Temporal Segmentation of Egocentric Videos to Highlight Personal Locations of Interest

Antonino Furnari, Giovanni Maria Farinella, Sebastiano Battiato

Department of Mathematics and Computer Science
University of Catania
{furnari,gfarinella,battiato}@dmi.unict.it

Abstract. With the increasing availability of wearable cameras, the acquisition of egocentric videos is becoming common in many scenarios. However, the absence of explicit structure in such videos (e.g., video chapters) makes their exploitation difficult. We propose to segment unstructured egocentric videos to highlight the presence of personal locations of interest specified by the end-user. Given the large variability of the visual content acquired by such devices, it is necessary to design explicit rejection mechanisms able to detect negatives (i.e., frames not related to any considered location) learning only from positive ones at training time. To challenge the problem, we collected a dataset of egocentric videos containing 10 personal locations of interest. We propose a method to segment egocentric videos performing discrimination among the personal locations of interest, rejection of negative frames, and enforcing temporal coherence between neighboring predictions.

Keywords: first person vision, egocentric video, context-based analysis, aware computing, video segmentation

1 Introduction and Motivation

Wearable cameras have recently become popular in many application scenarios including law enforcement [34], assistive technologies [19], life-logging [16] and social cameras [23]. Despite the large amount of information that such systems can potentially acquire, the exploitation of egocentric videos is quite difficult due to the lack of explicit structure, e.g., in the form of scene cuts or video chapters. Depending on the considered goal, long egocentric videos tend to contain much uninformative content like, for instance, transiting through a corridor, walking, or driving to the office. Therefore, as pointed out in [24], automated tools are needed to enable faster access to the information stored in such videos and index their visual content. Towards this direction, researches have investigated methods to produce short informative video summaries from long egocentric videos [1, 21, 35], recognize the actions performed by the wearer [18, 10, 26, 20, 27], and segment the videos according to detected ego-motion patterns [24, 25]. While current literature focuses on providing general-purpose methods which are usually optimized using data acquired by many users, we argue that, given

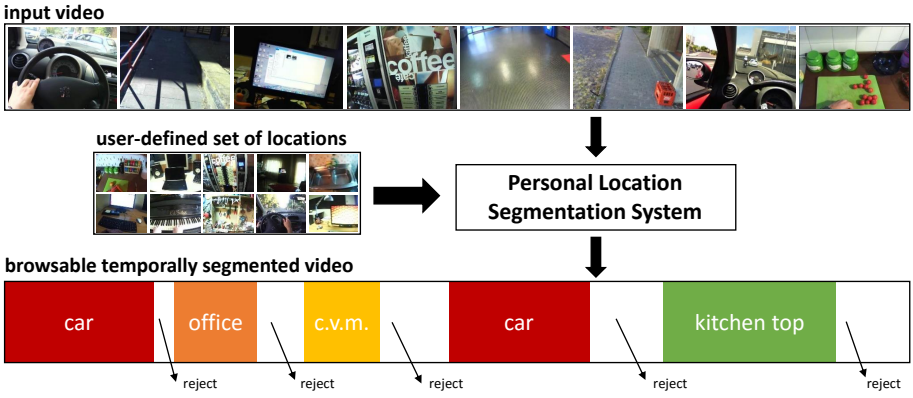


Fig. 1. Overall schema of the proposed temporal segmentation of an egocentric video.

the subjective nature of egocentric videos, more attention should be devoted to user-specific methods.

In this paper, we propose to segment unstructured egocentric videos into coherent shots related to user-specified personal locations of interest. Our notion of personal location builds on the one introduced in [12]: *a fixed, distinguishable spatial environment in which the user can perform one or more activities which may or may not be specific to the considered location.* According to this notion, a personal location is specified at the instance level (e.g., my kitchen, my office, my car), rather than at the category level (e.g., a kitchen, an office, a car). It should be noted that personal locations are very specific to the user defining them and should not be confused with the general concept of visual scene. Indeed, a given set of personal locations could include different instances corresponding to the same scene category (e.g., office vs lab office). Under such conditions, classical scene-tuned image descriptor such as GIST [22] would perform poorly as shown in [12]. Fig. 1 shows a schema of the investigated problem. The user defines a number of locations of interest by providing minimal training data in the form of short videos (e.g., a 10 seconds video per location). The user is just asked to wear his camera and briefly look around while he is in the considered location. Therefore, each training video is deemed to contain the most common views of the considered location. Given the input egocentric video and the user-defined set of locations, the task is to establish for each frame in the video if it is related to either one of the considered personal locations or none of them (i.e., it is a negative sample). We want to emphasize that in a real-world scenario in which the system is set up by the end user himself, training must be simple and achievable with few training data. Moreover, given the large variability exhibited by egocentric videos, it is unfeasible to ask the user to acquire a significant quantity of negative samples [12]. Therefore, we assume that only positive samples of different locations are provided by the user and propose a method to detect negative samples automatically, without training on them. We would like to note that avoiding to learn from negative frames is not limiting from a performance stand point. In fact, as we show in

