Project 4: Video Search

Marc Beillevaire, Mateusz Buda, Ted Cassirer, Laura Jacquemod, Christoph Kaiser

Search Engines and Information Retrieval, Department of Computer Science
KTH, Sweden

Abstract

Because digital information is created at a very fast pace, it has become necessary to develop tools for efficient search. This is particularly true with videos, which gather an important amount of information. Whereas most search engines focus on the metadata around the video: title, description, etc., we propose a novel approach, segmenting the video in order to search on its visual content. We use an algorithm to segment videos as well as a search engine in order to provide the best videos to a query. After comparing different ways of segmenting, retrieval techniques and ranking measures, a final system was found that performs quite well, seeing that the precision at 3 is nearly of 0.7.

Keywords: Visual Information Retrieval, Artificial Intelligence, Ranking Measures

1. Introduction

Visual searching has become a widely researched topic in the recent years. However, the commercial search engines that currently exist do not allow users to search visually into a large part of the visual content, specifically videos. Many different use cases might appear: a company might want to find all appearances of its logo or a student might want to find the segment in a lecture video where a particular slide is presented. Also a user can simply be looking for a particular advertising without remembering its brand. All of these problems can not be solved using today's search engines whereas this problem is extending, seeing that the amount of digital information is constantly increasing. This can be shown by the popular platform ”YouTube”, which sees around 300 hours of videos uploaded every minute [7].

Hence, a search engine allowing users to search for videos through content-based visual queries (CBVQ) is becoming essential. It is obviously important that its precision and recall, speed and ease to use need to be as good as the current search engines.

In this paper, we start off by an overview of some related work in the following section 2. Then we focus on what data we gathered and which methods we use for a successful search implementation in section 3. After explaining the results we got from our experimental setup in section 4, we assess the correctness of our search engine in section 5 and further more discuss possible future modifications or further improvements in section 6.

2. Related work

2.1. Neural description of videos

As explained by M. Flickner [12], humans can easily distinguish the content of an image and its perceptual organization whereas it is more laborious for computers which are better at extracting semantic descriptions. Because of this, automatic object recognition has been an important subject of research for the last two decades. This has become even more true with the outbreak of autonomous robots and cars which need to estimate surrounding objects in real time. In this case, frames of videos are analyzed, as in this paper.

The initial research includes the work done by D. G. Lowe [11] in order to introduce a new class of local image features. Thanks to these different features, a robust recognition could be achieved in less than 2 seconds.

More recent and more advanced research on this subject have succeeded in providing quicker and more robust solutions, which are hence more useful in the case of videos. One recent method has been developed by Vinyals et al. from Google [17] and by Karpathy and Fei-Fei from Stanford [9]. They introduces a Neural Talk framework that uses long short-term memory (LSTM) recurrent neural network as a language model to produce human like descriptions from image representation obtained with convolutional neural network.

2.2. *Search engines*

CBVQ research on video databases has emerged in the end of the 20th century with the introduction of the Query By Image and Video Content (QBIC) system [12] and VideoQ [15] but it has not been fully explored yet. In fact, we are, to our knowledge, the first ones using automatic images captioning in videos in order to answer visual queries.

Past research that has been done in this field has focused on low-level visual features that are quantifiable properties such as shape, color, texture or motion. In order to use the search Engine VideoQ [15], the user has to, through its interface, draw the object with the right color, shape, size and motion. Even if it performs well with skiers for example, it is not easy enough to use to become a daily tool, as required. As more research and progress was done in this field, tools such as the one explained by A. Araujo et. al. [1] or Video Google [8] have been created. In both cases, the input that was required from the user in order to use the search engine was an image or a subpart of an image. Even if this search can be very useful in some cases, it does not cover most of the use cases, we presented. This is why we decided to use text queries describing visual aspects of a video in this paper - in order to meet most use cases by providing an easy and efficient search engine on videos.

3. Methods

First we explain how we chose the correct data to make our experiments in section 3.1. Then, we focus on finding a good way of recognizing objects in video frames using deep learning in 3.2. Section 3.3 presents a method of extending the previously generated descriptions. Next an overview of the additionally used metadata is given in section 3.4. Finally, in section 3.5, we compare different retrieval methods and weightings in order to provide the users a search engine that gives correct videos according to their query.

