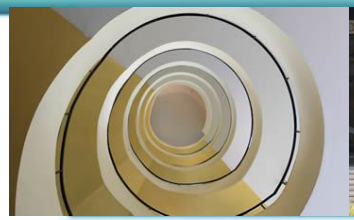
Final Mini Project Demonstration

Project Title	:Searching a Video Database using Natural Language Queries
Project ID	:MPW20MHR01
Project Guide	:Dr Mamatha H.R
Project Team	:Shubha M (PES1201701540), Shrutiya M (PES1201700160), Kritika Kapoor (PES1201701868)






Project Abstract and Scope

SEARCHING A VIDEO DATABASE:

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 is built.

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)

Literature Survey

Several studies related to this have been published. Many video data models are concept based and have an object oriented approach.

1. **BilVideo**
   - BilVideo 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.

Literature Survey(contd)

2. Natural language querying for video databases, Information Sciences

- Another similar system 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.

But on the other hand, using natural language to describe an image using deep neural networks provides a good language model to extract information from the images.

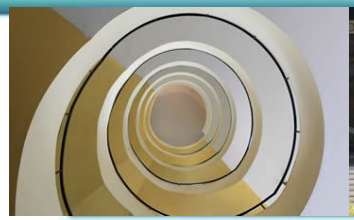
Literature Survey(contd)

3. Person search with natural language description

- Another related project tackles the problem of searching a person in huge image databases using natural language based description.
- 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.

4. Where to Play: Retrieval of Video Segments using Natural-Language Queries

- Another model uses Densecap to generate multiple captions per image and conducts a tracking by caption to retrieve video segments.
- Densecap is a model whose architecture consists of a CNN, a dense localization layer, and an RNN language model that produces the labels.
- It uses Skip-thoughts vector for sentential encoding and for performing semantic similarity.






Design Approach

Our design approach consists of the procedure using two approaches, namely being, Densecap and NeuralTalk2. On the whole, the model that we have executed consists mainly of 3 sequential parts:

i. Generate Captions
ii. Creating Tracklets
iii. Voice based Search.

1. **Generating Captions**
   - This is the first step towards building this system. The video clip is split into frames.
   - 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.

Design Approach

2. Creating Tracklets

The creation of tracklets vary for the NeuralTalk2 and Densecap approach. Both the approaches have been described in detail below.

a. NeuralTalk2 -

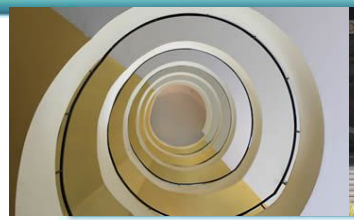
- The first tracklet is represented by the caption of the first image and the rest of the frames are sequentially compared with the tracklet upto the previous frame.
- The criteria of inclusion of the current frame into the tracklet is the semantic similarity exceeding the agreed upon threshold.
- Moreover, 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.

b. Densecap -

- The **Densecap model**, which 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, was developed with the ideas of both object detection and caption generation and given an image as input.

Design Approach

- This model generates on an average of 80 captions per image, with corresponding weights, and bounding boxes of the objects all dumped into a json file.
- 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, i.e a snapshot image of the video, is associated with on an average of 75 captions, sorted by weights and the first five highest weights and hence the most significant captions for these frames are used.
- Similar to the previous strategy, every new frame will be compared with the previous track's caption and semantic similarity will be calculated. If it crosses a threshold, the new frame will be added to this tracklet. The idea of the cutting threshold mentioned in the Neuraltalk2 approach is utilized here too.






Design Approach

- After semantic similarity, the bounding box coordinates outputted by Densecap are taken into consideration the data is analyzed by considering the euclidean distance of the bounding boxes of two frames to decide how similar two frames are, to be added into the same tracklet.
- 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.

Design Approach

3. **Voice based Query**

- 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 order to make this work accurately, Google's Universal sentence-encoder is employed. Using a word based embedding approach and obtaining average of word embedding will mostly not represent the actual meaning of a sentence. Hence, a sentence encoder was chosen. Universal sentence-encoder is a pre-trained sentence-encoder that uses tf hub for sentence embedding.
- Finally, after extracting the sentence vector, cosine similarity metric is used as the similarity measure.

Design Approach

Benefits of the approaches used:

- Most of the video data models built so far are usually **content and rule based**, wherein 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, and which require certain level of annotation.
- 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. This information is used for querying which makes our model **domain-independent** and serves general purpose.

Drawbacks:

1. Neuraltalk2

There is lesser localization as the whole track is formed on the basis of a caption. The spatial relations between the objects in the frames are not given importance.

2. Densecap

Tracklets formed can be extremely small.

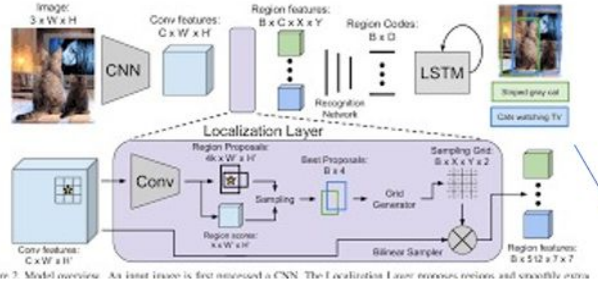
Design Constraints, Assumptions & Dependencies

1. The application of video retrieval requires a lot of computational power. **NVIDIA CUDA** enabled GPU was used for better computational abilities. **GPU** enabled DenseCap and NeuralTalk2 implemented in Torch are used to generate the frames.
2. The program to create tracklets was run on **Google Colab**. However just running 10 frames took about 4 hours. And the video we had had 1950 frames. We found a hack to enable 35gb ram TPU. We then decided to divide the entire task into smaller tasks to get intermediate tracks and then later merge them. However, it was still not very feasible with the resources we had.
3. We needed computational resources better than google colab and we have exhausted all our options. Colab is ineffective as it continuously keeps disconnecting. Due to limited resources, the model was tested using a one minute video
4. Right now we are just running commands on Colab because it is computationally very expensive and the results are shown there.

