D5.6

Multimodal content mining and linking framework – v2

v 1.1 / 2017-02-10

Hervé Le Borgne (CEA), Eleftherios Spyromitros-Xioufis (CERTH), Symeon Papadopoulos (CERTH), Yiannis Kompatsiaris (CERTH), Alexandru Ginsca (CEA), Adrian Popescu (CEA)

The current deliverable is a technical report accompanying the second version of the USEMP multimedia mining and linking modules. This deliverable is an update of D5.3 “Multimodal content mining and linking module – v1”. Similar to D5.3, it documents the underlying principles and methodologies, the exposed functionality, the respective implementation details, and the conducted evaluation experiments.
In particular, the following modules are discussed: a) multi-concept detection, b) visual-textual joint representation and c) private/non-private image classification. Visual-textual joint representation classification is a new module, while the others are updates of modules introduced in D5.3 and D5.5.

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	5
Deliverable lead org.	CEA
Deliverable type	Prototype
Authors	Hervé Le Borgne (CEA), Eleftherios Spyromitros-Xioufis (CERTH), Symeon Papadopoulos (CERTH), Yiannis Kompatsiaris (CERTH), Alexandru Ginsca (CEA), Adrian Popescu (CEA)
Reviewers	Georgios Petkos (CERTH) Rob Heyman (iMinds)
Version	1.1
Status	Final
Dissemination level	RE: Restricted Group
Due date	2016-06-30
Delivery date	2016-10-18 (revised 2017-02-10)

Version	Changes
0.1	ToC from CEA
0.2	First round of contributions from CERTH about private/non-private image classification
0.3	Contributions from CEA on concept detection
0.4	Second round of contributions from CERTH
0.6	Final version with refinements after review
1.1	Updated version addressing comments received from the third annual review

Table of Contents

1.Introduction

This deliverable provides a description of the USEMP multimodal[1] annotation, retrieval and location detection modules implemented during the second iteration of the project. The introduction first gives an overview of the role of multimodal mining in USEMP, of the research methodology and of multidisciplinary interactions within the project.

The main objectives of the deliverable are:

a) to clarify the usage of multimodal mining modules in the USEMP framework;
b) to show how textual and visual modalities can be effectively combined in order to improve the overall quality of multimedia mining results;
c) to detail the research approaches adopted, including implementation details;
d) to present an evaluation of multimodal mining modules on relevant datasets;

This deliverable provides documentation on the second version of the prototype implementations of the USEMP multimodal mining and linking modules. It is an update of D5.3 “Multimodal mining and linking module – v1” and, given that the overall objectives of the project did not change, the introductory section is largely similar to that of D5.3.

The main objective of multimodal mining and linking is to combine text and visual content mining in order to endow the USEMP framework with the capability to **conduct inferences about OSN users’ interests and traits based on the multimodal content** they share and interact with. Naturally, multimedia fusion is only doable if a document contains text and image components and, whenever this condition is not met, text mining (D5.1 and D5.4) or visual content mining (D5.2 and D5.5) should be used instead. Inferences over multimodal documents are most often extracted for individual documents, but are subsequently used in other parts of the project, as follows:

- Direct exploitation of multimodal inferences in the platform implemented in WP7;
- Combination with behavioral cues processed as part of the privacy scoring framework (T6.1) and integration in the USEMP platform.

Most of the multimedia efforts carried out in USEMP follow a late fusion scheme, in which textual and visual modalities are processed independently, followed by an integration of their outputs. During the second iteration of the project, we continue the work alongside this framework by building on previous efforts for **visual concept detection** (Section 2) and **private/non-private image classification** (Section 4) but also address the early fusion approach for multimedia mining by proposing a new **visual-textual joint representation** (Section 3).

1.1. Research methodology and contributions

Research on multimodal mining and linking is successively shaped by the conclusions of upstream research from other disciplines: legal studies (WP3), social science (WP4), user studies and system design (WP4, WP2). The links with these research streams are discussed in more detail in Subsection 1.2. Naturally, multimodal mining relies on the

[1] Here, multimodal refers to content processing techniques that make use of both textual and visual content at the same time.

modules available for text mining (D5.1 and D5.4) and visual content mining (D5.2 and D5.5). The overall objective is to leverage complementary contributions from individual textual and visual modalities in order to improve the obtained inferences. Assuming that the features for the involved modalities are already available, there are two main types of multimodal fusion: (1) early fusion – the features are combined in a common space before performing any further processing (i.e. machine learning for classification or similarity computation for retrieval) and (2) late fusion – a complete processing is performed for each modality and results are combined only at the end. In each case, the most effective methods stemming from the previous WP5 deliverables were selected as the basis for the modules, with preference given to reusing modules wherever possible. To assess the usefulness of the proposed prototypes, evaluation was carried out with suitable publicly available datasets or approaches.

Multimedia fusion work done during the second iteration of USEMP development cycles relied both on existing NLP and computer vision approaches and novel contributions directly targeting multimedia representations. We consider that it results in a number of interesting research contributions, including:

- In D5.2 we introduced a 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 - CNN, which fall under the family of Deep Learning methods) and is tested on large-scale datasets. In this second iteration, we introduce *D-CL,* an improved version of *Semfeat*, with the purpose of addressing some of its drawbacks by: (a) increasing the interpretability of detected concepts, (b) pushing forwards high level concepts and (c) improving the descriptors performances in classification tasks.
- We perform an exploratory study with the scope of investigating the performance of bi-modal representations. We introduce a new representation method, called Multimedia Aggregated Correlated Components (MACC), which aims at reducing the gap between the projections of visual and textual features by embedding them in a local context reflecting the data distribution in the common space.
- A new method for private/non-private image classification was introduced in D5.5. There, we performed an assessment of CNN and *Semfeat* features on this novel task, investigated the importance of user-centred feedback for the improvement of performance and presented the first version of a dedicated dataset was created as part of this task. Here, we continue this line of work by: a) enriching the dataset and re-evaluating generic and personalized privacy classification models on the expanded version, b) providing easily comprehensible explanations of the classification outputs using a new semantic feature representation (*semfeat-lda*) that is based on a new privacy aspect modeling approach, c) exploring the potential of discovering groups of users with similar privacy concerns and providing meaningful visualizations of the different groups and d) integrating a pilot version of the privacy-aware image classification module into DataBait

1.2. Multidisciplinary issues[2]

Multimodal mining operates on a combination of text and visual mining results and is thus mainly dealing with approaches from natural language processing, computer vision and machine learning. However, the presented research was considerably shaped by the rest of the USEMP disciplines, and at the same time provides actionable feedback to them. In the following, we provide a concise account of the inter-play between multimodal mining research and the different disciplines of the project.

D5.6 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 the coordination of legal aspects, clarified practical implications of multimodal mining related and were turned into specific requirements that were implemented:

- The USEMP end-users should be clearly informed about their rights and obligations when engaging with the platform.
- Processing of personal data should be subjected to a declaration of USEMP work to national Data Protection Agencies.
- Processing of sensitive information, such as user personally identifiable information, should be considered separately and be subjected to a specific declaration.
- Copyright issues should be carefully considered for training data used during USEMP and, more importantly, for any commercial implementation of its results after the end of the project
- Ensuring that all USEMP components have clear IP rights (in case of reusing existing components).

Work on trade secrets and intellectual property done 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 system requirements are central here:

- [SR02] [3]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

[2] Multidisciplinary issues are, to a large extent, common to all WP5 deliverables and this section has thus similar content in D5.1, D5.2 and D5.3.
[3] The requirement notation is the one used in the deliverables that extracted them.

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 the expected behavior of the developed USEMP tools. While all requirements are important, the following ones have particular impact on multimedia mining modules:

- 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 explainability of the automatic decisions made by the system.
- Req. 10 asking for 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 text and image 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 D5.6 tools can be used both directly in the USEMP interface (D7.2), and as part of the privacy disclosure framework created in D6.1, to complement social network mining inferences. For instance, user locations can be extracted from texts and images and can then be displayed directly by the USEMP interface to inform the user about her degree of exposure on a certain privacy dimension (e.g. location). In a more complex functioning mode, multimodal data representations can be combined with social interaction data (such as likes, comments) to improve the quality of predictions.

2. Multi-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, are illustrated in user content shared on OSNs and scalability in terms of recognizable concepts should be a core requirement, along with detection accuracy.

The remarks drawn from the pilot feedback detailed in D8.5 support the user interest for visual concept detection. However, one of the comments that stood out was the high specificity of the concepts that are shown to users, which may lead to some of them being deemed irrelevant. Through the second iteration of the concept detection module, besides improving our semantic descriptor by highlighting more relevant concepts, we also focus on pushing forward higher level concepts that are automatically inferred from the large set of individual concept detectors coupled with external knowledge sources.

In D5.1 and D5.3, we proposed a concept detection approach that is applicable to the very large number of concepts which can appear in OSN users' image streams and a semantic image descriptor, named *Semfeat*. Most semantic features in the literature (Jain et al., 2015), (Ginsca et al., 2015) consider visual concepts independently from each other whereas they are often linked together by some semantic relationships (i.e. hyponymy, hypernymy, exclusion, etc.).

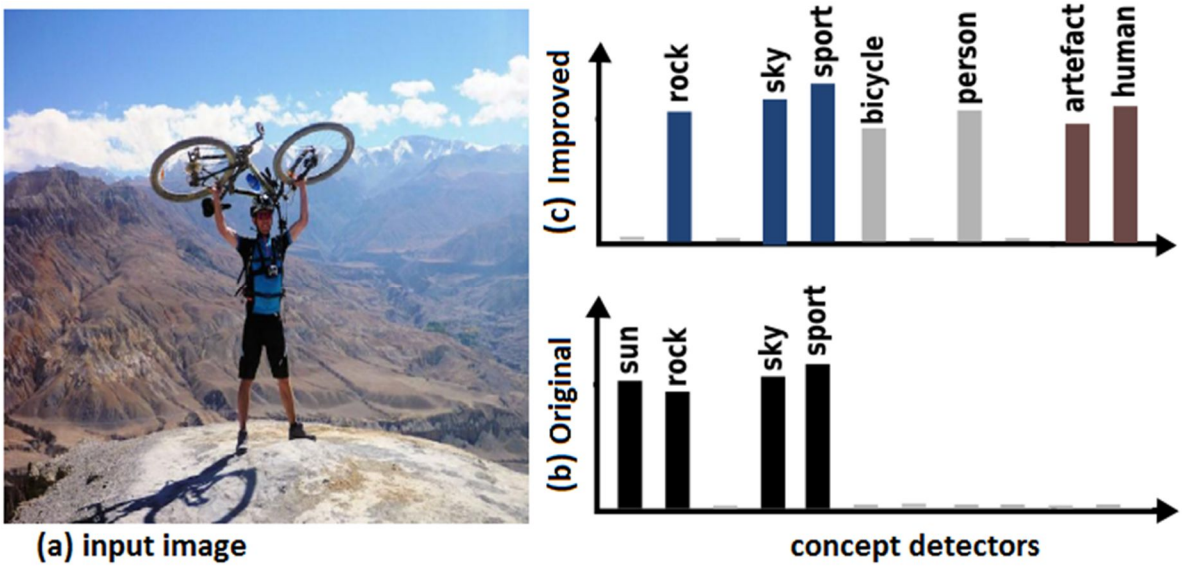

Figure 1: We propose a semantic representation that computes the concepts' presence differently according to their categorical level. For an input image (a) with multiple objects our previous approach would output the concepts illustrated in black (b) and miss useful concepts, such as person and bicycle. In contrast, the proposed scheme captures properties of the image that are useful for categorization (c), e.g. superordinate concepts (brown), basic-level (gray) and subordinate (blue) concepts, making the representation more relevant.

Here, we take into account the relations between concepts using existing human knowledge, such as semantic hierarchies (e.g. WordNet), which makes our approach a "top-down" scheme. More precisely, our main contribution consists of identifying three types of concepts

into an existing hierarchy, according to their categorical level, and processing them differently to design the semantic feature. It is nevertheless not easy to determine the categorical levels to which a concept belongs. Hence, we propose a method to identify the three groups in practice, for a given supervised classification problem. The novel semantic representation is named Diverse Concept-Level feature (D-CL). Compared to bottom-up approaches, an advantage of the proposed top-down scheme appears when the concept detectors fail at the subordinate level because category is finer thus harder to identify.

In Figure 1, we present an example of the improvement brought by the novel semantic descriptor introduced in the second iteration of the visual concept detection module. Besides injecting higher order concepts (e.g. person, human), we are also able to propose additional details of the image's content (i.e. detecting bicycle, whereas this concept was not retained by our first version of the module). This improvement also benefits the privacy disclosure framework developed within WP6. For instance, the privacy dimension "hobbies" can be better estimated by capturing the "bicycle" concept.

2.1. Related work

The problem of object class recognition in large scale image databases is a topic of high interest in the vision community. In parallel to the mainstream data-driven approach, based on convolutional neural networks, several works adopted a concept-driven scheme to design semantically grounded image features, that we name semantic features in the following.

Given the availability of large-scale image datasets, an image representation based on a bench of object detectors is a promising way to handle natural images according to their category. These object detectors are more generally considered as the outputs of base classifiers. Such approaches offer a rich high-level description of images that is close to the human understanding. Semantic features are also scalable in terms of number of concepts thus being able to cope with a wide variety of content. An exception is the work of (Bergamo & Torresani, 2012) that introduces "meta-classes" to address this aspect. Those meta-classes are "abstract" categories (do not really exist in the real-world) that capture common properties among many object classes. They are built using spectral clustering on low-level features of images among a set of categories. The restrictive assumption of this method is the dependence of the meta-class learning on the visual low-level features.

The classical formulation of semantic features exploits all classifier outputs but it was recently shown that forcing the semantic representation to be sparse (by setting the lowest values to zero) can be beneficial both in terms of scalability and performance (Ginsca et al., 2015). Nevertheless, semantic features with a large set of concept detectors often contain a high number of visually similar concepts to describe the same object.

The current trend in object classification is to exploit mid-level features obtained with deep convolutional neural networks, such as VGG-Net (Simonyan & Zisserman, 2015). Built on top of such mid-level representations, we focus on semantic features that: (i) include a rich representation of images, (ii) provide a humanly understandable description of content, and (iii) are more flexible since concepts are learned independently from each other. This semantic-based approach has been introduced by (Torresani et al., 2010) with a limited number of concepts. They used nonlinear LP-beta classifiers to learn each concept detector.

We previously explored in D5.1 linear SVMs for building semantic features and showed their effectiveness when the features are constrained to be sparse.

