Video to Text Summarisation and Timestamp generation to detect important events

D Shah[1,2], M Dedhia[1,3], R Desai[1,4], U Namdev[1,5] and P Kanani[1,6]

[1]Synapse, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

[2]dhirajssh@gmail.com, [3]m11dedhia@gmail.com, [4]rushildesai01@gmail.com, [5]uditinamdev@gmail.com, [6]pratikkanani@gmail.com

Abstract. With the advent of modern technology and the subsequent rise of efficient storage devices we are witnessing a rise in the number of media that is available to us. Among the most common media, the only one that takes up huge spaces on physical storage devices are videos. The primary reason for that is the addition of higher resolution videos and a greater frame rate. It is quite necessary to come up with summarisation techniques that help us understand the most important parts of the video. Apart from that, summarisation also helps us skip the non-essential parts of the video. This technology can be utilised to cut short on the time wasted on searching through the most relevant parts of the video. This paper tries to focus on the fundamental problem of summarising long videos and converting them into shorter sections that can effectively convey the same content if one were to see the entire video. Introducing timestamps also helps the viewer in jumping to the crucial events of the video. This paper makes use of deep learning algorithms such as Convolutional Neural Networks and Recurrent Neural Networks. These serve as a means of comparing different frames and generating end results.

Keywords—Summarisation, Deep learning, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Cosine similarity, Rectified Linear Unit (ReLU)

1. Introduction

Due to the simplicity of use, rapid sharing, and high image quality, large amounts of video are being uploaded on social media outlets like Facebook and YouTube. There are a number of solutions available for organising and searching still images. Similar video approaches work effectively for short snippets but fall apart when used for longer videos. Computer vision algorithms have greatly aided the organisation and search of still picture data; however, these techniques are typically computationally expensive. Technically, processing videos that are several hours long remains a difficulty. Some parents may record long segments of their baby's first birthday party to ensure that such important moments are preserved. There might be a lot of unnecessary transition time and imagery in these videos.

To address these issues, we propose techniques that make use of recent improvements in video annotation and video summarisation to reduce hour-long recordings to a manageable textual summary and construct video indexes or timestamps so that viewers can jump right to the part of the video they want to see.

The entire process of generating text summaries has been divided into four broad parts, each one imperative to the entire architecture. Firstly, we identify interesting segments from the video using advanced methods such as measurement of optical flows of frames showcasing motion within

themselves. A lot of pre-processing steps are applied, and these form an important chunk of the model. We use a variety of methods to calculate several scores that cumulatively represents a final score which takes a lot of different aspects into consideration. Further explanation under Fitness scores. The contributions of this paper include:

- A way of extracting both, the important events along with their timestamps, and the summary from a single computation pipeline. This helps in cutting down the computation time drastically as compared to having different methods for the summary generation and the important events extraction.
- A different way of calculating superframe cut scores, motion scores and uniqueness scores of each frame in the video. The scores take colourfulness, contrast, edges and superframe quality into consideration.

2. Literature Review

Through Video summarisation and Indexing, this paper aims to reduce the viewing time of convoluted videos and ease large-scale browsing of videos by producing succinct summaries which are representative of the original videos. This will allow users to search through only the highlights of the video. Moreover, further summarisation of the short clips into text also helps to grasp only the essential parts of the video for a reader. There have been several works which have incorporated various different methods in achieving the summary of a video. It has been observed that research in Video Summarisation has been diverse and advanced in recent years. May it be unsupervised or supervised, the goal of a summarisation is to produce a compact visual summary that encapsulates the most informative parts of a video.

