Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 1, 2025

Using Deep Learning REST APIs, Vid-Sum summarizes videos

[1] B. Santhosh, [2] P. Karthik, [3] V. Kranthi Kumar, [4] Ch. Akash, [5] N. Jeevan Reddy, [6] Dr. Nidamanuru Srinivas Rao, [7] Mr.V. Pradeep Kumar,

[1,2,3,4,5]UG Scholar, Dept. of CS, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

[6] Associate Professor, Dept. of CSE, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

[7] Assistant Professor, Dept. of EEE, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

Abstract—

The goal of video summarizing is to condense a lengthy film into a concise one that conveys the main points and mood of the source material. This is useful since it allows us to get the main points from a shorter film without having to watch the whole thing. The majority of supervised learning video summarization algorithms now employ CNNs, with a few also using recurrent neural networks. We present VidSum, a Deep Learning architecture for video summarization. For video summarization, we use a combination of convolutional neural networks and long short-term memory (LSTM) networks. With the help of our deep learning model, we can identify which frames are most significant at certain points in time and then create video summaries that are both coherent in terms of time and include all of the relevant information from the movie. When tested on the well-known TVSum and SumMe datasets, our model achieved better results than competing algorithms when it came to video summarization. The following terms are associated with deep learning: supervised learning, video summarization, LSTM (long short term memory), and convolutional neural networks (CNNs).

I. INTRODUCTION

We engage with and consume a great deal of video data these days because of the proliferation of social media and online video providers such as Netflix and YouTube. The internet receives hundreds of hours of video content every minute. Beyond that, video data has multiplied in the last decade and is still growing at an incredible pace, all because cameras are everywhere, from inexpensive CCTV security cameras to our cellphones. Finding effective video summarizing algorithms that extract the most important parts of a film and make human labor unnecessary has therefore become a pressing need. By providing a brief overview of a movie, video summarizing allows for more efficient viewing of massive video datasets. Reducing the original video to a few key frames can make videos more educational, impressive, and engaging. There are a plethora of practical uses for video summarization. Video surveillance is one example that is highly relevant. Humans aren't particularly efficient when it comes to watching 24-or even 72-hour video surveillance recordings. Thanks to Video Summarization, investigators can save countless hours of footage by simply watching a condensed version of the entire surveillance feed. The model can sift through hours of footage of a parking lot, for instance, and pull out the crucial moments to include in the summary. The court system would greatly benefit from this as it would speed up investigations and be quite useful in legal situations. Automated movie trailers may also be made using Video Summarization. Directors and movie editors would have less work to do if this happened. You may use it to make stunning video highlights from longer events, like weddings and birthday parties, instead of watching the entire 6-7 hour lengthy footage. In addition, we can use video summarization to make better image and video carousels or combined short 1-2 minute videos right on our phones from our photo galleries. It's very similar to how we make them now, but with better results. Thus, video summarization is highly beneficial in the real world and is a continuous field of study, with many researchers contributing to it and working on it.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 1, 2025

II. RELATED WORK

For video summary, several various methods have been used. Unsupervised Learning [16] was used by Kanafani et al. [2] to summarize videos. Similarly, Shemer et al. [13] and Jadon et al. [12] have used Unsupervised Learning to extract crucial moments from videos.
Besides that, a lot of research articles on video summarization have made use of supervised learning, including methods like recurrent neural networks [18], convolutional neural networks [17], etc. [20]. Video summaries have been created using supervised learning by Zhu et al. [4] and Elfeki et al. [15]. Convolutional Neural Networks have been used by Huang et al. [14], Rochan et al. [10] and Fajtl et al. [11], while Vasudevan et al. [9] have utilized both Convolutional and Recurrent Neural Networks. For video summary, several really novel methods have also been used. Graphs, in which nodes represent specific parts of a video, have been used by Papalampidi et al. [7]. Famous TV program videos have had their text and colors extracted by Hohman et al. [8] for the purpose of summarization. When training their networks to generate video summaries, Apostolidis et al. [3] and Fu et al. [1] made use of General Adversarial Networks [22].

