Jotter: An Approach to Summarize the Formal Online Meeting

1st Sumedh S Bhat
Computer Science Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, Karnataka, India
sumedhbhat01@gmail.com

2nd Uzair Ahmed Nawaz
Computer Science Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, Karnataka, India
uzairahmed.blr@gmail.com

3rd Sujay M
Computer Science Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, Karnataka, India
sujaym.10@gmail.com

4th Nameesha Tantri
Computer Science Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, Karnataka, India
nameesha.tantri@gmail.com

5th Vani Vasudevan
Computer Science Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, Karnataka, India
vani.v@nmit.ac.in

2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIE) | 979-8-3503-1646-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/AIKIE60097.2023.10390455

***Abstract*—This paper examines various meeting summarising techniques. It also presents a hybridised technique for meeting summarization that combines abstractive and extractive techniques. Text extraction from meetings is accomplished in two ways: audio during the meeting and text recognition from the screen. These extracted texts are then compiled to create a meeting summary. In contrast to extractive text summarising, which concatenates essential sentences from paragraphs, abstractive text summarization focuses on providing a coherent summary of the given content. The majority of recent progress on abstractive summarization has been using RNN or Recurrent Neural Networks, although RNN-based algorithms fail when dealing with long sequences. This paper proposes hybridizing these two methods sequentially for meeting summarization to create a better overall summary of the meeting. This research endeavour seeks to convert conference video and audio into text documents and then run our hybridised text summarizer model on them. The result is a summarised text document that the various stakeholders, whether present at the conference or not, could utilise for rapid and concise reference. It will help the people who are not been able to attend the meeting can get the main points discussed in the meeting through summary.**

Keywords—Text Summarization, Deep Learning, Extractive Technique, Abstractive Technique, Artificial Intelligence, Object Detection, Machine Learning

I. Introduction

In the beginning, the text is extracted from the frames using natural language processing. The audio from the meeting is recorded, and a speech recognition model is applied to convert it to text. Following that, the summarizer is given the text that was generated from these sources so that it can summarise the text.

Extractive summarization generates a summary by means of ranking the sentences. Although, the effectiveness of extractive summarization is highly dependent on the sentence feature's quality. Whereas, most of the other algorithms demanded that the feature extraction to represent sentences be crafted by hand. In recent years, combining the earlier methods of deep learning has grown more prevalent. Embedding techniques for pre-trained words have been accomplished. Outstanding results in a variety of NLP tasks.

Extractive Summarization using BERT (Bidirectional Encoder Representations from Transformers) is done by tokenizing the paragraphs into sentences for the BERT model and is clustered by using the K Means algorithm to select only the sentences that are close to the centroid that achieves more accurate results. BERT for Text Embedding uses the BERT library, which is provided by the "hugging face" organisation.

We use an algorithm called TFRSP, also known as Text Frequency Ranking Sentence Prediction, that employs both extractive and abstractive summarization techniques. The Term Frequency-Inverse Document Frequency algorithm is combined with the Text Ranking algorithm in the process to generate an extractive summary. The sequence-to-sequence model is utilised in abstractive summarization which is a supervised algorithm with training and test datasets used to generate an abstractive summary.

II. Literary Review

We intend to implement three major modules in our application. The extraction of frames, speech recognition, and text summarization. Using Python's MoviePy library, We are able to take frames out of videos. Even the number of frames per second we can extract can be specified. Speech can be recorded and converted to text in a variety of ways. Deep learning, SVM or Support Vector Machine, and Minimum Distance Classifiers are some of the techniques available. Each method had benefits and drawbacks. It is best to utilize a deep learning technique that aids in generalization and offers a more accurate recognition model due to the significant quantity of noise that is generated while recording the meeting. This paper proposes using Google's speech recognition via a python library SpeechRecognition and PyAudio, PortAudio for speech recording. It is the second most accurate speech recognition software, with an accuracy of 79%.

