Searching a Video Database using Natural Language Queries

Shubha M[1], Kritika Kapoor[1], Shrutiya M[1], and Mamatha H. R.[1]

[1] PES University, Bangalore Karnataka 560085, India
[1]{shubhamhegde6,kapoorkritika66,shrutiya03}@gmail.com,
mamathahr@pes.edu

Abstract. This paper describes an application that achieves voice based natural language query, search and extracted video segment playing after the search in order to query the content of the videos in a user-friendly manner. Two different models were explored for the same. The first model is implemented using an image captioning approach. Two different image captioning methods are used for creating tracklets, namely Densecap and NeuralTalk2. NeuralTalk2 generates a single appropriate caption for the entire image whereas Densecap generates multiple captions corresponding to specific regions of interest in the image. These captions are used to preprocess the video and create semantically similar tracklets. Given a video and a voice-based natural language query, this system will produce video tracklets from the video that are semantically relevant to the query. The second model uses an audio processing approach. Here, first the transcripts generated by YouTube are collected. The voice query is taken as input and the most relevant segments of the video are retrieved using on-the-fly generation of tracks and merging if required. For finding Semantic similarity in both the models, first Universal Sentence Encoder (by Google) which uses a deep averaging network encoder (DAN) for converting the sentences into 512 dimensional vectors is used and then cosine similarity between the vectors is calculated.

Keywords: Image captioning, NeuralTalk2, Densecap, Video search, Audio Search, Audio Processing, Natural Language Queries, Parallel Processing.

1 Introduction

There has been immense growth in video technologies in the previous decade due to the advent of video sharing platforms and increased use of devices to capture videos for personal use leading to a rise in the use of video databases. And hence, video databases have become popular in various fields. Video databases need easy to use interfaces to retrieve video segments. A natural language interface for querying video databases has immense importance. The scope of querying video databases is huge. For example, consider the following queries:

1. A red car in front of a white building. (This can be queried in security footage database)

2. Man, in a blue jacket next to a woman. (This sort of queries can be used in journalism for identification)
3. Ball in goal post. (This can be queried in sports events video databases)
4. Derivation of Naive Bayes (College lecture videos.)

Knowing the significance of such video querying systems, several attempts have been made to build video data models. However, most of these models are usually content and rule based. In such models, the semantic content which includes objects, activities, and spatial properties of objects are taken into consideration. Objects are detected using bounding rectangles and are primarily used for querying event based and spatial relations. However, these are usually specific to the application and domain dependent. These models focus on mining characteristic concepts or objects or activities and constructing a sentential query through a predefined rule-based structure. Since these models follow a rule-based approach, it can cater to only those queries that come under the predefined rules or graphs that were used to extract the semantic meaning from the sentence. Certain level of annotation is required.

However, the model prescribed by us does not need annotation of the video database. Instead image captioning tools are employed to extract information about the objects and events through deep neural networks in the image captioning approach. This information is used for querying which makes our model domain-independent and serves general purpose.

In addition to the visual features of the video, the model also focuses on the audio features. This can be used on all videos with audio to seek to the part where anything related to the given query was present (Semantically similar). By giving a keyword as the query, we can retrieve the video segments of a huge video where that or anything similar to that is being explained as in the case of a college lecture video.

In addition to the video data model, a voice based natural language querying interface is used in our system. A natural language querying interface is more useful to query the video segments as it is more flexible and user-friendly. There is no need to learn how to use an online tool or a separate querying language.

2 Related Work

Several studies related to this have been published. Many video data models are concept based and have an object-oriented approach. BilVideo [1] is one such system that uses POS tagging information to group the specified queries as object, spatial, and trajectory queries. It constructs the queries as Prolog facts and forwards it to the query processing engine. This uses the knowledge base and object-relational database to provide the results. BilVideo has a visual interface (Web-based) for query specification unlike our natural language interface. Natural language querying interface is more desirable than other forms of interfaces since it provides more flexibility where the user can use his/her own sentences for querying.

