



## D5.2

---

### Visual mining and linking modules – v1

---

v 1.0 / 2015-01-09

---

Symeon Papadopoulos (CERTH), Etienne Gadeski (CEA), Adrian Popescu (CEA), Herve Le Borgne (CEA), Giorgos Orfanidis (CERTH), Meropi Pavlidou (CERTH)

---

The current deliverable is a technical report accompanying the first version of the USEMP visual mining and linking modules. In particular, it documents the underlying principles and methodologies, the exposed functionality, the respective implementation details, and the conducted evaluation experiments. In addition, it highlights the importance of each module for the project use cases and the multi-disciplinary issues arising from their deployment. In particular, the following modules are included and discussed: a) large-scale and privacy-aware concept detection, b) location and POI detection, c) face detection and recognition, d) logo and product recognition, and e) near-duplicate detection.



---

|                        |   |
|------------------------|---|
| Project acronym        | USEMP   |
| Full title             | User Empowerment for Enhanced Online Presence Management        |
| Grant agreement number | 611596  |
| Funding scheme         | Specific Targeted Research Project (STREP)                      |
| Work program topic     | Objective ICT-2013.1.7 Future Internet Research Experimentation |
| Project start date     | 2013-10-01  |
| Project Duration       | 36 months   |

---

|                       |   |
|-----------------------|---|
| Workpackage           | WP5   |
| Deliverable lead org. | CERTH   |
| Deliverable type      | Prototype   |
| Authors               | Symeon Papadopoulos (CERTH)<br>Etienne Gadeski (CEA)<br>Adrian Popescu (CEA)<br>Herve Le Borgne (CEA)<br>Giorgos Orfanidis (CERTH)<br>Meropi Pavlidou (CERTH) |
| Reviewers             | Niels van Dijk (ICIS)<br>Noel Catterall (HWC)   |
| Version               | 1.0   |
| Status                | Final   |
| Dissemination level   | RE: Restricted Group  |
| Due date              | 2014-12-31  |
| Delivery date         | 2015-01-09  |

---



---

#### **Version Changes**

---

- 0.1 Initial draft by Symeon Papadopoulos
  - 0.2 Updated structure and introduction by Symeon Papadopoulos
  - 0.3 Contributions from CEA
  - 0.4 Contributions from Symeon Papadopoulos and CERTH
  - 0.5 Contributions and editing by Symeon Papadopoulos
  - 0.6 Contributions from CEA
-

- 
- 0.7 First complete version available for internal review
  - 0.8 Refined version based on comments of Niels van Dijk
  - 0.9 Refined version based on comments of Noel Catterall
  - 1.0 Addition of missing command line inputs and minor revisions by CEA and CERTH
-

# Table of Contents

- 1. Introduction** ..... 3
  - 1.1. Scope of the deliverable ..... 3
  - 1.2. Visual mining and linking in USEMP ..... 3
  - 1.3. Research methodology and contributions ..... 5
  - 1.4. Multidisciplinary issues ..... 5
- 2. Concept detection** ..... 7
  - 2.1. Related work ..... 7
  - 2.2. Method description ..... 8
  - 2.3. Evaluation and testing ..... 9
    - 2.3.1. Concept detection and retrieval experiments ..... 9
    - 2.3.1. Privacy-oriented multimedia concepts experiment ..... 13
  - 2.4. Implementation and usage ..... 15
  - 2.5. Next steps ..... 16
- 3. Location and POI detection** ..... 18
  - 3.1. Related Work ..... 18
  - 3.2. Method description ..... 19
    - 3.2.1. Visual language model and Geo-Visual Ranking (GVR) ..... 19
    - 3.2.2. POI recognition based on CNN ..... 21
  - 3.3. Evaluation and testing ..... 22
    - 3.3.1. Location Estimation on MediaEval Placing Task 2014 ..... 22
    - 3.3.2. POI-location recognition on Oxford5k, Holidays and Div150Cred ..... 23
  - 3.4. Implementation and usage ..... 24
    - 3.4.1. Visual language model and Geo-Visual Ranking (GVR) ..... 24
    - 3.4.2. POI recognition based on CNN ..... 25
  - 3.5. Next steps ..... 27
- 4. Face detection and recognition** ..... 28
  - 4.1. Related Work ..... 28
  - 4.2. Method description ..... 28
  - 4.3. Evaluation and testing ..... 29
  - 4.4. Implementation and usage ..... 29
  - 4.5. Next steps ..... 29
- 5. Logo and product recognition** ..... 30
  - 5.1. Related Work ..... 30

- 5.2. Method description .....30
- 5.3. Evaluation and testing .....30
- 5.4. Implementation and usage .....31
- 5.5. Next steps .....32
- 6. Near-duplicate detection .....33**
  - 6.1. Related Work .....33
  - 6.2. Method description .....33
  - 6.3. Evaluation and testing .....33
    - 6.3.1. Results on WebTransforms .....35
    - 6.3.2. Results on Image Copy Detection dataset .....36
  - 6.4. Implementation and usage .....36
  - 6.5. Next steps .....36
- 7. Conclusions and future work .....37**
- 8. References .....38**

# 1. Introduction

---

This deliverable provides documentation on the first version of the prototype implementations of the USEMP visual mining and linking modules. This introductory section first delineates the scope of the deliverable; it proceeds with an overview over the delivered visual mining and linking modules. It continues with a description of the adopted research methodology and concludes with a discussion of the multi-disciplinary issues involved in the development and deployment of the presented modules.

## 1.1. Scope of the deliverable

This deliverable offers documentation on the delivered prototype implementations of the USEMP visual mining and linking modules. The deliverable addresses the following objectives: a) to make clear the role and usage of each module in the USEMP system, b) to describe the underlying research approaches and expose a number of technical implementation details, and c) to present the achieved experimental results and discuss aspects related to the deployment and integration of the modules in the system.

Although much of the deliverable content is addressed to the public community of interested researchers and practitioners, part of the discussion is dedicated to USEMP-specific aspects, contextualizing the work within the project background, work structure and plan.

## 1.2. Visual mining and linking in USEMP

The primary goal of building a number of visual mining and linking modules in USEMP is to endow the system with the capability to **conduct inferences about OSN users' interests and traits based on the visual content of the images** they share. These inferences are produced on a per-image level, but are subsequently exposed to the USEMP privacy scoring framework (documented in D6.1), where they are aggregated and combined (together with additional inferences based on text processing – see D5.1 – and complementary online trails and cues – also discussed within T6.1) to build and update rich user profiles. The types of information that can be inferred by processing users' images include a wide variety of personal information such as:

- Interests and activities (e.g. sports, arts, activism)
- Habits (e.g. smoking, drinking)
- Favorite brands and products (e.g. mobile phones, clothes)
- Home location and list of visited places
- Social interactions (i.e. people appearing in the same image)
- Social affinities (i.e. people sharing similar content)

Due to the variety of personal information to be mined, a number of visual mining and linking modules and approaches need to be employed, and have therefore been the subject of research and development within USEMP. One of the main researched approaches is **concept detection** (Section 2), i.e. the identification of entities, objects and themes of interest that are depicted in images. Concept detection is a versatile information extraction tool that can be used to detect a large variety of the aforementioned types of information,

including interests, activities, and habits, contributing to the recognition of depicted scenes. A particular set of concepts that are interesting for USEMP, the so-called *private concepts*, are separately discussed in the same section. A second important approach is **location and POI detection** (Section 3), which attempts to estimate the location of where an image was captured (i.e. the location of the depicted scene) based on visual cues as well as with the help of matching the image of interest to other images with known location (e.g. based on Exif or OSN platform-specific metadata). A third approach deals with the **detection and recognition of faces** in an image (Section 4), which typically reveals information about a users' social interactions and links. A further approach deals with the **detection of logos and products** in images (Section 5), which is useful for detecting the association between users and brands, and can be used as a value proxy for OSN users. Finally, a versatile visual mining and linking USEMP module is **near-duplicate detection** (Section 6), which does not directly lead to the extraction of personal information, but is of particular value to the USEMP personal information mining framework for two reasons:

- It enables to match an unknown image to one or more images with “known” attributes (e.g. known location, depicted topic and concepts), and in that way to propagate the known attributes to it.
- It makes it possible to infer latent links (or similarities) between OSN users (based on the similarity of the content they share), and in that way to propagate known information for those users to the “unknown” user, of whom the images are observed. A reference method for propagating user labels based on the connections/links between users is further discussed in D6.1.

Figure 1 illustrates the foreseen usage of the USEMP visual mining and linking modules listed above in a few exemplary cases.

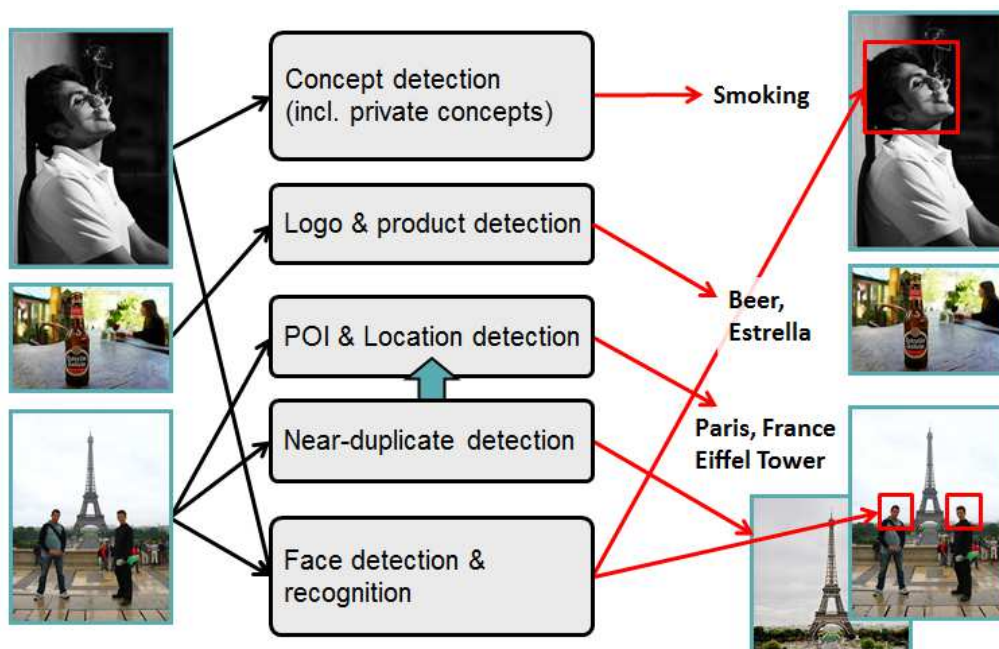


Figure 1. Example use and outputs of USEMP visual mining modules. To avoid overloading the image with content, only a small number of possible interconnections and outputs are presented.

## 1.3. Research methodology and contributions

The conducted research described in this deliverable was to a large extent shaped by the desiderata and insights coming from the USEMP disciplines (social science, legal studies, user studies, system design) as will be discussed in more detail in the next subsection. Having specified the main objectives of the visual mining and linking research in close collaboration with the above disciplines, the next step was to perform an extensive analysis of existing work on each of the studied fields (a summary of which is included in a dedicated subsection for each module). In each case, the most effective methods for the problem at hand were selected as the basis for the modules; subsequently, existing implementations of the selected methods were reused wherever possible, while in some cases development work was necessary to build the target method (each of the following sections specifies the respective details). Finally, to assess the reliability and quality of the prototyped solutions, they were evaluated using suitable publicly available datasets. In case no such datasets were available, new ones were created with a focus on the problems of interest.

Although much of the work performed in this first research and development iteration relied on existing computer vision and machine learning approaches, we consider that it resulted in a number of valuable research contributions. In particular:

- A new concept-level feature representation (Semfeat) that is particularly effective both for conventional concept detection settings and, importantly, for transferring concept models to new sets of concepts. The proposed representation is grounded on state-of-the-art computer vision advances (Convolutional Neural Networks, which fall under the family of Deep Learning methods) and is tested on large-scale datasets, as well as on datasets focused on *private concepts*.
- A number of location estimation and POI detection approaches were developed and tested on a number of publicly available datasets, and a promising approach was developed based on the use of CNN and transfer learning.
- A highly effective and efficient near-duplicate detection method was developed that outperforms most state-of-the-art approaches and scales gracefully with the addition of large amounts of distractor images.
- Preliminary implementations of face and logo recognition modules have been tested leading to competing results and helping highlight the limitations of existing methods.

## 1.4. Multidisciplinary issues

Although visual mining is mainly dealing with approaches from the areas of computer vision, image processing and machine learning, the presented research was considerably shaped by the rest of the USEMP disciplines, and at the same time provided actionable feedback to them. In the following, we provide a concise account of the inter-play between visual mining research and the different disciplines of the project.

D5.2 is informed by work done in WP2, WP3, WP4 and WP9 and it provides valuable input for WP6 and WP7. The legal analysis carried out in WP3, and more particularly in T3.6 which deals with coordination of legal aspects, clarified practical implications of visual content mining related to: processing of sensitive information (e.g. so that concept detection models and evaluation were tailored to effectively detecting sensitive information), copyright issues related to data used during training, ensuring that all USEMP components have clear IP



rights (in case of reusing existing components). Work on trade secrets and intellectual property carried out as part of D3.2 explored the tensions between profile representations on the end-user side, within OSNs and created in USEMP and made clear the complex interplay between these actors, as well as their respective rights and obligations.