Previously, unsupervised methods were employed for the automated process of video summarisation. [1] proposed a method for producing static video summaries on the basis of colour feature extraction from video frames as well as k-means clustering algorithm while [2] performs unsupervised video summarisation that finds visually important shots by using title-based image search results. Criteria such as diversity, representativeness and relevance are also outlined by the researchers to select the most important frames from a video [3, 4]. Additionally, various heuristics were defined in order to showcase the significance of the frames and the scores were used to determine the characteristic frames to be used in the summarised video. For example, [5] focuses more on the most significant persons and objects with whom the camera wearer interacts, and thus introduces regional cues like frequency of occurrence, gaze, and closeness to hands that are indicative of high-level saliency in egocentric video. This helps them predict the relative importance of new regions on the basis of these cues. Some Researchers [6] have proposed a summarisation algorithm that makes use of web-image based prior information to automatically select the maximally informative frames from a video in an unsupervised manner. Using the SVM framework, they discover ‘canonical viewpoints or subclasses in web-images which further helps in identifying the keyframes used for summarisation. [7] uses a deep summarisation network (DSN) to create video summaries. This predicts the probability of selecting a video frame and then based on the probability distributions forms the video summaries. By proposing a label-free reinforcement learning algorithm for the problem, they outperformed other state-of-the-art unsupervised alternatives and produced better results than even some of the other supervised methods. We on the other hand, make use of a mathematical framework and calculate scores based on different properties such as colour, movement and uniqueness and identify the frames scoring at the top of such score-based ranking.

Many Researchers have also used supervised methods for the summarisation process. [8-10] viewed video summarisation as a subset selection problem. [8] proposed a sequential determinantal point process model to explicitly learn subset selection from human-given summaries. The standard determinantal point process model (DPP) treats the frames of the videos as randomly permutable items meaning it fails to capture their inherent sequential nature. The seqDPP overcomes this flaw. [9] uses a submodular function to optimise a global objective function of the desirability of selected frames whereas [10] proposed subshot-based summarisation technique to nonparametrically transfer summary structures from training videos to test videos. Another method used was Long Short-Term Memory

(LSTM) network which is a type of recurrent neural network (RNN) capable of learning order dependence in sequence prediction problems. LSTM was used by [11] for modelling the variable-range temporal dependency amongst video frames which helped in deriving compact and representative video summaries. [12] makes use of fully convolutional sequence models for summarisation of videos by adapting popular semantic segmentation networks for the same. This is quite different from the LSTM methods used in the previous works like [11] which processed the frames in a sequential order. On the other hand, [12] the models employed processed all frames simultaneously, which helps take full advantage of GPU parallelisation. Results obtained by such methods have better scope in comparison to the traditional unsupervised methods. Since these methods specifically learn from human-created summaries, they are better in predicting how a person would summarise an input video. We use a similar approach with an RNN where the superframes that we generated give us captions as sentences for each segment and we apply a loss function to attain a very accurate caption and minimize the loss.

A lot of methods mentioned above focus on identifying the important keyframes and subshots without really evaluating these summarised videos. Moreover, navigating long videos becomes a difficult task and hence captioning the video in the form of text becomes necessary. Video captioning consists of feature extraction and caption generation based on the extracted features from input video clips. A lot of works [13-16] used the encoder-decoder framework to achieve video captioning. Some researchers [13] made use of an encoder-decoder framework which was based on RNN, specifically LSTM which was used to encode the video into a compact representation and also the video representation is decoded into a video caption. By proposing a hierarchical-RNN framework for video captioning, [14]'s model is able to generate paragraphs for long videos as well. Though this produced interesting results, there were many limitations for the same. Another researcher's work [15] also comprised an encoder-decoder framework for video captioning like the ones mentioned above but they used their model to learn a video representation shared between the encoder and the video captioning parts and hence it was different from the other related models. They incorporated the usage of multi-task learning which jointly modelled both the video summarisation and the video captioning and generated a captioned summary that enabled users to index as well as go through only the essential parts of the video. Since their model only implemented a simple encoder-decoder RNN, it could not surpass the performance which could have been achieved by using a hierarchical or a bi-directional RNN. Additionally, S. Sah [16] proposed methods to produce video summaries of long videos and also generate annotations for key frames extracted from the interesting segments of the whole video. They further summarised the captions to generate textual summaries which represented the input video. In our paper, for captioning we derive features of superframe cuts, and those features are fed into the RNN as described below in the methodology section. Each segment generates a sentence or a phrase as the output describing the action happening in that part of the video. RNN was used for this purpose.

