# The IMOTION System at TRECVID 2016: The Ad-Hoc Video Search Task

Claudiu Tănase Luca Rossetto Ivan Giangreco Heiko Schuldt
Department of Mathematics and Computer Science
University of Basel, Switzerland
{c.tanase | luca.rossetto | ivan.giangreco | heiko.schuldt}@unibas.ch

Stéphane Dupont Omar Seddati
Université de Mons
Mons, Belgium
{stephane.dupont | omar.seddati}@umons.ac.be

## ABSTRACT

In this paper, we describe the details of our participation to the TRECVID Ad-Hoc Video Search (AVS) 2016 with the IMOTION system.

## 1. INTRODUCTION

This paper describes the details of our participation to the TREC Video Retrieval Evaluation 2016 [1] Ad-Hoc Video Search (AVS) task with the IMOTION system. The AVS task considers an end-user looking for a video segment in a collection that has not been previously manually annotated. In this task, 30 queries were released by NIST, for which the system should return a ranked result list of at most 1000 shot IDs each. The test data set is composed of 4593 Internet Archive videos with a total duration of 600 hours (IACC.3).

In this paper, we present the IMOTION system. The IMOTION system is a video retrieval system that comes with support for a large variety of query paradigms, e.g., query-by-sketch, query-by-example, query-by-motion, querying using semantic concepts. It supports querying using multiple query containers, e.g., using a hand-drawn sketch, a still image, a motion flow field or by specifying a semantic concept. The IMOTION system is built in a flexible and modular way and can easily be extended to support further query modes or feature extractors. The IMOTION system has participated to the Video Browser Showdown 2015 [7] and 2016 [6]. For the TRECVID AVS task, we only consider querying using semantic concepts.

This paper is structured as follows: Section 2 describes in detail the submitted runs, Section 3 presents a description of the system. In Section 4, we discuss the results of our runs. Finally, Section 5 concludes.

## 2. SUBMITTED RUNS OVERVIEW

The submitted runs have been named based on the priority ordering, so that 'Run 4' is the lowest priority run. Runs 4 and 3 are fully automated, runs 1 and 2 are manually assisted.

**IMOTION_4** captions for the test keyframes were automatically generated using DenseCap, ranked based on cosine similarity to the query payload text in a 400 topic LSI text feature space trained on a recent Wikipedia text dump and maxpooled (see Figure 1).

**IMOTION_3** this run linearly combines the following subruns, depicted in Figure 2:

- 300-dimensionality word2vec embeddings are used to create a query-to-class similarity matrix between the 30 topic and 325 custom classes aggregated from an AlexNet ILSVRC classifier. Topic scores are obtained by dot multiplying the classifier output matrix with this matrix.
- 4096-dimensionality feature vectors representing activations in the seventh fully connected layer in a VGG16 neural network were extracted from IACC.3 keyframes, MSCOCO and Flickr30k images. Using the same LSI model as in Run4, we scored captions on the MSCOCO and Flickr30k and used these textual similarity scores as target values for random forest and linear support vector regressors.

**IMOTION_2** linearly combines the following sub-runs (Figure 3):

- a query-to-class similarity matrix is manually constructed. Topic scores are obtained by dot multiplying the classifier output matrix with this matrix.
- 4096-dimensionality feature vectors representing activations in the seventh fully connected layer in a VGG16 neural network were extracted from IACC.3 keyframes, MSCOCO and Flickr30k images. 1957 training examples for the 30 queries were manually collected and RBF and chi-square kernel SVMs were trained on them.

**IMOTION_1** Score fusion by simple summing scores of the runs 4, 3 and 2, as shown in Figure 4.

# 3. SYSTEM DESCRIPTION

The following sections describe individual system components involved in scoring the runs.

## 3.1 Extra training data

In order to establish a relationship between visual and semantic textual data, we made use of datasets containing images which are annotated with short textual descriptions created by humans. The two datasets used in this work were the MSCOCO [3] caption dataset consisting of roughly 120 thousand images with one caption per image and the Flickr30k [8, 5] dataset which contains 30 thousand images and has five captions per image.

## 3.2 CNN feature extraction

To obtain features capable of describing the semantic content of an image, we use output of the $7^{th}$ fully connected layer of a neural network [2]. We used a pre-trained model based on the BVLC CaffeNet Model[1] but converted for a CPU-based DNN runtime[2]. The output of this layer is a 4096-dimensional sparse vector.

We complement the 4096 dimension features with categorical class scores also obtained from CNNs. Using top ranking $n$-grams from a web search engine we end up with 325 entry level categories representing objects or scenes referred to with common language (dog, kitchen, etc.). By collecting all images from hypernyms of an entry-level synset (e.g. 'poodle' and 'dalmatian' for 'dog') from ImageNet we obtain a new extended training set for 325 classes. A modified AlexNet image classifier is trained on this dataset, and is applied on the IACC.3 test keyframes [6].