The use case analysis in D2.1 and the associated requirements defined in D2.2 served as guidelines for the implementation of technical components. In particular, the following requirements are central here:

- [SR02] “The system may be able to process the information within one second such that the user can make informed decisions on their past data without long delays. In the event data processing is to take longer, a progress bar should be presented. A maximal extent of 10 seconds will be aimed for.” This requirement has strong implications in terms of processing speed for the implemented components.
- [SR04] “The system may be able to make best effort associations between data placed onto OSN(s) and the profile attributes which can be inferred from such data.” This requirement is a counterpart of [SR02] that focuses on component performance, which should closely follow state of the art developments.
- [SR11] “The system may be able to get fruitful insights on how relevant a user’s profile is for different stakeholders.” Through inferences made by technical components, the end-users should be able to have insightful information on how her profile is seen by OSNs and, possibly, by other stakeholders.

In D4.1, a comprehensive list of social requirements was established, which offers a user-side view of functionalities that need to be implemented by USEMP tools. Of particular interest here are:

- Req. 1 asking for more transparency about privacy problems at an institutional level and notably OSNs in this context.
- Req. 2 demanding a backward link between inferences and raw data which generated them to improve the accountability and provenance of the automatic decisions made by the system.
- Req. 10 asking for a low impact on browser speed of the USEMP plug-in, a requirement which is tightly linked to [SR02] mentioned above.

The extensive market analysis done in D9.3 showed that existing privacy enhancing tools and privacy feedback and awareness tools deal mostly with volunteered and/or observed data. A strong opportunity in USEMP is to provide users with a more complete view of how their data could be handled and exploited by OSNs. Another conclusion of D9.3 is that existing content visual mining tools are not tailored for privacy enhancement and, consequently, an adaptation step is needed in order to better satisfy domain requirements. Downstream, insights gained with D5.2 tools can be used both directly in the USEMP interface (D7.2), and as part of the privacy scoring framework created in D6.1, to complement social network mining inferences. For instance, user locations can be extracted from images and can be displayed directly by the USEMP interface to inform the user about her degree of exposure on this core privacy dimension. In a more complex functioning mode, logo and product recognition from images can first be used to link the user to different brands and then can be combined with social interactions (such as links, comments, shares etc.) in order to derive a value estimate for the shared image.

## 2. Concept detection

---

Concept detection is the core visual mining module of USEMP because it enables the project tools to make privacy related inferences from raw images and thus build much more detailed privacy profiles. According to the insights provided by WP2, WP4 and WP6 analyses, a very large variety of concepts<sup>1</sup>, i.e. entities, objects and themes of interest depicted in images, are illustrated in user content shared on OSNs and scalability in terms of recognizable concepts should be a core requirement, along with detection accuracy. To cope with these requirements and to keep abreast with latest developments in computer vision, the majority of concept detection experiments are performed using deep learning features. Focus is put on feature transfer from an initial training set to a larger number of concepts and on learning visual concepts from manually curated resources but also directly from the Web. This last line of research is particularly important in order to improve concept detection scalability with no or little manual effort. Of particular interest is the detection of privacy-related multimedia concepts, i.e. concepts, of which the inference could be perceived as deducing private information about the depicted individual, and to this end, a first set of separate experiments was performed in this direction. The results of concept detection can be either used as such or integrated with other cues (including textual and social network mining insights) to inform the users about privacy dimensions defined in D6.1. For instance, even with a straightforward use of concept detection results, it is possible to infer *privacy-related concepts* such as smoking or drinking directly from images or to build detailed consumer profiles by detecting recurrent concepts. In addition, inferences per image are aggregated by the privacy scoring framework described in D6.1 to produce user-level scores about the defined private profile dimensions.

### 2.1. Related work

Concept detection in images is usually cast as a supervised classification problem in which the description of images as a feature vector plays a central role. (Zhang et al., 2012) provide a detailed analysis of low-level features that were most often used for image classification, including global (wavelets, color histograms, etc.) and local features (bags of visual words). While recent, this review is illustrative of the very fast progress in computer vision, since it does not cover at all the recent achievements of a new generation of neural network approaches that are becoming the standard approach to deal with a large variety of image mining tasks. In particular, Deep Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012) have emerged as an efficient end-to-end way to represent images. Image representation is no longer designed based on prior knowledge but hierarchically learned from image pixels to higher-level primitives such as edges, corner, and object parts. CNNs have recently demonstrated impressive image classification performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC<sup>2</sup>) (Russakovsky et al., 2014). Compared to visual features such as Fisher Vector (Perronin et al., 2010), the use of CNN brought down the ILSVRC error rate from 0.26 to 0.15 in 2012, 0.11 in 2013, 0.066 in 2014. Moreover,

---

<sup>1</sup> The term *concept* is an established term in the multimedia analysis and computer vision research communities and is typically associated with a topic, entity, object or theme depicted in an image.

<sup>2</sup> <http://www.image-net.org/challenges/LSVRC/2014/> (consulted on 5/1/2015)

CNN-based feature extractors, such as Caffe (Jia, 2013) or Overfeat (Sermanet et al., 2013), were publicly released to facilitate research in this area.

These extractors provide pre-trained models and facilitate the extraction of features for new image collections, such as the ones handled here. When learning a large number of classifiers, the availability of a quality background visual resource is an important factor. The straightforward solution is to use manually labeled datasets (Russakovsky et al., 2014) but this approach is not scalable. Consequently, approaches for obtaining training examples from Web images have gained popularity. The source of images may vary from crawlers to search engines (Schroff et al., 2011) or photo sharing platforms (Li et al., 2012) and since these sets of images are inherently noisy, it is important to devise concept detection methods which are resilient to noise.

## 2.2. Method description

Our main objective is to propose a concept detection approach that is applicable to the very large number of concepts which can appear in OSN users' image streams. In order to achieve this objective, we adopt a feature transfer approach similar in spirit with that of (Razavian et al., 2014), though at a much larger scale, which ensures a good balance between concept detection accuracy and scalability. Feature transfer involves learning a deep learning model using a large set of initial concepts of general nature, e.g. "person", "sky", "forest", "building", etc. (typically in the range of 1,000) with a convolutional neural network and then exploiting the outputs of an intermediary layer of this model as features to model concepts that do not appear in the training set. The underlying assumption here is that the CNN model is able to generalize beyond its initial scope due to the fact that the initial concepts and the newly modeled ones share enough visual properties for a proper modeling of the latter. For instance, assuming that concepts "person" and "mountain" are contained in the set of initial concepts, then a new concept "hiking" could be thought of as a combination of those concepts (and some additional ones). Detection accuracy is ensured through the use of state of the art CNN models such as the reference model provided in Caffe (Jia, 2013)<sup>3</sup>. Scalability is a second central requirement in USEMP and it is dealt with in different forms:

- *Scalable and efficient machine learning.* Schroff et al. (2011) showed that techniques from the linear SVM family give good results in image classification based on local features derived from SIFTs. In initial experiments, we tested different linear solvers available from the LIBLINEAR library<sup>4</sup> with CNN features and our results confirmed their conclusions. Confirming the robustness of the features no notable differences were observed between the solvers tested. Furthermore, we also tested non-linear SVMs (notably the RBF kernel) and a slight improvement is obtained but with a high computational cost for both training and testing phases. Consequently, a L2-regularized L2-loss SVM was consistently used in our experiments. This solver is particularly interesting at test time since the classification score of a concept is done through a simple dot product computation.

---

<sup>3</sup> Experiments were carried out with publicly available pre-trained CNN models and more advanced ImageNet 2014 Challenge models are not yet publicly available. Models will be updated during the second period to keep abreast with the latest developments in the field.

<sup>4</sup> Liblinear website <http://www.csie.ntu.edu.tw/~cjlin/liblinear> (accessed on 23/12/2014)

- *Training with both manually validated images and Web data.* The public availability of ImageNet (Deng et al., 2009), a manually curated visual resource that includes over 14 million images illustrating 22,000 concepts, had a positive influence on research dealing with large scale concept detection. In USEMP, we exploit this dataset but also want to explore the use of Web resources as an even more comprehensive collection of visual data. This exploration is motivated by the fact that, while large, ImageNet does not cover concepts which are very relevant when dealing with privacy<sup>5</sup>. For instance, it offers very limited coverage of named entities (such as artefact names, places, products), which are needed to provide feedback about the consumer profile of a user. To overcome this problem, we carry out experiments with Flickr groups and Wikipedia concepts as a complementary source to ImageNet.
- *Concepts are learned independently from each other.* In a majority of classification tasks, such as the ImageNet Challenge (Russakovsky et al., 2014) it is assumed that all concepts are known in advance and a one-versus-all approach is used for learning. However, in practice it is often the case that new concepts need to be added to cope with user needs and, when working with tens of thousands of classes, relearning all models with a one-versus-all approach is highly impractical. Consequently, learning is done independently using a one-versus-many approach that is inspired by existing work from (Bergamo & Torresani, 2012) and (Scroff et al., 2011). Differently from them, we use a much more populated negative class in order to increase model robustness.

The use of pre-trained CNN models for feature transfer was thoroughly tested for general purpose concept detection, with application to image retrieval and annotation, but also for a small dataset of privacy-related visual concepts. Models are learned for 17,000 ImageNet<sup>6</sup> concepts which are represented by at least 100 images and for 30,000 Flickr groups, with 300 images per group. When a test image is provided to the system, it is compared to all existing concept models to decide which one(s) are present in the image. We noticed that a hard-coded threshold is not appropriate for this decision since classification scores vary from one concept to another and the decision about the presence of a concept is taken by using concept-adapted thresholds. These thresholds are learned by performing a 5-fold cross validation of each concept in which a fold is kept for testing and the others for training. The threshold value is then computed as the average classification score of images tested during cross-validation.

## 2.3. Evaluation and testing

### 2.3.1. Concept detection and retrieval experiments

Concept detection is usually evaluated in a classification setting, such as the one proposed by the ImageNet challenge (Russakovsky et al., 2014) that tests accuracy with 1000 diversified ImageNet concepts. The pre-trained CNN model used in our experiment was already successfully tested on this dataset in (Krizhevsky et al., 2012) and then fine-tuned in

---

<sup>5</sup> Here, we are mainly referring to concepts related to the privacy scoring framework of D6.1, which aims at capturing different notions of private information in a single information schema.

<sup>6</sup> Only the concepts which have at least 100 associated images were modeled to obtain a robust representation.

(Jia, 2013) (error rate ~15% when 5 guesses out of 1000 are allowed<sup>7</sup>). (Razavian et al., 2014) propose further tests on the same model on a large number of standard computer vision datasets and show that feature transfer is effective in these cases. However, the number of concepts is relatively limited (tens to hundreds, depending on the dataset) and a ground truth (aka gold standard)<sup>8</sup> is available each time. In our case, direct testing of tens of thousands of concepts in a classification setting is not straightforward since a ground truth is not directly available and we propose instead to test the quality of the models in an image retrieval setting. The main advantages of this evaluation are that: (1) models learned from different data sources can be compared on the same test data; (2) content-based image retrieval and automatic annotation can be compared with human annotations to study if there is a quality gap between them and (3) the models are used in an actual application and this gives a good indication about their practical usefulness and the quality of their results. The potential disadvantages are that: (1) model quality is evaluated in an application different from the final one (i.e. user profiling) and (2) only a part of the concepts are actually evaluated in retrieval. Disadvantage (1) is mitigated by the fact that, while technological evaluation is done indirectly, the perceived quality of concept detection will be assessed during user studies in lab and living lab studies. Disadvantage (2) is common to any technological evaluation since no ground truth exists for all concepts learned for integration in the USEMP system.

The main experiments are run using the ImageCLEF 2010 Wikipedia Retrieval dataset, a dataset that provides a realistic sample of Web images. It was created as part of the ImageCLEF evaluation campaign<sup>9</sup> and is publicly available. It includes 237,434 Wikimedia images that were extracted from a large set of Wikipedia articles covering a wide range of publicly available content. The dataset is thus fit for ad-hoc image retrieval experiments, in which any query can be submitted to the process. To ensure comparability with other methods tried on this dataset, we report mean average precision (MAP)<sup>10</sup> scores. Two types of experiments are carried out on the same dataset in order to compare retrieval based on automatic content processing and manual annotations:

- Content based image retrieval (CBIR) – the input given to the system is an example image and the task is to retrieve images that are similar to the query. Here we create an intermediate semantic representation of images, named Semfeat, in which each image is represented by the N strongest classification scores among the concepts modeled, with N typically in the range of tens. Two flavors of Semfeat are evaluated, i.e. Semfeat<sub>IN</sub> and Semfeat<sub>FG</sub>, based on ImageNet and Flickr group visual concept models respectively. For Semfeat<sub>FG</sub>, image reranking is applied using a kNN based algorithm to clean noisy images from an initial set of 300 images. Put simply, we use the group images as positive examples and a diversified image pool as negative examples and rank higher those images which are closely related to other group

---

<sup>7</sup> This is the official evaluation measure proposed by the Challenge organizers.

<sup>8</sup> By ground truth we refer to a reference dataset that enables the comparison of the automatically produced outputs (e.g. detected concepts) to the *correct* ones, where correct typically means that they have been validated by human annotators. Such a comparison enables the computation of scores reflecting the correctness/accuracy of different algorithms.

<sup>9</sup> ImageCLEF website <http://www.imageclef.org> <http://www.imageclef.org/> (accessed on 22/12/2014)

<sup>10</sup> MAP is a standard evaluation measure in information retrieval which accounts for the accuracy of results by taking into account the number of relevant images but also their position in the ranked list.

images in a ranking which combines positive and negative examples. Results are reported with the top 80% of images retained for each image. Other reranking thresholds were tested and close results were obtained in these cases. A negative class of 100,000 diversified images is used for concept learning.

