International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 3, June 2023, pp. 2752~2760
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i3.pp2752-2760

Multimodal video abstraction into a static document using deep learning

Muna Ghazi Abdulsahib, Matheel E. Abdulmunim
Department of Computer Sciences, University of Technology-Iraq, Baghdad, Iraq

Article Info

Article history:

Received Jul 12, 2022
Revised Aug 21, 2022
Accepted Aug 31, 2022

Keywords:

Deep learning
Keyframe extraction
Shot and scene boundary detection
Text summarization
Video summarization

ABSTRACT

Abstraction is a strategy that gives the essential points of a document in a short period of time. The video abstraction approach proposed in this research is based on multi-modal video data, which comprises both audio and visual data. Segmenting the input video into scenes and obtaining a textual and visual summary for each scene are the major video abstraction procedures to summarize the video events into a static document. To recognize the shot and scene boundary from a video sequence, a hybrid features method was employed, which improves detection shot performance by selecting strong and flexible features. The most informative keyframes from each scene are then incorporated into the visual summary. A hybrid deep learning model was used for abstractive text summarization. The BBC archive provided the testing videos, which comprised BBC Learning English and BBC News. In addition, a news summary dataset was used to train a deep model. The performance of the proposed approaches was assessed using metrics like Rouge for textual summary, which achieved a 40.49% accuracy rate. While precision, recall, and F-score used for visual summary have achieved (94.9%) accuracy, which performed better than the other methods, according to the findings of the experiments.

Corresponding Author:

Muna Ghazi Abdulsahib
Department of Computer Sciences, University of Technology-Iraq
Baghdad, Iraq
Email: MUNA.G.Abdulsahib@uotechnology.edu.iq

1. INTRODUCTION

Video is defined as a series of frames made up of many forms of data, such as text, images, audio, and metadata. Education, surveillance, entertainment, medicine, and sports are all areas where video is frequently used. As a result, a large number of digital recordings are now available on websites, social media platforms, and public video archives. The video took a long time, required a large quantity of storage, and used a lot of bandwidth when transmitted via the internet. The user may not be interested in the complete video's content, or the user may not have enough time to watch the entire video. People may only want to see the abstraction of the video rather than the entire video [1], [2]. Abstraction is a process that delivers a fast overview of the most important or relevant information within the original content. As a result, it saves users time, reduces storage, increases transmission speed across the web, and improves access efficiency. Due to its sequence-to-sequence nature, summarizing video, audio, pictures, and text is a difficult task [3], [4]. Deep learning methods have recently proven to be useful in a variety of domains, including image classification, summarization, machine translation, discourse identification, and text-to-speech production [5]–[8].

The process of video summarization is important for giving an abstract view of a long video. It represents the most relevant information about the various occurrences in the film in the shortest amount of time. Digital videos are typically made up of a variety of media, including audio, images, and text. Static (keyframes) and dynamic (video skims) summary are the two methods of video summarization. A static video summary represents a summary of a film's major contents by extracting the most informative keyframes. Dynamic video produces short video segments from original videos based on both their visual and audio content. Some summarization algorithms rely solely on visual data, while others combine visual and textual data. As a result, dynamic summaries can include all three types of visual, textual, and audio data. Artificial Intelligence approaches will be used to generate summaries in the same way that humans would [9]–[11].

A text summary is a technique for reducing a large text to a short text by deleting the less relevant material. While keeping the primary information that allows users to quickly identify the most interesting information [7], [12], [13]. There are two types of text summaries: the first is extractive summaries. Which is summarizing works by making a copy of a few key sentences from the source material. It uses statistical analysis to select words, sentences, and paragraphs from the original material based on their importance and then concatenates these key elements of the document to generate a summary [14]–[16]. The second is abstractive summaries. Humans summarize documents in a similar fashion to abstractive text summarization. Abstractive text summarization begins by comprehending the document before paraphrasing, which includes new words, phrases, and rephrasing. It employs a sophisticated heuristic algorithm. It compresses data well and reduces redundancy. Abstract summarizing is more difficult to achieve than extractive text summarization [7], [15], [16].

Scenes, shots, and frames make up the structure of a video. A video scene is made up of a series of interconnected shots captured from various camera settings. A video shot is a collection of interconnected frames captured by a single camera action. The smallest unit in a video is a frame, which displays a single image. The scene types are sequential scene, concurrent scene, and hybrid scene. A sequential scene is one in which an event occurs in a series of consecutive places, the contents of which are related and visually comparable. A concurrent scene is made up of shots in which two or more people are conversing. Multiple sequential and concurrent shot groupings create a hybrid scene. Scene segmentation is important in many applications, including summarization, indexing, and retrieval. It begins by partitioning the video into distinct shots, then extracting shot features. Comparable shots are joined to generate scenes. The three basic categories of scene boundary detection are visual, audio-based, and hybrid scene segmentation [9], [17]–[21].