3.1. *Data*

The first and obvious step for such a project is to find the right videos. Within the scope of this project the real origin of a video does not really matter. This is the case because the project, as already stated in the introduction, mainly researches the possibilities for a search based on Visual Information Retrieval.

In today's use of video platforms, users have a high possibility of responding in some form to what they see. This is often implemented as comments or votes. Because such data can also be very useful, here presented Search Engine takes additional metadata like title and description, but also user influenced data like up-/down-votes and views into account.

To form an adequate testset, 1000 videos were selected at random. As a matter of time these videos and all the above mentioned information have been gathered from the currently largest video-database and platform YouTube, which provides access to roughly estimated over half a billion hours of videos [7].

Our dataset comes from academic use only videos crawled using YouTube API [5]. The crawling took place in 2007 and 2008 so we only used raw links and discarded all metadata that we retrieved on our own to make them more up-to-date. The crawling strategy was to start with the initial set of videos from the list of "Recently Featured", "Most Viewed", "Top Rated" and "Most Discussed", for "Today", "This Week", "This Month" and "All Time". Those videos are of depth 0. Then for each of those videos, the next depth to be fetched comes from the related videos list. We used the data that was crawled in 27 July 2008 of depth 3 to make them vary in topic and type i.e. song videos, documentary film, advertisements, sport events, etc. It was first filtered to discard videos that were removed or with restricted access. To make the evaluation easier we selected only English videos by checking the language of the title. From the processed YouTube video links we randomly selected 1000 video links. The final set of videos also varies in length from about 30 seconds to over 1 hour.

3.2. *Video to text mapping*

To automatically generate visual content description for video frames we used the Neural Talk framework as proposed by Vinyals et al. from Google [17] and by Karpathy and Fei-Fei from Stanford [9], which is described in section 2.1.

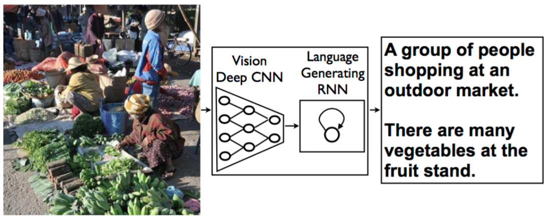
Vision
Deep CNN
Language
Generating
RNN
A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

Figure 1: End-to-end neural network based model consisting of a vision CNN followed by a language generating RNN. It generates complete sentences in natural language from an input image, as shown on the example above. (Figure from [17])

For the language model we used an LSTM trained on MSCOCO with 512 hidden units as in [17]. It is fed with an image representation consisting of 4096 dimensional feature vector. Models that meet this criteria include VGG CNN models. We used VGG_CNN_S model described in [4] that achieved 13.1% top-5 error on ILSVRC-2012 validation set [3] and VGG_ILSVRC_16 from [16] that achieved 7.5% on the same task. For both of them we have chosen the representation from the last but one fully connected layer. In the figures 2 and 3 we present example descriptions generated with the models.

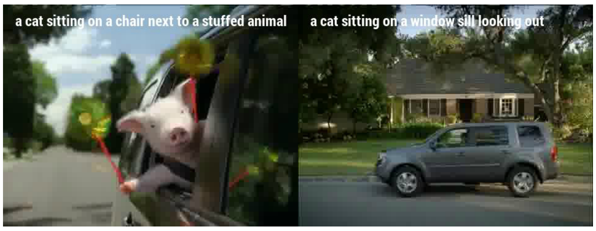
a cat sitting on a chair next to a stuffed animal
a cat sitting on a window sill looking out

Figure 2: Examples of video frames descriptions generated with image representation extracted from VGG_CNN_S model. "A cat sitting on a chair next to a stuffed animal. A cat sitting on a window sill looking out."

