Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Time-Sensitive Egocentric Image Retrieval for Finding Objects in Lifelogs

Degree's Thesis

Sciences and Telecommunication Technologies Engineering

Author: Cristian Reyes Rodríguez

Advisors: Xavier Giró-i-Nieto, Eva Mohedano and Kevin McGuinness

Universitat Politècnica de Catalunya
Dublin City University
2015 - 2016

Abstract

This work explores diverse practices for conducting an object search from large amounts of egocentric images taking into account their temporal information. The application of this technology is to identify where personal belongings were lost or forgotten.

We develop a pipeline-structured system. Firstly, the images of the day being scanned are sorted based on their probability to depict the forgotten object. This stage is solved by applying an existing visual search engine based on deep learning features. Secondly, a learned threshold selects the top ranked images as candidates to contain the object. Finally the images are reranked based on temporal and diversity criteria.

Furthermore, we build a validation environment for assessing the system's performance aiming to find the optimal configuration of its parameters. Due to the lack of related works to be compared with, this thesis proposes an novel evaluation framework and metric to assess the problem.

Resum

Aquest treball explora diverses pràctiques per realitzar cerca d'objectes en grans volums d'imatges egocèntriques considerant, a més, la informació temporal d'aquestes amb l'objectiu d'identificar on s'han deixat, perdut o oblidat els objectes personals.

Desenvolupem un sistema amb estructura seqüencial d'etapes. En primer lloc, es duu a terme una cerca de les imatges que tenen més probabilitat de descriure l'objecte. Aquesta etapa es realitza aplicant motors de cerca visual ja existents basats en *deep learning*. En segon lloc, un llindar après escull les millors imatges com a candidates a contenir l'objecte. Finalment, les imatges són reordenades temporalment aplicant criteris de diversitat.

A més, construïm un entorn de validació del funcionament del sistema amb l'objectiu de trobar la configuració òptima dels seus paràmetres. Donat que no hi ha treballs similars amb els què ens poguem comparar, el treball defineix un entorn i una mètrica per a l'avaluació del problema.

Resumen

Este trabajo explora diversas prácticas para realizar búsqueda de objetos en grandes volumenes de imágenes egocéntricas considerando, además, la información temporal de estas con el objetivo de identificar el lugar donde se han dejado, perdido o olvidado objetos personales.

Desarrollamos un sistema con estructura secuencial de etapas. En primer lugar, se lleva a cabo una búsqueda de las imagenes con más probabilidad de describir el objeto. Esta etapa se realiza aplicando motores de búsqueda visual ya existentes basados en *deep learning*. En segundo lugar, un umbral aprendido escoge las mejores imágenes como candidatas a contener el objeto. Finalmente, las imágenes son reordenadas temporalmente aplicando criterios de diversidad.

Además, construimos un entorno de validación del funcionamiento del sistema con el objetivo de encontrar la configuración óptima de sus parámetros. Dado que no hay trabajos similares con los que nos podamos comparar, el trabajo define un entorno y una métrica para la evaluación del problema.

Acknowledgements

First of all, I want to express my grattitude to my advisors Eva Mohedano, Xavier Giro-i-Nieto and Kevin McGuinness for making possible to work on this project, guiding and teaching me and helping me during this period.

I would also like to refer to Erasmus+ Programme for Sutdent Mobility in the European Union, Generalitat de Catalunya and Centre de Formació Interdisciplinària Superior (CFIS) for funding my stay in the Dublin City University.

Also acknowledge Marc Carné, Mònica Chertó, Andrea Calafell, Albert Gil, Noel E. O'Connor, Cathal Gurrin and many others for being open to help me every time I needed.

Finally, I would like to thank my family and my friends, specially those who have been by my side and have given me inestimable support making so extraordinary these years at university.

Revision history and approval record

Revision	Date	Purpose
0	06/06/2016	Document creation
1	14/06/2016	Document revision
2	24/06/2016	Document revision
3	26/06/2016	Document approval

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Cristian Reyes	cristian.reyes@estudiant.upc.edu
Xavier Giró i Nieto	xavier.giro@upc.edu
Eva Mohedano	eva.mohedano@insight-centre.org
Kevin McGuinness	kevin.mcguinness@dcu.ie

Written by:		Reviewed and approved by:		Reviewed and approved by:	
Date	06/06/2016	**Date**	20/06/2016	**Date**	20/06/2016
Name	Cristian Reyes	**Name**	Xavier Giró	**Name**	Eva Mohedano
Position	Project Author	**Position**	Supervisor	**Position**	Supervisor

Contents

List of Figures

List of Tables

1 Introduction

1.1 Statement of purpose

The interest of users in having their lives digitally recorded has grown in the last years thanks to the advances on wearable sensors. Wearable cameras are among the most informative ones. Egocentric (First-Person) vision provides a unique perspective of the visual world that is inherently human-centric. Since egocentric cameras are mounted on the user (typically on the user's head but also on the chest), they are ideal to gather visual information from our everyday interactions. In the near future, everything we do might be captured into our *lifelog*, processed to extract meaning and used to support us in our daily life. Some of these devices can generate very large volumes of images daily. This fact makes it necessary to develop automatic, efficient and scalable analysis techniques in order to build useful applications upon them.

Thanks to the first person point of view, there have been recent advances in areas such as personalized video summarization, understanding concepts of social saliency, activity analysis with inside-out cameras (a camera to capture eye gaze and an outward-looking camera) or recognizing human interactions and modeling focus of attention. However, in many ways we are, as a community, only beginning to understand the full potential and limitations of the first person model.

People interact several times with their personal belongings along the day and, unfortunately, sometimes they lose or simply forget them somewhere unintentionally. Once an object has been lost, people may think: *"There should exist some kind of system to help me find my object right now!"* The usage of wearable cameras can help the user to answer this question. It must be noted that is not a simple task to find out what is a useful image to help the user in this situation. The user can go through hundreds of images recorded along the day trying to find the one that contains his or her object.

Information retrieval is the activity of obtaining information relevant to a given search from a collection of information resources. In this work, we assess the potential of egocentric vision to help the user answer the question *Where did I put my ...?*. We address it as a **time-sensitive image retrieval** problem. This thesis explores the design of a retrieval system for this purpose, focusing on the visual as well as on the temporal information.

In particular, the main contributions of this project are:

- Adapt a previous work on instance search [16] to the specific field of Egocentric Vision.
- Explore strategies to enhance the last appearances of the objects versus the previous ones in order to take advantage of the temporal information.
- Determine an appropriate metric to assess the performance of the system and its capability to solve the main question.
- Provide an annotated ground truth for the NTICR-Lifelog dataset [3] as well as a baseline performance to the scientific community to allow future research on this field.

- Explore the usefulness of saliency maps [17] to improve the performance of the retrieval part based on the visual similarity.

During the development of this thesis, an extended abstract has been submitted and accepted at the 4th Workshop on Egocentric (First-Person) Vision of the Conference on Computer Vision and Pattern Recognition (CVPR) 2016. We also plan to submit the improvements achieved since then in the next edition of the Lifelogging Tools and Applications 2016 Workshop at the Association for Computing Machinery conference on Multimedia (ACM-MM) 2016.

1.2 Requirements and specifications

This project has been developed as a tool that could be used for other students or developers in the future to be recycled or improved.

The requirements of this project are the following:

- Design a system capable of helping the users to find their personal objects once they have been forgotten or lost.
- Adapt previous work on instance search to the egocentric vision field.
- Find a way to exploit the temporal information in order to improve the system's performance.
- Evaluate the results and set a baseline for further research on this problem.
- Contribute to the scientific dissemination of the work.

The specifications are the following

- Develop algorithms in Python.
- Use GPUs to perform the high demanding computation experiments.
- Use the software platform *Caffe* [8] as the basic deep learning framework.

1.3 Methods and procedures

Our goal is to rank the egocentric images captured during a day based on their likelihood to depict the location of a personal object. To do that, we design a system based on a pipeline structure which is composed of the following stages: ranking by visual similarity, partition between candidate/non-candidate images and temporal-aware reranking within each class.

The first stage is based only on the visual information. It builds a ranking of the images of the day based in their likelihood to contain the object in them. Then, a classification between candidate and non-candidate images is performed in order to bound the set of images in what to focus on. Finally, a temporal block takes care of the temporal information enhancing those images that are closer in time to the instant that search is thrown.

1.4 Work Plan

This project was developed at the Insight Center for Data Analytics at the Dublin City University. This mobility was founded by the Erasmus+ Programme of the European Union.

It was advised in Dublin by Phd candidate Eva Mohedano and post-doctoral researcher Kevin McGuinness. The work was also closely advised by Professor Xavier Giro-i-Nieto from the Universitat Politècnica de Catalunya through weekly videocalls.

The established work plan has been followed, with a few exceptions and modifications explained in the section 1.5.