A keyframe is also known as a selective or representative frame. A keyframe is the frame in which the content of a video shot or a subshot is best represented. To select keyframes, features such as color histogram, shapes, and edges are used [22]–[25]. Many applications, such as video summarization, video retrieval, indexing, compression, searching, and action identification, require keyframe extraction. It reduces the amount of video data by deleting irrelevant data. Processing all frames necessitates a large amount of computation and memory. These keyframes should contain all of the visual information for the entire video. Due to the length of videos, keyframe extraction is more difficult, necessitating the use of efficient methods to detect keyframes. These methods can be categorized into four categories: sampling, shot boundary detection, clustering, and other keyframe extraction methods. The most straightforward technique is to use the first, middle, or last frame as a keyframe. However, one disadvantage of these approaches is that the recovered keyframes may not be capable of portraying the video material [26]–[34].

A recurrent neural network (RNN) is a supervised deep learning model. RNNs can deal with sequences of vectors in both the input and output. RNNs are capable of efficiently processing successions of different lengths of inputs and outputs. Each RNN layer has its own timestep, and the weights are distributed across time. Short-term memory causes vanishing gradients in RNNs, which occurs when training a long sequence by RNN. Gating RNN is a technique for solving the problem of disappearing gradients. Long short-term memory (LSTM) and gated recurrent unit (GRU) are two well-known gating techniques [7], [35]–[37]. The convolutional neural network (CNN) is a sort of multi-layer neural network. Feature extraction and classification are the two real aspects of CNN. The system will run a series of convolutions and pooling operations in the feature extraction section to separate the features. The pooling techniques will be employed to reduce the size of the input as well as the convolution results. The output layer, on the other hand, will be a network of fully connected layers that will serve as a classifier for the retrieved features [37]–[39].

Andra and Usagawa [40] have proposed summarizing the transcript of a lecture video by using attention with a recurrent neural network. The suggested method consists of the following steps: First, the preprocessing step. Then the transcript is segmented based on topic. The segments are classified by the topics based on similarity. The PowerSeg method is used to segment the text window by extracting multiple feature vectors. The cosine similarity is then used to compute the similarity between each segment. And finally, summarize the transcript using encoder and decoder recurrent neural networks, which contain three hidden layers of LSTM. However, the experimental results were evaluated using a rough measure across two datasets. The first dataset is a lecture video series from Stanford University. The second dataset is used for artificially generated lecture video fragmentation. These transcript videos are summarized manually and used as ground truth in the training model. Dilawari and Khan [41] have suggested video sequence summarization using deep learning. Which uses the VGG-16 CNN pretrained deep model to present textual descriptions of the visual content of video frames. These textual descriptions are then fed into the recurrent model of two bidirectional LSTM to present an abstractive textual summarization. Producing video events in text form can help users comprehend long videos in less time. The video description was evaluated using the Microsoft research video to text (MSRVTT) dataset. The text summarization was evaluated using a rough measure using the CNN/Daily Mail dataset. For their proposed video textual description and summarization, they produced the UET Surveillance dataset.

Agyeman *et al.* [42] have suggested summarizing soccer videos using deep learning. The video was first divided into segments and then applied a deep model to each segment. To extract features, a deep learning model employs a residual network (ResNet34)-based 3D-convolutional neural network (3D-CNN). After that, they applied the LSTM network to detect video soccer highlights. All the highlights from the video segments are then connected into a summary video. For evaluation, video summarization used the mean opinion score (MOS) measure. Two datasets were used: UCF101 and Soccer-5, which is a newly created annotated soccer dataset. Hussain *et al.* [43] have suggested summarizing multi-view video industrial surveillance using deep learning. The video was divided into shots according to target appearance, using the CNN model, and then stored with a timestamp in a novel lookup table. After that, it is sent to the cloud, which uses the CNN model to extract features. And fed into bidirectional long short-term memory to obtain probabilities of informativeness and non-informative frames. Finally, generating a summary from these has the highest probability of informative frames. The measures used for evaluation on the Office dataset are recall, precision, and F1-score. Emon *et al.* [44] have suggested cricket video summarization using deep learning. The video was divided into shots using Kernel temporal segmentation, which detects shot boundaries when they appear different in visual features. The deep network is used for prediction of the frame important score. They used a convolutional neural network of type ResNet152 and reinforcement learning with a reward function. Then compute the shot importance score from the average value of the frame scores. After that, they selected the shot with the highest score within a specific length of summary video. A new dataset created of cricket videos is called CricSum. For evaluation, they performed the F1-score as an objective measure as well as the mean opinion score as a subjective measure to compute the degree of human judgment.