- Automatic annotation based retrieval - the input given to the system is a text query and the task is to retrieve images which belong to the same topic (where the correct image-topic association is decided by human annotators who evaluate the results). Here, we use Flickr to automatically create visual models for single concepts in the query ( $Auto_{UNI}$ ) but also pairs of concepts ( $Auto_{BI}$ ) which appear in the Wikipedia queries. The same negative class of 100,000 diversified images used in CBIR experiments is also used here for concept learning.

In both cases, results are compared against a state-of-the-art textual retrieval system that exploits Wikipedia for efficient query expansion and tf-idf modeling of the collection (Popescu & Grefenstette, 2011). To our knowledge, this system obtains the best performance among the textual approaches tested on the Wikipedia retrieval collection. For CBIR, we compare our approach with the previous state-of-the-art, obtained with a Fisher vector representation of images (Perronin et al., 2010).

The 2010 query set contains 70 diversified queries, with 118 associated images (i.e. one or two example images per query). Retrieval over the Wikipedia collection is challenging because the image content is highly diversified. In addition, some topics are represented by only few relevant images. The Wikipedia Retrieval ground truth was built using a pooling approach and is therefore incomplete (Tsirikika et al., 2012). Since the new experiments are radically different from the ones used to create the ground truth, many new relevant results are found but would not be taken into account when computing performance, thus penalizing the newly tested approaches. To improve comparability, we extend the original ground truth by pooling the runs corresponding to the experiments reported here. This extension is realized using conditions similar to the original assessment, i.e. same topic narratives and a majority voting with three relevance judgments per image.

|        |      | Content based image retrieval |       |                       |                       | Automatic annotation |                    |
|--------|------|-------------------------------|-------|-----------------------|-----------------------|----------------------|--------------------|
| Method | Txt  | Fisher                        | Caffe | Semfeat <sub>IN</sub> | Semfeat <sub>FG</sub> | Auto <sub>UNI</sub>  | Auto <sub>BI</sub> |
| MAP[%] | 26.7 | 5.55                          | 13.73 | 16.24                 | 18.31                 | 18.4                 | 21.49              |

*Table 1. Txt is a state of the art result obtained with human annotation of the Wikipedia collection. Fisher is a CBIR experiment using the Fisher vector features presented in (Perronin et al., 2010). Caffe stands for the features extracted with the reference model from (Jia, 2013). Semfeat<sub>IN</sub> and Semfeat<sub>FG</sub> are the semantic descriptors obtained by concatenating classifier scores and sparsifying the obtained vector. Auto<sub>UNI</sub> and Auto<sub>BI</sub> are experiments with fully automatic annotation of the collection images.*

The obtained results are summarized in Table 1. While results obtained with human annotations of the images (Txt) are still better than the rest, the quality gap is significantly reduced compared to previous results in literature. Our best CBIR experiment (Semfeat<sub>FG</sub>) obtains a MAP score of 18.31% while a run with Fisher Vectors reaches only 5.55%. This difference is even more important if we take into account the fact that Fisher representations include over 100,000 dimensions while Semfeat representations have only a few dozens of active dimensions and retrieval can be considerably sped-up with the use of inverted indexes. Equally important is the fact that Semfeat representations improve results over the usage of raw Caffe features. These results are interesting in themselves but also because



they can be reused in T5.3 to create a late fusion approach that exploits CBIR for vocabulary free image annotation. This approach is interesting for concepts that do not have associated visual models but will still be associated through images by exploiting the annotations associated to the  $k$  visually most similar images (neighbors) of an image. The quality of automatic annotation is equally very interesting, with the best configuration (Auto<sub>BI</sub>) reaching 21.49%, only six MAP points behind a sophisticated text retrieval approach. Auto<sub>BI</sub> is obtained at the price of modeling concept pairs but interesting performance is obtained even when modeling only single concepts which appear in a query (MAP=18.4 for Auto<sub>UNI</sub>).

























































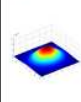

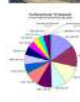




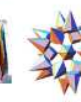
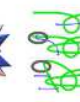

| Query image   | Textual topic<br>Group annotations   | Top 10 similar images   |   |   |   |   |   |   |   |   |   |
|---|--|---|---|---|---|---|---|---|---|---|---|
|    | <b>stars and galaxies</b><br>lomo<br>sun<br>astronomy<br>space<br>stars                          |    |    |    |    |    |    |    |    |    |    |
|    | <b>tennis player on court</b><br>tennis<br>dance<br>judo<br>sport<br>wimbledon                   |    |    |    |    |    |    |    |    |    |    |
|    | <b>mountains with sky</b><br>colorado<br>alaska<br>carpathian mountains<br>montana<br>washington |    |    |    |    |    |    |    |    |    |    |
|    | <b>palm trees</b><br>palm<br>tree<br>silhouette<br>clouds<br>sky                                 |    |    |    |    |    |    |    |    |    |    |
|   | <b>solar panels</b><br>solar<br>corten<br>greenroof<br>green<br>mill                             |   |   |   |   |   |   |   |   |   |   |
|  | <b>DNA helix</b><br>pencils<br>rainbow<br>blue<br>illustration<br>colors                         |  |  |  |  |  |  |  |  |  |  |

Figure 2. Illustration of content based image retrieval results obtained with Semfeat, a semantic image descriptor built on top of convolutional neural network features. Results are presented for diversified query images and they show that the approach works well when the query concepts are well covered in Semfeat. This is not, for instance, the case for the DNA helix image since there are no Flickr group models for this concept.

We illustrate CBIR results obtained with Semfeat<sub>FG</sub> in Figure 2. Obtained results are of good quality for topics related to natural images<sup>11</sup>, such as “tennis player on court”, “mountains with sky” and “palm trees” and also for “stars and galaxies”, a query which is well mapped to the CNN model used to create Semfeat. An exception is “solar panels”, a query for which there is confusion between the glass surface of the panels and that of modern buildings. Results are equally poor for “DNA helix” as a result of the fact that the CNN model does not contain concepts which are visually similar to “DNA helix” and feature transfer fails in this case.

<sup>11</sup> Natural images here denote photos of real-world scenes, i.e. images that are not computer-generated, graphics, diagrams, etc.






















































































































| Query                           | Annotation type | Top 10 results  |   |   |   |   |  |   |   |   |   |
|---------------------------------|-----------------|---|---|---|---|---|--|---|---|---|---|
| Spider with cobweb              | Automatic       |    |    |    |    |    |    |    |    |    |    |
|                                 | Manual          |    |    |    |    |    |    |    |    |    |    |
| Basketball game close up        | Automatic       |    |    |    |    |    |    |    |    |    |    |
|                                 | Manual          |    |    |    |    |    |    |    |    |    |    |
| Yellow buses                    | Automatic       |    |    |    |    |    |    |    |    |    |    |
|                                 | Manual          |    |    |    |    |    |    |    |    |    |    |
| DNA helix                       | Automatic       |    |    |    |    |    |    |    |    |    |    |
|                                 | Manual          |    |    |    |    |    |    |    |    |    |    |
| Portrait of Jintao Hu           | Automatic       |    |    |    |    |    |    |    |    |    |    |
|                                 | Manual          |   |   |   |   |   |   |   |   |   |   |
| ISS international space station | Automatic       |  |  |  |  |  |  |  |  |  |  |
|                                 | Manual          |  |  |  |  |  |  |  |  |  |  |

Figure 3. Illustration of text retrieval results obtained using automatic image annotations and respectively manual user annotations. Results obtained with automatic annotations are equivalent or even better compared to manual annotations for topics whose concepts are well represented in the basic CNN features used for annotation (spider, basketball, bus). Poor results are obtained for the concepts which are not well covered by the initial CNN features (DNA helix, faces). Interestingly, for ISS, automatic annotations return results of photos taken from the Station, while manual annotations return results which depict the Station itself.

Figure 3 illustrates text retrieval results based on automatic and human annotations on the ImageCLEF Wikipedia collection. Similar to CBIR results, good retrieval quality is obtained for topics, whose associated relevant results are natural images including visually coherent objects and scenes (“basketball game close up”, “yellow buses”). Poor results are obtained for concepts not well covered by the CNN model (“DNA helix”, “portrait of Jintao Hu”). This indicates that dedicated models need to be built for different conceptual domains. In the next section, we present an example of domain adaptation for location and POI recognition and we will further explore this research path during the second year of the project.

### 2.3.1. Privacy-oriented multimedia concepts experiment

Although the previous experiments on existing publicly available datasets are valuable for assessing the performance of the developed modules in generic settings, we are also interested in developing models that are tailored to the USEMP requirements. In particular, we are interested in concepts that relate to different aspects of private information with respect to the depicted individuals or to the users that share them online. Ultimately, we aim at developing concept models that are in line with the privacy scoring model of D6.1. As a first step, we conducted a small-scale experiment on the following set of five concepts:



drinking alcohol, smoking, extreme sports, political beliefs and luxurious living. Figure 4 illustrates one exemplary image from each of these concepts.



Figure 4. Example images from the privacy-oriented concepts of the manually curated dataset (faces have been obscured for protecting the anonymity of depicted individuals). From left to right: a) smoking, b) drinking alcohol, c) extreme sports (climbing), d) political beliefs (participation in demonstrations), e) luxurious living (yacht).

While in operation, the visual mining module of USEMP will detect such concepts in the content of OSN users and will make them aware of potential consequences. For instance, it is known that insurance companies apply price discrimination practices for people based on their social media profile (Raman et al., 2012), while several employers are known to engage in discriminatory personnel selection practices on the basis of such information (Acquisti & Fong, 2012). Furthermore, it is noteworthy that Facebook has recently started developing automatic methods that try to detect “embarrassing” photos and warn users before they upload them (calling this service “digital assistance”)<sup>12</sup>.

In terms of visual mining, detecting privacy-oriented concepts is a challenging task due to the fact that several of those concepts exhibit very high visual variability (e.g. extreme sports, luxurious living), while others are visually expressed by small objects that are hard to detect (e.g. smoking) or could be confused with very similar ones (e.g. drinking alcohol versus drinking soda). As a first step towards building and evaluating privacy-oriented concept models, we carried out a preliminary dataset collection around the five selected concepts using Flickr Groups and Google Image Search as sources for the collection. We submitted text queries with the concept labels to the Flickr Group search API and the Google Image search API. We then performed a manual selection of the returned Flickr Groups and images and at a second step also collected and manually filtered the images posted on the selected Flickr Groups. Table 2 reports the number images collected per concept. We recognize that this is only a small dataset and a very limited concept set, however for the first iteration of development, it serves as a first proof-of-concept test bed, which will be extended in the coming period to more concepts and more images per concept, using more scalable data collection and annotation approaches that require less manual annotation work.

| smoking | drinking | extreme sports | political beliefs | luxurious living |
|---------|----------|----------------|-------------------|------------------|
| 401     | 402      | 752            | 923               | 190              |

Table 2. Privacy-oriented image dataset.

To evaluate the potential of detecting concepts such as the above, we performed a set of preliminary experiments using an existing concept detection approach based on Approximate Lalacian Eigenmaps (ALE). We used SIFT features computed on a dense grid (based on the implementation of the VLFeat library<sup>13</sup>) and the VLAD aggregation scheme (Jegou et al., 2012), and trained SVM models (using LibSVM) on the ALE vectors produced with the

<sup>12</sup> <http://www.wired.com/2014/12/fb/> <http://www.wired.com/2014/12/fb/> (accessed on 26/12/2014)

<sup>13</sup> <http://www.vlfeat.org/> <http://www.vlfeat.org/> (accessed on 26/12/2014)

method of (Mantziou et al., 2013) using 5-fold cross-validation. The concept detection accuracy was measured using the recognition rate  $r$  defined as  $p/q$ , where  $p$  and  $q$  stand for the numbers of correctly classified data points and total data points respectively.

|       |     | <b>smoking</b> | <b>drinking</b> | <b>extr. sports</b> | <b>pol. beliefs</b> | <b>lux. living</b> |
|-------|-----|----------------|-----------------|---------------------|---------------------|--------------------|
| ALE   | $r$ | 98.3           | 97.5            | 95.5                | 96.4                | 99.7               |
| Prior | $r$ | 15.0           | 15.1            | 28.2                | 34.6                | 7.1                |

Table 3. Concept detection accuracy of ALE and prior concept probabilities.

Table 3 reports the obtained results. Despite the fact that the results indicate a very high detection accuracy (compared to the random baseline), we should be cautious with respect to their validity in more realistic scenarios. To this end, we plan to conduct tests on a larger scale and to also measure the performance of the Semfeat representation (which will be the approach that will be deployed in the system) described in subsection 2.2.

## 2.4. Implementation and usage

There are two main phases of developing the models, namely training and testing. Training can be performed offline because visual models do not change at test time, while testing needs to be performed online in order for results to be provided to the user in real time. The implementation of concept detection is done in C++. CNN feature extraction was realized using the ImageNet reference model provided along with the Caffe framework (Jia, 2013)<sup>14</sup>. After testing different layers of the deep model, the best results were obtained with the output of the last fully connected layer before classification (named fc7 in Caffe). All vectors are L2-normalized to reduce the negative effect of inter-image feature intensity variation. The resulting features have 4096 dimensions, which are considered as mid-sized vectors in the computer vision community. Features are extracted using a GTX Titan Black GPU card and the processing of an image takes less than 10 msec.

As we mentioned, concept learning is implemented using the L2-regularized L2-loss SVM from LIBLINEAR, with parallelization on a small cluster with 48 cores for faster learning. In this setting, learning 17,000 concepts takes less than 24 hours. Given that models need to be learned 5 times during the cross-validation step that adapts classification thresholds to individual concepts, this step takes less than 5 days. Regarding the deployment in the USEMP system, model training is assumed to have been completed and only the resulting models are provided to the partner in charge of integration, along with C++ wrappers that enable feature extraction with Caffe and, then, concept prediction for each new image. The concept prediction step is performed online and takes around 100 msec for 17,000 concepts on a single core, assuming that the concept models are loaded in main memory. If needed, this process can run in parallel on several cores to speed up execution.