1.4.1 Work Packages

- WP 1: Written documentation
- WP 2: State of the art
- WP 3: Analysis and development of software
- WP 4: Analysis of datasets
- WP 5: Experimental part
- WP 6: Oral defense

1.4.2 Gantt Diagram

The following figure shows a summary of the thesis plan. We provide a detailed version of it in section 7, figure 7.1

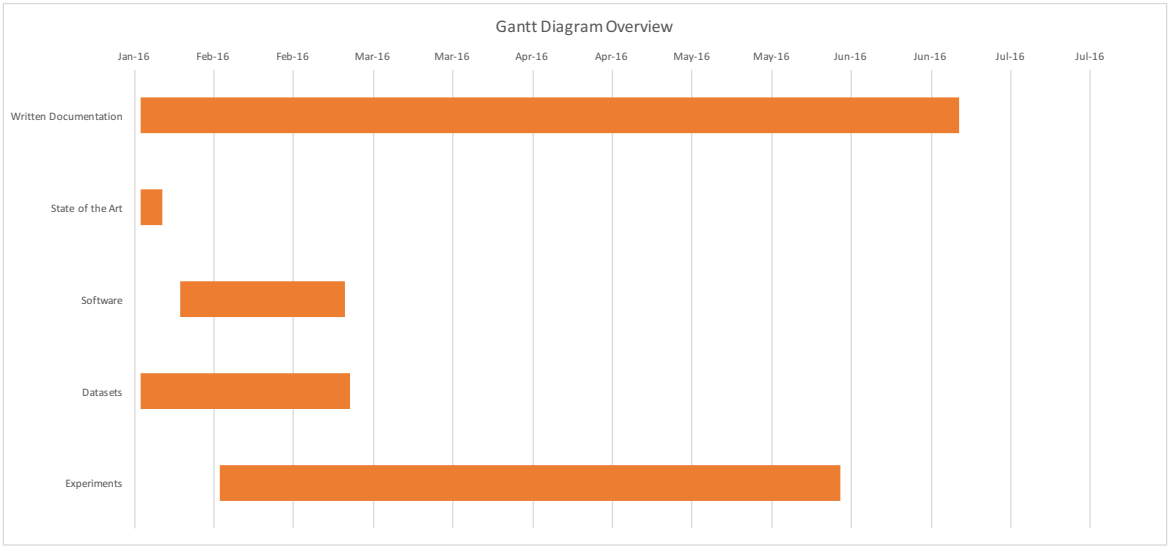


Figure 1.1: Gantt Diagram of the Degree's Thesis

1.5 Incidents and Modification

Due to the huge annotation effort required for the manual annotation of the whole dataset, it was restricted to a part of it.

During the project we decided to increase the workload of the tasks in the WP 5 by exploiting the usage of visual saliency maps in the system.

Some experimentation was limited by the task of finding the optimal thresholds 4.3. It is a high computation consuming task and in some stages it was necessary to work with non-optimal thresholds in order to predict if some changes might improve the system before spending time training.

2 State of the art

2.1 Lifelogging applications

The early applications of lifelogging were mostly related to healthcare, and generally rely on a single or reduced set of sensors (*e.g.* accelerometer only, wearable camera only, etc.). The potential for lifelogging is greater than these initial use cases. Bell and Gemmell (2009) in their book "Total Recall" argue why lifelogging will revolutionize health care, productivity, learning and social society [7].

While the potential applications of lifelogging are not yet well understood, it is possible that app developers will come up with ingenious applications and tools that fulfill some of the lifelogging visions [1]. Search and information retrieval are the fundamental tools for many kinds of lifelogging applications. Reciprocally, lifelogging provides new challenges for information retrieval and user experience modeling.

The concept of lifelog implicitly refers to data, which can be of many types. The recording of statistical data, for example values related to physical activity, is a highly widespread practice nowadays which uses motion sensors. However, there are also other sources of data to create lifelogs such as wearable cameras. These devices allow to use visual lifelogs in plenty of situations.

Lifelogs in memory rehabilitation and memory assistance have become an active area of research aiming, for instance, to overcome the difficulties some people have with short term memory recall [23], specially for people with Alzheimer's and other dementias.

Visual lifelogs are also exploited by object recognition which brings several applications. Human-object interactions have been recognized by combining object recognition, motion estimation and semantic information [20] or using likelihood models based on hand trajectories [6]. Object recognition is not only useful for object-based detections but also for event identification using the object categories that appear in an image [13] or activity recognition based on the objects' frequency of use. This work [25] recognized handled objects and associated kitchen tasks from a fixed wall-mounted camera.

Lifelogs are not necessary linked to an individual usage. There are also population-based lifelogging applications where information recorded by many individuals is gathered in order to take conclusions of a group's behavior. A good example of this is described in [10], where healthcare workers in a clinical practice would typically log their work at the end of their shift but in this case they used visual lifelogs to trigger their own recall of their day. In particular this was used in an analyses to better understand the information needs of clinicians in hospitals in Finland.

Even though the appearance of lifelogging devices may be found as an aid for many applications, users must also keep in mind that there exists also an specific legislation related to data protection in many countries that can oppose limits to their use [15, 24].

2.2 Visual Search

The use of wearable cameras to create lifelogs requires automatic engines to make profitable applications upon them. Content-Based Image Retrieval (CBIR) has been an active area of research since the past decade. A lot of work is still being developed in this area, which includes various applications such as security, medical imaging, audio and video retrieval. It shows the growing interest of many researchers in this field, which results in the development of new tools and techniques.

Image retrieval based only on the visual content requires a representation in order to be searched and classified. Visual descriptors, or image descriptors, are representations of the contents in images or videos which aim to express patterns in pixels contained in a digital image. They describe elementary characteristics such as the shape, the color, the texture or the motion and they allow searching contents in a fast and efficient way. The description of the audiovisual content is not a trivial task and it is essential for the efficient usage of these sorts of files. A good visual descriptor should have the ability to manipulate intensity, rotation, scale and affine transformations.

Nowadays, Convolutional Neural Networks (CNN) are considered as the dominant approach to extracting visual descriptors for many computer vision tasks since the pioneer work carried out by **Krizhevsky et al.** [9] in 2012. The layers of a CNN have neurons arranged in 3 dimensions: width, height and depth. The neurons inside a layer are only connected to a small region of the layer before it, called a receptive field. In our work we use a pre-trained Convolutional Neural Network called *vgg16* [22], using the *conv-5_1* layer to extract the features of the images.

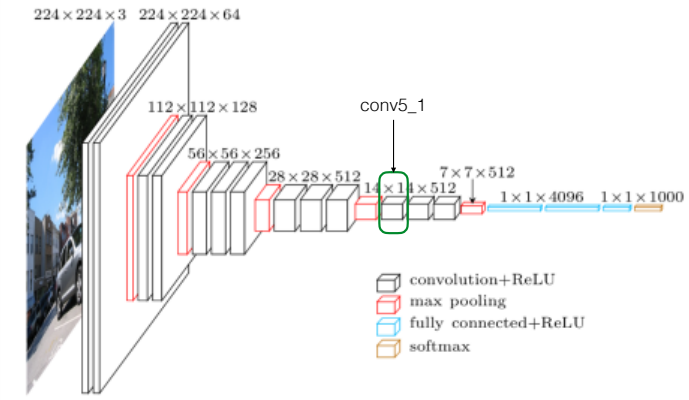


Figure 2.1: Structure of the *vgg16* CNN. [21]

Once the visual descriptors have been obtained, a similarity metric to compare the descriptors between images is needed in order to perform image retrieval. The Bag of Words model (BoW) is a popular representation used in natural language processing and information retrieval. In this

model, a text is represented as the multiset or bag [1] of its words, disregarding grammar and even word order but keeping multiplicity. Furthermore, the BoW model has also been used for computer vision because it generates scalable and sparse representations that can be stored in inverted file structures [26]. These structures allow to do computationally fast dot products of sparse vectors.

Image representation based on the BoW model has its basis in the fact that an image can be treated as a document. Text words are something absolutely clear whereas "visual words" in images need to be defined. To achieve this, a classic pipeline includes following three steps: feature detection, feature description, and codebook generation [12]. A definition of the BoW model can be the "*histogram representation based on independent features*" [5]. Content Based Image Retrieval (CBIR) appears to be the early adopter of this image representation technique [18]. We considered to use a previous work [16] based on this framework as a tool for performing the visual search in our system.

The final step for the BoW model is to convert vector-represented patches to *codewords* (analogous to words in text documents), which also produces a *codebook* (analogy to a word dictionary). A codeword can be considered as a representative of several similar patches. One simple method is performing k-means clustering over all the vectors of a training set [11] or, in case of high demanding problems, do it with a random subset of the vectors for training. Codewords are then defined as the centers of the learned clusters. The number of the clusters is the codebook size (analogous to the size of the word dictionary), this number is usually large 25,000 in [16], as an example. Then, each patch in an image is assigned to a certain codeword through the clustering process creating and *assignment map* and the image can be represented by the histogram of the codewords. Finally the descriptor is the L_2-normalized histogram of the codewords.

[1] A multiset (or bag) is a generalization of the concept of a set that, unlike a set, allows multiple instances of the elements. For example, $S_1 = \{a, a, b\}$ and $S_2 = \{a, b, b\}$ are different multisets although they are the same set $S = \{a, b\}$. The multiplicity of a in S_1 is 2.

3 Methodology

Our goal is to rank the egocentric images captured during a day based on their likelihood to depict the location of a personal object. In our problem we have defined the following sets of images as inputs to the system:

- The **query set** Q: For each object or category that is going to be searched for, a set of images containing the object is necessary to define the query for the system.
- The **target set** I: For each day, this set contains around 2,000 images captured along the day.