A small number of scholars have also turned to Reinforcement Learning [21] in an effort to address the issue of video summarization. Zhou et al. [6] and Lan et al. [5] trained their deep learning models using Reinforcement Learning. Video summary has been subjected to every conceivable learning technique, including unsupervised, supervised, and reinforcement learning. However, supervised learning approaches are the most successful. We opted for supervised learning since, in this particular case, the model outperforms other approaches when it comes to learning. Among the many methods explored to address the issue of video summarization, supervised learning has produced some of the most promising outcomes to yet. Our model incorporates both Convolutional Neural Networks and Long Short-Term Memory (LSTM) Networks [19]. For the following reasons, we opted for LSTM networks rather than alternative options: A. Recurrent neural networks have the issue of vanishing gradients, which prevents them from learning on extended data sequences. LSTM is able to address the Vanishing Gradient Problem because it is directly related to the forget gate's activation function.
B. On huge datasets, LSTM Networks outperform Gated Recurrent Units (GRUs) [23]. Even with massive datasets, their accuracy is excellent. C. In contrast to Convolutional Neural Networks (CNNs), which are limited to finding spatial characteristics and patterns in image and audio data, Long Short-Term Memory (LSTM) Networks can handle sequential data and formulate predictions based on it.

III. PROPOSED APPROACH

Partitioning a movie into its individual frames is the first step in our suggested technique. Following their input into the CNN, Convolutional Layers of the network retrieve spatial characteristics from the video frames. The LSTM network is then used to assess the time-based relevance of these frames based on the retrieved characteristics. The next step is to determine if each of these images will be included in the final summary by calculating their Temporal Intersection over Union with the ground truth. If they are, then they are classified as yes. The last stage is to construct the summary movie by compiling all the photos that were classed as yes.
There are five stages to our suggested solution:

A. Extracting Frames from video file:

A video is really just a collection of video sequences, and each of those sequences contains pictures. Consequently, we start by segmenting the video. Afterwards, we take still pictures out of the video. As a result, we can see a video in its entirety by examining its individual frames. We use the Python module OpenCV for this purpose. There are typically 24 frames per second (fps) in CCTV footage, meaning that there are 24 pictures captured in a single second. As a result, the model incorporates 24 still photos for every second of video. Additionally, our model is compatible with 30fps and 60fps.

B. Feature Extraction from the extracted frames:

These retrieved frames are essential for our model to learn the characteristics. We do this by extracting spatial characteristics from the photos using convolutional neural networks. Following the aforementioned process of video frame extraction, our model now begins to learn. To prevent the model from being overfit, we have additionally included dropout layers. Feature extraction from picture frames by our model is provided by the output of these layers.

C. Temporal interest proposals:

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 1, 2025

Finding the most crucial frames to include in the summary is the next step. Here, we use Long Short Term Memory (LSTM) cells. Since LSTM cells may also learn from their predecessors, we employ them. The time-based interest video segments are crucial to the success of any film or video. To establish the significance of a scene, we must be aware of the scene that came before it. Because we employed LSTM, our model has "memory" and can recall prior sequences, allowing it to more accurately identify which parts of the movie are most relevant to the summary. The model then uses LSTM to suggest, in a time-coherent fashion, which frames are interesting and significant.

D. Classification on the basis of Temporal Intersection over Union:

At last, we choose whether a frame will be included in the final summation. We consider a proposal to be positive and assign it a score of 1 if its Temporal Intersection over Union (tIOU) is greater than 0.6 when compared with the Ground Truth summary frames provided in the dataset. If this is the case, the proposal is included in the final summary. Any frame with a Temporal Intersection over Union value below 0.6 is excluded from the final summary since it is deemed negative and assigned a score of 0. Therefore, we have maintained the Temporal Intersection-Over-Union Threshold at 60% when training our model.

E. Amalgamation of Frames:

To create the summary video, you must collect all the frames that are good and have been awarded a score of 1. The final video summary is produced by merging these frames using the OpenCV library. On display in Figure 1 is the flow diagram depicting our suggested method for video summarization, which we call VidSum.

