EasyChair Preprint

№ 3519

Smart Surveillance for Smart City

Atanu Mandal, Amir Sinaeepourfard and Sudip Kumar Naskar

September 25, 2020

Smart Surveillance for Smart City

Atanu Mandal
Department of Computer Science & Engineering
Jadavpur University
Kolkata, India
atanumandal0491@gmail.com

Amir Sinaeepourfard
Department of Computer Science
Norwegian University of Science & Technology
Trondheim, Norway
a.sinaee@ntnu.no

Sudip Kumar Naskar
Department of Computer Science & Engineering
Jadavpur University
Kolkata, India
sudip.naskar@cse.jdvu.ac.in

***Abstract*—In recent years, video surveillance technology has become pervasive in every sphere. The manual generation of the description of videos requires huge time and labor and sometimes important aspects of videos are overlooked in human summaries. The present work is an attempt towards the automated description generation of Surveillance Video. The proposed method consists of the extraction of key-frames from a surveillance video, object detection in the key-frames, natural language (English) description generation of the key-frames and finally summarizing the descriptions. The key-frames are identified based on a mean square error ratio. Object detection in a key-frame is performed using *region convolutional Neural Network (R-CNN)*. We used *Long Short Term Memory (LSTM)* to generate captions from frames. *Translation Error Rate (TER)* is used to identify and remove duplicate event descriptions. Tf-idf is used to rank the event descriptions generated from a video and the top-ranked description is returned as the system generated a summary of the video. We evaluated the MSVD data set to validate our proposed approach and the system produces a *Bilingual Evaluation Understudy (BLEU)* score of 46.83.**

***Index Terms*—smart city, smart surveillance, video summarization, content-based video retrieval**

I. Introduction

During the era of the Internet of Things (IoT), population growth has increased a lot. IoT is an enabler for improving different aspects of public and private life with applications ranging from retail to home, health-care to transport, etc. where IoT systems are used for monitoring, crowd-sourcing and facilitating the shared environment. As these applications monitor and profile their users, they have obvious privacy implications. One of the major achievements obtained through IoT in today's world is *Smart Environment*.

A smart city is an urban improvement vision to incorporate various IoT and information and communication technology (ICT) arrangements safely to deal with a city's advantages. The city's benefits incorporate, yet are not constrained to, neighborhood offices data frameworks, schools, libraries, transportation frameworks, clinics, power plants, water supply systems, waste administration, law requirement, and other group administrations. The objective of building a smart city is to enhance personal satisfaction by utilizing innovation to enhance the proficiency of administrations and address occupants' issues. ICT permits city authorities to communicate straightforwardly with the group and the city base and to screen what is going on in the city, how the city is developing, and how to empower a superior personal satisfaction [1]. The video feed from different sources can be used for the security of the city. For solving this purpose, the city Event Detection Platform can be used.

Communication in terms of visual is the conveyance of ideas and information in forms that can be seen. Visual communication in part or whole relies on eyesight. Visual communication is a broad spectrum that includes signs, typographic, drawing, graphic design, illustration, industrial design, advertising, animation, color, and electronic resources. Visual communication contains image aspects. The interpretation of images is subjective and to understand the depth of meaning, or multiple meanings, communicated in an image requires analysis.

During the current era describing the visual content using natural language processing has received immense interest, expressly in a single sentence, whereas Describing the events of Video feeds has shown less interest even though it has large beneficially in the field of having analysis in unlawful scenes, human-machine interaction. For describing the events of Video feeds we proposed **Visual Data Analysis (VDA)** which not only provides the main events along with the other events in the video feeds which is useful to analyze and surveillance any unlawful scenes. The system which provides information on the activity always useful for the security of the society.

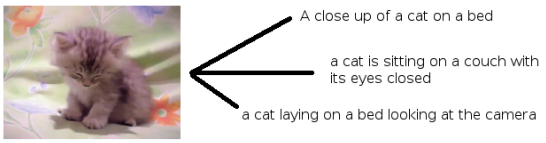

Fig. 1. Output of VDA

In this paper, we discuss our approach targeted the summarization of a feed from a surveillance camera while evaluated with the data provided. Figure 1 shows a key-frame of a video feed and its relevant output. From the frame, it is impossible to identify whether the cat is on bed or couch. But the ground truth is it is sitting on a couch.