The system has a pipeline structure which may be divided into two main stages: a visual aware block followed by a temporal aware block. The visual aware block is based on encoding the convolutional features of CNN using the BoW aggregation scheme [16]. The temporal aware block is composed by a first step that selects candidate images and a second one which takes care of enhancing the ranking based on the temporal information. We included configuration flags for each stage in order to determine the most appropriate set up. These flags are shown in figure 3.1 and are deeper described in the following sections.

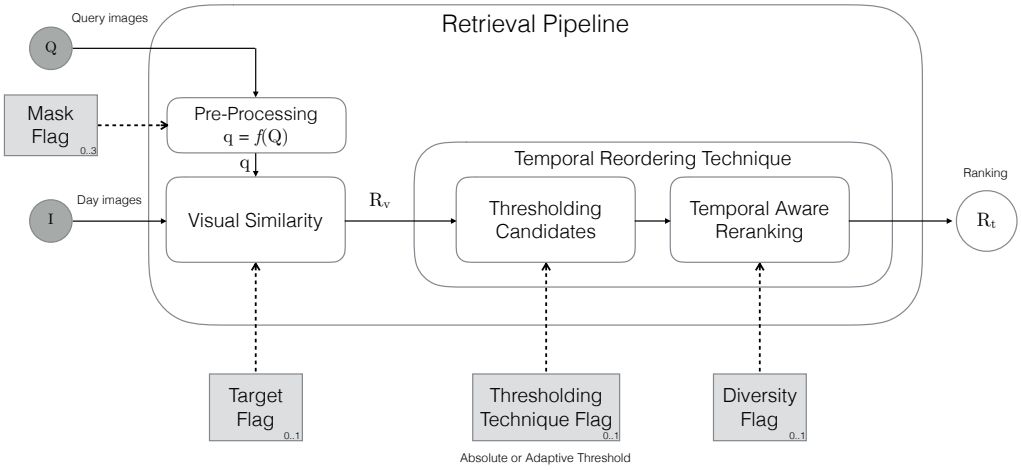


Figure 3.1: Global architecture of the pipeline based system.

3.1 Baseline

Up to the author's knowledge, there is no previous work on finding lost objects in lifelogging datasets. Thus, there is no baseline from other authors for the whole system, that is using visual and temporal information. We decided to define as a baseline the easiest approach for the resolution of the problem: a simple temporal sorting of the images based on their time stamp, being the first image of the ranking the last one taken by the camera. This would mimic the case where the user would visualize the whole sequence of images taken during the day in reverse order. This would be the most obvious action of a person who is looking for the location of a lost personal item.

Even though there are no works to be compared with in the field of lifelogging, there exist multiple works which have explored the problem of visual search from a few examples of the query object. As introduced in section 2, we have used an existing image retrieval system [16] and we applied further stages in order to improve its performance when looking for objects in egocentric images.

3.2 Ranking by visual similarity

The goal of this stage is to create a ranking R_v of the I set for a given Q set. This ranking is based only on the visual information of the images. In order to do that, we have explored different configurations and variations of the BoW model described in section 2.

A function $f : Q \longmapsto f(Q) \in \mathbb{R}^n$ aims at building a single **query vector** $\vec{q}$ by gathering the information of all images in $Q = \{q_1, q_2, ..., q_{|Q|}\}$, leading to obtain $\vec{q} = f(Q)$. This function f can be defined in several ways. To choose a specific definition of f, the system includes the `Mask Flag`.

Three different approaches have been explored to define f:

- **Full Image (FI)**: The $\vec{q}$ vector is built by averaging the frequencies of the visual words of all the local CNN features from the query images.
- **Hard Bounding Box (HBB)**: The $\vec{q}$ vector is built by averaging frequencies of the visual words that fall inside a query bounding box that surrounds the object. This approach considers only the visual words that describe the object.
- **Soft Bounding Box (SBB)**: The $\vec{q}$ vector is built by averaging frequencies of the visual words of the whole image, but weighting them depending on their distance to the bounding box. This allows introducing context in addition to the object. Weights are computed as the inverse of the distance to the closest side of the bounding box and are L_2-normalized.

A similar procedure is applied to the set of target images I, the daily images in our problem. A function $g : I \to \mathbb{R}^n$ is defined to build a feature vector $\vec{i_j} = g(i_j)$ for each image $i_j \in I$. Three different definitions of the g function have been studied which bring to the `Target Flag`:

- **Full Image (FI)**: The $\vec{i_j}$ vector is built using the visual words of all the local CNN features from the i_j image.

- **Center Bias (CB)**: The $\vec{i_j}$ vector is built using the visual words of all the local CNN features from the i_j image but it inversely weightens the features with the distance to the center of the image.

- **Saliency Mask (SM)**: The $\vec{i_j}$ vector is built using the local CNN features of the whole image, but this time weighting their frequencies with the help of a visual saliency map. The saliency of an item – be it an object, a person, a pixel, etc. – is the state or quality by which it stands out relative to its neighbors. A saliency map, in image processing, is a map depicting the areas of the image which have a high saliency.

  As it is described in section 2, the assignment map is extracted from the *conv-5_1* layer of the *vgg16* pre-trained convolutional neural network. This fact implies that the assignment map is 32×42. We assumed that each visual word was related to a set of pixels in the original image.

  Saliency maps were obtained using another pre-trained CNN: *SalNet* [17]. This network produces maps that represent the probability of visual attention on an image, defined as the eye gaze fixation points.

  According to this, we decided to down-sample the saliency maps to the same size of the assignment maps by applying the mean function to each local block. Once the down-sampling is obtained, a vector $w = (w_1, ..., w_{32\times42})$ is built and L_2-normalized. That is, $w := \frac{w}{\|w\|_{L_2}} = \frac{w}{\sqrt{\sum w_k^2}}$ where $w_k = \text{mean}(S_{i,j})$. This vector contains the weights related to each assignment of the assignment map.

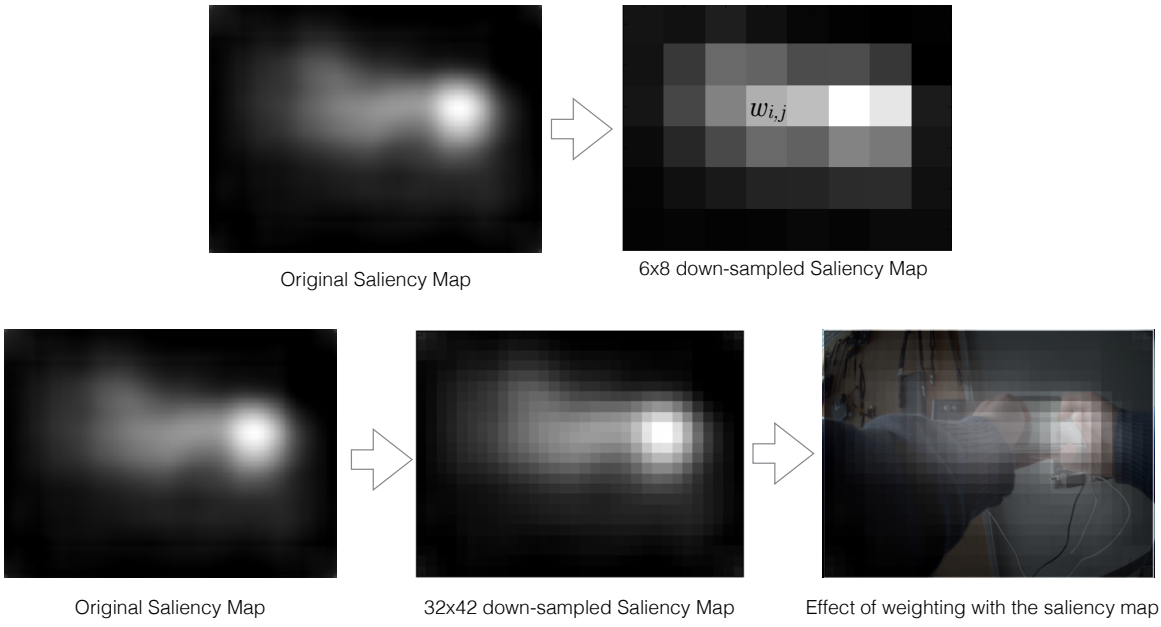


Figure 3.2: First row: Simplified example of down-sampling by mean. Second row: Obtaining a set of weights for the assignment map based on the saliency map.

This $\vec{i_j}$ feature vector is compared to the $\vec{q}$ query feature vector in order to compute the Cosine Similarity [1] between $\vec{i}$ and $\vec{q}$ and obtain the ν score. Then the visual ranking R_v is produced by ordering the images in I according to their ν score.

[1] Cosine Similarity$(a,b) = \cos(\widehat{a,b})$ Note that it is always between 0 and 1 as vectors have non-negative components.

3.3 Detection of candidate moments

The visual ranking R_v provides an ordered list of the images based on their likelihood to contain the object. It must be noticed that in our problem this might not always be useful. The last appearance of the object does not have to be the most similar to the query in visual terms. Taking this into account we introduced a post-processing to the visual search ranking.

The first step in this post-processing is determining which of the images in the ranked list should be considered as relevant to contain the query object. This is achieved by thresholding the list and considering as relevant Candidates (C) the upper part of the list, and Discarded (D) images the lower one. Two different thresholding techniques were considered in order to create the C and $D = I \setminus C$ sets.

