MotoSum: A Video Summarization Experiment

Luis Vitor Zerkowski[1], Flavio Soares[2], Luc Van Gool[3]

[1]University of São Paulo, São Paulo

[2]Advisor from University of São Paulo, São Paulo

[3]Advisor from ETH Zürich, Zürich

Introduction

In a world totally immersed in digital media where almost everyone has unrestricted access to cameras and filming devices, it can be very difficult to keep up with the core of the content one consumes online. In the motorcycle world, this problem can be even worse, since it is common for riders to record very long videos of their rides. By automatically producing diverse, representative, and generally reasonable summaries, the video summarization pipeline attempts to address this problem.

Another contribution of this paper is the MotoSum benchmark, a dataset of motorcycle ride videos with frames labeled by our own team based on the TVSum [1] and SumMe [2] datasets approaches. Finally, it is worth noting that the annotation tool has been developed in such a way as to easily adapt it for use on the Amazon Mechanical Turk crowdsourcing platform.

Discussions and Results

To account for the project proposal, some summarization pipelines were developed.

Uniform Sampling: A Reference Model

To this end, a fairly simple first prototype of sampling the original video was developed. In this prototype, the strategy was basically to split a video of length L into 100 segments of size $\frac{L}{100}$, and to compile 15 of these segments - 15% of the original video - into a final summary. The segments, finally, were chosen at an uniformly spaced interval of length $\frac{L}{15-1}$. For every frame of the video chosen from the sampling, its entire segment was selected. The idea of this kind of selection is precisely to guarantee that the length L' of the summary will be 15% of the size L of the original video, and that segments from various parts of the video will be selected.

Pretrained DSNet: The First Agent

The DSNet is a supervised neural network specialized in the task of video summarization, and it was trained on a dataset called TVSum. The flowchart in Figure 1 schematizes how the network works.

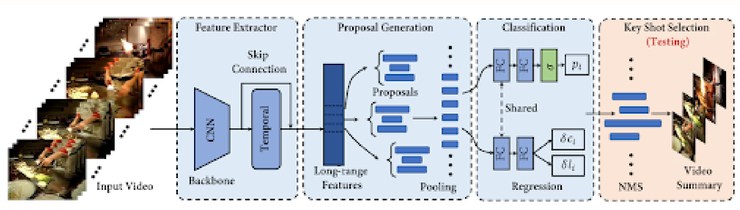
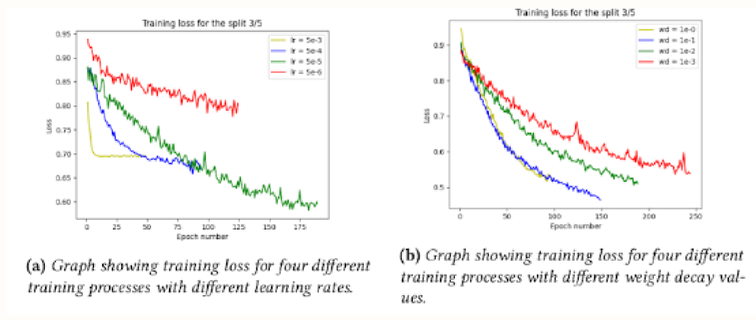


Figure 1: Schematic illustrating the architecture of the Anchor Based DSNet network [3].

Retrained DSNet: A Better Agent

Having a first functional agent in hand and already trained on a database of various videos, the next experiment naturally moves in the direction of a new training process. This is because we can use an already implemented robust architecture to train an agent with a new database, *MotoSum*, a domain-specific database of motorcycle rides developed for the project.

For the network optimization we use only the Adam optimizer and for the network initialization we used Xavier initialization. For the training itself, we tested two hyperparameters, as shown by the graphs in Figure 2. The first of these, given by (a), was the learning rate, and the second one, given by (b), was the weight decay.

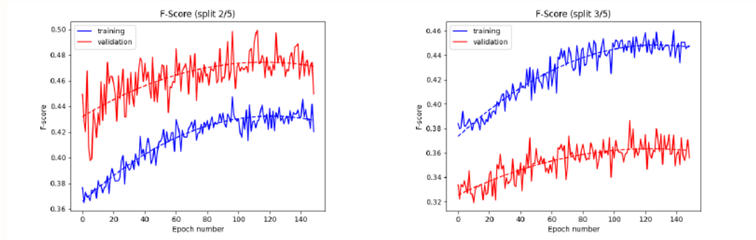


(a) *Graph showing training loss for four different training processes with different learning rates.*

(b) *Graph showing training loss for four different training processes with different weight decay values.*

Figure 2

Despite the success of training the model on the third split, it cannot be said that the same was true for all other splits. For three of the five splits, the f-score tracking metric for the validation set was significantly higher than the training f-score, as shown in the graphs (a) and (b) in Figure 3 comparing the model f-scores on the second and third splits. This kind of phenomenon is a reflection of a small dataset with very specific data, two weaknesses of our database and typical for the video summarization task.