The paper is organized as follows. In Section II, we established the works previously done in the field. In Section III, discussion on our approach is made whereas Section IV,

we provided the result of our evaluation and Section V, we discussed our results.

II. Related Works

In this section, we discussed the previous methods that have been applied in the field. Till today the work for the betterment of this process is ongoing and no one particular method is foolproof and the quest for more accurate results is still on. There are two major methods one is a single view and another is a multi-view.

Huge progress has been made using a variety of ways to summarize a single view video in an unsupervised manner or developing supervised algorithms. Various strategies have been studied, including clustering [3], attention modeling [4], saliency-based linear regression model [5], superframe segmentation [6], kernel temporal segmentation [7], crowd-sourcing [8], energy minimization [9], storyline graphs [10], long short-term memory [11] and maximal bi-clique finding [12]. Recently, there has been a growing interest in using sparse coding (SC) to solve the problem of video summarization since the sparsity and reconstruction error term naturally fits into the problem of summarization.

Another approach is the Multi-view video summarization. It is a more challenging problem due to the inevitable thematic diversity and content overlaps within multi-view videos than a single video. To address the challenges encountered in multi-view settings, there have been some specifically designed approaches that use random walk over Spatio-temporal graphs [13] and rough sets to summarize multi-view videos. A work proposed by Author [14] uses bipartite matching constrained optimum path forest clustering to solve the problem of multi-view video summarization. An online method has also proposed by Author [15]. However, this method relies on inter-camera frame correspondence, which can be a very difficult problem in uncontrolled settings. The work in [16] and [17] also addresses a similar problem of summarization in non-overlapping camera networks. Learning from multiple information sources such as video tags [18], topic-related web videos, and non-visual data, is also a recent trend in multiple web video summarization.

Author [2] addressed the problem of summarizing multi-view videos via joint embedding learning. The embedding helps in capturing content correlations in multi-view datasets without assuming any prior correspondence between the individual videos. On the other hand, the sparse representative selection helps in generating multi-view summaries as per user length requests without requiring an additional computational cost. Performance comparisons on six standard multi-view datasets show marked improvement over some mono-view summarization approaches as well as state-of-the-art multi-view summarization methods.

Many researchers mentioned that videos can also be considered as having a hierarchical structure if video shoots and video scenes are accounted, which are two higher levels other than frames. A video shot consists of a series of frames obtained from the camera and associated camera effects such as zooming, panning, and tilting. In addition to video obtained directly from the camera, these shots can be combined using special editing effects such as fade-in/fade-out, dissolve, etc. Video scene can be defined as a combination of several shots stitched together which represents a relatively complete semantic content. Some key-frames are also introduced to characterize each shot or scene. So, a hierarchical representation has at least four levels i.e., key-frames, shots, scenes, and complete video. Although this structure can be used as an approach to video representation, it lacks the semantic content required by general users. Anjum et al., aims to extract useful information, semantics, and highlights from raw video and further elaborate it by annotation, to be used later on for further content-based indexing and retrieval.

