Searching and annotating 100M Images with YFCC100M-HNfc6 and MI-File

Giuseppe Amato
ISTI-CNR
Via G.Moruzzi, 1
Pisa, Italy 56124
giuseppe.amato@isti.cnr.it

Fabrizio Falchi
ISTI-CNR
Via G.Moruzzi, 1
Pisa, Italy 56124
fabrizio.falchi@isti.cnr.it

Claudio Gennaro
ISTI-CNR
Via G.Moruzzi, 1
Pisa, Italy 56124
claudio.gennaro@isti.cnr.it

Fausto Rabitti
ISTI-CNR
Via G.Moruzzi, 1
Pisa, Italy 56124
fausto.rabitti@isti.cnr.it

ABSTRACT

We present an image search engine that allows searching by similarity about 100M images included in the YFCC100M dataset, and annotate query images. Image similarity search is performed using YFCC100M-HNfc6, the set of deep features we extracted from the YFCC100M dataset, which was indexed using the MI-File index for efficient similarity searching. A metadata cleaning algorithm, that uses visual and textual analysis, was used to select from the YFCC100M dataset a relevant subset of images and associated annotations, to create a training set to perform automatic textual annotation of submitted queries. The on-line image and annotation system demonstrates the effectiveness of the deep features for assessing conceptual similarity among images, the effectiveness of the metadata cleaning algorithm, to identify a relevant training set for annotation, and the efficiency and accuracy of the MI-File similarity index techniques, to search and annotate using a dataset of 100M images, with very limited computing resources.

CCS CONCEPTS

• **Information systems → Image search**;

KEYWORDS

Image Search, Image Annotation, Deep Learning

ACM Reference format:
Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro, and Fausto Rabitti. 2017. Searching and annotating 100M Images with YFCC100M-HNfc6 and MI-File. In *Proceedings of CBMI '17, Florence, Italy, June 19-21, 2017,* 4 pages.
https://doi.org/10.1145/3095713.3095740

1 INTRODUCTION

Deep Convolutional Neural Networks (DCNNs) have recently shown impressive performance on a number of multimedia information retrieval tasks [7, 13]. In particular, the activation of the DCNN hidden layers has been also used in the context of transfer learning and content-based image retrieval [4, 12]. In fact, Deep Learning methods are "representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level" [8]. These representations can be successfully used as features in generic recognition or visual search tasks.

In this paper we present a public on-line Content-Based Image Retrieval system indexing about 100M images. It allows searching for similar images to the query and also to annotate the query images. The searched dataset is YFCC100M [14] that is the largest Creative Commons image dataset available today. The image search engines relies on the deep features contained in YFCC100M-HNfc6, which we extracted from YFCC100M. The 4,096-dimensional features vectors were indexed using MI-File [2], a permutation-based approximated data structure. A cleaned subset of metadata and images of the YFCC100M dataset was identified and used as training set to perform unsupervised automatic image annotation [10] of submitted queries.

The on-line demo is available at http://mifile.deepfeatures.org. A screen-shot of the interface can be seen in Figure 1

2 THE YFCC100M-HNFC6 DATASET

The YFCC100M-HNfc6 dataset [1] consists of visual deep features extracted from the Yahoo Flickr Creative Commons 100 Million (YFCC100M)[1]. The YFCC100M dataset was created in 2014 as part of the Yahoo Webscope program. YFCC100M consists of 99.2 million photos and 0.8 million videos uploaded to Flikcr between 2004 and 2014 published under a Creative Commons commercial or non commercial license. Metadata associated with each media, as for instance, user tags, user descriptions, etc., are also included in the in the YFCC100M dataset.

