Location recognition on lifelog images via a discriminative combination of generative models

Alessandro Perina
alessandro.perina@iit.it
Matteo Zanotto
matteo.zanotto@iit.it
Baochang Zhang
baochang.zhang@iit.it
Vittorio Murino
vittorio.murino@iit.it

Pattern Analysis and Computer Vision (PAVIS)
Istituto Italiano di Tecnologia
Genova, Italy

Abstract

This paper presents a generative framework aimed at the analysis of a "visual lifelog" captured by wearing a camera for long periods of time. Here, we focused on location recognition and we propose the use of an ensemble of heterogeneous generative models able to capture the different aspects that characterize each location. We defined the likelihood of the ensemble as the likelihood of a mixture model whose components are the individual models themselves. Our results set the new state of the art on all the tasks associated with the SenseCam-32 dataset and outperform Bayesian model averaging and several other discriminative combination techniques. From a theoretical perspective, this paper proposes a principled (discriminative) combination of heterogeneous generative models able to cope with extremely challenging classification tasks and it demonstrates that combining such diverse heterogeneous models is indeed advantageous.

1 Introduction

It is a common belief that in the near future, wearable technology will be the next computing revolution. Such wearable systems are intended to be used in a seamless way like a piece of clothing and they are at the basis of "lifelogging", a rather old concept originated with the idea of wearing a gadget or computer that records a large portion of a person's daily life [1, 2]. An overview of the main challenges in lifelogs analysis can be found in [17].

Among all wearable sensors, the first lifelogging cameras are recently becoming available for a large number of people to use. Examples are *SenseCam*, *Autographer* and *Narrative* cameras. All of them use a passive record-it-all approach, automatically shooting a photo every 10-30 seconds. However, the soon-to-be enormous amount of images must be organized in order to be useful, and simply using temporal arrangement of the shots is totally unsatisfactory.

Figure 1: The first 54 images of a lifelog. Notice the high blur (red boxed images) and the dramatic changes in illumination (yellow box)

The analysis of lifelog images has yet to receive much attention from the computer vision community. However, recent works of Grauman's group [12, 13] aimed at temporally segmenting videos for summarization are related to analysis of lifelogs due to the characteristics of the data of interest. Other works focused on analyzing social interactions [16], actions performed or undergone [19, 22, 23], and detecting novel thus likely interesting events [14]. The unsupervised approach of [5] is probably the first (and only) attempt to deal with real lifelogs images. Its goal was discovering the locations visited in several days worth of images captured with a SenseCam.

Likewise [5], this paper focuses on location recognition. By exploiting recent and classical generative models used for scene understanding [4, 6, 7, 8, 9, 10, 11], we propose a new framework which learns in a principled way a discriminative combination of weights combining the different heterogeneous generative models in multiple complexities[1].
To the best of our knowledge, this is the first attempt to combine such diverse and heterogeneous ensemble of generative models and our choice is motivated by an intuitive and a theoretical reason. First, the locations one visits in his life are so different that it cannot exist a single model which would be able to fit well everywhere. This was also noted in [27]. For example, our favorite grocery store, the one we usually go to, could nicely be modeled by a full bag-of-word approach like LDA [6], whereas locations like a kitchen is probably well recognized by looking at the objects that contains, and finally contained environments like our work cubicle or our car may well be modeled by an exemplar based-method [8] or by a panoramic reconstruction method like the epitome [4]. The second motivation for the need for model combination is more theoretical: when none of the models in an ensemble is the true data generator model (TDG), there usually exists a combination that can replicate the behavior of the TDG more closely than any individual model on its own [21, 25].
The rest of the paper proceeds as follows; in Sec. 2, we introduce the probem and we give an overview of our approach. In Sec. 2.1, we will describe the generative models for scene analysis we considered, and in Sec. 2.2, we will present our model combination technique. Finally, in Sec. 3 we will present the results, and Sec. 4 will draw some conclusions.