We have discovered several ways to summarise the text. We discovered and tested several algorithms, including BERT, TSRFP, Sentence Ranking, and KNN. The sentence ranking algorithm generated summaries that were computationally cheaper but did not look as close to a human written summary. The BERT and KNN required greater resources but generated more humane summaries. Using the TSRFP model, we attempted to solve the problems. The summary is produced by using extractive summarization as an input for abstractive text summarization. This reduced the computational power required while still producing a summary that is more akin to a human-written summary.

III. Proposed Methods

By gathering information from the presenter's speech and the presentation and using speech recognition and handwriting text recognition, respectively, we propose to summarize the meeting. After being tokenized and filtered for relevant sentences, the collected data is sent into an extractive text summarizer. This information is subsequently given to the abstractive text summarization model, which creates a considerably shorter and more precise summary.

There are various implementation models for every module of the suggested task. However, our initial suggestion is a mixed system. The issue will be choosing the ideal model and improving it further. Everyone's perspective on how to summarize a meeting varies greatly. The models that have been implemented are separate from one another and have not yet been brought together to form a complete application.

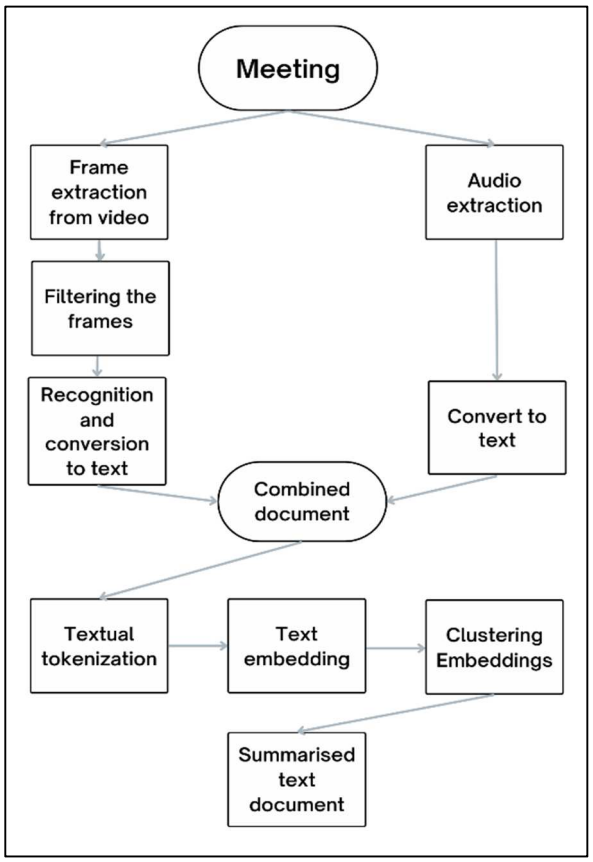

Fig. 1. Architectural diagram

Furthermore, as far as we are aware, no other programs have succeeded in providing a real-time meeting report. In order to create a web extension for lecture summaries, we want to create a web application. With the help of this study, we can distill an hour-long meeting into a single document that is simple to use and offers a succinct summary. Both business and education could make use of this application. The proposed system's architecture is shown in Figure 1. The investigation of numerous libraries, such as the hugging face transformers, MoviePy, SpeechRecognition, PyAudio, and PortAudio, provided us with a thorough understanding of the libraries available on the market.

A. Speech Recognition

1) Deep Learning: Deep learning is one of the most used speech recognition techniques. It is utilised by voice recognition software such as Google's. Deep learning predicts speech using an artificial neural network. [1] This strategy requires extensive training data to be effective. The audio data must initially be separated into practise data (70%) and testing data (30%). Using the Mel-frequency cepstral coefficient, the audio input is transformed into a vector, and subsequently, speech is identified using deep learning (DL). This model produced an accuracy of 66.22%.