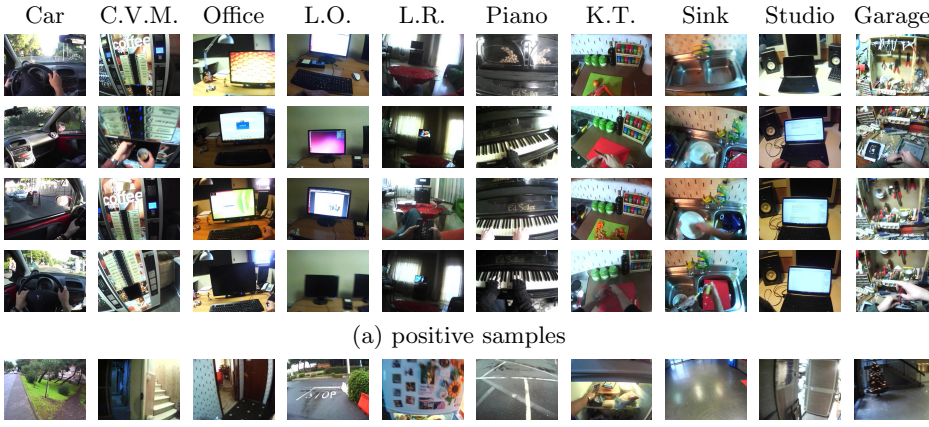
the experiments, even when negative samples are available for learning purposes, training a multi-class classifier to correctly detect them is not trivial.

The proposed method uses a Convolutional Neural Network (CNN) to discriminate among different locations and a Hidden Markov Model (HMM) to enforce temporal coherence among neighbouring predictions. Differently from previous works, we treat the rejection of negative samples explicitly and introduces a non-parametric method to reject negative frames. Being non-parametric, our method does not need any negative samples at training time. We discuss the computational performances of the proposed method and also suggest a simplified system which is efficient enough to run in real-time. This allows possible uses in real-time, assistive-related applications. The main contributions of this paper are summarized in the following: 1) we study the problem of segmenting egocentric videos using minimal user-provided data and propose a dataset comprising more than 2 hours of labelled egocentric videos covering 10 different locations plus various negative environments, 2) we propose a method for egocentric video segmentation and negative sample rejection which trains only on the available positive samples, 3) we show how CNNs can be exploited in this domain (where training data is assumed to be scarce) experimenting a series of simple architectural tweaks to avoid over-fitting during fine-tuning and optimize computational performances. Experiments show that the proposed system outperforms baselines and existing approaches by a good margin and with an accuracy of over the 90% on the challenging sequences included in the proposed benchmark dataset.

The remainder of the paper is organized as follows. Section 2 summarizes the related work. Section 3 describes the dataset. Section 4 presents the proposed system. Section 5 reports the experiments and discusses the results. Finally, Section 6 concludes the paper.

2 Related Work

Researchers have explored the issues and opportunities related to first person vision ever since the 90s. Relevant endeavors have focused on investigating contextual awareness and localization [29, 3, 33], improving human-machine interaction [30, 2], understanding and recognizing human activities [10, 11, 5, 8], indexing and summarizing egocentric videos [24, 21, 25]. In particular, our work is related to previous studies on contextual awareness in wearable and mobile computing. In [9], efficient computational methods for scene categorization are proposed for embedded devices. In [29], some basic tasks and locations related to the Patrol game are recognized from egocentric videos in order to assist the user during the game. In [3], personal locations are recognized from egocentric video based on the approaching trajectories observed from the camera point of view. In [33], a context-based vision system for place and scene recognition is proposed and deployed on a wearable system. In [31], still images of sensitive spaces are detected for privacy purposes combining GPS information and an image classifier. In [5], Convolutional Neural Networks and Random Decision Forests



(a) positive samples

(b) negative samples

Fig. 2. Some sample frames from the proposed dataset.

are exploited to recognize human activities from egocentric images. In [12], a benchmark of different wearable devices and image representation techniques for personal context recognition is proposed.

While current literature focuses primarily on providing general-purpose methods which can rely on data acquired by multiple user, we focus on a personalized scenario in which the user himself provides the training data and sets up the system. Under such conditions, it is not possible to rely on a big corpus of supervised data, since it is not feasible to ask the user to collect and label it. Moreover, differently from related works, we explicitly consider the problem of rejecting negative samples, i.e., recognizing locations the user is not interested in, so to discard irrelevant information.

3 Proposed Dataset

We collected a dataset of egocentric videos related to ten different personal locations, plus various negative ones. The considered locations arise from a possible daily routine: Car, Coffee Vending Machine (C.V.M.), Office, Lab Office (L.O.), Living Room (L.R.), Piano, Kitchen Top (K.T.), Sink, Studio, Garage. The dataset has been acquired using a hardware configuration similar to the best performing in the benchmark proposed in [12]: a Looxcie LX2 camera equipped with a wide angular converter. Such configuration allows to acquire videos at a resolution of 640×480 pixels and with a Field Of View of approximately $100°$. The use of a wide-angular device is justified by the ability to acquire a large amount of information on the scene, albeit at the cost of radial distortion, which in some cases requires dedicated computation [14, 13]. Fig. 2 shows some example frames from the dataset. The dataset exhibits a high degree of intra-class variability (e.g., Car and Garage classes) and small inter-class variability in some cases (e.g., Office, Lab Office and Studio classes).