The contribution of this paper is to abstract video into a static document based on both visual and audio media. This paper proposed a method for text summarization using a hybrid deep learning model. Proposed methods for detecting shot boundaries, scene boundaries, and keyframe extraction based on hybrid features, which provide robustness and improve detection performance by selecting the strongest and most adaptable characteristics.

2. RESEARCH METHOD

This section provides the proposed method with a general block diagram and algorithms, to perform video abstraction into a static document. The suggested video abstraction is based on multi-model video data, which includes both audio and visual data. The semantic information that is happening in the scenes is often directly reflected in the auditory information that distinguishes the activities. The main video abstraction steps are segmenting the input video into scenes and getting the textual summary of the audio and visual summary for each scene. Then present a static PDF document with visual and textual abstractions for each scene in sequential order.

This architecture is composed of two phases, which are visual summary and text summary. The first phase is visual summary, in which visual hybrid features are used to suggest methods for detecting scene boundaries (Scene BD), shot boundaries (SBD), and keyframe extraction. The text summary phase converts audio into textual representation using automatic speech recognition (ASR), then uses deep learning to conduct abstractive text summarization. The proposed method's general block diagram is shown in Figure 1. This section covers the details of the suggested method for each phase.

a. The first phase is visual summary, in which hybrid features including discrete wavelet transform (DWT) and grey level co-occurrence matrix (GLCM) are used to detect cut shot boundaries, keyframe extraction, and scene boundaries from video. The following are the details:

- The first step is preprocessing, which involves extracting frames from video. The frames are then resized to 256×256. This step has the benefit of cutting down on computation time.
- The next step is to extract the first feature, which is about texture features. This includes grayscale conversion of frames. The Haar wavelet function is then employed within the DWT to compute it. This feature allows for efficient, robust, and flexible feature selection. Then for each frame, extract the low low (LL) sub band from DWT, which contains the most significant characteristics data.
- After that, the second feature is extracted to represent the texture feature. This includes grayscale conversion and extraction of the GLCM function, which is a powerful feature. GLCM determines the

frequencies of a nearby connection between two pixels at angles of 0°, 45°, 90°, and 135°, as well as the distance 1 between them. The GLCM is then normalized, and the correlation is calculated.

- Then next comes the step of matching similarities. This stage compares the similarity of features in consecutive frames. The Euclidean distance is used to compute the similarity match for the first and second features. The average of the matching vectors of both features is then computed, yielding a vector of average matching.
- Next, it determines a local adaptive threshold that is superior to the global threshold. Using a global threshold for all frames is inefficient since video material changes rapidly, making it difficult to determine a global threshold that fits all frames. Local thresholds, on the other hand, vary depending on the content of video frames. As a result, in the proposed method, local thresholds are derived using the mean and standard deviation for an average matching vector for each window size of 250 frames. The formula (1) is used to calculate local thresholds:

$$Th = (mean + standard\ deviation) * c \tag{1}$$

where c is set to (3.7) and gained through trial and error until the best performance results are obtained.

After that, a comparison is made between the frames' average matching vector and a threshold value. The frame that corresponds to the index is regarded as a cut shot boundary detection if it is greater than the threshold value.

- This step is to extract the keyframe. A keyframe that represents the most important content is extracted from each shot. Because the LL sub band from DWT is already computed during the shot boundary process, keyframe selection criteria based on DWT are applied. For each shot, the frame with the highest DWT standard deviation is extracted as the keyframe. Because greater DWT indicates the amount of information included in a frame, it is possible to efficiently summarize the visual contents.

Finally, scene boundaries detection step is applied. After the segmentation of video into shots, the most representative keyframes are extracted. The scenes' boundaries are detected based on visual information. A sequential scene is one in which an event occurs in a number of locations, the contents of which are related and visually comparable. The mean square error metric is used to group shots with a high visual similarity into the same scene. After detection of the scene boundaries, the most representative keyframe in each scene is extracted, which represents the video's visual abstraction.