(a) *Graph showing f-score for validation and training sets during training process of the second split.*

(b) *Graph showing f-score for validation and training sets during training process of the third split.*

Figure 3

DSNet + ERFNet: A New Agent

Despite how well our previous agent works, we can build even better versions of it. This is because our feature extraction network is still too generic, being able to deal with all kinds of videos. It is not necessary, however, to handle such a diverse domain, since our agent should only summarize egocentric videos in the motorcycle ride environment.

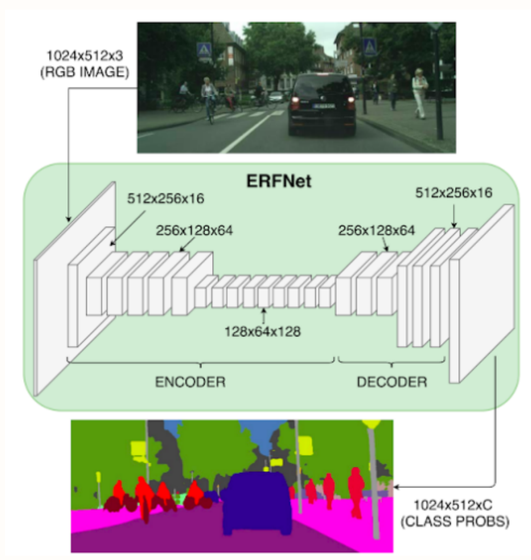


Figure 4: Schematic illustrating the architecture of the ERFNet. [4]

Results

Table 1 shows the agents' performances in terms of diversity, representativeness, and image quality score. It is worth noting that these values were calculated considering the average of each score among all the generated summary videos - compared to their respective original videos from the dataset in the case of diversity and representativeness. In addition, we use a sampling rate of 15 in the videos, optimizing the computations without any major quality tradeoff, since the videos are all 30 FPS.

Model	Diverse	Representative	BRISQUE	NIQE
Reference	0.67951	0.33452	52.71482	11.94220
Pre-trained	0.71394	0.53834	52.98820	12.03526
Retrained	0.71358	0.51296	51.87517	12.07534

Table 1: Scores of each model.

Bonus Agents: Query Filtering and Stat-Based Filtering

The idea of the first agent is to incorporate more subjectivity to the model, leaving it up to the user to decide which are the most relevant points of the video. In this way, the model receives a video and a title provided by the client himself, and tries to computationally generate the best summary of the original video considering the textual highlight under consideration. To achieve the model's proposal, it is necessary for us to use an embedded semantic space in which we project both words and images. In a shared features space, it is possible to perform similarity analysis betweenwords and frames, allowing a filtering of the original video before using the summarization model. Beyond all experimentation based on machine learning and, more specifically, on neural networks, one can also build agents based on feature filtering. In the motorcycle racing environment, features such as speed, leaning angle of the bike, and even external predicates - a certain location on the map, or characteristics of the roads driven through, for example - are excellent references to summarize a video.

Conclusions and Next Steps

The focus of this project was to give a broad overview of video summarization, introducing the field and presenting several ways to approach its problems. Also part of the plan is a pedagogical journey through several areas of computer science that contribute to a better understanding of the summarization task. Building the knowledge from its classical basis to the use of state-of-the-art methods, this project is not only the implementation of several experiments, but also the development of a product effectively deployed in a company's application.

As for the project itself, there is still much to be done. Testing of new summarization models, especially of an unsupervised approach, using a lighter semantic segmentation network to improve diversity on the motorcycle rides environment, improvements in the query filtering branch, making the semantic image-language space much more robust, well integrated and fast to query, new filters based on ride data, and much more. For now, let this article serve as a technical and descriptive reference of what has been done so far to guide the next initiatives.

References

[1] Song et al., 2015 *Tvsum: Summarizing web videos using titles*

[2] Gygli et al., 2014 *Creating summaries from user videos*

[3] Zh et al., 2020 *DSNet: A Flexible Detect-to-Summarize Network for Video Summarization*

[4] Romera et al., 2017 *Erfnet: efficient residual factorized convnet for real-time semantic seg- mentation*