Artificial Bee Colony Algorithm Optimization for Video Summarization on VSUMM Dataset

Vinsent Paramanantham[1]
Research Scholar
Faculty of Computer Science and Engineering,
Sathyabama Institute of Science and Technology
(Deemed to be University)

Dr. S. SureshKumar[2]
Principal, Swarnandhra College of
Engineering and Technology, Narasapur, AP, India.

***Abstract*—This paper attempts to prove that the Artificial Bee Colony algorithm can be used as an optimization algorithm in sparse-land setup to solve Video Summarization. The critical challenge in doing quasi(real-time) video summarization is still time-consuming with ANN-based methods, as these methods require training time. By doing video summarization in a quasi (real-time), we can solve other challenges like anomaly detection and Online Video Highlighting. A simple threshold function is tested to see the reconstruction error of the current frame given the previous 50 frames from the dictionary. The frames with higher threshold errors form the video summarization. In this work, we have used Image histogram, HOG, HOOF, and Canny edge features as features to the ABC algorithm. We have used Matlab 2014a for doing the feature extraction and ABC algorithm for VS. The results are compared to the existing methods. The evaluation scores are calculated on the VSUMM dataset for all the 50 videos against the two user summaries. This research answers how the ABC algorithm can be used in a sparse-land setup to solve video summarization. Further studies are required to understand the performance evaluation scores as we change the threshold function.**

***Keywords*—Artificial Bee Colony optimization; video summarization; online video highlighting; sparse-land; anomaly detection; image histogram; HOG; HOOF; canny edge**

I. Introduction

Since campuses, roads, and public places are monitored constantly by video surveillance, the adaptation of VS will be imperative. Skimming through a huge corpus of video data to derive meaningful summarization requires efficient VS techniques. The need of the hour is to come up with techniques that can be easily deployed and require less training of the algorithms as in ANN methods. Some of the frameworks work well in the object tracking environment or any others. In this framework, we have come up with a common approach to do VS, as seen in the result section evaluated across multiple genres of videos table reference. The main motivation behind this work is three-fold. Firstly, to prove the use of the ABC optimization algorithm in a sparse-land setup. Secondly, to apply this approach to a real-time (quasi) framework similar to [1]. Thirdly, to adapt any domains online video content so that it can be used to solve other challenges in real-time like anomaly detection [2].

The challenge to any video summarization is to adapt to any domain, some of the frameworks work well on a certain domain as the methods are restricted or concentrated for a particular purpose like choosing humans and vehicle [3]. Methods like the sparse-land approach give the liberty to adapt to any domain videos, which is also proven in this work by the evaluation scores across multiple genre videos in VSUMM dataset in Table I, II. There has been a keen interest in the sparse-land based approach in the literature [4, 1, 5, 6, 7, 8], hence taking this approach in this paper is proven.

The rest of the paper is organized as follows: Section II briefs about the related works in VS. Section III describes the proposed ABC method for the VS framework, Section IV deals with the proposed methodology, Section V discusses the experimental results. Finally, Section VI concludes the paper.

II. Related Works

The optimization algorithms play a vital role in selecting the right frames for video summarization and updating the dictionary D. Various studies on optimization algorithm and its performance metrics are based on storage reduction and computation time, as discussed in [9]. In this paper, we have evaluated the ABC algorithm against a well know dataset VSUMM [10], and the results benchmarked against a known dataset. In this section, we will go through optimization algorithm selection and different strategies to do VS. In the literature, we find lots of methods and techniques to do VS, based on clustering [11], saliency-based methods calculating the frame importance score on egocentric VS[lee2012discovering], traditional approaches with SVD [12]. In recent years there is an enormous amount of papers based on ANN [13, 14, 15, 16, 17, 18, 19]. ANN methods involve training in supervised, unsupervised approaches which may not be suitable for a near real-time VS. Graph-based methods [20, 21, 22, 23] also requires the data porting into a graph database before computation which specializes in keyframe retrievals and browsing system. Among all of the methods, the sparse-land based approach to solve VS still stands out of other techniques due to its simplicity in solving VS as an optimization problem. The other features can be easily plugged and played with any optimization algorithm, as demonstrated in [9], flexibility in selecting the right dictionary shapes and elements and support quasi (real-time) in solving the VS [1], followed by anomaly detection [2]. We also see recent advancement in the sparse-land approach using CSC(Convolutional Sparse Coding Model) as on par with the current ANN methods [24].

