Video Summarization

Swathi Konduru, Vijayendra D Avina, Tashina Jinan

May 2023

Abstract

A study by Microsoft Research found that most viewers abandon online videos after only a few seconds. The viewers abandon the video mostly because of inaccurate content of their desire. Video summarization can help to capture the viewer's attention by providing a shorter and more engaging summary of the video content. Video summarization using machine learning involves using automatically identify the most important parts of a video and condense them into a shorter summary. Both supervised and unsupervised learning methods have been successfully used in video summarization tasks, including video skimming, keyframe selection, video captioning etc. The focus of the proposed work is to evaluate the performance of k-means clustering which is an unsupervised learning method and CNN-based LSTM model in terms of generating summarised videos. The generated summarized videos are compared to the user summaries in SumMe dataset. The Jaccard similarity score, which measures the overlap between the generated summary and the user data, was significantly higher for the unsupervised model compared to the supervised model. The study also demonstrated the robustness and generalizability of the approach by testing it on multiple datasets with varying characteristics. Overall, this research provides an effective approach for generating video summaries using both supervised and unsupervised learning techniques, which can handle diverse video content.

1 Introduction

The amount of data that is available on the internet has been increasing exponentially with the growing technology. According to the statistics, as of 2021, it is said that over 500 hours of video content is uploaded to YouTube every minute and it is expected to increase rapidly in the coming years [1]. It is difficult to find the relevant and the necessary content in the vast amount of data. When you search for a video tutorial on a particular topic, you come across multiple videos that have different lengths and cover various aspects of the process. Some videos might show you the video that explains everything from start to finish which is helpful for beginners, while others might focus on specific steps or techniques. A study conducted by Microsoft found that humans have an attention span of just 8 seconds, which is lower than that of a goldfish [2]. Therefore, with limited time and short attention span, it is overwhelming and time consuming to watch all these videos.

Video Summarization plays a crucial role in managing and accessing large video collections by providing a shorter version of a video while still maintaining the semantic meaning or the original essence of the video. It also helps in enhancing user experience by allowing them to quickly browse through the video content and find what they are interested in. Manual video summarization can be time consuming and subjective process that relies on the judgement of human editors.

In recent years, the use of advanced technologies such as Artificial Intelligence and machine learning for automated video summarization gained significant attention because of the faster and objective process. Automated video summarization using AI and machine learning has numerous applications, including video search, content recommendation, and video surveillance. By summarizing the video, the search and recommendation algorithms can provide more accurate and relevant results to the users. In the case of video surveillance, automated summarization can be used to quickly identify the critical events and activities, reducing the workload of the human operators.

With the growth in video content, there is an increasing need for efficient video summarization techniques to help users navigate and consume the content. The goal of the proposed work is to develop an AI-based video summarization system that can accurately and efficiently generate summaries of videos, enabling users to quickly get an overview of the content and determine whether it meets their needs. By contributing to the field of video summarization, this project aims to help users better manage and consume the growing library of online video content.

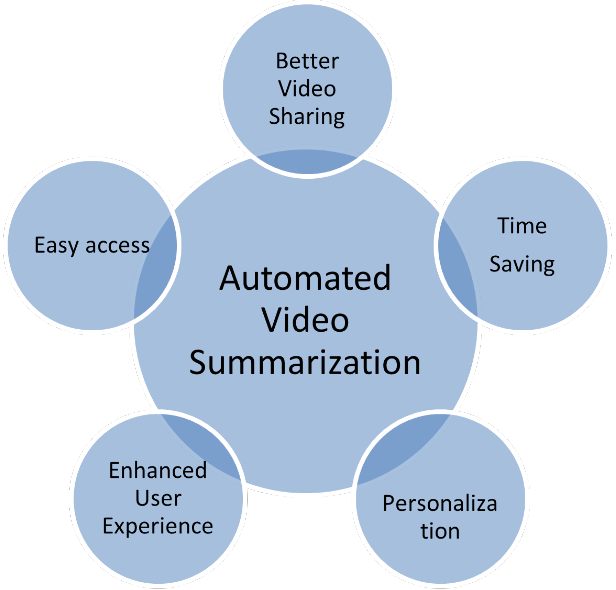


Figure 1: Advantages of Automated Video Summarization

Some of the advantages of Automated Video Summarization is illustrated in Figure 1

- Better Video Sharing: Summarized videos can be easily shared on social media platforms, making them more accessible and shareable.

- Time- saving: The lengthy videos are quickly summarized allowing users to get the content of the video without having to watch the whole video, thus saving time and help gain easy access to the content they need.