- **Threshold on Visual Similarity Scores** (TVSS): This technique consists in building the set of the candidate images as $C = \{p \in I : \nu_p > \nu_{th}\}$, where ν_{th} is a learned threshold . It is, basically, an absolute threshold that the visual scores have to overcome to be considered as candidates.
- **Nearest Neighbor Distance Ratio** (NDRR): This strategy is inspired by a previous work by Loewe [14]. Let ν_1 and ν_2 be the two best scores in the ranked list, then the candidates set is defined as $C = \left\{i \in I : \frac{\nu_i}{\nu_1} > \rho_{th}\frac{\nu_2}{\nu_1}\right\}$. In this case, it is an adaptive technique which sets the threshold depending on the ratio of the scores of two best visually ranked images.

Both techniques require to set either ν_{th} or ρ_{th}. These values cannot be chosen arbitrarily. The appropriate way to choose them is to perform an optimization of the system performance over a training set in order to predict the suitable values, as it is described in section 4.3. We also wanted this to be independent of the categories and be able to set a unique threshold for the system.

3.4 Temporal aware reranking

Once candidate images have been selected, the next, and last, step takes care of the temporal information. The temporal-aware reranking introduces the concept that the lost object may not be in the location with the best visual match with the query, but in the last location where it was seen.

Two rankings R_C and R_D are built by reranking the elements in C and D, respectively, based on their time stamps. The final ranking R_t is built as the concatenation of $R_t = [R_C, R_D]$ (which considers the best candidate to be at the beginning of the list). Thus, R_t always contains all the images in I and we ensure that any relevant image will appear somewhere in the ranking, ever after the thresholded cases. We propose two strategies to exploit the time stamps of the images:

- **Decreasing Time-Stamp Sorting**: This is the most simple approach we can consider at this point. Just a simple reordering of the C and D sets to build the R_C and R_D rankings from the latest to the earliest time-stamp. This configuration will be applied in all experiments, unless otherwise stated.

- **Interleaving**: This other approach introduces the concept of diversity. We realized that the rankings tend to present consecutive images of the same moment when using the straightforward sorting.

  This is an expected behavior due to the high visual redundancy of neighboring images in an egocentric sequence. Therefore, we considered that exploding this fact with a diversity technique could be useful. As the final goal of this work is determining the location of the object, showing similar and consecutive images to the user is uninformative. By introducing a diversity step, we force the system to generate rank list of diverse images, which may increase the chances of determining the object location by looking at the minimum of elements in the ranked list.

  Our diversity-based technique has its basis in the interleaving of samples. In digital communication, interleaving is the reordering of data that is to be transmitted so that consecutive samples are distributed over a larger sequence of data in order to reduce the effect of burst errors. Adapting it to our domain, we interleave images from different scenes in order to put a representative of each scene at the top of the ranking. Thus, if the first candidate is not relevant, we avoid the second to be from the same scene and, therefore, it is more likely to be relevant.

  In order to do that, we used this scheme after the candidate selection stage also described in figure 3.3.

  1. Make a list with all the images in I sorting by their time-stamp in decreasing order. That is, the later image the first. For each image $i \in I$ it must be known whether it belongs to C or D. Such as, $O = \{i^C_{n-1}, ..., i^C_m, i^D_{m-1}, ..., i^D_l, i^C_{l-1}, ..., i^C_k, i^D_{k-1}, ..., i^D_1\}$.
  2. Split into sub-lists using the transitions $C \rightarrow D$ or $D \rightarrow C$ as a boundary.
  3. Build a new list R_C by adding the first image of each sub-list containing elements in C maintaining time-stamp in decreasing order. Then, the second image of each sub-list and so on. Thus, $R_C = \{i^C_{n-1}, i^C_{l-1}, i^C_{n-2}, i^C_{l-2}, ...\}$. Build R_D analogously.
  4. Concatenate R_C and R_D to obtain the final ranking $R_t = [R_C, R_D]$.

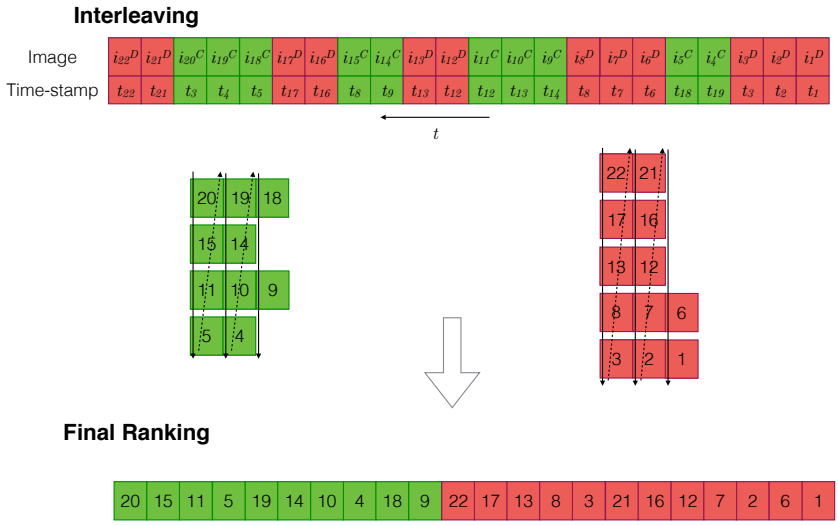


Figure 3.3: Scheme of the interleaving strategy used.

4 Results

This chapter presents the results obtained with the techniques presented in Chapter 3 to improve the baseline system described in Section 3.1.

The evaluation of an information retrieval system is the process of assessing how well does it meet the information needs of its users. In order to do that, an evaluation environment must be set up. To measure an information retrieval system effectiveness in the standard way, it is required:

1. A document collection, a dataset.
2. A collection information needs, expressible as queries.
3. A set of relevance judgments, usually a binary assessment of either *relevant* or *non-relevant* for each query-document pair.
4. A specific metric that reflects quantitatively the user's experience with the system.

To approach properly this problem, it shall be split into two sub-problems. On one hand, the visual mechanism should be assessed with a specific ground truth and metric to evaluate, only, its visual performance. After this evaluation an optimal configuration for the visual stages could be chosen. Then, the whole system should be evaluated over a new ground truth, this time including the temporal information. That would allow to study and optimize independently the visual and temporal stages.

However, it was unfeasible for us to face the problem in that way because it requires annotating around 35,000 images. Thus, we decided to only evaluate its performance at the end of the pipeline and optimize the parameters that thetermine the visual filtering based on the performance after the temporal reranking.

4.1 Datasets

An annotated dataset is needed to do the evaluation of the performance of the retrieval system. This evaluation follows these steps. The system does the search in the dataset and generates a ranking. Then the performance is evaluated taking into account the positions in the ranking where the relevant images are found.

In this case, the dataset corresponds to a set of images containing personal objects. The annotation consists of tagging each image depending on whether it is relevant or not for the user when looking for the forgotten or lost objects. These tags will compose what is called the *ground truth*.

As we wanted to use egocentric images, a study on existing egocentric datasets was done. These datasets did not contain annotations of specific personal objects that could appear in the images. This fact brought us to decide between building and annotating a suitable dataset from

scratch or, alternatively, annotating an existing dataset according to the requirements of our retrieval. We opted for the second option.

After this initial analysis, two candidate were compared to decide which one to use for the experiments. On one hand, the The Egocentric Dataset of the University of Barcelona (EDUB) [2], which is composed of 4912 images acquired by 4 people using the Narrative Clip 1 [1] wearable camera. It is divided in 8 different days, 2 days per person. On the other hand, the NII Testbeds and Community for Information access Research (NTCIR) Lifelog dataset, which is composed of 88185 images acquired by 3 people using the Autographer [2] wearable camera during 90 days, 30 days per person.

We chose the NTCIR-Lifelog dataset for two reasons. Firstly, it had 30 days of the same user versus the 2 days of the EDUB. This was really interesting when performing evaluations in order to understand a general behavior. Secondly, the Autographer camera used in the NTCIR-Lifelog dataset used a wide angle lens. This fact was really helpful to make the images more likely to include personal objects versus the EDUB images acquired with the Narrative.

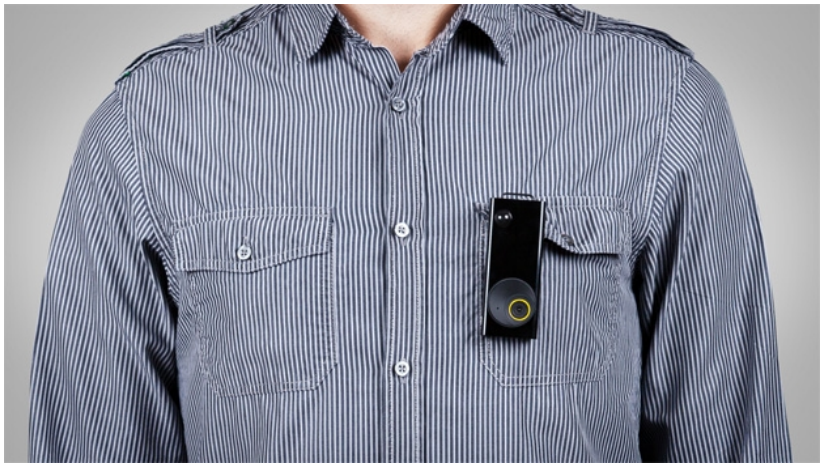


Figure 4.1: Autographer wearable digital camera developed by OMG Life.