Sequence	Context transitions	Length
1	Car → N → Office → N → Lab Office	00:11:27
2	Office → N → Lab Office	00:05:55
3	Lab Office → N → Office → N → C.V.M.	00:07:24
4	TV → N → Piano → N → Sink	00:11:40
5	Kitchen → N → Sink → N → Piano	00:10:41
6	Kitchen → N → Sink → N → TV	00:11:18
7	Piano → N → Sink → N → TV	00:04:57
8	Studio → N → Car → N → Garage	00:06:51
9	Car → N → Garage → N → Studio	00:05:17
10	Car → N → Studio → N → Garage	00:06:05
	Total length	01:21:35

Table 1. A summary of the location transitions contained in the test sequences. "N" represents a negative segment (to be rejected by the final system).

As discussed in the introduction, we assume that the user is required to provide only minimal data to define his personal locations of interest. Therefore, the training set consists in 10 short videos (one per each location) with an average length of 10 seconds per video. The test set consists in 10 video sequences covering the considered personal locations of interest, negative frames and transitions among locations. Each frame in the test sequences has been manually labeled as either one of the 10 locations of interest or as a negative. Table 1 summarizes the content of the test sequences with the related transitions. The dataset is also provided with an independent validation set which can be used to optimize the hyper-parameters. The validation set contains 10 medium length (approximately 5 to 10 minutes) videos of activities performed in the considered locations (one video per location). Validation videos have been temporally subsampled in order to extract exactly 200 frames per location, while all frames are considered for training and test videos. We have also acquired 10 medium length videos containing negative samples from which we uniformly extract 300 frames for training and 200 frames for validation. Negative samples are provided in order to allow comparisons with methods which explicitly learn from negatives. Please note that the proposed method does not need to learn from negatives and hence it discards them at training time.

The proposed dataset contains 2142 positive, plus 300 negative frames for training, 2000 positive, plus 200 negative frames for validation and 132234 mixed (both positive and negative) frames for testing purposes. The dataset is available at the web page *http://iplab.dmi.unict.it/PersonalContexts/*.

4 Proposed Method

Given an egocentric video as an ordered collection of image frames $\mathcal{V} = \{I_1, \ldots, I_n\}$, our system must be able to 1) correctly classify each frame I_i as one of the considered locations, 2) reject negative frames, 3) segment temporally coherent sub-sequences related to the locations of interest. The system eventually returns the segmentation $\mathcal{S} = \{C_1, \ldots, C_n\}$, where $C_i \in \{0, \ldots, M-1\}$ is the class

label associated to frame I_i ($C_i = 0$ representing the negative class label) and M is the total number of classes including negatives ($M = 11$ in our case - 10 locations, plus the negative class). Rejection of negative samples is usually tackled increasing the number of classes by one and explicitly learning to recognize negative samples. However, this procedure requires a number of training negative samples which may not be easily acquirable by the user in a real-world scenario. Indeed, given the large variability of visual content acquired by wearable devices, it would be infeasible to ask the user to acquire a sufficient number of representative negative samples. Therefore, we propose to treat negative rejection separately from classification and introduce a non-parametric rejection mechanism which does not need negative samples at training time.

We first consider a multi-class component which is trained solely on positive samples to discriminate among the considered positive $M - 1$ classes. Since the multi-class model ignores the presence of negative frames, it only allows to estimate the posterior probability:

$$p(C_i|I_i, C_i \neq 0). \tag{1}$$

We propose to quantify the probability $p(C_i = 0|I_i)$ of a given frame I_i to be negative as the uncertainty of the multi-class model in predicting the class labels related to last k frames (in our experiments we use $k = 30$, which is equivalent to one second at 30 fps). Specifically, considering that both the visual content and class label are deemed to change slowly in egocentric videos, we assume that the past k frames $\mathcal{I}_i^k = \{I_i, I_{i-1}, \ldots, I_{\max(i-k+1,1)}\}$ are related to the same class. Such assumption may be imprecise when $\mathcal{I}_i^k$ contains the boundary between two different locations. However, such cases are rather rare and if k spans over one second or less, the assumption only affects the boundary localization accuracy and is not expected to have a huge impact on the overall accuracy. Since the multi-class model has been tuned only on positive samples, we expect it to exhibit low uncertainty when the frames in $\mathcal{I}_i^k$ belong to one of the positive classes, while we expect a large uncertainty in the case of negative samples. Similarly to [15], we measure model uncertainty computing the variation ratio of the distribution of labels $\mathcal{Y}_i^k = \{y_i, \ldots, y_{max(i-k+1,1)}\}$ predicted within $\mathcal{I}_i^k$ by maximizing the posterior probability in Eq. (1): $y_i = \arg\max_j p(C_i = j|I_i, C_i \neq 0), j = 1, \ldots, M - 1$. We finally assign the probability of I_i being a negative sample as follows:

$$p(C_i = 0|I_i) = 1 - \frac{\sum_j \mathbb{1}(y_j = \tilde{\mathcal{Y}}_i^k)}{\#\{\mathcal{Y}_i^k\}} \tag{2}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function and $\tilde{\mathcal{Y}}_i^k$ represents the mode of $\mathcal{Y}_i^k$. It should be noted that the definition reported in Eq. (2) is totally arbitrary and encodes the belief that the model should agree on similar inputs if they are positive samples. In practice, given a number of predictions computed within a small temporal window, we quantify the probability of having a negative sample as the fraction of labels disagreeing with the mode.