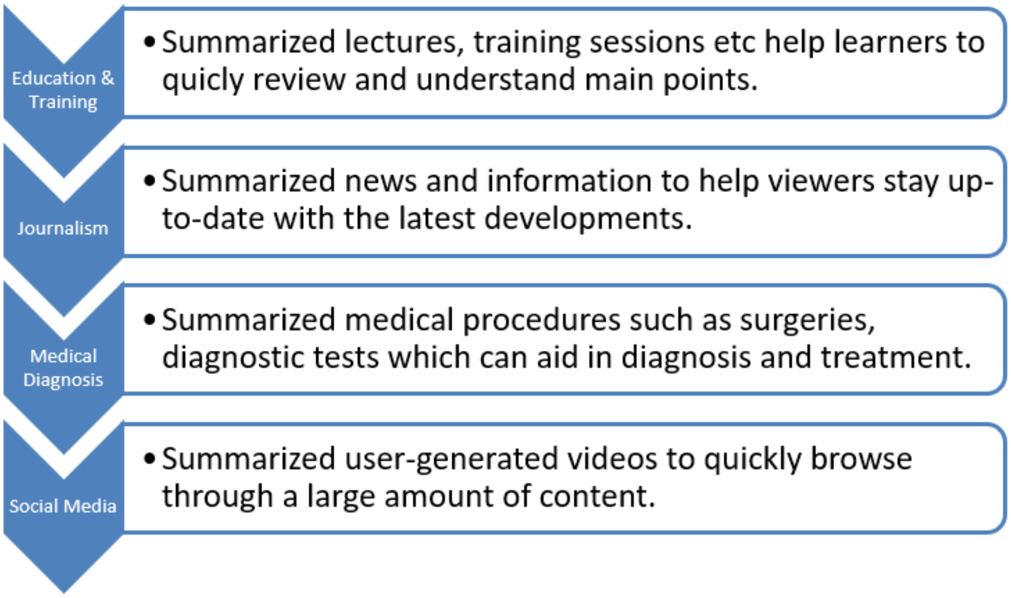
- Storage efficiency: Videos take up a lot of storage space and summarizing them can reduce the file size significantly. This is particularly important in industries like security and surveillance where large amounts of video data need to be stored.

- Enhanced user experience: Summarized videos provide an enhanced user experience by allowing users to digest the most important parts of a video quickly and easily.

Also, providing a condensed version of longer video of user's interest increases the user satisfaction with the recommendations, leading to an enhanced experience. Additionally, by analyzing the user's interaction with the video summaries, personalized recommendations can be refined further, providing more accurate and relevant content suggestions.

1.1 Applications

Video Summarization has various applications in different fields that include Entertainment where video highlights or recaps of sporting events, concerts, and other live performances can be created. It can also be used to create trailers, teasers, highlights of audio launch sessions for movies and TV shows. Surveillance and Security is another field where video summarization plays an important role as it can be used to quickly identify events of interest from a long video footage to identify criminal activity or security breaches. Other applications are illustrated in Figure 2



- Summarized lectures, training sessions etc help learners to quicly review and understand main points.
- Summarized news and information to help viewers stay up-to-date with the latest developments.
- Summarized medical procedures such as surgeries, diagnostic tests which can aid in diagnosis and treatment.
- Summarized user-generated videos to quickly browse through a large amount of content.

Figure 2: Applications of Automated Video Summarization

Video Summarization has been widely researched in the field of computer vision and machine learning. The earliest approaches focused on keyframe extraction which selects important frames from a video and presents them as a summary. Later, the research focused on extracting semantic features from video frames and using them to summarize the video content. Recent research has leveraged deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to automatically generate video summaries. Some approaches have used a supervised learning framework to train models on annotated video datasets, while others have employed unsupervised learning methods to identify key frames or segments without explicit labels. There has also been research exploring the use of reinforcement learning, attention mechanisms, and multi-modal fusion techniques for video summarization. Additionally, there have been efforts to incorporate user preferences and feedback to generate personalized video summaries. Some of the existing works are discussed in the literature review section below.

2 Literature Review

In [3], Gygli et al. proposed a method to generate video summaries automatically from user generated content. The proposed method extracts visual and audio features from the video, clusters similar segments, and selects important frames and audio segments to generate a summary. Support Vector Machines (SVM) was used to predict the important scores of the frames in the video while K-means clustering was used to group the frames into segments. MATLAB was used for implementation and the model was evaluated on the SumMe dataset using performance metrics such as F-score, precision, and Recall.

In [4], Otani et al. proposed a video summarization method that uses deep semantic features for better summarization. They extracted deep features from each video frame using the VGG-16 model, and then used these features to represent the video frames as vectors. They then applied k-means clustering to these vectors to group similar frames together and select representative frames from each cluster to create a summary.