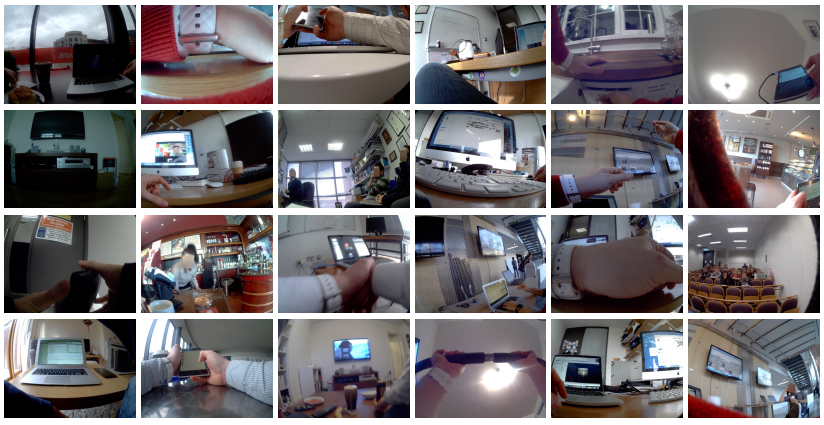


Figure 4.2: Collection of images contained in the NTCIR-Lifelog dataset.

[1] http://getnarrative.com/narrative-clip-1.
[2] http://www.autographer.com.

4.1.1 Definition of Queries

When performing a search the system needs an input of some images of the object in order to look for it. To carry out the experiments, and after doing an exhaustive analysis of the dataset, we decided to work with 4 object categories: **mobile phone**, **laptop**, **watch** and **headphones**. The set Q was built containing 5 images of the own dataset for each category. The whole object was present in these images and occupied most of it.

Figure 4.3: Set of images Q used to build the query vector $\vec{q}$ for each category.

4.1.2 Annotation of the Dataset

The annotation task consists in tagging the images that are going to be considered as relevant to the user. As the project was evolving, we modified and defined more appropriate ways of annotation. It must be noticed that only the images of one user, around 30,000, were annotated.

In a first approach, we decided to annotate the 3 last occurrences of each object. That is, the ground truth was composed by a maximum of 3 images per category containing the full object in them. After preliminary experiments we realized that this was not the best option, because it implies that the system is supposed to find all the three images, but they might depict different locations while only the last location where the object appears is of interest for our application.

The second approach, consisted in using only the last image containing each object. So, the ground truth was composed by a single image per category, and we called it One-Relevant Ground Truth. After some experimentation and a qualitative analysis of the rankings, we realized that the visual aware block was performing better that what the metric was suggesting. The fact was that some images from the same location as the one annotated as relevant were easier to find. It may be because of the object orientation or even because they were clearer than the one annotated as relevant (the last appearance in the location). This way of annotating seemed to be more accurate as well as realistic but it brought up some new dilemmas. The system was now asked to find the last image containing the object but if it found an image from the same scene as the relevant, it was considered as an error.

This fact lead to the last and final annotation strategy. We decided to extend the annotations of the dataset following this guideline: *"We will consider as relevant those images that would*

help to find out where was the last time that the camera saw the object". This strategy made us considering as relevant all the images that were from the same location and shown the object. Any of them would help the user to find his or her object. We called this the Extended Ground Truth.

4.2 Evaluation metric

In order to assess the performance of the system, an evaluation metric must be chosen to be able to compare quantitatively how do the different configurations perform. This metric has to be as realistic as possible and has to have the ability to measure exactly whether the system helps or not to the user when he or she looks for the objects.

4.2.1 Mean Average Precision

At a very early stage of the project, the Mean Average Precision (MAP) was employed to evaluate the performance because of its popularity among the image retrieval community.

The Mean Average Precision of the system is obtained by computing the mean of the Average Precisions (AP) of the Q queries, as presented in Equation 4.1.

$$\mathrm{MAP} = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{AP}(q) \tag{4.1}$$

At the same time, the average precision for a class is obtained by averaging the precision at position k ($P@k$ or $P(k)$) of all images in the ranked list which actually belong to the set R of relevant images. The $P@k$ is defined as the amount of relevant elements in the first k elements of the ranked list, divided by k. Equation 4.2 formulates this metric.

$$\mathrm{AP} = \frac{1}{|R|} \sum_{k=1}^{N} \mathrm{P}(k) \cdot \mathbb{1}_R(k) \tag{4.2}$$

where $|R|$ corresponds to the amount of images in the set R of relevant images, and $\mathbb{1}_R(k)$ is a binary function that activates when the element k in the ranked list belongs to R. This metric, considers all the relevant images in the ground truth. Given that in our problem we do not need to find all relevant documents, but only one of them, we also decided to use another.

4.2.2 Mean Reciprocal Rank

The Mean Reciprocal Rank (MRR) [4] is the average of the reciprocal ranks of results for a sample of queries Q, being the reciprocal rank of a query response the multiplicative inverse in the rank of the first relevant answer q^*. For a day d its mathematical expression is:

$$\mathrm{MRR_d} = \frac{1}{|Q_d|} \sum_{q \in Q_d} \frac{1}{q^*} \tag{4.3}$$

We have defined the Averaged-MRR ($\mathrm{A\text{-}MRR}$) to refer to the average of MRRs obtained across all days. Given a set of days $D = \{d_1, d_2, ..., d_k\}$ the expression of the Averaged Mean Reciprocal Rank is

$$\mathrm{A\text{-}MRR} = \frac{1}{|D|} \sum_{d \in D} \mathrm{MRR_d} = \frac{1}{|D|} \sum_{d \in D} \frac{1}{|Q_d|} \sum_{q \in Q_d} \frac{1}{q^*} \tag{4.4}$$

Mean Reciprocal Rank is associated with a user model where the user only wishes to see one relevant document. Assuming that the user will look down the ranking until a relevant document is found, and that document is at rank n, then the precision of the set they view is $\frac{1}{n}$, which is also the reciprocal rank measure.

4.3 Training

As it has been discussed in the previous sections 3.3, in order to apply some strategies it is necessary to use certain values that can not be chosen arbitrarily.

In many areas of information science, finding predictive relationships from data is a very important task. Initial discovery of relationships is usually done with a training set while a test set is used for evaluating whether the discovered relationships hold.

More formally, a **training set** is a set of data used to discover potentially predictive relationships. A **test set** is a set of data used to assess the strength and utility of a predictive relationship. Thus, we decided to divide our dataset composed by 24 days [3] into this two subsets using 9 days for the training and the remaining 15 for the test.

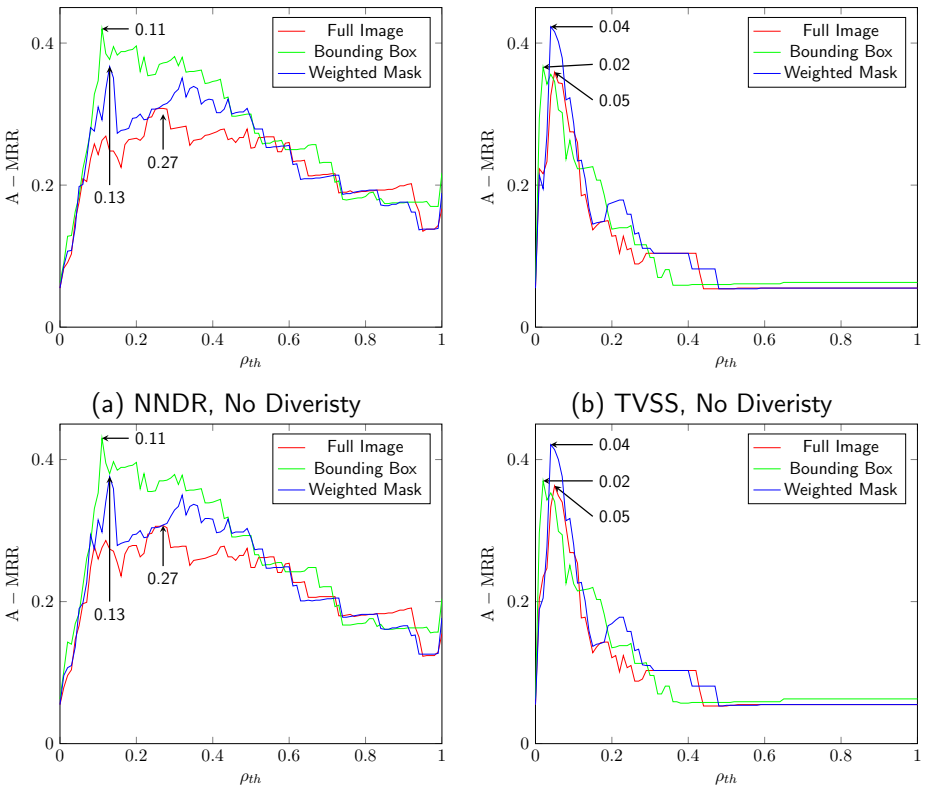
The values that we wanted to train were basically the construction of the codebook for the visual words as well as the thresholds used in both techniques described in 3.3, TVSS and NNDR.