3. Methodology

After reviewing the current state of the art architectures, the summarisation and captioning model was constructed using supervised learning. We don't have labelled data about timestamps where according to human interpretation, important events occur or the video segments that could perfectly replace the entire video in conveying the entire idea behind the video. The entire process of generating text summaries has been divided into four broad parts, each one imperative to the entire architecture. Interesting segments are identified from the video using advanced methods such as measurement of the optical flow of frames showcasing motion within themselves. Other than that, image pyramids have been used to downsize or down sample each frame in order to reduce its resolution so that the video could be processed more efficiently. Although these methods constitute pre-processing of the data, they form an important chunk of the model. A mix of these methods are used to calculate several scores that cumulatively look after each and every part of the video. Further explanation is given under the Scoring Functions section. These calculations are used in a linear fashion to generate a final score to rank superframes that have been shortlisted in comparison with each other. This approach is used to eliminate superframes that might have made their way to the most interesting ones just due to one score but haven't

been classified as important by other scores. Since not all scores are as important as others, we add weights to them to affect the final score in a manner that gives us the desired result. This also incorporates the adjustments required to give higher priorities to scores that directly relate to content in the video and not just metrics of the videos such as sharpness, colour saturation, etc. The combined score is henceforth known as Fusing Score.

3.1. Dataset

Microsoft Common Objects in Context or COCO dataset has been used to train and test the caption generating model. COCO has been used because it is one of the most popular image datasets, with applications like object detection, segmentation, key-point detection and captioning [17]. This dataset consists of approximately 328K images. The most important feature of this dataset is that every image has at least five different manually written captions. The video used for testing the model is about New York City and its tourist attractions, which is Expedia's promotional video [18]. The video used is approximately 6 minutes long.

3.2. Scoring Functions

This method begins by calculating the superframe cut scores, motion scores and uniqueness scores of each frame in the video. The various different scores that are calculated are:

3.2.1. Colourfulness [19]. The degree to which an image contains colour. It also suggests how luminescent a frame is and what kind of colour degradation it has gone through. The given formulae have been used:

$$\widehat{C} = \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb}, \quad (1)$$

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2}, \quad (2)$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \quad (3)$$

3.2.2. Contrast [19]. To compute the Contrast Score, each and every frame in the superframe segment is converted to Luminance [16]. Contrast score is defined as the standard deviation of the luminance pixels; hence it is converted to grayscale first.

3.2.3. Edges [20]. The edges are computed using Laplacian transform. It is used for measuring the sharpness of each frame. The edges are taken into consideration to ensure that the keyframes selected have a certain level of sharpness.

3.2.4. Uniqueness Score. This can also be referred to as the super frame quality score that uses the above three scores in a non-linear fashion.

3.2.5. Superframe Score. The superframe score is the overall score of the superframe segment which is the average of the fusion of all three scores, colourfulness, contrast and edges calculated for each frame. The superframe score is a mixture of the uniqueness score and the optical flow using Lucas-Kanade method. In Lucas-kanade method points are given to track, and optical flow vectors of those points are received. The Lucas-kanade method fails when there is large motion. So, pyramids are being used. On going upwards in the pyramid, small motions are filtered out and large motions become small motions. By using Lucas-Kanade, optical flow along with scale is received [21].

