Towards Distributed, Semi-Automatic Content-Based Visual Information Retrieval (CBVIR) of Massive Media Archives

Christian Kehl and Ana Lucia Varbanescu
Instituut voor Informatica
Universiteit van Amsterdam
Email: Christian.Kehl@uni.no, a.l.varbanescu@uva.nl

Within our research, we study the possibilities of semi-automatic Content-Based Visual Information Retrieval (CBVIR) for large scale, growing media archives. The growing amount of audio-visual data, shown in table I, poses a challenge to media archives, such as the Dutch National Institute for Sound and Video (NISV) and comparable institutions (BBC, France Télévision). While visual indexing in practice is commonly performed manually, we propose to use Visual Object Classification (VOC) approaches for tagging archived and novel content items with respective labels. Our initial system design in figure 1 is the result of an interdisciplinary workshop (*ICT with Industry 2013*) that connects ideas from Computer Vision, High-Performance Computing and Information Retrieval, based on industry demand. Initial commercial applications (e.g. Orpheus [1] and pixolution [2]) provide rudimentary solutions for small datasets.

data size [hours of video]	500.000
data growth [hours per day]	approx. 50
data size (storage)	12 Petabytes
average content item length [minutes]	20

Table I
TYPICAL CURRENT DATA VOLUME OF MEDIA ARCHIVES, ON THE EXAMPLE OF NISV.

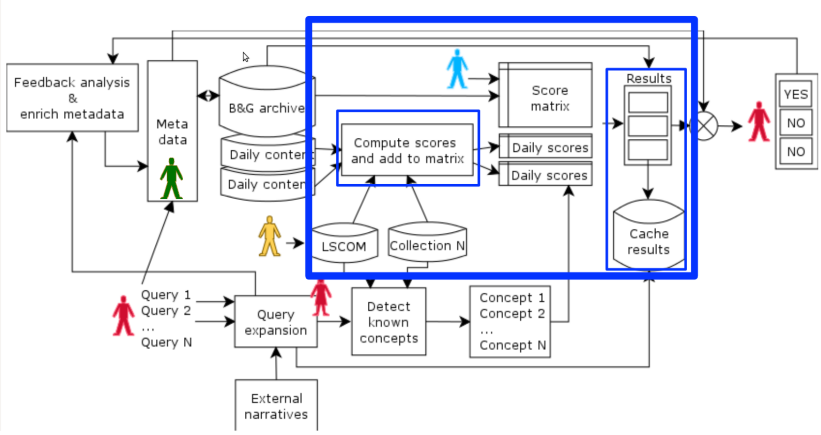

Figure 1. Initial system design of the "ICT for Industry" workshop 2013

Within the scientific VOC community, a paradigm shift occurred within recent years. Formerly, approaches such as template matching or Bag-of-Words (BoW) [1] were used for image [2]- and video [3] classification of large-scale repositories with GPU acceleration [4]. Since the introduction of Convolutional Neural Networks (CNNs) for solving VOC challenges by Krizhevsky et al. in 2012 [5], the

[1] img(Anaktisi) Image Retrieval software - http://orpheus.ee.duth.gr/anaktisi/
[2] pixolution - fusing visual and keyword search - http://fusion.pixolution.de/

approach has shown remarkable success in classifying large-scale, static-size image [6]- and video [7] repositories. Advances on CNNs are commonly demonstrated on static-size datasets, connected to classification challenges such as "Pascal VOC" [8] and ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9].

CNNs achieve superior classification precision by learning from massive training datasets, which is a time-consuming and memory-restricted process. Krizhevsky et al. already employed GPU Computing at the convolution layers [5]. Recently, the method has split into a data-parallel stage (including the convolution- and pooling layers of the Neural Network (NN)) of I/O-bounds operations, and a model-parallel stage (including fully-connected layers at the of the NN) of memory-bound operations. Krizhevsky proposed a parallelization stategy for both stages [10]. Other research groups explored different data [11]- and model [12] parallelization strategies. Exploiting data parallelism is prominent for speeding up CNNs in Deep Learning packages (e.g. Theano [13]).

With our research, we explore the challenges of dynamic, fast-growing data collections (e.g. modern media archives) and the tradeoffs they impose on existing CBVIR methods. Our approach addresses the challenges of such evolving datasets (i.e. a significant amount of content and tags changes after the initial training), which differs from the existing methods for challenge repositories (e.g. ILSVRC, PASCAL VOC). CNN classification scores follow their training set. We hence propose a new parametrization scheme for classifiers, taking the classification variance within different training sets as key assumption. We plan to steer the learning stage via adapting the training set's sample rather than the VOC parameters (e.g. NN architecture and connectivity). Additionally, as shown in figure 1, we plan to incorporate user feedback on the classification quality in the classifier parametrization.

We propose a semi-automatic implementation for a prototype system. The flexible design uses workflows to organise the data flow, which allows the integration of different VOCs as white-box models. Parallel workflow frameworks also facilitate the system's scaling across computing platforms, such as the DAS-4 cluster[3], allowing the transparent implementation of parallelization strategies [10]–[13] on accelerators. WS-VLAM [14][4] is a viable workflow implementation, which organises the data flow according to state charts (such as fig. 2). In specific test cases, available CNNs can be used as visual object classifier.

