learn the meaning of the video by detecting motion of the camera and dividing the [16] tries to do so by detecting and clustering low-level features, and detecting differences between frames. Often the original algorithms for key frame extraction can be be extended to video skimming by choosing a continuous set of frames around key frames to produce a video skim.

2.3 Supervised and Unsupervised Methods

Selecting the important shots/frames of a video is done using both unsupervised and supervised methods. Unsupervised algorithms like [7, 11, 9, 17, 18, 1] use manually defined factors for comparing frames and then subsequently choosing the key ones. The evaluation metrics are generally based on the intuition of how the frames should be different from each other and should be placed far apart in a feature space. The intuition in this approach is to find features of interest in frames, clustering frames that have similar features and finally choosing parts of the video that are highly dissimilar.

With increasing availability of datasets with available human extracted summaries, there has been some promising recent work [12, 19, 20, 1, 21] in the domain of supervised learning. Supervised algorithms require training examples to train the model to learn which parts of a video are important. The general requirement for supervised learning algorithms is that they require annotated data which is available in the form of level of interest that a frame has, or binary data indicating if a frame is to be included in the summary or not. It is important, however, in all methods to take into account the sequential nature [22] of a video and how humans perceive a video. The results of supervised learning are promising, although, still not comparably different from some of the unsupervised methods.

3 Datasets

3.1 SumMe

[1] created this benchmark with an intention to automate summary evaluation for present and future video summarization techniques, and have a common benchmark to compare it on. This dataset consists of *25 videos* which are single-shot and range in length from *1-6 minutes*. The dataset contains summaries created by *15 to 18 users* with the constraint in length being that the summaries should be *5% to 15%* of the original video. The dataset also has the average scores of a frame being included in the summary from this data. The benchmark comes along with an automatic video summary evaluation scripts, which computes the f-score for given set of frames chosen by the algorithm. The f-score is measured against the human generated summaries. The dataset was created using crowd sourced annotation.

3.2 TVSum50

The videos in SumMe dataset contain only one shot, as they are home videos recorded using a single camers, which is a major limitation of the dataset. [2] created this TVSum50, in order to test algorithms based on shot detection. This dataset contains 50 videos from 10 wide range of categories. The dataset also focusses on title based summarization, and supplies a representative image for each video, which can be used for summarization.

4 Methods Used

4.1 Uniform Sampling

Uniform Sampling is one of the most primitive algorithms for video summarization, which basically selects every k^{th} frame in the summary, where k is defined by the summary length requirement. Taking the typical summary size as 15% of the original video, this implies that every 6^{th} or 7^{th} is required to be chosen in the final summary. Uniform sampling is simple in execution, and is often used as a baseline score. Video summarization, being the complex task it is, even uniform sampling remains important in context of video summarization. The approach based on superframes as specified in [1] is discarded for the simpler approach mentioned here, and the results obtained are shown in the table.

4.2 VSUMM

First proposed in [4], the technique has been one of the fundamental techniques in video summarization in the unsupervised setup. The algorithm uses the standard K-means algorithm to cluster features extracted from each frame. *Color histograms* are proposed to be used in [4]. Color histograms are 3-D tensors, where each pixel's values in the RGB channels determines the bin it goes into. Since each channel value ranges in $0 - 255$, usually, 16 bins are taken for each channel resulting in a $16 \times 16 \times 16$ tensor. Due to computational reasons, a simplified version of this histogram was computed, where each channel was treated separately, resulting in feature vectors for each frame belonging to R^{48}. The approach in [1] for clustering is slightly different. But, the simplified color histograms give comparable performance to the true color histograms. The *features extracted from VGG16 at the 2^{nd} fully connected layer* were tried as features as well. Since the original features belong to R^{4096}, the features were reduced in dimensionality using PCA to R^{500}.

One of the initial things studies using VSUMM was the effect of pre-sampling, that is only a fraction of the frames were considered for summary. This was computationally necessary as taking all the frames increased compute time substantially. As the sequence of frames are strongly correlated, *sampling at low rates was shown to have no effect on the final results*. For the rest of the algorithm, only every 5^{th} fram was taken for consideration. (Taking a high sampling rate such as every 30^{th} frame often meant that a large enough summary cannot be obtained for evaluation. Also summary quality started to reduce at such high rates. So, 5 was chosen.)