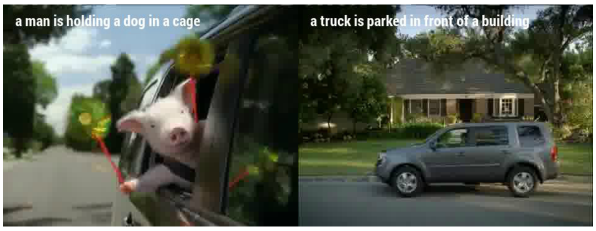
a man is holding a dog in a cage
a truck is parked in front of a building

Figure 3: Examples of video frames descriptions generated with image representation extracted from VGG_ILSVRC_16 model. "A men is holding a dog in a cage. A truck is parked in front of a building."

3.3. Text expansion

The Neural Talk framework aims to generate human-like descriptions and this is not exactly what we are most interested in when building our search engine. We want to have as much relevant text information as we can get. To expand the text obtained from Neural Talk we first extract nouns from it using part-of-speech tagger from Natural Language Toolkit [10]. Then for each of them we take synonyms from WordNet [13]. WordNet is a large lexical database of English that has hierarchical structure. It uses a concept of hypernyms and hyponyms. Hypernyms of a given word are its parents in a hierarchy. They are more general and more abstract. We take hypernyms of depth higher that 5 to discard words with low informativeness. Hyponyms of a given word are children from the very next level in the WordNet hierarchy. They are words that are more specific.

3.4. Gathering of Metadata

Another source of text data related to a video is the metadata. It consists of data provided by the user uploading the video as well as the ones looking at the video. As already mentioned in section 3.1, it is very common for today's platform to track this kind of information. Therefore we found it useful to include this already easy accessible data into our search engine even though it has not necessary something todo with Visual Information Retrieval.

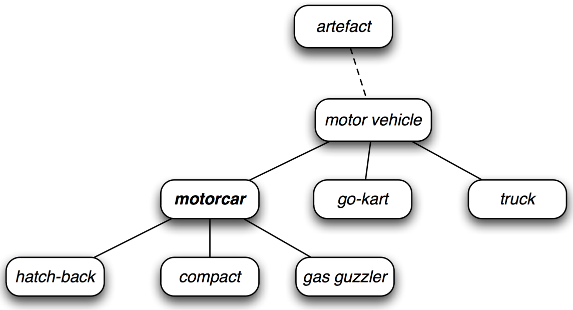

Figure 4: Fragment of WordNet Concept Hierarchy: edges indicate the hypernym/hyponym relation. (Figure from [10])

Gathered metadata contains the following information:

- title
- description
- category
- numerical statistics consisting of:
  - views count, v
  - up-votes count, u
  - down-votes count, d
  - adds to favorites playlist count, F
  - comments count, c

Because of our choice of YouTube as a video provider, these values are already tracked and accessible for each video. Important to mention though is that statistics about up-votes, down-votes and/or comments are might not available. This is because video uploaders can deactivate either the platforms voting system or comments or both.

3.5. Search engine

For all of the following retrieval methods, we first got the inverted index as explained by

C.D. Manning in [2] where, instead of saving the offset of a token in a text, we were saving its position in the video (number of frame).

Finally, the output of our search engine is an ordered list of links to the videos corresponding to the query. Not only do we return the link to the video, but also the best position in the video. For example, if the video is two-hours long, it is not convenient for the user to go through the whole video in order to find a match with the query. Therefore, we calculate the "best" position by searching for the frame with the more close neighbors that also answer to the query.

3.5.1. Boolean retrieval

The first retrieval method that was programmed on our search engine was a Boolean retrieval. In this way, our search engine gave results to multiword queries, phrase queries ie. looking for all the query terms in only one frame as well as close queries ie. searching of query terms in frames that were close to each other (with a threshold that had to be defined by the user).

In that case, even if no ranking was considered, the output was still ordered. In fact, the probability of getting a correct description from the Neural Talk framework is given by that same framework and we ordered the results in descending order of probability.

3.5.2. Ranked retrieval

Three different rankings were performed by our search engines and compared in order to find the most robust one.

- Tf-idf ranking: as explained by C. D. Manning [2, Chapter 6], we weight each term according to its frequency and the inverse document frequency. For each term, we therefore calculate:

  $tf - idf_{t,d} = tf_{t,d} * log(\frac{N}{df_t})$ with $tf_{t,d}$ the number of occurrences of this term in the document, N the total number of documents and df_t the number of documents with the term t. With this ranking, less frequent terms would weight more than frequent ones.