2 Combination of heterogeneous generative models

Lifelogs pose serious challenges to computer vision researchers. Cameras are usually worn around the neck or attached to clothes and this causes non-linear and unpredictable motion

[1] e.g., learned with with different parameters

which causes blur and rapid changes in the scene. Figure 1a shows 54 consecutive images taken in a period of ∼15 minutes over which the bearer changes location few times (kitchen, living room, garage). Notice how most of the frames are blurred, while few are highly blurred and difficult to understand even for a human. Moreover, the illumination exhibits dramatic changes over short time periods even when the bearer stays in the same location. Last but not least, temporal continuity of the stream of shots is very loose. Another intrinsic characteristics of lifelogs is that, in a real scenario, the labeled data available to accomplish a task (e.g., location recognition) are inherently scarce: most of the images, in fact, can only be labeled by the bearer of the camera and crowd-sourcing is difficult, if not impossible.

Motivated by the aforementioned problems, in this paper we introduce a framework aimed at helping in the automatic organization of lifelogs. Importantly, it only assumes that few labeled images for each location visited are available. At training time, several (M) heterogeneous generative models in multiple (K) complexities (which compose the ensemble $\mathcal{E}$ of cardinality $M \times K$) are learned. Generative models are in fact chosen for their generalization power which makes them suitable in situations where a small number of training samples is available. We will refer with $\mathcal{M}^l_{m,k}$ to the k-th complexity of model m, learned on images of class l[2]. As second step, for each class, we learn a (discriminative) combination of weights $\pi^l_{m,k}$ which combines the likelihoods of each model $\mathcal{M}$ and defines the likelihood of the ensemble $\mathcal{E}$.
At test time, given an unseen image, we first perform inference under the previously learned generative models, then we combine the likelihoods using $\pi^l_{m,k}$, and finally we take a classification decision comparing the likelihoods.

2.1 Data generating models

In the context of scene analysis and recognition, existing algorithms such as appearance-based spatial clustering models, like epitomes [4, 5] are impractical because of the extremely unfavorable imaging conditions.
A substantially different way to deal with diversity of imaging conditions and geometric variation in objects or entire scenes is the bag-of-words approach. In this representation, all the spatial information is discarded and images are represented as disordered "bags" of image features (or visual words) [6]. This simplifying assumption makes this approach more robust to image transformations and hence more suitable to analyze lifelogs. In this paper we considered $M = 6$ recent generative models [6, 7, 8, 9, 10, 11] in $K = 8$ complexities, and we combine them in an ensemble $\mathcal{E}$. The next section will detail these models adopting the following notation: we will use $\mathbf{c}^t$ to refer to the t-th bag and l^t for its label. c^t_z is the count of features z in bag t. Finally, we will indicate the likelihood of bag $\mathbf{c}$ under model $\mathcal{M}^l_{m,k}$ as with $\mathcal{L}(\mathbf{c}|\mathcal{M}^l_{m,k})$. In this paper we used quantized SIFT features, but any feature can be used; see the experimental section for details.

Counting Grids and extensions [7, 9, 10]. The counting grid model (**CG**) is a generalization of the epitome model [4] for bags-of-words (BoW). Instead of looking for co-occurrence of features, it enforces that bags are generated taking windows (a "scene" or "view") in a larger scene (the "visual world" or "panorama"), therefore it models the spatial interdependence in image feature histograms which standard BoW approaches omit. This model (and

[2] For convenience, any of the indexes may be omitted when no confusion arises.

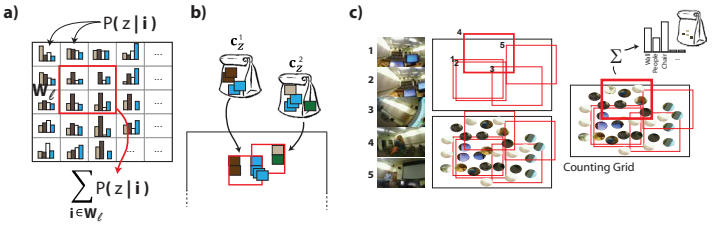

Figure 2: **a)** Counting Grid illustration. The counting grid is a grid of word distributions, bags are generated by picking a window. **b)** The feature mappings in the windows must be consistent in the overlapping region. **c)** Cartoon illustration of a counting grid learned using images from the location "conference room".

its extensions) are well suited for lifelogs as a lot of the variations we see in bags of image-derived features is due to camera movement.