The YFCC100M-HNfc6 feature dataset [1] was created using the Caffe [6] framework. In particular we used the neural network Hybrid-CNN whose model and weights are public available in the

CBMI '17, June 19-21, 2017, Florence, Italy

ACM ISBN 978-1-4503-5333-5/17/06…$15.00
https://doi.org/10.1145/3095713.3095740

[1] http://bit.ly/yfcc100md

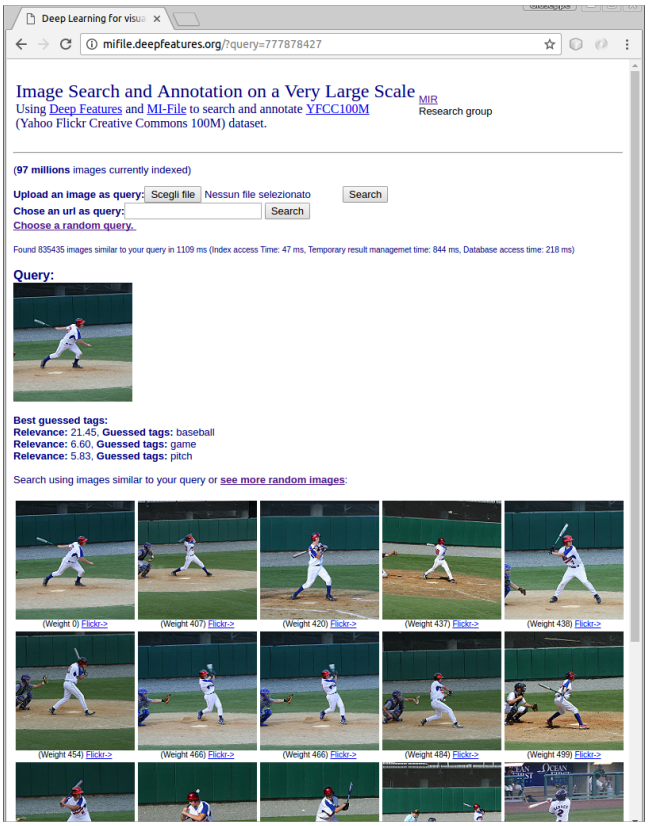

Figure 1: Screen shot of the on line image content based search engine

Caffe Model Zoo[2]. The Hybrid-CNN was trained on 1,183 categories (205 scene categories from Places Database and 978 object categories from the train data of ILSVRC2012 (ImageNet) with 3.6 million images [16]. The architecture is the same as Caffe reference network. The deep features we have extracted are activation of the fc6 layer.

The YFCC100M-HNfc6 feature dataset is public available at http://www.deepfeatures.org and is included in the Multimedia Commons initiative corpus. The Multimedia Commons initiative[3] is an effort to develop and share sets of computed features and ground-truths for the YFCC100M.

3 MI-FILE

The image search engine allows searching for similar images according to the deep features of the YFCC100M-HNfc6 features dataset in a database of about 100M images. To guarantee high efficiency and accuracy of the similarity search process, the deep features were indexed using a permutation based indexing technique [2].

Recently, permutation based indexes have attracted interest in the area of similarity search. The basic idea of permutation based indexes is that data objects are represented as appropriately generated permutations of a set of pivots (or reference objects). Let P be the set of pivots. An object o is represented as a permutation $\Pi_o = (p_1, \ldots p_n)$, of the pivots $p_i \in P$, sorted according to their distance from o. Similarity queries are executed by searching for data objects whose permutation representation is similar to that of the query. This, of course, assumes that similar objects are represented by similar permutations of the pivots.

One of the most promising permutation based approach is the MI-File. It uses an inverted file to store relationships between permutations. It also uses some approximations and optimizations to improve both efficiency and effectiveness. The basic idea is that entries (the lexicon) of the inverted file are the pivots P. The posting list associated with an entry $p_i \in P$ is a list of pairs $(o, \Pi_o^{-1}(i))$, $o \in C$, i.e. a list where each object o of the dataset C is associated with the position of the pivot p_i in Π_o.

As already mentioned, in [2] it was observed that truncated permutations (that is sequences of sorted pivots containing just the first elements of a permutation) can be used without huge lost of effectiveness. MI-File allows truncating the permutation of both data and query objects independently. We denote with l_x the length of the truncated permutation used for indexing and with l_s the one used for searching (i.e. the length of the query permutation).

The MI-File also uses a strategy to read just a small portion of the accessed posting lists, containing the most promising objects, further reducing the search cost. The most promising data objects in a posting list, associated with a pivot p_i for a query q, are those whose position of the pivot p_i, in their associated permutation, is closer to the position of p_i in the permutation associated with q. That is, the promising objects are the objects o, in the posting list, having a small $|\Pi_o^{-1}(i) - \Pi_q^{-1}(i)|$. To control this, a parameter is used to specify a threshold on the maximum allowed position difference (*mpd*) among pivots in data and query objects. Provided that entries in posting lists are maintained sorted according to the position of the associated pivot, small values of *mpd* imply accessing just a small portion of the posting lists.