Considering that $C_i = 0$ and $C_i \neq 0$ are disjoint events (and hence $p(C_i \neq 0|I_i) = 1 - p(C_i = 0|I_i)$), the probabilities reported in Eq. (1) and (2) can be

combined as follows:

$$p(C_i|I_i) = \begin{cases} p(C_i = 0|I_i) & \text{if } C_i = 0 \\ p(C_i \neq 0|I_i) \cdot p(C_i|I_i, C_i \neq 0) & \text{otherwise} \end{cases}. \quad (3)$$

The final class prediction for frame I_i (including the rejection of negative samples) can be obtained maximizing Eq (3) as follows:

$$C_i^* = \arg\max_j p(C_i = j|I_i) \quad (4)$$

Given the nature of egocentric videos, subsequent frames will be likely related to the same location, while a sudden change of location is a rare event. Such a prior can be taken into account during the computation of the final segmentation using a Hidden Markov Model (HMM). We consider the probability $p(\mathcal{S}|\mathcal{V})$ which, according to the Bayes' rule, can be expressed as follows:

$$p(\mathcal{S}|\mathcal{V}) \propto p(\mathcal{V}|\mathcal{S})p(\mathcal{S}). \quad (5)$$

Assuming conditional independence of the frames with respect to each other given their classes ($I_i \perp\!\!\!\perp I_j|C_i,\ \forall i, j \in \{1, 2, \ldots, n\}, i \neq j$), and applying the Markovian assumption on the conditional probability distribution of the class labels ($p(C_i|C_{i-1} \ldots C_1) = p(C_i|C_{i-1})$), Eq. (5) can be written as:

$$p(\mathcal{S}|\mathcal{V}) \propto p(C_1) \prod_{i=2}^{n} p(C_i|C_{i-1}) \prod_{i=1}^{n} p(I_i|C_i). \quad (6)$$

Probability $p(C_1)$ is assumed to be constant over the different classes and can be ignored when maximizing Eq. (6). Probability $p(I_i|C_i)$ can be inverted using the Bayes law $p(I_i|C_i) \propto p(C_i|I_i)p(I_i)$. Since I_i is observed, term $p(I_i)$ can be ignored, while $p(C_i|I_i)$ is estimated using Eq. (3). Eq. (6) can be hence written as:

$$p(\mathcal{S}|\mathcal{V}) \propto \prod_{i=2}^{n} p(C_i|C_{i-1}) \prod_{i=1}^{n} p(C_i|I_i). \quad (7)$$

The term $p(C_i|C_{i-1})$ is the HMM state transition probability. Transition probabilities in Hidden Markov Models can generally be learned from the data as done in [33], or defined ad hoc to express a prior belief as done in [31]. Since we assume that few training data should be provided by the user and no labeled sequences are available at training time, we define an ad-hoc transition probability as suggested by [31]:

$$p(C_i|C_{i-1}) = \begin{cases} \varepsilon, & \text{if } C_i \neq C_{i-1} \\ 1 - (M-1)\varepsilon, & \text{otherwise} \end{cases} \quad (8)$$

where ε is a small constant (we use the machine accuracy in double precision 2.22×10^{-16} in our experiments). The state transition probability defined in Eq. (8) enforces coherence between subsequent states and penalizes random state

changes. The final segmentation of the input egocentric video is obtained choosing the one which maximizes the probability in Eq. (7) by using the Viterbi algorithm [4]:

$$\mathcal{S}^* = \arg\max_{\mathcal{S}} p(\mathcal{S}|\mathcal{V}). \quad (9)$$

5 Experimental Settings and Results

Experiments are performed on the dataset described in Section 3. All compared methods are trained on the whole training set and evaluated on the test sequences. The validation set is used to tune hyper-parameters and select the best performing iteration in the case of CNNs. In Section 5.1, we study the performances of the proposed method, paying particular attention to optimization. Specifically, we evaluate different architectural tweaks which help reducing over-fitting when fine-tuning Convolutional Neural Networks on our small realistic dataset ($\approx$ 200 samples per class) and reduce computational requirements. Moreover, we discuss the influence of the different components included in our method (i.e., multiclass classifier, rejection mechanism, and HMM). In Section 5.2 we compare our method with respect to the state of the art.

5.1 Proposed Method: Optimization and Performances Evaluation

The multi-class classifier employed in the proposed method could be implemented using any algorithm able to output posterior probabilities in the form of Eq. (1). We consider Convolutional Neural Networks given their compactness and the superior performances shown on many tasks including personal location recognition [12]. In particular, following [12], we fine-tune the VGG-S network proposed in [7] on our training set. Since the VGG network has been trained on the ImageNet dataset, we expect the learned features to be related to objects and hence relevant to the task of location recognition, as highlighted in [36].

Optimization of the Multi-Class Classifier Fine-tuning a large CNN using a small training set ($\approx$ 200 samples per class) is not trivial and some architectural details can be tuned in order to optimize performances. Specifically, we assess the impact of the following architectural settings: 1) locking the convolutional layers (i.e., setting their relative learning rate to zero), 2) disabling dropout in the fully connected layers, 3) reducing the number of units in the fully connected layers from 4096 to 128, 4) removing the fully connected layers and attaching a logistic regression (softmax) layer directly to the last convolutional layer. In the following, we discuss different combinations of the aforementioned architectural settings in order to assess the influence of each considered setting. Results for these experiments are reported in Table 2 and Table 3.