But recent research has proved that multi-view works better than a single view. The multi-view method is different from single video summarization in two important ways. First, although the amount of multi-view data is immensely challenging, there is a certain structure underlying it. Specifically, there is a large number of correlations in the data due to the locations and fields of view of the cameras. So, content correlations, as well as discrepancies among different videos need to be properly modeled for obtaining an informative summary. Secondly, these videos are captured with different view angles, and depth of fields, for the same scenery, resulting in some unaligned videos. Hence, the difference in illumination, pose, view angle and synchronization issues pose a great challenge in summarizing these videos. So, methods that attempt to extract summary from single-view videos usually do not produce an optimal set of representatives while summarizing multi-view videos. Advantages of multi-view summarization are better to characterize the multi-view structure, The author [2] projects the data points into a latent embedding that can preserve both intra-view and inter-view correlations without assuming any prior correspondences or alignment between the multi-view videos, example, un-calibrated camera networks. Authors underlying idea hinges upon the basic concept of subspace learning, which typically aims to obtain a latent subspace shared by multiple views by assuming that these views are generated from this subspace. Second, the author [2] proposes a sparse representative selection method over the learned embedding to summarize the multi-view videos. Specifically, the author formulates the task of finding summaries as a sparse coding problem where the dictionary is constrained to have a fixed basis (dictionary to be the matrix of same data points) and the nonzero rows of the sparse coefficient matrix represent the multi-view summaries. Finally, to better leverage the multi-view embedding and selection mechanism, the author learns the embedding and optimal representatives jointly. Specifically, instead of simply using the embedding to characterize multi-view correlations and then selection method, proposes to adaptive change of the embedding concerning the representative selection mechanism and unify these two objectives in forming a joint optimization problem. With joint embedding and sparse representative selection, the final objective function is both non-smooth and non-convex.

Author [2] presents an efficient optimization algorithm based on half-quadratic function theory to solve the final objective function.

III. Proposed Methods

We propose the VDA model for Description of Surveillance Video in Smart City, where input is the video feed, and output is the sequence of sentences $\{x_0, x_1, ..., x_n\}$. We divided III into 4 parts for the brief discussion of our work. In Section III-A we briefed approach for extraction of key-frames from a video feed which has the useful information of the scenes. In Section III-B we briefed on the detection of objects from the key-frames and in Section III-C we discussed approach on the generation of the relation of the objects. In section III-D we discussed the method used for the representation of the final events.

A. Image Frame Extraction

The main part of image frame extraction from a feed is the frame size of the feed. Frame size i.e. frame per second (fps) vary from the recording device to the device. It can be from 25 fps to 300 fps. For content-based video retrieval for retrieving images meeting with specific visual features (such as scenes, lens, frames, and moving object in the video) can be used.

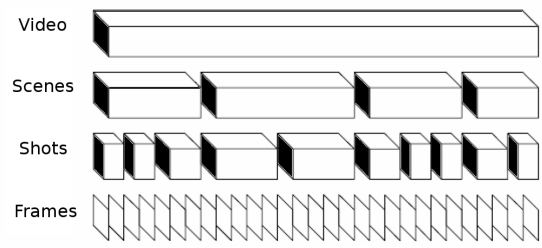

Fig. 2. Anatomy of a video

The method mainly includes Key Frame extraction. The key-frame extraction is a process that extracts the most representative image collections from the original video and refers to the image frame in the video sequence which is representative and also able to reflect the summary of a shot content. It is one of the key technology of content-based video retrieval and is the basis of video analysis and retrieval. By using the key-frame we can express the main content of lens clarity, and reduce the amount of video processing data and complexity greatly.

Author [19] provided an algorithm for the extraction of the key-frame using the method based on image frame information. Author proposed it in two phases, in which the first phase computes the threshold (T) using the mean and standard deviation of the histogram of absolute differences of consecutive image frames.

$$T = \mu_{adh} + \sigma_{adh} \tag{1}$$

where μ_{adh} is mean of absolute difference and σ_{adh} is the standard deviation of an absolute difference.

The second phase extracts key-frames comparing the threshold against the absolute difference of consecutive image frames.

Algorithm 1 KEY FRAME()

```
Extract frame one by one
Histogram difference between 2 consecutive frames
Find mean and standard deviation of absolute difference
Compute Threshold (T)
Compare the difference with T
if Step 5 > T then
    Select it as a key frame
else
    Go to Step 2
end if
Continue till end of loop
```

There are many algorithms proposed but many of those algorithms can miss few frames and chances of missing the important key-frames is higher. Algorithm 1 can miss a few frames as a key-frame. For this purpose, an algorithm is proposed where the system will extract all the image frames and keep the most dissimilar images as key-frames and remove similar images using Structural Similarity Measure (SSIM). SSIM [20] can be used because it remedies some of the issues of Mean Squared Error(MSE).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2}$$

SSIM attempts to model the perceived change in the structural information of the image, whereas MSE is estimating the perceived errors. The Equation 2 mentioned is used to compare two windows (small sub-samples) rather than the entire image as in MSE. Doing this leads to a more robust approach that can account for changes in the structure of the image, rather than just the perceived change.