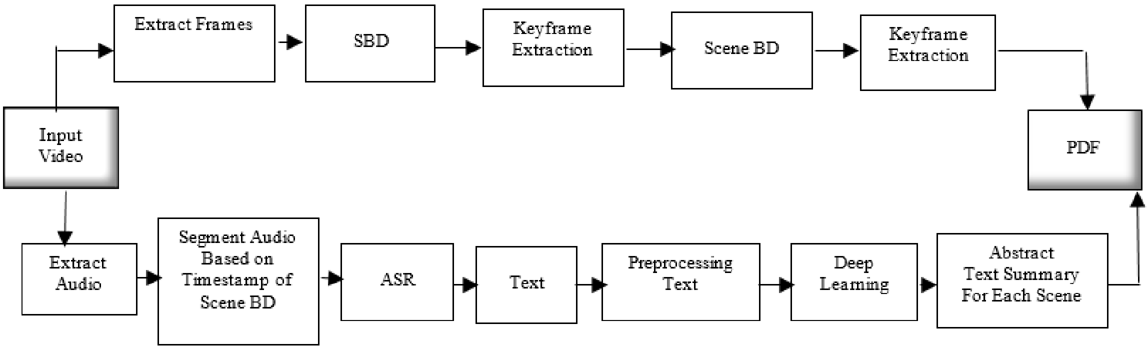

Figure 1. Block diagram of the proposed method

b. The second phase is text summarization, which employs convolutional recurrent neural network (CRNN) a hybrid deep learning approach that includes CNN and RNN for abstractive text summarization. The text was extracted by ASR from the video's audio. The following are the details:

- Extracting audio from video is the first step. Then split the audio into segments according to the timestamp of the scene boundary detection. Each audio segment is converted into textual representation using ASR.
- Next, after loading the dataset, the text preprocessing step will execute split words, lowercase, remove punctuations, remove stop words (stop words such as: "a," "the," and so on), and contractions. This phase improves the accuracy of the text summarization. Then divide the data into two sets: training and testing.
- After that, the encoder will use two one-dimensional convolutional neural network layers, the CNN part, followed by three long-short-term memory layers, the RNN part. This generates a vector representation that is used as the decoder's initial state.

- Then in the decoder, one LSTM layer will be employed, followed by an attention mechanism. The decoder can make use of intermediate hidden states in the encoder using this mechanism. Hence, the decoder can use all of this information to figure out which word will come next.
- Finally, the model's structure is designed to accommodate two stages: training and inference. In the training stage, it will compile and train the encoder and decoder models. Then, the inference stage will utilize those encoder and decoder models to predict the target summary.

Algorithm 1. Proposed video abstraction into a PDF static document.

```
Input: MP4 Video
Output: PDF
Process:
Step 1: Load a video.
Step 2: Visual summary:
Step 2.1: Extract frames from video.
Step 2.2: Resize the frames.
Step 2.3: Cut shot boundary detection:
Step 2.3.1: Convert frames to grayscale.
Step 2.3.2: Extract first feature: Compute DWT and extract LL.
Step 2.3.3: Extract second feature: Compute GLCM
Step 2.3.4: Compute the average of similarity matching (Hybirdmatch[i]) for hybrid
            features.
Step 2.3.5: Calculate local thresholds (LTH[j]) for every M frames.
Step 2.3.6: Compare if Hybirdmatch[i] > LTH[j], Then it considered as cut SBD.
Step 2.4: Extract a keyframe from each shot that has the highest DWT.
Step 2.5: Scene boundary detection:
Step 2.5.1: Calculate keyframe similarity for each shot, and compare it to the threshold to
            group them into the same scene.
Step 2.5.2: Extract a keyframe from each scene that has the highest DWT.
Step 3: Compute timestamp for each SceneBD.
Step 4: Text summary:
Step 4.1: Extract audio from video.
Step 4.2: Segment Audio based on timestamp of SceneBD.
Step 4.3: Convert each audio segment to text using ASR.
Step 4.4: Preprocessing text.
Step 4.5: Deep learning model:
Step 4.5.1: Encoder part: two layers CNN of a one-dimensional convolutional and three
            layers RNN of type LSTM.
Step 4.5.2: Decoder part: one-layer RNN of type LSTM.
Step 4.5.3: Perform attention mechanism for the output of encoder and decoder.
Step 4.5.4: Output dense layer for the output of decoder and attention.
Step 4.5.5: Compile and train the model.
Step 4.5.6: Save encoder and decoder model.
Step 4.5.7: Predict encoder and predict decoder model.
Step 4.5.8: Generate abstract text summary for each scene.
Step 5: Create a PDF document with visual and textual abstractions for each scene in
        sequential order.
Step 6: End.
```

3. RESULTS AND DISCUSSION

Experimental results of the tests were displayed in this section to demonstrate the performance of the proposed video abstraction method. It contains data on the videos that were used to assess the effectiveness of the suggested strategy, and present experimental results for both phases' visual and text summary. A comparison to a previous method is also included. The BBC archive, which includes BBC Learning English and BBC News, provided all of the evaluated video resources. Hato and Abdulmunem [25] provided the ground truth for the shots' boundary. Table 1 shows the video duration, number of frames, the number of ground truth of cut shots boundary and the number of ground truth of scene boundary for the video files that were examined.