Another similar system [2] is based on a content-based video data model that caters to spatio-temporal and trajectory-based queries. It uses the semantic content which includes objects, activities, events and spatial properties of objects. Information extraction techniques are used to extract the semantic representations of the queries. This semantic information is used to query the object database. Conceptual ontology module is implemented with WordNet. This uses word-based embeddings which may not extract the complete meaning of the sentence.

However, both these systems majorly require structured models to relate to the objects/concepts from the video database based on certain rules which makes the application specific to certain cases only and less general purpose.

Using natural language to describe an image using deep neural networks provides a good language model to extract information from the images. Image Captioning refers to the process of generating textual natural language description from an image – based on the objects and actions in the image. Densecap [3] is a model whose architecture consists of a CNN, a dense localization layer, and an RNN language model that produces the labels. It generated multiple captions (image captioning) corresponding to specific regions of interest (Object detection). NeuralTalk2 [4] is an image captioning project implemented in Torch. It is an end-to-end model which is implemented using a fine-tuned CNN (conventional VGGNet) followed by RNN and is trained on the MS COCO dataset. It generates a suitable caption given an image. Another related project tackles the problem of searching a person in huge image databases using natural language-based description [5]. It is a model built on NeuralTalk2 that uses RNN with Gated Neural Attention mechanism (GNA- RNN). These projects have immense importance in computer vision for natural language description of images. However, they are models for an image and haven't tackled the problem of video segment retrieval. But these methods can be easily extended to work with videos and forms the crux of our attempt.

Another model [6] uses Densecap to generate multiple captions per image and conducts a tracking by caption to retrieve video segments. It uses Skip-thoughts vector for sentential encoding and for performing semantic similarity. Our model is inspired from this paper. However, we explore two approaches for our model, a state-of-the-art image captioning technique, NeuralTalk2 which generates an image for the entire image and Densecap, which is an intersection of object detection and image captioning. We also use a newer sentence embedding provided by Universal Sentence Encoder (by Google) [7] which uses a deep averaging network encoder (DAN) for converting the sentences into 512 dimensional vectors. Henceforth, our paper mainly draws from NeuralTalk2, Densecap and aforementioned models.

3 Methodology

3.1 Overview

Our goal at designing a model to solve this problem is broadly divided into two approaches, image-captioning approach and an audio approach.

The image captioning approach, in brief is as shown, in Fig. 1, consists of three sequential parts –

1. Generate Captions
2. Creating Tracklets

3. Voice based Search

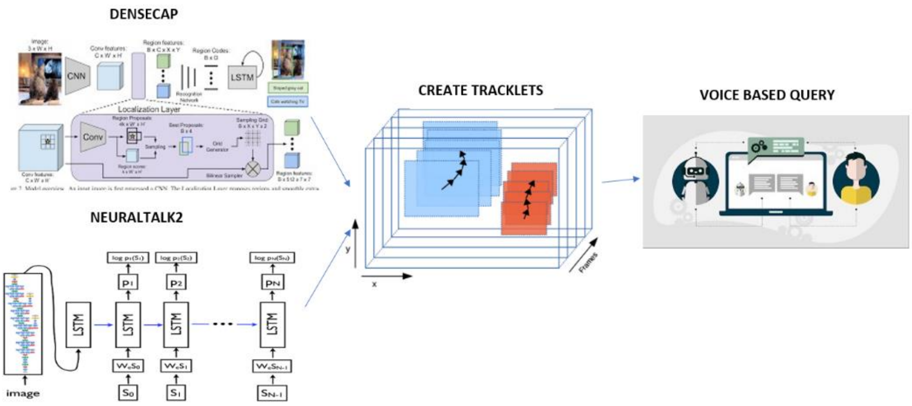

Fig. 1. Overall Structure, (a) Generate Captions for each frame (b) Create Tracklets (c) Voice Based Query. [3, 16, 17, 18]

The video clip is split into frames. Each frame is passed to an image captioning model to generate captions. In order to generate appropriate captions, we employ two different kinds of image captioning models, Densecap and NeuralTalk2, and compare them to see which gives better results. For every frame, the Densecap model outputs several bounding boxes with captions that best depicts the scenario. On the other hand, NeuralTalk2 is an efficient captioning code that generates one caption per image.