- Popularity ranking: in the same way as Google's PageRank, we ranked videos by their popularity on their website (in our case, on Youtube). We used the empirically found formula:

  $$P = \alpha\sqrt{\frac{10 * F + u - d}{10}} + \beta\sqrt[3]{v} + \gamma\sqrt{\frac{c}{10}} \quad (1)$$

  with the data from numerical statistics as introduced in section 3.4. Different values of α, β and γ were chosen in order to find when the best results were found.

- A combination of both previous types of ranking. The appropriate weight for each ranking will be found experimentally in order to get the best results possible.

4. Results

In order to choose the best parameters for our search engine and to assess if it was good or not, we did some measurements with the three following different queries:

- Q1: Large airplane
- Q2: Playing with a dog
- Q3: Man on a bench

We considered that the number of relevant videos in our dataset was of 10 for each query.

4.1. Selection of the retrieval model

4.1.1. Boolean retrieval

The first tests were done in order to find the best choices for Boolean retrieval:

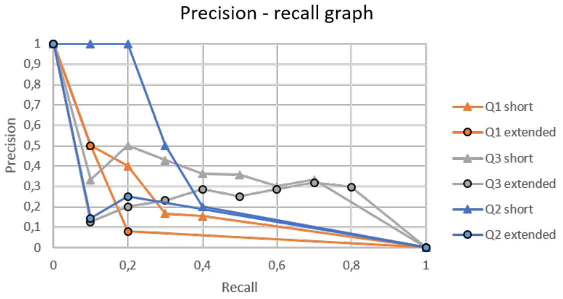

Figure 5: Precision - recall graph using the Boolean retrieval method to show normal and extended index.

- Normal index or with text expansion
- Search in close frames or all frames

Figure 5 summarizes the tests that were done with the three queries with both normal index and the extended one using text expansion as described in section 3.3. Seeing that the area under the curve is bigger with the normal index for the three queries, we can conclude that, with Boolean retrieval, it is better to use the normal index rather than its extended version.

By looking at close frames only, or more specifically at phrase queries, ie. only in one frame, obviously less results were obtained but their quality was also reduced. For example, with Q2, only 10 results were obtained with a precision of 0.1.

4.1.2. Ranked retrieval

More tests needed to be done to get the best search engine with ranked retrieval:

- Find the threshold probability
- Normal index or with text expansion
- Use metadata or not
- Optimize search or not
- Parameters for ranked retrieval

First, we decided to keep the frames' description if their probability of being true was over 10^{-6}. In this case, enough results were returned but it was still specific.

Then, we wanted to assess the importance of metadata. In this way, figure 6 shows different weights ω for the tf-idf with Q2. Not only did we find out that a higher weight was preferred (we chose for later 10) but also that the extended index was needed. In fact, words such as ”dog” or ”playing” never occurred in the initial descriptions so Q2 was the same as ”with a” which does not make sense. Hence, extended description need to be used for ranked retrieval.

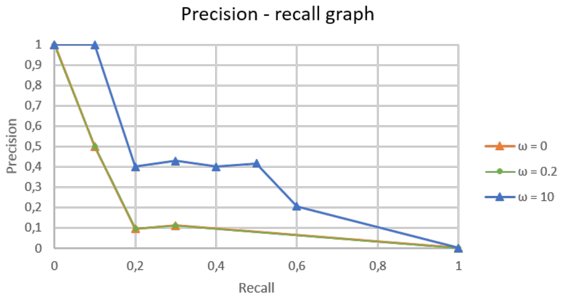

Figure 6: Precision - recall graph using the ranked retrieval method to different weights for metadata.

Concerning the optimization of the search, we decided not to use it in our final search engine. We implemented the optimization by index elimination, as explained by C. D. Manning in [2, Chapter 7]. In fact, as the following table shows with the example of Q2, the optimization can be considered as enough (time decreased by 2 ms) only when a high threshold is required. However, this threshold is highly query-dependent and could lead to very few results if it was used with another query.