The feature is said sparse when, for a given image, only a limited number of dimensions are non-zero. Several methods have been proposed to determine the level of sparsity for semantic features. Sparsification is the process that sets to zero the lowest output values and keeps activated only the other dimensions of a feature. For instance, (Li et al., 2010) managed this sparsity aspect at learning time through the regularization of logistic regression. Recent works such as (Jain et al., 2015) exhibit very good performances in image retrieval, and action classification, by retaining in the final feature, a small (but fixed) number of the largest classifier outputs (less than 100). In our scheme, the sparsity is a consequence of the proposed concept groups identification. In this work, the selection of concepts is done in regards to their identified categorical-level, yielding to representations containing only useful concepts. Contrary to former works, our sparse representation is adapted to each image, according to its actual content, and relative to the problem of interest.

2.2. Method description

In this section, we detail our proposed approach, a new method for multi-concept detection in images and the resulting semantic representation that takes into account available human knowledge. We first identify three types of concepts into an existing hierarchy (according to their categorical level) and then, process their concepts differently. It is nevertheless, not straightforward to identify these three groups, in practice.

2.2.1. Diverse concept-level feature

A semantic feature is a *F*-dimensional vector extracted from an image *I*, itself described by a mid-level feature *x*. The feature *x* could be any image descriptor such as Bag-of-Word or Fisher Kernel features, but also mid-level features such as those obtained from a fully-connected layer of a convolutional neural network. Each dimension of the semantic feature is the output of a classifier for the concept c_i evaluated on *x*.

While the concepts c_i are potentially linked together by some semantic relationships, in our first iteration of *Semfeat* (see D5.1), we considered them independently. We now propose to rely on existing human knowledge regarding the relations between concepts. Such a knowledge is, for instance, reflected into existing hierarchies such as WordNet that organize a large set of concepts according to “is-a” relationships, that is to say by defining hyponyms and hypernyms. An advantage of our approach is to remove the dependence to the basic visual descriptor and to introduce human-based information within the process of image representation design.

All the concepts considered in semantic features, are named according to existing *categories*. The name of a category is given according to a human judgment. We adhere to a commonly used concept classification:

- **Basic-level concepts** are the terms at which most people tend naturally to categorize objects, usually neither the most specific nor the most general available category but the one with the most distinctive attributes of the concept.

- **Superordinate concepts** are categories placed at the top of a semantic hierarchy and thus display a high degree of class inclusion and a high degree of generality. They include basic-level and subordinate concepts.
- **Subordinate concepts** are found at the bottom of a semantic hierarchy and display a low degree of class inclusion and generality. As hyponyms of basic-level concepts, subordinate categories are highly specific.

At the core of our approach, concepts are processed differently according to their categorical level. This asymmetrical process is based on a cognitive study proposed by (Jolicoeur et al., 1984), where they conclude that, concepts are processed differently by humans, i.e., it is purely perceptual for the basic-level and subordinate concepts, while it is inferred using stored semantic information, for superordinate concepts. In our scheme, basic-level and subordinate concepts are computed through a visual process, while superordinate concepts are processed semantically using the hyponym relations between concepts into hierarchies.

Figure 2: Illustration of concepts that our D-CL feature would predict, for two different images. It selects concepts from different categorical levels of a semantic hierarchy, i.e., superordinate, basic-level and subordinate concepts.

Figure 2 illustrates for two input images, the three types of concepts that would be retained by our scheme. More precisely, for an input image, the probability of a basic-level or a subordinate concept is the output of a binary detector for the concept c_i evaluated on the mid-level feature x, further normalized by a sigmoid function such that the final predicted score falls in the $[0,1]$ interval. The visual classifiers have been learned using images of the concept c_i as positive samples and images of a diversified class as negative samples.

Each concept classification model is obtained with L2-regularized linear SVMs, but other linear models could be used. Regarding the process of basic-level and subordinate concepts, even if it is similar, a particular difference is that, all basic-level concepts are selected in the final representation, while for subordinate concepts, we select only the most salient. This particular process for subordinate concepts avoids redundancy of information.

Concepts at the highest categorical level (superordinate) are computed, for an input image, through a *semantic classifier*. It is an inference of concepts that have at least one hyponym relation with the superordinate concept c_i. We thus define the subsumption function that aims to output the set of concepts having hyponym relations with an input concept. We further, define the semantic classifiers that are used to compute superordinate concepts. A subsumption function $\varsigma(\cdot)$ takes as input a concept c_i and a semantic hierarchy H with hyponymy relations and outputs a set C_i of concepts that are subsumed by the concept c_i, i.e., the concepts that have a hyponymy relation with the concept c_i in a semantic hierarchy. A semantic classifier is an operator that predicts the probability of presence of a concept c_i in

the image through a semantic inference of purely visual output classifiers. The proposed *Diverse Concept-Level* (D-CL) feature computes superordinate concepts through a semantic classifier and all other concepts, i.e. basic-levels and subordinates, using a visual classifier. It also selects all basic-level and superordinate concepts and retains only the most salient subordinate concepts. An illustration of the asymmetric process according to the type of concepts is presented in Figure 3.

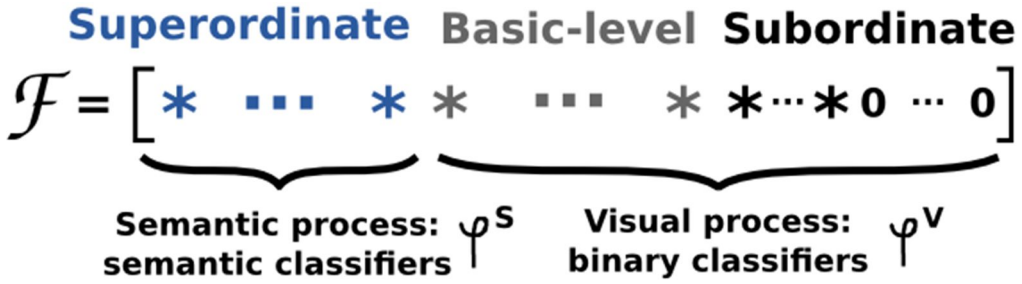

Figure 3: Illustration of the asymmetric process in our D-CL feature. Superordinate concepts are processed semantically through semantic classifiers, while basic-level and subordinate concepts are visually processed through binary classifiers. Stars and zeros represent output values $F_i(\cdot)$ of each concept of the D-CL feature $F(\cdot)$. Note that, concepts are grouped by categorical levels, but any order could be obtained in a real scheme.

2.2.2. Identifying concept groups

In this section, we detail how to identify the three groups of concepts (basic-level, superordinate and subordinate), in practice, for a given supervised classification problem. The D-CL feature is computed by activating all the basic-level concepts, all the superordinate concepts, the *K* most salient subordinate concepts and by deactivating all others.

Let *F(x)* be the D-CL feature of a mid-level feature *x* extracted for an image *I* contained in a targeted dataset. Let D^d be the set of *d* categories of the targeted dataset. While basic-level concepts are not available at a large scale, we propose to identify, in an offline phase, the set of basic-level concepts *BL* selected in our D-CL feature by matching it with the set of targeted dataset categories D^d. The latter, is based on the assumption that broader-datasets mostly contain categories at the basic-level. Specifically, all targeted dataset categories d_i are matched with concepts c_i to generate a set of basic-level concepts adapted to the dataset BL^d. In fact, this matching has the advantage to make our D-CL feature adaptable to the application context. Regarding the sets of superordinate *P* and most salient subordinate B^K concepts, they are therefore automatically selected through the subsumption function $\varsigma(\cdot)$ that takes as input concepts from BL^d and a semantic hierarchy *H* with "is-a" relations.

Selecting a portion of the whole concepts, and setting others to zero is closely related to the sparsification processes that sets to zero the lowest output values and keeps activated only the other concepts. Recent works underlined that such a property of sparsity has the advantage to be effective and computationally efficient. The added value of this work is the adaptability of the concept selection to the input images. Contrary to our previous effort, the sparsity is adapted to each image, according to its actual content, and relative to the problem of interest.

The D-CL feature is illustrated in Figure 4. It is able to capture from an image containing multiple objects, all the basic-level concepts (colored in dark green) adapted to the target dataset, all its superordinate concepts (colored in dark red) and the most salient subordinate ones (colored in dark blue). It results in a final representation capturing the most informative concepts for a target collection of images.

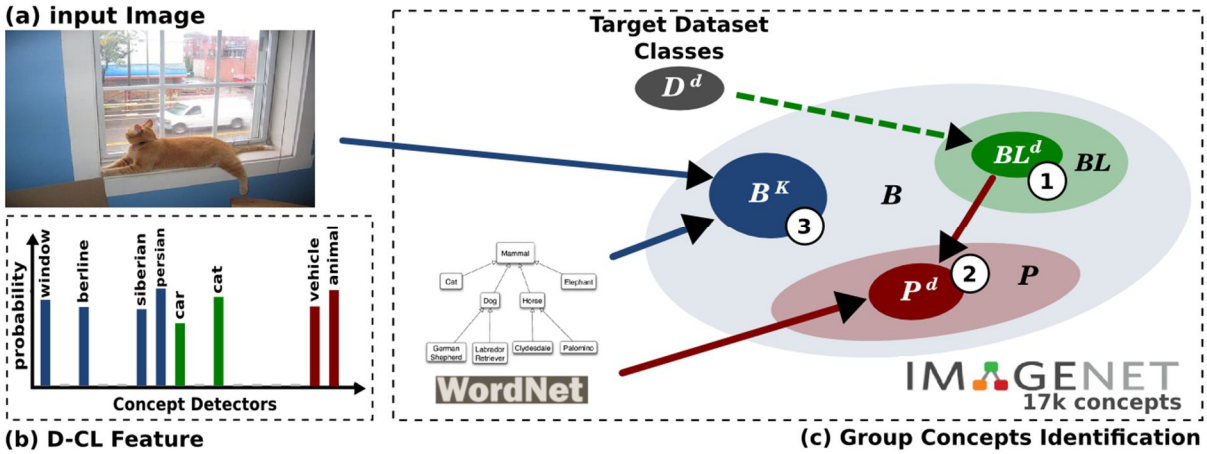

Figure 4: Illustration of the concept groups identification (c) in a practical case, for an input image (a) contained in a dataset collection. The proposed concept groups identification selects (1) in an offline phase (dashed arrow), the concepts of the target dataset categories as a portion (BL^d) of all basic-level concepts (BL), (2) the part (P^d) of its superordinate concepts (P) and in a final step (3) the most salient (B^K) subordinate concepts (B). For steps (2) and (3), a semantic hierarchy (WordNet) is used to compute the hyponymy relations. This latter, results in the final D-CL representation (b), to an activation of diverse concept levels (i.e., superordinate, basic-level and subordinate) and a deactivation of all other concepts.

2.3. Evaluation and testing

In order to evaluate the proposed "Diverse Concept-Level" feature we test its performance on three multi-object classification datasets and we compare it with the best semantic features in the literature. Finally, we evaluate the contribution of the asymmetrical process of concepts in the proposed D-CL descriptor by, first, evaluating the proposed semantic classifier and comparing it to traditional binary classifiers, and then, assessing the contribution of each concept group selection.

The effectiveness of the proposed diverse concept-level feature is tested in the context of multi-class object classification. It is evaluated according to a standard experimental protocol as reported in the recent literature on the following three datasets:

- Nus-Wide Object (Chua et al., 2009) is a multi-object classification dataset. As a subset of NUS-WIDE, it consists of 31 object categories and 36,255 images in total. It contains 21,709 images for training and 15,546 images for testing. Each image is labeled by one or several labels from the 31 categories.
- Pascal VOC 07 (Everingham et al., 2010) is a multi-object classification task. It is based on a dataset that contains 9,963 images, each image being labeled by one or several labels from 20 categories. We used the pre-defined split of 5,011 images for training and 4,952 for testing.

- Pascal VOC 12 (Everingham et al., 2012) is similar to VOC 2007 but its number of images is larger: 22,531 images are split into 11,540 images for training and 10,991 images for testing.

For all experiments, ImageNet (Deng et al., 2009) is used to learn our diverse concept-level representation. We especially use a subset of ImageNet with 17,462 concepts, containing more than 100 images each. Thus, we learn each individual concept detector using images representing the concept c_i as positive samples, and images of a diversified class as negative samples. Note that the concepts can be at any categorical-level of a semantic hierarchy, making our method applicable on top of any semantic feature. The set of basic-level concepts BL^d is matched with the set of targeted dataset categories, for each dataset.

Since all the concepts of ImageNet are organized in accordance to the WordNet hierarchy, we use it as input to the subsumption function $\varsigma(\cdot)$ to select the corresponding superordinate concepts P^d. Specifically, only the first and the fourth level of the WordNet hierarchy are used. This avoids redundancy of semantically close superordinate concepts. For the set of the K most salient subordinate concepts (B^K), the parameter K, is cross-validated on each training dataset using the usual training/validation split.

Semantic features (including the proposed D-CL), are built on top of any low-level or mid-level features (CNN). However, the quality of the D-CL feature will directly depend on the low/mid-level feature used. We thus, created semantic features on top of a competitive mid-level feature released in the literature, namely VGG-Net (Russakovsky et al., 2015). Note that, for a fair comparison, the same mid-level feature is used to build Classemes (Torresani et al., 2010) and Semfeat representations. For our study, fine tuning of the CNN may result into an improvement of the results at the cost of significant computational cost and the possible use of additive data. Such a specific optimization of the CNN has not been considered in our experiment, to insure their reproducibility with the available CNN models.

2.3.1. Multi-Object Classification Results

In Table 1, we test the D-CL feature for multi-object classification on the datasets presented above. The evaluation of our method lies in the context of semantic features. Thus we compare its performances to the following three baselines:

- Semfeat is the previous iteration of our semantic descriptor based on concept detectors. To fairly compare it to our novel method, we build it on top of the same middle level descriptor. This layer is used to learn the classifiers of the 17,462 concepts of ImageNet that contains more than 100 images. According to the original work, a sparsification over images is performed;
- Classemes+ is, for a fair comparison with other methods, our own implementation of Classemes (Torressani et al., 2010). We build it on top of a the 16th layer of VGG-16 with the same concepts as our method and Semfeat, that is to say 17,462 concepts of ImageNet containing more than 100 images. Like in the original work, no sparsification is considered;
- Meta-Class (Bergamo et al., 2012), is the output of 15,232 concept detectors. It is based on a concatenation of low-level features combined with a spatial pyramid histogram with 13 pyramid levels.