Fig. 3. Image Difference in SSIM = 0.7867

In figure 3 the two frames used are frames of the same feed which gives SSIM index of 0.7867, the reason is there is a huge difference in the frames.

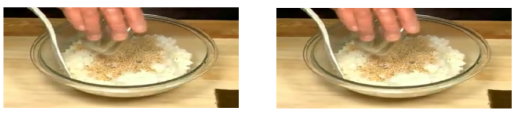

Fig. 4. Image Difference in SSIM = 0.1

In figure 4 the two frames are the consecutive frames hence provides the SSIM index of 0.1 means both are duplicate frames.

Algorithm 2 IMAGE FRAME($feed$, T_{ssim})

$TF \leftarrow fps * d$
$F \leftarrow \{f_1, f_2, ..., f_{TF}\}$
$K \leftarrow \{f_1\}$
$f_x \leftarrow f_1$
for $i = 2; i \leq TF; i++$ **do**
  if $SSIM \leq T_{ssim}$ **then**
    $Append(f_i, K)$
    $f_x \leftarrow f_i$
  else
    $Remove\ f_i$
  end if
end for
return K

The algorithm 2 is proposed for the extraction of useful frames from a video feed. In the algorithm, a set of Frames (F) has been created and blindly first frame has been selected as a member of the set of key-frame(K). The next frame is compared with the selected key-frame and if it satisfies the condition of the Threshold value for SSIM (T_{ssim}) then the frame is appended to the set of key-frames else the frame is removed from the set. This results in a set of key-frames which are meaning-full as well as use-full and leads to less space complexity.

B. Object Detection from Key Frames

In previous section III-A we extracted all the key-frames of a video. This section we proposed to detect the object using Single Shot Detection and MobileNets. This method, when combined, can be used for super-fast, real-time object detection on resourced constraints devices. We used the OpenCV module to load a pre-trained object detection network. This will enable us to pass input frames through the network and obtain the output bounding box (x, y) coordinate of each object. When we dealt with deep-learning-based object detection there are primarily 2 object detection methods.

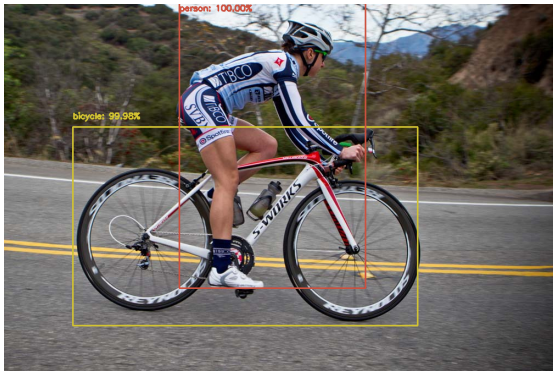

Fig. 5. Detection of Bicycle and a person

Faster R-CNN's [21] are the likely and most used method for object detection using deep learning, however, we found that technology is challenging to train, even with the fastest implementation R-CNN's the algorithm can be quite slow, on the order of 7 FPS. Whereas if we wish to look for speed author [22], has provided a solution which is capable of processing 40-90 FPS, but the result leaves the desired accuracy.