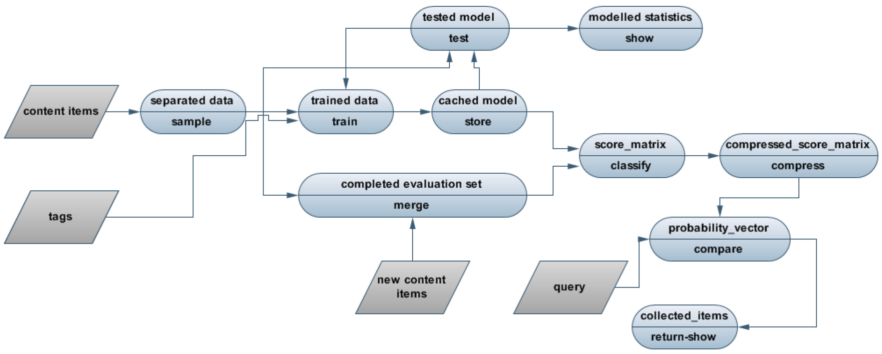

Figure 2. State chart of the visual classification system. Rhombi model datasets, while rounded squares model state transformations. For each transformation, the top row describes the output state of the transformation, the bottom row describes the transformation function.

Initial experiments were conducted on the CIFAR-10 [15] dataset, while novel tags and sample images were successively added from CIFAR-100 [15]. The experiments were conducted on GPU- and shared-memory parallel computing platforms. CNN architectures were prototyped in PyLearn2 [16]. The experiment architecture resembles the network of Krizhevsky et al. [5]. For testing the use of pre-trained networks and final network layer recomputation on tag- and image updates, a base model with a sigmoid non-linearity as last layer is used. Shown experiments adhere to the following scheme:

- Experiment 1: training the basic CIFAR-10 dataset with 100 $\frac{samples}{tag}$

[3]The Distributed ASCI Supercomputer 4 - http://www.cs.vu.nl/das4/about.shtml
[4]WS-VLAM implementation "pumpkin" - https://github.com/recap/pumpkin

- Experiment 2: training the base dataset with 100 $\frac{samples}{tag}$, for re-training of updates
- Experiment 3: training CIFAR-10 (100 $\frac{samples}{tag}$) with 3 added tags (10 $\frac{samples}{tag}$, under-represented)
- Experiment 4: training CIFAR-10 (250 $\frac{samples}{tag}$) with 3 added tags (10 $\frac{samples}{tag}$, highly under-represented)
- Experiment 5: training CIFAR-10 (100 $\frac{samples}{tag}$) with 3 added tags (all 100 $\frac{samples}{tag}$)
- Experiment 6: training a dataset replacing 3 original CIFAR-10 tags with 3 CIFAR-100 tags and adding all corresponding CIFAR-100 samples. This represents a generalisation after large media updates, using 100 $\frac{samples}{tag}$
- Experiment 7, 8 and 9: Equal datasets as experiments 3, 5 and 6, but only re-training the final softmax classification layer, using experiment 2 as base model

The evaluation of each experiments used the full test dataset of 1,000 samples per class from CIFAR-10, and all test images for added tags. The tag replacement was done to resemble generalisations. Average prediction error rates per experiment are given in table II. Table III shows the computation times per platform. The tested platforms include NVIDIA Tesla C2050 (plat.1)- and NVIDIA GTX680 (plat.2) accelerators, a workstation graphics adapter (plat.3), a shared-memory Intel SandyBridge CPU of the DAS-4 cluster (16 threads used, plat.5), and an Intel Xeon CPU (8 threads used, plat.4). Figure 3 shows the impact of the computing architecture on the runtime for the three major scenarios of full model computation, model precomputation and last-layer retraining.

Exp. 1	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9
0.89670	0.90551	0.89702	0.91935	0.86563	0.89881	0.91458	0.84435

Table II
MODEL PREDICTION ERRORS OF CNN-BASED VOC, USING DYNAMIC DATA SAMPLES.

Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9
00:25:52	00:49:16	00:26:33	00:31:19	00:27:20	00:27:21	00:05:08	00:05:14	00:05:21
00:19:13	00:23:49	00:10:51	00:13:13	00:11:18	00:11:17	00:02:55	00:02:42	00:02:47
04:04:04	07:42:21	04:12:25	05:38:37	04:28:12	04:27:19	00:43:06	00:45:29	00:45:32
04:09:06	07:49:06	04:35:59	06:33:25	05:02:56	04:49:39	00:47:14	00:49:21	00:50:28
01:15:54	02:42:19	01:18:45	n/a	n/a	n/a	00:18:19	n/a	n/a

Table III
TRAINING TIMES FOR EACH EXPERIMENT ON GIVEN PLATFOREMS (LISTING ORDER IN TEXT).