3.1. The experimental results for the first phase: a visual summary

Precision, recall, and F-score were used as evaluation measures to evaluate the effectiveness of the suggested shot boundary detection and scene boundary detection approaches. The number of true, false, and missed detections were calculated and compared to the ground truth of shots and scenes to determine these measures. The high values of these measures imply optimal performance. Tables 2 and 3 show the performance of the proposed SBD and scene BD approaches, respectively, on test videos utilizing precision, recall, and F-score.

Both Tables 2 and 3 show that the evaluation metrics have high values, indicating that the proposed approaches have a high level of accuracy performance. The proposed SBD method's average value of the F-score has been obtained (95.618%). While the proposed scene BD method's average F-score value was reached at (94.2%). As a result, the average visual summary accuracy for both the proposed SBD and Scene BD was 94.9%.

Table 1. The video materials that were tested

Name of The Video	Duration (mm:ss)	No. of Frames	No. of Ground Truth of Cut Shots	No. of Ground Truth of Scene
BBC Learning1	9:43	14578	68	13
BBC Learning2	8:18	12453	45	15
BBC Learning3	8:1	12026	72	15
BBC Learning4	9:40	14505	76	14
BBC News	2:54	4355	9	6

Table 2. The performance of the SBD proposed method

Name of The Video	Recall	Precision	F- score
BBC Learning1	100%	95.77%	97.84%
BBC Learning2	97.78%	100%	98.88%
BBC Learning3	98.6%	93.42%	95.95%
BBC Learning4	96.1%	94.81%	95.42%
BBC News	100%	81%	90%

Table 3. The performance of the scene BD proposed method

Name of The Video	Recall	Precision	F- score
BBC Learning1	100%	100%	100%
BBC Learning2	100%	100%	100%
BBC Learning3	100%	100%	100%
BBC Learning4	100%	100%	100%
BBC News	83%	62%	71%

3.2. The experimental results for the second phase: a text summary

After experiments with the suggested deep model design's training, the most appropriate and effective values of the technique parameters were used and are presented in Table 4. Here are some concepts used in deep model construction to avoid overfitting and make an approach to better performance results: i) to avoid overfitting, several hidden units are dropped out; ii) if the loss does not decrease for some specific epochs, reduce the learning rate by 0.1; iii) early-stop, which stops training when the model's performance begins to decrease; and iv) using the attention mechanism. Using this mechanism, the decoder can make use of intermediate hidden states in the encoder. Then all of that information is used to determine which word comes next.

Rouge-1, Rouge-2, and Rouge-L were used as evaluation measures to evaluate the effectiveness of the suggested hybrid deep learning encoder-decoder model for abstractive text summarization. Experiments were conducted using a news summary dataset [45]. This dataset consists of 98401 pairs of news articles and their reference summaries. Table 5 displays the average F-score value for rough 1, 2, and L.

The high value of the Rouge Score indicates a good performance. As shown in Table 5, the proposed approach appears to have a good performance measure. The comparison of the proposed method with different video abstraction methods is shown in Table 6. The study focuses on the type of multimodal, methodology, and the evaluation metric F-score for visual summary and F-score of Rouge-1 for textual summary. F-scores with a high value imply accurate performance. As seen in Table 6, the proposed method appears to have performed better than the others.

Table 4. The hyperparameter values in proposed deep models

Hyper Parameter	Value
Embedding dimensions	200 dimensions
Hidden size	200
Batch size	128
Activation	ReLU for CNN and Tanh for LSTM
Epochs	100 with using early-stopping
Optimizer	RMSprop optimizer
Initial learning rate	0.001
Dropout	0.4