Secondly, we create tracklets by tracking the frames based on captions. In contrast to the conventional methods, it pays attention to both coordinates of bounding boxes and semantic meanings of regions derived from the previous frames. As frame sequences pass, we obtain the semantic similarity between consecutive frame captions, compare the meanings of the boxes and create tracklets that are semantically relevant. As a result, several semantic tracklets are obtained, each containing frame information and a representative caption.

As the last part of our model, we take a voice based natural language query as input and retrieve the appropriate segments of the video that are best described by the input query. Using a speech recognition system and natural language processing, the model evaluates the semantic similarity between the input text query and the depictive captions of all the tracklets obtained from the video and outputs the most relevant track.

Secondly, the audio approach does not require pre-processing. For this approach, we process the transcripts of the video given as input. We employ an on-the-fly track generation technique by using semantic similarity as the base criteria. All the results are shown on the original video so that it's user friendly and the context of the concept being displayed in the video can be easily grasped by moving back and forth.

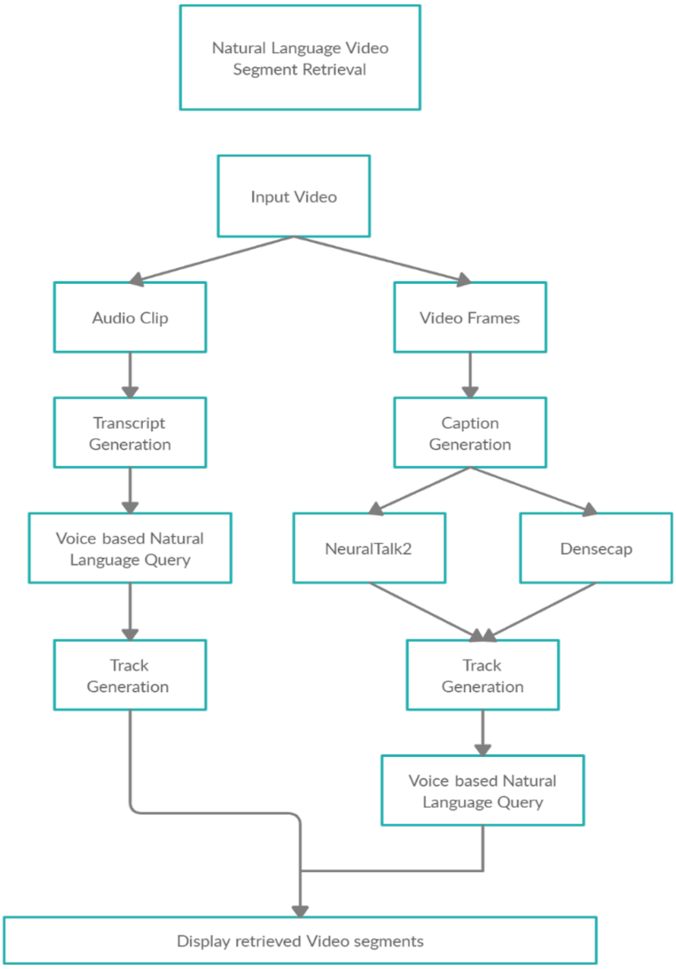
Natural Language Video Segment Retrieval
Input Video
Audio Clip
Video Frames
Transcript Generation
Caption Generation
Voice based Natural Language Query
NeuralTalk2
Densecap
Track Generation
Track Generation
Voice based Natural Language Query
Display retrieved Video segments

Fig. 2. Overview of the model approach

3.2 Video Approach

The video processing procedures for video retrieval is a three-step process. First, extract the visual features, followed by creation of tracklets. Lastly, retrieval of the most relevant videos. The three steps have been explained in detail below.