3.3. Keyframe Extraction

Each frame is extracted using OpenCV and its frames per second (fps) value is prefetched. Using fps and the length of the video the total number of frames in the video are found. The image frames in the video that symbolise and summarise it are referred to as keyframes. If shots can be detected and

transitions between them disregarded, keyframes can be used as shot boundaries. A threshold filter is applied to filter out all frames and their timestamps that lie above the given threshold. In our case, a threshold of 0.12 for summary generation and 0.20 for time stamps generation has been used. After getting the array index positions, the time duration of each segment is derived. From these segments the key frames are extracted using dimension reduction and dynamic clustering. First using singular value decomposition (SVD) the top 63 values from the histogram of each superframe segment are derived. This is important because each frame/column in the resultant vector can be represented by 63 dimensions. Thus, the dimensions of the feature-frame matrix are reduced. To find which all frames are similar, dynamic clustering of projected frames is used. This is done by creating clusters of consecutive frames with the help of Cosine Similarity to check whether the new frame is similar to the last cluster formed or not. Only those frames that have more than 25 clusters and are eligible to be a part of the superframe shot are filtered out. The other frames in the sparse clusters are excluded since they are considered as transitions between shots. The frames that are eligible make a shot and the shots' last added frame is chosen as a keyframe. The keyframes given as output are saved in png format. equations are an exception to the prescribed specifications of this template.

3.4. Model Description

This section describes the process of getting the training set, which is a combination of both image and caption, ready for model training. InceptionV3 has been used for feature extraction. InceptionV3 is a widely used image recognition model which has been developed by multiple researchers over the years [22]. The final classification layer of the inceptionV3 model has been cut off so that the last convolutional layer becomes the output layer. The output represents the extracted features. The shape of the output is 8x8x2048. These extracted features have then been cached onto the disk for use in the training process. Caching the extracted features on the RAM would be a lot quicker but it would also be highly memory intensive, so it is not preferable. These features are cached because processing each image with InceptionV3 could become a bottleneck while training.

For the next step, the captions are pre-processed and normalised. First, the captions are tokenised, only the top k most frequent words from all the captions are added in a dictionary. For this project, after testing different values, we have kept k = 5000 since it gives good results. Then, phrase to index and index to phrase mappings are created, because the words need to be converted into a number to be able to be fed into the model. To map back the index generated by the model to the word it represents, index to word mapping is used. Finally, all the captions are padded so that the vectors are of an equal size.

Coming to the actual model architecture, a Convolutional Neural Network (CNN) encoder and a Recurrent Neural Network (RNN) decoder have been used for the entire model. After extracting features from InceptionV3's lower convolution layer, we get a vector of shape (8, 8, 2048) which is compressed to (64, 2048). The vector is passed through the CNN encoder which contains a single fully connected layer with a Rectified Linear Unit (ReLU) activation function.

ReLU Activation function:

$$R(z) = \begin{cases} z & z > 0 \\ 0 & z <= 0 \end{cases} \qquad (4)$$

The CNN encoder is used to extract a set of feature vectors. The encoder passes the extracted features through a fully connected layer. The RNN, or in our case GRU, attends over the image to predict the next word of the caption, taking the previous generated word into consideration. We make use of the Bahdanau Attention model [23] which looks at a given set of tokens and only focuses or in more technical terms attends to those tokens that based on previous tokens add more context or meaning into the resultant set of sequential tokens. Once all vectors are initialised, training is started. The optimisation algorithm used is an Adam Optimizer [24] and the loss function used is Sparse Categorical Cross-Entropy [25]. Sparse Categorical Cross-Entropy is an integer-based version of the Categorical Cross-Entropy loss function [26].

Categorical Cross-Entropy Loss:

$$CCE(p,t) = -\sum_{c=1}^{C} t_{o,c} \log(p_{o,c}) \quad (5)$$

Each epoch and its corresponding loss are tracked to estimate how many training epochs would be the best. Once the model has been trained, it is tested by picking out random samples from the dataset and the provided caption and the caption generated from the model are compared [27].

The model is incorporated in this project in the following manner. Once the key-frames are extracted, a pair-wise comparison of the key-frames, using structural similarity, and the generated caption of each key-frame, using a cosine similarity function, is done. Cosine similarity function is used to measure the similarity between two non-zero vectors and is calculated as the "angle" between these two vectors.