Table 5. The rouge score in proposed abstractive text summarization deep model

Measures	Value
Rouge-1	40.49
Rouge-2	25.83
Rouge-L	44.07

Table 6. Comparison different video abstraction methods with proposed method

Reference	Multimodal Used	Methodology	Visual Metric	Textual Metric
[40]	Text	Using RNN to summarize the transcript of a lecture video	___	29.72
[41]	Visual	Present textual descriptions of the visual content using CNN, then summarized using RNN	___	40.21
[43]	Visual	Using CNN and LSTM to segment the videos into shots, then the most informative ones are included in the summary	90%	___
The proposed method	Visual and audio	Using hybrid features and CRNN, we obtain the audio's textual form, and then produce a PDF with visual and textual abstractions for each scene	94.9%	40.49

4. CONCLUSION

This paper introduces a video abstraction approach that is based on multi-model video data, which comprises both audio and visual data. The proposed method's general structure and algorithm were displayed. A comparison of the suggested technique to other techniques was also provided. The hybrid features DWT and GLCM are used to detect scene and shot boundaries. Based on the highest DWT, the most informative keyframes from each scene are chosen and combined into the visual summary. and CRNN deep learning, used for abstractive text summarization. This paper obtained the audio's textual form and then produce a PDF with visual and textual abstractions for each scene. The testing videos were provided from the BBC archive, which included BBC Learning English and BBC News. A deep model was also trained using a news summary dataset. Metrics including Rouge, precision, recall, and F-score were used to evaluate the proposed approaches' performance. According to the results of the experiments, the proposed strategy outperformed the other methods.

REFERENCES

[1] A. A. A. Karim and R. A. Sameer, “Static and dynamic video summarization,” *Iraqi Journal of Science*, vol. 60, no. 7, pp. 1627–1638, Jul. 2019, doi: 10.24996/ijs.2019.60.7.23.

[2] R. F. Ghani, S. A. Mahmood, Y. N. Jurn, and L. Al-Jobouri, “Key frames extraction using spline curve fitting for online video summarization,” in *2019 11th Computer Science and Electronic Engineering (CEEC)*, Sep. 2019, pp. 69–74, doi: 10.1109/CEEC47804.2019.8974340.

[3] N. Raphal, H. Duwarah, and P. Daniel, “Survey on abstractive text summarization,” in *2018 International Conference on Communication and Signal Processing (ICCSP)*, Apr. 2018, pp. 513–517, doi: 10.1109/ICCSP.2018.8524532.

[4] V. A and D. V Jose, “Speech to text conversion and summarization for effective understanding and documentation,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 3642–3648, Oct. 2019, doi: 10.11591/ijece.v9i5.pp3642-3648.

[5] M. M. Mahmoud and A. R. Nasser, “Dual architecture deep learning based object detection system for autonomous driving,” *Iraqi Journal of Computer, Communication, Control and System Engineering*, vol. 21, no. 2, pp. 36–43, Jun. 2021, doi: 10.33103/uot.ijccce.21.2.3.

[6] W. M. S. Abedi, I. Nadher, and A. T. Sadiq, “Modification of deep learning technique for face expressions and body postures recognitions,” *International Journal of Advanced Science and Technology*, vol. 29, no. 3S, pp. 313–320, 2020.

[7] D. Suleiman and A. Awajan, “Deep learning based abstractive text summarization: Approaches, datasets, evaluation measures, and challenges,” *Mathematical Problems in Engineering*, vol. 2020, pp. 1–29, Aug. 2020, doi: 10.1155/2020/9365340.

[8] D. Patel, N. Shah, V. Shah, and V. Hole, “Abstractive text summarization on google search results,” in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, May 2020, pp. 538–543, doi: 10.1109/ICICCS48265.2020.9120998.

[9] B. Sadiq, B. Muhammad, M. N. Abdullahi, G. Onuh, A. M. Ali, and A. E. Babatunde, “Keyframe extraction techniques: A review,” *ELEKTRIKA- Journal of Electrical Engineering*, vol. 19, no. 3, pp. 54–60, 2020.

[10] P. Gunawardena *et al.*, “Interest-oriented video summarization with keyframe extraction,” in *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Sep. 2019, pp. 1–8, doi: 10.1109/ICTer48817.2019.9023769.

[11] X. Ai, Y. Song, and Z. Li, “Unsupervised video summarization based on consistent clip generation,” in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, Sep. 2018, pp. 1–7, doi: 10.1109/BigMM.2018.8499188.

[12] R. M. Hadi, S. H. Hashem, and A. T. Maolood, “An effective preprocessing step algorithm in text mining application,” *Engineering and Technology Journal*, vol. 35, no. 2, pp. 126–131, 2017.

[13] K. Wanjale, P. Marathe, V. Patil, S. S. Lokhande, and H. Bhamare, “Comprehensive survey on abstractive text summarization,” *International Journal of Engineering Research and*, vol. V9, no. 09, pp. 832–834, Oct. 2020, doi: 10.17577/IJERTV9IS090466.