Generating Captions. This is the first step towards building this system. The video clip is split into fixed divisions called frames, by taking time as the main factor, by deciding an appropriate frame rate. After this, two varied image captioning models, namely NeuralTalk2 and Densecap were applied on each frame. NeuralTalk2 generated one appropriate caption describing the entire image whereas Densecap generated multiple captions corresponding to specific regions of interest, denoted by bounding boxes, in the image. The two models have been explained in detail below.

Densecap. Our first approach makes use of the Densecap model [3] that is a Fully Convolutional Localization Network architecture built using a Convolutional Network, the state-of-the-art VGG-16 model followed by a localization layer, Recognition Network and a RNN language model.

Densecap was developed with the ideas of both object detection and caption generation and given an image as input, it identifies relevant objects in the image by virtue of its object detection methodologies and using these objects, generates on an average of 80 captions per image (An example is shown in Fig. 3), with corresponding weights, and the corresponding bounding boxes of the objects all dumped into a json file. These captions and the coordinates of the boxes are then used to form tracklets in the next step.

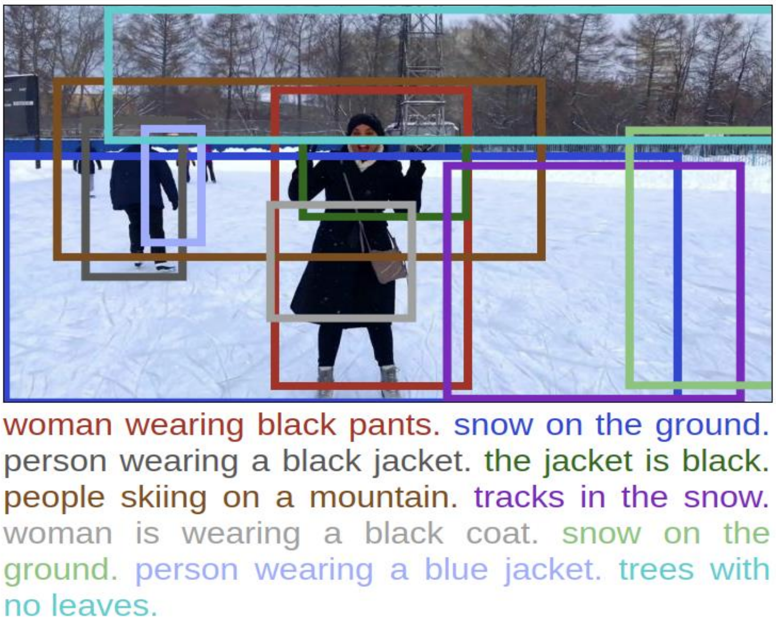
woman wearing black pants. snow on the ground.
person wearing a black jacket. the jacket is black.
people skiing on a mountain. tracks in the snow.
woman is wearing a black coat. snow on the ground. person wearing a blue jacket. trees with no leaves.

Fig. 3. Caption generated using Densecap [3]

NeuralTalk2. The second approach to image captioning involves the use of the NeuralTalk2 model [4], an efficient captioning code, extensively trained on the object-detection efficient image-captioning dataset Microsoft-COCO, written in Torch, a model which is implemented using a CNN (conventional VGGNet) followed by RNN.

Supporting CNN finetuning, an image is given as input to NeuralTalk2 and generates one appropriate caption per image. An example is shown in Fig. 4. The script is evaluated by sending the frames as a batch and a json file is generated which maps every image with its caption. This is later used in the making of the tracklets.