Formally, the basic counting grid is a set of distributions over features z on a 2-dimensional discrete grid indexed by $\mathbf{i} = (x, y)$. $\mathbf{E} = [E_x, E_y]$ describes the extent of the counting grid. Since each element of the grid is a normalized distribution, $\sum_z p(z|\mathbf{i}) = 1$ everywhere on the grid. The counting grid geometry is illustrated in Fig. 2a.

A given bag $\mathbf{c}$ is assumed to follow the distribution of visual words found in a window of the counting grid. In particular, each bag can be generated by first selecting a position $\mathbf{k}$ on the grid and placing a window of dimension $\mathbf{W}$ in that position. Then, all counts in this window are averaged to form a histogram representing the parameters of the Multinomial distribution from which the features in the bag are sampled. In other words, the position of window $\mathbf{k}$ in the grid is the only latent variable and once it is given, the probability of the bag of features can be computed as $p(c_z|\mathbf{k}) = \prod_z \left(\frac{1}{|\mathbf{W}|} \cdot \sum_{\mathbf{i} \in W_\mathbf{k}} p(z|\mathbf{i}) \right)^{c_z}$ where $W_\mathbf{k}$ indicates the particular window placed at location $\mathbf{k}$.

The subsequent work from the same authors introduced the Spring-Lattice Counting Grid [10] (**SLCG**), illustrated by Fig. 3a. Here, the feature bags originating from different image sections are mapped to different sub-windows in the counting grid in a configuration that is close to, but not exactly the same as the configuration of the source sectors.

The reconfigurable part model [11]. This model (**RPM**) represents a scene as a collection of parts arranged in a reconfigurable pattern as illustrated by Fig. 3c. Each image is divided into predefined sectors $\mathbf{S} = S_x \times S_y$ (like the tessellated version of [7, 9]), and a latent variable specifies which "region model" (e.g., road, furniture, grass, etc.) is assigned to each image region.

Latent Dirichlet Allocation and Mixture models [6, 8]. Mixture models (**MM**) have been employed successfully for scene and location recognition. The idea is to learn one mixture model per-class where each centroid captures a typical configuration of the scene. Torralba et al. [8] modeled locations with a mixture of C Gaussians over Gist descriptors.

Latent Dirichlet Allocation (**LDA**) is a widely-used hierarchical model of count data originally developed to aid in the analysis of text corpora. LDA learns a set of topics, which captures the co-occurrence of the visual words. The model allows mixing multiple topics to explain a single bag [6].

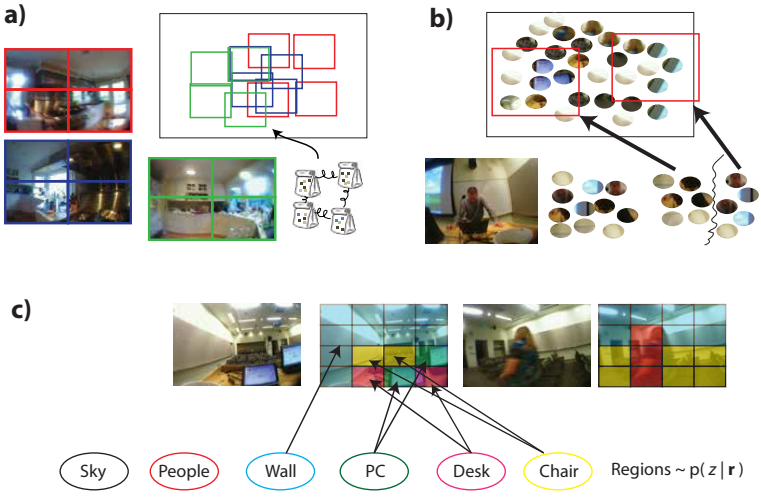

Figure 3: **a)** Spring-Lattice Counting Grid [10]. **b)** Componential Counting Grids. **c)** Reconfigurable part model.