Optimization algorithm from Evolutionary methods like ABC [25], PSO [26], GA [27], $ADMM$ [28] and $rmsprop$

[29] are quite common methods for optimization algorithm, In this paper, we have used the ABC method for optimization. In the recent literature, we can see ABC usage [30] for VS, where the authors have worked on another well-known dataset Summe [31] using segment level data on the Video for VS. The global effects of the entire video may not be captured well in such approaches [30].

[30] has used the ABC algorithm to identify key video segments and used clustering techniques to arrive at keyframes. The keyframes come from the center of the cluster. A region of interest approach is used to identify important frames, similar to the camshift algorithm proposed in our work to reduce the unwanted frames. The final reduction of keyframes is done via the hue histogram comparison. Also, the ABC algorithm has shown better convergence than other algorithms like PSO.

[9], in our previous approach, we have proposed four algorithms to test video summarization optimization time and storage reduction. The test was performed on random videos on youtube, whereas this paper accomplishes the performance of the ABC optimization algorithm against known VSUMM dataset in VS, also we have calculated the performance evaluation scores as indicated in the experiments and results section.

[1] has used ADMM optimization techniques in a sparse-land setup to solve VS challenge, these ideas are some of the key foundations in solving the VS framework along with dictionary initialization and sparse modeling. References for image restoration can be found in [32]. Image reconstruction is done with the current frame and frames from the dictionary. A high reconstruction error of α denotes more changes between frames. When the reconstruction error α is high the frame is included for summarization [33, 5, 34, 35].

A. Summary of the Contribution

Our contribution in this work is the usage of the ABC optimization algorithm in a sparse-land setup to do VS. The evaluation metrics $precision, recall, F1-Score$ are obtained for the individual video to showcase the working of the ABC algorithm on par with other methods as compared in Table III with earlier reference works [10]. The other two Tables I, II gives the $precision, recall, F1-Score$ for all the 50 individual videos in VSUMM dataset. This framework works as a near real-time(quasi-real-time) summarization and anomaly detection framework. The framework can also be easily extended to other advanced sparse-land setups such as CSC [24].

III. The ABC Optimization for VS Framework

The artificial bee colony (ABC) algorithm comes from the swarm intelligence branch. The ABC algorithm is modeled around the intelligent behavior of honey bee in performing their task efficiently to identify the target food locations [25, 36]. There are mainly three types of phase, Employed, onlooker, and scout bee phases. The employed bees are responsible for visiting the existing food sources, onlooker bees wait for the dance ceremony to select the next food source depending upon the performance of the bees, the scout bees do a random pickup of food sources. The main function of Employed phase is to update the X_{new} position variable and to find a suitable partner solution X_p, the update equation to calculate the new position is as shown in the below equation 1. X is the current solution and X_p is the partner solution. ϕ is a random value in the range [-1,1].

$$X_{new} = X + \phi(X - X_p), \phi \in [-1, 1] \tag{1}$$

The Onlooker bees are responsible for selecting the food sources with a highest nectar value $F(\theta_i)$, θ_i is the i^th food source, the probability of a cycle is given as $P(c) = \{\theta_i(c)|i = 1, 2, ...S\}$, (C: cycle, S:no. of food sources), probability function $p(X_i)$ for choosing the food sources as given below.

$$p(x_i) = \frac{F(\theta_i)}{\sum_{i=1}^{S} F(\theta_k)} \tag{2}$$

The scout bees do a random discovery of the food sources with the predefined limits specified by the search space limits $[X^{Min}, X^{Max}]$, the randomness of the food sources are determined by the below equation 3.

$$X_{i,j} = X_j^{min} + rand_{i,j} \times (X_j^{max} - X_j^{min}) \tag{3}$$

Where $i = 1, 2, ...S$, S is the number of food sources, $j = 1, 2, ...d$, d dimensional vector solution, $X^{min} = x_1^{min}, x_2^{min}, , .., x_d^{min}$ and $X^{max} = x_1^{max}, x_2^{max}, , .., x_d^{max}$, $rand_{i,j}$ is a value from a uniform distribution (0,1).