[14] P. Batra, S. Chaudhary, K. Bhatt, S. Varshney, and S. Verma, “A review: Abstractive text summarization techniques using NLP,” in *2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)*, Aug. 2020, pp. 23–28, doi: 10.1109/ICACCM50413.2020.9213079.

[15] E. Naser, “Word retrieval based on FREAK descriptor to identify the image of the English letter that corresponds to the first letter of the word,” *Engineering and Technology Journal*, vol. 38, no. 3B, pp. 150–160, Dec. 2020, doi: 10.30684/etj.v38i3B.1511.

[16] M. A. I. Talukder, S. Abujar, A. K. M. Masum, S. Akter, and S. A. Hossain, “Comparative study on abstractive text summarization,” in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2020, pp. 1–4, doi: 10.1109/ICCCNT49239.2020.9225657.

[17] H. Nam and C. D. Yoo, “Content adaptive video summarization using spatio-temporal features,” in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 4003–4007, doi: 10.1109/ICIP.2017.8297034.

[18] I. U. Haq *et al.*, “Movie scene segmentation using object detection and set theory,” *International Journal of Distributed Sensor Networks*, vol. 15, no. 6, Jun. 2019, doi: 10.1177/1550147719845277.

[19] T. H. Trojahn and R. Goularte, “Temporal video scene segmentation using deep-learning,” *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 17487–17513, May 2021, doi: 10.1007/s11042-020-10450-2.

[20] M. Haroon, J. Baber, I. Ullah, S. M. Daudpota, M. Bakhtyar, and V. Devi, “Video scene detection using compact bag of visual word models,” *Advances in Multimedia*, vol. 2018, pp. 1–9, Nov. 2018, doi: 10.1155/2018/2564963.

[21] A. Rao *et al.*, “A local-to-global approach to multi-modal movie scene segmentation,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 10143–10152, doi: 10.1109/CVPR42600.2020.01016.

[22] D. Narra, Y. Madhavee Latha, and D. Avula, “Content based temporal segmentation for video analysis,” in *2020 IEEE-HYDCON*, Sep. 2020, pp. 1–5, doi: 10.1109/HYDCON48903.2020.9242711.

[23] C. Huang and H. Wang, “A novel key-frames selection framework for comprehensive video summarization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 577–589, Feb. 2020, doi: 10.1109/TCSVT.2019.2890899.

[24] M. E. Abdulmunem and E. Hato, “Semantic based video retrieval system: Survey,” *Iraqi Journal of Science*, vol. 59, no. 2A, Apr. 2018.

[25] E. Hato and M. E. Abdulmunem, “Fast algorithm for video shot boundary detection using SURF features,” in *2019 2nd Scientific Conference of Computer Sciences (SCCS)*, Mar. 2019, pp. 81–86, doi: 10.1109/SCCS.2019.8852603.

[26] C. Lv and Y. Huang, “Effective keyframe extraction from personal video by using nearest neighbor clustering,” in *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2018, pp. 1–4, doi: 10.1109/CISP-BMEI.2018.8633207.

[27] J. Valognes, M. A. Amer, and N. S. Dastjerdi, “Effective keyframe extraction from RGB and RGB-D video sequences,” in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Nov. 2017, pp. 1–5, doi: 10.1109/IPTA.2017.8310120.

[28] M. Asim, N. Almaadeed, S. Al-maadeed, A. Bouridane, and A. Beghdadi, “A key frame based video summarization using color features,” in *2018 Colour and Visual Computing Symposium (CVCS)*, Sep. 2018, pp. 1–6, doi: 10.1109/CVCS.2018.8496473.

[29] S. Tippaya, S. Sitjongsataporn, T. Tan, M. M. Khan, and K. Chamnongthai, “Multi-modal visual features-based video shot boundary detection,” *IEEE Access*, vol. 5, pp. 12563–12575, 2017, doi: 10.1109/ACCESS.2017.2717998.

[30] G. S. N. Kumar, V. S. K. Reddy, and S. Srinivas Kumar, “Video shot boundary detection and key frame extraction for video retrieval,” in *Proceedings of the Second International Conference on Computational Intelligence and Informatics*, 2018, pp. 557–567.

[31] X. Yan, S. Z. Gilani, M. Feng, L. Zhang, H. Qin, and A. Mian, “Self-supervised learning to detect key frames in videos,” *Sensors*, vol. 20, no. 23, Dec. 2020, doi: 10.3390/s20236941.