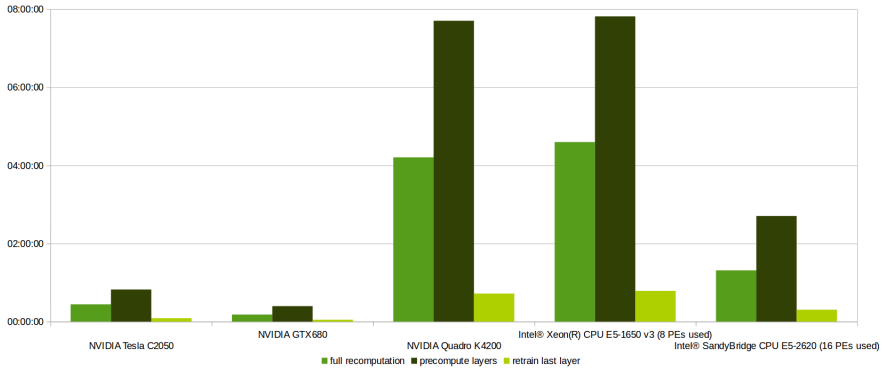

Figure 3. Visual comparison of runtimes on the assessed platforms, for the scenarios of full recomputation, model precomputation and last-layer retraining.

The comparably small CIFAR-10 dataset was chosen for the experiments due to time constraints.

Its prediction error rates are limited by the large amount of overfitting with respect to the parameter space. Despite the overfitting, we can conclude from table II that fully recomputed- and last-layer retrained experiments score comparably. This means the usage of retraining on update procedures appears to be a valid procedure to largely reduce computation times on data updates. A significant impact of tag under-representation of samples on the error rate was not observed in the experiments, which may be present on larger datasets. However, even within this small example, different sample rates have an impact on the final score. Higher samples rates score on average 0.8% better than lower sample rates. Tag generalisation has shown to be a good way to improve scores.

Accelerators are favourable to the method, due to its large amount of convolution operations. More interestingly, the experiments show the use of precomputed models for large-scale reclassification when introducing new image samples- and tags. Although model precomputation takes approximately twice the time of full model training, large amounts of updates (as shown in table I) justify precomputations. In combination with very fast last-layer retraining, the proposed method potentially outperforms full model recomputations (see fig. 3).

The next step is the further method evaluation on the ILSVRC 2010 model with additional ImageNet content and tags. We will moreover assess the impact of model retraining at different layers on accuracy and runtime. The measurements will give us further indications towards the accuracy-speed trade-off's and scalability. This also gives further insight to the theory of steering the classification via its data sample.

Acknowledgements

We thank the Dutch National Institute for Sound and Visual (NISV) for the positive collaboration, and the ASCI research school for its support and access to the DAS-4 cluster. We furthermore thank the anonymous reviewers for their constructive feedback.

References

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22, 2004, pp. 1–2.

[2] K. E. A. Van de Sande, T. Gevers, and C. G. M. Snoek, "Empowering visual categorization with the gpu," *Multimedia, IEEE Transactions on*, vol. 13, no. 1, pp. 60–70, Feb 2011.

[3] C. Snoek, K. Sande, O. Rooij, B. Huurnink, J. Uijlings, M. Liempt, M. Bugalhoy, I. Trancosoy, F. Yan, M. Tahir, K. Mikolajczyk, J. Kittler, M. Rijke, J. Geusebroek, T. Gevers, M. Worring, D. Koelma, and A. Smeulders, "The mediamill trecvid 2009 semantic video search engine," 2009. [Online]. Available: http://epubs.surrey.ac.uk/733282/

[4] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha, "Real-time bag of words, approximately," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, ser. CIVR '09. New York, NY, USA: ACM, 2009, pp. 6:1–6:8. [Online]. Available: http://doi.acm.org/10.1145/1646396.1646405

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 818–833.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[8] M. Everingham, S. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, pp. 1–39, 2014. [Online]. Available: http://dx.doi.org/10.1007/s11263-014-0733-5

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," 2014.

[10] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv preprint arXiv:1404.5997*, 2014.

[11] D. Scherer, H. Schulz, and S. Behnke, "Accelerating large-scale convolutional neural networks with parallel graphics multiprocessors," in *Artificial Neural Networks–ICANN 2010*. Springer, 2010, pp. 82–91.

[12] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le *et al.*, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1223–1231.

[13] W. Ding, R. Wang, F. Mao, and G. Taylor, "Theano-based large-scale visual recognition with multiple gpus," *arXiv preprint arXiv:1412.2302*, 2014.

[14] M. Baranowski, A. Belloum, and M. Bubak, "Mapreduce operations with ws-vlam workflow management system," *Procedia Computer Science*, vol. 18, no. 0, pp. 2599 – 2602, 2013, 2013 International Conference on Computational Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913005929

[15] A. Krizhevsky, "Learning multiple layers of features from tiny images," Technical Report, Computer Science Department, University of Toronto, 2009.

[16] I. J. Goodfellow, D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio, "Pylearn2: a machine learning research library," *arXiv preprint arXiv:1308.4214*, 2013. [Online]. Available: http://arxiv.org/abs/1308.4214