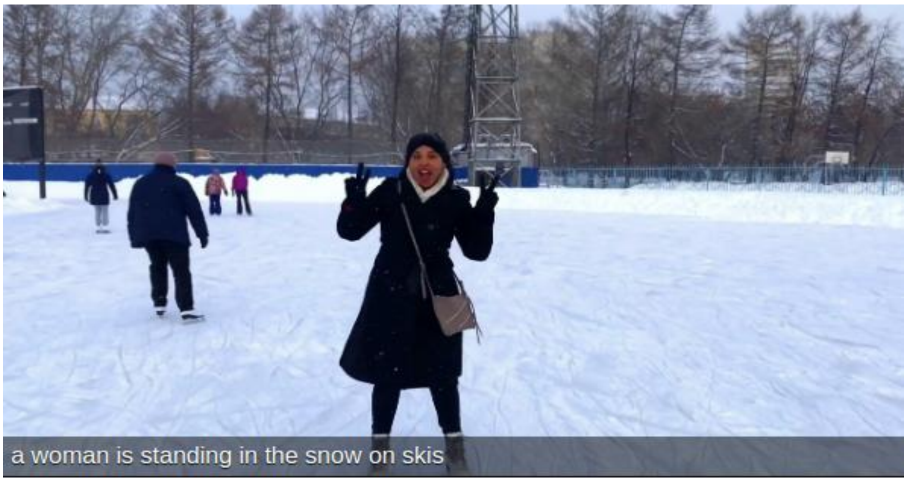

Fig. 4. Caption generated using NeuralTalk2 [8]

Two json files corresponding to the two models were obtained. These were then preprocessed necessarily to sequentially access the information about each frame in order to create semantic segments for the video.

Creating Tracklets. The second step of the application involves generating tracklets, for which slightly varying methods were used in accordance with the image captioning models used. The two methods are explained in detail below.

Densecap. To generate tracklets of a given video, the semantic meaning of the captions generated as well as the positions of the bounding boxes were taken into consideration.

The main idea to create tracklets using Densecap's json file was that given a frame, having N captions, N tracklets are begun each represented by these captions. A new frame is then compared with this frame, and if M captions are found to be similar, the unmatched N-M captions are registered as new tracklets, resulting in the total number of tracklets to be 2N-M.

A frame, as mentioned above, is associated with on an average of 75 captions, all sorted by the weights, as outputted by Densecap. For simplicity, each frame is represented by 5 captions with the highest weights and then the following procedure is carried out.

When a new frame is encountered, it is compared with the previous track's caption (because a new frame will form a part of the most recent track) and semantic similarity is calculated between the captions. Only if it crosses a particular threshold value, the current frame is added to the tracklet. Using this strategy, semantic meaning is taken care of, to form tracks. However, if the similarity measure does not cross the threshold, the tracklet retains the previous information and the next frame repeats the above mentioned procedure.

It is observed that when a tracklet is about to terminate, the semantic similarity with the forthcoming frames drops significantly. Hence, when the threshold value is not reached, a count is kept of the unmatched number of frames. Here, another threshold value is decided which takes care of the completion of a track. The count of unmatched frames represents discontinuity, and when this count reaches the pre-decided threshold, a tracklet is completed, representing a significant event in the video which is denoted by the caption of the first frame forming it.

Aside from semantic similarity, the bounding box coordinates generated by Densecap are also given weightage while forming tracklets. This is a necessary measure that cannot be ignored because Densecap's captions are generated based on the recognizable objects found in the bounding boxes. Similar to the procedure for similarity calculation of captions, every new frame's bounding boxes are compared with the bounding boxes of tracklet formed so far, and euclidean distance was chosen as the distance measure. Following the same thought process, a threshold is agreed upon and if the distance agrees with the threshold value, the new frame is added to the tracklet.

Although both these measures are important, first priority is given to caption similarity; the frames that cross the similarity threshold are then checked for bounding box closeness, hence completing tracklet formation by adopting both 'caption wise tracking' and 'tracking by detection' methods in a weighted manner.

NeuralTalk2. For the second approach to tracklet generation, the method is slightly altered and less elaborate than the above one, though the essence remains the same. This method concentrates on only 'caption wise tracking' as the image captioning method, used in this regard, outputs one caption per image and hence only semantic similarity of captions can be considered [4]. Employing similar techniques as the Densecap method, the first tracklet is represented by the caption of the first image and the rest of the frames are sequentially compared with the tracklet upto the previous frame. The criteria of inclusion of the current frame into the tracklet is the semantic similarity exceeding the agreed upon threshold. Moreover, the completion of tracklets is achieved by the number of frames calculated to be dissimilar to the tracklet caption exceeds the decided threshold, also referred to as cutting threshold.