In [5], Zhang et al. proposed a video summarization method that uses a Long Short-term Memory (LSTM) network which is a special type of RNN to learn to select important frames from a video. They trained the network on a set of annotated videos, and then used it to select frames that are representative of the video content. Here LSTM has shown its effectiveness to model the variable-range temporal dependency among video frames, so as to derive both representative and compact video summaries.

In [6], Zhou et al. proposed a video summarization method that uses deep reinforcement learning to select frames from a video. They trained a deep neural network to predict the value of a summary based on its diversity and representativeness and use this network as a reward signal for the reinforcement learning algorithm. The algorithm learns to select frames that maximize the reward, resulting in a summary that is both diverse and representative.

In [7], Apostolidis et al. proposed a video summarization method that uses a deep recurrent neural network to select important frames from a video. They trained the network on a set of annotated videos, and then used it to select frames that are representative of the video content. They also proposed a method for selecting the summary length based on the amount of information in the video.

In [8], Mahasseni et al. proposed an unsupervised video summarization method that uses adversarial training to learn a summary representation that maximizes the difference between the summary and the original video. They used a LSTM network to encode the video frames and train a discriminator network to distinguish between the summary and the original video. The generator network is trained to maximize the discriminator's error, resulting in a summary that captures the important content of the video.

Some of the limitations of the existing work on video summarization include:

- Lack of generalization: Many existing models are developed for specific video types or scenarios and may not generalize well to other types of videos.

- Overfitting: Some models may overfit to the training data and may not perform well on new, unseen data.

- Complexity: Some models are highly complex and require significant computational resources, making them impractical for real-time video summarization applications.

- Lack of interpretability: Some models may be difficult to interpret, making it hard to understand how they make their predictions and limiting their usefulness in certain applications.
- Limited scalability: Some models may not be scalable to large-scale video datasets, which can limit their usefulness in real-world applications.

The proposed work uses both supervised and unsupervised learning techniques to generate video summaries and to evaluate the performance of the techniques. The proposed work uses K-means clustering and Long Short-Term Memory to generate video summaries and the comparison of these two models can help identify the strengths and weaknesses of each approach and provide insights into when to use one approach over the other. The models are chosen after reviewing the existing work based on high performance. The proposed work evaluates the performance on SumMe dataset. This work explores the generalizability by testing the proposed approach on multiple datasets with varying characteristics. This exploration can demonstrate the robustness of your approach and its ability to handle diverse video content. The workflow of the proposed approached is discussed in the section below.

3 Methodology

The methodology for the proposed approach typically involves the following steps

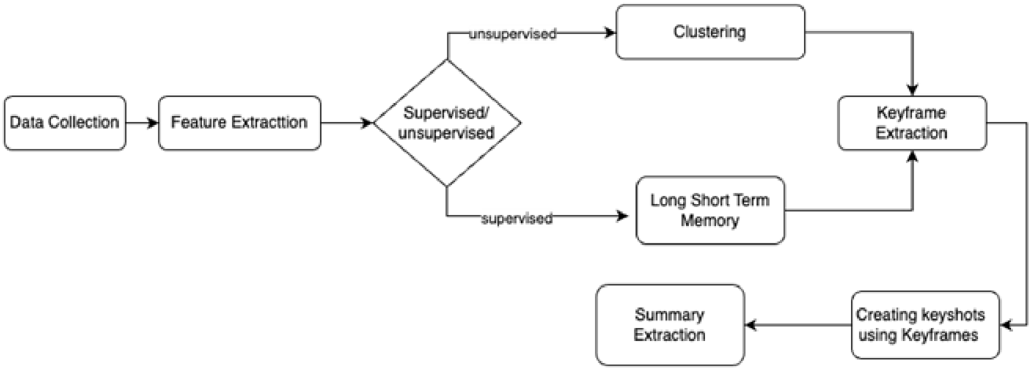


Figure 3: Overview of the proposed workflow.

3.1 Data collection

SumMe dataset was chosen for training and evaluating the summarization models. The SumMe dataset is a collection of 25 videos that have been manually annotated with keyframe-level importance scores by multiple human annotators. The videos were selected to cover a diverse range of content, such as documentaries, sports, and home videos.

3.2 Feature Extraction

The feature extraction process involves analyzing the color and motion features of video shots. First, the code initializes feature vectors to store the color and motion features of each frame within a shot. It then reads the first frame of a shot and computes its color feature using the mean color values. The Lucas-Kanade method is then used to compute the motion feature between the first and second frames. This process is repeated for all frames within the shot, with the motion features normalized and combined with the color features to create a shot feature vector. These shot feature vectors are then collected and returned as the features for the entire video.