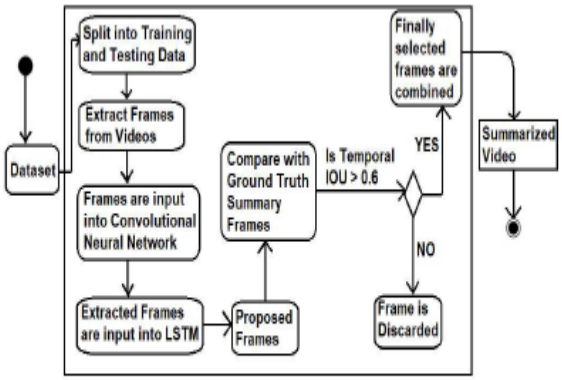

Fig. 1. Proposed Approach Flowchart of VidSum - Video Summarization using Deep Learning

IV. IMPLEMENTATION

These are the steps we implemented for our proposed solution:

A. Collecting the datasets:

Training data is necessary for us to train our model. Video clips and ground truth summaries provided by humans are required for this. We need this to ensure that the model can learn to prioritize video frames and to compare the recommended frames to the ground truth frames in terms of their temporal intersection and union. Consequently, we gathered the well-known TVSum [24] and SumMe [25] datasets, which are used for video summarizing and feature human summaries for each video across different genres. Testing data is also necessary for validation reasons; this will allow us to examine our model's efficacy in generating temporally consistent summaries and in proposing summary frames. To facilitate testing and training, we partitioned the TVSum and SumMe datasets accordingly. Using human-created ground truth user summaries as a guide, we sorted all of the video files and verified that none of them were corrupt. This must be completed in order for the model to acquire the necessary knowledge and skills from the input movies.

B. Preprocessing the input video files:

We developed a python script to pull still pictures out of the video. For this, we relied on the OpenCV Python package. In order for the Convolutional Neural Network to gain knowledge from these retrieved frames, this is carried out.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 1, 2025

C. Detecting and Extracting Features:

Using Maxpool2D layers interspersed with Convolutional layers, we developed a unique machine learning model. By training on the pictures' spatial properties, the Convolutional Neural Network improves its ability to forecast which frame should be chosen for the summary movie. Our convolutional neural network (CNN) is trained and features are extracted from the input video frames in Fig. 2.

```
20:18:52] Epoch: 18/300 Loss: 0.6001/0.6685/1.2686
20:18:55] Epoch: 19/300 Loss: 0.5809/0.6502/1.2311
20:18:57] Epoch: 20/300 Loss: 0.5701/0.6431/1.2132
20:19:00] Epoch: 21/300 Loss: 0.5741/0.6385/1.2127
20:19:02] Epoch: 22/300 Loss: 0.5608/0.6241/1.1849
20:19:05] Epoch: 23/300 Loss: 0.5562/0.6175/1.1737
20:19:07] Epoch: 24/300 Loss: 0.5497/0.6143/1.1640
20:19:09] Epoch: 25/300 Loss: 0.5456/0.5866/1.1322
20:19:12] Epoch: 26/300 Loss: 0.5324/0.6024/1.1348
20:19:14] Epoch: 27/300 Loss: 0.5167/0.6271/1.1438
20:19:17] Epoch: 28/300 Loss: 0.5318/0.5948/1.1265
20:19:19] Epoch: 29/300 Loss: 0.5243/0.6057/1.1300
20:19:21] Epoch: 30/300 Loss: 0.5178/0.5773/1.0951
20:19:24] Epoch: 31/300 Loss: 0.5020/0.5733/1.0753
20:19:26] Epoch: 32/300 Loss: 0.4968/0.5636/1.0604
20:19:28] Epoch: 33/300 Loss: 0.5022/0.5620/1.0641
20:19:31] Epoch: 34/300 Loss: 0.4909/0.5665/1.0574
20:19:33] Epoch: 35/300 Loss: 0.4924/0.5656/1.0579
```

Fig. 2. Convolutional Neural Network training and extracting features from the frames

D. Reducing Overfitting:

We included 0.5-chance dropout layers between the convolutional layers after realizing our model was overfitting. As a result, overfitting is lessened. The spatial characteristics retrieved from the picture frames by our model are given to us as output by these layers.

E. Feeding these frames into LSTM:

We have used LSTM to get a better understanding of the frames that are coherent in terms of time after we have extracted spatial features using convolutional layers. The "Memory" feature of LSTM allows it to recall the scene that came before the current one, allowing it to assess the significance of the current frame. The LSTM layers provide the output frames that they believe should be included of the video summary. These are the frameworks for proposals of temporary relevance.