Finally, in order to improve effectiveness of the approximate search, when the MI-File execute a k-NN query, it first retrieves $k \cdot amp$ objects using the inverted file, then selects, from these, the best k objects according to the original distance. The factor $amp \geq 1$, is used to specify the size of the set of candidate objects to be retrieved using the permutation based technique, which will be reordered according to the original distance, to retrieve the best k objects.

4 AUTOMATIC IMAGE ANNOTATION

The presented image search engine, in addition to search for semantically similar images, to the image query, also offers the possibility of annotating automatically the query images, with textual tags.

The Hybrid-CNN that we used to extract the deep features, is also able to associate an image with one of the 1,183 categories (205 scene categories from the Places Database and 978 object categories from ILSVRC2012) it was trained from. However, these categories are insufficient to associate relevant tags with any submitted query image. These categories, in fact, were not chosen according to the way people actually describe their pictures.

In this work, we address a special case of Automatic Image Annotation task [3, 9]. Specifically, the image annotation technique

[2]http://github.com/BVLC/caffe/wiki/Model-Zoo
[3]http://multimediacommons.wordpress.com/

that we defined is an Unsupervised Image Annotation approach [10], that is a method that uses the knowledge implicitly existing in a huge collections of unstructured texts describing images, and it is able to label images without training a model.

In the image search engine we used the tags and descriptions, contained in the metadata of the media in the YFCC100M dataset, as knowledge base for the automatic annotation engine. The annotation engine was obtained using a k-NN classification algorithm leveraging on the similarity between the deep features, as follows.

We first selected a subset of the YFCC100M images and metadata according a strategy, briefly described in next section, that identifies images with relevant textual descriptions and tags.

The deep features of the resulting selected image subset were indexed using again a MI-File index. When a image query to be annotated is received, the first 2,000 most similar images to the query are retrieved from the selected subset. Then, the 2,000 retrieved images, sorted according to their similarity to the query, are sequentially inspected by retrieving their selected textual descriptions. The terms in the accessed metadata are stemmed, using the Porter stemmer [11], and the count of the occurrences of the various stem, in all the 2000 retrieved metadata, is inclemently updated.

When metadata are accessed, the *id* of the owner of the retrieved images is considered as well. During the sequential scan of the 2,000 images, we take just one image per owner, in order to avoid bias due to the usage of several similar images published by the same user. The owner *id* is used to check that an image by the same owner was already considered, so the current one should be discarded.

The sequential scan of the 2,000 retrieved images stops as soon as matadata from 70 images are used (multiple images coming from the same owners are not used and not considered in the count). The value 70 was chosen as a good compromise between effectiveness and efficiency. The terms corresponding to the most frequent stems collected in the above process are suggested as tags for the query.

Some preliminary tests on tag prediction performance, using the benchmark suggested in Yahoo-Flickr Grand Challenge on Tag and Caption Prediction, [15], and without limiting the vocabulary to the 1540 tags to be predicted in the challange, gave a precision@5 of 0.22, a recall@5 of 0.15, and an accuracy@5 of 0.65.

5 METADATA CLEANING

Metadata of the media in the YFCC100M dataset contains the tags and the descriptions given by the users of Flickr. These metadata are often noisy and inaccurate. Sometimes images do not contain descriptions and tags; sometime the associated tags and descriptions are wrong; sometimes they are not useful.

The above mentioned k-NN classification algorithm, applied to the full YFCC100M metadata set sometimes gives results that are not very accurate. In addition, applying the k-NN classification algorithm to the full dataset requires issuing a nearest neighbour search to the 100M image database.

In order to reduce the cost of execution of the k-NN classification and to have, at the same time, a more accurate subset of images and metadata to be used as training set, we have defined a metadata cleaning algorithm that selected a subset of images with relevant metadata and a subset of associated metadata with relevant tags.

The metadata cleaning algorithm leverages on the capability of the deep features, contained in the YFCC100M-HNfc6 feature dataset, to assess the semantic similarity between image contents.