Table 2 is organized as follows. Each row of the table is related to a different experiment. The first column (Id) reports unique identifiers for the considered methods. The second column (Settings) summarizes the architectural settings

		Accuracy			Computational Performances	
Id	**Settings**	**Discrim.**	**+Rejection**	**+HMM**	**Dimensions**	**Time**
[*a*]		76.90	69.60	73.83	378 MB	13.23 ms
[*b*]	L	83.30	76.06	83.22	378 MB	13.13 ms
[*c*]	L ND	**94.53**	**85.00**	**88.63**	378 MB	13.10 ms
[*d*]	L 128	83.07	77.49	82.84	34 MB	10.32 ms
[*e*]	L ND 128	77.09	71.99	73.59	34 MB	10.28 ms
[*f*]	L LR	**92.31**	**81.00**	**85.37**	26 MB	10.23 ms

Table 2. Optimization of the multi-class classifier. Architectural settings: L the convolutional layers are locked, ND dropout is disabled, 128 fully connected layers are reduced to 128 units and reinitialized, LR fully connected layers are replaced by a single logistic regression layer. Reported times are average per-image processing times. Maxima per column are reported in **underlined bold digits**, while second maxima are reported in **bold digits**.

related to the specific method. The third column (Discrimination) reports the accuracy of the multi-class model alone (i.e., class labels are directly computed using Eq. (1)). Note that such accuracy values are computed removing all negative samples from the test set. The fourth column (+Rejection) reports the accuracy of the models after applying the proposed rejection method (i.e., labels are obtained using Eq. (4)). The fifth column (+HMM) reports the accuracy of the complete method including the Hidden Markov Model (i.e., final segmentation labels are obtained using Eq. (9)). Column 6 reports the size of the models in megabytes. Column 7 finally reports the average time needed to predict the class label of a single frame[1]. Table 3 reports per-class true positive rates for the considered configurations.

The reported results highlight the importance of tuning the considered architectural settings to improve both computational performances and accuracy. In particular, locking the convolutional layers allows to significantly improve the performances of the fine-tuned model (compare [*b*] to [*a*] in Table 2)[2]. Significant performance improvements are observable when the CNN is evaluated alone (Discrim. column) as well as when the model is integrated in the proposed system (columns +Rejection and +HMM). This result highlights how the unlocked network suffers from over-fitting, due to the high number of parameters to optimize with relatively few training data. It should be noted that, in our experiments, only convolutional layers are locked, while fully connected ones are still optimized. Locking convolutional layers, hence, allows to use part of the network as a bank of object-related feature extractors (the pre-trained convolutional layers), while optimizing the way such features are combined in the fully connected layers.

Disabling dropout has a positive impact when convolutional layers are locked and fully connected layers are fine-tuned ([*c*] vs [*b*]). This indicates that dropout is causing the model to underfit due to the scarcity of training data. Interestingly, when fully connected layers are reduced to 128 units and hence reinitialized with

[1] Times have been estimated running the CNN models on a NVIDIA GeForce GTX 480 GPU using the Caffe framework [17]. They include the rejection of negative frames but do not take into account the application of the Hidden Markov Model.

[2] SVM models are tested on a Intel(R) Core(TM) i7-3930K CPU @ 3.20GHz with LIBSVM [6].

		Per-Class True Positive Rate (TPR)										
Id	**Settings**	**Car**	**C.V.M.**	**Gar.**	**K.T.**	**L.Off.**	**Off.**	**Piano**	**Sink**	**Stud.**	**L.R.**	**Neg.**
[*a*]		91.28	**98.73**	**98.71**	**100.0**	**95.87**	**94.81**	**98.52**	**100.0**	99.40	**99.20**	36.91
[*b*]	L	90.71	**98.53**	**98.41**	99.60	93.83	**93.57**	98.48	99.00	98.50	98.91	47.77
[*c*]	L ND	75.57	92.42	87.60	97.95	84.08	71.67	93.32	96.69	94.09	89.73	**82.34**
[*d*]	L 128	**99.09**	94.36	74.90	89.46	93.51	84.66	98.16	98.90	**99.72**	99.09	51.22
[*e*]	L ND 128	**99.57**	95.43	98.31	**100.0**	**98.54**	90.25	**98.68**	**99.51**	**99.82**	**99.14**	36.51
[*f*]	L LR	94.53	78.93	85.88	78.39	89.91	60.28	93.57	96.91	97.46	98.20	**61.66**

Table 3. Per-class true positive rates for the considered configurations. See Table 2 for a legend.

Gaussian noise, disabling dropout seems to favor overfitting as one would generally expect (compare [*e*] to [*d*]). This behavior is probably due to the inclination of randomly reinitialized layers to easily co-adapt [28]. Reducing the dimensionality of the fully connected layers to 128 units helps reducing the dimensions of the network and improving its speed, but results in a substantial loss in accuracy due to the needed reinitialization of the weights (compare [*d*] to [*c*]).