The above approaches generate a set of tracklets each represented by the caption of the first frame it is formed by, and each tracklet storing information of the frames it constitutes. Hence, given an input video, it is processed upon by both of these methods and broken down into a set of tracks, each different from another forming the representative of different scenes in a video. Once these tracklets and their information was generated, the duration of every tracklet was calculated by mapping it to the input video duration and this data was stored in a file, which was then used for the next step, searching using a voice query.

Voice Based Query and Retrieval. Converting the input voice query to text using a speech recognition system, the query is compared with every representative caption of the final tracklets and outputs the tracks that are semantically relevant.

In these last two phases, creating tracklets and comparing queries, one of the major steps is comparing corresponding captions, deciding a particular threshold above which the captions are considered to be semantically similar or not. In order to make this work accurately, Google's Universal sentence-encoder [7] is employed. Using a word-based embedding approach and obtaining average of word embedding will most probably not represent the actual meaning of a sentence. Hence, a sentence encoder was chosen. Universal sentence-encoder [7] is a pre-trained sentence-encoder that is available on TensorFlow Hub for sentence embedding. The model is trained on a different set of tasks, supervised and unsupervised, and captures as much universal semantic information as possible to give an embedding of length 512. This is a simple Deep Averaging Network where input word embeddings are averaged together and passed through a multilayer perceptron deep neural network. Finally, after extracting the sentence vector, cosine similarity metric is used as the similarity measure. The advantage of using cosine similarity is that even if the two similar sentences are far apart by the Euclidean distance (because of the length of the sentence), chances are they might still be aligned closer together. Smaller the angle between the word embeddings, greater is the cosine value, hence higher similarity.

The retrieval of relevant clips was sped up by the use of parallel processing. After obtaining the final tracklets of the input video, semantic similarity was calculated with the representative caption of every tracklet on multiple processors simultaneously with the help of multiprocessing. All extracted tracklets were ranked according to the calculated cosine similarity measure, and presented to the user.

3.3 Audio Approach

The main idea for this approach was that it acts as an added approach to creating tracklets as audio might be important in certain situations and can provide vital information to tracklet formation. The clips generated as a result are formed mainly using semantic similarity as the base concept.

For this approach, we process the transcripts generated by YouTube for a particular video and first store all the transcripts with their respective start and end timings that would be used as fundamental information to create tracklets. The voice query is taken as input and the most relevant segments of the video are retrieved using an on-the-fly generation of tracks. For this, semantic similarity of each transcript is calculated with the input query. If it crosses a predetermined semantic-similarity threshold, i.e, a match is found, subsequent transcripts are again compared with the matched transcripts to continue the process of track formation.

Also, a count is kept of the unmatched number of tracklets and a cutting threshold of a predefined value is used after which when reached, that particular track terminates and the end time is stored. The algorithm continues to find other tracks relevant to the query in the entire video.

Now that we have the start and end timings of all the relevant tracks, merging of consecutive tracks is done if they are supposed to be continuous, based on predefined threshold, so that it ensures that a track is not cut into smaller tracks and presented.

Finally, the retrieved video segments are shown on the original video, so the user can either choose to continue or could navigate to before or after the specified video segment to properly understand the context of how the result is relevant to his query by moving back and forth.

4 Results and Evaluation

4.1 Specifics

This application of video retrieval requires a lot of computational power. NVIDIA CUDA enabled GPU was used for better computational abilities. (NVIDIA Graphics driver version 430.84, CUDA version 10.1, cudnn 7.6). GPU enabled Densecap and NeuralTalk2 are implemented in Torch.

A visually descriptive video was chosen as input for testing and evaluation. It was observed that the number of tracklets generated after pre-processing for Densecap was approximately 5 times more than that of NerualTalk2. Densecap is preferred for better localised query. Creating tracks using the captions generated by densecap was 5-6 times slower than using the ones generated by NeuralTalk2 since there were multiple captions per image that needed to be compared, hence multiple tracks initialized for one image.