2) Bidirectional Kalman Filter: For accurate results, speech recognition requires clear audio. Since the audio may contain sounds like noise, we are unable to provide proper and accurate audio as input during real-time conditions. These disturbances will produce inaccurate outcomes. The Kalman Filter could be used to remove these disruptions. Reduced non-stationary background noise can be achieved by using the Kalman filtering approach. Training and testing databases are first created from the database. The implementation of MFCC is for feature extraction. 90% accuracy was obtained during the testing, which is done in a noisy environment.

3) Support Vector Machine and Minimum Distance Classifier: The language of choice for speech recognition is typically English. People who don't speak English or are illiterate can use these technologies if speech recognition can be done in their native tongues. The system's goal is to create and put into use a conversion engine for several Indian languages. To categorize speech, the proposed system employs the Minimum Distance Classifier, Support Vector Machine (SVM), and MFCC feature extraction methods. The audio database is first constructed, followed by training and test databases. A feature vector is created once several features from the training dataset have been found during training. After features are collected using MFCC, a reference vector is created, and the testing phase is completed, the words with the best match to the given speech are taken into account. The precision [3] is calculated to be 93.625%.

4) Evaluating various speech recognition methods: Deep learning, machine learning, and statistical methods are all used in speech recognition. Each method has advantages and disadvantages. While susceptible to noise, feature extraction methods like Principal Component Analysis and Independent Component Analysis are useful for speech recognition systems with a limited vocabulary. RASTA or RASTA-PLP can provide dependable performance for noisy data. [4] Even while KNN and Naive Bayes classifiers are simple, they only work well with limited vocabulary sets. SVM performs better with medium-sized data, while training takes longer with large datasets [3]. Large datasets necessitate a drawn-out training period for neural networks [1].

B. Handwritten Text Recognition

1) Artificial Intelligence: Handwritten recognition is one of the most challenging pattern recognition tasks. It is used in a variety of settings, including courtrooms and postal services for postal code recognition. First, a test photograph with white backdrop and black text is taken. A genetic

algorithm is used to analyze the features of the extracted handwritten text, and the extraction method is utilized to extract the density of each text's pixels. Around 90% of the time, the system is thought to be accurate.

2) Deep Neural Networks((DNN): The accuracy and speed of handwriting detection have increased because to the development of deep neural networks. People can now rely on features and algorithms that were obtained from data instead of ones that were manually created[6]. All previous methods rely on a different system where the image is initially divided, the extracted image's angle, curve, and height are fixed, and there are constraints on the text's size and orientation. However, people's handwriting varies widely and is frequently extremely challenging to read, therefore these methods are either incorrect or ineffectual. A neural network, which is trained from data and is easily adaptable to new datasets, has all of these implicit processes and is unrestricted by any constraints. The structure is seen in Figure 2. With a 7.6% error rate compared to other cloud-based API is 14.6%, this algorithm [6] is trained at 145 DPI and outperforms all other models.

Figure 2 shows the Model Schematics. On the left, there is a CNN Encoder and Transformer Decoder. On the right, there is the Transformer Layer with an optionally Localised Self Attention.

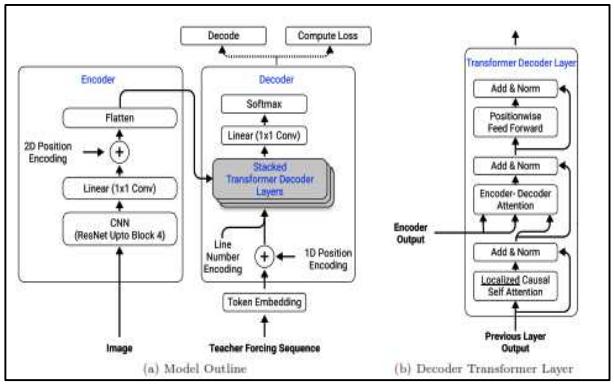

Fig. 2. DNN Model Schematics

C. Text Summarization

1) Extractive Text Summarization using Sentence Ranking: The most often occurring terms are those that are most pertinent, according to the primary idea underpinning extractive text summary. The technique first creates a frequency matrix, which is a list outlining the frequency with which each word occurs, and then it filters out the English words that are used the most. To determine a phrase's prominent context, the ratings of each word in the phrase are taken into account. The summarizer will extract the sentence fragments [5] with the highest weighted frequency and present a summary.