The intuition behind the metadata cleaning algorithms is the following. If two images are very similar (according to the similarity measured by way of the deep features), and their metadata contain the same tag, then that tag is probably relevant to the two images. This intuition was used to define a clustering algorithm that takes into account both the visual and the textual part.

The preliminary step is the creation of an inverted index where each stem, extracted from the textual metadata, is associated with the list of images that contain it in their metadata. Then, we run our clustering algorithm on the images of each posting list of the inverted file, to group together similar images that are associated with the same user defined tags.

The outline of the clustering algorithm that we defined resembles, somehow, a variation of the *dbscan* [5] clustering algorithm. Given a stem s, we scan the list of images in the posting list (that is the list of images having the corresponding term in their textual metadata). For each image we run a k nearest neighbour search query, with a very small k, on the full index containing all images. We used k equal to 5 in this prototype.

The metadata of the k retrieved images are accessed to check that they contain the stem s. This is done, simply by making the intersection between the posting list associated with s and the k retrieved images. The images resulting from the intersection (if any) are very similar one to the other and are associated with the same stem s. This means that the corresponding term is probably relevant to these images.

Let us call c the set of images remaining after the intersection between the posting lists associated to s and the k retrieved images. The set of images c are first eliminated from the posting list associated with s, so they are not longer used as queries. If there are no other clusters previously generated for the stem s the set of the remaining images c is used to create a new cluster, associated with the stem s. If there are other clusters associated with s, previously created, and there is an intersection between c and some other clusters, c and the intersecting clusters are all merged together to form just one cluster associated with s. The merged cluster contains very similar images, all associated with the same tag. Finally, if c does not intersect with any other clusters, also in this case, c is used to create a new cluster associated to the stem s. The owner *id*s of the images are also recorded, in order to avoid the potential bias due to creating clusters with wrongly tagged and very similar images all coming from the same owner, as already discussed above. The process is repeated for all stems of the inverted file. At the end of the process, each stem is associated with a list of clusters of very similar images. This process eliminates all images without tags, and significantly mitigates the imprecision of the user generated tags. It is unlikely to have large clusters of visually similar images all associated with a wrong tag.

The cleaning algorithm has currently selected about 16 thousand terms, with associated set of image examples. The total number of selected image is about one million. The number of image examples per term depends on the occurrence frequency of the term, in the data set, and ranges from about 20 thousands for the most