Componential Counting Grids [9]. The original CG model [7] is a mixture model, assuming the existence of a single source (e.g., one window) for all the features in one bag. On the other hand LDA is an admixture that allows mixing of multiple topics to explain a single bag. Componential Counting Grids (CCG) [9] combine the two models allowing to have multiple sources (e.g., mapping position in the grid) for each bag with a mathematical formulation which is identical to LDA: indeed, CCG represent an admixture of windows. Each source usually captures layers and/or parts of images, like objects or people, and then is mapped independently on the grid. Experimentally, the componential nature of the model helps to extract foreground and to capture parallax effects.
As for standard Counting Grids, also CCG can be extended to deal with image representations consisting of multiple bag-of-words, each associated to a section of the original image. This results in Tessellated Componential Counting Grids (**TCCG**) which will be used in the following.
For details on the likelihoods and the learning algorithms the reader is referred to the original papers.

2.2 Learning combinations of heterogeneous generative models

Learning errors can often be reduced by combining a set of models and this poses the challenge of finding effective ways to create combinations of learners. Bayesian model averaging (BMA) [20] provides a theoretically optimal framework for combining models, however it assumes that the *true* data generator model is present in the ensemble which is clearly a rather strong assumption. Several empirical works showed that BMA yields to worse results than ad-hoc methods (e.g., Bagging, Stacking, etc.) whenever its assumptions are not met. Indeed, as Minka pointed out [25], BMA acts more like soft model selection than like a strategy for model combination and it places too much weight on the model which provides the highest likelihood.

Here, we do not assume that any of the models in the ensemble is the true data generator model (TDG). Instead of searching for a linear combination that can more closely replicate the TDG behavior than any individual model on its own [21, 25], we look for a discriminative combination of weights $\pi_{m,k}$. Furthermore, we compute this combination per-class as, in general, different combinations of models could be better suited for different classes.

More formally, our goal is to combine the likelihoods of each model to generate the likelihood of an ensemble

$$\mathcal{L}(\mathbf{c}|\mathcal{E}^l) = \sum_m \sum_k \pi^l_{m,k} \cdot \mathcal{L}(\mathbf{c}|\mathcal{M}^l_{m,k}) = \sum_{u=m\times k} P(u) \cdot P(\mathbf{c}|u, \mathcal{M}^l_u) \tag{1}$$

If we assume that the mixing coefficients π's sum to 1, Eq. 1 becomes the likelihood of bag $\mathbf{c}$ under a *mixture of heterogeneous generative models* where each of the $M \times K$ components is a generative model. It is important to note that since our mixture components are different heterogeneous generative models, they can be separately learnt using all the available data to avoid overtraining.

Working in a one-vs-all setting, for each class l, we propose to compute the weights π^l which maximize the margin between the average conditional log-likelihood ratio (A-CLLR) of positive samples $\bar{F}^+_m$ and that of negative samples $\bar{F}^-_m$. This formulation is expected to robustly find weights which define a more discriminative mixture of models than standard Bayesian techniques [20, 21].
The average conditional log-likelihood of a set of bags, is defined as

$$A-CLL = \frac{1}{T}\sum_{t=1}^{T} \log p(l^t = l|\mathbf{c}^t) \tag{2}$$

Formally, we want to find the weights that maximizes the margin

$$F_m = \overbrace{\frac{1}{|\mathcal{D}_l|}\sum_{\mathbf{c}^t\in\mathcal{D}_l} \log\left(\frac{\sum_{m,k}\pi^l_{m,k}\cdot\mathcal{L}^{CV}(\mathbf{c}^t|\mathcal{M}^l_{m,k})}{\sum_c\sum_{m,k}\pi^l_{m,k}\cdot\mathcal{L}^{CV}(\mathbf{c}^t|\mathcal{M}^{l=c}_{m,k})}\right)}^{\bar{F}^+_m} - \overbrace{\frac{1}{|\overline{\mathcal{D}}_l|}\sum_{\bar{\mathbf{c}}^t\in\overline{\mathcal{D}}_l} \log\left(\frac{\sum_{m,k}\pi^l_{m,k}\cdot\mathcal{L}^{CV}(\bar{\mathbf{c}}^t|\mathcal{M}^l_{m,k})}{\sum_c\sum_{m,k}\pi^l_{m,k}\cdot\mathcal{L}^{CV}(\bar{\mathbf{c}}^t|\mathcal{M}^{l=c}_{m,k})}\right)}^{\bar{F}^-_m}$$
$$\text{subject to } |\pi^l| = 1 \quad \text{and} \quad \pi^l_{m,k} \geq 0 \ \ \forall m,k \tag{3}$$

where $\mathcal{L}^{CV}(c^t_z|\mathcal{M}^l_{m,k})$ is the likelihood of sample t under the model $\mathcal{M}^l_{m,k}$ learned through cross-validation, $\mathcal{D}_l$ and $\overline{\mathcal{D}}_l$, represent the set of training samples belonging and not belonging to class l, respectively, and $|\mathcal{D}_l|$ and $|\overline{\mathcal{D}}_l|$ are their cardinalities. For simplicity, we assumed a uniform prior over classes l.