Cosine Similarity:

$$\text{Cos}\,\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}} \quad (6)$$

Here, this function is indicative of the degree to which two consecutive frame captions are similar. A threshold is assigned for the similarity which when exceeded suggests that the two frame captions are very similar and contain almost similar kind of information. The ones that fall below the threshold are included in the final summary or the timestamp event.

The Figure (figure 1) depicts the entire schematic flow diagram of the video summary generation process. This diagram shows the process in a concise manner. The input video that is to be analysed is passed into a feature extractor that gives 5 scores namely Colourfulness, Contrast, Edges, Uniqueness and Superframe score. These scores are then used to generate sections of video called superframe segments which will be the input to the feature map encoder. The features extracted are then used in an RNN decoder as described previously making use of GRU units and Bahdanau attention model to generate sentences of those superframe segments. Finally, these sentences are analysed to generate the entire summary of the video.

4. Results

The video used for testing the model is a vacation travel guide for New York City. It's a promotional video of a travel company called Expedia [18]. This video is about 6 minutes long and underscores the important landmarks of the city like Central Park, Times Square, and Brooklyn Bridge to name a few. Such Travel videos help in attracting tourists and also giving a glimpse of life in a different country or city. Generating a summary and timestamp for this video would help the viewer in jumping to the most important parts which can be the major tourist attraction or a crucial event happening in the video.

4.1. Graphs

Various graphs were plotted to help determine the most important segments based on motion and combination of the scores. The Graph (figure 2) is a plot of Normalised motion scores against the corresponding frame number. We can see that the segments of video having scores greater than our threshold seem to be quite important and need to be included for summary generation. The graph shown (figure 3), ranks the segments based on the combined scores calculated using different scoring functions described above in methodology. Here, if the motion detected in the frame is low, a higher score is assigned to that frame. Even the sharpness of the frame should be more for a better score. As seen in the plot, the frame of the segment in the beginning of the video has motion nearly equal to zero and hence has a score which is almost 1. This is an anomaly and thus the first superframe segment is discarded. To get the important frames, a threshold is set. If the summation score or the combined score is less than 0.12, it's considered to be zero. By doing this, the frames with better scores are visible in the graph (figure 4).

4.2. Final Summary generation

The Summary generated by the model based on Superframe cuts:
(1) a mountain view over the desert is in it's lake waters

(2) a poster of a large amount of sheep on the land window
(3) all different clocks hanging from a fish
(4) a group of people waiting on a large city with a clock that looks at the top of a larger boat built in to a building and a brick building
(5) an outdoor figure of flowers in the grass
(6) the crowd of people walk along a city skyline near the docks
(7) a number of the character on the train on the train engine pulled by a bridge and another on shore
(8) an airplane with long brown an audience stands beside the sky
(9) at the contents outside each under a town
(10) a group of kids in a political event with <unk> a woman looking at a bus at building that's raised trains
(11) some people in front of a building
(12) a fire hydrant spraying water in the rain
(13) a crowd of people in a crowd holding lab furniture and some trees
(14) there are several red and white flowers with pink flowers
(15) a beautiful picture of cows walking out in a field under a wooded area
(16) the man on sidewalk wearing a red helmet tries to board a skateboard
(17) a big selection of cakes outside of this food shop
(18) some people walk beneath an umbrella
(19) a man licks several people standing around a long table having food
(20) adults dressed up
(21) some green <unk> running while bird feed a <unk> covered in the park
(22) a man grabbing umbrellas
(23) several buildings on the water next to a large boat on a large city sidewalk
(24) man waiting with a <unk> room
(25) some people are on boogie board.

Here '<unk>' is an unknown character. The frames with some of the important captions are shown below (Fig. 5 - 10).