IV. Proposed Methodology

The architectural flow for VS is similar to our previous work [9]. The features used are HOG(histogram of oriented gradients) with nine bins with a range of 20 degrees per bin, HOF(Histogram of Optical Flow), $HOOF$(Histogram of Optical Flow), Canny edge detection, the sample feature output of a frame can be seen in the below Fig. 1.

A. Preprocessing of Video Using

The camshift algorithm is used to preprocess the frames, a wide variety of applications can found for the camshift algorithm [37, 38, 39] including object tracking and frame rate and size reduction by only capturing the ROI areas. In our approach, we have used the camshift algorithm to reduce the number of frames. This is an important step to filter keyframes. The camshift algorithm usage and depiction can be seen in Fig. 2, similar methods can be seen in the literature [40].

CA{1,1}

	1	2	3	4
1	0.2662	0	0.0125	3600
2	0.187	0	0	0
3	0.3927	0.5	0	0
4	0.2276	0	0	0
5	0.6122	0	0	0
6	0.1907	0	0	0
7	0.3722	0.5	0	0

Fig. 1. The Feature Matrix for HOG, HOF, Image Histogram, Canny Edge

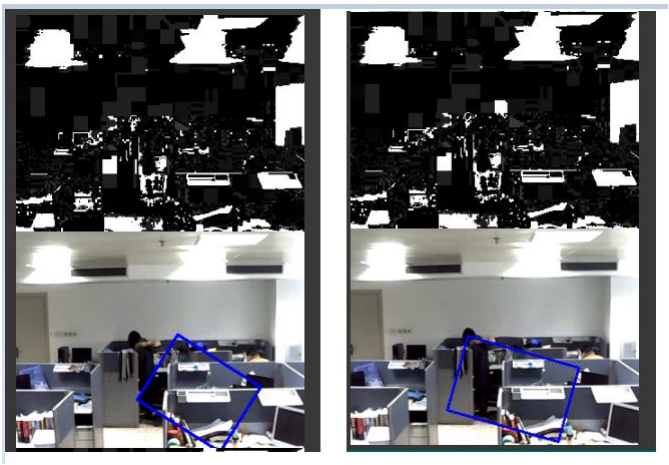

Fig. 2. The Camshift Algorithm usage for Frame Reduction

B. Features Used

The features used can be seen in the code listing Matlab code below and the values as depicted in Fig. 1. $currF$ is the current frame read, $Canny-edge$ variable Contain the Canny edges, HI is the histogram image, HOG is the histogram of oriented gradients [41], $HOOF$ is the Histograms of oriented optical flow.

```
% Matlab Code Listings of features
currF=rgb2gray(currFrame);
Canny-edge=mean2(edge(currF,'canny'));

HI1=mean(imhist(currF));

HOG = gradientHistogram(hx,hy',bin);

HOOF(1:size(ohog,1))=HOG;

CA{t}=[HOG HOOF Canny-edge HI1];
```

C. Dictionary of Key Frames

The atom selection for the dictionary is done using a similar approach as followed in [2, 1], We have selected 50 frames for dictionary comparison, the 50 frame is a selected as a computational limit, The current frame feature values are compared against and previous frames value as indicted in equation 4, where pre is the previous frames feature value, cu is the current frames feature value, the α is calculated by the ABC algorithm as indicated in the algorithm section, λ is initialized to a small value of 0.01, 50 k atoms. Dictionary selection is again a great way to start the summarization with good representation from the video data, the dictionary initialization is discussed in [42, 43, 44].

$$\min_{x} \& f(x) = \sum_{i=1}^{k}(((pre - cu) \times \alpha)) + (\lambda \times \sum_{i=1}^{k}(\alpha)); \quad (4)$$

D. Threshold as the Reconstruction Error

The threshold α is calculated as a mean of the 50 frames in the current cycle comparison from the dictionary, as we increment by 50 frames for the next comparison. The threshold α as compared with the value from equation 4 when there is a higher reconstruction error (higher value of α), we include the frame for summarization.