2) Abstractive Text Summarization using BERT Model: BERT and K-Means are also used by extractive text summarization. The paragraphs are tokenized into sentences using the BERT model using a pipeline. Clustering is embedded using K-means. The last step is to choose the sentences closest to the centroid. Uses the PyTorch-pre-trained-BERT package for the BERT for Text Embedding algorithm. Clustering is accomplished with the help of K-Means and Gaussian Mixture Models. A very modest number of phrases are required so that the model can successfully capture the context of the entire lecture. The model [20] included additional sentences which were beyond what was requested.

3) K Nearest Neighbours (KNN): The degree of feature values and feature similarity are taken into account in a modified version of the KNN algorithm. After selecting the K distinct words that are similar as the K nearest neighbours, the label for the beginner entity is decided by voting on the labels of the K closest neighbours. A binary category has been assigned to the text summation. A string of text that has been broken up into paragraphs using carriage returns makes up the input. [9].

4) TFRSP: This algorithm combines extractive and abstractive summarization techniques. In the first step, the Term Frequency-Inverse Document Frequency algorithm is coupled with the Text Ranking algorithm. The output of the is given as an input to the abstractive algorithm. The summary is generated using abstractive text summarization. The sequence-to-sequence model, a supervised learning technique with training and testing datasets, is used for abstractive text summarization. The summary increases the ROUGE score of the existing methods by 38.42 per cent.

Currently, there exists a way to summarize the video by using the MSR-VTT dataset which consists of over 10,000 YouTube clips along with their summaries but to use this model we need to process the entire video lecture at once which is inefficient and time-consuming. This method is a more improved implementation by using screenshots and audio from the host instead of the entire video. This allows us to the pipeline by using threads which reduces the time consumed while processing the video. Since this is used mainly for live meetings this reduces the time for the generation of the report.

Our application can record audio, video, and text from it. It can also translate speech to text and provide a summary. There is no market-available software that can offer all these features and effectively summarize the meeting. Therefore, there is room for potential future expansion and improvement of our research.

IV. Results

The proposed application focuses on three different modules:

A. Key-Frame Extraction

The user launches the summarizing program, which extracts the keyframes (Figure 3) from the movie. A comparison algorithm is run through all the frames sequentially to check the similarity of the frames. If two frames are within the threshold, then the latest frame will be made the keyframe and the previous frame will be discarded. This helps reduce the redundancy of data and provides a more accurate measure of the words that have been utilized in the meeting. This method is selected assuming the host erases the canvas once complete and can give a more definitive measure of the different sections in the meeting. Figure 4 shows the key frames extracted from the chosen video using Convolutional Neural Networks by comparing the extracted images to select only the frames that are distinct from the preceding frames that have been extracted.

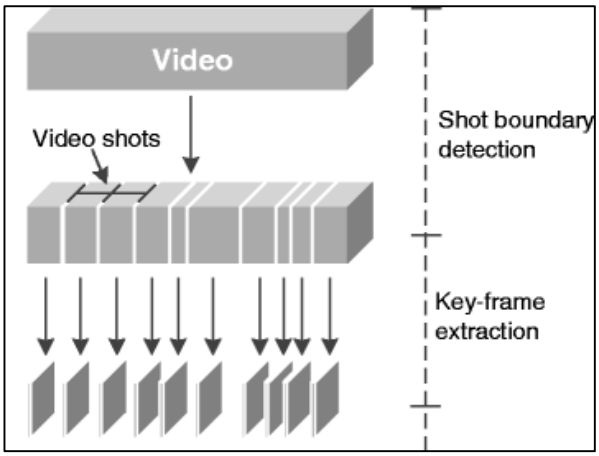

Fig. 3. Diagrammatic representation of a key frame extraction from video.