We assume that we can express each of the 30 queries as a linear combination of these 325 class scores. In order to estimate these weights we extract word2vec [4] embeddings of dimensionality 300 of all the terms in the query payload and all the names of the 325 categories. We consider the Euclidean distance between two embeddings proportional to the semantic distance between the words. We compute a query-to-class similarity matrix by min-pooling the inverted distance between all the terms of one query and each class. Scores for runs 2 and 3 are computed by simply multiplying the score matrix with the query-to-class similarity matrix.

## 3.3 Automated captioning and retrieval

For run4 we generate dense automatic captions using DenseCap[3]. This results in a variable number (in average around 13) most likely captions for every test keyframe. When computing semantic keyframe-to-keyframe or keyframe-to-topic distances we min-pool pairwise distances between all generated captions for a keyframe.

In order to estimate semantic similarity between text captions we build our own text retrieval module. We use as training corpus a recent 13GB text dump of Wikipedia[4] on which we perform simple tokenizing (without stemming). We build a dictionary of the 100K most frequent words after filtering out all words with less than 20 occurrences or with occurrence in more than 10% of documents.
Using this dictionary we extract bag-of-words feature vectors from the available captions in the MSCOCO and Flickr30k datasets and we compute the *tf-idf* coefficients from the joint corpus. Using the transformed vectors we train a 400 topic LSI model. Textual similarity between two strings is computed as cosine distance between their representations in latent topic space. When comparing text queries, the starting *find shots of* is stripped in preprocessing.

## 3.4 Classification and regression

For the classification step (used in Runs 1 and 2) we used SVMs with nonlinear RBF and chi-square kernels. The training set consists of "fc7" features described in section 3.2 extracted from the 1957 manually collected images. Optimizing the SVM hyperparameter $\gamma$ as well as the regularization parameter $C$ is accomplished by gridsearch with values between $1e-4$ and 10 in logarithmic increments. Cross-validation is performed with a stratified 3-fold strategy, and multiclass is enforced through one-vs-rest. The mean value for the the classification score (accuracy) on cross-validation is at around 0.8. Estimator scores are converted to probabilities by using Platt's rule. The estimated probabilities from the 2 classifiers (gaussian and chi-square) are combined into one probability score for each shot using the "or rule"

$$p_{shot} = 1 - (1 - p_{RBF}) * (1 - p_{\chi^2}) \quad (1)$$

For the regression step we used random forest and support vector regression. The training data consisted of "fc7" features obtained from all the images in the MSCOCO and Flickr30k datasets. The regression target values represent text similarity (as defined in section 3.3) between the image's human annotation and the 'payload' part of each query (i.e., with the 'find shots of' stripped). For most captioned images there were several annotations per image: we max-pool the similarity value at image level. Because of time and memory concerns we were only able to train with a linear kernel implementation based on LIBLINEAR.

## 3.5 Fusion

Submission scores for runs 2,3 and 4 have been computed based on late (score) fusion. Given the lack of any validation data and the assumed low occurrence rate of true examples we decided to simply assign a weight of 1 to all features participating in weighted fusion.

# 4. RESULTS

Table 2 shows the detailed results of the four submitted runs.

In the manually-assisted runs, the IMOTION system was ranked at position 6 (run 1) and position 8 (run 2) out of 22, respectively. In the fully automatic runs, on the other hand, the IMOTION system ranked at position 19 (run 3) and position 20 (run 4) out of 30.

# 5. CONCLUSION

It it not surprising that the manually-assisted runs largely outperformed the automatic runs, since the former relied on

[1] https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet
[2] https://github.com/pluskid/Mocha.jl
[3] http://cs.stanford.edu/people/karpathy/densecap/
[4] available at https://dumps.wikimedia.org/enwiki/20160701/