Initial approach to the problem was to generate a key frame summary of 15% length, was to set $K = 15 * \text{frames}/100$ in K-means algorithm. The results for this approach are listed in VSUMM column using color histograms, using VGG16 features at fc7 in VSUMM+VGG16 column. The VGG16 approach was discarded to color histograms, because there was no substantial gain, and VGG16 features were computationally expensive to compute.

There are two problems with the above approach. After pre-sampling at every 5^{th} frame, the number of choices to be included in the 15% summary is only the 20% of original video. The second problem is that it produces discontinuous set of frames, which does not correlate well with how humans would generate video summary, giving relatively poorer scores. As suggested in [1], video skims of approximately 1.8 seconds centred around some key frames was used to generate summary. This slight modification reduced the number of key frames to be chosen using K-means clustering, and at the same time allowed for more continuous video summaries. The results are shown to improve using this technique in the VSUMM skims column.

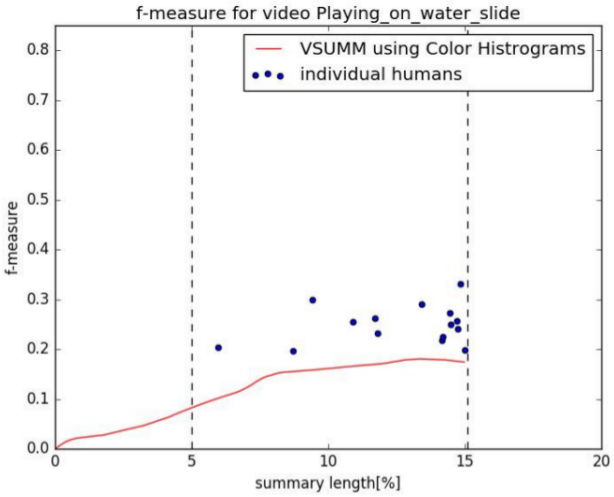

Figure 1: F-Measure vs Summary Length VSUMM [3]

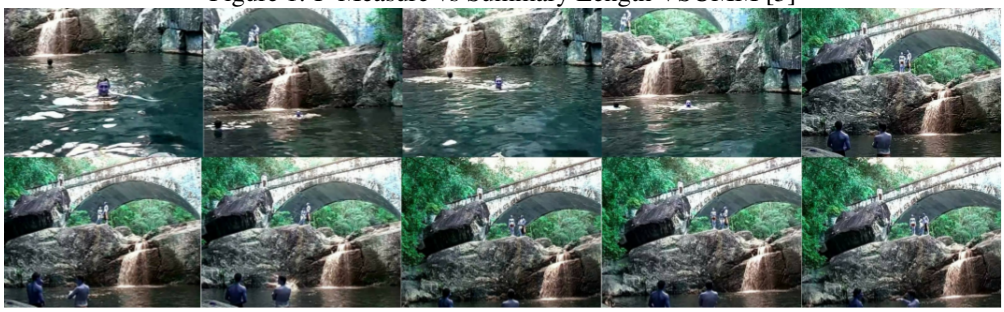

Figure 2: 10 key frames for "Playing on water slide" using VSUMM [3]

4.3 VGraph

This method proposed by [23] is based on partitioning the video into different shots using colour features. The key frames are extracted from nearest neighbour graph built from the shot representative frames. To extract frames from the nearest neighbour graph (NNG), first, the nearest neighbour of the shot representative frames are found. Colour features are used to calculate the similarity between two frames. To calculate pairwise similarity between colour histograms of representative frames, Bhattacharyya distance [24] is used. Then a reverse nearest neighbour graph (RNNG) is formed from the NNG. Finally, each strongly connected component in the RNNG is identified as a cluster and a frame is randomly chosen as a representative of the cluster. On the SumMe dataset most of the videos are single shots, so we had to reduce the cutoff which is used for detecting shot boundary. In some videos like Air Force One the algorithm was not able to detect any shot boundary thus this method failed to create the summary for them, this condition was quite rare. In most of the cases key frames were detected and to create a summary a continuous set of frames was chosen around the detected frame.