F. Comparing with ground truth summary video frames:

In order to determine the Temporal Intersection over union, we compare the frames from the ground truth summary video in the dataset with the frames from the suggested frames (Fig. 3).

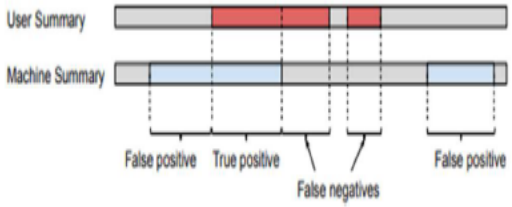

Fig. 3. Comparing Proposed Frames with Ground Truth Summary Frames

G. Classifying these frames on the basis of Temporal IOU:

For a frame to be included in the final summary movie, its Temporal Intersection over Union must be greater than 0.6. We then mark it as positive. If a frame's Temporal Intersection over Union is less than 0.6, we mark it as negative and remove it from the final summary video. Figure 4 shows the results of the calculation of the Temporal IOU (Intersection Over Union).

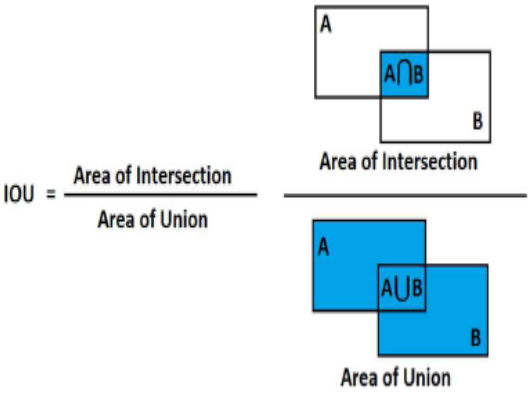

Fig. 4. Calculating Temporal Intersection over Union

H. Creating Summary Video:

The frames that had a Temporal Intersection Over Union (tIOU) more than 0.6 are included in the summary movie, but the frames with a tIOU less than 0.6 are removed. The next step is to merge the frames. To create the summary movie, all of the chosen frames are blended together in the proper

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 1, 2025

sequence. For this, we rely on the OpenCV python package.

I. Training the model:

Over the course of 30 hours, we trained the model 10,000 times. In order to improve the model's accuracy, we trained it again after making many adjustments. Overfitting was prevented by including a couple dropout layers. The goal was to improve the model's ability to understand the crucial parts of a video clip and make it more generalizable. Our model is trained on the SumMe dataset (Fig. 5) and the TVSum dataset (Fig. 6).

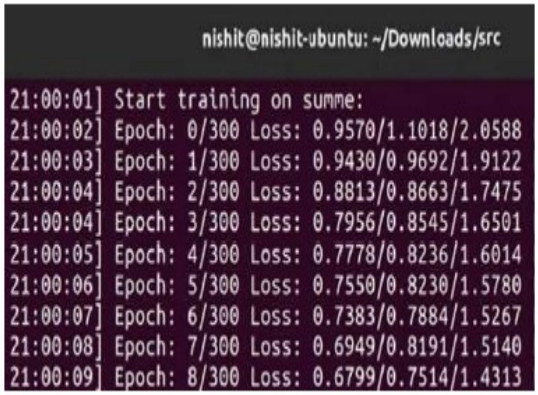

Fig. 5. Training our model on SumMe Dataset

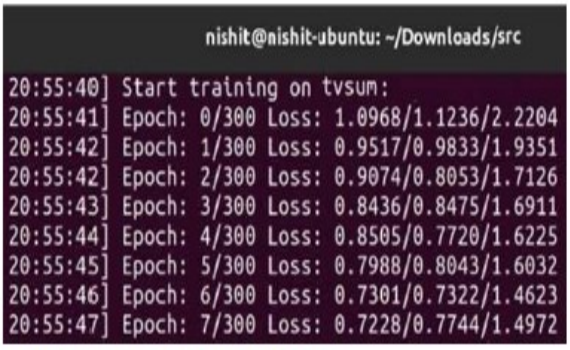

Fig. 6. Training our model on TVSum Dataset

V. TESTING AND OBSERVATIONS

A. Testing:

After separating the TVSum and SumMe datasets for training and testing, we put the model to the test by comparing it to the testing subset of the original datasets. To improve the model's accuracy, we tweaked its hyperparameters and adjusted its settings.