3.3 Unsupervised Learning

In unsupervised learning, models are trained on unlabeled data to identify patterns that help classify, label and/or group the data points without any guidance or input from the user in performing the task. The goal is to fins clusters within the data to make predictions and meaningful insights. Some of the applications of unsupervised learning include anamoly detection, dimensionality reduction, density estimation, and feature extraction.

3.3.1 Clustering

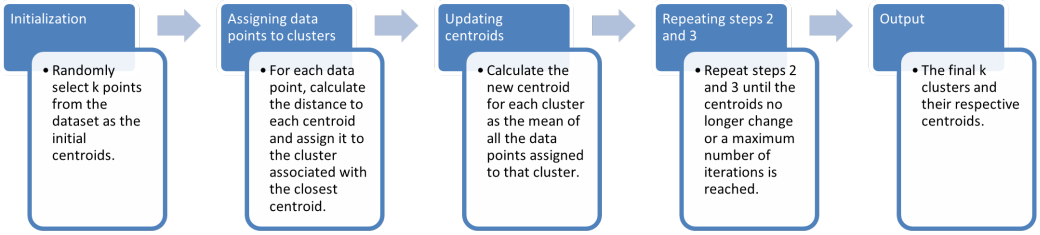

Initialization
• Randomly select k points from the dataset as the initial centroids.
Assigning data points to clusters
• For each data point, calculate the distance to each centroid and assign it to the cluster associated with the closest centroid.
Updating centroids
• Calculate the new centroid for each cluster as the mean of all the data points assigned to that cluster.
Repeating steps 2 and 3
• Repeat steps 2 and 3 until the centroids no longer change or a maximum number of iterations is reached.
Output
• The final k clusters and their respective centroids.

Figure 4: K-means clustering.

K-means clustering is an unsupervised machine learning algorithm used for partitioning a dataset into k clusters based on similarity in the data points. This algorithm works by iteratively assigning each data point to the closest centroid and then updating the centroid based on the new members in the cluster until convergence. This algorithm is widely used for clustering and has many practical applications, such as image segmentation, customer segmentation, and anomaly detection. However, it has some limitations, such as its sensitivity to initialization and the need to specify the number of clusters k in advance.

3.4 Supervised Learning

Supervised learning is a type of machine learning where an algorithm learns from labeled data, which includes both input data and their corresponding output labels. During training, the algorithm is given input-output pairs and learns to map input to output by adjusting its internal parameters. The goal of supervised learning is to make accurate predictions on new, unseen data by generalizing from the training data. Examples of supervised learning include classification and regression problems.

3.4.1 Convolution Neural Network with Long Short Term Memory

A Convolutional Neural Network uses convolution filters to the input to process sequential data while Long-short-term-memory loops over the sequential data and learn long-term dependencies between

Model: ”sequential”		
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 1, 64)	384
max_pooling1d (MaxPooling1D)	(None, 1, 64)	0
dropout (Dropout)	(None, 1, 64)	0
lstm (LSTM)	(None, 1, 32)	12416
dropout_1 (Dropout)	(None, 1, 32)	0
lstm_1 (LSTM)	(None, 16)	3136
dropout_2 (Dropout)	(None, 16)	0
dense (Dense)	(None, 1)	17
Total params	15953	
Trainable params	15953	
Non-trainable params	0	

Table 1: CNN-LSTM model summary

time-steps. A CNN-LSTM model uses both convolution and LSTM layer to learn from training data. In the proposed architecture 1, the first layer is convolutional layer which takes in one dimensional data and performs feature extraction to extract important features from the input data. This is followed by a maxpooling layer which reduces the dimensionality of the output from the previous layer to reduce overfitting. Dropout layer takes output from maxpooling layer and randomly drops out the percentage of neurons during the model training to prevent the model from overfitting. This is followed by the LSTM layer, which processes a list of extracted features from previous layers taking account of temporal dependencies between them. This is followed by dropout layer to prevent overfitting. There is another LSTM layer followed by dropout layer to process the output of previous LSTM layer. Dense layer is the final layer which is responsible for processing output from previous layer and generating a single output value to perform classification or regression tasks.

3.5 Keyframe extraction

Keyframe extraction is the process of selecting representative frames from a video. For unsupervised model, the frames closest to selected centroids can be selected as keyframes. The distance between each frame and the centroid can be calculated based on features extracted from the frames, such as colour or motion. Finally, the keyframes can be returned as a list of indices pointing to the selected frames. For supervised leaning, If the output of the frame meets a certain threshold, then those frames are considered as keyframes.