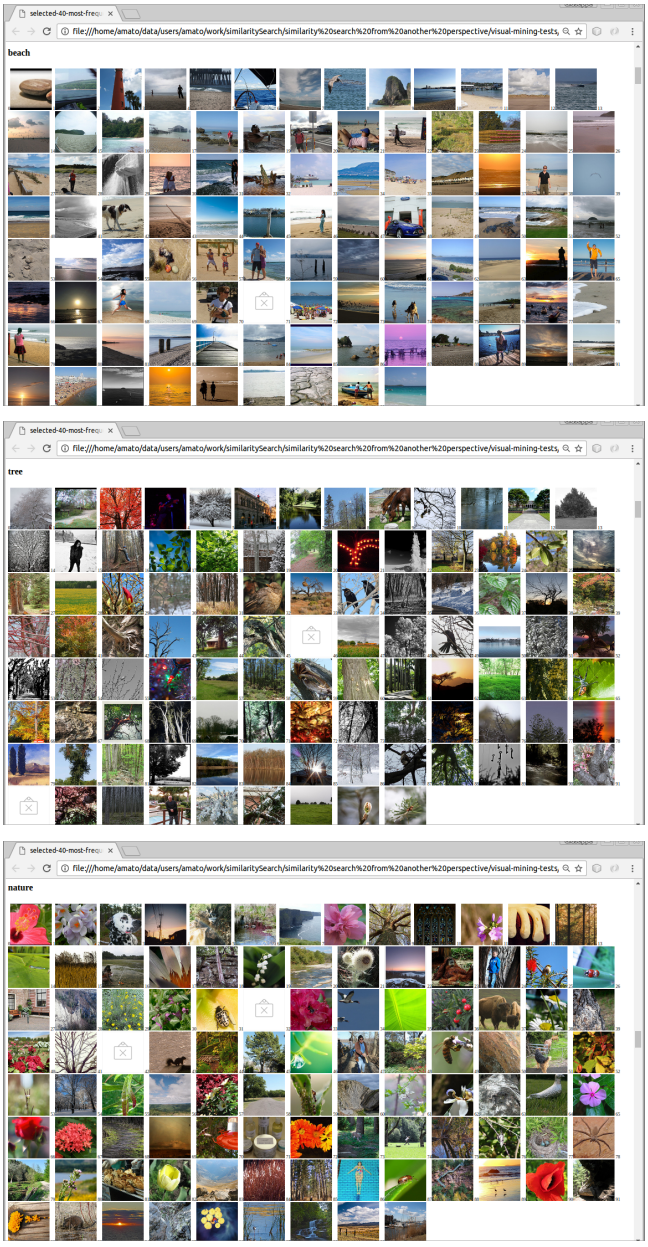

Figure 2: Example of classes generated by the clustering algorithm. From the top to the bottom: Beach, Tree, Nature. A sample of 100 images is shown.

frequent selected term (flower), up to 2 for the less frequent selected term (scale insect). Figure 2 shows an example of image clusters generated.

6 CONCLUSION AND FUTURE WORK

In this work, we present an on-line CBIR system which indexes, using MI-File, a dataset of deep features extracted from 100M images that are part of the well-known and public available YFCC100M dataset. The system, in addition to search for image from a dataset of 100 million images, is also able to suggest tag annotations for the query image. The automatic image annotation algorithm is based on a k-NN classifier executed on top of an automatically cleaned subset of the images and metadata from the YFCC100M dataset.

This system demonstrates the effectiveness of the deep features extracted and contained in the publicly available YFCC100M-HNfc6 feature dataset, the efficiency of the MI-File indexing approach, and the accuracy of the annotation strategy used.

ACKNOWLEDGMENT

This work was partially supported by Smart News, Social sensing for breaking news, co-founded by the Tuscany region under the FAR-FAS 2014 program, CUP CIPE D58C15000270008. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

REFERENCES

[1] Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro, and Fausto Rabitti. 2016. *YFCC100M-HNfc6: A Large-Scale Deep Features Benchmark for Similarity Search.* Springer International Publishing, Cham, 196–209. https://doi.org/10.1007/978-3-319-46759-7_15

[2] Giuseppe Amato, Claudio Gennaro, and Pasquale Savino. 2014. MI-File: using inverted files for scalable approximate similarity search. *Multimedia tools and applications* 71, 3 (2014), 1333–1362.

[3] Kobus Barnard and David Forsyth. 2001. Learning the semantics of words and pictures. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Vol. 2. IEEE, 408–415.

[4] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2013. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531* (2013).

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI Press, 226–231. http://dl.acm.org/citation.cfm?id=3001460.3001507

[6] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[9] Xirong Li, Tiberio Uricchio, Lamberto Ballan, Marco Bertini, Cees G. M. Snoek, and Alberto Del Bimbo. 2016. Socializing the Semantic Gap: A Comparative Survey on Image Tag Assignment, Refinement, and Retrieval. *ACM Comput. Surv.* 49, 1, Article 14 (June 2016), 39 pages. https://doi.org/10.1145/2906152

[10] Luis Pellegrin, Hugo Jair Escalante, Manuel Montes-y Gómez, and Fabio A. González. 2016. Local and global approaches for unsupervised image annotation. *Multimedia Tools and Applications* (2016), 1–26.

[11] M. F. Porter. 1997. Readings in Information Retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Chapter An Algorithm for Suffix Stripping, 313–316. http://dl.acm.org/citation.cfm?id=275537.275705

[12] Ali Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 806–813.

[13] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[14] Bart Thomee, Benjamin Elizalde, David A Shamma, Karl Ni, Gerald Friedland, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Commun. ACM* 59, 2 (2016), 64–73.

[15] Bart Thomee, Pierre Guarrigues, Liangliang Cao, and David A. Shamma. 2016. A Yahoo-Flickr Grand Challenge on Tag and Caption Prediction! https://multimediacommons.wordpress.com/tag-caption-prediction-challenge/. (2016). [Online; accessed 14-March-2017].

[16] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*. 487–495.