The similarity threshold in constructing semantic tracklets varied from 0.6 to 0.8. Time taken to find similarity between two sentences was approximately 12 seconds. The cutting threshold was set to 5 frames, and the minimum track size was also set to 5 frames, i.e. only tracks with length greater than or equal to 5 frames were retained as valid semantic tracks. For the given input query, a set of tracks with semantic similarity of representative caption and the query higher than the threshold value was proposed by the application. Table 1 represents the retrieval rates with and without parallel processing and the speedup achieved.

Table 1. Retrieval Rates and speedup

Approach	Without Parallel Processing	With Parallel Processing	Speedup
NeuralTalk2	25min	22s	68.18
Densecap	4hrs	1 min 40s	144

The specifics for the audio approach were along the same lines as that of the video approach. The semantic similarity threshold was set to 0.6 and the consecutive transcript similarity threshold was set to 0.5. The cutting threshold was set to 2 transcripts.

4.2 Evaluation

In video retrieval applications, relevance of the retrieved videos according to the input query decide the accuracy of the system. The performance of such models can be judged only by the users who give the input query as the nature of this problem is purely subjective. To quantify this measure of accuracy, 40 users were asked to give a query of their choice after watching the video. Each output clip from both the video approaches were flagged as relevant / non-relevant by the user who gave the corresponding query.

We have used two measures of accuracy namely, MAP and MRR. Mean Average Precision (MAP) calculates the mean of average precision for each individual query as shown in (1). Average Precision collectively measures the relevance of all the ranked retrieved videos.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{1}$$

On the other hand, Mean Reciprocal Rank (MRR) gives the relative score of the first relevant item according to its rank. As shown in (2) rank_i represents the rank of the first relevant result of the i-th query, averaged over all the queries, Q.

$$\text{MRR} = \frac{1}{Q}\sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{2}$$

Table 2 represents the metric scores of MRR and MAP for both the approaches.

Table 2. Accuracy Scores

Approach	MRR	MAP
NeuralTalk2	62.77%	60.66%
Densecap	57.33%	54.00%

Both image captioning (NeuralTalk2) and Densecap work appreciably well. NeuralTalk2 is a good measure when the aim is analyzing the whole scene whereas the object details are clearer in Densecap. Hence, Densecap proves better for localised queries. For instance, 'Russian Flag' had 0 outputs for NeuralTalk2 but Densecap gave the relevant clip. However, there can be repeated scenes in Densecap since it generates many captions for an image that may vary slightly but on a whole keep the meaning similar. Creating tracks from Densecap is more computationally exhaustive due to a greater number of captions per image and the added distance computation of the bounding boxes. Considering the accuracy measures, NeuralTalk2 certainly works better than Densecap and is also faster for preprocessing and retrieval, the only trade-off being localization of query.

The audio approach, built considering specific domains such as University lecture videos and retrieving clips corresponding to the given query was computationally expensive as every transcript had to be compared with the input query, and retrieval rate for a particular query was observed to be 24 hours for a video with 868 transcripts.

4.3 Examples

In order to test the application built on the image captioning, i.e. NeuralTalk2 approach, a video was taken and this model was performed on [9] as input as it had quite a few descriptive scenes and seemed appropriate for the project. A voice query of "a woman is skiing" was given, on which a search was performed, and [10], [11] and [12] were generated as outputs by the model.

Similarly, to test the application built on the Densecap approach, a video was taken and this model was performed on the same video as NeuralTalk2 input, for easy comparison and monitoring performance i.e. [8]. A voice query of "a woman is skiing" was given, on which a search was performed, and [13] and [14] were generated as outputs by the model.

On similar lines, to assess the accuracy and efficiency of the audio approach, the model was performed on a lecture video of SVM by Patrick Winston, MIT Professor i.e. [15]. A voice query of "constraints" was given to obtain all the scenes where the professor explains the constraints of the SVM derivation. Eight clips were generated and their start and end time being (462s, 548s), (831s, 876s), (1045s, 1055s), (1134s, 1149s), (1355s, 1604s), (1849s, 1881s), (2058s, 2115s), (2313s, 2380s).