Algorithm 1 ABC Algorithm

$CostFunction \leftarrow @(x)Sphere(x)$;
$nVar \leftarrow 5$; $VarMin \leftarrow -10; VarMax \leftarrow 10$;
$MaxIt \leftarrow 10; nPop \leftarrow 10$;
$L = round(0.6 * nVar * nPop); (TrialLimit)$
$InitializationPopulationArray$
$pop \leftarrow repmat(emptybee, nPop, 1)$;
$\#InitBestSolutionEverFound\ BestSol.Cost \leftarrow \inf$;
$\#CreateInitialPopulationbyrandomsample$
for $i \leftarrow 1 : nPop$ **do**
  $updatethebestcost$
  $BestSol = pop(i)$;
end for
$\#ABCMainLoop$
$\#Choosepartner\ K\ randomly, != i$
for $it \leftarrow 1 : MaxIt$ **do**
  for $i \leftarrow 1 : nPop\#RecruitedBees$ **do**
    $\#NewBeePosition$ *by eqn 1*
    $newbee.Position \leftarrow pop(i).Position + \phi \times (pop(i).Position - pop(k).Position)$;
  end for
  #Calculate Fitness Values and Selection Prob
  for $i \leftarrow 1 : nPop$ **do**
    $F(i) \leftarrow exp(-pop(i).Cost\%MeanCost)$;
  end for
  Onlooker Bees
  for $m \leftarrow 1 : nOnlooker$ **do**
    $newbee.Cost \leftarrow CostFunction(newbee.Position)$;
  end for
  #Scout Bees
  for $i \leftarrow 1 : nPop$ **do**
    $pop(i).Cost \leftarrow C - Function(pop(i).Position)$;
  end for
  for $i \leftarrow 1 : nPop$ **do** $pop(i).Cost < BestSol.CostBestSol \leftarrow pop(i)$;
  end for
  $return\ alpha \leftarrow min(BestCost)$;

V. Experimental Result

In this section, we discuss the results obtained using the ABC optimization algorithm on a well-known dataset VSUMM [10]. The dataset consists of 50 videos from different genres and user summary keyframes for each video. In this experiment, we have compared the results for two user summaries and given the evaluation for each user summary against the automated summary generation as available in VSUMM dataset [10].

The average evaluation scores obtained in Table III indicate the approach using the ABC algorithm in a sparse-land approach is close to other results as compared to [10].

Fig. 3 depicts the results of one of the video # 30 from the VSUMM dataset giving a clear indication of the frame number matches and $+/-1$ frame matches, hence the results obtained demonstrate the approach for sparse-land based VS, a full framework for VS, anomaly detection, and online-highlighting. This approach is open to include any other Text/NLP [45, 46, 47, 48, 49, 50] based feature inputs. Frame importance rankings [45] with NLP caption generation methods [51, 52] combined with other video features are recent advancements in video summarization features [53, 50].

A. Evaluation of Video Summary

The evaluation is based on the proposed approach as discussed in [54, 10] called Comparison of User Summaries (CUS). The user summary is composed of many user summaries and taken a common score approach in the VSUMM dataset. The results in our approach called the automatic summary are compared with two user summaries as depicted in Fig. 3.