Method	Nus-Wide Object (20%)	Pascal VOC 2007 (45%)	Pascal VOC 2012 (30%)
Classemes+	70.3	82.4	81.7
Meta-Class	36.5.	48.4	49.3
VGG-16 (fc8)	67.3	77.4	77.2
Semfeat	74.7.	82.8	81.7
D-CL	**76.0**	**85.1**	**83.0**

Table 1. Overall performance (mean Average Precision in %) of the following methods, ObjectBank, Classemes, Classemes+, Picodes, Meta-Class, VGG-16 (fc8), Semfeat and our approach (D-CL) on Nus-Wide Object, Pascal VOC 2007 and Pascal VOC 2012. We mention, for each dataset (in parenthesis), the rate of images labelled with multiple labels.

Regarding the classification protocol, each class of the datasets is learned by a one-vs-all linear SVM classifier and we use mean Average Precision (mAP) to evaluate the performances. For each dataset, the cost parameter of the SVM classifier and the parameter K are optimized through cross-validation on the training images, using the usual train/validation split. Results are reported in Table 1. Our novel descriptor significantly outperforms all the other representations. On Pascal VOC 2007, D-CL has better performances than the four baselines Classemes+ (+2.7 points of mAP), Meta-Class (+35), VGG-16-fc8 (+7.7) and Semfeat (+2.3 points of mAP). Similar improvements are observed on Pascal VOC 2012 and Nus-Wide Object datasets. However, we note that, compared to all baselines, the improvements of the proposed D-CL feature, is much better on Pascal VOC 2007 than Pascal VOC 2012 and Nus-Wide Object. This result is aligned with the expectation since Pascal VOC 2007 contains a larger part (45%) of images labeled by multiple classes, compared to Pascal VOC 2012 and Nus-Wide Object, that contain only 30% and 20%, respectively.

2.3.2. Accuracy of Semantic Classifiers

In this section, we assess the effectiveness of the proposed semantic, and compare it with purely visual classifiers, i.e., binary classifiers on generic concepts (i.e. concepts that have at least one hypernym relation with another concept).

Through this analytic investigation, we look for evidence that superordinate concepts are semantically processed by humans, rather than by a visual perception processing. Thus, we evaluate the proposed semantic classifier and the visual classifiers on superordinate concepts only. Regarding our experiment, the selection of superordinate concepts imposes to set to zero all the basic-level and subordinate concepts. Thereby, the experiment has been conducted on the context of multi-class object classification through the Pascal VOC 07 dataset. All the images of the dataset have been re-labeled at superordinate level, e.g. all images labeled as bird, dog, cow, horse or sheep are now labeled as animal, all images labeled as chair, sofa or table are now labeled as furniture, etc. (see the second column of Table 2 for the re-labeling of other classes). Hence, we learn each superordinate class of the dataset by a one-vs-all SVM classifier. The cost parameter of the SVM classifier is optimized through cross-validation on the training dataset, using the usual train/validation split. Performance results of both classifiers are reported in Table 2 using average precision (AP in %) for each class and mean Average Precision (mAP in %) over all classes in the last row.

Superordinate	Basic-level	Visual	Semantic
Animal	bird - cow - dog - horse - sheep	92.9	97.7 (+4.8)
Electronic equipment	tv monitor	52.1	72.6 (+20.5)
Furniture	chair - sofa - table	70.0	74.9 (+4.9)
Person	person	77.2	85.7 (+8.5)
Plant	potted plant	26.5	40.5 (+14.0)
Vehicle	airplane - bike - boat - bus - car - mbike - train	93.4	96.9 (+3.5)
Vessel	bottle	18.7	31.4 (+12.7)
mAP		**61.5**	**71.4 (+9.9)**

Table 2. Evaluating purely visual binary classifiers (denoted as Visual) and our proposed semantic classifiers (denoted as Semantic) for superordinate concepts (first column) of Pascal VOC 07 dataset classes (second column). The improvements of semantic classifiers over visual classifiers are shown in parentheses. Note that, the class person of Pascal VOC 2007 is already at the highest level in the WordNet hierarchy.

The average precision of each superordinate concept computed through binary classifiers (denoted as Visual) and the proposed semantic classifier (denoted as Semantic), are presented in the last two columns, respectively. Remarkably, the proposed semantic classifier clearly outperforms binary classifiers (purely visual) for all the superordinate concepts. From this study, we conclude that the superordinate concepts are better recognized by D-CL, due to its ability to compensate low within-category resemblance of generic concepts.

2.3.3. Concept Groups Selection Sensitivity

We evaluate now the contribution of the concepts from different groups (i.e. categorical levels) on a multi-object classification task (Pascal VOC 2007). To this end, we need to isolate each group of concepts in the D-CL representation by selecting them individually and setting other groups to zero. It results in four special cases of the D-CL feature, (i) selecting only superordinate concepts denoted as *Superordinate*, (ii) selecting only basic-level concepts denoted as *Basic-level* (iii) selecting only subordinate concepts denoted as *Subordinate* and (iv) selecting only the K most salient subordinate concepts, denoted as *K-Subordinate*. We also evaluate the contribution when selecting all the concept groups in the representation, e.g., superordinate, basic-level and subordinate concepts, denoted as *Fusion 1*. Finally, we report the results obtained by the proposed D-CL concept groups selection, corresponding to the selection of, all the superordinate and basic-level concepts and the K most salient subordinate concepts. It is also a fusion of other groups of concepts that we denote as D-CL. Results are reported in Table 3. For each concept group selection, a check-mark represents the concept groups that had been selected in the final representation. The last column gives the mAP obtained for the different concept selections.

Selecting only superordinate concepts (P) leads to very bad results, compared to basic-level concepts only (BL), which are their-self lower than subordinate concepts only (B). Selecting only the K most salient subordinate concepts (BK) obtains lower performances than selecting them all. Surprisingly, for the fusion, it is better with the selection of the K most salient concepts (the proposed D-CL) than with the selection of all subordinate concepts (Fusion 2). This experiment shows that the proposed D-CL selection gives a most effective semantic representation.

Concept Groups Selection	P	BL	B	BK	mAP
Superordinate	✓				44.4%
Basic-level		✓			76.1%
Subordinate			✓		82.1%
K-Subordinate				✓	78.9%
Fusion 1	✓	✓	✓		82.7%
Fusion 2 (D-CL)	✓	✓		✓	85.1%

Table 3. Evaluation of the contribution of different concept groups selection (check-mark = selected group) in the proposed semantic feature on Pascal VOC 2007 dataset.

2.4. Implementation and usage

We rely on the same implementation used for our first iteration of the concept detection module that is described in D5.3. The difference lies in the use of a different pre-computed concepts model (denoted by *concept-models* in our implementation). In our extraction pipeline, we replace the previous *Semfeat* model with the newly introduced *D-CL*.

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)[4]. 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.

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 concept detection wrapper can be called with the following command:

[4] 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)

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

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

Table 1. Concept detection usage.

3. Visual-textual joint representation and mining

Upstream work done in WP2, WP4 and WP6 provided valuable insights about the usage and impact of both textual and visual cues. A diverse panoply of concepts, i.e. entities, objects and themes of interest depicted in multiple modalities (e.g. image, text, video) are present in user content shared on OSNs. Some of them have a clear visual representation (e.g. animals, drinks etc.) and can be inferred using purely computer vision techniques, as shown in D5.2 and D5.5, while for others textual cues might be required as well (e.g. sexual orientation, political views, opinions) and impose the use of text mining, as we illustrated in D5.1 and D5.4. However, items that share both modalities (e.g. images with associated textual description, title or tags) present a widespread use in most current OSNs. Relying on approaches in which we process each modality independently worked successfully by their integration through the privacy disclosure framework carried out in WP6. To fully take advantage of these types of data, it is important to propose approaches which leverage both the powerful visual cues and the textual annotations produced by users. We perform here an exploratory study aimed at investigating the performance of bi-modal representations. This gives us the opportunity to devise rich multimedia representations that support both multi-modal and cross-modal tasks.

To deal with this problem, we put forward a new representation method for the projections on a common space, called Multimedia Aggregated Correlated Components (MACC). It aims to reduce the gap between the projections of visual and textual features by embedding them in a local context reflecting the data distribution in the common space. Given a database of multimedia documents, we first perform Kernel Canonical Correlation Analysis (KCCA)s and build a codebook from all the projections of visual and textual features on the KCCA common space. Subsequently, for each multimedia document, visual and textual features are projected on this common space, then coded using the codebook and eventually aggregated into a single MACC vector that is the multimedia representation of the document (see Figure 5).

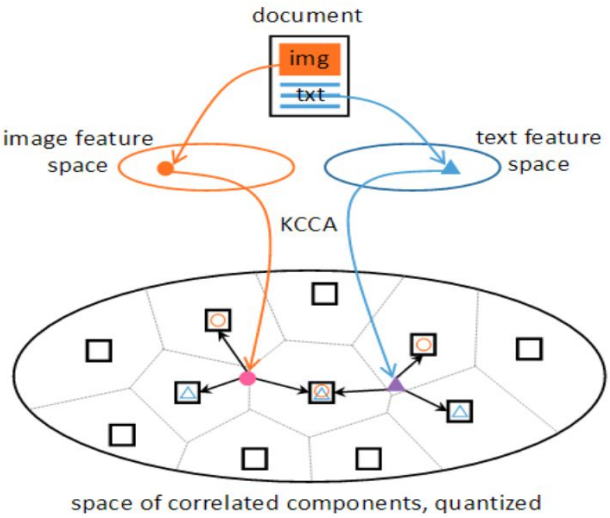

Figure 5: Visual and textual contents of a document are projected onto a common space that has been previously quantized. Both points, corresponding to the same document, are encoded according to a common vocabulary before their aggregation.

3.1. Related work

In bi-modal image classification, visual and textual content are employed together for solving the task. Cross-modal tasks like text illustration or image annotation require instead to "translate" information from one modality to another. But to address cross-modal tasks it is necessary to devise methods that are able to link the two modalities more closely. This is accomplished through the development of a common, latent representation space resulting from a maximization of the relatedness between the different modalities. The methods typically rely on Canonical Correlation Analysis or its kernel extension (Hardoon et al., 2014) (Hwang and Grauman, 2012) and on deep learning (Ngiam et al., 2011) (Srivastava and Salakhutdinov, 2012).

Given a set of documents described along two different modalities like image and text, Kernel Canonical Correlation Analysis (KCCA) aims to find maximally correlated manifolds in the feature spaces associated to the two modalities. While mainly considered for cross-modal tasks, a common representation space also has the potential to improve the results obtained in bi-modal tasks. For images described by both a visual and a textual content, bi-modal tasks typically focus on semantics. The common representation space favors inter-related information that usually highlights semantics and discounts modality-specific information. In the recent literature, various (K)CCA-based approaches have been proposed to deal with either cross-modal or bi-modal tasks. CCA was first applied to cross-modal retrieval in (Hardoon et al., 2014), where its kernel extension KCCA was also introduced in order to allow for more general, non-linear latent spaces. Since not all the words (or tags) annotating an image have equal importance, (Hwang and Grauman, 2012) proposed a method taking advantage of their importance when building the KCCA representation. The importance of a word for an image is obtained from the order of words in the annotations provided by users for that image. (Gong et al., 2012) put forward a multi-view (K)CCA method: a third view, explicitly representing image's high-level semantics, is taken into account when searching for the latent space. This "semantic" view corresponds to ground-truth labels, search key-words or semantic topics obtained by clustering tags. This first group of approaches focuses on investigating complete representations of data for building a robust common space. Nevertheless, they directly use the projections of the textual and visual descriptors the KCCA common space in order to perform cross-modal tasks.

Approaches in a second group aim to build upon these direct projections on the KCCA common space. Specifically, (Costa Pereira et al., 2014) proposed semantic correlation matching (SCM), where the projections of image and text features by (K)CCA are first transformed into semantic vectors produced by supervised classifiers with respect to pre-defined semantic classes. These vectors are then used for cross-modal retrieval. (Ahsan et al., 2014) employed the con-catenation of textual and visual KCCA-descriptors as inputs of a clustering algorithm to perform a bi-modal task, social event detection. Our proposal follows this second group of approaches. The novelty of our work compared to existing methods is to build a common vocabulary for image and text on the KCCA space and to represent multimedia documents by aggregating their visual and textual descriptors defined on this common vocabulary.

3.2. Method description

We describe a new representation of multimedia documents relying on an aggregation of the projections of visual and textual content defined on a common vocabulary. Since (K)CCA aims to find a projection space where the correlation between modalities is maximized, we named this new representation "Multimedia Aggregated Correlated Components" (MACC). We then present an extension for completing the MACC representations of documents for which only one modality is available. While MACC addresses problems with the representation of bi-modal documents, this extension focuses on actual cross-modal cases.

For data simultaneously represented in two different vector spaces, CCA finds maximally correlated linear subspaces of these spaces. Let X^T and X^I be two random variables, taking values in $R_d{}^T$ and respectively $R_d{}^I$. Consider N samples $\{(x^T{}_i , x^I{}_i)\}^N \subset R_d{}^T \times R_d{}^I$. CCA simultaneously seeks directions $w^T \in R_d{}^T$ that maximize the correlation between the projections of x^T onto w^T and of x^I onto w^I.

Kernel CCA (KCCA) aims to remove the linearity constraint by using the "kernel trick" to first map the data from each initial space to the reproducing kernel space (RKHS) associated to a selected kernel and then looking for correlated subspaces in these RKHS.

3.2.1. Aggregation of textual and visual information in the projection space

Let us consider a document with a textual and a visual (image) content. A feature vector x^T is extracted from its textual content and another feature vector x^I from the visual one. In what follows, we assimilate a document to a couple of feature vectors (x^T , x^I). A set of such data is a set of couples $X = \{(x^T{}_i , x^I{}_i), i = 1 . . . N \}$. By applying KCCA to this data, as explained, we obtain $2N$ points (vectors) belonging to a common vector space where the two modalities are maximally correlated. In this space, a document (x^T , x^I) is represented by two points, p^T that is the projection of x^T and p^I the projection of x^I. Ideally, since they represent the same document, p^T and p^I should be closer to each other than to any other point in the projection space. We propose to create a unified representation for each document, by the following process:

1. define a unifying vocabulary in the projection space,
2. describe both p^T and p^I according to this vocabulary,
3. aggregate both descriptions into a unique representative vector of the document.

Simply said, the "unified vocabulary" is obtained by quantizing the projection space, then $p^T$and p^I are projected to this codebook and sum pooled to get the final representation.