5 Conclusion

In this paper, we accomplish a system that uses a state of art deep learning approach for retrieval of semantically relevant video segments given a query. We used two broad techniques, an image captioning approach and an audio approach. The image captioning approach mainly consists of two different image captioning techniques: NeuralTalk2 (one image caption for an image) and Densecap (multiple captions corresponding to specific regions in the image) for extracting information about the video content. The video was preprocessed using this information to create tracklets which is a spatio-temporal representation of the video's content. The system has a natural language voice-based interface for querying which is user friendly and is more adaptable. Both NeuralTalk2 and Densecap work well. Image captioning is good when one wants to analyze the whole scene whereas the object details are clearer in Densecap. The audio approach on the other hand uses an on-the-fly generation technique, which on given the voice query generates semantically similar tracklets after processing of the transcripts. But the cons of this approach are that as semantic similarity is the base for forming tracklets, results might not be accurate as other factors in the time domain and frequency domain also need to be considered. This would be useful in domains where the audio is more important than the video like University lecture videos.

Acknowledgments

This work was done in part or full under the Intel Corporation Student Contest 2020 in association with the Center for Innovation and Entrepreneurship (CIE) at PES University. We acknowledge all the support from Intel Corporation towards this work.

References

1. Kucuktunc, O., Gudukbay, U. and Ulusoy, O., 2007. A natural language-based interface for querying a video database. IEEE MultiMedia, 14(1), pp.83-89.
2. Erozel, G., Cicekli, N.K. and Cicekli, I., 2008. Natural language querying for video databases. Information Sciences, 178(12), pp.2534-2552.
3. Johnson, J., Karpathy, A. and Fei-Fei, L., 2016. Densecap: Fully convolutional localization networks for dense captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4565-4574)
4. Karpathy, A. and Fei-Fei, L., 2015. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3128-3137)
5. Li, S., Xiao, T., Li, H., Zhou, B., Yue, D. and Wang, X., 2017. Person search with natural language description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1970-1979)..
6. Sangkuk Lee, Daesik Kim, Myunggi Lee, Jihye Hwang, and Nojun Kwak. 2016. Where to Play: Retrieval of Video Segments using Natural-Language Queries. In Proceedings of 20, 2017, April (UNDER REVIEW IN ACM MM), 8 pages. DOI: 10.475/123 4
7. Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., ... & Sung, Y. H. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.
8. Karpathy, A., Neuraltalk2 image captioning,Github repository, Available: .https://github.com/karpathy/neuraltalk2
9. Fabina Travels (2020) Russia: 1 Minute Travel Vlog. In: YouTube. https://youtu.be/rnQLGLvlSIQ. Accessed 13 May 2020.
10. Kritika Kapoor (2020) NeuralTalk2-results-1. In: YouTube. https://youtu.be/HuUamlD-neSg. Accessed 13 May 2020.
11. Kritika Kapoor (2020) NeuralTalk2-results-2. In: YouTube. https://youtu.be/HigfoDnZsbY. Accessed 13 May 2020.
12. Kritika Kapoor (2020) NeuralTalk2-results-3. In: YouTube. https://youtu.be/t-U2jXjfzik. Accessed 13 May 2020.
13. Kritika Kapoor (2020) Densecap-results-1. In: YouTube. https://youtu.be/g14sERx98qw. Accessed 13 May 2020.
14. Kritika Kapoor (2020) Densecap-results-2. In: YouTube. https://youtu.be/lq9eHwS6Uq4. Accessed 13 May 2020.
15. MIT OpenCourseWare (2014) 16. Learning: Support Vector Machines. In: YouTube. https://youtu.be/_PwhiWxHK8o. Accessed 13 May 2020.
16. Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164)
17. E. Bochinski, V. Eiselein and T. Sikora, "High-Speed tracking-by-detection without using image information," 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, 2017, pp. 1-6, doi: 10.1109/AVSS.2017.8078516.
18. TechGig News, https://content.techgig.com/flipkart-partners-with-iit-patna-for-machine-translation-research/articleshow/77694928.cms, last accessed 2020/08/25.