In order to devise a more compact model, we finally consider replacing the fully connected layers with a logistic regressor (i.e., a layer with 10 units followed by softmax). In this case, the locked convolutional layers of the VGG-S network are used as feature extractors, while predictions are performed combining them using a simple logistic regressor classifier. This configuration allows to greatly reduce memory and time requirements at the cost of a modest loss in terms of accuracy (compare [*f*] to [*c*], [*d*], [*e*]).

Among all compared method, the most accurate is [*c*], followed by the computationally efficient [*f*]. Both methods outperform the others by a good margin. Moreover, it is worth noting that [*f*] is more than 90% smaller and 20% faster than [*c*] while only about 3% less accurate. Such result is particularly interesting in real-time scenarios involving low-resources and embedded devices (e.g., in smart glasses or in a drone). Finally, as can be noted from Table 3, only the two best configurations (methods [*c*] and [*f*]) succeed in correctly rejecting negative samples, while other methods yield lower true positive rates.

Performances of the proposed method As discussed above, columns 3 to 5 in Table 2 report performances related to the main components involved in the proposed method, i.e., multi-class classifier, rejection mechanism and Hidden Markov Model. As can be noted, high accuracies can be achieved when discriminating among a finite number of possible locations (column Discrim.). The need for a rejection mechanism in real-world scenarios makes the problem much harder, decreasing classification accuracy by 10% in average (compare Discrim. with +Rejection columns). These results suggest that more efforts should be devoted to effective rejection mechanisms in order to make current classification systems useful in real world applications. Indeed, any real system devoted to distinguish among a number of classes must be able to deal with the negative ones. Enforcing temporal coherence using a Hidden Markov Model generally helps reducing the gap between simple discrimination and discrimination + rejection (consider for instance methods [*c*] and [*f*]). The effects of the rejection and HMM modules are qualitatively illustrated in Fig. 3. As can be noted, sim-

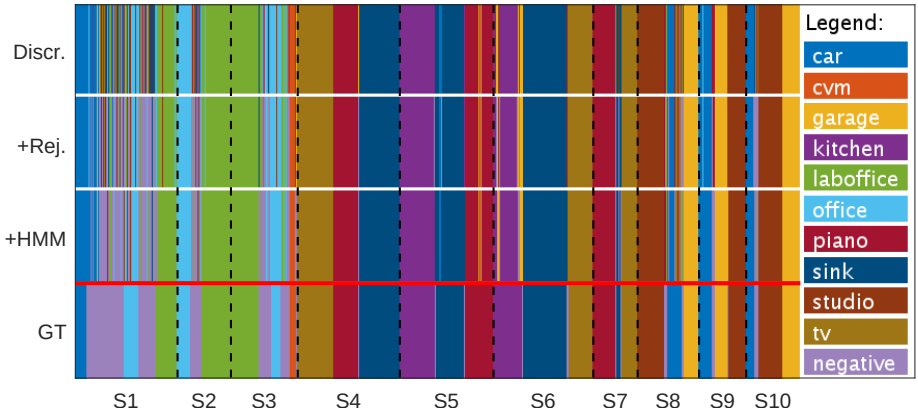


Fig. 3. Graphical representation of the labels produced by the proposed method (method [*c*] in Table 2). Each row reports the concatenation of labels produced for all test sequences. Boundaries between sequences are highlighted with black dashed lines and "S1" ... "S10" labels. The visualization is intended to qualitatively assess the influence of the rejection and HMM components on the performances of the overall system. Specifically, the first three rows report labels obtained using the multi-class classifier, the proposed rejection mechanism and the HMM, similarly to what discussed for Table 2. The last row reports the ground truth. Detailed visualizations for each sequence are available in the supplementary material available online. Best seen in color.

ple class discrimination (top row) yields noisy predictions when ground truth frames are negative. The rejection mechanism (second row) successfully detects negative segments. The use of a HMM (third row) finally helps reducing sudden changes in the predicted labels.

5.2 Comparison with the State of the Art

To assess the effectiveness of the proposed method, we compare it with respect to two baselines and an existing method for personal location recognition [12]. The first baseline tackles the location recognition problem through feature matching. The system is initialized extracting SIFT feature points from each test image and storing them for later use. Given the current frame, SIFT features are extracted and matched with all images in the training set. To reduce the influence of outlier feature points, for each considered image pair, we perform a geometric verification using the MSAC algorithm [32] based on an affine model. Classification is hence performed considering the training set image presenting the highest number of inliers and selecting the class to which it belongs. In this case, the most straightforward way to perform rejection probably consists in setting a threshold on the number of inliers: if an image is a positive, it is expected to yield a good match with some example in the dataset, otherwise only weak matches should be obtained. Since it is not clear how such a threshold should be arbitrarily set, we learn it from data. To do so, we first normalize the number of inliers by the number of features extracted from the current frame. We then

		Accuracy			Comp. Performances	
Id	**Settings**	**Discrim.**	**+Rejection**	**+HMM**	**Dimensions**	**Time**
[c]	L ND	**94.53**	**85.00**	**88.63**	378 MB	13.10 ms
[f]	L LR	**92.31**	**81.00**	**85.37**	26 MB	10.23 ms
[g]	SIFT	34.64	33.16	–	71 MB	5170.1 ms
[h]	L ND NE	73.84	76.42	79.69	378 MB	12.82 ms
[i]	SVM [12]	87.76	74.14	79.64	423 MB	97.83 ms

Table 4. Comparisons with the state of the art. Methods [c] and [f] are reported from Table 2 for convenience. Architectural settings: L the convolutional layers are locked, ND dropout is disabled, LR fully connected layers are replaced by a single logistic regression layer, SIFT the SIFT feature matching baseline, NE the model is trained on both positive and negative samples, SVM classification based on one-class and multiclass SVM classifiers.