Output:

Video Description: The input video consists of a college lecture with slides about the topic.

Video Duration: 4:21

Frames Extracted: 24

Fig. 4. Frames extracted from the video

B. *Speech Recognition(SR)*

The application breaks up the audio into distinct segments after identifying the host's voice in it. These segments are produced using the time frames that were recorded during the frame extractor's production of unique frames. After being recorded, the audio is saved in the.wav format. The speech from the wav file is recognized using the Google Speech API and grouped with the relevant text created from the keyframe. The workflow for converting speech to text is shown in Figure 5, and the code and output for the SR module are shown in Figure 6.

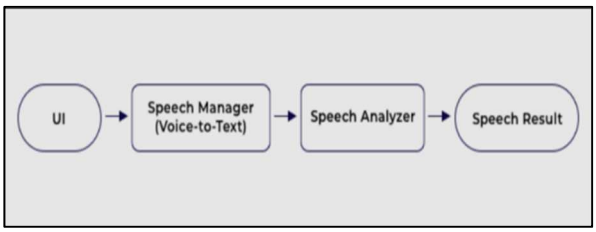

Fig. 5. Workflow of speech-to-text conversion

```
pip install SpeechRecognition pydub

Requirement already satisfied: SpeechRecognition in /home/sumedh/anaconda3/lib/python3.9/site-packages (3.8.1)
Requirement already satisfied: pydub in /home/sumedh/anaconda3/lib/python3.9/site-packages (0.25.1)
Note: you may need to restart the kernel to use updated packages.

import speech_recognition as sr

filename = "MLtrainingWav.wav"

r = sr.Recognizer()

with sr.AudioFile(filename) as source:
    audio_data = r.record(source)
    text = r.recognize_google(audio_data)
    print(text)

trial speech recognition for machine learning algorithm
```

Fig. 6. The SR module

C. *Text Summarization*

The text generated from the above modules is passed to the text summarizer which summarizes the context and returns a summarized text document. Figure 7 shows the workflow of text summarization and Table I presents the original text and summarized words with % of reduction achieved.

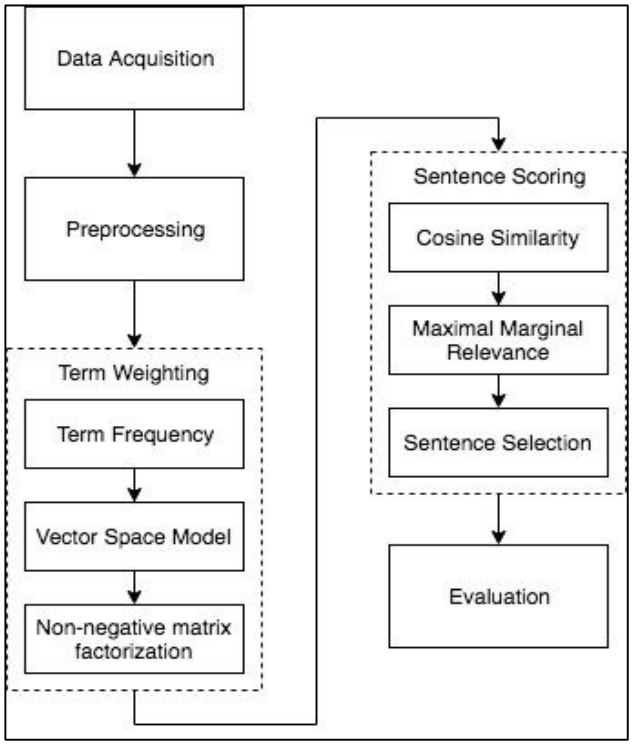

Fig. 7. Workflow of Text Summarization

TABLE I. SAMPLE INPUT TEXT AND ITS SUMMARY WITH REDUCTION PERCENTAGE

Sample Number	No. of words	No. of words after summarization	Reduction Percentage
1	183	86	53.01 %
2	155	83	46.45 %
3	92	52	43.47 %
4	149	63	57.72 %
5	210	89	57.62 %

After completely integrating all the modules together the application was run on a few small lecture videos to determine the time taken to generate the summary as depicted in Table II.