We iterated on the model's modifications and optimizations to boost the

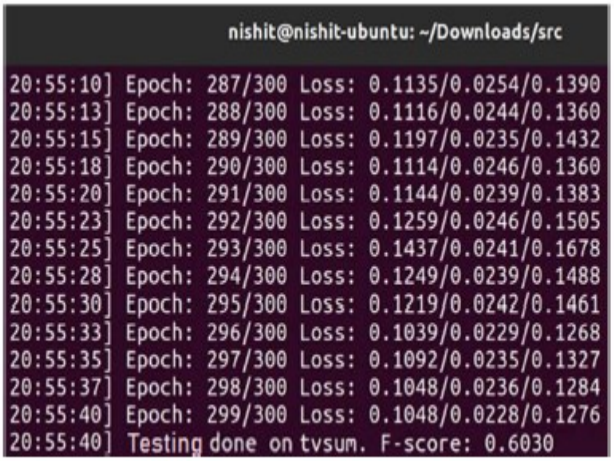

Fig. 7. Testing our model on TVSum Dataset

precision of the model. In the end, we were able to attain testing F-Scores of 60.3% on the TVSum Dataset and 48.9% on the SumMe Dataset, as seen in Figures 7 and 8, respectively.

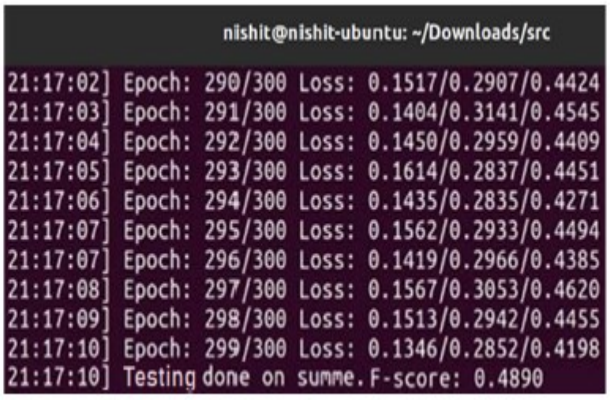

Fig. 8. Testing our model on SumMe Dataset

B. Observations:

We tried adjusting the Temporal IOU Threshold from 0.6 to 0.7 and 0.4, however the resulting summary video lacked temporal coherence and yielded poor results. We discovered that setting the Temporal IOU Threshold to 0.6 yielded the most effective summary films. Furthermore, as seen in Figure 9, the optimal value for the Temporal Intersection Over Union criterion was 0.6, which yielded the greatest F-Score. Therefore, the optimal value for the Temporal IOU Threshold was 0.6.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 1, 2025

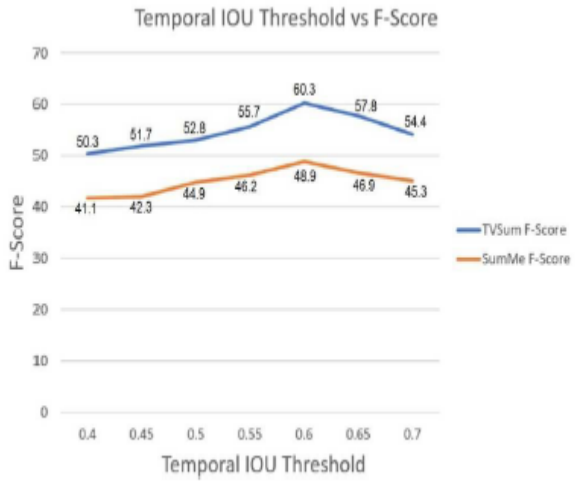

Fig. 9. Temporal IOU Threshold vs F-Score

VI. RESULTS

Ultimately, we were able to get testing FScores of 60.3% on the TVSum Dataset and 48.9% on the SumMe Dataset after thirty hours of training and fine-tuning the model. In terms of Video Summarization, they represent an improvement over earlier methods. Comparing our F-Score to those of other authors' work on the subject of video summarization using the TVSum and SumMe datasets is shown in Table I.