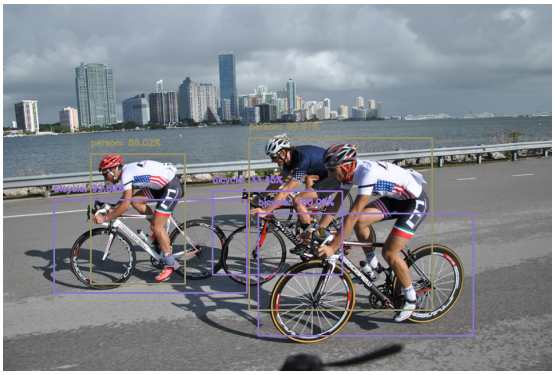

Fig. 6. Detection of group of person and bicycle

Many previous works use an existing network architecture like VGG but the problem is that this network architecture is very large around 200-500MB. So it can be said that these networks are unsuitable of resource constraints devices due to the sheer size and the resulting number of computations. In our proposed model, we used MobileNets [23] combined with SSDs usually because they are designed for resource constraints devices and provide fast, and efficient deep-learning methods to object detection. Using the model, it is easier to detect a single object (fig. 5) and multiple objects (fig. 6).

Training and result: The model was trained on COCO Dataset and finely tuned on PASCAL VOC. As a result of which it can detect multiple objects.

Using the proposed we can detect the objects, which will be passed to Convolutional Neural Network (CNN) to get the features i.e. relations of the objects in the key-frame which was then passed to LSTM for caption generation, where the objects were passed as encoder and feature as a decoder.

C. Caption Generation of the events in Key Frames

This section we proposed to form the relative sentence of the events occurred using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) combined to create the caption of the key-frame.

In our model, we take the set of objects detected of a frame, F as input, and is trained to maximize the likelihood $p(S|I)$ of producing a target sequence of words $S = \{S_1, S_2, ..., S_n\}$ where each word S_i comes from a given dictionary, that describes the frame. We used CNN to create a dense feature vector that is used as a feature input into the LSTM network in the encoder. For the decoder of the LSTM, we created feature vector which is the relation between the objects of the key-frame.

The LSTM unit proposed in [24], for an input x_i at time step t, the LSTM computes a hidden/control state h_t and a

memory cell state c_t which is an encoding of everything the cell has observed until time t.

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ g_t &= \phi(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \phi(c_t) \end{aligned} \tag{3}$$

where σ is the sigmoidal non-linearity, ϕ is the hyperbolic tangent nonlinearity, $\odot$ represents the element-wise product with the gate value, and the weight matrices denoted by W_{ij} and biases b_j are the trained parameters.

In sentence language modeling, LSTM is predicting the next word in a sentence. In the model, we created an embedding of the frames. This embedding is then fed as an initial stage into the LSTM. This becomes the first previous state to the language model, influencing the next predicted words. At each time-step, the LSTM considers the previous cell state and outputs a prediction for the most probable next value in the sequence. This process is repeated until the end token is sampled, signaling the end of the caption.

For generation of caption we used the Beam Search algorithm, a heuristic-based that finds the most promising nodes. It generates all possible next paths, keeping only the top N best candidates at each iteration. As the number of nodes to expand from its fixed, this algorithm is space-efficient and allows more potential candidates than a best-first search. A caption generator utilizes a beam search to improve the quality of sentences generated. At each iteration, the generator passes the previous state of the LSTM (the initial state is the image embedding) and the previous sequence to generate the next softmax vector. The top N most probable candidates are kept and utilized in the next inference step. This process was continued until either the max sentence length is reached or all sentences have generated the end-of-sentence token.

For generating a caption the idea of graph search problem can be used, where the nodes are the words and the edges are the probability of moving from one node to another. Finding the optimal path involves maximizing the total probability of the sentence. Sampling and choosing the most probable next value is a greedy approach to generating a caption. But experimentally it is computationally efficient but can lead to a sub-optimal solution. Given all possible words, it would not be computationally efficient to calculate all possible sentences and determine the optimal sentence. This rules out using a search algorithm such as Depth First Search or Breadth-First Search to find the optimal path.

D. Text Representation

In the above section III-C we generated the caption of the key-frames which provide all the events of the video feed. But in certain conditions due to Threshold as mentioned T_{ssim} there are possibilities of duplication of frames. To resolve the problem we used Translation Error Rate (TER). TER is a method used to determine the amount of post-editing required for a machine translation job. The automatic metric measures the number of actions required to edit a translated segment inline with one of the reference translations. It's quick to use, language-independent and corresponds with post-editing effort. We took each line as a document and calculate the unique events in the video feed using the threshold of 50%. Using the TER we were able to find all the relevant events of the feed.