3.2.2. Codebook learning

As for the bag of words (BoW) model, we learn a codebook $C = \{c_1, .., c_k\}$ of k codewords with k-means directly in the projection space. A crucial point is that all the projected points, coming from both textual and visual modalities are employed as input to the k-means algorithm. Hence, the clustering potentially results into three types of codewords (that are centers of the clusters). Some are representative of textual data only, others of visual data only, while some clusters contain both textual and visual projection points. The codebook is thus intrinsically cross-modal and can serve as "common vocabulary" for all the

points in the projection space, whether they result from the projection of a textual content or of a visual one.

3.2.3. MACC representation

A bi-modal document (x^T, x^I) is projected on the KCCA projection space of dimension d into (p^T, p^I). Each of these points is then encoded by its differences with respect to its nearest codewords. The modality-specific representations v^T and v^I result from the concatenation of the d-dimensional vectors v_i^T and respectively v_i^I. The projection space obtained with KCCA has dimension d, so the modality-specific encoded vectors v^T and v^I, as well as the MACC vector v, have a size of $D = d \times k$, where k is the size of the codebook C. The vectors v^T and v^I are component-wise differences of p^T and p^I with some codewords.

There is also another advantage in our context, where some codewords may be representative of "modality-specific" Voronoi cells, i.e. clusters that contain projected points of only one modality after k-means. Therefore, by encoding p^T and p^I according to several codewords, it is more likely to include information from both modalities. Hence, the "modality vectors" v^T and v^I are not exactly modality-specific since they benefit from a sort of "modality regularization" with the multimodal codebook. Yet another advantage is that if p^T and p^I are close enough then they certainly share one or several nearest codewords.

All this indicates that the MACC representation is a soft synthesis of the contributions of both modalities that compensates for the imperfection of the KCCA projection space in the context of bi-modal tasks.

3.2.4. MACC completion with the missing modality

The MACC representation proposed in the previous section is defined when the multimedia document it describes has both a visual and a textual content. But this condition does not hold for several important multimedia tasks. In particular, for cross-modal tasks, data in the reference base and/or the query usually come from only one modality. In such a case, we estimate MACC representations by completing unimodal data with suitable information that concerns the missing modality and is obtained from an auxiliary dataset.

Consider an auxiliary dataset containing m documents where both visual and textual contents are present. Let A be the set of pairs of KCCA projections of the visual and textual features of these documents on the common space, with $A = \{(q^T, q^I)\}$, $q^T \in A^T$, $q^I \in A^I$, $|A| = m$. In practice, the auxiliary dataset could be the training data used to obtain the KCCA space.

To explain the completion process, let us consider a document with textual content only, described by a feature vector x^T that is projected as p^T on KCCA space. The same development could be symmetrically applied to a document having only visual content. A "naive" choice would be to combine p^T with a vector obtained from its μ nearest neighbors among the points projected from the other modality (visual modality in this case). Preliminary experiments (not reported here) have shown that such a strategy is far from being optimal. We propose instead to find the auxiliary documents having similar projected content in the available modality (textual modality in this case) and to use the projections of the visual content of these documents to complete p^T.

3.3. Evaluation and testing

We evaluate the proposed representation for bi-modal classification on the PascalVOC 07 dataset and for cross-modal retrieval on the FlickR 8K and FlickR 30K collections.

A description of Pascal VOC07 is given in Section 2. FlickR 8K (Rashtchian et al., 2010) and FlickR30K (Young et al, 2014) contain 8000 and 31783 images respectively. Each image was annotated by 5 sentences using Amazon Mechanical Turk. These datasets have the same 1000 images for validation and 1000 images for testing. While the training set of FlickR 8K contains 6000 images, the one of FlickR 30K is much larger containing 29783 images. We report the Recall@K metric, i.e. the fraction of times the ground-truth image is found among the top K images.

To represent visual content we use the 4096-dimensional features of the Oxford VGG-Net (Simonyan and Zisserman, 2014), L2-normalized. This representation was shown to provide very good results in several classification and retrieval tasks. To represent texts (sets of tags or sentences, respectively) we employ the features built from Word2Vec (Mikolov et al., 2013), an efficient method for learning vector representations of words from large amounts of unstructured text data. In our experiments, textual features are 300-dimensional L2-normalized vector representations.

3.3.1. Limitations of KCCA projections

As previously mentioned, the common representation space obtained with KCCA only provides a coarse association between modalities. Several data analysis results shown here highlight this problem. Table 5 reports several average distances between KCCA projections of the training data (10022 points in Pascal VOC07 and 12000 points in FlickR 8K). We denote by $d_{intramodality}(I)$ and $d_{intramodality}(T)$ the average within modality distances between image and respectively text projected points.

Next, $d_{intermodality}(sample)$ is the average distance between visual projection and associated textual projection on the KCCA space of a training sample, while $d_{intermodality}(overall)$ is the average distance between visual and textual projections over all training data. The values obtained in Table 5 show that projected points are closer to their within-modality neighbors than to their corresponding points in the other modality.

Average Distance	Pascal VOC07	FlickR 8K
$d_{intramodality}(I)$	1.18 ± 0.16	1.17 ± 0.13
$d_{intramodality}(T)$	1.11 ± 0.19	0.75 ± 0.13
$d_{intermodality}(sample)$	1.39 ± 0.07	1.02 ± 0.12
$d_{intermodality}(overall)$	1.42 ± 0.06	1.28 ± 0.10

Table 5. Average distances between projections on KCCA space.

For a better visualization, we computed the centers of gravity of the visual and respectively textual points, then projected all the points onto the line that joins these two centers. In Figure 6, we report the distribution of these projected points. The separation in the KCCA space between data points from the two modalities appears very clearly, for both Pascal VOC07 and FlickR 8K datasets.

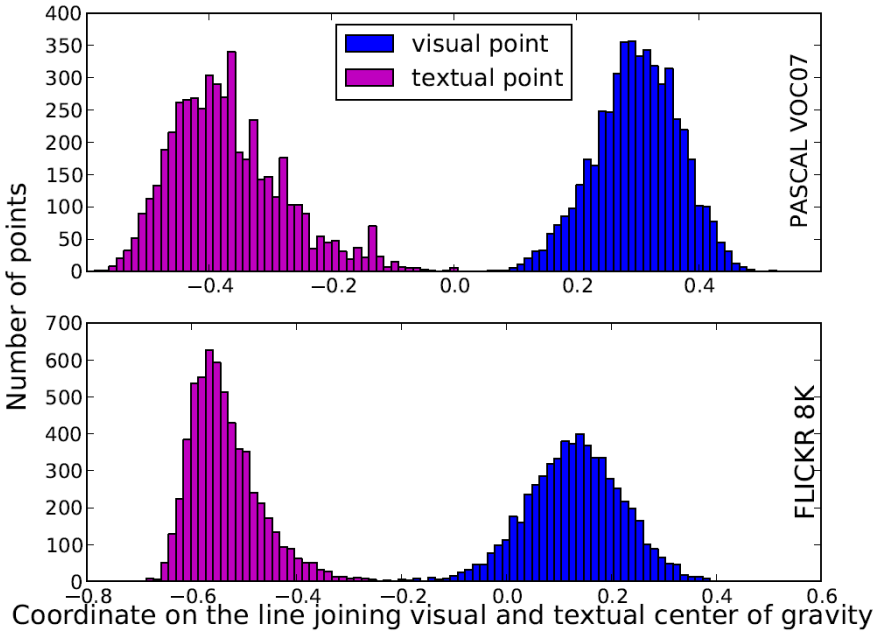

Figure 6: Separation between modalities on the KCCA space.

Given this separation between modalities on the common space, the clusters we obtain with k-means contain mostly data from a single modality (image or text). Table 6 shows the number of clusters associated to each modality in Pascal VOC07 and FlickR 8K. They are qualified as “visual” or “textual” according to the majority of points they contain, but each cluster can have both visual and textual points. The clusters are used for codebooks in the following. The value of k is chosen on a validation set.

Dataset	# projected points	k	# visual clusters	# textual clusters
Pascal VOC07	10022	16	12	4
FlickR 8K	12000	8	6	2

Table 6. Distribution of textual and visual KCCA-projected points into clusters.

3.3.2. Image classification

The KCCA is learnt on the 5011 training data, with both visual and textual content. We used the seminal KCCA implementation from (Hardoon et al., 2014). The dimension of the “common” projected space is set to d = 150. All 5011 training data are then projected on this common space and a codebook C is learnt with k-means from this set (2×5011 = 10022 points) for k ∈ {8, 16, 32}.

The first evaluation considers the classification of documents having both a visual and a textual content, such that a MACC representation (of size d × k) of each document is directly obtained using the previously built codebook. For each category, we learn a SVM classifier with linear kernel, following a one-versus-all strategy.

With such settings, the best result we obtain on the testing set is a mAP of 90.37, with (k = 16), resulting into a 2400-dimensional MACC representation. However, when a full cross-validation is conducted on the training set, we obtain a mAP of 90.12 with k = 32.

Baseline	Size of representation	mAP (%)
VGG-Net	4096	86.10
Word2Vec	300	82.50
VGG-Net+Word2Vec	4396	86.16
$KCCA_{img}$	150	84.84
$KCCA_{img}$	2400	85.29
$KCCA_{txt}$	150	82.01
$KCCA_{txt}$	2400	82.60
MACC	2400	90.12

Table 7. Pascal VOC07: comparison with baselines.

We compare our image classification result to several baselines that uses the same features as MACC in Table 7. For the VGG-Net (respectively Word2Vec) baseline, classifiers are trained and tested on VGG-Net (respectively Word2Vec) features only, i.e. using the visual (respectively textual) content alone. For the VGG-Net+Word2Vec baseline, representations for both training and testing data are obtained by early fusion, i.e. by concatenating VGG-Net features and Word2Vec features. For the $KCCA_{img}$ (respectively $KCCA_{txt}$) baseline, the visual (respectively textual) features are first projected on the KCCA common space for both training and testing data and then used for classifiers learning. We consider two different sizes of the KCCA common space, 150 and 2400, so that the results can be compared to our 2400-dimensional MACC representation (built from a 150-dimensional common space, with 16 codewords). The results in Table 6 show that the MACC approach outperforms all the mentioned baselines.

3.3.3. Image retrieval

In the context of image retrieval, KCCA is learnt on the 6000 training documents with both visual and textual content. To select the parameters, a grid search is performed employing the validation set of 1000 documents. The visual and textual features of the training documents are then all projected on this common space and a codebook is learnt from this set of 12000 (= 2 × 6000) points.

For the text-to-image retrieval task the training dataset of FlickR 8K is used as auxiliary dataset A. As shown in Table 8, the proposed approach has higher R@1, R@5 and R@10 than the other image retrieval methods in the recent literature on the FlickR 8K dataset.

Our method also significantly outperforms several recent deep learning approaches (Karpathy and Fei-Fei, 2015) (Chen and Zitnick, 2015) that use content representation similar to ours. Furthermore, the MACC representation achieves better results than (Chen

and Zitnick, 2015), the current state-of-the-art on both FlickR 8K and FlickR 30K image retrieval, in which the VGG-Net features are also employed.

Approach	R@1	R@5	R@10
(Karpathy and Fei-Fei, 2015)	15.2	37.7	50.5
(Chen and Zitnick, 2015)	18.5	45.7	58.1
MACC (F8k)	33.9	65.6	77.5
MACC (F30k)	35.3	66.0	78.2

Table 8. Image retrieval results on FlickR 30K. MACC parameters are cross-validated on FlickR 8k (F8k) or FlickR 30k (F30k)

3.4. Implementation and usage

There are four main components of the MACC representation extraction tool: extracting visual features, extracting textual features, learning the MACC model (learning the KCCA common space and the codebook) and generating the MACC representation. Note that the model is learned offline.

The implementation of the visual feature extraction uses the same Caffe framework pipeline as the one for visual concept detection (see Section 2). The difference stems in the CNN model that is used. The ImageNet reference model is replaced here by the VGG[5] model. (Simonyan and Zisserman, 2014). The resulting features have 4096 dimensions and are extracted using a GTX Titan Black GPU card.

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

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

For textual feature extraction, we use the Genism[6] toolkit implementation of the word2vec (Mikolov et al., 2013) “skip-gram” embeddings model. The model is pre-trained on the GooleNews corpus[7].

Textual feature extraction is realized with:

> **extract_embeddings.py** [w2v-model] [text-in] [text-vec] [size] [threads]

Finally, the MACC representation is extracted as follows:

> **extract_macc.py** [vis_desc] [text_desc] [kcca-model] [codebook] [mmodal_desc]

The commands and parameter files are explained in Table 9.

Program	Description
extract_features.bin	Binary for feature extraction provided with Caffe
extract_embeddings.py	Python script for textual embeddings feature extraction based on

[5] The Caffe implementation of the VGG model is publicly available at https://gist.github.com/ksimonyan/211839e770f7b538e2d8#file-readme-md

[6] https://radimrehurek.com/gensim

[7] https://code.google.com/archive/p/word2vec/

	the Gensim toolkit.
extract_macc.py	Python wrapper used for extracting the MACC descriptors. The scripts first calls Matlab dependencies for generating the KCCA features before compressing them using the pre-computed codebook.
Parameter	**Description**
caffe-model	CNN model used to compute features
proto-file	Configuration file needed to compute features
caffe-layer	Layer of the CNN architecture used for feature extraction. FC7 for the Caffe reference model
tmp-leveldb	File for output features in leveldb format. Deprecated
num-batches	Number of batches used for faster extraction. Typically one batch with several images.
vis-ascii	File for output features in ASCII format.
mode	Indicates if GPU or CPU should be used for feature extraction. GPU is strongly recommended since CPU extraction is very slow
gpu-name	If GPU is used and there are several available, indicates which one should be preferred. This argument is optional and points to gpu-name=0 (i.e. default GPU).
w2v-model	Word2Vec model trained on a large text corpus.
text-in	File with the textual descriptions in ASCII format.
text-vec	Output file which stores the corresponding embeddings features.
size	The size of the embeddings feature vector. Possible values are 50, 100, 300.
threads	Number of threads used for extracting the textual features.
vis_desc	File containing the visual descriptors descriptions of the multimedia items for which the MACC descriptor is to be extracted.
text_desc	File containing the corresponding textual descriptions of the multimedia items for which the MACC descriptor is to be extracted.
kcca-model	Pre-trained kcca model.
codebook	Codebook learned on a training set used in extracting the MACC descriptor.
mmodal_desc	Output file containing the single bi-modal MACC representation of multimedia items.

Table 9. MACC feature extraction usage.