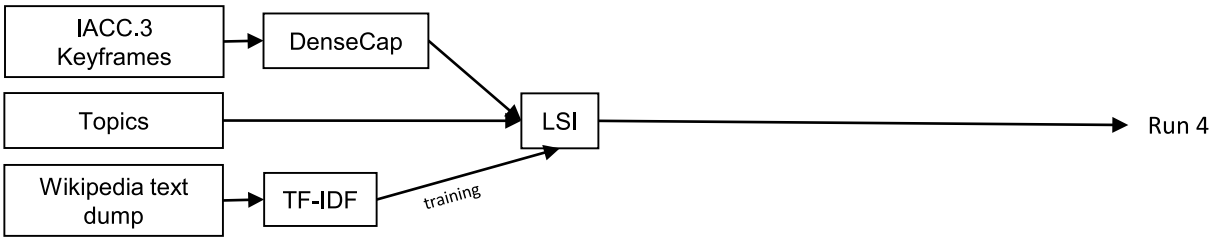


**Figure 1: Flow diagram of Run 4**

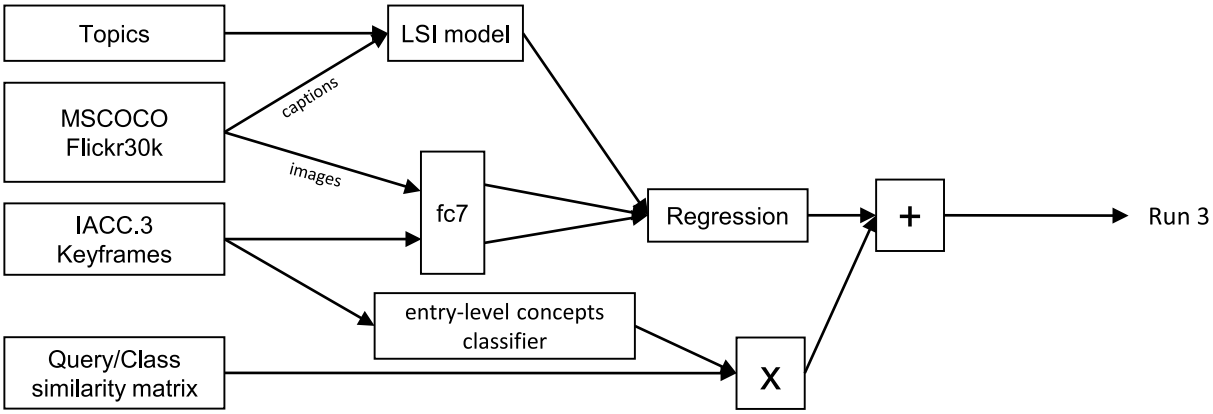


**Figure 2: Flow diagram of Run 3**

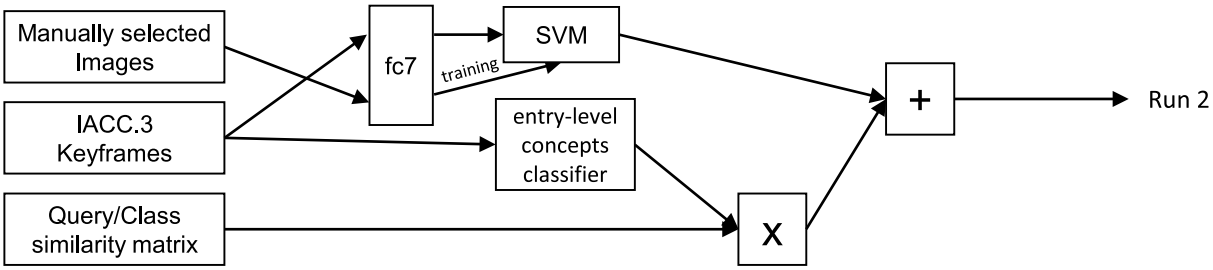


**Figure 3: Flow diagram of Run 2**

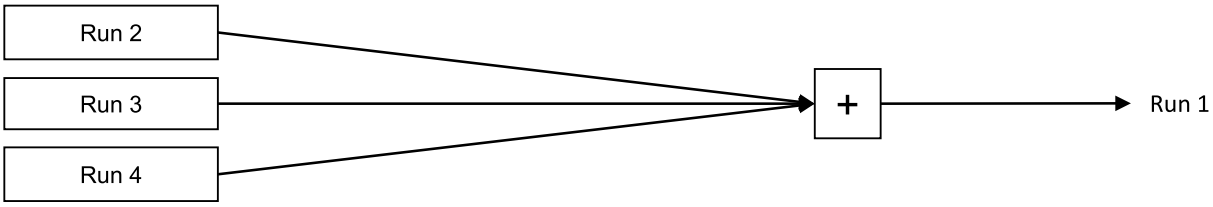


**Figure 4: Flow diagram of Run 1**

**Table 1: Mean extended inferred average precision per run**

| Run | Submission Type | Name | mAP |
|---|---|---|---|
| 1 | Manually-assisted | Fuse all | 0.047 |
| 2 | Manually-assisted | word2vec325manual | 0.046 |
| 3 | Fully automatic | word2vec325auto | 0.014 |
| 4 | Fully automatic | InvertedDeepCap | 0.012 |

manually collected examples and classification, while the latter depended on the quality of caption retrieval for providing regression targets. Results of the manual runs have confirmed that established techniques used in previous TRECVID SIN editions — higher order features from CNN upper layers combined with discriminative classifiers like SVMs — are still relevant in the AVS context.