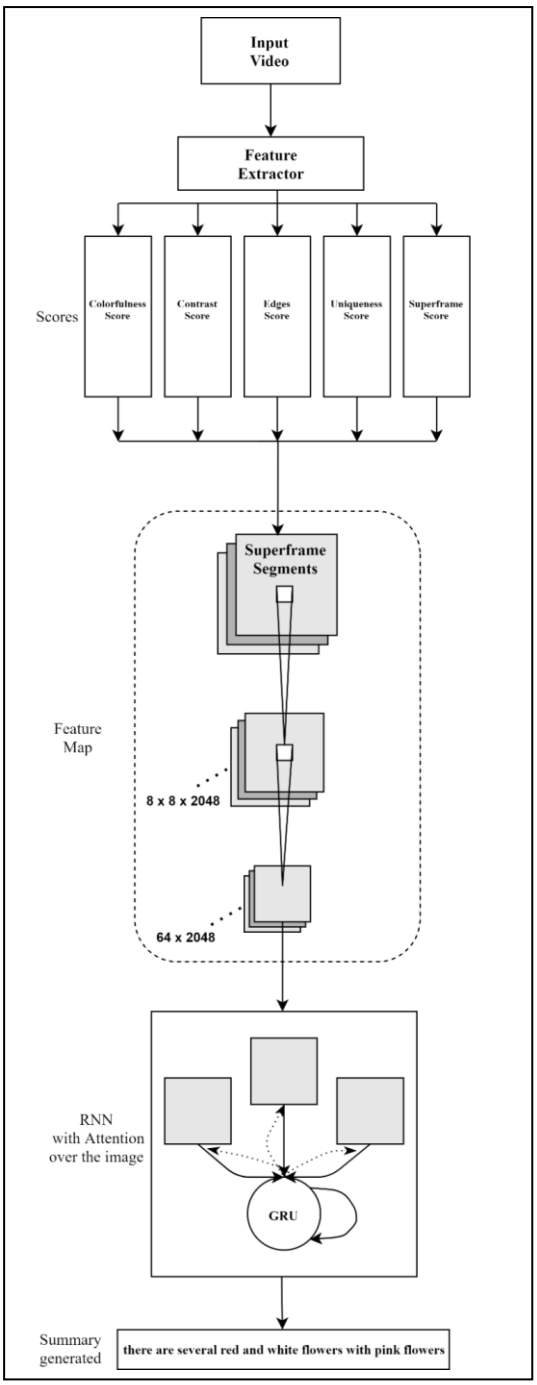

Figure 1. Schematic Flow Diagram

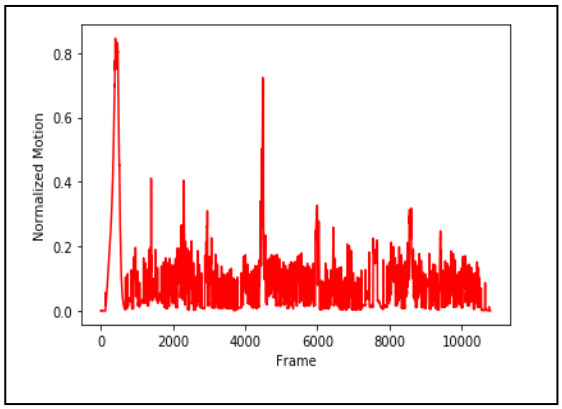

Figure 2. Normalization Motion vs Frame

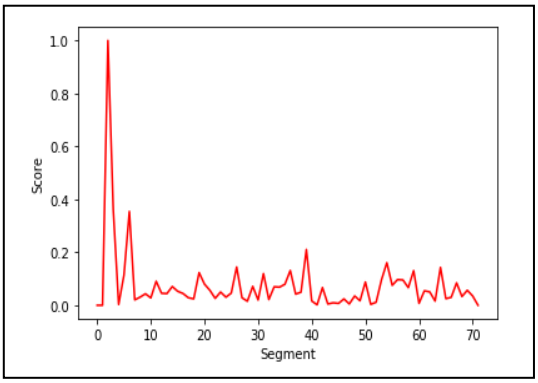

Figure 3. Score vs Segment

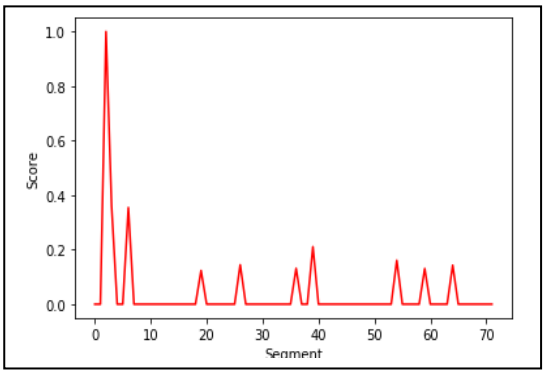

Figure 4. Score vs Segment (threshold)

4.3. Timestamp Generated

The Table (table 1) shows the timestamps generated with the corresponding important events that were detected by the model. Timestamps were generated for all the keyframes from the top superframe segments which had the total score greater than 0.20. There happened to be 8 such frames.

Table 1. Timestamps Generated.

Timestamp (s)	Events
17	Many flowers in the hole
20	An outdoors pools surrounded by sky on take off
32	A bird laying on top of a bench
33	A room that includes a bunch of <unk> by the market
35	A clock tower with a clock tower in the middle of a building
196	A child and a deep in the middle of a grove of flowers
198	A yellow stand <unk> across a lush greenfield
200	A man on a skateboard in the side

Figure 5. Frame No. 5

Caption: an outdoor figure of flowers in the grass

Figure 6. Frame No. 10

Caption: a group of kids in a political event with <unk> a woman looking at a bus at building that's raised trains.

Figure 7. Frame No. 11

Caption: some people in front of a building

Figure 8. Frame No. 14

Caption: There are several red and white flowers with pink flowers

Figure 9. Frame No. 17

Caption: a big selection of cakes outside this food shop

Figure 10. Frame No. 18

Caption: Some people walk under an umbrella

These were some of the frames that generated captions from the testing dataset. There were frames like frame no. 10 (figure. 6) that have given poor captions that do not aptly describe the given frame. Rest of them gave accurate captions that perfectly describe the frame.

The frames (figure. 11 - 12) shown below are at definite timestamps which were generated by the model. Important events from the video are selected and two of them are included here. This will help the viewer to choose and move to the mentioned events and save their time.