- **Visual Words Codebook**: To apply the BoW framework it is necessary to construct a visual codebook in order to map vectors to their nearest centroid. This codebook was built using k-means clustering method [4], actually we used an algorithm performing an acceleration technique based on approximate nearest neighbors to deal with the high dimensional codebooks, on local CNN to fit the codebook generated by 25,000 centroids as in [16].
- **Threshold values ν_{th} and ρ_{th}**: In order to predict what would be the best value for these parameters, the same procedure was applied for both. We performed a sweep from 0 to 1 with a step-size of 0.01. For each of these values the MRR was computed and averaged across the 9 days that composed the training set. Therefore, this $\mathrm{A\text{-}MRR}$ can

[3] We had 30 days recorded by the user but 3 of them were used to build the Q set and the other 3 did not contain useful images for our problem.

[4] k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

be understood as a function of the threshold, so an optimal argument can be chosen. Figure 4.4 show the curves obtained and the optimal values chosen when training with saliency maps for the g function. When training using other configurations for the g function, the optimal thresholds ρ_{th} and ν_{th} remained in the same values despite $\mathrm{A\text{-}MRR}$ was slightly different.



(a) NNDR, No Diveristy

(b) TVSS, No Diveristy

(c) NNDR, Diveristy

(d) TVSS, No Diveristy

Figure 4.4: Training the thresholds using SM for g.

4.4 Test

Once the codebook and the threshold values ν_{th} and ρ_{th} are learned, everything is ready to assess the performance of the system over the test set.

The following tables show the results obtained when computing the $\mathrm{A\text{-}MRR}$ over the 15 days that define the test set and taking into account all the different configurations that can be obtained according to the configuration parameters.

4.4.1 Numerical Results

As it is described in section 3, there are several different configurations for the system. We aim to determine which of the possible approaches is the best for each flag in the system summarized in figure 4.1.

Flag	Possible Approaches
Query mask flag $f(Q)$	Full Image (FI) Hard Bounding Box (HBB) Soft Bounding Box (SBB)
Target processing flag $g(i)$	Full Image (FI) Center Bias (CB) Saliency Maps (SM)
Thresholding flag	Nearest Neighbor Distance Ratio (NNDR) Threshold on Visual Similarity Scores (TVSS)
Temporal reordering flag	Time-stamp Sorting Interleaving

Table 4.1: Configuration parameters summary.

Each of the following tables contain the A - MRR values obtained for all possible configurations given a determined approach of g. Being the Time Sorting the set baseline, and the Visual Ranking the performance at a medium stage to be able to understand the power of the temporal enhancing stages.

$f(Q)$	Time Sorting	Visual Ranking	NNDR	TVSS	NNDR+I [5]	TVSS+I
FI		0,157	0,216	0,213	0,231	0,223
HBB	0,051	0,139	0,212	0,180	0,216	0,184
SBB		0,163	0,171	0,257	0,169	**0,269**

Table 4.2: A - MRR using Full Image for g.

$f(Q)$	Time Sorting	Visual Ranking	NNDR	TVSS	NNDR+I	TVSS+I
FI		0,156	0,191	0,205	0,206	0,215
HBB	0,051	0,130	0,212	0,170	0,216	0,174
SBB		0,162	0,160	0,240	0,161	**0,258**

Table 4.3: A - MRR using Center Bias for g.

$f(Q)$	Time Sorting	Visual Ranking	NNDR	TVSS	NNDR+I	TVSS+I
FI		0,150	0,240	0,274	0,249	**0,283**
HBB	0,051	0,173	0,200	0,136	0,206	0,147
SBB		0,178	0,168	0,242	0,174	0,257

Table 4.4: A - MRR using Saliency Maps for g.

[5] I stands for Interleaving

4.4.2 Discussion

The previous tables allow to interpret the following facts:

- Comparing the results obtained with any of the system configurations (the four last columns) versus the intermediate stage (the visual ranking R_v) and the defined baseline (the temporal sorting), we can conclude that both the visual and the temporal stages improve the $\mathrm{A\text{-}MRR}$. In other words, the system would be more helpful for the users when they try to solve the task of looking for their objects than if they go backwards through the egocentric images taken by their devices.
- Comparing the same approach with or without temporal diversity reveals that diversity improves the $\mathrm{A\text{-}MRR}$. This fact leads to assume that it is always a good choice to use the interleaving technique to implement this temporal diversity although further approaches might work even better.
- Comparing the results of table 4.2 and 4.3, it is shown that building the g function using the Center Bias does not improve the results obtained when using the Full Image, despite this is a good strategy to use in many other tasks related to CBIR. This is probably due to the fact that the objects in egocentric images do not have to be located in the center of the image as it is taken unintentionally.
- The usage of Saliency Maps for the g function decreases the $\mathrm{A\text{-}MRR}$ when combined with the HBB and it remains practically at the same value for SBB for the f function. Nevertheless, it outperforms any other approaches when combined with the FI configuration for f. Suggesting that the local convolutional features of the background of the images in Q may be found in the salient places of the I images so an improvement is achieved.
- Results show that in order to explore the optimal configuration, it is not possible to optimize independently at each stage because changing a parameter at one stage may affect the best choice for other stages. As it was introduced at the very beginning of this chapter, the problem shall be split into a two sub-problems, first assessing the visual part and then the whole system.

5 Budget

This project has been developed using the resources provided by the Image Processing Group of UPC. However, we made an estimation of how much would it cost to handle this project on a cloud computing platform such as Amazon Web Services, being the pricing of a suitable GPU capable device 0.64 €/h. Also estimating that it would run experiments during an average of around 100 hours/week it would lead to a cost of 1,280 €. Regarding software, there would be no costs as everything we used was open source.

So, the main costs of this project comes from the computing resources and the salary of the researches. I considered that my position as well as the one of the Phd student supervising me were of junior engineer, while the two professors who were advising me had a wage/hour of a senior engineer. I also considered the total duration of the project was 20 weeks, as depicted in the Gantt diagram in Figure 1.1.

	Amount	**Wage/hour**	**Dedication**	**Total**
Junior engineer	1	8,00 €/h	30 h/week	4,800 €
Junior engineer	1	8,00 €/h	4 h/week	640 €
Senior engineer	2	20,00 €/h	2 h/week	1,600 €
Computing services	1	0.64 €/h	100 h/week	1,280 €
			Total	**8,320 €**

Table 5.1: Budget of the project

6 Conclusions

The main objective of this project was to design a retrieval system to find personal objects in egocentric images. We built it using state of the art techniques to take into account the visual information and exploring strategies to take profit of the time stamps. Compared to the defined baseline, the contributions reported in this document have shown that the system is helpful for the task. So, we can consider that the main goal has been successfully achieved. We also think these results might be useful as a baseline for further research on this field.

The facts of not having any annotated datasets as well as not having any baselines to be compared with have made it an evolutionary task. We had to face up to making several decisions such as choosing a good dataset to work with, choosing an appropriate way to annotate it regarding to the problem that we wanted to solve, exploring and choosing proper strategies to take profit of the temporal information of the images as well as discarding approaches that did not seem to be helpful at early or medium stages of the research.

One of the most relevant results of this work is the fact that, although being a common strategy in many other tasks, applying weighted masks from the center did not improve the results. Whereas using saliency maps improved significantly the results as it was expected.

Another think to point is the fact that the different flags do not seem to be independent as it can be interpreted from the results. Changing the set-up for a specific flag makes the other flags switch in their optimal configuration. This encourages to do further research to find out the possible relationships and to build more complete validation environments.

Though all these difficulties, it has been a worthy experience and a nice task to research aiming to a useful application in such an up-to-date field. Learning what CNNs are, how to work with them, what are they capable of and come to understand the restriction that big data brings to resources as well as the vital importance of optimizing in computationally terms.

As a future work, we suggest to explore different approaches for the temporal reordering stage that might improve the system performance. Referring to the visual part, a fine-tunning could be performed in order to adapt the network to the egocentric images and improve its improve the accuracy when extracting the local convolutional features.

7 Appendices

As appendices we can found our extended abstract as well as its poster accepted to the 4th Workshop on Egocentric Vision included in the CVPR 2016 [19]. There is also the detailed Gantt diagram.

Where did I leave my phone ?

Cristian Reyes, Eva Mohedano, Kevin McGuinness and Noel E. O'Connor
Insight Centre for Data Analytics
Dublin, Ireland
cristian.reyes@estudiant.upc.edu
eva.mohedano@insight-centre.org

Xavier Giro-i-Nieto
Universitat Politecnica de Catalunya
Barcelona, Catalonia/Spain
xavier.giro@upc.edu

1. Introduction

The interest of users in having their lives digitally recorded has grown in the last years thanks to the advances on wearable sensors. Wearable cameras are one of the most informative ones, but they generate large amounts of images that require automatic analysis to build useful applications upon them. In this work we explore the potential of these devices to find the last appearance of personal objects among the more than 2,000 images that are generated everyday. This application could help into developing personal assistants capable of helping users when they do not remember where they left their personal objects. We adapt a previous work on instance search [3] to the specific domain of egocentric vision.

2. Methodology

Our goal is to rank the egocentric images captured during a day based on their likelihood to depict the location of a personal object. The whole pipeline is composed of the following stages: ranking by visual similarity, partition between candidate/non-candidate images and temporal-aware reranking within each class.

2.1. Ranking by Visual similarity

Given a certain set of query images Q depicting the object to be found, the algorithm starts by producing a ranking of the images of the day I ordered by their visual similarity score ν. This score is computed according to [3], which uses a bag of visual words model built with local features from a convolutional neural network (CNN).