4.4 LSTM

While generating video summaries, the sequential nature of the video frames needs to be considered, as well as the interdependence that they display. A user, when viewing the video, does not look merely at the visual information, but delves into the semantic meaning that the video conveys, while also following the storyline of the video as and when required. A user takes all of this information into account before deeming whether a frame is interesting or not. It becomes a necessity to use a model that takes temporal features into account. LSTMs have been popularly used to model sequential data in domains such as speech recognition, document classification etc. The model

proposed here uses features of frames of the video and feeds them to the LSTM chains. These chains run opposite to each other and are not connected to each other. the outputs of the LSTMs and the input features are then fed into a *Multi Layer Perceptron*. If the features of the frames are denoted by $x = x_1, x_2, \ldots x_T$ the scalar output y_t giving the importance of each frame is defined as:

$$y_t = f_I(h_{forward}, h_{backward}, x_t)$$

It is this importance parameter that is used to train the model using annotated data. The features of the frame that can be shallow features like *SIFT* [25], *color histograms* and *texture histograms* or deep features extracted using *convolutional neural netwrks*. Skeleton code for the model based on LSTMs can be found in the project repository [3].

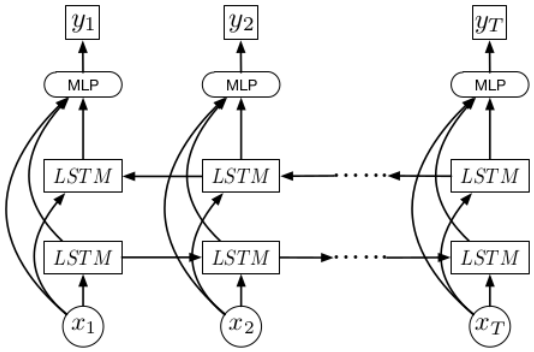

Figure 3: Proposed LSTM Model [3]

4.5 SIFT

This method proposed an approach to detect video shots by first detecting key objects in each frame, and then, tracking motion of objects to detect shot boundaries. To track motion it uses *FLANN index KD tree*. It further goes on to detect key frames in each shot based on euclidean distance between consecutive frames. Thirdly, it calculates entropy of each frame and combines all these three properties to find global key frames of the video. These key frames form the summary.

5 Results & Proposed Future Work

The results shown below are the F-scores of various models and humans on the SumMe dataset [1]. From the standard benchmarks shown below, we see that the algorithms are nowhere close to the upper bound, or even the best human. No algorithm dominates the results, i.e. outperform the other algorithms on most videos. This is because even in this small set of data, we have a wide variety of videos. Some algorithms suit a specific type of videos while fail miserably at others. For example, the SIFT algorithm seems to perform well on videos with high motion. In the future, a mixture of experts model could be trained to widen the range of videos that the model can deal with.