One of the possible causes for the modest performance of the automated runs is in the unreliability of the LSI model's ranking of the captions. For example, in topic 524 all captions containing 'beard' were higher ranked than all captions matching 'white robe'. Even with perfect transfer, this imbalance would manifest directly in the final ranking of the shots. Given the submission list is limited at 1k, the performance consequently degrades.

A serious challenge compared with previous TRECVID editions was the lack of training and validation data for the imposed topics. One direct consequence was the inability to fine-tune parameters for the classification and fusion components. However, even with simple score summing used as fusion, the MAP improvements from Run4 up to Run1 are confirmed as significant by the TRECVID randomized test.

## 6. ACKNOWLEDGMENTS

This work was partly supported by the Chist-Era project IMOTION with contributions from the Belgian Fonds de la Recherche Scientifique (FNRS, contract no. R.50.02.14.F), the Scientific and Technological Research Council of Turkey (Tübitak, grant no. 113E325), and the Swiss National Science Foundation (SNSF, contract no. 20CH21_151571).

## 7. REFERENCES

[1] G. Awad, J. Fiscus, M. Michel, D. Joy, W. Kraaij, A. F. Smeaton, G. Quéenot, M. Eskevich, R. Aly, and R. Ordelman. Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *Proceedings of TRECVID 2016*. NIST, USA, 2016.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[5] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2641–2649, 2015.

[6] L. Rossetto, I. Giangreco, S. Heller, C. Tănase, H. Schuldt, S. Dupont, O. Seddati, M. Sezgin, O. C. Altıok, and Y. Sahillioğlu. Imotion–searching for video sequences using multi-shot sketch queries. In *International Conference on MultiMedia Modeling*, pages 377–382. Springer, 2016.

[7] L. Rossetto, I. Giangreco, H. Schuldt, S. Dupont, O. Seddati, M. Sezgin, and Y. Sahillioğlu. IMOTION – a content-based video retrieval engine. In *International Conference on MultiMedia Modeling*, pages 255–260. Springer, 2015.

[8] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

**Table 2: Mean extended inferred average precision per query and run**

| Task | Run 1 | Run 2 | Run 3 | Run 4 |
|---|---|---|---|---|
| 501 | 0.01 | **0.02** | 0.00 | 0.00 |
| 502 | 0.13 | **0.18** | 0.00 | 0.00 |
| 503 | **0.25** | 0.24 | 0.01 | 0.00 |
| 504 | **0.05** | **0.05** | 0.00 | 0.00 |
| 505 | **0.04** | 0.03 | 0.00 | 0.00 |
| 506 | 0.19 | **0.25** | 0.00 | 0.00 |
| 507 | 0.17 | **0.19** | 0.00 | 0.00 |
| 508 | 0.01 | 0.01 | **0.02** | 0.01 |
| 509 | **0.06** | 0.01 | 0.01 | 0.00 |
| 510 | 0.00 | 0.00 | 0.00 | 0.00 |
| 511 | **0.01** | **0.01** | 0.00 | 0.00 |
| 512 | 0.00 | **0.02** | 0.00 | 0.00 |
| 511 | **0.01** | **0.01** | 0.00 | 0.00 |
| 514 | 0.01 | 0.01 | **0.02** | 0.00 |
| 515 | 0.00 | 0.00 | 0.00 | 0.00 |
| 516 | 0.00 | 0.00 | 0.00 | 0.00 |
| 517 | **0.01** | **0.01** | **0.01** | 0.00 |
| 518 | **0.06** | 0.04 | 0.05 | 0.02 |
| 519 | 0.06 | 0.01 | **0.12** | 0.09 |
| 520 | 0.02 | **0.03** | 0.00 | 0.00 |
| 521 | 0.00 | **0.01** | 0.00 | 0.00 |
| 522 | **0.07** | 0.04 | 0.05 | 0.03 |
| 523 | **0.03** | **0.03** | 0.01 | 0.00 |
| 524 | **0.02** | 0.01 | 0.00 | 0.00 |
| 525 | **0.02** | 0.00 | 0.01 | **0.02** |
| 526 | **0.02** | 0.01 | 0.01 | **0.02** |
| 527 | 0.00 | 0.00 | 0.00 | 0.00 |
| 528 | 0.13 | 0.13 | 0.05 | **0.16** |
| 529 | 0.01 | 0.00 | **0.06** | 0.00 |
| 530 | 0.00 | 0.00 | 0.00 | 0.00 |
| Median | **0.019** | 0.012 | 0.002 | 0.001 |
| Average | **0.047** | 0.046 | 0.014 | 0.012 |