A feature vector $q = f(Q)$ is generated from the set of images in Q that depict the object to locate. Three different approaches have been explored to define f:

a) No Mask: The q vector is built by averaging the visual words of all the local CNN features from the query images.

b) Mask: The q vector is built by averaging the visual words of the local CNN features that fall inside a query bounding box that surrounds the object. This allows to consider only the visual words that describe the object.

c) Weighted Mask: The q vector is built by averaging the visual words of the local CNN features of the whole image, but this time weighted depending on their distance to the bounding box. This allows to consider the context in addition to the object.

2.2. Detection of Candidate Moments

As a second step, a thresholding technique is applied to the ranking in order to partition the I set into two subsets named Candidates (C) and Discarded (D) moments.

Two different thresholding techniques were considered in order to create the C and $D = I \setminus C$ sets: TVSS (Threshold on Visual Similarity Scores) and NNDR (Nearest Neighbor Distance Ratio). The TVSS technique builds $C = \{i \in I : \nu_i > \nu_{th}\}$. The NNDR technique is based in the one described by Loewe [2]. Let ν_1 and ν_2 be the two best scores, then it builds $C = \left\{i \in I : \frac{\nu_i}{\nu_1} > p_{th}\frac{\nu_2}{\nu_1}\right\}$.

2.3. Temporal-aware reranking

The temporal-aware reranking step introduces the concept that the lost object is not in the location with the best visual match with the query, but in the last location where it was seen. Image sets R_C and R_D are built by reranking the elements in C and D, respectively, based on their time stamps. The final ranking R is built as the concatenation of $R = [R_C, R_D]$.

We considered two strategies for the temporal reranking: a straightforward sorting from the latest to the earliest timestamp, or a a more elaborate one that introduces diversity.

The diversity-aware configuration avoids presenting consecutive images of the same *moment* in the final ranked list. This is especially important in egocentric vision, where sequential images in time often present a high redundancy. Our diversity-based technique is based in the interleaving of samples, which is frequently used in dig-

ital communication. It consists in ordering temporally the images in I but knowing for each image if it belongs to C or D. So we might have something similar to $O = \{i_1^D, ..., i_{k-1}^D, i_k^C, ..., i_{l-1}^C, i_l^D, ..., i_{m-1}^D, i_m^C, ..., i_{n-1}^C\}$. Then $R_C = \{i_k^C, i_l^C, i_m^C, i_{k+1}^C, i_{l+1}^C, i_{m+1}^C, i_{k+2}^C, ...\}$ and R_D is built analogously.

3. Experiments

3.1. Dataset annotation

Our work has been developed over the NTCIR Lifelogging Dataset [1] which consists of anonymised images taken every 30 seconds over a period of 30 days. Each day contains around 1,500 images.

This dataset was annotated for this work with five personal objects which could be lost: a phone, headphones, a watch and a laptop. In particular, they were tagged as *relevant* the last appearance of the object within each day.

Queries were defined by considering that the user had a collection of images of the object, not only one. The Q set contained from 3 to 5 images per category. These images showed the objects clearly and were used to build the q vector. This assumption is realistic as the object to be found could be defined from past appearances from the same dataset.

3.2. Training

The proposed system presents some parameters that were learned with the training part of the dataset.

A visual vocabulary for Bag of Words was learned from around 14,000 images of 9 days, generating a total of 25,000 centroids. The thresholds ν_{th} and p_{th} respectively were also learned on the same 9 days used for training. The optimal values found are detailed in Table 1.

	No Mask	Mask	Weighted Mask
ν_{th}	0.04	0.01	**0.04**
p_{th}	0.17	**0.11**	0.14

Table 1. Optimal thresholds. In bold those that gave highest mAP

3.3. Test

For evaluating the performance, Mean Average Precision (mAP) was computed for each day, taking into account all the categories. Then these values have been averaged over 15 test days and presented in Table 2.

Applying a thresholding technique has demonstrated to be helpful, as the combination of the object masking and the NNDR thresholding technique has shown the best results.

It must be noticed that mAP is not the best measure in diversity terms, so despite the fact that mAP decreases, the images that form the top of the ranking have shown to be from more diverse scenes as it is shown in Figure 1.

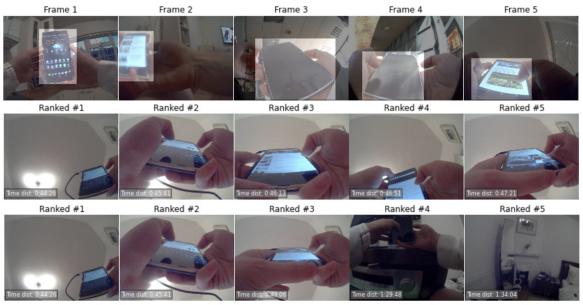


Figure 1. Results obtained for a search in category phone for a certain day. First row are the images that form Q with mask, second row results using NNDR and third results using NNDR + Div.

	No Mask	Mask	Weighted Mask
Temporal Ordering	0.051	0.051	0.051
Visual Similarity	0.102	0.082	0.111
TVSS	0.113	0.111	0.139
NNDR	0.086	**0.176**	0.093
TVSS + Div	0.096	0.082	0.118
NNDR + Div	0.066	**0.166**	0.049

Table 2. mAP results obtained when testing over 15 days.

4. Conclusions

This work has presented a good baseline for further research on the problem of finding the last appearance of an object in egocentric images.

Instance search based on bags of convolutional local features has shown promising results on egocentric images. Thresholding and temporal diversity techniques have improved the performance of visual only cues.

We plan to extend the annotations to neighbor images that may also depict relevant information to locate the location where the object was found. This way, not only one image would be considered as relevant, as assumed in the presented experiments.

References

[1] C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, and R. Albatal. Ntcir lifelog: The first test collection for lifelog research. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pisa, Italy, July 2016. ACM.

[2] D. G. Loewe. Distinctive image features from scale-invariant keypoints. page 20, 2004.

[3] E. Mohedano, A. Salvador, K. McGuinness, F. Marques, N. E. O'Connor, and X. Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 2016.

Where did I leave my phone ?

C. Reyes[1], E. Mohedano[1], K. McGuinness[1], N. E. O'Connor[1], X. Giro-i-Nieto[2]

[1] Insight Centre for Data Analytics, Dublin, Ireland [2] Universitat Politècnica de Catalunya, Barcelona, Catalonia/Spain

IEEE 2016 Conference on Computer Vision and Pattern Recognition

CVPR 2016

Motivation

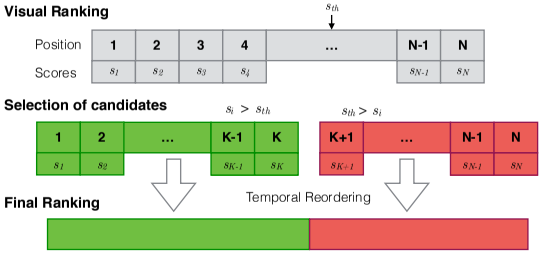
In this work we explore the potential of wearable cameras to find the last appearance of personal belongings among a large volume of images that are generated every day.

This application could help into developing personal assistants capable of helping users when they do not remember where they left their personal objects.

Our goal is to rank the egocentric images captured during a day based on their likelihood to depict the location of a personal object.

Methodology

The whole pipeline is composed of the following stages:

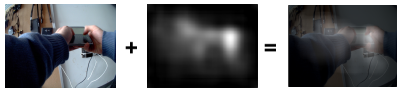


Visual Ranking

- **Bag of Words** structure for encoding the features extracted using a **Convolutional Neural Network.** [1]
- Three **masking strategies** to extract the CNN features for the queries:

- **3 to 5 visual examples** to **depict** it from different points of view.

- **Saliency maps** for the target images to down-weight de background.

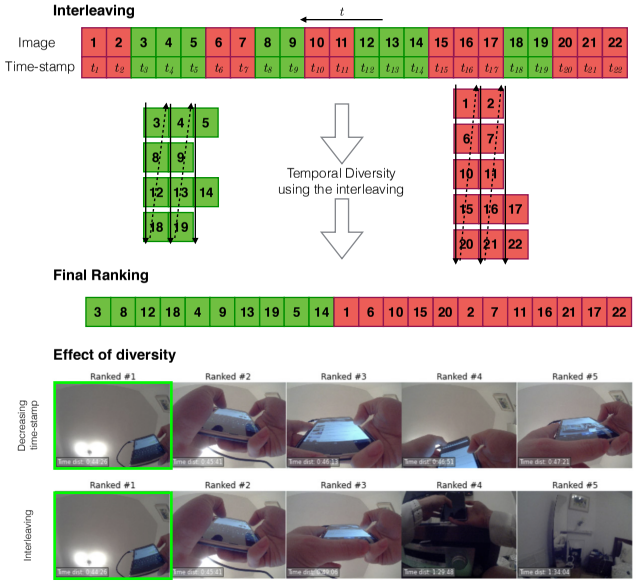


Ranking Threshold

Absolute (TVSS) $C = \{i \in I : \nu_i > \nu_{th}\}$

Adaptive (NNDR) $C = \left\{i \in I : \frac{\nu_i}{\nu_1} > p_{th}\frac{\nu_2}{\nu_1}\right\}$

Temporal Reranking



Dataset Annotation

- NTCIR Lifelogging Dataset [2] *not public*
- Annotation of the last daily appearance over 35,000 images from 30 days
- Categories: watch, phone, laptop and headphones

Results

- Mean Average Precision

	No Mask	Mask	Weighted Mask
Temporal Ordering	0.051	0.051	0.051
Visual Similarity	0.102	0.082	0.111
TVSS	0.113	0.111	0.139
NNDR	0.086	**0.176**	0.093
TVSS + Div	0.096	0.082	0.118
NNDR + Div	0.066	**0.166**	0.049

Table 2. mAP results obtained when testing over 15 days.