Video Name	Random	Upper bound	Worst Human	Mean Human	Best Human	Uniform	Cluster	Attr.	Gygli
Base jumping	0.144	0.398	0.113	0.257	0.396	0.168	0.109	0.194	0.121
Bike Polo	0.134	0.503	0.19	0.322	0.436	0.058	0.13	0.076	0.356
Scuba	0.138	0.387	0.109	0.217	0.302	0.162	0.135	0.2	0.184
Valparaiso Downhill	0.142	0.427	0.148	0.217	0.4	0.154	0.154	0.231	0.242
Bearpark climbing	0.147	0.33	0.129	0.217	0.267	0.152	0.158	0.227	0.118
Bus in Rock Tunnel	0.135	0.359	0.126	0.217	0.27	0.124	0.102	0.112	0.135
Car railcrossing	0.14	0.515	0.245	0.217	0.454	0.146	0.146	0.064	0.362
Cockpit Landing	0.136	0.443	0.11	0.217	0.366	0.129	0.156	0.116	0.172
Cooking	0.145	0.528	0.273	0.217	0.496	0.171	0.139	0.118	0.321
Eiffel Tower	0.13	0.467	0.233	0.312	0.426	0.166	0.179	0.136	0.295
Excavators river crossing	0.144	0.411	0.108	0.303	0.397	0.131	0.163	0.041	0.189
Jumps	0.149	0.611	0.214	0.483	0.569	0.052	0.298	0.243	0.427
Kids playing in leaves	0.139	0.394	0.141	0.289	0.416	0.209	0.165	0.084	0.089
Playing on water slide	0.134	0.34	0.139	0.195	0.284	0.186	0.141	0.124	0.2
Saving dolphines	0.144	0.313	0.095	0.188	0.242	0.165	0.214	0.154	0.145
St Maarten Landing	0.143	0.624	0.365	0.496	0.606	0.092	0.096	0.419	0.313
Statue of Liberty	0.122	0.332	0.096	0.184	0.28	0.143	0.125	0.083	0.192
Uncut Evening Flight	0.131	0.506	0.206	0.35	0.421	0.122	0.098	0.299	0.271
paluma jump	0.139	0.662	0.346	0.509	0.642	0.132	0.072	0.028	0.181
playing ball	0.145	0.403	0.19	0.271	0.364	0.179	0.176	0.14	0.174
Notre Dame	0.137	0.36	0.179	0.231	0.287	0.124	0.141	0.138	0.235
Air Force One	0.144	0.49	0.185	0.332	0.457	0.161	0.143	0.215	0.318
Fire Domino	0.145	0.514	0.17	0.394	0.517	0.233	0.349	0.252	0.13
car over camera	0.134	0.49	0.214	0.346	0.418	0.099	0.296	0.201	0.372
Paintball	0.127	0.55	0.145	0.399	0.503	0.109	0.198	0.281	0.32
mean	0.139	0.454	0.179	0.311	0.409	0.143	0.163	0.167	0.234
relative to upper bound	31%	100%	39%	68%	90%	31 %	36 %	37 %	52 %
relative to average human	45%	146%	58%	100%	131%	46 %	53 %	54 %	75 %

Video Name	Gygli	VSUMM	VSUMM + VGG16	VSUMM Skims	SIFT	Uniform	VGraph
Base jumping	0.121	0.083356	-	0.081787	0.234	0.085364	0.268
Bike Polo	0.356	0.078369	-	0.084036	0.196	0.074112	0.032
Scuba	0.184	0.145599	0.145599	0.14567	0.144	0.145059	0.021
Valparaiso Downhill	0.242	0.201909	-	0.20309	0.19	0.19899	0.250
Bearpark climbing	0.118	0.156611	-	0.15226	0.146	0.160377	0.253
Bus in Rock Tunnel	0.135	0.029341	-	0.031258	0.177	0.030199	0.031
Car railcrossing	0.362	0.386466	-	0.365478	-	0.363804	0.095
Cockpit Landing	0.172	0.096021	-	0.098027	0.035	0.089413	0.039
Cooking	0.321	0.023172	-	-	0.192	0.023748	0.039
Eiffel Tower	0.295	0.123115	-	0.131469	0.004	0.119034	0.236
Excavators river crossing	0.189	0.326871	0.326871	0.328678	-	0.328008	0.158
Jumps	0.427	0.174919	-	0.15363	-	0.176264	0.149
Kids playing in leaves	0.089	0.424418	-	0.438694	0.366	0.426775	0.349
Playing on water slide	0.2	0.174321	0.174321	0.177334	0.232	0.168675	0.040
Saving dolphines	0.145	0.229369	-	0.230782	0.121	0.212642	-
St Maarten Landing	0.313	0.039482	-	0.038585	0.12	0.040343	0.030
Statue of Liberty	0.192	0.070949	-	0.074132	0.208	0.068651	-
Uncut Evening Flight	0.271	0.251676	-	0.249076	0.256	0.253156	0.177
paluma jump	0.181	0.047268	0.047268	0.044189	0.092	0.048565	0.226
playing ball	0.174	0.258244	0.258779	0.274092	0.222	0.239955	0.171
Notre Dame	0.235	0.223917	-	0.217702	-	0.229265	0.182
Air Force One	0.318	0.065103	-	0.06423	0.07	0.066812	-
Fire Domino	0.13	0.003367	-	0.029354	0.247	0.002603	0.240
car over camera	0.372	0.038304	-	0.039166	-	0.035693	0.032
Paintball	0.32	0.233006	-	0.225485	-	0.224332	0.228
mean	0.234	0.155	-	0.162	0.171	0.152	0.174
relative to upper bound	52 %	34.44%	-	35.78%	38%	34%	32.49%
relative to average human	75 %	49.84%	-	51.70%	55%	49%	47.44%