TABLE II. VIDEO SUMMARIZER RESULTS

Sample Number	Video Length	Total Time Taken by the Module
1	15 mins	325 secs
2	30 mins	574 secs

To get these values, the modules are run independently of the video but when taking into consideration that a lot of these operations are happening simultaneously as the lecture is being played, the time that a user must wait to get the summary is almost negligible.

V. Conclusion

The numerous techniques for summarizing the sessions are presented in-depth and completely in this paper. Text summarization that is abstractive and extractive is used in the suggested system. Different stakeholders who were unable to attend the meeting or who simply want to know the main topics of the meeting may utilize this summary.

References

[1] Kongthon, Alisa; Sangkeettrakarn, Chatchawal; Kongyoung, Sarawoot; Haruechaiyasak, Choochart (October 27–30, 2009). Implementing an online help desk system based on conversational agents. MEDES '09: The International Conference on Management of Emergent Digital EcoSystems. France: ACM. doi:10.1145/1643823.1643908.

[2] Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.

[3] A. P. Singh, R. Nath and S. Kumar, "A Survey: Speech Recognition Approaches and Techniques," 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2018, pp. 1-4, doi:10.1109/UPCON.2018.8596954.

[4] N. Chumuang and M. Ketcham, "Model for Handwritten Recognition Based on Artificial Intelligence," 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), 2018, pp. 1-5, doi: 10.1109/iSAI-NLP.2018.8692958.

[5] J. N. Madhuri and R. Ganesh Kumar, "Extractive Text Summarization Using Sentence Ranking," 2019 International Conference on Data Science and Communication (IconDSC), 2019, pp. 1-3, doi: 10.1109/IconDSC.2019.8817040.

[6] S. S. Desai, D. Rajput and K. Patil, "An approach for Text Recognition from Document Images," 2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC), 2020, pp. 1-5, doi: 10.1109/B-HTC50970.2020.9297939.

[7] Du, Y. et al. (2020) “PP-OCR: A practical ultra-lightweight OCR system,” arXiv [cs.CV]. doi: 10.48550/ARXIV.2009.09941.

[8] Ghadage, Y. H. and Shelke, S. D. (2016) “Speech to text conversion for multilingual languages,” in 2016 International Conference on Communication and Signal Processing (ICCSP). IEEE, pp. 0236–0240.

[9] Jo, T. (2017) “K nearest neighbor for text summarization using feature similarity,” in 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE). IEEE, pp. 1–5.

[10] Jolad, B. and Khanai, R. (2019) “An Art of Speech Recognition: A Review,” in 2019 2nd International Conference on Signal Processing and Communication (ICSPC). IEEE, pp. 31–35.

[11] Lakkhanawannakun, P. and Noyunsan, C. (2019) “Speech Recognition using Deep Learning,” in 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC). IEEE, pp. 1–4.

[12] Ozsoy, M. G., Alpaslan, F. N. and Cicekli, I. (2011) “Text summarization using Latent Semantic Analysis,” Journal of information science, 37(4), pp. 405–417. doi: 10.1177/0165551511408848.

[13] Raundale, P. and Shekhar, H. (2021) “Analytical study of Text Summarization Techniques,” in 2021 Asian Conference on Innovation in Technology (ASIANCON). IEEE, pp. 1–4.

[14] Sharma, N. and Sardana, S. (2016) “A real time speech to text conversion system using bidirectional Kalman filter in Matlab,” in 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, pp. 2353–2357.

[15] Singh, S. S. and Karayev, S. (2021) “Full page handwriting recognition via image to sequence extraction,” in Document Analysis and Recognition – ICDAR 2021. Cham: Springer International Publishing, pp. 55–69.