- Mean Reciprocal Rank

$$\mathrm{MRR}_q = \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{q^*}$$

	No Mask	Mask	Weighted Mask
Temporal Ordering	0.051	0.051	0.051
Visual Similarity	0.157	0.139	0.163
TVSS	0.216	0.212	0.171
NNDR	0.213	0.180	0.257
TVSS + Div	0.231	0.216	0.169
NNDR + Div	0.223	0.184	**0.269**

Table 1: MRR over 15 days using Full Image on target.

	No Mask	Mask	Weighted Mask
Temporal Ordering	0.051	0.051	0.051
Visual Similarity	0.150	0.173	0.178
TVSS	0.240	0.200	0.168
NNDR	0.274	0.136	0.242
TVSS + Div	0.249	0.206	0.174
NNDR + Div	**0.283**	0.147	0.257

Table 2: MRR over 15 days using Saliency Maps on target.

Conclusions

Good baseline for further research on this problem.

Instance search based on bags of convolutional local features has shown promising results on egocentric images.

Thresholding and temporal diversity techniques have improved the performance of visual only cues.

References

[1] Mohedano E, Salvador A, McGuinness K, Giró-i-Nieto X, O'Connor N, Marqués F. "Bags of Local Convolutional Features for Scalable Instance Search". ACM ICMR 2016.

[2] Gurrin, C., Joho, H., Hopfgartner, F., Zhou, L., and Albatal, R. "NTCIR Lifelog: The First Test Collection for Lifelog Research". SIGIR 2016

Annotations

Acknowledgements

Co-funded by the Erasmus+ Programme of the European Union

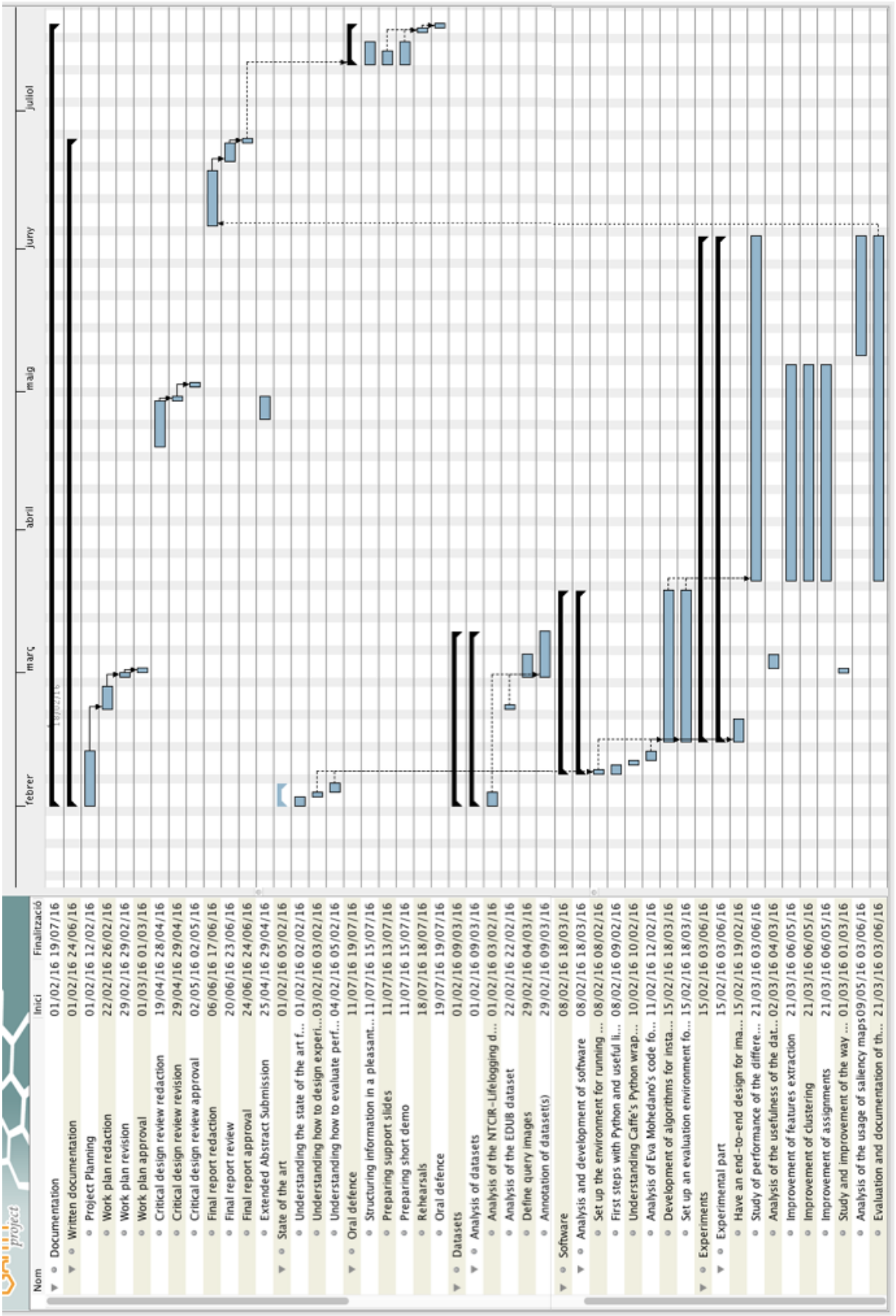


Figure 7.1: Detailed Gantt Diagram of the Thesis.

Bibliography

[1] Marc Bolaños, Mariella Dimiccoli, and Petia Radeva. Towards storytelling from visual lifelogging: An overview. *CoRR*, abs/1507.06120, 2015.

[2] Marc Bolaños and Petia Radeva. Ego-object discovery. *CoRR*, abs/1504.01639, 2015.

[3] F. Hopfgartner L. Zhou C. Gurrin, H. Joho and R. Albatal. Ntcir lifelog: The first test collection for lifelog research. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2016.

[4] Nick Craswell. *Encyclopedia of Database Systems*, chapter Mean Reciprocal Rank, pages 1703–1703. Springer US, Boston, MA, 2009.

[5] Li Fei-Fei, Rob Fergus, and Antonio Torralba. Recognizing and learning object categories, 2007.

[6] Abhinav Gupta, Aniruddha Kembhavi, and Larry S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1775–1789, October 2009.

[7] Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. Lifelogging: Personal big data. *Foundations and Trends® in Information Retrieval*, 8(1):1–125, 2014.

[8] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrel. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia Open Source Competition*, 2014.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[10] Sanna Kumpulainen, Kalervo Järvelin, Sami Serola, Aiden R Doherty, Alan F Smeaton, Daragh Byrne, and Gareth JF Jones. Data collection methods for task-based information access in molecular medicine. 2009.

[11] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision*, 43(1):29–44, June 2001.

[12] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[13] Li-Jia Li and Fei-Fei Li. What, where and who? classifying events by scene and object recognition. In *ICCV*, pages 1–8. IEEE Computer Society, 2007.

[14] David G. Loewe. Distinctive image features from scale-invariant keypoints. page 20, 2004.

[15] Katina Michael. Wearable computers challenge human rights.

[16] Eva Mohedano, Amaia Salvador, Kevin McGuinness, Ferran Marques, Noel E. O'Connor, and Xavier Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 2016.

[17] Junting Pan, Kevin McGuinness, Elisa Sayrol, Noel O'Connor, and Xavier Giro-i Nieto. Shallow and deep convolutional networks for saliency prediction. CVPR 2016.

[18] Guoping Qiu. Indexing chromatic and achromatic patterns for content-based colour image retrieval. *Pattern Recognition*, 35(8):1675–1686, 2002.

[19] Cristian Reyes, Eva Mohedano, Kevin McGuinness, N. O'Connor, and X. Giró-i Nieto. Where did i leave my phone? Extended abstract at the CVPR Workshop on First-Person (Egocentric) Vision. Las Vegas, NV, USA. 2016.

[20] M. S. Ryoo and J. K. Aggarwal. Hierarchical recognition of human activities interacting with objects. In *CVPR*. IEEE Computer Society, 2007.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[23] G Stix. Photographic memory wearable cam could help patients stave off effects of impaired recall. *Scientific America*, 1, 2011.

[24] Christian Thiel and Christoph Thiel. *Enforcing Data Privacy in the Age of Google Glass*, pages 220–229. Springer Fachmedien Wiesbaden, Wiesbaden, 2014.

[25] Jianxin Wu, Adebola Osuntogun, Tanzeem Choudhury, Matthai Philipose, and James M. Rehg. A scalable approach to activity recognition based on object use. In *In Proceedings of the International Conference on Computer Vision (ICCV), Rio de*, 2007.

[26] Justin Zobel and Alistair Moffat. Inverted files for text search engines. *ACM Comput. Surv.*, 38(2), July 2006.