Figure 11.

Caption: a child and a deep in the middle of a grove of flowers

Timestamp: 196 seconds

Figure 12.

Caption: a man on a skateboard in the side

Timestamp: 200 seconds

5. Conclusion

Summarisation helped us skip the non-essential parts of the video. This technology can also be utilised to cut short on valuable time wasted by jumping to the most relevant parts of the video. The core problem of summarising long videos that take up a lot of space has been addressed in this paper. This paper helps convert long videos into shorter sections that convey the same meaning. Timestamps would help the viewer to skip to the crucial events of the video.

This research describes a coherent technique for video summarisation as well as timestamps. Frame-to-frame motion, image quality, cinematography, and personal choice are all combined in a unique manner to obtain fascinating portions in both short and lengthy films. From the most significant parts of a video, keyframes are removed and converted to textual captions using an CNN encoder-RNN decoder neural network. Human evaluation of the summary and timestamps generated by this method indicate promising results. A key drawback of this model is that the captions generated at times are not the best for representing a particular keyframe. With the provision of more datasets that contain true values on both key segments and associated captions, one could potentially find a solution.

References

[1] Avila SEF, Lopes APB, Luz A, and Araújo AA 2011 VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method *Pattern Recognition Letters* **vol**. **32 no. 1** pp. 56–68

[2] Song Y, Vallmitjana J, Stent A, and Jaimes A 2015 TVSum: Summarizing web videos using titles *Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 5179–5187

[3] Lu Z and Grauman K 2013 Story-driven summarization for egocentric video *Proceedings / CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition* pp. 2714–2721

[4] Liu T and Kender JR 2002 Optimization algorithms for the selection of key frame sequences of variable length *European Conference on Computer Vision* pp. 403–417