The implementation is at a prototype level and serves as a proof of concept for the extraction and use of the novel multimodal descriptor.

4. Private/non-private image classification

Private/non-private image classification is a new research topic that arose out of the need for protecting users from sharing images featuring sensitive content to Online Social Networks. This line of research is highly relevant to USEMP since a privacy-aware image classification module can significantly contribute towards users' privacy awareness and control. Our initial work on this topic was documented in D5.5 where we highlighted the need for developing personalized image privacy classification models and conducted preliminary experiments that revealed the limitations of generic models and demonstrated the potentials of model personalization. Those experiments were based on a preliminary version of a new real-world dataset (*YourAlert*) that we created for the purposes of this study. In months M28-M35[8] of the project, we continued our work on this topic. In particular, we a) expanded the *YourAlert* dataset and re-evaluated generic and personalized privacy classification models on the expanded version to confirm our previous findings, b) we worked towards providing easily comprehensible explanations of the classification outputs using a new semantic feature representation (*semfeat-lda*) that is based on a new privacy aspect modeling approach, c) explored the potential of discovering groups of users with similar privacy concerns and providing meaningful visualizations of the different groups (again based on *semfeat-lda*) and, importantly, d) integrated a pilot version of the privacy-aware image classification module into DataBait, and evaluated it in the context of the final pilot studies. All these latest developments are detailed in this section[9].

4.1. Related work

Most modern OSNs allow users to control the privacy settings of their shared content. Yet, the typical user finds it difficult to understand and correctly configure the offered access control policies (Madejski et al., 2012). As a result, several studies (Liu et al., 2011, Madejski et al., 2012) have identified a serious mismatch between the desired and the actual privacy settings of online shared content. This discrepancy motivated the development of mechanisms that aid users in selecting appropriate privacy settings. In the work of Naini et al. (2015), for instance, the authors focused on Facebook posts and evaluated prediction models that make use of users' previous posts and profile preferences in order to suggest appropriate privacy settings for new posts. Despite achieving high performance, the authors noticed differences in user behaviors and concluded that personalized privacy models could further improve results.

As we had already pointed out in D5.5, the work of Zerr et al. (2012) was among the first to consider the problem of privacy-aware image classification. That work focused on developing models that capture a generic ("community") notion of privacy by using a training dataset (PicAlert) consisting of collectively annotated (as private or public), publicly available Flickr photos. Extending that work, Squicciarini et al. (2014) experimented with combinations of visual and metadata-derived features and achieved better prediction accuracy on the same dataset. Squicciarini et al. (2014) also attempted to solve a more complex privacy

[8] The bulk of research and development work was carried out in M28-M33, while an analysis of data collected during the final pilot studies was conducted near the end of the project.
[9] A research paper describing this work was recently presented at ICMR 2016 (Spyromitros-Xioufis et al., 2016).

classification problem where three types of disclosure were defined for each image (view, comment, download) and the task was to assign one of five privacy levels ('Only You', 'Family', 'Friends', 'SocialNetwork', 'Everyone') to each type of disclosure. As in (Zerr et al., 2012), their models captured only a generic perception of privacy.

Differently from the majority of previous works, in D5.5 and, more recently, in (Spyromitros-Xioufis et al., 2016) we highlighted the limitations of generic image privacy classification models and proposed effective personalization methods. To the best of our knowledge, (Buschek et al., 2015) is the only other work that considers privacy classification of personal photos. However, Buschek et al. (2015) evaluate only purely personalized models, assuming that each user provides sufficient amount of feedback. In contrast, our method achieves high performance even at the presence of very limited user-specific feedback by leveraging feedback from other users. Moreover, while Buschek et al. (2015) use only metadata-based features (location, time, etc.) and simple visual features (colors, edges, etc.), we employ state-of-the-art CNN-based semantic visual features that facilitate comprehensible explanations of the classification outputs.

4.2. Method description

4.2.1. Personalized image privacy classification models

In D5.5 we explained the need for personalized privacy classification models and briefly described a personalization method that in addition to user-specific training examples also uses training examples provided by other users (via explicit or implicit feedback) in order to achieve good generalization performance even in cases where the amount of user feedback is limited. Here, we describe our method in more detail and point to its relation to methods from the domains of *transfer* (Pratt, 1992) and *multi-task* learning (Caruana, 1997).

Provided that sufficient amount of feedback is available from each user, one could rely only on user-specific examples for training personalized privacy classification models. This, however, might require considerable effort from the user and cannot be taken for granted. As a result, user-specific privacy classification models might not be able to generalize well. To overcome this problem, we propose the development of *partially-personalized* models that are learned using a combination of user-specific training examples and examples from other users. The intuition behind such an expansion of the training set is that, although each user has a personal notion of privacy, there are also similarities between different users (since everyone is affected to some degree by general trends and norms) and the expansion of the training set is tailored exactly towards the exploitation of such similarities. Importantly, in order to retain the personalized nature of the models, we assign higher weights to the user-specific examples, effectively increasing their influence on the resulting model.

More formally, given a set of users $U = \{u_1, u_2, \dots, u_k\}$ and assuming that each user $u_i \in U$ has provided ground truth annotations for a set of personal images $I_{u_i} = \{im_{u_i}^1, im_{u_i}^2, \dots, im_{u_i}^n\}$, a user-specific dataset $D_{u_i} = \{(\boldsymbol{x}_{u_i}^1, y_{u_i}^1), (\boldsymbol{x}_{u_i}^2, y_{u_i}^2), \dots, (\boldsymbol{x}_{u_i}^n, y_{u_i}^n)\}$ can be constructed where $\boldsymbol{x}_{u_i} = [x_{1_{u_i}}, x_{2_{u_i}}, \dots, x_{d_{u_i}}]$ is a vector representation of im_{u_i} and y_{u_i} equals 1 if the image is annotated as private, and 0 otherwise. The typical approach is to train a personalized classifier $h_{u_i}: \boldsymbol{X} \rightarrow Y$ (where $\boldsymbol{X} = R^d$ and $Y = \{0, 1\}$ are the domains of $\boldsymbol{x}$ and y respectively) using only examples from D_{u_i}. Instead of that, we propose that each classifier h_{u_i} is trained on $\bigcup_{i=1}^{k} D_{u_i}$, i.e. the union of all user-specific datasets, and personalization is achieved by

assigning a higher weight w to the examples of D_{u_i}. Example weights are directly handled by some learning algorithms (e.g. decision trees) while other learning algorithms can be "forced" to take weights into account by including duplicates of specific examples in the training set. The effect of weighting is that the classifier is biased towards correct prediction of more highly weighted examples and is commonly used in supervised learning techniques, e.g. cost-sensitive learning (Elkan, 2001) and boosting (Freund & Schapire, 1996).

We note that our approach resembles techniques from the domains of transfer and multi-task learning (Pratt, 1992, Caruana, 1997), commonly referred to as *instance sharing* or *instance pooling*. In fact, if we consider the privacy classification of the images of each user as a different learning task, the problem of personalized image privacy classification can be considered as an instance of multi-task learning. Multi-task learning methods are known to work better than methods that treat each learning task independently in cases where the tasks are related and there is lack of training data for some of the tasks (Alvarez et al., 2012), two conditions that hold for the problem that we tackle here.

4.2.2. YourAlert: a realistic image privacy classification benchmark

The need for building and realistically evaluating personalized image privacy classification models, motivated the composition of a new benchmark dataset through a user-study that asked users to provide privacy annotations for photos of their personal collections. In D5.5 we presented a preliminary version of this dataset consisting of 584 photos contributed by 10 different users and described the characteristics that make it stand out from existing ones, i.e. a) the fact that it consists of personal user photos including really private ones and b) the fact that it captures the variability in privacy perceptions of different users. In the months that followed, more users participated in the user-study and the dataset was thus expanded.

In total, we received feedback from 27 users (22 males and 5 females), with ages ranging from 25 to 39 years. Each user contributed approximately 16.4 private and 39.5 public photos (on average) for a total of 1511 photos. Importantly, the final version of the dataset (features and privacy annotations) was made publicly available for future benchmarks[10]. Note that those users were directly recruited by CERTH and CEA and were different than the ones participating in the pilot studies. This was done for the following two reasons: a) using a completely independent set of users (and the corresponding observations) for training the private image classification models would ensure that when the module was tested through DataBait, the test users would be totally "unknown" (from a machine learning point of view) to the system, b) at the time of the first experiments, the integrated system did not provide facilities for image privacy-oriented annotation, hence it was necessary to quickly set up an independent mechanism to collect user feedback.

4.2.3. Semantic visual features: a privacy aspect modeling approach

One of the main limitations of previous work on privacy-aware image classification was the fact that classification outputs were either not justified or justified in a non-intuitive and hardly-comprehensible manners. In D5.5, we described how using *semfeat*, a type of semantic visual features, we can provide intuitive feedback about why an image is classified as private or public, as well as build user privacy profiles and gain interesting insights into

[10] http://mklab.iti.gr/datasets/image-privacy/youralert/

users' privacy perceptions. In this subsection, we briefly review *semfeat* and then introduce a new type of semantic features, called *semfeat-lda,* that deal with a limitation of *semfeat* – the fact that *semfeat*'s vocabulary is not privacy oriented – and allow for even more intuitive justifications and privacy insights.

Figure 7: A hardly comprehensible justification (green rectangles highlighting the most discriminative local patches) provided for a private classification by PicAlert (Zerr et al., 2012).

The *semfeat* features are obtained by exploiting the outputs of a large array of classifiers, trained on images represented with standard convolutional neural network features (*cnn*). In D5.5, the last fully connected layer of the Caffe reference model (Jia, 2013) was used to extract the *cnn* features. Here we use the more recent VGG-16 model (Simonyan & Zisserman, 2014) which obtained one of the top results during the ImageNet 2014 challenge. The VGG-16 model consists of 16 layers and is learned with the training set of the ImageNet ILSVRC 2014 dataset (Russakovsky et al., 2015) that includes 1,000 specific classes and approximately 1.2 million images. These classes cover a wide range of domains and the obtained model has thus good performance in transfer learning tasks as attested by (Ginsca et al., 2015). We use the output of the last fully connected layer (*fc7*), which consists of 4,096 dimensions. Given the newer *cnn* features described above, *semfeat* are calculated as described in D5.5. In short, 17,462 concept models are learned independently as binary classifiers with a ratio of 1:100 between positive and negative examples, with the negative class including images of ImageNet concepts that are disjoint from the 17,462 concepts that we modeled. Finally, the features are sparsified by retaining only the top $n = 100$ classifier outputs and setting the rest equal to zero.

Compared to *cnn* and other low-level visual features, *semfeat* have the advantage that they can be used to justify classification outputs. An example is shown in Figure 8 where a privacy-oriented classification is accompanied by an automatically generated cloud of the most discriminative image tags. However, having been constructed for general purpose concept detection, the *semfeat* vocabulary contains many concepts that are unrelated to privacy and/or are too specific (e.g. 'knitwear', 'Freudian', 'smoker' - 'cigar-smoker'). As a result, many of the most discriminative image tags cannot be easily linked to privacy or are even confusing.

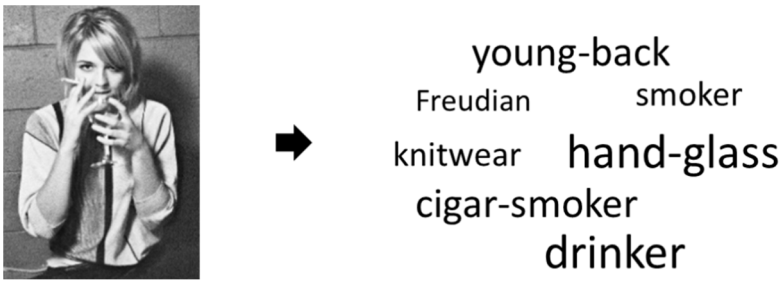

Figure 8: A private image along with an automatically generated cloud of the most discriminative *semfeat* concepts.

To address this limitation, we developed a privacy aspect modeling approach that maps the concepts of the *semfeat* vocabulary into a number of privacy-related latent topics using *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003). More specifically, each image is treated as a document consisting of its top $n = 10$ *semfeat* concepts and a private image corpus is created by combining the private images of the YourAlert and PicAlert datasets. LDA, and in particular the Mallet implementation (McCallum, 2002), is then applied on this corpus to create a topic model with 30 topics. Among the detected topics, six privacy-related ones are selected after manual inspection: '*children'*, '*drinking'*, '*erotic'*, '*relatives'*, '*vacations'*, '*wedding'* (Table 10). Given such a topic model, the *semfeat* concepts of each image can be mapped to the privacy-related topics (using Gibbs sampling inference), leading to a new, higher-level semantic representation that we refer to as *semfeat-lda*. Figure 9 shows the projection of the most discriminative *semfeat* concepts of Figure 8 to the six privacy-related topics. Obviously, this assignment to privacy-related topics represents an even more intuitive justification of the classifier's output.

Topic	Top-5 *semfeat* concepts assigned to each topic
children	dribbler child godson wimp niece
drinking	drinker drunk tippler thinker drunkard
erotic	Slattern erotic cover-girl maillot back
relatives	great-aunt second-cousin grandfather mother great-grandchild
vacations	seaside vacationer surf-casting casting sandbank
wedding	groom bride celebrant wedding costume

Table 10: Privacy-related topics along with top-5 *semfeat* concepts assigned to each topic.

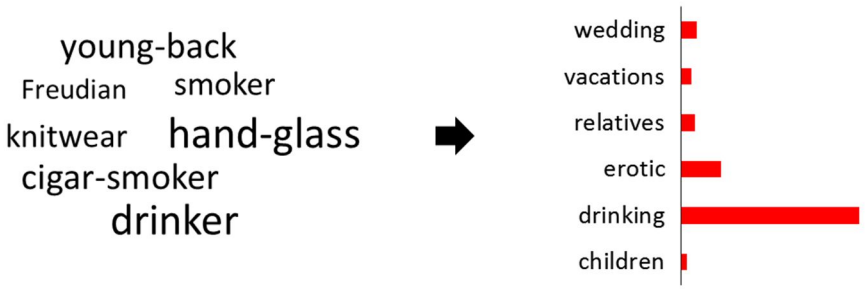

Figure 9: The projections of the most discriminative *semfeat* concepts of Figure 8 to the six privacy-related topics.