The feature extraction wrapper can be called with the following command:

```
extract_features.bin [caffe-model] [proto-file] [caffe-layer] [tmp-leveldb] [num-batches]
[tmp-ascii] [mode] [gpu-name]
```

The L2 normalization of features is realized with:

<sup>14</sup> The Caffe reference model is publicly available at <https://github.com/BVLC/caffe/wiki/Model-Zoo> <https://github.com/BVLC/caffe/wiki/Model-Zoo> (accessed on 22/12/2014)

`normalizer_L2 [tmp-ascii] [tmp-ascii-l2] [num-dimensions] [liblinear-header]`

The concept detection wrapper can be called with the following command:

`compute_similarity [num-concepts] [num-dimensions] [concept-models] [tmp-ascii-l2] [tmp-concepts] [top-concepts]`

The commands and parameter files are explained in Table 4. This extraction assumes that the Caffe suite is already running on the server, with GPU enabled and that the same CNN model used to create concept models is readily available.

| Program                           | Description  |
|-----------------------------------|--|
| <code>extract_features.bin</code> | Binary for feature extraction provided with Caffe  |
| <code>compute_similarity</code>   | C++ binary used to compute the most salient concepts of an image.  |
| File                              | Description  |
| <code>caffe-model</code>          | CNN model used to compute features   |
| <code>proto-file</code>           | Configuration file needed to compute features  |
| <code>caffe-layer</code>          | Layer of the CNN architecture used for feature extraction. FC7 for the Caffe reference model   |
| <code>tmp-leveldb</code>          | File for output features in leveldb format. Deprecated   |
| <code>num-batches</code>          | Number of batches used for faster extraction. Typically one batch with several images.   |
| <code>tmp-ascii</code>            | File for output features in ASCII format.  |
| <code>mode</code>                 | Indicates if GPU or CPU should be used for feature extraction. GPU is strongly recommended since CPU extraction is very slow   |
| <code>gpu-name</code>             | If GPU is used and there are several available, indicates which one should be preferred. This argument is optional and points to <code>gpu-name=0</code> (i.e. default GPU). |
| <code>tmp-ascii-l2</code>         | L2 normalized version of the features written in ascii format ( <code>tmp-ascii</code> ). The normalized version is written in liblinear format.                             |
| <code>num-dimensions</code>       | Number of dimensions of each model. Assuming that fc7 layer of Caffe reference models is used, this is 4096.   |
| <code>liblinear-header</code>     | Value needed for liblinear formatting. Typically '+1'.   |
| <code>num-concepts</code>         | Number of modeled concepts. In USEMP, 17462 concepts are used.   |
| <code>concept-models</code>       | File which contains pre-computed concept models, in ASCII format.  |
| <code>tmp-concepts</code>         | Output file which stores the list of most relevant concepts for the current image. Concepts are ranked by decreasing classification score.                                   |
| <code>top-concepts</code>         | Number of most salient concept retained for each image.  |

*Table 4. Concept detection usage.*

The implementation is mature and minor effort is needed in order to integrate concept detection in the USEMP system. Consequently, this brick will be used from the very beginning of user tests which are due to start on February 2015.

## 2.5. Next steps

The obtained results are already very encouraging and future work is directed in two main directions, namely improvement of scientific quality of results and integration in USEMP. From a scientific point of view, we will focus on: (1) developing the work on privacy-related concepts and a tighter integration with the WP6 scoring framework, e.g. to target attribute

values that are of sensitive nature (e.g. smoking, drinking, political beliefs) and of commercial interest, e.g. concepts related to the IAB taxonomy<sup>15</sup>, as well as look into the broader problem of public versus private image classification (Zerr et al., 2012); (2) using more advanced CNN architectures, such as the ones devised for the ImageNet 2014 Challenge (Russakovsky et al., 2014); (3) go beyond the one-CNN-model-fits-all approach used until now and learn dedicated CNNs for different conceptual domains and (4) exploit CBIR results for vocabulary free annotation as part of D5.3 on multimedia fusion.

From a USEMP integration perspective, concept detection will be included in the architecture and used during the first user tests which are scheduled in February 2015 as part of WP8 activities. Furthermore, the concept detection module included in the system will be updated when improvements are obtained as a result of scientific advances.

---

<sup>15</sup> <http://www.iab.net/QAGInitiative/overview/taxonomy> (accessed on 26/12/2014)

## 3. Location and POI detection

In USEMP, location detection from image content is important in particular for building the user location profile, which is one of the eight core privacy dimensions determined through work done in WP4 and WP6. As mentioned in the introduction, this tool is particularly important when no useful textual annotations are available, a situation which arises often enough on social media. It is thus mainly useful for the first use case of the project as it contributes to raising user awareness about what can be automatically extracted from their data. In USEMP, we have investigated two variants of the problem: a) generic location estimation using both Visual language model and Geo-Visual Ranking (GVR) approaches, and b) location estimation as a Point Of Interest (POI) recognition problem, relying on the use of deep learning features adapted to this particular domain.

### 3.1. Related Work

Location detection from images at large scale was first presented in (Hays & Efros, 2008), where global image features were used to compute similarity between the input image (of which the location is to be predicted) and a set of images with known locations. Then, mean-shift clustering is performed on the set of top  $k$  most similar images (where  $k$  is in the order of 100) and the centroid of the largest cluster is returned as the location estimate. The obtained median error is in the range of 2000 km, which makes this method hard to use in practice. More recent work focused on the use of local images descriptors, such as SIFT (Crandall et al., 2009), and their aggregated versions (Choi & Li, 2014) in order to improve location prediction accuracy. Despite important progress was achieved compared to (Hays & Efros, 2008), performance remains relatively low, with under 4% of images placed at <1 km of their true location in (Choi & Li, 2014). It is noteworthy that the aforementioned methods rely on the use of Nearest-Neighbour (NN) retrieval schemes (such as the ones described in Section 2), and hence on the assumption that some very similar image with known location exists in an index that is built for this purpose. When such an assumption is violated, one may expect such methods to fail to produce a reasonable estimate. This is illustrated in Figure 1.

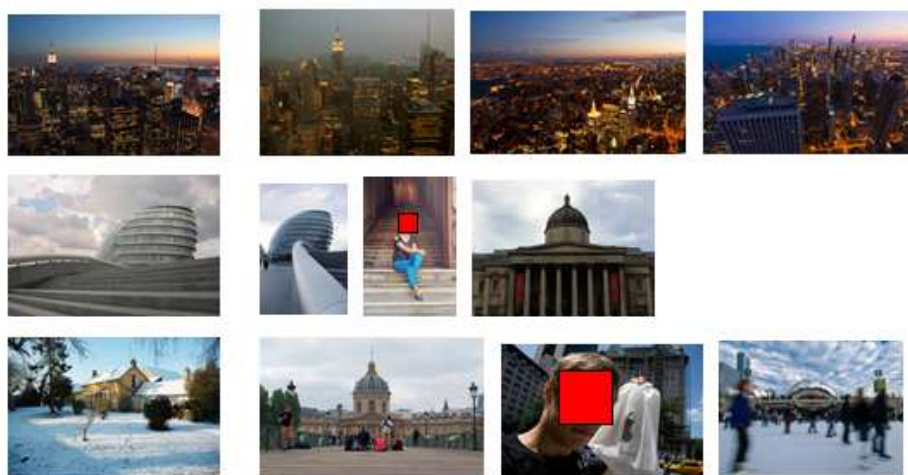


Figure 5. Examples where NN-based location estimation techniques work (first two rows) and fail (last row). In each row, the leftmost image is the “unknown” (query) image and the three images to the right are the top-3 most similar neighbors (using SURF+VLAD features to compute similarity).



Building on the recent success of deep learning based image representations (i.e. CNN features), location detection was cast as a POI detection task and experiments were carried out notably with standard POI retrieval datasets such as Oxford5k (Philbin et al., 2007) and INRIA Holidays (Jégou et al., 2008). Razavian et al. (2014) use a crude approach in which off-the-shelf generic CNN features are associated to a spatial search algorithm which splits the image in patches and then compares pairs of patches to improve performance and report MAP=68% on Oxford5k and MAP=84.3% on Holidays. Babenko et al. (2014) perform domain adaptation through the use of domain related data (i.e. POI images) for learning CNN features. A part of POI images are manually checked in order to retain only POIs whose associated images include a small quantity of noise. Babenko et al. (2014) report MAP=54.5 on Oxford5k and MAP=79.3 on Holidays. While interesting in terms of scalability due to their smaller dimensionality, the results obtained with CNN-based approaches still lag behind those obtained with high dimension representations of SIFTs on these databases.

## 3.2. Method description

Due to the complexity of the problem at hand and the importance of location as a privacy dimension, we explored two different variants of the problem: a) generic location estimation with the use of Visual language model and Geo-Visual Ranking (GVR), and b) POI recognition using CNN representations and transfer learning.

### 3.2.1. Visual language model and Geo-Visual Ranking (GVR)

We experimented with two different approaches for generic location estimation. The first approach explores the potential of using a “visual” language model, while the second one is based on the Geo-Visual Ranking (GVR) method by (Li et al., 2013).

**Visual language model:** This approach was motivated by the tag-based geolocation approach of (Kordopatis-Zilos et al., 2014). The general idea is to form a similar framework but instead of the textual user-assigned tags to make use of visual words and of a “visual” language model. In brief, the earth surface is divided into rectangular cells of approximately 1km side length. Given such a grid, an approach used to build a corresponding language model is described in (Popescu & Ballas, 2012). More specifically, the most probable cell for a query (test) image is retrieved based on the respective tag probabilities. A tag probability in a particular cell is calculated as the total number of different Flickr users that used the tag inside the cell, divided with the total count of different users in all cells. All this information is available from the training dataset (in the case of MediaEval, this is distributed by the organizers, but other publicly available geolocated datasets can be used<sup>16</sup>).

Two different language models were built using visual codebook sizes of 10k and 100k visual words<sup>17</sup>. The method used to estimate the visual words was under examination since different approaches produced different tags and thus results. All approaches used the popular Speeded Up Robust Features (SURF) (Bay et al., 2008). Because the number of SURF features per image is typically much higher than the average number of textual tags

---

<sup>16</sup> See also the related discussion in D5.1.

<sup>17</sup> The concept of *visual word* is the counterpart of a word (term) in text but extracted from an image. One may think of a visual word as a distinct visual pattern. In order to make the representation of new images compact, we typically create visual vocabularies (codebooks) of standard size and map the visual patterns of the new images to the closest possible patterns of these codebooks.

(for example 1000 versus 10), we kept only some of the SURF descriptors for each image. We experimented with retaining a fixed number of words  $k$ , and this meant keeping the words with higher frequency. The problem with this simple approach is that a lot of words have similar frequency and it is therefore not possible to sort them in a deterministic way. A second approach was to keep all words that have frequency above a certain threshold, typically above  $n=2$  or 3 occurrences. Another issue that arose here was that some images did not have any word with at least  $n$  occurrences and thus these images would not be associated to any word at all. As a final solution this approach was combined with a fallback parameter for those images that do not collect any word. For those images the words assigned would be the ones with at least  $n-1$  votes, provided that  $n-1 > 1$ .

**GVR-based method:** This approach is based on VLAD+SURF features and the IVFADC indexing scheme by (Jegou et al, 2011). Four visual vocabularies of  $k=128$  centroids were used to aggregate the SURF features and produce the VLAD vectors. Next a joint dimensionality reduction using PCA and whitening (Jegou et al, 2012) was applied to these vectors to reduce their dimensionality from  $D=32768$  to 1024. The next step involved coarse quantization using a codebook with 8192 centroids and product quantization on the residual vectors using an  $8 \times 10$  scheme (Jegou et al, 2011). The above scheme provided an efficient way to retrieve nearest neighbors for each query image.

To improve the robustness of the results in comparison with using just the 1-NN, we retrieve multiple NNs for each query image. The number  $m$  of results to be retrieved is a parameter of the problem. Having retrieved the top  $m$  NNs, we group them into  $M$  geographical clusters of maximum radius  $r_{geo}$  creating  $M$ . Each of these clusters represents a potential candidate for the query image. The criterion with which we are going to decide which candidate is to be chosen is the main difference in the approaches mentioned before. As it has already been mentioned in (Hays et al, 2008), one criterion is the number of cluster members, after rejecting clusters smaller than a user-defined threshold, while (Li et al, 2013) chose to use the sum of visual similarity between neighbors and the query for this purpose. We have experimented with various methods:

1. The mean distance score between the cluster members and the query. This was our first attempt which proved to be unreliable.
2. The median distance score between the cluster members and the query instead. The intention was to be less prone to outliers but the results turned out to be similar.
3. The cluster's size (max number of members) was found to perform better than the previous criteria. Of course this method could just fail in cases where the true nearest neighbors was just a single candidate.
4. The sum of inverted distance scores between the cluster members and the query, which further improved results.
5. A hybrid version of the first and third approaches where a formula combining both criteria was introduced. If  $d_{mean}$  was the mean distance of each cluster and  $n$  the cluster size then the new distance score was calculated as  $d_{mean} (a-n)/a$ , where  $a$  is a weight parameter. The idea behind this score is to take into consideration both the mean distance score which gives a rough estimation of cluster distance and at the same time also the cluster size which implies higher confidence in the retrieved set of NNs. This method seemed to further improve performance over the sum approach.