6 What We Learnt

- Structuring a research problem by surveying the existing literature and then developing an intuition about how to approach solving the problem in concern.
- It is not a menial task to move from theory to practice. We learnt how to practically implement train and debug machine learning algorithms.
- The role played by features and how engineering features for the task at hand is important to learn a good model.

- To extract features for video summarization, we delved into the basics of computer vision and finding features of interest using popular techniques used in CV.
- These techniques included SIFT, CNNs for feature extraction, generating color and texture histograms among more.
- Using a common repository of code and working in a team by sharing and using modular code fragments throughout the project.
- Libraries/Tools Used: Python, Keras, TensorFlow, OpenCV, Scikit-learn, Scipy, FFmpeg

References

[1] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *European conference on computer vision*. Springer, 2014, pp. 505–520.

[2] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5179–5187.

[3] (2016) Video summarization repository. [Online]. Available: https://github.com/architsharma97/VideoSummarization

[4] S. E. F. De Avila, A. P. B. Lopes, A. da Luz, and A. de Albuquerque Araújo, "Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[5] N. Ejaz, I. Mehmood, and S. W. Baik, "Efficient visual attention based framework for extracting key frames from videos," *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 34–44, 2013.

[6] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1. IEEE, 1998, pp. 866–870.

[7] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan, "Large-scale video summarization using web-image priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2698–2705.

[8] G. Kim, L. Sigal, and E. P. Xing, "Joint summarization of large-scale collections of web images and videos for storyline reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4225–4232.

[9] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization." in *CVPR*, vol. 2, no. 6, 2012, p. 7.

[10] W. Wolf, "Key frame selection by motion analysis," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 1228–1231.

[11] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern recognition*, vol. 30, no. 4, pp. 643–658, 1997.

[12] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Advances in Neural Information Processing Systems*, 2014, pp. 2069–2077.

[13] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using delaunay clustering," *International Journal on Digital Libraries*, vol. 6, no. 2, pp. 219–232, 2006.

[14] D. Liu, G. Hua, and T. Chen, "A hierarchical visual model for video object summarization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2178–2190, 2010.

[15] K. Chorianopoulos, "Collective intelligence within web video," *Human-centric Computing and Information Sciences*, vol. 3, no. 1, p. 1, 2013.

[16] Z. Liu, E. Zavesky, B. Shahraray, D. Gibbon, and A. Basso, "Brief and high-interest video summary generation: evaluating the at&t labs rushes summarizations," in *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*. ACM, 2008, pp. 21–25.

[17] T. Liu and J. R. Kender, "Optimization algorithms for the selection of key frame sequences of variable length," in *European Conference on Computer Vision*. Springer, 2002, pp. 403–417.

[18] H.-W. Kang, Y. Matsushita, X. Tang, and X.-Q. Chen, "Space-time video montage," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1331–1338.

[19] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3090–3098.

[20] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarizatio," *arXiv preprint arXiv:1603.03369*, 2016.

[21] W.-L. Chao, B. Gong, K. Grauman, and F. Sha, "Large-margin determinantal point processes." UAI, 2015.

[22] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," *arXiv preprint arXiv:1605.08110*, 2016.

[23] K. Mahmoud, N. Ghanem, and M. Ismail, "Vgraph: an effective approach for generating static video summaries," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 811–818.

[24] T. Kailath., "The divergence and bhattacharyya distance measures in signal selection," in *Proceedings of the IEEE Communication Technology*, 1967, pp. 811–818.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.