3.6 Creating keyshots using keyframes

The keyframes extracted from either k-means clustering or CNN-LSTM model are then used to create keyshots. The neighbourign frames of each keyframe are selected based on either temporal proximity or visual similarity to form a shot. Overall, the process of choosing keyframes and grouping them into key shots is an important step in developing a meaningful video summary that brings to life the essence of the original video.

3.7 Summary generation

All the selected Keyshots from the previous step are sorted temporally and stitched together to generate a summary video. K-means clustering algortithm sorts the keyshots based on the timestamps. The keyframe closer to the centroid in each cluster is taken as representative frame and these representative frames are sorted based on their timestamps to generate temporal sequence of keyshots. While CNN-LSTM also considers timestamps to sort keyshots, it also considers other factors such as motion, visual saliency etc.

4 Evaluation

The summaries generated by both k-means clustering algorithm and CNN-LSTM model are compared to user generated summaries to determine how well the models were able to capture the relevant and significant parts of the video.

4.1 K-means Clustering

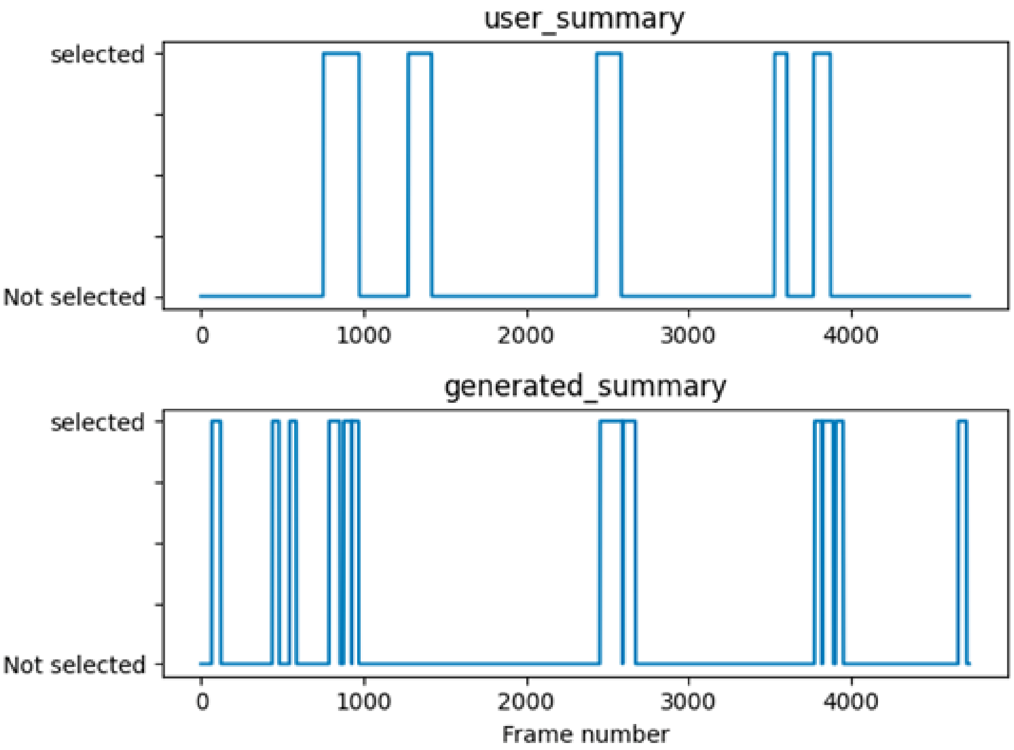


Figure 5: K-means clustering vs user generated summary.

Figure 5 shows the comparison of the unsupervised model with one of the user summaries which is selected from SumMe dataset.