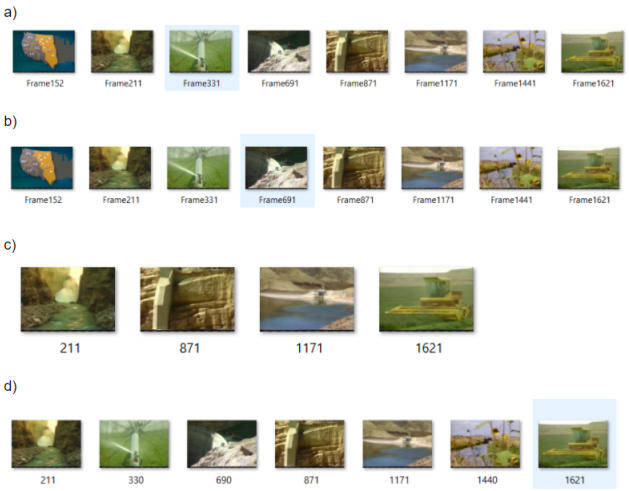

Fig. 3. CUS Evaluation Method for Video #30 in VSUMM Dataset, (a) User1 Summary (b) User2 Summary (c) Automatic Summary from ABC Method (d) Automatic Summary from ABC Method with +/- 1 Frame Number. Frame Numbers for Each Group is Listed Below

a) 152, 211, 331, 691, 871, 1171, 1441, 1621
b) 152, 211, 331, 691, 871, 1171, 1441, 1621
c) 211, 871, 1171, 1621
d) 211, 330, 690, 871, 1171, 1440, 1621

Precision, recall, and F1-score are the common metrics to measure the performance of the VS framework, the formulas are followed from [54, 10]. The evaluation metrics for precision, recall, F1-score is depicted in Tables I and II against both the user summary in VSUMM dataset. The equation depicted below 5, 6, 7 are used for the evaluation metrics with the automated summary generated by our approach, the comparison scores are mean accuracy rate CUSA(precision) Error rate CUSE(Recall), and F1-Score. The F1-score obtained by our approach is close enough to other methods [10], by balancing the threshold parameters in ABC algorithm we can improve the F1-Score, also we need to take care of other scores that get affected like precision and recall. Finding the right balance with all the parameters of our ABC approach for video summarization and evaluation by F1-Score is another open challenge.

$$Precision = \frac{N_{matched}}{N_{AS}} \tag{5}$$

$$Recall = \frac{N_{matched}}{N_{US}} \tag{6}$$

$$F1 - Score = \frac{2 \times P \times R}{P + R} \tag{7}$$

VI. Conclusion

In this work, we propose the ABC optimization algorithm for Video summarization to reduce long video to short video, removing redundant frames. We have compared the performance metrics for evaluations with the known dataset VSUMM. The comparison metrics have given a better score with other methods with reasonable performance. This method can be easily used for (quasi) real-time VS and anomaly detection, also extendable with other advanced sparse-land approaches as CSC (Convolutional Sparse Coding Model) [24], and K-SVD approaches [55, 56]. Finding an optimal threshold function or value for summarization is still open as the performance measure gets affected as we decrease or increase the threshold function.

References

[1] B. Zhao and E. P. Xing, "Quasi real-time summarization for consumer videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2513–2520.
[2] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *CVPR 2011*. IEEE, 2011, pp. 3313–3320.
[3] T. Hussain, K. Muhammad, A. Ullah, Z. Cao, S. W. Baik, and V. H. C. de Albuquerque, "Cloud-assisted multiview video summarization using cnn and bidirectional lstm," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 77–86, 2019.
[4] R. Panda and A. K. Roy-Chowdhury, "Multi-view surveillance video summarization via joint embedding and sparse optimization," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2010–2021, 2017.
[5] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1600–1607.
[6] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan, "From keyframes to key objects: Video summarization by representative object proposal selection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1039–1048.
[7] S. Mei, G. Guan, Z. Wang, S. Wan, M. He, and D. D. Feng, "Video summarization via minimum sparse reconstruction," *Pattern Recognition*, vol. 48, no. 2, pp. 522–533, 2015.
[8] F. Dornaika and I. K. Aldine, "Decremental sparse modeling representative selection for prototype selection," *Pattern Recognition*, vol. 48, no. 11, pp. 3714–3727, 2015.
[9] D. S. K. Vinsent Paramanantham, "A real time video summarization for youtube videos and evaluation of computational algorithms for their time and storage reduction," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 6, no. 4, pp. 176–186, 2018.
[10] S. E. F. De Avila, A. P. B. Lopes, A. da Luz Jr, 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.