TABLE I. OUR FINAL F-SCORE COMPARED TO PREVIOUS WORKS

Model	TVSum F-Score	SumMe F-Score
FCSN (Rochan et al.) [10]	52.7	41.5
VASNet (J. Fajtl et al.) [11]	59.8	47.7
DR-DSN (Zhou et al.) [6]	57.6	41.4
Ours	60.3	48.9

Test Results: F- Figures 10 and 11, respectively, show the scores that our model earned on the TVSum and SumMe datasets.

```
20:55:33] Epoch: 296/300 Loss: 0.1039/0.0229/0.1268
20:55:35] Epoch: 297/300 Loss: 0.1092/0.0235/0.1327
20:55:37] Epoch: 298/300 Loss: 0.1048/0.0236/0.1284
20:55:40] Epoch: 299/300 Loss: 0.1048/0.0228/0.1276
20:55:40] Testing done on tvsum. F-score: 0.6030
```

Fig. 10. Final Testing F-Scores of our model on TVSum Dataset

```
21:17:07] Epoch: 296/300 Loss: 0.1419/0.2966/0.4385
21:17:08] Epoch: 297/300 Loss: 0.1567/0.3053/0.4620
21:17:09] Epoch: 298/300 Loss: 0.1513/0.2942/0.4455
21:17:10] Epoch: 299/300 Loss: 0.1346/0.2852/0.4198
21:17:10] Testing done on summe. F-score: 0.4890
```

Fig. 11. Final Testing F-Scores of our model on SumMe Dataset

Figure 12 shows the F-Scores our model earned on the SumMe Dataset, while Figure 13 shows the F-Scores on the TVSum Dataset, both of which were compared to other video summarization approaches. Images taken from a summary film that our algorithm generated from CCTV surveillance footage are shown in Figure 14.

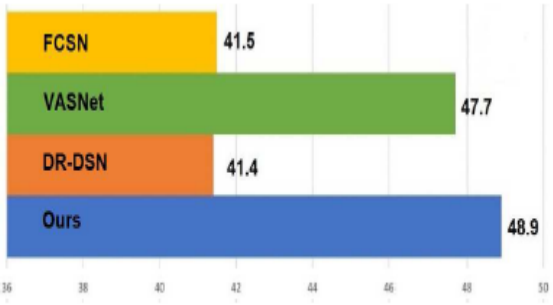

Fig. 12. Our Final F-Score as compared to other video summarization techniques on SumMe Dataset

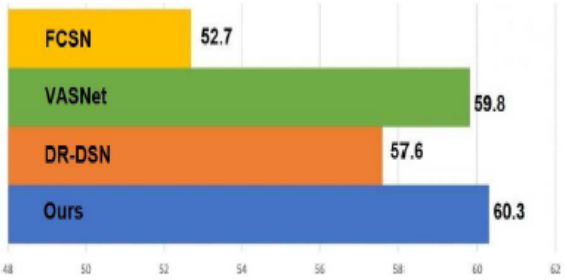

Fig. 13. Our Final F-Score as compared to other video summarization techniques on TVSum Dataset

VII. CONCLUSION

Consequently, our video summarizing methodology is presented in this study. Our video summarizing model approaches video summarization as a problem of temporal interest detection, in contrast to existing algorithms that just learn the significance score of each video frame. It outperforms state-of-the-art supervised models in video summarization thanks to its ability to learn temporal coherence between video frames. Ultimately, our model improved to the point where it could independently extract relevant video portions for inclusion in the summary and exclude

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 1, 2025

irrelevant sequences altogether. Since our model could effectively generate a high-quality summary video including all the crucial elements of a video clip, we were able to enhance accuracy for the video summarizing issue. Our goal moving forward is to make our unified structure more interest-based coherent. In order to make our model work better, we might also try to make the transitions between scenes in a video more seamless in time.

Fig. 14. Screenshots from summary video created by our model on CCTV surveillance footage

REFERENCES

[*1] T. -J. Fu, S. -H. Tai and H. -T. Chen, "Attentive and Adversarial Learning for Video Summarization," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2019, pp. 1579-1587, doi: 10.1109/WACV.2019.00173.*

[2] Kanafani, Hussain, et al. Unsupervised Video Summarization via Multi-Source Features. arXiv, 26 May 2021. arXiv.org, https://doi.org/10.48550/arXiv.2105.12532.