To maximize F_m we must solve a nonlinear constrained optimization problem. We used the Broyden-Fletcher-Goldfarb-Shanno algorithm. It is an interior-point algorithm with a approximation of the Hessian based on difference between successive gradient vectors.
This technique requires the computation of the partial deriviative of F_m w.r.t. $\pi_{m,k}$, which can be computed as follows for the first term of Eq. 3

$$\frac{\partial F^+_m}{\partial \pi^l_{m,k}} = \sum_{c^t} \frac{\mathcal{L}(\mathbf{c}^t|\mathcal{M}^l_{m,k})\cdot(\sum_c\sum_{m,k}\pi^l_{m,k}\cdot\mathcal{L}(\mathbf{c}^t|\mathcal{M}^{l=c}_{m,k})) - \sum_c\mathcal{L}(\mathbf{c}^t|\mathcal{M}^{l=c}_{m,k})\cdot(\sum_{m,k}\pi^l_{m,k}\cdot\mathcal{L}(\mathbf{c}^t|\mathcal{M}^l_{m,k})}{\sum_{m,k}\pi^l_{m,k}\cdot\mathcal{L}(\mathbf{c}^t|\mathcal{M}^l_{m,k})\cdot\sum_c\sum_{m,k}\pi^l_{m,k}\cdot\mathcal{L}(\mathbf{c}^t|\mathcal{M}^{l=c}_{m,k})}$$

where we dropped the superscript "CV" for brevity. The derivative of the second term of Eq. 3 is similar and we did not report it in the paper.

Method	Cit.	Tessellation	Complexities
CG	[7]	$\mathbf{S} = 1 \times 1$	$\mathbf{E} = [10 \times 10, 15 \times 15, \ldots 45 \times 45]$, $\mathbf{W} = 5 \times 5$
tCCG	[9]	$\mathbf{S} = 4 \times 4$	$\mathbf{E} = [10 \times 10, 15 \times 15, \ldots 45 \times 45]$, $\mathbf{W} = 5 \times 5$
SLCG	[10]	$\mathbf{S} = 2 \times 2$	$\mathbf{E} = [10 \times 10, 15 \times 15, \ldots 45 \times 45]$, $\mathbf{W} = 5 \times 5$
LDA	[6]	$\mathbf{S} = 1 \times 1$	**Topics** $= [10, 20, 30, 50, 70, 90, 110, 130]$
Mixt.Mod.	[8]	$\mathbf{S} = 1 \times 1$	**Centers** $= [2, 3, 4, 5, 6, 7, 8, 9]$
RPM	[11]	$\mathbf{S} = 4 \times 4$	**Regions** $= [2, 3, 4, 5, 6, 7, 8, 9]$

Table 1: Models and complexities in the ensemble $\mathcal{E}$. For [10] and [11] we considered the image tessellation $\mathbf{S}$ used in the original papers.

At this point it is worth noting how our technique allows to exploit all the data in both the generative and discriminative steps. This is crucial as lifelogs can't have a lot of training data and standard method would overtrain [3]. In particular, the benefits of this framework are:

- Being based on a mixture of heterogeneous models, all the training data can be used to learn each component of the $\{m,k\}$ mixture.
- The weights are computed per-class. Our hypothesis in fact is that different locations are better modeled by different algorithms (or combinations). This was already noticed in [27] in the context of indoor images.
- To discriminatively learn the combination of weights, Eq. 3 makes use of *all* the data. This is important as it allows to avoid overtraining of the discriminative part of the model.
- The method computes an actual likelihood $\mathcal{L}(\mathbf{c}|\mathcal{E}^l)$ which can be further used in conjunction with other generative models, for example a hidden Markov model to exploit the temporal relations between images.