[5] Lee YJ, Ghosh J, and Grauman K 2012 Discovering important people and objects for egocentric video summarization *Computer Vision and Pattern Recognition (CVPR)* pp. 1346–1353

[6] Khosla A, Hamid R, Lin CJ, and Sundaresan N 2013 Large-scale video summarization using web-image priors *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision*

and Pattern Recognition pp. 2698–2705
[7] Zhang Y, Kampffmeyer M, Zhao X, and Tan M Deep 2019 reinforcement learning for query-conditioned video summarization *Applied Sciences* **vol 9**, p. 750
[8] Gong B, Chao WL, Grauman K, and Sha F 2014 Diverse sequential subset selection for supervised video summarization *Advances in Neural Information Processing Systems* **vol 27**
[9] Gygli M, Grabner H, and Gool V 2015 Video summarization by learning submodular mixtures of objectives *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 3090–3098
[10] Zhang K, Chao WL, Sha F, and Grauman K 2016 Summary transfer: Exemplar-based subset selection for video summarization *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 1059–1067
[11] Zhang K, Chao WL, Chao F, and Grauman K 2016 Video summarization with long short-term memory *European Conference on Computer Vision* pp. 766–782.
[12] Rochan M, Ye L, and Wang Y 2018 Video summarization using fully convolutional sequence networks *European Conference on Computer Vision* pp. 358–374.
[13] Venugopalan S, Rohrbach M, Donahue J, Mooney, R Darrell T, and Saenko K 2015 Sequence to sequence – video to text *International Conference on Computer Vision (ICCV)*
[14] Yu H, Wang J, Huang Z, Yang Y, and Xu W 2016 Video paragraph captioning using hierarchical recurrent neural networks *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2016 pp. 4584–4593
[15] Chen BC, Chen YY, and Chen F 2017 Video to text summary: Joint video summarization and captioning with recurrent neural networks
[16] Sah S, Kulhare S, Gray A, Venugopalan S, Prud'hommeaux E, and Ptucha R 2017 Semantic text summarization of long videos *IEEE Winter Conference on Applications of Computer Vision (WACV)* pp. 989–997
[17] Lin TY et al. 2014 Microsoft COCO: Common objects in context *European Conference on Computer Vision* pp. 740–755.
[18] 2013 New York City Vacation Travel Guide | Expedia *www.youtube.com* https://youtu.be/MtCMtC50gwY (accessed Aug. 20, 2021)
[19] Hasler D and Suesstrunk S 2003 Measuring colourfulness in natural images *Proceedings of SPIE - The International Society for Optical Engineering* **vol**. **5007** pp. 87–95
[20] Aziz F, Wilson RC, and Hancock ER 2012 Shape signature using the edge-based Laplacian *International Conference on Pattern Recognition* pp. 1594–1597.
[21] 2017 OpenCV: Optical Flow *Opencv.org* https://docs.opencv.org/3.3.1/d7/d8b/tutorial_py_lucas_kanade.html (accessed Aug. 23, 2021).
[22] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, and Wojna Z 2016 Rethinking the inception architecture for computer vision
[23] Bahdanau D, Cho K and Bengio Y 2014 Neural machine translation by jointly learning to align and translate ArXiv, **vol**. **1409**
[24] Kingma D and Ba J 2014 Adam: A method for stochastic optimization *International Conference on Learning Representations*
[25] Mannor S, Peleg D and Rubinstein R 2005 The cross entropy method for classification in Machine Learning *Proceedings of the Twenty-Second International Conference (ICML 2005)* pp. 561–568
[26] 2019 How to use sparse categorical crossentropy in Keras? *MachineCurve MachineCurve* https://www.machinecurve.com/index.php/2019/10/06/how-to-use-sparse-categorical-crossentropy-in-keras/
[27] Xu K et al. 2015 Show, attend and tell: Neural image caption generation with visual attention *Proceedings of the 32nd International Conference on Machine Learning PMLR* **vol**. **37** pp. 2048–2057