### 3.2.2. POI recognition based on CNN

In this approach, we construct domain adapted CNN features for POI recognition which is a domain adaptation of the training step of our concept detection work. The work closest to ours was presented in (Babenko et al., 2014), where authors also train a neural network dedicated to the tourism domain. However, their approach differs from ours in several important aspects: (1) the amount of employed supervision by them is significantly higher; (2) no image re-ranking is tested to reduce the influence of noise; (3) training is done on an highly unbalanced dataset, a modeling choice which is likely to downgrade results; (4) initialization is done with the weights learned from ImageNet; (5) the data sources are different - Yandex in (Babenko et al., 2014) versus Flickr here. Similar to (Babenko et al., 2014) and (Razavian et al., 2014), we study domain transfer – i.e. CNN features are learned on a training set with a limited number of POIs and then tested on datasets which include POIs and locations that are not included in the training set.

Problem (1) is central because it limits the scalability of the learning process. Put simply, if one is able to train efficiently without pre-processing of Web images associated to POIs or with automatic re-ranking of the images, the annotation effort needed for creating the training set becomes negligible. Consequently, we test three feature configurations:

- POI-CNN-N - direct use of Flickr image sets, without re-ranking.
- POI-CNN-A - automatic re-ranking based on the correlations between the images associated to each POI.
- POI-CNN-W - weakly supervised re-ranking in which candidate images are compared against a limited amount of manual annotations.

POI-CNN-A and POI-CNN-W strategies are implemented using a linear SVM-based re-ranking method (Schroff et al., 2011), applied to a generic CNN description of image content. We train CNN models using POI-CNN-N, POI-CNN-A and POI-CNN-W strategies. To ensure comparability, the same set of POIs is used in all three cases, with the top ranked 1000 images used for each POI. Training is done with Caffe (Jia, 2013). Furthermore, PCA is applied to features in order to obtain more compact representations.

Following (Razavian et al., 2014), we also propose feature augmentation (A) in order to further adapt features to the location retrieval task. Augmentation includes three main operations which can be exploited together or separately:

- Spatial search – to cope with object location and scale variability, image patches are extracted at different scales from the query and the test set. Then, all query patches are compared to test set patches to rank patch pair distance. Finally, the similarity between two images is given by the average of the minimum distances found for each query patch.
- Feature augmentation – pipeline including: L2-normalization, PCA-compression, feature whitening and renormalization.

Query expansion – pseudo-relevance feedback method in which an enriched representation of the query is built after a first retrieval process before launching a new retrieval with the updated version as query.



### 3.3. Evaluation and testing

The experimental evaluation of our methods is also organized in two main parts following the two-level treatment of the problem: a) evaluation of the generic location estimation methods described in subsection 3.2.1 on the dataset provided by the MediaEval Placing Task 2014, and b) evaluation of the POI/location recognition methods of subsection 3.2.2 on the Oxford5k, Holidays and Div150Cred datasets.

#### 3.3.1. Location Estimation on MediaEval Placing Task 2014

A number of experiments were carried out on the small test set (25,500 images) of the MediaEval Placing Task 2014 dataset (Choi et al., 2014). The training set used by all methods was 1M images randomly sampled from the corresponding 5M training set.

Table 5. Location estimation results using the visual language model approach. Table 5 presents the experimental results of the visual language model approach. Overall, the location estimation performance seems to be very poor across all settings. We observe that selecting visual words based on the frequency threshold is somewhat better compared to ranking by frequency and selecting the top 30 or 40. We also note that the finer the cell size (for the cells of the language model grid) the worse the results. Given that the best obtained accuracy was 2.57% for the range of 10km, we decided that this approach is not suitable for the problem at hand and hence to not invest further in it.

|                  | Cell size $10^{-c}$ (km) |        |        |                        |        |        |                        |        |        |
|------------------|--------------------------|--------|--------|------------------------|--------|--------|------------------------|--------|--------|
|                  | c = 1                    | c = 2  | c = 3  | c = 1                  | c = 2  | c = 3  | c = 1                  | c = 2  | c = 3  |
|                  | Codebook size 10k        |        |        |                        |        |        |                        |        |        |
|                  | Words with frequency > 3 |        |        | 30 most frequent words |        |        | 40 most frequent words |        |        |
| Accuracy         |                          |        |        |                        |        |        |                        |        |        |
| 100 m            | 0.0                      | 0.004  | 0.06   | 0.0                    | 0.004  | 0.06   | 0.0                    | 0.004  | 0.07   |
| 1 km             | 0.12                     | 0.41   | 0.27   | 0.13                   | 0.41   | 0.31   | 0.13                   | 0.40   | 0.29   |
| 10 km            | 2.57                     | 1.83   | 1.38   | 2.63                   | 2.02   | 1.34   | 2.63                   | 1.99   | 1.31   |
| 100 km           | 4.59                     | 2.78   | 2.07   | 4.67                   | 3.38   | 1.85   | 4.67                   | 3.36   | 1.76   |
| 1000 km          | 23.70                    | 24.07  | 23.23  | 24.46                  | 25.70  | 26.05  | 24.46                  | 25.85  | 26.92  |
| 10000 km         | 92.55                    | 91.31  | 90.26  | 92.98                  | 92.63  | 92.23  | 92.99                  | 92.78  | 92.55  |
| Median error (m) | 5844.9                   | 5897.2 | 6017.1 | 5844.5                 | 5898.1 | 6020.9 | 5871.1                 | 5865.6 | 6085.3 |

Table 5. Location estimation results using the visual language model approach.

We continued with a number of experiments on variants of the GVR approach. For the sake of brevity, we only present the results of Table 6, which are the highest for the radii of 1km and 10km.

| $k$              | 100    | 100    | 300    | 400    | 400    | 500    | 500    | 600    |
|------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| $r_{geo}$        | 0.1    | 1.0    | 1.0    | 10.0   | 1.0    | 1.0    | 1.0    | 1.0    |
| a                | 10     | 20     | 10     | 10     | 20     | a = 10 | a = 20 | a = 10 |
| 100 m            | 0,83   | 0,71   | 0,63   | 0,62   | 0,65   | 0,58   | 0,62   | 0,56   |
| 1 km             | 1,61   | 1,68   | 1,62   | 1,60   | 1,60   | 1,53   | 1,56   | 1,49   |
| 10 km            | 2,53   | 3,03   | 3,19   | 3,18   | 3,15   | 3,24   | 3,24   | 3,20   |
| 100 km           | 3,32   | 3,96   | 4,41   | 4,34   | 4,32   | 4,45   | 4,45   | 4,44   |
| 1000 km          | 14,46  | 15,70  | 17,07  | 17,29  | 17,04  | 17,38  | 17,16  | 17,36  |
| 10000 km         | 81,54  | 83,16  | 84,85  | 85,04  | 84,58  | 85,38  | 84,89  | 85,78  |
| Median error (m) | 6661,0 | 6261,3 | 5911,3 | 5844,6 | 5875,0 | 5765,9 | 5816,3 | 5750,9 |

Table 6. Location estimation results using the variants of the GVR approach.

Analyzing all obtained results, we could conclude the following:

- The size of the training set has a noticeable effect on the performance. Results using the same method and parameters but different training set size can vary in performance from 0.28% gain (or 84.85% relative gain) in 100m and single 1-NN retrieval to 0.66% gain (or 24.62% relative gain) in 10km radius.
- The parameter controlling the geo-neighbor radius ( $r_{geo}$ ) has an important effect on the results we get for different evaluation radii. This parameter controls the size of the clusters since this in fact represents the cluster's geographical spread. The interesting part is that we can get higher geo-location performance using smaller  $r_{geo}$  in low evaluation radii (100m for instance) but with the cost of lower performance for larger evaluation radii. A good compromise seems to be  $r_{geo}=1$ .
- The number of top NN images  $k$  is also a parameter which affects the results. Using too many candidates means it is more likely to include images that are relatively close to the query but it is also harder to identify the correct ones. There seems to exist a trade-off at roughly  $k=500$  candidates.
- Finally, the weight  $a$  is controlling the tradeoff between a single NN score and a mean cluster score. The first one gives more exact and also unstable results while the second gives less accurate but more robust results. Values around  $a=10$  and  $a=15$  seems to produce the best results.

While the best attained accuracy (3.24% for the radius of 10km) was considerably higher compared to the one achieved by the language model approach, it is still considered to be low and insufficient for integration in the USEMP system, as it is expected to lead to frequent mispredictions of location, which could be frustrating for the end users. In conclusion, the conducted experiments with both generic location estimation approaches have not resulted in satisfactory performance. To this end, the POI and location recognition approaches of subsection 3.2.2 were subsequently tested.

### 3.3.2. POI-location recognition on Oxford5k, Holidays and Div150Cred

Evaluation is carried out using the Oxford5k (Philbin et al., 2007) and INRIA Holidays (Jégou et al., 2008), the two standard datasets used by the multimedia retrieval community. While interesting, these two collections have rather low complexity and size and we also test our methods using the Div150Cred dataset, introduced as part of the MediaEval 2014 Diverse Social Images Task<sup>18</sup>. This dataset includes 150 POIs of different types (churches, museums, squares, parks etc.). It is thus more challenging than Oxford5k and Holidays both in terms of image diversity but also in terms of scale. The evaluation measure used for all datasets is Mean Average Precision (MAP), with higher scores showing better performance. For Oxford5k and Holidays, we report different state-of-the-art results, including the best ones are aware of. To compare our approach with existing work on Div150Cred, we compute results with bags-of-visual words with a very large vocabulary (named BoVW), a simpler implementation of (Arandjelovic & Zisserman, 2012) and also with the Caffe reference model, an off-the-shelf CNN feature extractor (Jia, 2013).

The results of Table 7 demonstrate that domain adaptation of CNN learning is useful since the results are clearly better with the proposed method than with any of the existing CNN methods, including (Babenko et al., 2014) and (Razavian et al., 2014). More importantly, differences are highest for the Div150Cred dataset, i.e. the one which is closest to a use of

---

<sup>18</sup> <http://www.multimediaeval.org/mediaeval2014/diverseimages2014/> (consulted on 27/12/2014)

the method in real settings, and very good results are obtained in all configurations, including the compressed features which employ PCA with 512 dimensions. When compared to state-of-the-art results on Oxford (Arandjelovic & Zisserman, 2012), our methods still lag behind (76.9% versus 92.7%). On Holidays, it has similar performance to that reported by (Tolias et al., 2013), 87.5% versus 88%. In contrast, the comparison between BoVW, a slightly simplified version of (Arandjelovic & Zisserman, 2012), and our methods on Div150Cred is clearly favorable to the later, 2.7% versus 9.8% - 12.5%.

| Method                           | Feature dimension | Feature augm. |   |    | Oxford5k | Holidays | Div150Cred |
|----------------------------------|-------------------|---------------|---|----|----------|----------|------------|
|                                  |                   | S             | A | QE |          |          |            |
| Inverted index approaches        |                   |               |   |    |          |          |            |
| (Arandjelovic & Zisserman, 2012) | 1M                |               |   | x  | 92.9     | N/A      | N/A        |
| (Tolias et al., 2013)            | 65K               |               |   |    | 83.8     | 88       | N/A        |
| BoVW                             | 1M                |               |   |    | 72.9     | 51.2     | 2.7        |
| Existing CNN methods             |                   |               |   |    |          |          |            |
| (Babenko et al., 2014)           | 4k                |               |   |    | 54.5     | 79.3     | N/A        |
| Caffe (Jia, 2013)                | 4k                |               |   |    | 38.2     | 73       | 3.7        |
| Caffe (Jia, 2013)                | 3 – 16k           | x             | x |    | 71.2     | 87.1     | 6.4        |
| Proposed CNN methods             |                   |               |   |    |          |          |            |
| POI-CNN-N                        | 4K                |               |   |    | 65.7     | 77       | 10.3       |
| POI-CNN-A                        | 4K                |               |   |    | 64.6     | 76.5     | 9.8        |
| POI-CNN-W                        | 4K                |               |   |    | 67.1     | 76.3     | 9.8        |
| POI-CNN-N <sub>512</sub>         | 512               |               |   |    | 68.6     | 78       | 11.9       |
| POI-CNN-A <sub>512</sub>         | 512               |               |   |    | 66.4     | 77.9     | 10.9       |
| POI-CNN-W <sub>512</sub>         | 512               |               |   |    | 69.3     | 77.7     | 10.8       |
| POI-CNN-N <sub>512</sub>         | 3 – 16k           | x             | x |    | 76.9     | 87.5     | 12.5       |
| POI-CNN-A <sub>512</sub>         | 3 – 16k           | x             | x |    | 76.7     | 86.9     | 11.5       |
| POI-CNN-W <sub>512</sub>         | 3 – 16k           | x             | x |    | 76.7     | 86.6     | 11.8       |

Table 7. POI detection experimental results.

- An unexpected finding has been that training without re-ranking (POI-CNN-N), with automatic re-ranking (POI-CNN-A) and with weakly supervised re-ranking (POI-CNN-W) resulted in similar performance levels, with POI-CNN-N being slightly superior on Div150Cred. A possible explanation of these results is that, when training CNNs, the overall diversity of the training set has a more important role than the presence of a reasonable amount of noise. This finding has important implications because, if replicated in other domains, it would indicate that it is possible

## 3.4. Implementation and usage

### 3.4.1. Visual language model and Geo-Visual Ranking (GVR)

Here, we present the basic sequence of commands that are necessary to construct location models and prepare location prediction files. For the sake of brevity, further details regarding the meaning of the arguments and parameters are provided in the respective readme files accompanying this report.