4.3. Evaluation and testing

In D5.5 we evaluated the performance of generic and personalized image privacy classification models on a preliminary version of the YourAlert dataset. Here we repeat the evaluation using the full version of the dataset to confirm our previous findings. Moreover, we

evaluate additional variations of personalized models and show how the new *semfeat-lda* representation can be used to create user privacy profiles and identify groups of users with similar image privacy concerns. We follow the same experimental setup as in D5.5, i.e. we use L2-regularized logistic-regression as the classification algorithm (with tuning of the regularization parameter) and the area under the ROC curve (AUC) as the measure of classification accuracy.

4.3.1. Limitations of generic image privacy classification models

In this subsection, we evaluate the performance of generic image privacy classification models when applied in a realistic setting where different users have different perceptions of image privacy. To this end, we conduct the following experiment: a generic image privacy classification model is trained using a randomly chosen 60% of the PicAlert dataset and then tested on: a) the remaining 40% of PicAlert and b) the YourAlert dataset. In the first case, we have an idealized evaluation setting, similar to the one adopted in (Zerr et al., 2012), while in the second case we have an evaluation setting that better resembles the test conditions that a privacy classification model will encounter in practice. To ensure the reliability of the performance estimates, we repeat the above evaluation procedure five times (using different random splits of PicAlert) and take the average of the individual estimates.

Figure 10 shows the AUC scores obtained on PicAlert (light blue bars) and YourAlert (orange bars) when different visual features are used. Besides *cnn* and *semfeat,* on PicAlert we also evaluate the performance using quantized SIFT (*bow*) and edge-direction coherence (*edch*) features, the best performing of the visual features used in (Zerr et al., 2012). The results on PicAlert indicate that generic models built with *cnn* and *semfeat* lead to significantly better results than models built with *edch* and *bow*, as we obtain a near-perfect 0.95 AUC score (about 20% better than the AUC score obtained with *bow*). However, we notice that the performance drops significantly (about 24%) when the models are applied on YourAlert. These results are in accordance with those presented in D5.5.

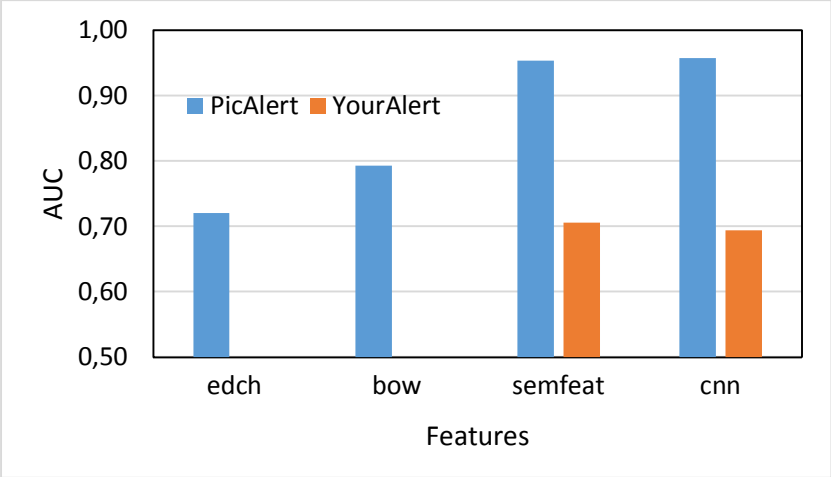

Figure 10: Performance of generic image privacy classification models on PicAlert and YourAlert.

Figure 11 presents a per-user performance breakdown for generic models based on *cnn* and *semfeat* features (i.e. a separate AUC score is calculated for each user based on his/her own images). We note that there is a large variability in performance across users. For instance, using *semfeat* features, near perfect AUC scores are obtained for users $\{u_1, u_8, u_{27}\}$ while the AUC scores are worse than random for users $\{u_9, u_{23}, u_{16}, u_{14}\}$ suggesting that the privacy

perceptions of these users deviate strongly from the average notion of privacy. For this type of users, as well as for those for which the performance of the generic models is close to random (about 40% of users), building personalized privacy classification models is essential to develop a useful alerting mechanism. The performance of personalized models is studied in the next subsection.

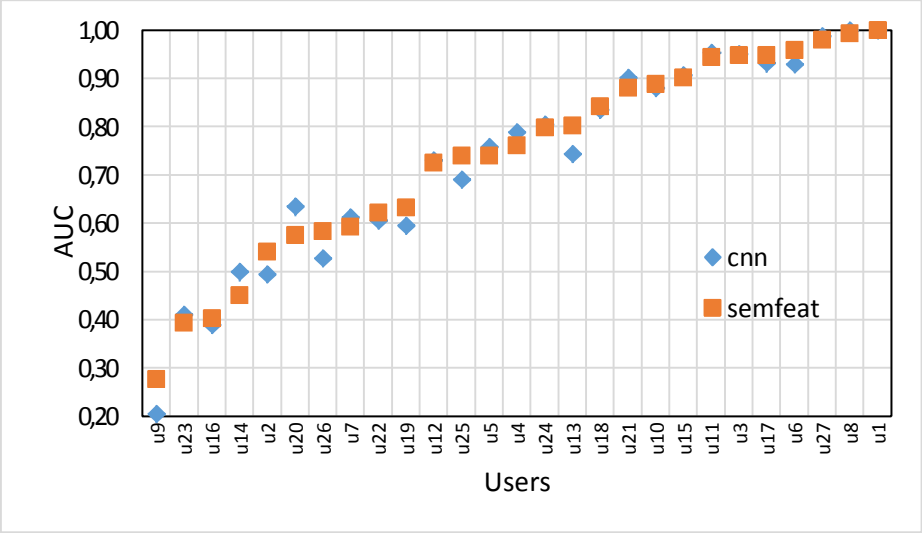

Figure 11: Per-user performance of generic models based on *semfeat* and *cnn* features.

4.3.2. Personalized image privacy classification models

This subsection compares the performance of generic privacy classification models to that of models leveraging user feedback in order to adapt to specific users. Specifically, we evaluate two types of personalized model on YourAlert.

- *user*: Fully personalized models that use only user-specific training examples, i.e. a specific model is built for each YourAlert user from examples that have been provided by this user only.
- *hybrid*: Partially-personalized models that use a mixture of user-specific and generic training examples. We experiment with two sources of generic examples: a) from PicAlert (*hybrid-g* variant), b) from other users of YourAlert (*hybrid-o* variant). As in the case of *user*, a different model is built for each user. In both variants we experiment with assigning different weights on the user-specific examples.

In theory, *user* models are expected to perform better when a sufficient number of user-specific examples are available, while *hybrid* models are expected to be advantageous with a limited amount of user feedback.

Figure 12 plots the AUC scores obtained on YourAlert by *user* and *hybrid* models trained on $\{5,10,15,20,25,30,35\}$ user-specific examples using *semfeat* and *cnn* features. The upper part of the figure reports results for the *hybrid-g* variant while the lower part reports results for the *hybrid-o* variant. For each *hybrid* variant, four variations are constructed, each one using a different weight ($w \in \{1,10,100,1000\}$) for the user-specific examples to facilitate a study of the impact of the weight parameter. In addition to the performance of these personalized models, the figure also shows the performance of two types of generic model to allow a direct comparison: a) *generic*: a model trained on a subset of PicAlert (the same subset as the one used in *hybrid-g*) and b) *other*: a model trained using only examples from other

YourAlert users, i.e. a different generic model is built for each user, using the same generic examples as the one used in *hybrid-o*.

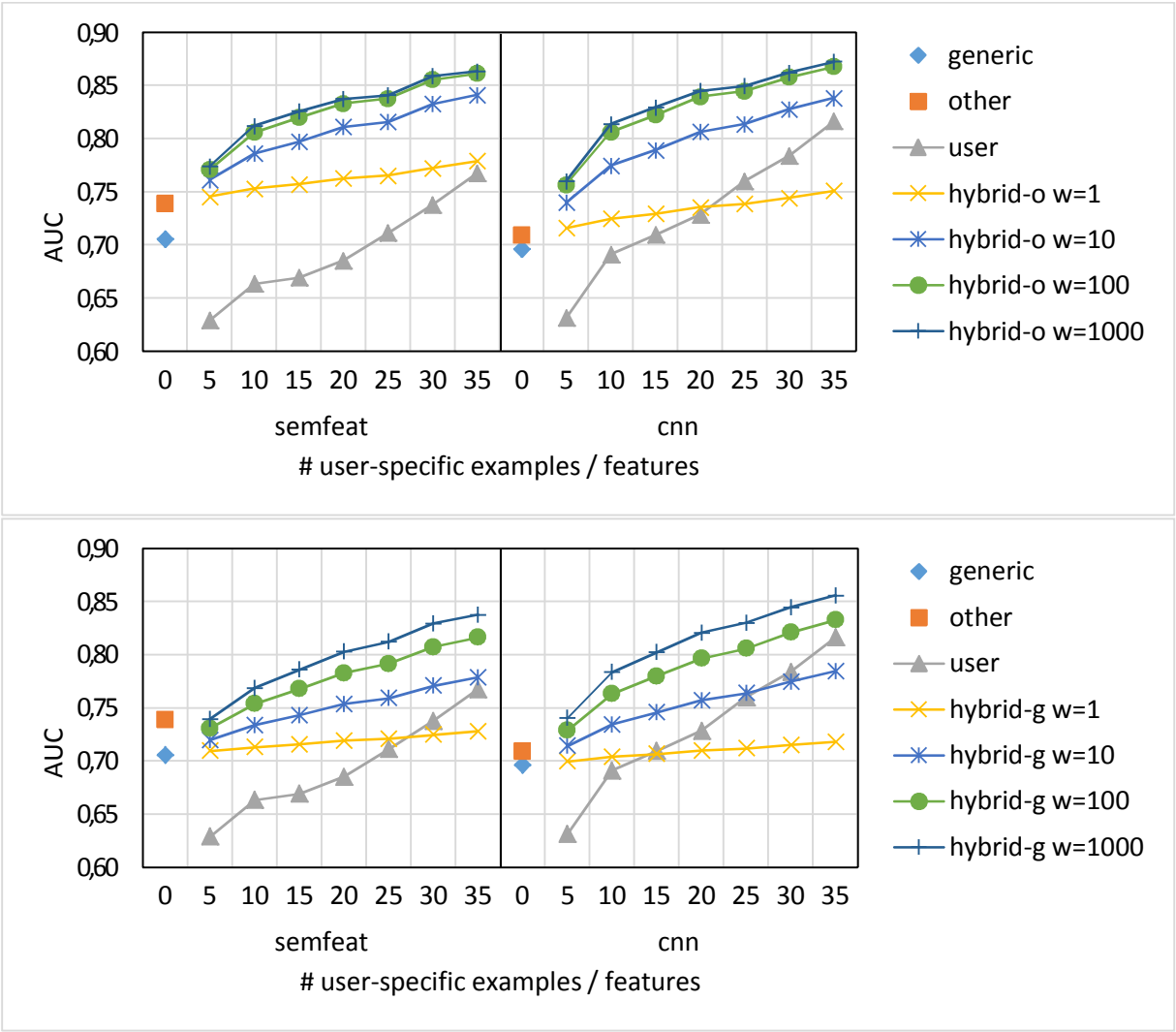

Figure 12: Performance of personalized models as a function of user-specific training examples. *hybrid-o* (top) vs *hybrid-g* (bottom).

With respect to the generic models, we see that *other* has similar performance with *generic* when *cnn* features are used and better in the case of *semfeat* features. These results suggest that although the examples of YourAlert come from users that adopt a personal, potentially different, notion of privacy, they are equally useful as the PicAlert examples for learning a generic privacy model.

With respect to the personalized models, we see that the performance of *user* models increases sharply as more user-specific training examples become available. When *semfeat* features are used we see that *user* models obtain similar performance with the generic models (*generic* and *other*) with as few as about 30 examples. The situation is even better when *cnn* features are used as we see that the performance of *user* models catches up to the performance of the generic models with as few as 15 examples and improves by about 15% when 35 user-specific examples are used.

With regard to the partially-personalized, *hybrid* models we observe that they outperform significantly the fully personalized *user* models (with both types of features), especially for

smaller numbers of user-specific training examples. As expected, the gap closes as more user-specific training examples become available. However, we see that for all values of user-specific examples (up to at least 35) *hybrid* models provide significantly better performance than both *user* and the generic models.

Comparing the two variants of the *hybrid* models (*hybrid-g* and *hybrid-o*), we see that they exhibit similar performance, in accordance with our previous observations about *generic* and *other* models. Importantly, in both cases we see that assigning a higher weight to user-specific examples is crucial for obtaining better performance. In the case of *hybrid-g* we observe that the performance keeps improving as we increase w, with $w = 1000$ leading to the best results. In the case of *hybrid-o*, using $w > 100$ does not improve the performance further. This difference is attributed to the fact that the initial ratio of user-specific to generic examples is higher in the case of *hybrid-o* models. Overall, we see that the best personalized model (*hybrid-o* with *cnn* features) can boost the performance of the best generic model (*other* with *semfeat* features) by about 4% when the user provides feedback for 10 images to about 18% when the feedback increases to 35 images.

Figure 13 presents a per-user performance breakdown for *generic, user and hybrid-o* ($w = 1000$) *models based on cnn* features (*user* and *hybrid-o* use 35 user-specific examples). *hybrid-o* and *user* provide better performance than *generic* for the majority of users, particularly for those that are poorly predicted by the *generic* model. Moreover, we see that *hybrid-o* is equally good or better than *user* with very few instances of a noticeable degradation.

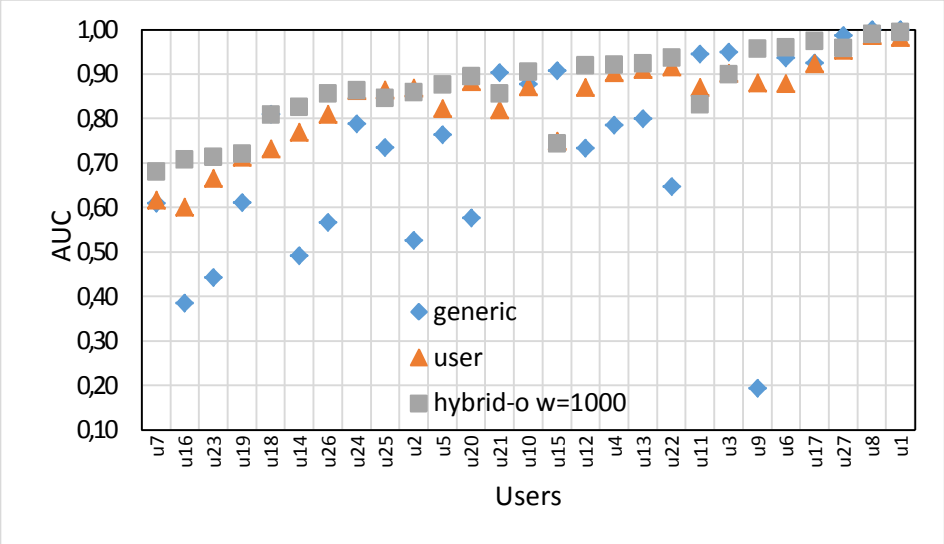

Figure 13: Per-user performance of *generic*, *user* and *hybrid-o* ($w = 1000$) models based on *cnn* features.