[32] V. Parikh, J. Mehta, S. Shah, and P. Sharma, “Comparative analysis of keyframe extraction techniques for video summarization,” *Recent Advances in Computer Science and Communications*, vol. 14, no. 9, pp. 2761–2771, Dec. 2021, doi: 10.2174/2666255813999200710131444.

[33] S. Pandey, P. Dwivedy, S. Meena, and A. Potnis, “A survey on key frame extraction methods of a MPEG video,” in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, May 2017, pp. 1192–1196, doi: 10.1109/CCAA.2017.8229979.

[34] E. F. Naser, “Compare between histogram similarity and histogram differencing for more brief key frames extraction from video stream,” *Journal of Physics: Conference Series*, vol. 1897, no. 1, May 2021, doi: 10.1088/1742-6596/1897/1/012022.

[35] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, “Neural abstractive text summarization with sequence-to-sequence models,” *ACM/IMS Transactions on Data Science*, vol. 2, no. 1, pp. 1–37, Feb. 2021, doi: 10.1145/3419106.

[36] H. Liang, “Research on pre-training model of natural language processing based on recurrent neural network,” in *2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Sep. 2021, pp. 542–546, doi: 10.1109/ICISCAE52414.2021.9590748.

[37] B. Marapelli, A. Carie, and S. M. N. Islam, “RNN-CNN MODEL:A Bi-directional long short-term memory deep learning network for story point estimation,” in *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, Nov. 2020, pp. 1–7, doi: 10.1109/CITISIA50690.2020.9371770.

[38] D.-J. Choi, J.-H. Han, S.-U. Park, and S.-K. Hong, “Comparative study of CNN and RNN for motor fault diagnosis using deep learning,” in *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, Apr. 2020, pp. 693–696, doi: 10.1109/ICIEA49774.2020.9102072.

[39] H. Abdullah and H. Abduljaleel, “Deep CNN based skin lesion image denoising and segmentation using active contour method,” *Engineering and Technology Journal*, vol. 37, no. 11A, pp. 464–469, Nov. 2019, doi: 10.30684/etj.37.11A.3.

[40] M. B. Andra and T. Usagawa, “Automatic lecture video content summarizationwith attention-based recurrent neural network,” in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, Mar. 2019, pp. 54–59, doi: 10.1109/ICAIIT.2019.8834514.

[41] A. Dilawari and M. U. G. Khan, “ASoVS: Abstractive summarization of video sequences,” *IEEE Access*, vol. 7, pp. 29253–29263, 2019, doi: 10.1109/ACCESS.2019.2902507.

[42] R. Agyeman, R. Muhammad, and G. S. Choi, “Soccer video summarization using deep learning,” in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, Mar. 2019, pp. 270–273, doi: 10.1109/MIPR.2019.00055.

[43] 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, Jan. 2020, doi: 10.1109/TII.2019.2929228.

[44] S. H. Emon, A. H. M. Annur, A. H. Xian, K. M. Sultana, and S. M. Shahriar, “Automatic video summarization from cricket videos using deep learning,” in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, Dec. 2020, pp. 1–6, doi: 10.1109/ICCIT51783.2020.9392707.

[45] K.Vonteru, “News summary,” *Kaggle,V2*. 2019, Accessed: May 31, 2022. [Online]. Available: https://www.kaggle.com/datasets/sunnysai12345/news-summary?select=news_summary_more.csv.

BIOGRAPHIES OF AUTHORS

Muna Ghazi Abdulsahib received the B.Sc. and M.Sc. degrees in computer science in the Department of Computer Science from the University of Technology-Iraq in 2008 and 2014, respectively. She has been a lecturer in computer science at the University of Technology-Iraq since 2018. She is currently a university lecturer in computer science in the Department of Computer Science at the University of Technology-Iraq. Her research interests include the applications of image processing, multimedia, and artificial intelligence. She can be contacted at email: MUNA.G.Abdulsahib@uotechnology.edu.iq.

Matheel E. Abdulmunim received the B.Sc., M.Sc., and Ph. D. degrees in computer science in the Department of Computer Science from the University of Technology-Iraq in 1995, 2000, and 2004 respectively. She has been a professor of computer science at the University of Technology-Iraq since 2017. She is currently a university professor in computer science in the Department of Computer Science at the University of Technology-Iraq. She has authored or coauthored more than 100 refereed journal and conference papers and 3 books. Her research interests include the applications of image processing, multimedia, pattern recognition, and artificial intelligence. On August 1, 2010, this entry was published. She can be contacted at email: matheel.e.abdulmunim@uotechnology.edu.iq.