3 Experiments

We considered the portion of lifelog collected in [5][4]. The authors selected and labeled a random subset of the lifelog (5000 images) and created the SenseCam-32 dataset, which comprises images from 32 different locations the camera bearer visited.
In all the experiments we used SIFT as visual words extracted from 8×8 patches spaced by 4 pixels. The obtained descriptors were clustered in $Z = 200$ visual words. The count matrices $\mathbf{c}^t$ are obtained counting the number of occurrences of visual word z in image t. For each of the generative models, we considered the complexities reported in Tab. 1, the cardinality of ensemble is $|\mathcal{E}| = 6 \times 8 = 48$[5].

As a first test, to further motivate our approach, we performed a series of One-vs-All trials on the SenseCam-32 dataset comparing the generative models in $\mathcal{E}$. In Fig. 4a, we report the best scoring method (in terms of AUCs) for each category. Results are very intuitive: LDA scores well on large unconstrained environments like grocery stores or parks, RPM and

[3] See the experimental evaluation for more details.

[4] Data available at www.alessandroperina.com

[5] Concerning counting grids, it is well know that they are not sensitive to the window size $\mathbf{W}$, as long as it is sufficiently big. The only parameter that matters is the capacity of the model $\kappa = |\mathbf{E}|/|\mathbf{W}|$ which measures how many independent windows can fit into the grid. For this reason we only considered $\mathbf{W} = 5 \times 5$.

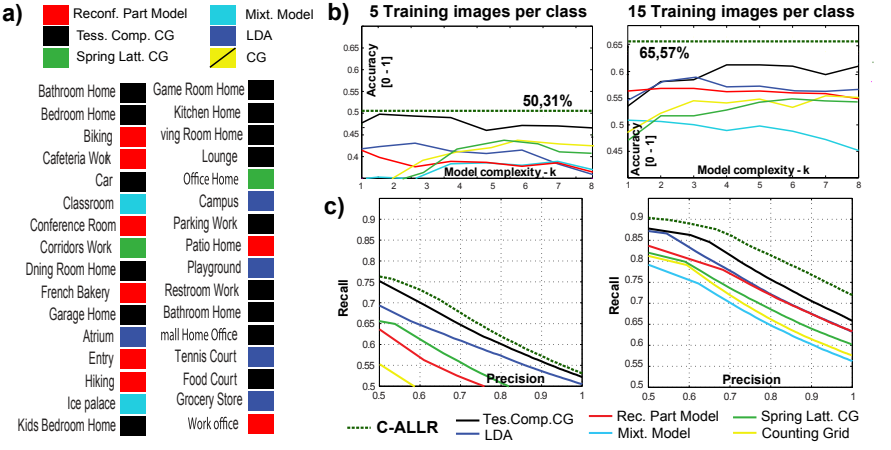

Figure 4: **a)** One-Vs-All results on SenseCam-32 Dataset. Despite, overall TCCG [9] provides better classification accuracy, LDA tends to score better on large unconstrained outdoor scenes, while RPM on contained scenes characterized by a small amount of visual cutter. Model Combination Results on **b)** SenseCam-32 [10] dataset, and **c)** the batches of day images.

SLCG are better suited for small fixed locations without much foreground, on the remaining classes TCCG tends to work better.

As second test, we evaluated the classification accuracy of each model separately, and several combination methods. In particular: *i)* several generative Bayesian fusion methods [20, 21, 30] and Naive Bayes, *ii)* discriminative fusion methods [29], and *iii)* kernel methods based on features [28] or built from the log-likelihood values $\mathcal{L}$'s [31].
Bayesian model averaging [20] accounts for uncertainty of model correctness $P(\mathcal{M})$ by integrating over the model space and weighting each model by the probability of it being the "correct" model, e.g., $P(\mathbf{c}|l) = \int_{\mathcal{M}} P(\mathbf{c}|l,\mathcal{M}) \cdot P(\mathcal{M})$. Differently from [20], the Bayesian model combination technique of [21] integrates out the uncertainty about which *model combination* is correct. Finally, [30] is a combination method that assumes that the combining weights are generated by a Dirichlet distribution.
We also considered a weighted Naive Bayes classifier whose attributes are the models themselves, e.g., $P(l,<m,k>) = P(l) \cdot \prod_{m,k} \mathcal{L}(\mathbf{c}|\mathcal{M}^l_{m,k})^{w_{m,k}}$, where $w_{m,k}$ are feature-dependent weights, equal to 1 for Naive Bayes, but which can also be trained discriminatively [29] by maximizing the conditional log-likelihood or the mean-squared error.
Finally, we also considered mutual information kernels [31] built starting from the log-likelihoods.