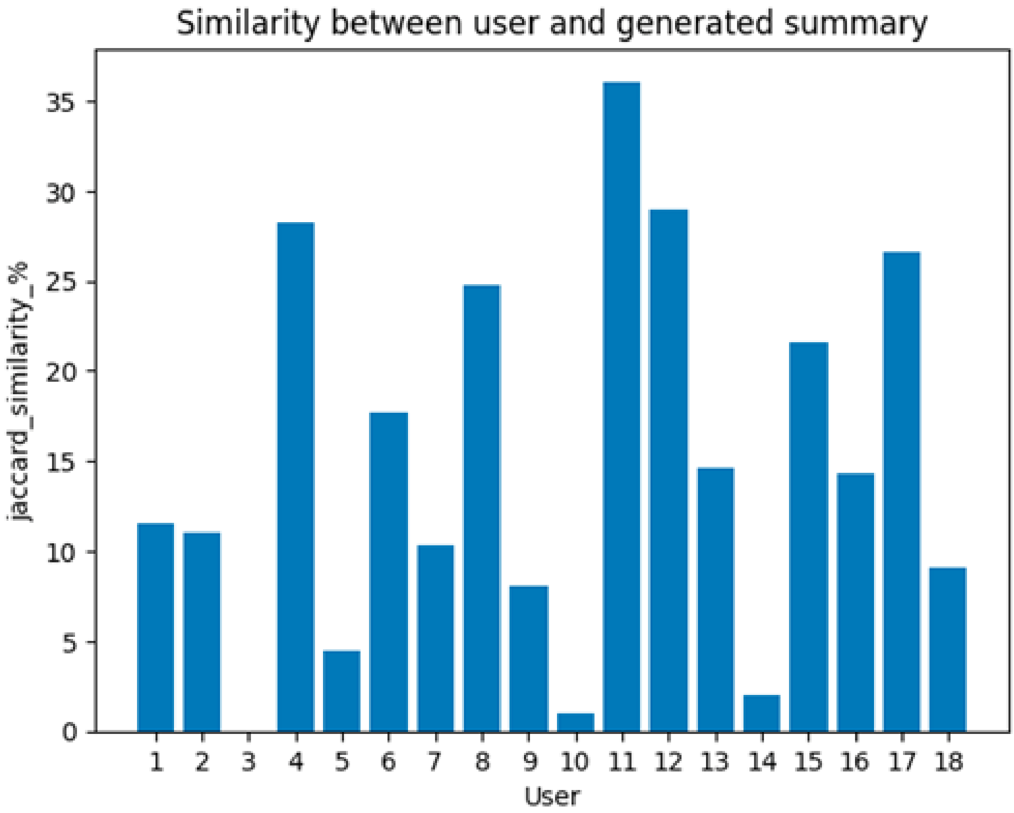


Figure 6: Similary in K-means clustering vs user generated summary.

Figure 6 is the summary generated by unsupervised model is compared to the user data given in SumMe dataset using Jaccard similarity score in percentage. The highest similarity is about 35%.

4.2 CNN-Long Short Term Memory

The model is evaluated in terms of precision(1), recall(2) and F1-score(3) 2. Precision is the proportion of true positives (correctly identified diseased plants) among all positive predictions (total predicted diseased plants). High precision indicates that the model has low false positive rates. Recall (or sensitivity) is the proportion of true positives among all actual positive cases (total number of diseased plants). High recall indicates that the model has low false negative rates. F1-score is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance, taking both precision and recall into account. These metrics might not be useful in all the use cases because sometimes the best performing models might not provide summarizes that could match the preferences of all the users. In such cases, it is useful and ideal to consider other metrics as well.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1\ Score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{3}$$

TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

Weighted metrics	
Precision	0.79
Recall	0.85
f1-score	0.80

Table 2: Performance metrics

Figure 7 shows the comparison of the supervised model with one of the user summaries which is selected from SumMe dataset. There is a lot of discontinuation of shots observed which can lead to jittery video, a longer summary length minimizes this effect.

Figure 8 is the summary generated by unsupervised model is compared to the user data given in SumMe dataset using Jaccard similarity score in percentage. The highest similarity is about 22%. Which is not high as unsupervised learning model.