[3] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris and I. Patras, "AC-SUM-GAN: Connecting Actor-Critic and Generative Adversarial Networks for Unsupervised Video Summarization," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 8, pp. 3278-3292, Aug. 2021, doi: 10.1109/TCSVT.2020.3037883.

[4] W. Zhu, J. Lu, J. Li, and J. Zhou. 2021. DSNet: A Flexible Detect-to- Summarize Network for Video Summarization. Trans. Img. Proc. 30 (2021), 948–962. https://doi.org/10.1109/TIP.2020.3039886

[5] S. Lan, R. Panda, Q. Zhu, A.K. Roy-Chowdhury. (2018). "FFNet: Video Fast-Forwarding via Reinforcement Learning". 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6771-6780.

[6] K. Zhou, Y. Qiao, and T. Xiang. 2018. "Deep reinforcement learning for unsupervised video summarization with diversityrepresentativeness reward". In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18). AAAI Press, Article 929, 7582–7589.

[7] P. Papalampidi, F. Keller, and M. Lapata, "Movie Summarization via Sparse Graph Construction", AAAI 2021, 2021

[8] Fred Hohman, Sandeep Soni, Ian Stewart, and John Stasko, 2017, A Viz of Ice and Fire: Exploring Entertainment Video Using Color and Dialogue, VIS4DH Workshop 2017

[9] Arun Balajee Vasudevan, Michael Gygli, Anna Volokitin, and Luc Van Gool, 2017, Query-adaptive Video Summarization via Quality-aware Relevance Estimation, MM '17: Proceedings of the 25th ACM International Conference on Multimedia, pp. 582–590, https://doi.org/10.1145/3123266.31232972017

[10] Mrigank Rochan, Linwei Ye and Yang Wang, 2018, Video Summarization Using Fully Convolutional Sequence Networks, ECCV 2018, 2018

[11] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino, 2019, Summarizing Videos with Attention

[12] Shruti Jadon, and Mahmood Jasim, 2020, Unsupervised video summarization framework using keyframe extraction and video skimming, 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 2020, doi: 10.1109/ICCCA49541.2020.9250764

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 1, 2025

[13] Yair Shemer, Daniel Rotmanand, and Nahum Shimkin, 2019, ILSSUMM: Iterated local search for unsupervised video summarization, 2020 25th International Conference on Pattern Recognition (ICPR), 2020, doi: 10.1109/ICPR48806.2021.9412068

[14] Jia-Hong Huang, and Marcel Worring, 2019, Query controllable video summarization, ICMR '20: Proceedings of the 2020 International Conference on Multimedia Retrieval, June 2020, pp. 242–250, doi: 10.1145/3372278.3390695

[15] Mohamed Elfeki, Liqiang Wang, and Ali Borji, 2019, Multi stream dynamic video summarization, WACV 2022, pp. 185-195

[16] Muhammad Usama, Junaid Qadir, Aunn Raza, and Hunain Arif, 2019, Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges, IEEE Access (Volume: 7), doi: 10.1109/ACCESS.2019.2916648

[17] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi, 2017, Understanding of a convolutional neural network, 2017 International Conference on Engineering and Technology (ICET), doi: 10.1109/ICEngTechnol.2017.8308186

[18] Alex Sherstinsky, 2018, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network, 2018

[19] Greg Van Houdt, Carlos Mosquera, Gonzalo Nápoles, 2020, A Review on the Long Short-Term Memory Model, Springer Artificial Intelligence Review, Dec 2020, doi: 10.1007/s10462-020-09838-1

[20] Vladimir Nasteski, 2017, An overview of the supervised machine learning methods, Dec 2017, doi: 10.20544/HORIZONS.B.04.1.17.P05

[21] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath, 2017, A Brief Survey of Deep Reinforcement Learning, 2017

[22] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, 2014, Generative Adversarial Nets, NeurIPS, 2014

[23] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, NeurIPS 2014, 2014

[24] TVSum Dataset, [Online], Available: http://people.csail.mit.edu/yalesong/tvsum/

[25] SumMe Dataset, [Online], Available: https://zenodo.org/record/4884870#.YorKa6hByUl