4.3.3. Image privacy insights via *semfeat-lda*

Besides facilitating easily comprehensible explanations of privacy classifications (as exemplified in Figure 9), *semfeat-lda* features can help in creating user privacy profiles. To construct a privacy profile for each user we compute the centroid of the *semfeat-lda* vectors of his/her private images. This vector facilitates a summary of the user's concerns with respect to the six privacy-related topics that were identified by the privacy-aspect modelling approach. Given such a representation for each user, cluster analysis can be performed to identify recurring privacy themes among users. To illustrate this use of *semfeat-lda*, we

performed k-means ($k = 5$) clustering on the users of YourAlert and present the clustering results in Figure 14. We see that each cluster captures a different privacy theme. Users clustered at *c0*, for instance, are primarily concerned about preserving the privacy of their *vacations* while users clustered at *c2* are mainly concerned about the privacy of *children* and of photos related to *drinking*.

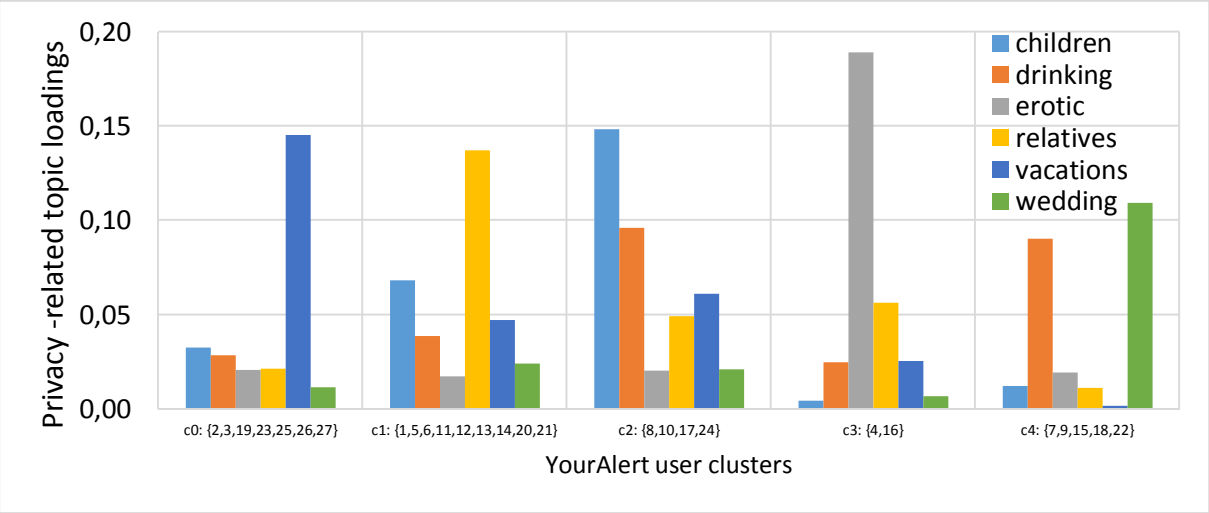

Figure 14: Clustering of YourAlert users based on privacy-related topics.

4.4. Implementation and usage

In terms of implementation, in D5.5 we provided an executable jar file implementing the presented image privacy classification methods and tag-cloud visualizations. In addition to that, we now make available the complete source code of our methods and our experimental testbed as part of a Github project (https://github.com/MKLab-ITI/image-privacy). The project page contains additional information on how to replicate the experimental results as well as a description of the expected data format and pointers to the datasets that we used.

Most importantly, significant effort was made to develop an image privacy classification module and integrate it into Databait. This integration was designed such that it could serve several goals simultaneously. In particular, we wanted to: a) increase the awareness of Databait users with respect to photos of private nature that they might have shared online and help them align the intended sharing settings with the actual ones for each individual photo, b) evaluate the performance of our image privacy classification models in a real-world setting, and c) collect feedback from Databait users that will allow us to further improve the performance of our models as well as make them more expressive (able to classify a user's photos into finer-grained privacy classes and to predict the type of personal information that each photo could potentially reveal).

Having the above goals in mind, we designed[11] an extension of the Databait interface that is depicted in Figure 15. As shown in the figure, to avoid making drastic changes to the experience of current users, we integrated all the changes into the existing “photo insights” view of Databait. More specifically, users still browse their images by clicking on particular concepts, but each image is now surrounded by a border with a color that indicates whether

[11] We designed a mock-up of the extension to the UI that was then implemented and is currently an integral part of Databait.

the image is predicted as private ("sensitive") or public ("less sensitive") by our image privacy classification algorithm. A small descriptive text was added to the bottom-left of the page to explain the meaning of the border colors and to explain that the accuracy of the algorithm can be improved if the user clicks on an image and answers two multiple-choice questions, thus motivating user feedback. When an image is clicked, instead of navigating the user to the image's page on Facebook, a pop-up window opens that, in addition to the image itself, a cloud of the most prevalent image tags, and the image's privacy prediction, also shows two multiple choice questions that we ask users to answer.

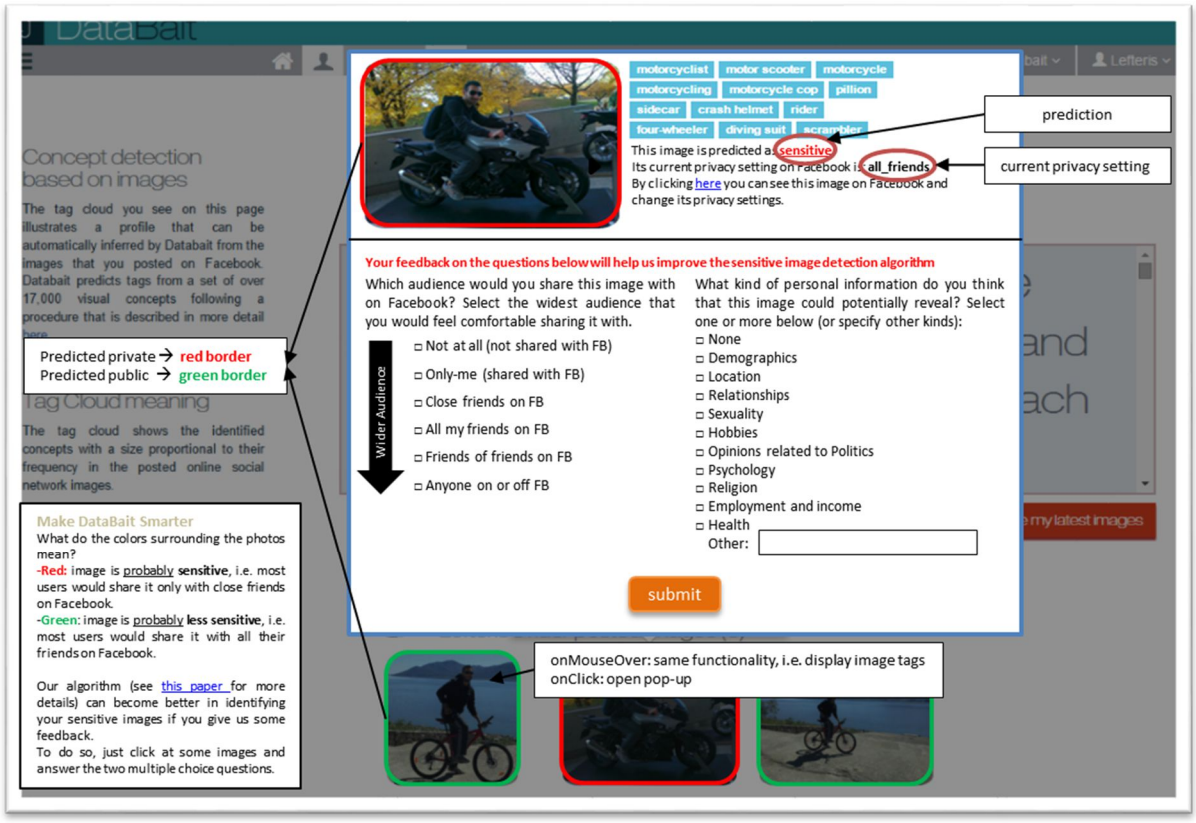

Figure 15: Photo privacy extension of Databait.

The first question (Q1) asks users to indicate the widest audience that they would feel comfortable sharing the image with on Facebook among the following six options: 'Not at all (not shared with Facebook)', 'Only-me (shared with Facebook)', 'Close friends on Facebook', 'All friends on Facebook', 'Friends of friends on Facebook', 'Anyone on or off Facebook'. The users' answers to this question will help us achieve several goals simultaneously: a) we will be able to evaluate the accuracy of the provided predictions; b) it will allow us to build personalized models for the users of Databait (since during the initial deployment of the module there is no user-feedback available, the predictions for all users are inevitably based on a generic image privacy classification model[12]); c) it will allow us to develop image privacy classification models able to classify a user's photos into finer-grained privacy classes, corresponding to the different OSN audiences photos can be shared with.

[12] We use the generic model that lead to the best performance according to our experiments on the YourAlert dataset, i.e. *other* with *semfeat* features.

The second question (Q2) asks users to indicate one or more kinds of personal information that each photo could potentially reveal. The list includes the 10 higher level privacy dimensions described in WP6 but the user is also allowed to optionally specify additional kinds. The users' answers to this question will help us develop image privacy classification models that in addition to predicting the sensitivity of each image, will also be able to predict the privacy aspect that each image is associated with. These predictions can help in explaining the classifications outputs, acting complementarily to the justifications that can be provided by the privacy aspect modelling approach presented in Subsection 4.2.3.

Another important feature of the photo privacy extension is the display of the image's current privacy setting on Facebook. This is important because it facilitates easy identification of images whose sharing settings are different from the suggested (predicted) or the intended ones. It should be stressed out that the current version of Facebook's API (v2.6) provides information about the privacy settings only at an album-level. However, many types of Facebook photos (e.g. profile pictures, cover photos, mobile uploads, etc.) have individually defined privacy settings - that do not match the settings of the album they belong to – that cannot be directly retrieved from the API. To overcome this limitation, we came up with a workaround that allowed us to retrieve the individual privacy settings of some of these photos. In particular, we noticed that some photos are part of Facebook posts, in which case their privacy settings match the privacy settings of the corresponding posts. In these cases, we replaced the album-level privacy settings of the photos with the ones from the corresponding posts. Note, however, that this could not be done for all photos and, as a result, the displayed information might not always be accurate.

4.5. Analysis of Pilot User Feedback

During the last months of the project, the integrated version of the image privacy classification module was used and evaluated by the users that participated in the final pilots. In particular, users had the chance to see the extended user interface and were encouraged to provide their feedback on the two privacy-related questions that we described in the previous subsection. In this subsection, we present an analysis of the collected feedback which allows the extraction of some interesting conclusions with respect to users' concerns regarding image privacy as well as the usefulness of the developed image privacy classification module.

In total, we collected feedback for 655 images from 54 different users. On average, each user provided feedback for 12.13 images, with a minimum of 1 image and a maximum of 28 images. Table 11 shows for each audience category the total number of images assigned to it by the users (via an answer to Q1) as well as the number of images whose Facebook sharing settings (at the time the feedback is received) make them accessible to a larger audience than the intended one, i.e. cases posing higher privacy risks. Since, as explained above, the limitations imposed by Facebook's API do not allow us to recover the actual (image-level) privacy settings for all images, the reported numbers represent lower limits. Thus, to provide a better indication of the actual number of privacy risks per category, we also report the percentage of high-risk images among the images of which the privacy settings could be accurately recovered.

First, we note that for the majority of images (564 out of 655), users have selected one of the less sensitive audience categories. In particular, more than half of the images (for which feedback has been provided) have been tagged as "All my friends on FB" (388), followed by

"Anyone on or off FB" (113) and "Friends of Friends on FB" (63). However, there is also a significant number of images for which a more sensitive audience category has been selected (91 out of 655) and, interestingly, there are even 9 photos assigned to the "Not at all (not shared with FB)" category. Importantly, we see that there is a significant number of confirmed privacy risks (63 out of 655 in total) which is probably a large underestimation of the actual number given the high (19.9%) percentage of risks among images with recovered privacy settings. Even more importantly, we notice that images assigned to more sensitive audience categories exhibit much higher percentages of risk, suggesting that users find it more difficult to correctly adjust the privacy settings of images intended for smaller audiences, i.e. more sensitive ones.

	Audience category	# images	# risks	% with risk among images with recovered privacy settings
more sensitive	Not at all (not shared with FB)	9	1	100.0%
	Only-me (shared with FB)	13	7	100.0%
	Close friends on FB	69	22	91.6%
less sensitive	All my friends on FB	388	21	10.3%
	Friends of Friends on FB	63	12	57.1%
	Anyone on or off FB	113	0	0.0%
	Total	655	63	19.9%

Table 11: Number of images and privacy risks per audience category.

In addition to the conclusions drawn from user responses to Q1, interesting conclusions can be drawn by analyzing user responses to Q2. Figure 16 shows how users' responses to Q2 are distributed to privacy dimensions, separately for images assigned to the three more sensitive audience categories and for images assigned to the three less sensitive categories. By comparing the two distributions, we see that images which are considered as more sensitive by users, are usually those revealing information related to their relationships (21%), location (19%), health (15%) and hobbies (15%). Images that are considered as less sensitive, on the other hand, are less frequently associated with information related to users' relationships (11%) and health (6%), and more frequently associated with information related to their hobbies (27%). Moreover, by examining the responses (31 in total) of users who specified additional kinds of personal information (by filling out the "other" input field in the pop-up box), we find that about half of them (15) concern the aspect of "leisure/vacations" but there are also aspects such as "family/friends/children" (8 responses), "artistic preferences" (2 responses), "education" (1 response) and "appearance" (1 response).