Using Q1, we tried two different values for α, β and ω for the equation 1 and our results are shown in Figure 7. We only tried both sets of values as they made sense: giving each part of the formula the same weight or giving more to the number of likes and views than to the

Optimizing	**No**	**t = 1**	**t = 1.5**	**t = 2**
Time (in μs)	3 404	3 079	3 185	780
Nb of docs	624	624	231	14

Table 1: Time optimization and number of documents retrieved according to t, tf-idf threshold

number of comments. From the results we got, we decided to choose $\alpha = \beta = \gamma = 0.3$.

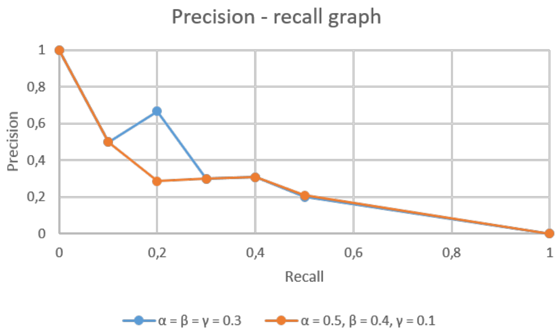

Figure 7: Precision - recall graph using different weights for the popularity formula.

Finally, we assessed the weight, ω, to give to popularity compared to tf-idf. This was done with the three different queries and is summarized in the Figures 8 and 9. Even if the best weight obviously depends on the query, we decided to choose $\omega = 0.2$ as it was the one that gave the highest area under the curve in average.

Search	**Boolean**	**Ranked**
Q1	0.178	0.404
Q2	0.230	0.092
Q3	0.323	0.450
Average	0.246	0.315

Table 2: Size of the area under the curve according to the retrieval model and the query considered

To conclude, the calculated areas under the curve for each retrieval model using the best related parameters, are given in Table 2. Seeing that the greatest the area, the best the search engine, we decided to use ranked retrieval and not Boolean retrieval for our final search engine. In fact, even if for Q2 the Boolean retrieval is better than the ranked retrieval, it is the general case that is important.

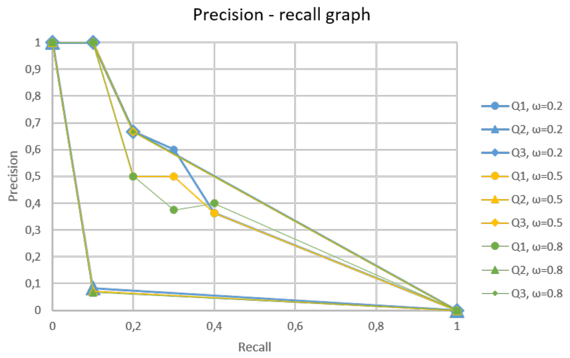

Figure 8: Precision - recall graph using different weights for the popularity compared to the tf-idf.

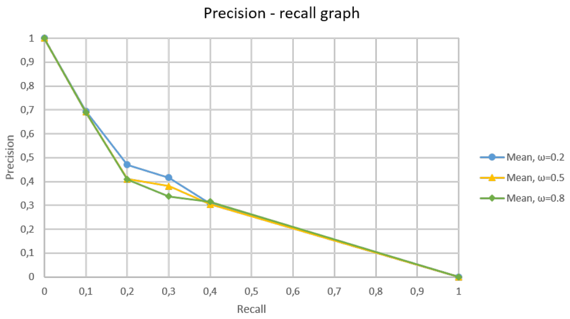

Figure 9: Average precision - recall graph using different weights for the popularity compared to the tf-idf.

The following parameters were chosen for the ranked retrieval:

- Threshold probability of 10^{-6}
- Taking into consideration the popularity of the video (with the parameters $\alpha = \beta = \gamma = 0.3$) with the weight of 0.2 for the ranking
- Using the metadata (title and description) with a weight of 10 for the tf-idf
- No optimization

4.2. Relevance feedback

Our search engine features a relevance feedback system, whose main principle is to check the relevance of the results, and then perform a new query taking the previous results relevance into account. We used the Roccio algorithm with parameters $\alpha = 1.0$ and $\beta = 0.5$ ($\gamma = 0$).