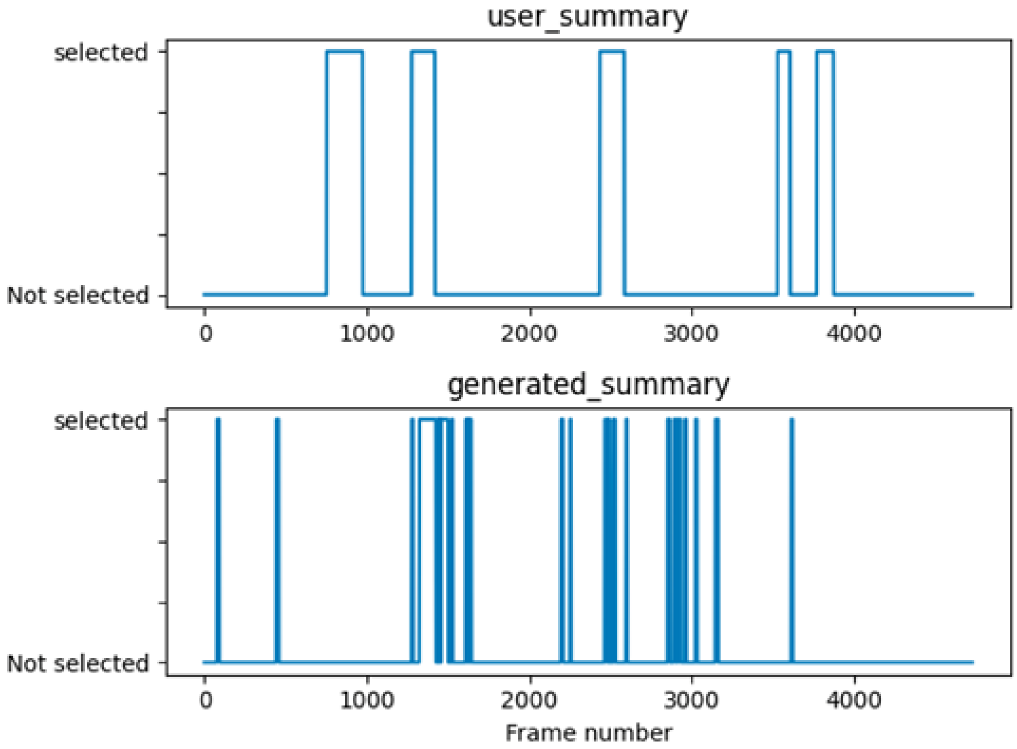


Figure 7: LSTM vs User generated summary.

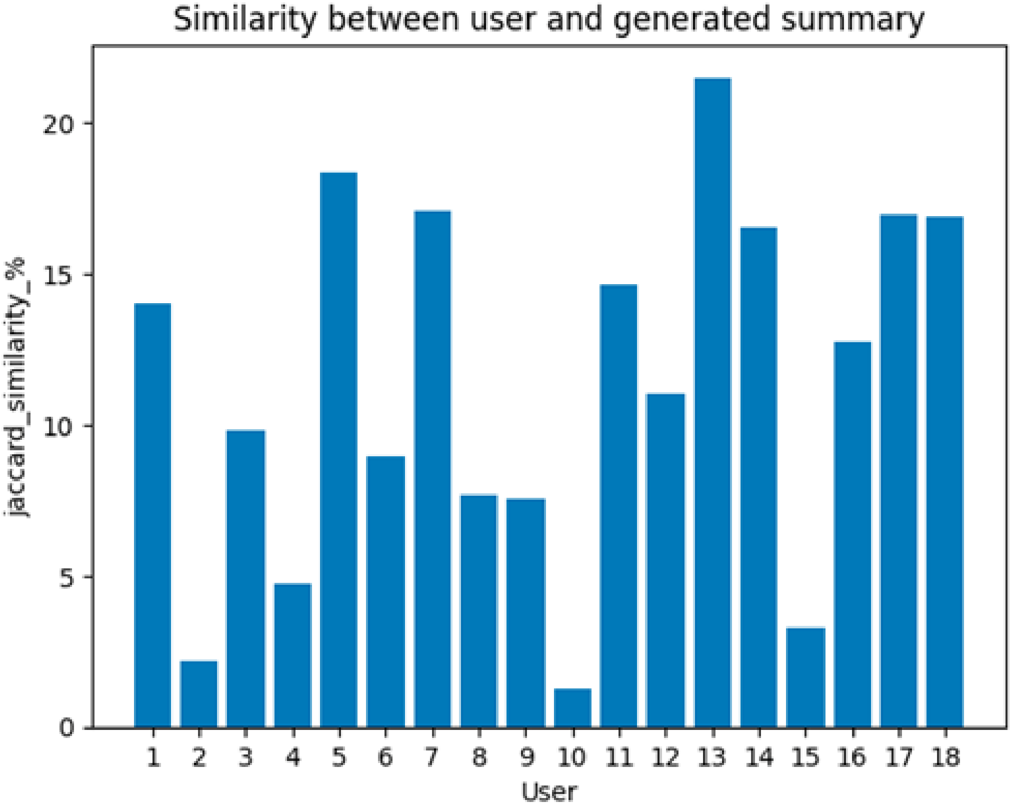


Figure 8: Similarity of LSTM vs user generated summary.

4.3 Comparison of supervised vs unsupervised learning models

Both k-means clustering and CNN-LSTM are capable of generating summarized videos but the approach used to summarize a video differs. K-means clustering groups visually similar frames into clusters and the significant frames from each cluster are selected to generate a summary. Whereas CNN-LSTM model is trained on labeled data in which importance of each frame is predicted and score is assigned for each frame based on it. Both algorithms have their own advantages and limitations. K-means clustering is fast and simple but it doesn't analyze the complex features in a frame like CNN-LSTM. The choice of algorithm ultimately depends on the requirements of the task.