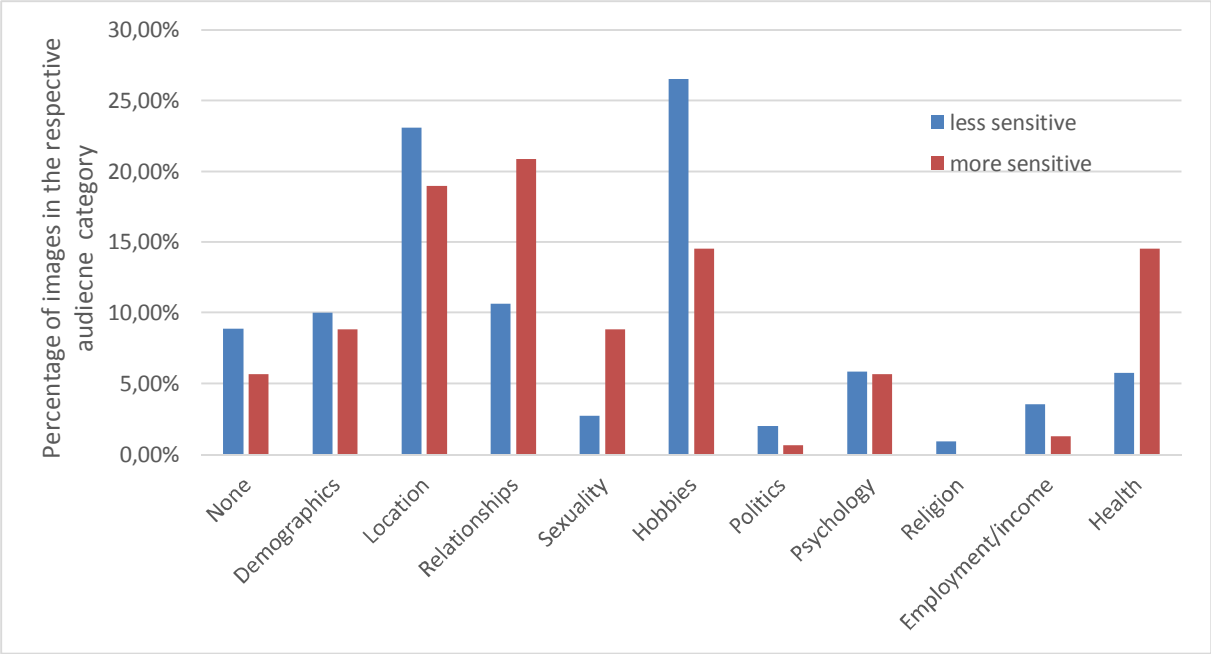

Figure 16: Distribution of users' responses to Q2 to privacy dimensions.

Besides allowing the extraction of interesting conclusions regarding users' image privacy concerns, the feedback provided by the users can serve a number of additional goals, as discussed in Subsection 4.4. Here, we focus on one of these goals, i.e. we use the feedback in order to assess the accuracy of the predictions made by the generic image privacy model that we deployed in Databait during the pilots. In order to do that, we treat users' responses on Q1 as ground truth privacy labels, after mapping the three less sensitive audience categories to the public privacy class and the three more sensitive categories to the private class. This mapping was necessary because the deployed model was trained on images labeled as either public or private and could therefore make only binary predictions. Note, that the ground truth labels generated after this mapping are fully compatible with those used to train the model since YourAlert images were annotated based on the following definitions of public and private images:

- public: "images you would share with all your OSN friends or even make public"
- private: "images you would share only with close OSN friends or not at all"

Having made the ground truth labels compatible with the model's outputs, it is then straightforward to proceed with the model's evaluation. So far (see Subsection 4.3), classification accuracy has been measured in terms of AUC, a measure that operates directly on the model's probability outputs, assessing the model's ability to assign higher probabilities on private compared to public images. When all 655 images are considered, the model's AUC is 0.71, quite close to the 0.74 AUC measured on YourAlert. When a separate AUC is calculated for each user based on his/her images, we obtain the scores of Figure 17[13]. In accordance with the results of subsection 4.3, we observe a large variability of performance between users (AUC ranges from 0 to 1) as a result of using a generic model.

[13] Note that we show results only for 10 of the 55 users that provided feedback because the rest of the users provided feedback only for images belonging to one of the two privacy classes (after the transformation) rendering AUC undefined.

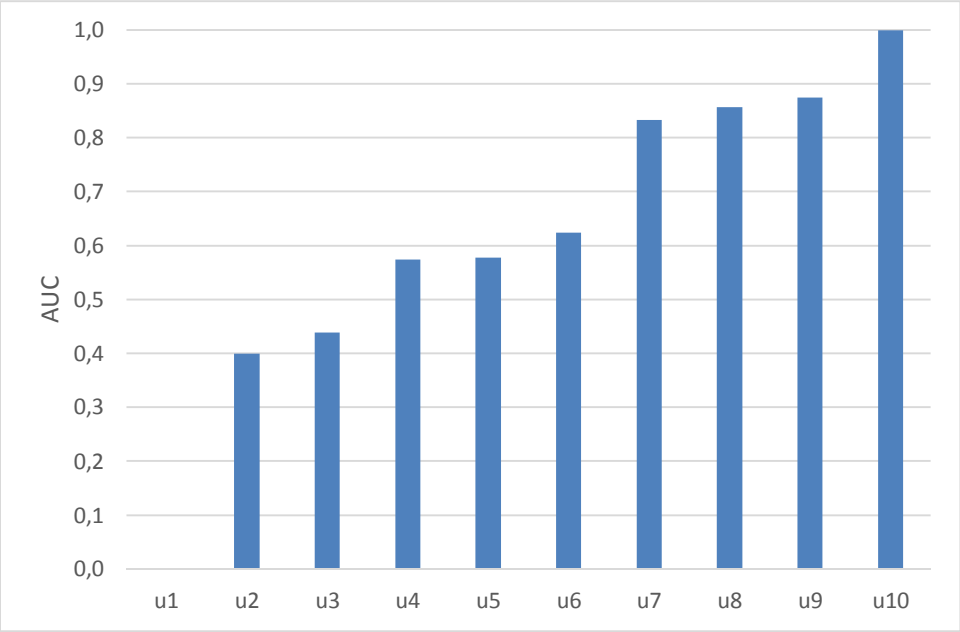

Figure 17: Model's performance (AUC) on each user.

Since the deployment of the model on Databait required hard public/private classifications rather than probability scores, complementarily to the AUC-based evaluation we also evaluated the model's hard classifications, obtained by applying a 0.4 decision threshold. This was preferred over the default 0.5 threshold in order to increase recall on the private class. The obtained confusion matrix is shown in Table 12. We see that the model classified correctly 69.9% of all users' images, managing to obtain a recall of 59% on the private class. If the default 0.5 decision threshold was applied instead, we would end up with the confusion matrix of Table 13, i.e. 82.7% correctly classified images but a much lower 18.4% recall on the private class.

		predicted	
		public	private
actual	public	*404*	159
	private	38	*54*

Table 12: Confusion matrix obtained using a 0.4 decision threshold.

		Predicted	
		public	private
actual	public	*525*	38
	private	75	*17*

Table 13: Confusion matrix obtained using a 0.5 decision threshold.

Overall, we can conclude that the deployed generic image privacy model could provide quite accurate and useful privacy predictions. Nevertheless, based on the findings of (Spyromitros-Xioufis et al., 2016) we expect that even more useful predictions can be obtained by using user feedback in order to build partially or fully-personalized models (depending on the amount of feedback each user provides).

5. Conclusions

During the second iteration of the project, work on developing multimedia mining modules was conducted in three main directions: improving visual concept detection, enriching private/non-private image classification and proposing a new visual-textual joint representation. After identifying the shortcomings of the first version of the concept detection module, we detailed a process of improving both the interpretability of identified concepts and the effectiveness of the semantic descriptor built upon the individual detectors by incorporating external knowledge. An important challenge that we identified is to offer not only correct detections but also to present the concepts using general terms, so that it is easier for a user to interpret the detection. In order to achieve this, we took into account the relations between concepts using human existing knowledge, as expressed in semantic hierarchies. Besides improving the set of concept detections, we also showed that the newly obtained semantic descriptors constitute a powerful image representation in classification tasks. This was confirmed by evaluating the descriptors on several known datasets.

We then investigated means of jointly processing visual and textual data linked to the same multimedia item. We introduced a new representation method for projecting the two modalities on a common space. It aims to reduce the gap between the projections of visual and textual features by embedding them in a local context reflecting the data distribution in the common space. The effectiveness of the proposed representation was confirmed by the strong performances obtained on bi-modal and cross-modal retrieval tasks.

Finally, we continued the line of work that concerns private/non-private image classification. Previous positive results were confirmed on an extended version of *YourAlert,* the dataset that we created for the purposes of this study, by re-evaluating generic and personalized privacy classification models. Of particular importance was the integration of a pilot version of the privacy-aware image classification module into DataBait, and its evaluation in the context of the final pilot studies. We also focused on providing easily comprehensible explanations of the classification outputs. In order to achieve this, we used a new semantic feature representation (*semfeat-lda*) that is based on a new privacy aspect modeling approach. The same approach aided us to get relevant visualizations of different groups of users with similar privacy concerns.

6. References

U. Ahsan and I. Essa. (2014). Clustering social event images using kernel canonical correlation analysis. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pat-tern Recognition Workshops, CVPRW '14, pages 814–819, Washington, DC, USA, 2014. IEEE Computer Society.

M. A. Alvarez, L. Rosasco, and N. D. Lawrence. (2012). Kernels for vector-valued functions: A review. Foundations and Trends in Machine Learning, 4(3):195{266, 2012.

A. Bergamo and L. Torresani. (2012). Meta-class features for large-scale object categorization on a budget. In Computer Vision and Pattern Recognition.

D. M. Blei, A. Y. Ng, and M. I. Jordan. (2003). Latent dirichlet allocation. JMLR, 3:993-1022.

D. Buschek, M. Bader, E. von Zezschwitz, and A. D. Luca. (2015). Automatic privacy classification of personal photos. In Human-Computer Interaction - INTERACT, 2015.

R. Caruana. (1997). Multitask learning. Machine Learning, 28(1):41-75, 1997.

X. Chen and C. Lawrence Zitnick. (2015). Mind's eye: A recurrent visual representation for image caption generation. In CVPR, June 2015.

T.S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. (2009). Nus-wide: A real-world web image database from national university of Singapore. In Proceedings of ACM Conference on Image and Video Retrieval, CIVR, 2009.

J. Costa Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. Lanckriet, R. Levy, and N. Vasconcelos. (2014). On the role of correlation and abstraction in cross-modal multimedia re-trieval. TPAMI, 36(3):521–535, 2014.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In CVPR , 2009

C. Elkan. (2001). The foundations of cost-sensitive learning. In IJCAI, 2001.

M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. (2010). The pascal visual object classes challenge. International journal of computer vision (IJCV), 88(2):303{338, 2010.

M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. (2012). The pascal visual object classes challenge 2012.

Y. Freund and R. E. Schapire. (1996). Experiments with a new boosting algorithm. In ICML, 1996.

A. L. Ginsca, A. Popescu, H. Le Borgne, N. Ballas, P. Vo, and I. Kanellos. (2015) Large-scale image mining with flickr groups. In Multimedia Modelling, MM, 2015.

Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. (2014). A multi-view em-bedding space for modeling internet images, tags, and their semantics. IJCV, 106(2):210–233, Jan. 2014.

D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-Taylor. (2004). Canonical correlation analysis: An overview with application to learning methods. Neural Computing, 16(12):2639–2664, 2004.

S. J. Hwang and K. Grauman. (2012). Learning the relative importance of objects from tagged images for retrieval and cross-modal search. IJCV, 100(2):134–153, Nov. 2012.

M. Jain, J. C. van Gemert, T. Mensink, and C. G. M. Snoek. (2015). Objects2action: Classifying and localizing actions without any video example. In International Conference on Computer Vision, ICCV, 2015.

L. Jia Li, H. Su, L. Fei-fei, and E. P. Xing. (2010). Object bank: A high-level image representation for scene classification & semantic feature sparsification. In NIPS 2010.

Y. Jia. (2013). Caffe: An open source convolutional architecture for fast feature embedding.

P. Jolicoeur, M. A. Gluck, and S. M. Kosslyn (1984). Picturesand names: Making the connection. Cognitive Psychology, 16(2):243-275, 1984

A. Karpathy and L. Fei-Fei. (2015). Deep visual-semantic align-ments for generating image descriptions. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

Y. Liu, P. K. Gummadi, B. Krishnamurthy, A. Mislove. (2011). Analyzing facebook privacy settings: user expectations vs. reality. In Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference, 2011.

M. Madejski, M. L. Johnson, S. M. Bellovin. (2012). A study of privacy settings errors in an online social network. In Tenth Annual IEEE International Conference on Pervasive Computing and Communications, 2012.

A. K. McCallum. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. (2013). Distributed representations of words and phrases and their compositionality. CoRR, abs/1310.4546,

K. D. Naini, I. S. Altingovde, R. Kawase, E. Herder, C. Niederee. Analyzing and predicting privacy settings in the social web. In User Modeling, Adaptation and Personalization, 2015.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. (2011). Multimodal deep learning. In Proceedings of the 28th inter-national conference on machine learning (ICML-11), pages 689–696, 2011.

L. Y. Pratt. (1992). Discriminability-based transfer between neural networks. In NIPS, 1992.

C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. (2010). Collecting image annotations using amazon's mechanical turk. In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, CSLDAMT '10, pages 139–147, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. (2015) ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211-252, 2015.

K. Simonyan and A. Zisserman. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014

E. Spyromitros-Xioufis, S. Papadopoulos, A. Popescu, Y. Kompatsiaris. (2016). Personalized Privacy-aware Image Classification. Proc. International Conference on Multimedia Retrieval (ICMR), New York, USA, June 6-9, 2016.

A. C. Squicciarini, C. Caragea, & R. Balakavi. (2014). Analyzing images' privacy for the modern web. In Proceedings of the 25th ACM conference on Hypertext and social media (pp. 136-147). ACM.

N. Srivastava and R. R. Salakhutdinov. (2012). Multimodal learning with deep boltzmann machines. (2012). In Advances in neural information processing systems, pages 2222–2230.

A. Tonge and C. Caragea. (2016). Image privacy prediction using deep features. In AAAI Conference on Artificial Intelligence, 2016.

L. Torresani, M. Szummer, and A. Fitzgibbon. (2010). Efficient object category recognition using classemes. In European Conference on Computer Vision, ECCV, 2010.

P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. TACL, 2:67–78, 2014.

S. Zerr, S. Siersdorfer, J. Hare, & E. Demidova. (2012). I Know What You Did Last Summer!: Privacy-Aware Image Classification and Search. SIGIR.