As our target is to retrieve the most important description we proposed to rank the document from most important to least important. To satisfy the problem we used Term Frequency (tf) and Inverse Document Frequency (idf). Using the video feed on YouTube [1] we received multiple events. We took all the unique words from the events and found the weights of appearing in a document and count the terms of appearance. Later, we tried to measure how much information does the word provides that is, whether the term is common or rare across all documents. We just multiplied the tf and idf to get tf-idf weighting for each.

$$W_{t,d} = log(1 + tf_{t,d}) * log_{10}(\frac{N}{df_t}) \tag{4}$$

To get the most frequent and useful we sorted the lines from top to bottom.

IV. Dataset, Evaluation and Result

This section describes the evaluation of our approach. We evaluated our output with a video description corpora, namely the Microsoft Video Description corpus [27](MSVD).

A. Microsoft Video Description Corpus (MSVD)

MSVD is based on web clips with short human-annotated sentences. The Microsoft Video description corpus is a collection of Youtube clips collected on Mechanical Turk by requesting workers to pick short clips depicting a single activity. The videos were then used to elicit single sentence descriptions from annotators. The original corpus has multilingual descriptions, in this work we use only the English descriptions. We did minimal pre-processing on the text by converting all text to lower and remove punctuation.

B. Microsoft Video Description Corpus Evaluation

In this section, we have provided the result of data provided of YouTube and their data as reference and evaluated our result with BLEU matrices.

The test set contains 1931 videos. Considering the diverse nature of the data sets and the limitations of automatic evaluation metrics, the results compared to different benchmark techniques, shown in Table I using BLEU only.

For the MSVD dataset, we compared our model with a benchmark [25] and get achieving a result of 46.83. As for each dataset videos, the overall proposed method provides an average result, which can be overcome and a better result can be found. During dealing with the results it has come into

[1] https://www.youtube.com/watch?v=ZbzDGXEwtGc

TABLE I
PERFORMANCE OF VIDEO DESCRIPTION METHODS ON **MSVD** DATASET.

Sl. No.	Techniques / Models / Methods	Year	BLEU
4	h-RNN [25]	2016	49.9
5	**VDA**	**2020**	**46.83**
3	TDDF [26]	2017	45.8

light that in human translated results synonyms of the word of system generated word has been used like humans used aeroplane, airplane, plane, whereas system generates airplane. All are synonyms of each other but produced great impact on the BLEU score. If synonyms are taken into actions the score can be much higher than the currently provided score and can even outperform the results of [25]. Our BLEU score was calculated using the tool used in MOSES.

C. Microsoft Video Description Corpus Evaluation Result

During our evaluation process, we found that the data-set provided by MSVD have focused on the main events which are described by human beings, whereas our system can provide main events along with other relevant events. Using a dataset provided in YouTube[2], MSVD focuses on main events i.e. "An airplane is flying in the sky" whereas our system can provide whether the sky is cloudy or not and whether the airplane is a fighter or commercial aircraft.

V. CONCLUSION

This paper proposed an approach to generate events of a video feed using our proposed model VDA. In contrast to related work, we construct descriptions, where key-frames are first to read sequentially and then sentences are generated sequentially. This allows us to handle multiple events of a feed. Our model achieves good performance on the MSVD dataset and excellent performance can be obtained by using synonyms of the words/objects. Despite its conceptual simplicity, our model significantly benefits for providing additional data and is slightly depends upon the quality of the video, suggesting that it has a high model capacity, and can learn complex temporal structure in the input and output sequences for challenging movie-description datasets like MPII-MD.

REFERENCES

[1] P. Rizwan, K. Suresh and M. R. Babu, *Real-time smart traffic management system for smart cities by using Internet of Things and big data* 2016 International Conference on Emerging Technological Trends (ICETT), Kollam, 2016, pp. 1-7. doi: 10.1109/ICETT.2016.7873660

[2] Rameswar Panda and Amit K. Roy-Chowdhury. *Collaborative Summarization of Topic-Related Videos* 2017; arXiv:1706.03114.