**Visual language model:** For the creation of a location model for the training set, the following needs to be executed (assuming a JRE is installed):

```
java -cp multimedia-indexing.jar gr.iti.mklab.visual.examples.UrlBoWIndexingMT temp/
files/metafile_train.txt train/BoW false 3 2 codebooks/SURF_64_c10000_N3327193_BoW.txt
false false - surf GreaterThanK 3 10000 false 0 100
```

This execution does not make use of pre-clustering. To use pre-clustering, run:

```
java -cp multimedia-indexing.jar gr.iti.mklab.visual.examples.UrlBoWIndexingMT temp/
files/metafile_train.txt train/BoW false 3 2
codebooks/SURF_64_c10000_N3327193_BoW_initCen10.txt true false - surf GreaterThanK 3
10000 false 0 100
```

A similar command should be executed for the test set, as a preparatory step to prepare the corresponding bag-of-words model.

To prepare a location estimation file, the following should be executed:

```
java -cp multimedia-indexing.jar gr.iti.mklab.distanse.AccuracyResultsV2
files/NameCoords_LatLon_test.ser
results/Results_IVFPQ_w0.1_r1_k5_raw3_GVR_MT_weightedMean_r1.0 25500
```

**Geo-Visual Ranking:** As a first step at training time, an IVFPQ index needs to be created using the following command:

```
java -cp multimedia-indexing.jar gr.iti.mklab.visual.examples.UrlIndexingMT temp
files/metafile_train.txt index false 3 2
codebooks/surf_l2_128c_0.csv,codebooks/surf_l2_128c_1.csv,codebooks/surf_l2_128c_2.csv,code
books/surf_l2_128c_3.csv 128,128,128,128 files/pca_surf_4x128_32768to1024.txt 1024 60 0 100
false 256 8192 files/pq_1024_128x8_rp_ivf_8192k_SURF.csv
files/qcoarse_SURF_1024d_8192k.csv
```

Also a serialized HashMap of the training set coordinates should be created:

```
java -cp multimedia-indexing.jar gr.iti.mklab.tag.TagManipulationUtils
files/metafile_train.txt files/ files/NameCoords_LatLon_train.ser String
```

Finally, a serialized HashMap of the all test set coordinates should be created:

```
java -cp multimedia-indexing.jar gr.iti.mklab.tag.TagManipulationUtils
files/All_test_metadata files/ files/NameCoords_LatLon_test.ser Coords
```

The location estimation is performed using the same command as for the visual language model.

### 3.4.2. POI recognition based on CNN

As previously mentioned, POI detection represents a domain adaptation of the training phase of concept detection, which consists in the use of domain relevant concepts to learn a dedicated CNN model. Training is done using the Caffe framework, with a dedicated model learned for each configuration (POI-CNN-N, POI-CNN-A, POI-CNN-W), and is performed offline. It takes approximately 10 days for each configuration on an NVIDIA K20 GPU. CNN feature extraction is realized with the POI-CNN-N model using the Caffe framework (Jia, 2013) and takes less than 20 msec on a K20 GPU. Testing is done using a nearest neighbor approach implemented in C++ and takes approximately 10 seconds for a 1 million images dataset on a single core. However, it can easily be done in parallel on several cores to speed-up execution.

The feature extraction wrapper can be called with the following command:

`extract_features.bin` [`caffe-model`] [`proto-file`] [`caffe-layer`] [`tmp-leveldb`] [`num-batches`]  
 [`tmp-ascii`] [`mode`] [`gpu-name`]

The L2 normalization of features is realized with:

`normalizer_L2` [`tmp-ascii`] [`tmp-ascii-l2`] [`num-dimensions`] [`liblinear-header`]

The location detection wrapper can be called with the following command:

`compute_similarity_location` [`ref-size`] [`num-dimensions`] [`ref-features`] [`tmp-ascii-l2`]  
 [`tmp-similar`] [`top-similar`]

The commands and parameter files are explained in Table 8. This extraction assumes that the Caffe suite is already running on the server, with GPU enabled and that the same CNN model used to create concept models is readily available.

| Program                                  | Description  |
|--|--|
| <code>extract_features.bin</code>        | Binary for feature extraction provided with Caffe. The feature extraction command is the same as for concept detection.  |
| <code>compute_similarity_location</code> | C++ binary used to compute the most similar images from a reference dataset given a query image. This step reuses the same program as the concept detection but with different inputs. |
| File                                     | Description  |
| <code>caffe-model</code>                 | CNN model used to compute features   |
| <code>proto-file</code>                  | Configuration file needed to compute features  |
| <code>caffe-layer</code>                 | Layer of the CNN architecture used for feature extraction. FC7 for the Caffe reference model   |
| <code>tmp-leveldb</code>                 | File for output features in leveldb format. Deprecated   |
| <code>num-batches</code>                 | Number of batches used for faster extraction. Typically one batch with several images.   |
| <code>tmp-ascii</code>                   | File for output features in ASCII format.  |
| <code>mode</code>                        | Indicates if GPU or CPU should be used for feature extraction. GPU is strongly recommended since CPU extraction is very slow   |
| <code>gpu-name</code>                    | If GPU is used and there are several available, indicates which one should be preferred. This argument is optional and points to <code>gpu-name=0</code> (i.e. default GPU).           |
| <code>tmp-ascii-l2</code>                | L2 normalized version of the features written in ascii format ( <code>tmp-ascii</code> ). The normalized version is written in liblinear format.                                       |
| <code>num-dimensions</code>              | Number of dimensions of each model. Assuming that fc7 layer of Caffe reference models is used, this is 4096.   |
| <code>liblinear-header</code>            | Value needed for liblinear formatting. Typically '+1'.   |
| <code>ref-size</code>                    | Size of the reference dataset from which similar images are extracted (i.e. number of images)  |
| <code>ref-features</code>                | Pre-computed features for the reference dataset.   |
| <code>tmp-similar</code>                 | Output file which stores the list of most relevant concepts for the current image. Similar images are ranked by decreasing similarity score.   |
| <code>top-similar</code>                 | Number of most similar images retained for each query image.   |

Table 8. Usage of POI-based location detection.

### 3.5. Next steps

The experimentation conducted so far has demonstrated that creating POI/location-specific models using CNN features is a more practical approach compared to NN-based methods for location estimation. Hence, future work will focus on refining and maturing the CNN-based approach with the goal of integrating it in the USEMP system.

More specifically, from a scientific point of view, future work includes improving location detection via training with larger amounts of Web data, exploiting more complex CNN architectures and altering these architectures to better account for the spatial context of image pixels. From a USEMP integration point of view, image based location detection will shortly be integrated with text based detection presented in D5.1 into a multimodal location detection framework as part of D5.3. Further downstream, this framework will be provided as a simple to use library for integration by the USEMP system developed as part of WP7.



## 4.Face detection and recognition

---

Faces are a core component of our identity. Data relating to facial characteristics is legally treated as a special category of sensitive information (“biometric data”) in European data protection law. A lot of research has been devoted to this problem for biometric applications but also for the development of Web services. In USEMP, face detection and recognition tools can be used to give feedback to the user about his/her occurrence in OSN images. More interestingly, it can be combined with results of other visual mining modules in order to associate a user with brands, products and other concepts that are detected on images depicting the user.

### 4.1. Related Work

Non-parametric Nearest-Neighbor (NN) based classifiers, which had been under-valuated for a while, are recently receiving a considerable interest to reduce the gap between their performance and that of parametric classifiers (Behmo et al., 2010; Boiman et al., 2008; McCann & Lowe, 2012). The interest devoted to these machine learning algorithms is due to their interesting properties in terms of computational complexities and scalability to large scale classification tasks. They can easily handle a large number of classes and are well suited to deal with dynamically changing (especially growing) datasets, where the arduous parameter retraining task required by parametric classifiers is no longer necessary. Authors of (Boiman et al., 2008) argued that the main drawbacks of common practices that had been used with NN classifiers included a) the quantization step to BoVW signatures, and b) the degradation of descriptor discriminative power due to information loss in quantization. The latter is less appropriate to large diversified classes, especially when a small number of labeled images relative to the class complexity are available for training. Under the Naive-Bayes assumption, a new NN based classification algorithm was introduced by the same authors, enabling to avoid the quantization step, inherent in the BoVW-like signatures. The classifier decision relies therefore on Image-to-Class distances computed within the space of local descriptors. Their proposed algorithm, called Naive-Bayes Nearest-Neighbor (NBNN), despite being very simple, leads to a classification performance that ranks among the top leading learning-based image classifiers. Authors of (McCann & Lowe, 2012) recently propose a local-NBNN (LNBNN) that exploits a locality constraint in the feature space to improve NBNN in terms of classification accuracy, runtime and ability to manage even more classes. Their work is inspired from recent advances in BoVW signature generation approaches that show the importance of the locality constraint to increase classification accuracy (Liu et al., 2011; Wang et al., 2010; Yu et al., 2009).

### 4.2. Method description

To detect one or several faces into an image, we use the Viola-jones face detector of OpenCV with a Haar Cascade (frontal face only). Then, SIFT local features (Lowe, 2004) are extracted into the image and feed a non-parametric classifier directly inspired from the approach of (McCann & Lowe, 2012), namely local Naïve Bayes Nearest Neighbors (LNBNN). During the testing phase, the images are processed similarly to the training phase and we use the “learned” model (there is no actual learning since the model is only the concatenation of local signatures) to detect the closest face. Knowing its label (e.g. person name), we return it to the user.

### 4.3. Evaluation and testing

We manually collected a small dataset of 18 famous people, with 10-20 images per person. This was collected with the help of Google Image search, resulting in 254 images in total but the face was not detected in four of them, which were then removed from the learning base. Learning is processed in less than 2 minutes on a single core (Xeon X5650@2.67 GHz).

For testing, we considered 10 persons belonging to the learning database and 2 more that were not in the base. We collected some videos of these persons and segmented them (temporally) manually. Each key frame of the sequence (1217 in total) was tested independently then the results were improved using the temporal coherence (i.e. relying on neighboring key frames to make recognition more robust). Detailed results are given (“unknown were not in the learning database) in Table 9.

| Person name            | # videos (test) | # good (classif. rate) |
|------------------------|-----------------|------------------------|
| Cameron Diaz           | 4               | 4 (100%)               |
| Charlize Theron        | 3               | 2 (67%)                |
| D. Letterman           | 1               | 1 (100%)               |
| Emma Watson            | 1               | 1 (100%)               |
| Georges Clooney        | 15              | 8 (55%)                |
| Glenn Close            | 3               | 2 (66%)                |
| Katie Holmes           | 1               | 1 (100%)               |
| Meryl Streep           | 5               | 5 (100%)               |
| Unknown (N. Sarkozy)   | 1               |                        |
| Paris Hilton           | 1               | 1 (100%)               |
| D. Pujadas             | 1               | 1 (100%)               |
| Unknown (Teri Hatcher) | 2               |                        |

Table 9. Face recognition experimental results.

### 4.4. Implementation and usage

The module has been implemented in C++ and is ready for integration. To this end, we provide a library compiled with GCC under an x86\_64 linux system. The only dependence to an external library is that to OpenCV 2.4.9 to be able to read an image (the required library is provided as well). To learn a base, the following command should be used:

```
Face_reco labelled_images.lst base_name --learn
```

where labelled\_images.lst is an ASCII file with two columns: first is the label (e.g. person name) and the second the path to the corresponding image. It then creates an index file base\_name.

For testing, the following command should be used:

```
Face_reco testing_images.lst base_name
```

where testing\_images.lst is an ASCII file the path to the testing images (one per line). It returns the label.

### 4.5. Next steps

While interesting, the results reported here seem to be far below those reported very recently (Taigman et al., 2014). Given the utility of face recognition in the USEMP system, we intend to conduct further research in this direction and explore the potential of an implementation based on CNN dedicated to face recognition.



## 5. Logo and product recognition

---

Logo and product recognition are useful in order to create consumer profiles for users, one of the core privacy dimensions determined in WP4 and WP6. Through automatic recognition, a profile will include specific brands and products that are likely to be of interest for a particular user. Visual mining can then be combined with the results of text mining (named entity recognition to be integrated in T5.1) and with user likes (analyzed in T6.1) to obtain a more complete profile. Logo and product recognition is mainly useful for the first use case of the project as it contributes to determining the value of the user's personal data. The first experiments carried out for logo recognition were done by adapting an existing method.

### 5.1. Related Work

Philbin et al. (2007) proposed an image mining methodology that represents SIFT-based local features in a very high dimensional space. Since these features are sparse, they can be represented using an inverted index structure in order to speed up retrieval. In a follow-up work, Arandjelovic & Zisserman (2012) proposed a different feature augmentation method, including query expansion and geometric verification, which we also implemented in our system. Romberg et al. (2011) adapted local features for logo detection and increased the importance of the spatial layout of local features and of the composition of basic spatial structures, such as edges and triangles. A cascaded index was also implemented in order to speed-up logo recognition.

### 5.2. Method description

We chose to use the general visual mining methodology proposed by Philbin et al. (2007). As we mentioned, it consists of a classical inverted index file approach relying on local descriptors. The local descriptors are augmented according to the proposal of (Arandjelovic et Zisserman, 2012). We computed a codebook of size 1M (one million) with an approximate K-means algorithm (Philbin et al., 2007). The cosine similarity is used to search similar images in the inverted index.