[11] A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Transactions on circuits and systems for video technology*, vol. 9, no. 8, pp. 1280–1289, 1999.
[12] A. Packialatha and A. Chandrasekar, "Effective video summarization using eigen based classification," *Transylvanian Review*, no. 2, 2016.
[13] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *ECCV*. Springer, 2016.
[14] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," *arXiv preprint arXiv:1801.00054*, 2017.
[15] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, "Summarizing videos with attention," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 39–54.
[16] P. Koutras and P. Maragos, "Susinet: See, understand and summarize it," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
[17] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, 2015, pp. 843–852.
[18] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo, "Unsupervised extraction of video highlights via robust recurrent auto-encoders," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4633–4641.
[19] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial lstm networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 202–211.
[20] Y. He, C. Gao, N. Sang, Z. Qu, and J. Han, "Graph coloring based surveillance video synopsis," *Neurocomputing*, vol. 225, pp. 64–79, 2017.
[21] Z. Ji, Y. Zhang, Y. Pang, and X. Li, "Hypergraph dominant set based multi-video summarization," *Signal Processing*, vol. 148, pp. 114–123, 2018.
[22] M. Paul and M. M. Salehin, "Spatial and motion saliency prediction method using eye tracker data for video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.
[23] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis, "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos," in *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on*. IEEE, 2009, pp. 2012–2019.
[24] I. Rey-Otero, J. Sulam, and M. Elad, "Variations on the convolutional sparse coding model," *IEEE Transactions on Signal Processing*, vol. 68, pp. 519–528, 2020.
[25] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.
[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.
[27] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973.
[28] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.
[29] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.
[30] T. Bhattacharjee, S. Saha, A. Konar, and A. K. Nagar, "Static video summarization using artificial bee colony optimization," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 777–784.
[31] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Video summarization using deep semantic features," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 361–377.
[32] F. Bach, J. Mairal, J. Ponce, and G. Sapiro, "Sparse coding and dictionary learning for image analysis," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2010.
[33] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 2272–2279.
[34] E. Elhamifar, G. Sapiro, and R. Vidal, "Sparse modeling for finding representative objects," *preparation*, vol. 4, no. 6, p. 8.
[35] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 66–75, 2011.
[36] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm," *Applied soft computing*, vol. 8, no. 1, pp. 687–697, 2008.
[37] J. G. Allen, R. Y. Xu, J. S. Jin *et al.*, "Object tracking using camshift algorithm and multiple quantized feature spaces," in *ACM International Conference Proceeding Series*, vol. 100. Citeseer, 2004, pp. 3–7.
[38] A. D. Mohammed and T. Morris, "A robust visual object tracking approach on a mobile device," in *Information and Communication Technology-EurAsia Conference*. Springer, 2014, pp. 190–198.
[39] C. Zhang, Y. Qiao, E. Fallon, and C. Xu, "An improved camshift algorithm for target tracking in video surveillance," in *Conf. of 9th. Information Technology & Telecommunication*, 2009, pp. 19–26.
[40] P. Korshunov and W. T. Ooi, "Reducing frame rate for object tracking," in *International Conference on Multimedia Modeling*. Springer, 2010, pp. 454–464.
[41] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.
[42] R. Rubinstein, M. Zibulevsky, and M. Elad, "Learning sparse dictionaries for sparse signal approximation," Computer Science Department, Technion, Tech. Rep., 2009.
[43] ——, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on signal processing*, vol. 58, no. 3, pp. 1553–1564, 2009.
[44] S. Ibrahem, Y. M. Abd El-Latif, and N. M. Reda, "Anovel data dictionary learning for leaf recognition."
[45] B. A. Plummer, M. Brown, and S. Lazebnik, "Enhancing video summarization via vision-language embedding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5781–5789.
[46] S. Yeung, A. Fathi, and L. Fei-Fei, "Videoset: Video summary evaluation through text," *arXiv preprint arXiv:1406.5824*, 2014.
[47] S. Sah, S. Kulhare, A. Gray, S. Venugopalan, E. Prud'Hommeaux, and R. Ptucha, "Semantic text summarization of long videos," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 989–997.
[48] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *European conference on computer vision*. Springer, 2014, pp. 540–555.
[49] A. Sharghi, B. Gong, and M. Shah, "Query-focused extractive video summarization," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–19.
[50] A. Sharghi, N. d. V. Lobo, and M. Shah, "Text synopsis generation for egocentric videos," *arXiv preprint arXiv:2005.03804*, 2020.
[51] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8739–8748.
[52] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu, "Bidirectional attentive fusion with context gating for dense video captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7190–7198.
[53] 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.
[54] M. Guironnet, D. Pellerin, N. Guyader, and P. Ladret, "Video summarization based on camera motion and a subjective evaluation method," *EURASIP Journal on Image and Video Processing*, vol. 2007, no. 1, p. 060245, 2007.
[55] A. Mohammed, S. Yildirim, M. Pedersen, Ø. Hovde, and F. Cheikh, "Sparse coded handcrafted and deep features for colon capsule video summarization," in *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2017, pp. 728–733.
[56] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