[3] Almeida, Jurandy, Neucimar J. Leite, and Ricardo da S. Torres. *Vison: Video summarization for online applications* Pattern Recognition Letters 33.4 (2012): 397-409.

[4] Ma, Yu-Fei, et al. *A generic framework of user attention model and its application in video summarization* IEEE transactions on multimedia 7.5 (2005): 907-919.

[5] Lee, Yong Jae, Joydeep Ghosh, and Kristen Grauman. *Discovering important people and objects for egocentric video summarization* Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[6] Gygli, Michael, et al. "Creating summaries from user videos." European conference on computer vision. Springer, Cham, 2014.

[7] Potapov, Danila, et al. *Category-specific video summarization* European conference on computer vision. Springer, Cham, 2014.

[8] Khosla, Aditya, et al. *Large-scale video summarization using web-image priors* Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.

[9] Pritch, Yael, et al. *Webcam synopsis: Peeking around the world* Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007.

[10] Kim, Gunhee, Leonid Sigal, and Eric P. Xing. *Joint summarization of large-scale collections of web images and videos for storyline reconstruction* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[11] Zhang, Ke, et al. *Video summarization with long short-term memory* European conference on computer vision. Springer, Cham, 2016.

[12] Chu, Wen-Sheng, Yale Song, and Alejandro Jaimes. *Video co-summarization: Video summarization by visual co-occurrence*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[13] Fu, Yanwei, et al. *Multi-view video summarization* IEEE Transactions on Multimedia 12.7 (2010): 717-729.

[14] Kuanar, Sanjay K., Kunal B. Ranga, and Ananda S. Chowdhury. *Multi-view video summarization using bipartite matching constrained optimum-path forest clustering* IEEE Transactions on Multimedia 17.8 (2015): 1166-1173.

[15] Ou, Shun-Hsing, et al. *On-line multi-view video summarization for wireless video sensor network* IEEE Journal of Selected Topics in Signal Processing 9.1 (2015): 165-179.

[16] De Leo, Carter, and B. S. Manjunath. *Multicamera video summarization from optimal reconstruction* Asian Conference on Computer Vision. Springer, Berlin, Heidelberg, 2010.

[17] Leo, Carter de, and Bangalore S. Manjunath. *Multicamera video summarization and anomaly detection from activity motifs* ACM Transactions on Sensor Networks (TOSN) 10.2 (2014): 27.

[18] Wang, Meng, et al. *Event driven web video summarization by tag localization and key-shot identification* IEEE Transactions on Multimedia 14.4 (2012): 975-985.

[19] Sheena, C. V., and N. K. Narayanan. *Key-frame extraction by analysis of histograms of video frames using statistical methods* Procedia Computer Science 70 (2015): 36-40.

[20] Wang, Zhou, et al. *Image quality assessment: from error visibility to structural similarity* IEEE transactions on image processing 13.4 (2004): 600-612.

[21] S. Ren, K. He, R. Girshick and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017. doi: 10.1109/TPAMI.2016.2577031

[22] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788. doi: 10.1109/CVPR.2016.91

[23] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto and Hartwig Adam. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications* 2017; arXiv:1704.04861.

[24] Wojciech Zaremba and Ilya Sutskever. *Learning to Execute* 2014; arXiv:1410.4615.

[25] H. Yu, J. Wang, Z. Huang, Y. Yang and W. Xu, *Video Paragraph Captioning Using Hierarchical Recurrent Neural Networks* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 4584-4593.

[26] X. Zhang, K. Gao, Y. Zhang, D. Zhang, J. Li and Q. Tian, *Task-Driven Dynamic Fusion: Reducing Ambiguity in Video Description* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6250-6258.

[27] Chen, David L., and William B. Dolan. *Collecting highly parallel data for paraphrase evaluation* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.

[2]https://www.youtube.com/watch?v=ZbzDGXEwtGc