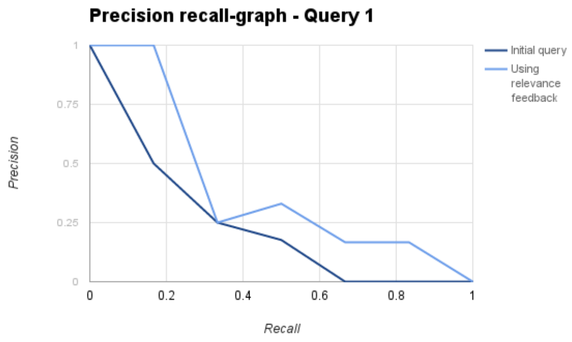

Figure 10: Precision - recall graph for Q1 and relevance feedback query

The relevance feedback here seems to work quite well and more large airplane videos are retrieved after applying it. Yet it is not always for every query, as the results on our second query ”playing with a dog” are better for the first query without the relevance feedback.

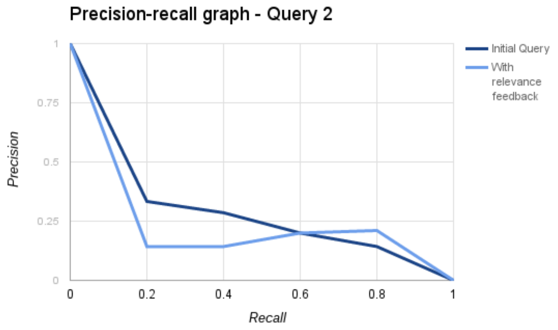

Figure 11: Precision - recall graph for Q2 and relevance feedback query

This is probably due to the absence of other relevant videos, thus the search engine cannot bring more good content. Our dataset is limited to only 1000 videos which is not much. With this result on ”playing with a dog”, we could then conclude that our video search engine performed well on the few examples of dogs it managed to find.

4.3. Selection of the description model

As mentioned before we used two models to generate the description of video frames. Results for the second model VGG_CNN_S that is smaller and gives considerably worse image representation than the first model, together with models comparison are presented in the Table 3.

Search	**CNN_S**	**ILSVRC_16**
Q1	0.150	0.404
Q2	0.500	0.092
Q3	0.185	0.45
Average	0.278	0.315

Table 3: Size of the area under the curve according to the model used for image representation and the query considered

Precision-recall graph for each query is shown in the Figure 12. For the second query ”Playing with a dog” it performs much better but average area under the curve is smaller than for VGG_ILSVRC_16 model.

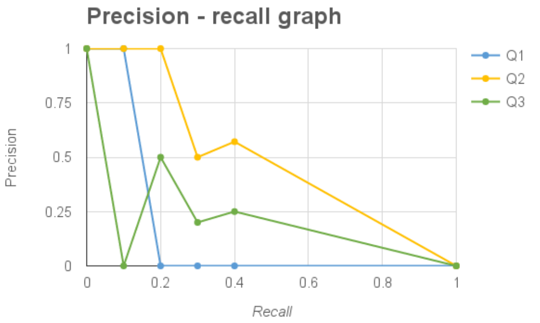

Figure 12: Precision - recall graph using VGG_CNN_S model to extract video frame representation.

We can assume that the better model performs on images classification task, the better image representation it gives. Also, the more powerful CNN is used, the better descriptions are generated according to BLEU scores [14] [9]. In our case, having two reference points only allow us to conclude that better BLEU scores improve search engine performance. However, we are not able to infer how the trend exactly looks like and whether it saturates at some point or not. Nevertheless, it confirmed our intuition that better image representation resulting in more accurate image description improves our search engine performance.

5. Conclusion

Video search and specifically visual queries represent emerging research areas that have a lot of progress to do. In this paper, we have focused on visual queries through text - words describing objects as well as actions taking place in the frames of the videos. Our experiments with different mappings of video to text as well as different retrieval methods have allowed us to find the best parameters for such a search engine. For example, our search engine is using ranked retrieval with a combination of popularity and tf-idf for the ranking. Hence, we provide users a video search engine available online. It shows considerable success for the top results: the first three results are close to the query even though the following results often seem to have nothing to do with the query.

The interesting and unique contribution of our paper to video search is the fact of using a Neural Talk framework in order to be able to use textual searches. Moreover, as required, our search engine is fast and instinctive to use, as Figure 13 shows. However, its precision and recall are not as high as the ones of other video search engines seeing that these ones only consider metadata.