		Per-Class True Positive Rate (TPR)										
Id	**Settings**	**Car**	**C.V.M.**	**Gar.**	**K.T.**	**L.Off.**	**Off.**	**Piano**	**Sink**	**Stud.**	**L.R.**	**Neg.**
[c]	L ND	75.57	92.42	**87.60**	**97.95**	84.08	**71.67**	**93.32**	**96.69**	94.09	89.73	**82.34**
[f]	L LR	**94.53**	78.93	85.88	78.39	**89.91**	60.28	**93.57**	**96.91**	**97.46**	**98.20**	61.66
[g]	SIFT	4.90	5.55	0.02	71.45	15.37	16.62	84.98	22.21	12.80	79.77	24.22
[h]	L ND NE	**78.16**	**95.23**	71.48	97.53	73.54	50.03	71.95	93.43	**95.70**	73.49	**95.72**
[i]	SVM [12]	74.97	**98.16**	**97.63**	**98.45**	**88.60**	**92.27**	79.13	69.25	59.16	**99.13**	06.58

Table 5. Per-class true positive rates of the compared methods. See Table 4 for a legend.

select the threshold which best separates the validation set from the training negatives. To speed up computation, input images are rescaled in order to have a standard height of 256 pixels (the same size to which images are resized when fed to CNN models), keeping the original aspect ratio.

The second considered baseline consists in a CNN trained to discriminate directly between locations of interest and negatives. In contrast with the proposed method, the baseline explicitly learns from negative samples. Hence, in our settings, the model is trained on 11 classes comprising 10 locations of interest, plus the negative class. This baseline is implemented adopting the same architecture as the one of method [c], which is the best performing configuration in our experiments. It should be noted that training negatives are independent from validation and test negatives. We also compare our method with respect to the one proposed in [12]. Such method performs negative rejection and location recognition using a cascade of One-Class and multiclass SVM classifiers trained on features extracted employing the VGG network [7].

Table 4 and Table 5 compare the performances of the considered methods. As can be noted, the proposed methods [c] and [f] retain the highest accuracies in Table 4. Requiring about 5 seconds to process each frame, the SIFT matching method ([g] in Table 4) is the slowest among the compared ones. Moreover, SIFT matching achieves poor results on the considered task, which indicates that it is not able to generalize to new views of the same scene and to cope with the many variabilities typical of egocentric videos. It should be noted that, since the SIFT baseline does not output any probability values, the HMM cannot be applied in this case.

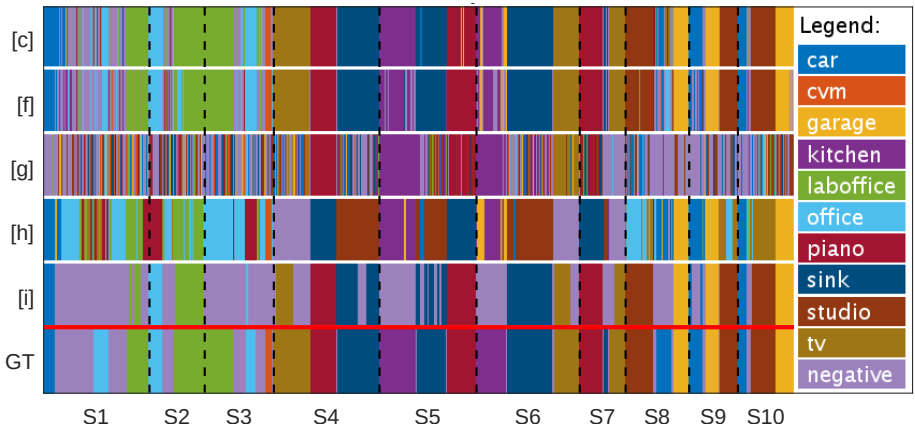


Fig. 4. Graphical representation of the segmentation results produced by the considered methods (see Table 4). Detailed visualizations for each sequence are available in the supplementary material. Best seen in color.

Baseline [*h*] retains a high TPR on negative samples (see Neg. column in Table 5). However TPRs related to other classes and the accuracy of the overall system are lower when compared to the proposed approaches. This indicates how learning from negative samples is not trivial in the proposed problem. The method introduced in [12] is outperformed by the proposed methods (compare [*i*] to [*c*]-[*f*]) and gives inconsistent results in the rejection of negative frames (see column Neg. in Table 5). Moreover, the proposed approaches are significantly faster and have smaller size. Fig. 4 finally reports segmentation results of all compared methods for qualitative assessment.

6 Conclusion

We have proposed a method to segment egocentric videos in order to highlight personal locations of interest. The system can be trained with few positive samples provided by the user. Convolutional Neural Networks are used to discriminate among positive locations, while a non-parametric rejection method is used to reject locations not specified by the user. A Hidden Markov Model is employed to enforce temporal coherence among neighboring predictions. We show how the architecture of the employed CNN can be tuned to optimize performances both in terms of accuracy and computational requirements. The effectiveness of the proposed method is assessed comparing it with respect to two baselines and a state of the art method. Future works will concentrate on studying the generalization ability of the method by considering multiple users in the personal location of interest recognition problem.