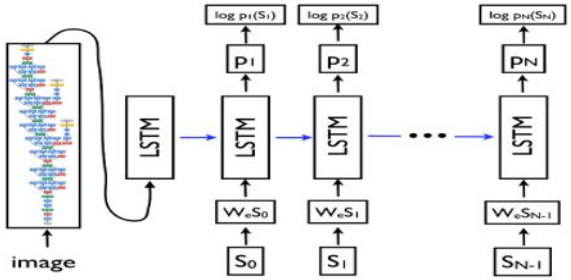
Design Description/UI Design

Model:

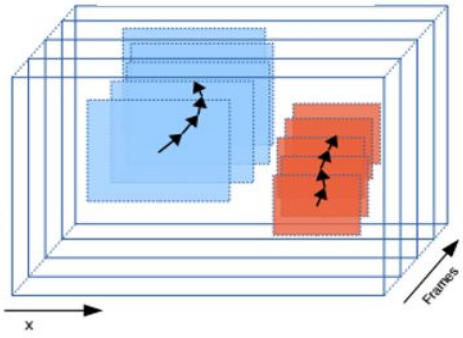
DENSECAP



NEURALTALK2



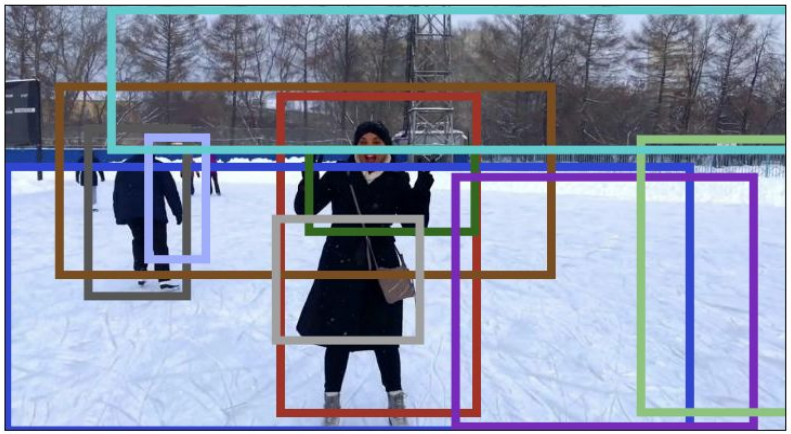
CREATE TRACKLETS



VOICE BASED QUERY

Design Description/UI Design

Image Captioning Model Results:



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.

Captions generated by Densecap

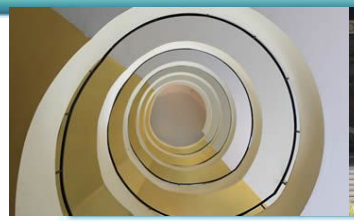
Captions generated by NeuralTalk2

Technologies Used

1) Speech to text API
2) DenseCap: Fully Convolutional Localization Networks for Dense Captioning and NeuralTalk2 that generates a single appropriate caption for the entire image (both use torch/torch7, torch/nn, torch/nngraph, torch/image, lua-cjson, qassemoquab/stnbhwd, jcjohnson/torch-rnn,torch/cutorch and torch/cunn)
3) NVIDIA CUDA based GPU for computational capabilities.
4) For **Semantic similarity**,Universal Sentence Encoder (by Google) which uses a deep averaging network encoder (DAN) for converting the sentences into 512 dimensional vectors is used
5) The program to create tracklets was run on **Google Colab**.

Project Progress So far

1. We are using an **image captioning approach** to solve this problem.
2. 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.
3. These captions are used to preprocess the video and create semantically similar **tracklets**.
4. For **Semantic similarity**, 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.
5. 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.






Project Progress So far

1. The results are put up in the following slides
2. Since the objective of the project is satisfied and two different methods have been explored for the same, it is completed satisfactorily.

Project Demo

- Due to limited resources, the model was tested using a one minute video. The video was then split in frames using a particular frame rate with help of opencv library. The number of frames generated was 519 followed by the creation of semantic tracklets.

- 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. It took around 25 minutes to process a query. Padding was added to the retrieved videos less than one second.

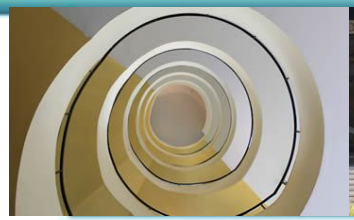
NeuralTalk2 Approach

In order to test the application, a video was taken and this model was performed on :
https://youtu.be/rnQLGLvlSIQ
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
https://youtu.be/HuUamlDneSg
https://youtu.be/HigfoDnZsbY
https://youtu.be/t-U2jXjfzik
were generated as outputs by the model.
We have to optimize the padding part which will lead to better results.

Densecap Approach

In order to test the application, a video was taken and this model was performed on the same video as NeuralTalk2 input, for easy comparison and monitoring performance, i.e:
https://youtu.be/rnQLGLvlSIQ
A voice query of "**a woman is skiing**" was given, on which a search was performed, and
https://youtu.be/g14sERx98qw
https://youtu.be/lq9eHwS6Uq4
were generated as outputs by the model.
We have to optimize the padding part which will lead to better results.

PES UNIVERSITY






Thank You