[16] Sinha, A., Yadav, A. and Gahlot, A. (2018) “Extractive Text Summarization using Neural Networks,” arXiv [cs.CL]. doi: 10.48550/ARXIV.1802.10137.

[17] Zhang, Y., Meng, J. E. and Pratama, M. (2016) “Extractive document summarization based on convolutional neural networks,” in IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society. IEEE, pp. 918–922.

[18] S M, Meena & M P, Ramkumar & R.E, Asmitha & Selvan, Emil. (2020). Text Summarization Using Text Frequency Ranking Sentence Prediction. 1-5. 10.1109/ICCCSP49186.2020.9315203. (Accessed: June 22, 2022).

[19] Dalal, V. and Malik, L. (2013) “A Survey of Extractive and Abstractive Text Summarization Techniques,” in 2013 6th International Conference on Emerging Trends in Engineering and Technology. IEEE, pp. 109–110.

[20] Miller, D. (2019) “Leveraging BERT for Extractive Text Summarization on Lectures,” arXiv [cs.CL]. doi: 10.48550/ARXIV.1906.04165

[21] Y. Tonomura, A. Akutsu, K. Otsuji and T. sadakata, "Videomap and videospaceicon: T ools for anatomizing video content", Proc. ACM INTERCHI'93, 1993.

[22] H. Ueda, T. Miyatake and S. Yoshizawa, "Impact: An interactiv enatural-motion-picture dedicated multimedia authoring system", Pr oc. ACM SIGCHI'91, 1991-Apr.

[23] A. Fermain and A. Tekalp, "Multiscale content extraction and representation for video indexing", Pr c. SPIE 3229 on Multimdia Storage and Archiving Systems II, 1997.

[24] D. DeMenthon, V. Kobla and D. Doermann, Video summarization by curv e simplification, 1998.

[25] M. Yeung, B. Yeo, W. Wolf and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences", Pr oc SPIE on Multimedia Computing and Networking, vol. 2417, 1995.

[26] Chowdhary CL, Reddy GT, Parameshachari BD. Computer Vision and Recognition Systems: Research Innovations and Trends. CRC Press; 2022 Mar 9.

[27] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, "Indexing b y laten t semantic analysis", Journal of the American Society for Information Science, vol. 41, pp. 391-407, 1990.

[28] W. Press, Numerical R ecip esin C: The A rt of Scientific Computing, England, Cambridge:Cambridge University Press, 1992.

[29] Nayak JP, Parameshachari BD, Soyjaudah KS, Banu R, Nuthan AC. Identification of PCB faults using image processing. In2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) 2017 Dec 15 (pp. 1-4). IEEE.

Contribution of the research work:

1) Effective Data Management: By reducing long films into shorter representations without sacrificing important information, video summarizing aids in the efficient management of huge video collections. This is especially important for applications with constrained bandwidth and storage capacity.

2) Improved Video Browsing and Retrieval: By giving consumers succinct summaries of video information, summarization techniques make it easier for users to browse and retrieve videos efficiently. This makes it easier for viewers to find specific information within videos and rapidly select relevant ones.

3) Time-Efficient Analysis: By employing summarization approaches to extract salient features and critical moments from long videos, researchers can analyze them more quickly and with less work. This makes it possible for researchers to concentrate on important sections, which improves the effectiveness of their analysis.

Data Collection:

1) Public Datasets: Specifically selected datasets made available to the public are frequently used by researchers doing video summarization studies. These datasets, which offer annotated video footage that may be utilized for training and assessment, include the SumMe, TVSum, and YouTube-8M datasets.
2) Custom Video Collection: To build a custom dataset suited to their particular study needs, researchers are welcome to gather their own video data. In order to generate a complete dataset that accurately represents multiple video kinds, this may entail filming or getting films from a variety of sources, making sure that the content, duration, and genres are diverse.
3) Web scraping: A variety of internet sources, including social media, news websites, and video-sharing platforms, can be used to collect videos through web scraping techniques. Nonetheless, adhering to copyright and licensing laws is essential.