### 5.3. Evaluation and testing

We conducted an experiment to evaluate the method using FlickrLogos-32, a publicly available dataset<sup>19</sup>, which facilitates comparison between our approach and the state of the art. The dataset contains photos showing brand logos and is meant for the evaluation of logo retrieval and multi-class logo detection/recognition systems on real-world images. We collected logos of 32 different logo brands by downloading them from Flickr. All logos have an approximately planar surface. The retrieved images were inspected manually to ensure that the specific logo is actually shown. The whole dataset is split into three disjoint subsets, each containing images of all 32 classes. The training set consists of 10 images that were hand-picked such that these consistently show a single logo under various views with as little background clutter as possible. The other two partitions Pv (validation set) and Pt (test set = query set) contain 30 images per class. Unlike the training set, these images contain at least

---

<sup>19</sup> Available at <http://www.multimedia-computing.de/flickrlogos/> <http://www.multimedia-computing.de/flickrlogos> (accessed on 23/12/2014)

one instance of a logo but in several cases multiple instances. Both partitions Pv, and Pt include another 3000 images downloaded from Flickr with the queries "building", "nature", "people" and "friends". These images are the negative images and complete the dataset.

We evaluated our method using a *retrieval* evaluation approach:

- Images from the training and validation set are indexed, including non-logo ones (4280 images)
- The 960 images of the query set (logos) are used as queries.
- The mean average precision (MAP) is used to measure the detection accuracy.

Our method resulted into a MAP of 0.48 while the best reported result in the literature on this benchmark is 0.55 (Romberg et al., 2011). This performance gap is mainly explained by the fact that (Romberg et al., 2011) fine-tune their features for logo detection while we adapt standard high dimensional image features for this particular task.

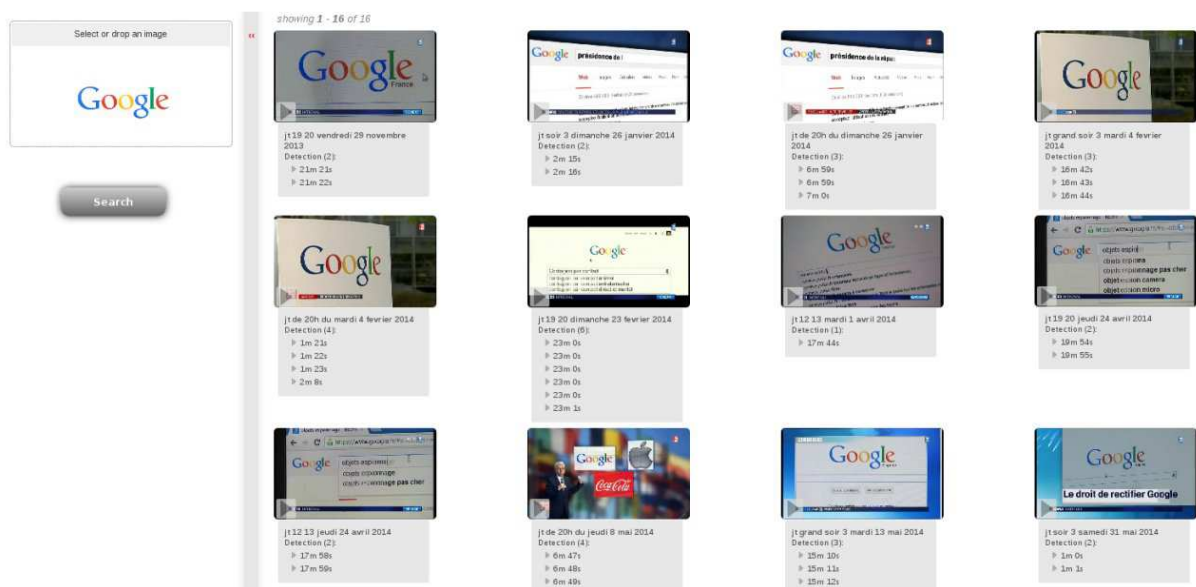


Figure 6. Illustration of logo detection process. The query image is to the left of the figure and its nearest neighbours to the right.

An illustration of logo detection is presented in Figure 6. Illustration of logo detection process. The query image is to the left of the figure and its nearest neighbours to the right., with an image in which the Google logo occupies a significant part of the query image. The same logo appears in similar images, many times embedded in more complex images. A qualitative examination of logo detection results shows that, as expected, the method fails when the logo is too small, partially obscured or under poor illumination conditions.

## 5.4. Implementation and usage

The module has been implemented in C++ and is ready for integration. To this purpose, we provide a library compiled with GCC under the x86\_64 linux system. The only dependence to an external library is that to OpenCV 2.4.9 to be able to read an image (the required library is provided as well).

To test which logos are found into the database, the following command should be used:

```
getdescr queries.lst HESSIAFF -o G_BINARY -p param.flickrlogo32 query.tmp
search_inindex query.tmp flickrlogos.ii flickrlogos.p8r.AK1000000.cb
```

where queries.lst contain the path of all test images, and the rest of parameters denote the rest of the necessary files. A further documentation of the method usage is given in Table 10.

| <b>Program</b>               | <b>Description</b>                      |
|------------------------------|---|
| Getdescr                     | Compute the signature                   |
| Search_invidex               | Search into an inverted file            |
| <b>File</b>                  | <b>Description</b>                      |
| param.flickrlogo32           | Parameter file to compute the signature |
| flickrlogos.ii               | The inverted index                      |
| flickrlogos.p8r.AK1000000.cb | The codebook                            |

*Table 10. Logo recognition usage.*

## 5.5. Next steps

This module will be integrated into the USEMP system, with a model able to recognize (at least) the 500 popular logos (trademarks), with popularity determined by the number of associated images in a Web search engine. This number will be definitively fixed after the first pilot studies (WP8) and further exchanges with WP2 and WP7 partners. We intend to continue the development of the method in order to improve its performance incrementally. Following work done for general concept detection and location detection, we will equally explore the possibility to recognize logos using CNN representations. To do this, we will train dedicated models over semi-automatically cleaned web data.

Another important line of work concerns product recognition, which is considered as a subcase of concept detection. In this case, a dedicated CNN model will be learned for a fixed training set made of product images and then exploited through feature transfer in order to recognize a larger number of products.

## 6. Near-duplicate detection

---

Near-duplicate detection is useful in different USEMP related tasks, including the estimation of image location (based on the locations of known similar images) and the identification of similar users (i.e. users who share similar content) that can be further leveraged by social network mining methods as the ones described in D6.1. This module can thus be exploited for both use cases of the project.

### 6.1. Related Work

Early copy detection systems (Edward et al., 1998) mainly exploited global descriptors (Law-To et al., 2007). They were often used to detect repeated “clips” into a video stream (Naturel & Gros, 2005; Döhring & Lienhart, 2009). Although fast to compute, these methods usually suffered from poor robustness to severe image transforms. They were thus progressively replaced by local feature based approaches (Berrani et al., 2003; Joly et al., 2007; Douze et al. 2010). Such approaches rely on detecting regions around points of interest, and then describing them with robust descriptors such as SIFT (Lowe, 2004), SURF (Bayet et al., 2006) or robust local binary patterns (Heikkilä et al., 2009). Local features are quantized according to a predefined codebook and then indexed into an inverted file, which corresponds to a forward index described by bag-of-features (BoF) with hard coding (Sivic & Zisserman, 2003), and can thus be used for efficient retrieval. Further improvements have been proposed to compress or enhance this representation, making the retrieval process faster and more precise. In image classification, it has been shown that finer coding strategies can lead to better BoF representation (Huang et al., 2014). However, for retrieval, few of these strategies have been tested. The improvement proposed by (Jégou et al., 2008) better takes into account the local features quantization and add an efficient way to consider the spatial arrangement of interest points in each image. For very large scale databases, local descriptors are usually aggregated in a unique vector that describes the deviation of a given image with respect to an average representation of the visual world. This (VLAD or Fisher Kernel) vector is then compressed in order to be able to efficiently search a large database while maintaining good precision (Jégou et al. 2012).

### 6.2. Method description

The method proposed by CEA cannot yet be described since a patent deposit is currently pending. It will be however fully described in the updated version of this deliverable which is due at the end of the second reporting period.

### 6.3. Evaluation and testing

Experiments are carried out with two different benchmarks. The *Image Copy Detection dataset* is a benchmark that has been released recently (Thomee et al., 2013) and enables comparison to state-of-the-art methods. We also propose our own benchmark (*WebTransforms*) that includes other transformations that we consider important such as small rotation, flip and partial blur (e.g. on a human face) and that has been extended to a larger scale using 14 million of distractor images. All experiments were conducted on an Intel Xeon processor @ 2.10 GHz, 32 GB of RAM under Linux.



Figure 7. Examples of image transformations for *WebTransforms*, including rotation blur, crop, flip and image incrustation.

*WebTransforms* dataset and evaluation protocol. We propose a dataset in which 13 transforms and the identity operation (see Table 11) have been separately applied to the 5,011 training image of the PascalVOC'07 test dataset (Everingham et al., 2007), resulting into 70,154 images (i.e. each image resulted into 14 derivative images). We also merged our dataset into 2M and 14.2M distractor images extracted from Flickr to see how it behaves with relatively large-scale databases. A visual representation of the transformations is available in Figure 7. We evaluate the search performance on this dataset with the recall@14, i.e. the fraction of relevant images retrieved at the top 14 positions, averaged for all the queries.

| #  | Name               | Short description   |
|----|--------------------|---|
| 1  | identity           | original image  |
| 2  | blur               | moderate image blurring                                       |
| 3  | partial blur       | severe image blurring on a small portion of the image         |
| 4  | rotation           | centered, angle -10   |
| 5  | flip               | left-right  |
| 6  | rcrop              | 80% area, random center                                       |
| 7  | crop1              | 44% area, centered  |
| 8  | crop2              | 25% area, centered  |
| 9  | image incrustation | Lena image positioned in the center and covering 25% of image |
| 10 | text incrustation  | big colored text superposed                                   |
| 11 | sepia              | sepia filtering   |
| 12 | compress           | 10% JPEG compression ratio                                    |
| 13 | resize1            | scale factors: x = 0:6; y = 1                                 |
| 14 | resize2            | scale factors: x = 1:2; y = 0:8                               |

Table 11. Transforms used in the *WebTransforms* dataset.

*Image Copy Detection* dataset and protocol. The authors of (Thomee et al., 2013) introduced a dataset to evaluate image detection methods for searching duplicate images on the web. This dataset comprises 6K query pictures taken at various locations in the world. A set of 60 transformations have been applied to these images leading to a total of 360,000 images. Transformations were chosen after a survey which involved 45 persons familiar with image processing who were asked to report the most common transformations they encountered when looking at images with their favorite search engine. Following the evaluation protocol from (Thomee et al., 2013), we merged near-duplicate images with a 2M Flickr images collection that was collected specifically for this experiment. We also compared our method with the GIST descriptor (Oliva & Torralba, 2001) and TOP-SURF (Thomee et al., 2010), used with default parameters and a 1M words dictionary. The search quality is measured by the mean average precision (MAP) computed over the 6,000 queries for each transform.



### 6.3.1. Results on WebTransforms

The evaluation is conducted with two settings of our descriptor: (I) 20 bytes signature (USEMP-20), (II) 68 bytes signature (USEMP-68). We compare our method with GIST descriptor, a global descriptor popular for web-scale image indexing that is one of the best methods for content-based duplicate image detection (Thomee et al., 2013), as well as to a recently developed and publicly available high-quality implementation of SURF+VLAD and Product Quantization (Spyromitros-Xioufis et al., 2014)<sup>20</sup>. Results are reported in Table 12.

| Settings     | 1   | 2     | 3     | 4     | 5     | 6     | 7     |
|--------------|-----|-------|-------|-------|-------|-------|-------|
| USEMP-20     | 100 | 100   | 100   | 64.28 | 99.98 | 78.31 | 6.35  |
| USEMP-68     | 100 | 100   | 100   | 79.33 | 100   | 89.66 | 9.10  |
| GIST         | 100 | 100   | 100   | 96.95 | 25.66 | 99.54 | 59.37 |
| SURF+VLAD    | 100 | 99.36 | 99.92 | 99.52 | 98.12 | 100   | 98.44 |
| USEMP-68+2M  | 100 | 99.96 | 99.96 | 51.65 | 100   | 73.32 | 1.84  |
| GIST+2M      | 100 | 100   | 100   | 94.21 | 15.15 | 99.26 | 36.4  |
| USEMP-68+14M | 100 | 99.92 | 99.86 | 39.47 | 100   | 65.95 | 1.18  |

(a)

| Settings     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | Avg. |
|--------------|-------|-------|-------|-------|-------|-------|-------|------|
| USEMP-20     | 0.44  | 53.02 | 99.16 | 92.64 | 99.9  | 100   | 100   | 78.1 |
| USEMP-68     | 0.92  | 90.16 | 99.82 | 99.26 | 99.74 | 100   | 100   | 83.4 |
| SURF+VLAD    | 68.37 | 97.45 | 99.18 | 99.98 | 98.4  | 96.85 | 99.44 | 96.8 |
| GIST         | 4.33  | 81.38 | 91.3  | 98.82 | 99.5  | 100   | 100   | 82.6 |
| USEMP-68+2M  | 0.11  | 80.1  | 99.58 | 97.78 | 99.56 | 100   | 100   | 78.9 |
| GIST+2M      | 0.94  | 74.34 | 89.22 | 98.12 | 99.28 | 100   | 100   | 79.1 |
| USEMP-68+14M | 0.1   | 75.53 | 99.48 | 96.95 | 99.48 | 100   | 99.96 | 77   |

(b)

Table 12. Results of copy detection on the WebTransform dataset, including: (a) results for transforms 1 to 7 and (b) results from transforms 8 to 10 and averages for each method. Performance is measured using  $R@14$ , which is the number of correctly retrieved transforms in the first 14 results.