APPENDIX

Results of the ABC algorithm can be found in the following links and a short Video Description for the VS processing: https://github.com/VinACE/ABC-VSUMM

TABLE I. EVALUATION METRICS AGAINST USERSUMMARY 1 (VSUMM1 SUMMARY).

	Automatic summary			Automatic Summary with +/- 1 Frame		
Video #	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
v21	1	0.53	0.69	1	0.6	0.75
v22	0.25	0.25	0.25	0.75	0.75	0.75
v21	1	0.53	0.69	1	0.6	0.75
v22	0.25	0.25	0.25	0.75	0.75	0.75
v23	0.33	0.33	0.33	0.46	0.46	0.46
v24	0.72	0.72	0.72	0.72	0.72	0.72
v25	1	0.5	0.67	1	0.6	0.75
v26	0.41	0.41	0.41	0.76	0.76	0.76
v27	0.54	0.54	0.54	0.7	0.69	0.69
v28	1	0.95	0.97	1	1	1
v29	0.44	0.44	0.44	0.55	0.55	0.55
v30	0.5	0.5	0.5	0.88	0.88	0.88
v31	0.27	0.27	0.27	0.36	0.36	0.36
v32	0.73	0.73	0.73	0.82	0.82	0.82
v33	0.14	0.14	0.14	0.47	0.47	0.47
v34	0.78	0.78	0.78	1	1	1
v35	0.73	0.73	0.73	0.73	0.73	0.73
v36	0.63	0.63	0.63	0.63	0.63	0.63
v37	0.2	0.2	0.2	0.2	0.2	0.2
v38	0.64	0.64	0.64	0.71	0.71	0.71
v39	0.36	0.36	0.36	0.36	0.36	0.36
v40	0.54	0.54	0.54	0.92	0.92	0.92
v41	0.8	0.8	0.8	1	1	1
v42	1	1	1	1	1	1
v43	0.68	0.68	0.68	0.81	0.81	0.81
v44	0.54	0.54	0.54	0.73	0.73	0.73
v45	0.14	0.14	0.14	0.29	0.29	0.29
v46	1	1	1	1	1	1
v47	0.8	0.8	0.8	0.8	0.8	0.8
v48	0.86	0.86	0.86	1	1	1
v49	0.46	0.46	0.46	0.54	0.54	0.54
v50	0.75	0.75	0.75	0.86	0.86	0.86
v51	0	0	0	0.14	0.14	0.14
v52	0.625	0.625	0.625	0.875	0.875	0.875
v53	0.4	0.4	0.4	1	1	1
v54	0.86	0.86	0.86	1	1	1
v55	0.5	0.5	0.5	0.5	0.5	0.5
v56	0.5	0.5	0.5	0.75	0.75	0.75
v57	0	0	0	0.1	0.1	0.1
v58	0.78	0.78	0.78	1	1	1
v59	0.5	0.5	0.5	1	1	1
v60	0.56	0.56	0.56	0.91	0.91	0.91
v61	0.71	0.71	0.71	1	1	1
v62	0.75	0.75	0.75	1	1	1
v63	0	0	0	0.22	0.22	0.22
v64	0.93	0.93	0.93	1	1	1
v65	1	1	1	1	1	1
v66	0.78	0.78	0.78	1	1	1
v67	0.86	0.86	0.86	1	1	1
v68	1	1	1	1	1	1
v69	1	1	1	1	1	1
v70	0.6	0.6	0.6	1	1	1
Mean Score	**0.6119**	**0.5915**	**0.5985**	**0.7709**	**0.7547**	**0.7607**

TABLE II. EVALUATION METRICS AGAINST USERSUMMARY 2 (VSUMM2 SUMMARY).