Also, if given the same video twice to a k-means clustering model, it will generate two different summarized videos because it is a random initialization algorithm and the inital centroids are selected randomly. The difference between these two videos will be relatively small. But, CNN-LSTM will generate the same video both the times since it is a deterministic model.
Figure9 shows the comparison of summary generated by unsupervised model and supervised model.

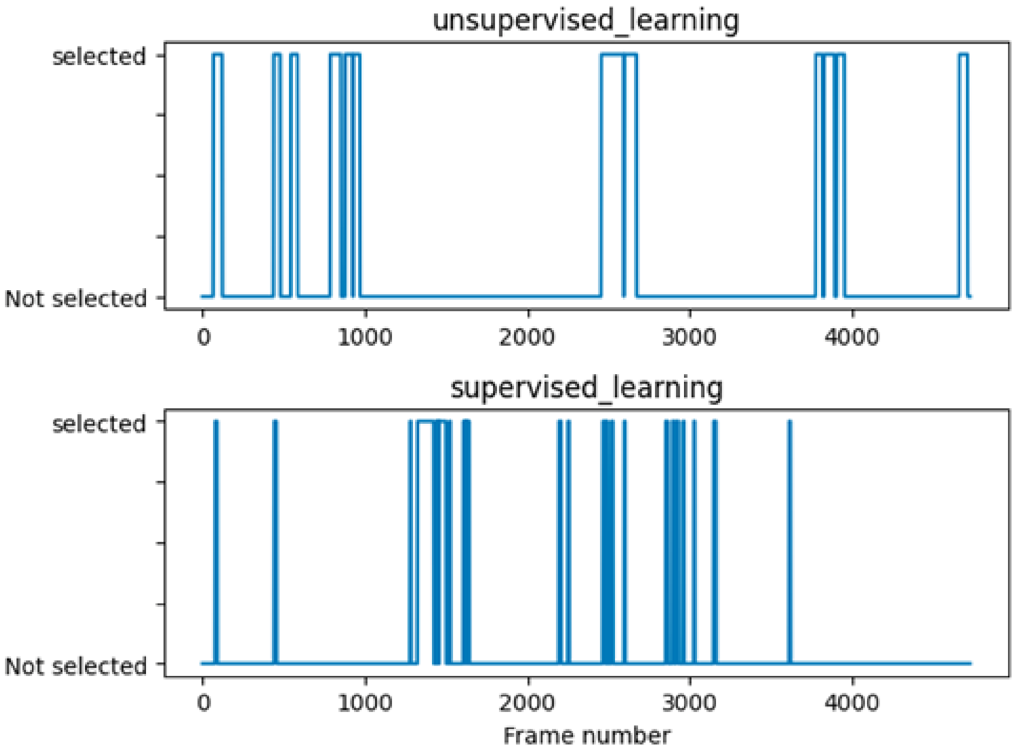


Figure 9: Comparison of Supervised vs Unsupervised learning models.

5 Conclusion

The proposed work describes the methodology for generating video summaries using both supervised and unsupervised learning techniques. The SumMe dataset was used for training and evaluating the summarization models. The video shots were analyzed for their color and motion features, and the feature vectors were collected to create shot feature vectors. K-means clustering was used for unsupervised learning to partition the data into clusters based on similarity in the data points. Supervised learning was performed using a Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) layers. The proposed architecture of the CNN-LSTM model included convolutional and maxpooling layers, dropout layers to prevent overfitting, LSTM layers to process a list of extracted features, and a dense layer to generate a single output value for classification or regression tasks. The performance of the models was evaluated using the F1 score on the SumMe dataset. it can be concluded that the unsupervised model outperformed the supervised model in generating summaries for the SumMe dataset. The Jaccard similarity score, which measures the overlap between the generated summary and the user data, was significantly higher for the unsupervised model, with a maximum similarity score of 35%, compared to 22% for the supervised model.
The future work would be to take feedback from the user on the summaries generated by the proposed models. Also, the proposed work could be extended by including video-to-text summarization.

References

[1] S. Foley, "Youtube might be worth over $100 billion," *The Street*, April 24 2018.

[2] J. Gibson, "You now have a shorter attention span than a goldfish," *Time*, May 14 2015.

[3] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pp. 505–520, Springer, 2014.

[4] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Video summarization using deep semantic features," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, pp. 361–377, Springer, 2017.

[5] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pp. 766–782, Springer, 2016.

[6] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[7] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Video summarization using deep neural networks: A survey," *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1838–1863, 2021.

[8] 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*, pp. 202–211, 2017.