References

1. Aizawa, K., Ishijima, K., Shiina, M.: Summarizing wearable video. In: International Conference on Image Processing. vol. 3, pp. 398–401 (2001)
2. Antifakos, S., K.N.S.B., Schwaninge, A.: Towards improving trust in context-aware systems by displaying system confidence (2005)
3. Aoki, H., Schiele, B., Pentland, A.: Recognizing personal location from video. In: Workshop on Perceptual User Interfaces. pp. 79–82 (1998)
4. Bishop, C.M.: Pattern recognition and Machine Learning. Springer (2006)
5. Castro, D., Hickson, S., Bettadapura, V., Thomaz, E., Abowd, G., Christensen, H., Essa, I.: Predicting daily activities from egocentric images using deep learning. International Symposium on Wearable Computing (2015)
6. Chang, C., Lin, C.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
7. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: British Machine Vision Conference (2014)
8. Damen, D., Leelasawassuk, T., Haines, O., Calway, A., Mayol-Cuevas, W.: You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In: British Machine Vision Conference (2014)
9. Farinella, G.M., Ravì, D., Tomaselli, V., Guarnera, M., Battiato, S.: Representing scenes for real–time context classification on mobile devices. Pattern Recognition 48(4), 1086–1100 (2015)
10. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. IEEE International Conference on Computer Vision pp. 407–414 (2011)
11. Fathi, A., Li, Y., Rehg, J.M.: Learning to recognize daily actions using gaze. In: European Conference on Computer Vision. vol. 7572, pp. 314–327 (2012)
12. Furnari, A., Farinella, G.M., Battiato, S.: Recognizing personal contexts from egocentric images. In: Workshop on Assistive Computer Vision and Robotics (ACVR) in conjunction with the IEEE International Conference on Computer Vision (2015)
13. Furnari, A., Farinella, G.M., Bruna, A.R., Battiato, S.: Generalized Sobel filters for gradient estimation of distorted images. In: International Conference on Image Processing (2015)
14. Furnari, A., Farinella, G.M., Puglisi, G., Bruna, A.R., Battiato, S.: Affine region detectors on the fisheye domain. In: International Conference on Image Processing. pp. 5681–5685 (2014)
15. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. arXiv preprint arXiv:1506.02142 (2015)
16. Gurrin, C., Smeaton, A.F., Doherty, A.R.: Lifelogging: Personal big data. Foundations and Trends in Information Retrieval 8(1), 1–125 (2014)
17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM Multimedia. vol. 2, p. 4 (2014)
18. Kitani, K.M., Okabe, T., Sato, Y., Sugimoto, A.: Fast unsupervised ego-action learning for first-person sports videos. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3241–3248 (2011)
19. Lee, M.L., Dey, A.K.: Lifelogging memory appliance for people with episodic memory impairment. In: Proceedings of the 10th International Conference on Ubiquitous Computing. pp. 44–53 (2008)

20. Li, Y., Ye, Z., Rehg, J.M.: Delving into egocentric actions. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 287–295 (2015)
21. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. In: Computer Vision and Pattern Recognition. pp. 2714–2721 (2013)
22. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision 42(3), 145–175 (2001)
23. Ortis, A., Farinella, G.M., D'Amico, V., Addesso, L., Torrisi, G., Battiato, S.: RECfusion: Automatic video curation driven by visual content popularity. In: ACM Multimedia (2015)
24. Poleg, Y., Arora, C., Peleg, S.: Temporal segmentation of egocentric videos. In: Computer Vision and Pattern Recognition. pp. 2537–2544 (2014)
25. Poleg, Y., Ephrat, A., Peleg, S., Arora, C.: Compact CNN for indexing egocentric videos. arXiv preprint arXiv:1504.07469 (2015)
26. Ryoo, M.S., Rothrock, B., Matthies, L.: Pooled motion features for first-person videos. arXiv preprint arXiv:1412.6505 (2014)
27. Spriggs, E.H., De La Torre, F., Hebert, M.: Temporal segmentation and activity classification from first-person sensing. In: Computer Vision and Pattern Recognition Workshops. pp. 17–24 (2009)
28. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)
29. Starner, T., Schiele, B., Pentland, A.: Visual contextual awareness in wearable computing. In: International Symposium on Wearable Computing. pp. 50–57 (1998)
30. Starner, T., Weaver, J., Pentland, A.: Real-time american sign language recognition using desk and wearable computer based video. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(12), 1371–1375 (1998)
31. Templeman, R., Korayem, M., Crandall, D., Apu, K.: PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces. In: Annual Network and Distributed System Security Symposium. pp. 23–26 (2014)
32. Torr, P., Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78(1), 138–156 (2000)
33. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. International Conference on Computer Vision (2003)
34. White, M.D.: Police officer body-worn cameras: Assessing the evidence. Washington, DC: Office of Community Oriented Policing Services (2014)
35. Xu, J., Mukherjee, L., Li, Y., Warner, J., Rehg, J.M., Singh, V.: Gaze-enabled egocentric video summarization via constrained submodular maximization. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2235–2244 (2015)
36. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems. pp. 487–495 (2014)