	Automatic summary			Automatic Summary with +/- 1 Frame		
Video #	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
v21	1	0.5	0.67	1	0.75	0.86
v22	0.25	0.25	0.25	0.75	0.75	0.75
v23	0.31	0.31	0.31	0.46	0.46	0.46
v24	0.67	0.67	0.67	0.67	0.67	0.67
v25	1	0.56	0.71	1	0.67	0.8
v26	0.5	0.5	0.5	0.83	0.83	0.83
v27	0.33	0.33	0.33	0.44	0.44	0.44
v28	1	0.94	0.97	1	1	1
v29	1	0.94	0.98	1	1	1
v30	0.27	0.27	0.27	0.45	0.45	0.45
v31	0.25	0.25	0.25	0.5	0.5	0.5
v32	0.64	0.64	0.64	0.73	0.73	0.73
v33	0.23	0.23	0.23	0.69	0.69	0.69
v34	0.86	0.86	0.86	1	1	1
v35	0.67	0.67	0.67	0.67	0.67	0.67
v36	0.38	0.38	0.38	0.63	0.63	0.63
v37	0.33	0.33	0.33	0.33	0.33	0.33
v38	0.68	0.68	0.68	0.79	0.79	0.79
v39	0.33	0.33	0.33	0.33	0.33	0.33
v40	0.4	0.4	0.4	0.9	0.9	0.9
v41	0.78	0.78	0.78	1	1	1
v42	1	1	1	1	1	1
v43	0.7	0.7	0.7	0.8	0.8	0.8
v44	0.63	0.63	0.63	0.88	0.88	0.88
v45	0.16	0.16	0.16	0.33	0.33	0.33
v46	1	1	1	1	1	1
v47	0.75	0.75	0.75	0.75	0.75	0.75
v48	0.86	0.86	0.86	1	1	1
v49	0.55	0.55	0.55	0.67	0.67	0.67
v50	0	0	0	0	0	0
v51	0.25	0.25	0.25	0.5	0.5	0.5
v52	0.5	0.5	0.5	0.83	0.83	0.83
v53	0.25	0.25	0.25	1	1	1
v54	0.86	0.86	0.86	1	1	1
v55	0.5	0.5	0.5	0.5	0.5	0.5
v56	0.4	0.4	0.4	0.6	0.6	0.6
v57	0	0	0	0.12	0.12	0.12
v58	0.89	0.89	0.89	1	1	1
v59	0.375	0.375	0.375	1	1	1
v60	0.56	0.56	0.56	1	1	1
v61	0.75	0.75	0.75	1	1	1
v62	1	1	1	1	1	1
v63	0	0	0	0.17	0.17	0.17
v64	0.92	0.92	0.92	1	1	1
v65	1	1	1	1		1
v66	0.78	0.78	0.78	1	1	1
v67	0.83	0.83	0.83	1	1	1
v68	1	1	1	1	1	1
v69	1	1	1	1	1	1
v70	0.6	0.6	0.6	1	1	1
Mean Score	**0.5999**	**0.5787**	**0.5865**	**0.7664**	**0.7348**	**0.7596**

TABLE III. MEAN ACCURACY RATE CUSA (PRECISION) AND MEAN ERROR RATE CUSE (RECALL) COMPARED AGAINST OTHER METHODS [10] AND OURS.

	OV	**DT**	**STIMO**	**VSUMM1**	**VSUMM2**	**Our-Summ1**	**Our-Summ2**
$Precision/CUS_A$	0.7	0.53	0.72	0.85	0.7	**0.61**	**0.59**
$Recall/CUS_E$	0.7	0.53	0.72	0.85	0.7	**0.61**	**0.59**