The 68 bytes descriptor outperforms the 20 bytes one by 5.3% and is slightly better than GIST by 0.8%. Most of the performance loss is due to severe crops (transforms 7 and 8). The VLAD+SURF and Product Quantization implementation seems to perform all competing approaches. However, the USEMP-20 and 68 descriptors behave gracefully when merged with a large scale image collection (2M and 14M distractors) since the performance drops only by 4 points with 2M distractors, giving a similar performance to GIST (GIST+2M). Once again, this is mainly due to severe crops, although one can observe a significant drop for rotation as well. However according to (Thomee et al., 2013) rotations are not often encountered on the Web. With 14 million distractors, the performance of our descriptor drops by further 2 points only. Preliminary experiments with the SURF+VLAD and Product Quantization method indicate that it is affected to a much higher extent by an increasing number of distractor images.

A further advantage of the USEMP approach is also that it is much faster. For instance, GIST requires 316 msec per query (average on 5011 queries) to search into the 70K images database and SURF+VLAD with product quantization needs 31 msec, while the USEMP

<sup>20</sup> Method implementation is available on: <https://github.com/socialsensor/multimedia-indexing>

approach needs only 4 msec. On a 200 times larger database (14M images) the USEMP method requires 845 msec with a single core and scales almost linearly to the number of used cores, requiring 53 msec with 16 cores.

### 6.3.2. Results on Image Copy Detection dataset

For this experiment we used setting the 68 bytes signature. We propose an evaluation which includes both speed and accuracy and thus measure the matching time, the mean average precision (MAP) and the average rank per duplicate for each transformation. Description and matching time for each method are reported in Table 13.

| Method   | Feature extraction (sec) | Matching time (sec) | MAP (%) |
|----------|--------------------------|---------------------|---------|
| TOP SURF | 0.340                    | 2.2                 | 93.7    |
| GIST     | 0.05                     | 9.0                 | 93.2    |
| USEMP    | 0.005                    | 0.120               | 99.1    |

Table 13. Results of copy detection on the Image Copy Detection dataset.

The USEMP method is at least one order of magnitude faster than other methods. A MAP of 99.1% is reported over all 60 transformations, while GIST and TOP-SURF have MAP scores of 93.2% and 93.7% respectively.

## 6.4. Implementation and usage

The module has been implemented in C++ and is ready for integration. To this purpose, we provide a library compiled with GCC under an x86\_64 linux system. The only dependence to an external library is to OpenCV 2.4.9 to be able to read an image (the required library is provided as well).

To create an index, use the following command:

```
Compute_dlphash images.lst index.dlph
```

where images.lst is an ASCII file containing the paths of all the images and index.dlph is the resulting index. To search into this index and get the 5 nearest neighbors, use:

```
Compute_dlphash queries.lst queries.dlph
```

```
Search_direct index.dlph queries.dlph 34 5
```

where queries.lst is an ASCII file containing the paths of all the images and index.dlph is the resulting set of signatures. To get a different number of neighbors change the number 5 into the wanted one.

## 6.5. Next steps

The copy detection module is now stable and minor effort is required for its integration in the USEMP system. Except for minor refinements and bug fixes, no additional effort will be allocated to this task in the remainder of the project.

## 7. Conclusions and future work

---

During the first iteration of the project, work on developing visual mining and linking modules was conducted in all directions described in the USEMP DoW. Recent advances in computer vision related to deep learning, coupled with our own contributions, allowed us to go well beyond the initial project objectives concerning concept detection both in terms of accuracy and of detected concepts. We have focused on scaling-up the learning process by learning concepts with little or no manual verification of images. A challenge we identified is to replace the currently used one-fits-all CNN model with domain adapted models in order to improve the detection accuracy. Given the central role of the concept detection module, we will pursue work notably concerning privacy-oriented multimedia concepts and the link between these concepts and the privacy scoring framework developed as part of WP6. A challenge here is to align visual concepts with privacy dimensions, especially concerning the personal data value part of the framework. Very interesting results were also obtained with a domain adaptation of CNN models for POI detection and here work will be pursued notably in D5.3 in order to integrate this module with text-based location recognition. Face detection was studied with standard models and we would like to update these models following recent progress in the field described in (Taigman et al., 2014). Logo and product recognition were also explored using standard techniques, which give interesting results but are probably close to the performance limits of these techniques. Here too, we will explore the use of CNNs to learn logo and product models. An interesting problem to tackle is the fact that logos/products often occupy a small area of the image and object localization methods will be tested to cope with this problem.

In parallel to visual mining modules improvement, we will focus on their integration in the USEMP system. As we mentioned, we will first integrate the concept detection module for pre-pilot studies to be carried out in February 2015. Then, we will progressively integrate the other modules with the overall objective of reaching full integration by the end of the second reporting period (September 2015).



## 8. References

---

- A. Acquisti, C. M. Fong. (2012). An Experiment in Hiring Discrimination Via Online Social Networks. Social Science Research Network Working Paper Series, Apr. 2012
- R. Arandjelovic, A. Zisserman. (2012). Three things everyone should know to improve object retrieval. Proceedings of International Conference on Computer Vision 2012
- A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky. (2014). Neural Codes for Image Retrieval. Proceedings of ECCV 2014
- H. Bay, A. Ess, T. Tuytelaars, L Van Gool. (2008). Speeded-up robust features (surf). Computer.Vision.and Image Understanding 110(3), pp. 346–359, june 2008
- R. Behmo, P. Marcombes, A. Dalalyan, V. Prinet (2010). Towards Optimal Naive Bayes Nearest Neighbor. In European Conference on Computer Vision (ECCV), pages 171–184
- A. Bergamo, L. Torresani. (2012). Meta-class features for large-scale object categorization on a budget. Proceedings of IEEE CVPR 2012
- S.-A. Berrani, L. Amsaleg, P. Gros. (2003). Robust content-based image searches for copyright protection. Proceedings of the 1st ACM International Workshop on Multimedia Databases, New York, NY, USA, MMDB '03, pp. 70–77, ACM
- O. Boiman, E. Shechtman, M. Irani. (2008) In defense of nearest-neighbor based image classification. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8
- E. Y. Chang, J. Z. Wang, C. Li, G. Wiederhold. (1998). Rime: a replicated image detector for the world wide web. Photonics East (ISAM, VVDC, IEMB), pp. 58-67
- J. Choi, X. Li. (2014). The 2014 ICSI/TU Delft Location Estimation System. Working notes of MediaEval Placing Task 2014
- J. Choi, B. Thomee, G. Friedland, L. Cao, K. Ni, D. Borth, B. Elizalde, L. Gottlieb, C. Carrano, R. Pearce, D. Poland. (2014). The placing task: A large-scale geo-estimation challenge for social-media videos and images. Proceedings of the 3rd ACM International Workshop on Geotagging and Its Applications in ACM Multimedia, 2014
- D. Crandall, L. Backstrom, D. Huttenlocher, J. Kleinberg. (2009). Mapping the World's Photos. Proceedings of WWW 2009
- J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Proceedings of CVPR 2009.
- I. Döhring, R. Lienhart. (2009). Mining TV broadcasts for recurring video sequences. Proceedings of CIVR, New York, NY, USA, pp. 28:1–28:8, ACM
- M. Douze, H. Jegou, C. Schmid. (2010). An image-based approach to video copy detection with spatio-temporal post-filtering. IEEE Transactions on Multimedia, pp. 257–266
- M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results. Online: <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>
- M. Heikkila, M. Pietikainen, C. Schmid (2009). Description of interest regions with local binary patterns. Pattern Recognition 42(3), pp.425–436

- Y. Huang, Z. Wu, L. Wang, T. Tan. (2014). Feature coding in image classification: A comprehensive study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3), pp. 493–506, March 2014
- B. Ionescu, A. Popescu, M. Lupu, H. Muller (2014). Retrieving Diverse Social Images at MediaEval 2014: Challenge, Dataset and Evaluation. Working Notes of MediaEval 2014.
- H. Jégou, M. Douze, C. Schmid. (2008). Hamming embedding and weak geometric consistency for large scale image search. *Proceedings of ECCV*, 2008
- H. Jegou, M. Douze, C. Schmid. (2011). Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(1), pp. 117–128, 2011
- H. Jegou, O. Chum. (2012). Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening. *Proceedings of ECCV 2012*, pp. 774–787, Springer, 2012
- H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, C. Schmid. (2012). Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(9), pp. 1704–1716
- Y. Jia. (2013). Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding. <http://caffe.berkeleyvision.org>
- A. Joly, O. Buisson, and C. Frelicot. (2007). Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia* 9(2), pp.293–306
- G. Kordopatis-Zilos, G. Orfanidis, S. Papadopoulos, Y. Kompatsiaris. (2014). Socialsensor at MediaEval Placing Task 2014. *Proceedings of MediaEval 2014 Working Notes*
- A. Krizhevsky, I. Sutskever, G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. *Proceedings of NIPS 2012*
- J. Law-To, O. Buisson, L. Chen, V. Gouet-brunet, A. Joly, N. Boujemaa, I. Laptev, F. Stentiford. (2007). Video copy detection: a comparative study. *Proceedings of CIVR*, 2007, pp. 371–378.
- X. Li, G.M. Snoek, M. Worring, A. Smeulders. (2012). Harvesting Social Images for Bi-Concept Search. *IEEE Transactions on Multimedia*, 14 (4)
- L. Liu, L. Wang, X. Liu. (2011). In Defense of Soft-assignment Coding. *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2486–2493
- D. G. Lowe (2004). Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision* 60(2), pp. 91–110, November 2004
- E. Mantziou, S. Papadopoulos, Y. Kompatsiaris (2013). Large-scale semi-supervised learning by approximate laplacian eigenmaps, VLAD and pyramids. *Proceedings of 14th IEEE Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pp. 1-4
- S. McCann, D. Lowe (2012). Local Naive Bayes Nearest Neighbor for Image Classification. *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR) 2012*
- X. Naturel, P. Gros (2005). A fast shot matching strategy for detecting duplicate sequences in a television stream. *Proceedings of the 2nd International Workshop on Computer Vision Meets Databases*, New York, NY, USA, CVDB '05, pp. 21–27, ACM
- A. Oliva, A. Torralba (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision* 42(3), pp. 145–175
- F. Perronin, Y. Liu, J. Sanchez, H. Poirier. Large-scale image retrieval with compressed Fisher vectors. *Proceedings of CVPR 2010*

- J. Philbin, O. Chum, O. M. Isard, J. Sivic, A. Zisserman (2007). Object Retrieval with Large Vocabularies and Fast Spatial Matching. Proceedings of CVPR 2007
- A. Popescu, N. Ballas. (2014). CEA LIST's participation at MediaEval 2012 Placing Task. Proceedings of MediaEval 2012 Working notes
- A. Popescu, G. Grefenstette. (2011). Social media driven image retrieval. Proceedings of ACM ICMR 2011
- A. S. Raman, J. L. Barloon, D. M. Welch. (2012). Social media: Emerging fair lending issues. The Review of Banking and Financial Services, 28(7), July 2012
- A. S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson. (2014). CNN Features off-the-shelf: an Astounding Baseline for Recognition. Proceedings of CVPR 2014 workshops
- S. Romberg, L. Garcia Pueyo, R. Lienhart, R. van Zwol. (2011). Scalable Logo Recognition in Real-World Images. Proceedings of ACM International Conference on Multimedia Retrieval 2011 (ICMR11), Trento
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei. (2014). ImageNet Large Scale Visual Recognition Challenge. arXiv technical report: <http://arxiv.org/abs/1409.0575>
- F. Schroff, A. Criminisi, A. Zisserman (2011). Harvesting image databases from the web. IEEE PAMI. 33(4), pp. 754--766
- P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun. (2013). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. arXiv technical report: <http://arxiv.org/abs/1312.6229>
- J. Sivic, A. Zisserman. (2003). Video google: A text retrieval approach to object matching in videos. Proceedings of ICCV 2003 vol.2, pp. 1470–1477
- E. Spyromitros-Xioufis, S. Papadopoulos, I. Kompatsiaris, G. Tsoumakas, I. Vlahavas (2014). A Comprehensive Study Over VLAD and Product Quantization in Large-Scale Image Retrieval. IEEE Transactions on Multimedia 16(6), pp. 1713-1728
- Y. Taigman, M. Yang, M. Ranzato, L. Wolf. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. Proceedings of IEEE CVPR 2014
- B. Thomee, M.J. Huiskes, E.M. Bakker, M.S. Lew. (2013). An evaluation of content-based duplicate image detection methods for web search. Proceedings of ICME 2013, pp. 1–6
- B. Thomee, E. M. Bakker, M. S. Lew. (2010). Top-surf: a visual words toolkit. Proceedings of ACM Multimedia 2010, pp. 1473–1476, ACM
- G. Tolias, Y. Avrithis, H. Jegou. (2013). To aggregate or not to aggregate: selective match kernels for image search. Proc. of International Conference on Computer Vision 2013.
- T. Tsirikas, J. Kludas, A. Popescu. (2012). Building Reliable and Reusable Test Collections for Image Retrieval: The Wikipedia Task at ImageCLEF. IEEE MultiMedia, 2012.
- J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong. (2010). Locality-constrained linear coding for image classification. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3360–3367, 2010.
- K. Yu, T. Zhang, Y. Gong. (2009). Nonlinear learning using local coordinate coding. Advances in Neural Information Processing Systems (NIPS), 22:2223–2231, 2009.

D. Zhang, M. Islam, G. Lu. (2012). A review on automatic image annotation techniques. *Pattern Recognition*, 45 (1), 2012

S. Zerr, S. Siersdorfer, J. Hare, E. Demidova. (2012). Privacy-aware image classification and search. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '12)*. ACM, New York, NY, USA, 35-44