We compared the results with our approach and we tested on the original SenseCam-32 dataset [10] using the standard validation procedure, varying the number of training images. Results are reported in Fig.4b and Tab.2. As shown, our combination method always outperforms each individual model in the ensemble, even with a very limited number of training images. For the competitor methods, when possible, we also computed a per-class combination weights. As shown in Tab.2, this always led to lower accuracy and it is clearly due to overtraing. It is well known, in fact, that discriminative models require more data.
Finally, it is also worth noting how our method sets the new state of the art on this dataset, also outperforming standard discriminative approaches like SVMs on the BoW descriptors

	A-CLLR	BMA [21]	BMC [21]	RBC [30]	NB	wNB (CLL) [29]	wNB (MSE) [29]	MI-K [31]
Single	n.p.	58,4%	57,3%	61,6%	62,1%	60,1 %	61,4 %	n.p.
per-class	**65,6%**	n.p.	n.p.	56,3%	n.p.	59,2%	58,8%	61,0%

Table 2: Comparison with other model combination techniques (15 Training images), *n.p.* stands for "not possible". Also for 5 training images A-CLLR scored the best, while [29, 31] overtrained

(48,9%) or the spatial pyramid kernel [28] (56.45%) which, as noted in [10], overtrains.

One of the benefits of the combination of Eq. 1, is that it effectively computes a likelihood $\mathcal{L}(\mathbf{c}|\mathcal{E})$, which can be then naturally used in conjunction with a hidden Markov model to exploit the weak temporal relationships between the images of a lifelog [8, 10]. In a real word scenario in fact, a user would download the lifelogging camera at the end of the day and images can be analyzed in batch.
From the authors of [10] we obtained the 2 days-worth of SenseCam images (e.g., two days completely labeled) used in their last experiment, for a total of 3753 images. The goal here was to compute the place posterior probabilities at time t, given all the previous images $P(l^t = k|\mathbf{c}^{1:t})$. We employed the same paradigm of [10], learning the models in $\mathcal{E}$ from SenseCam-32 images and using the forward-backward procedure to recursively assigning the label to the day's images. We fixed the HMM's observation likelihood to the likelihood of a model or the ensemble. Unlike [8], we used EM to estimate the transition matrix $A_{k|c} = P(l^t = k|l^{t-1} = c)$ and the place posteriors in an unsupervised way, simply fitting the likelihood to the day's images. Results are shown in Fig. 4c in terms of precision-recall curves. Again, our fusion technique increased the performance.

As final test, we considered the 67-indoor scene dataset [27] where the authors noted that different indoor locations are better classified by exploiting different properties. Differently from the SenseCam-32, here there is plenty of training data. Combining generative models by maximizing Eq.3 yielded to 33,8% classification accuracy which outperforms by a large margin each model in $\mathcal{E}$ and it is very close to discriminative multi-cue methods[6].

4 Conclusions

We studied the problem of location recognition on visual lifelogs. The task is very challenging because of the scarcity of training data, extreme image conditions and because the locations one visits are very diverse: from outdoor environments to large indoor locations like a grocery store to a cubicle or car's interiors. While the outdoor locations are well modeled using global properties, indoor locations are better discovered by recognizing objects or by reconstructing the scene. This motivated us to propose a model combination method, robust to overtraining and based on an ensemble of heterogeneous generative models. Our approach combines evidence from different complexities of different models, outperforming each model on its own and other advanced techniques. To the best of our knowledge, this is the first time this task has been successfully carried out with such diverse ensemble and this paper shows how this is a viable and effective option.