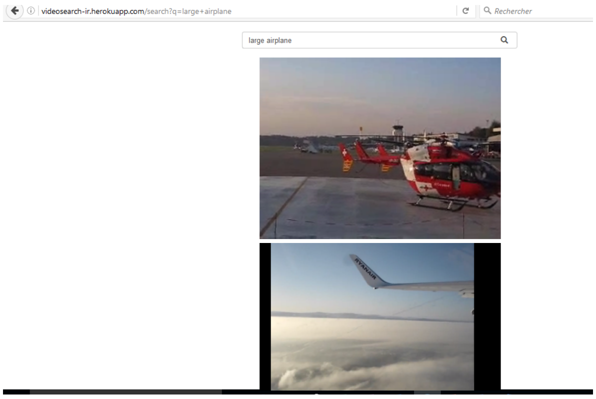

Figure 13: Screenshot of our online search engine used for Q2.

6. Discussion

Even though our search engine gave fairly good top results, its performance could be greatly improved. This is mostly due to the performance of the Neural Framework. In fact, with both language models, even if the framework's results are supposed to be ”often quite accurate” [17], they were most of the time inaccurate. Because of it, the textual descriptions we got often had nothing to do with the frame, therefore not being able to provide good results. This is a very tricky problem because neural descriptors are recent research subjects and a lot of improvement is still to be made. Therefore, it is necessary to look at enhancements in this field in order to update our search engine and get better results. Moreover, it is possible that the best retrieval model depends on the language model and vice versa. This would mean that our reasoning might not be fully independent.

To provide more data to the search engine in order to have better results, voice recognition and subtitles could also be used. In this way, not only would the visual aspect of the video be taken into consideration but also the speeches.

Concerning the search engine, the following changes need to be tested in order to provide a precision and a recall as high as possible:

- Extend not only the index but also the query
- Take into consideration the probability that the description of a frame is correct in order to calculate the ranking
- Save in the index not only the ID of the video and the number of the frame for each term but also the position in the description in order to allow subphrase queries

Finally, it is true that our current search engine has a dataset of a total of 1,000 videos only. To have performance results and to be more relevant for users, more videos need to be indexed. As described by H. Sundaram and S. Chang in [6], problems of performance might occur if the videos are not properly indexed and retrieved. This needs to be further worked on in order to guarantee a good search engine.

7. References

[1] V. Chandrasekhar D. Chen et. al. A. Araujo, M. Makar. Efficient video search using image queries. *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3082 – 3086, Oct. 2014.

[2] H. Schtze C. D. Manning, P. Raghavan. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *British Machine Vision Conference*, 2014.

[4] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *CoRR*, abs/1405.3531, 2014.

[5] Xu Cheng, Cameron Dale, and Jiangchuan Liu. Dataset for ”Statistics and Social Network of YouTube Videos”.

[6] S.-F. Chang H. Sundaram. Efficient video sequence retrieval in large repositories. *Storage and Retrieval for Image and Video Databases*, 7(108), December 1998.

[7] 2016 Statistic Brain Research Institute. YouTube Statistics, 2015.

[8] A. Zisserman J. Sivic. Video Google: a text retrieval approach to object matching in videos. *Ninth IEEE International Conference on Computer Vision, 2003.*, 2:1470 – 1477, Oct. 2003.

[9] Andrej Karpathy and Fei-Fei Li. Deep Visual-Semantic Alignments for Generating Image Descriptions. *CoRR*, abs/1412.2306, 2014.

[10] Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[11] D. G. Lowe. Object recognition from local scale-invariant features. *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, 2:1150 – 1157, Sept. 1999.

[12] W. Niblack J. Ashley M. Flickner, H. Sawhney. Query by image and video content: the QBIC system. *Computer*, 28:23 – 32, Sep. 1995.

[13] George A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November 1995.

[14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL ’02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[15] H.J. Meng H. Sundaram D. Zhong S. F. Chang, W. Chen. A fully automated content-based video search engine supporting spatiotemporal queries. *IEEE Transactions on Circuits and Systems for Video Technology*, 8:602 – 615, Sep. 1998.

[16] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

[17] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014.