[6] See a benchmark in [10]

References

[1] S.Mann Wearable Computing: A First Step Toward Personal Imaging IEEE Computer, vol 32(2),1997

[2] G.Bell, J.Gemmell Total Recall: How the E-Memory Revolution Will Change Everything. (Book) Penguin Group, 2009

[3] R. Data, W. Ge, J. Li and J.Wang Toward Bridging the Annotation-Retrieval Gap in Image Search by a Generative Modeling Approach ACM Multimedia 2006

[4] K.Ni, A.Kannan, A.Criminisi and J.Winn Epitomic Location Recognition *IEEE Trans. Pattern Anal. Mach. Intell*, 31(12), 2009

[5] N.Jojic, A.Perina, V.Murino Structural Epitome: a way to summarize one's visual input NIPS 2010: 1027-1035

[6] Fei-Fei Li, P.Perona A Bayesian Hierarchical Model for Learning Natural Scene Categories CVPR (2) 2005: 524-531

[7] A.Perina, N.Jojic Image analysis by counting on a grid CVPR 2011: 1985-1992

[8] A.Torralba, K.P.Murphy, W.T.Freeman and M.A.Rubin Context-based vision system for place and object recognition ICCV 2003: 273-280

[9] A.Perina, N.Jojic Capturing layers in image collections with componential models: from the layered epitome to the componential counting grid. CVPR 2013

[10] A.Perina, N.Jojic Spring Lattice Counting Grids: Scene recognition using deformable positional constraints ECCV 2012

[11] S.Parizi, J.Oberlin, P.Felzenszwalb Reconfigurable Models for Scene Recognition CVPR 2012

[12] W.Lee, J.Ghosh, K.Grauman Discovering Important People and Objects for Egocentric Video Summarization CVPR 2012

[13] K.Lu, and K.Grauman Story-Driven Summarization for Egocentric Video CVPR 2013

[14] O.Aghazadeh, J.Sullivan and S.Carlsson Novelty Detection from an Ego-Centric Perspective CVPR 2012

[15] A.Fathi, X.Ren, J.Rehg Learning to Recognize Objects in Egocentric Activities CVPR 2011

[16] A.Fathi, X.Hodgins, J.Rehg Social Interations: A First-Person Perspective CVPR 2012

[17] J.Machajdik, A.Hanbury, A.Garz and R.Sablatnig Affective computing for wearable diary and lifelogging systems: An overview OAGM 2011

[18] X. Ren and M. Philipose Egocentric recognition of handled objects: Benchmark and analysis CVPR Workshops 2009.

[19] M.S.Ryoo and L.Matthies First-Person Activity Recognition: What Are They Doing to Me CVPR 2013

[20] J.Hoeting, D.Madigan, A.Rafery, C.Volinsky Baysuian Model Averaging: A tutoral Statistical Science 14:382

[21] K.Monteith, J.Carroll, K.Seppi and T.Martinez Turning Bayseian Model Averaging into Baysian Model Combination IJCNN, 2011

[22] K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast Unsupervised Ego-Action Learning for First-Person Sports Video CVPR 2011

[23] H.Pirsiavash and D.Ramanan Recognizing Activities of Daily Living in First-Person Camera Views CVPR 2012

[24] A.R. Doherty, S.E. Hodges, A.C. King, A.F. Smeaton, E. Berry, C.J. Moulin, A. Lindley, P. Kelly, and C. Foster Wearable Cameras in Health: The State of the Art and Future Possibilities American Journal of Preventive Medicine (editorial), 44(3), 2013

[25] T.Minka Bayesian Model Averaging is not model combinations MIT Media Lab Note, 2000

[26] http://research.microsoft.com/EN-US/UM/CAMBRIDGE/PROJECTS/SENSECAM/publications.htm

[27] A. Quattoni, and A.Torralba Recognizing Indoor Scenes CVPR 2009

[28] S.Lazebnik C.Schmid and J.Ponce Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories CVPR 2006

[29] N.Zaidi, J.Cerquides, M.Carman, G.Webb Alleviating Naive Bayes attribute independence assumption by attribute weighting J.Mach.Learning Research, 14 (2013)

[30] J.Cerquides, R. De Mantaras Robust Bayesian Linear Classifier Ensembles ECML 2005

[31] M. Seeger Covariance kernels from Bayesian generative models NIPS 2001