

# Personal-Location-Based Temporal Segmentation of Egocentric Videos for Lifelogging Applications

Antonino Furnari\*, Sebastiano Battiato, Giovanni Maria Farinella

*Department of Mathematics and Computer Science*

*University of Catania*

---

## Abstract

Temporal video segmentation is useful to exploit and organize long egocentric videos. Previous work has focused on general purpose methods designed to deal with data acquired by different users. In contrast, egocentric video tends to be very personal and meaningful for the specific user who acquires it. We propose a method to segment egocentric video according to the personal locations visited by the user. The method aims at providing a personalized output and allows the user to specify which locations he wants to keep track of. To account for negative locations (i.e., locations not specified by the user), we propose a negative rejection method which does not require any negative sample at training time. For the experiments, we collected a dataset of egocentric videos in 10 different personal locations, plus various negative ones. Results show that the method is accurate and compares favorably with the state of the art.

*Keywords:* Egocentric Vision, Lifelogging, Personal Locations, Temporal Segmentation

---

## 1. Introduction

Wearable devices allow people to acquire a huge quantity of data about their behavior and activities in an automatic and continuous fashion [1]. The

---

\*Corresponding author

*Email addresses:* `furnari@dmi.unict.it` (Antonino Furnari), `battiato@dmi.unict.it` (Sebastiano Battiato), `gfarinella@dmi.unict.it` (Giovanni Maria Farinella)

practice of acquiring data of one’s own life for a variety of purposes is commonly  
5 referred to as lifelogging. While the technology to acquire and store lifelog data  
coming from different sources is already available, the real potential of such data  
depends on our ability to make sense of it. Wearable cameras, in particular, can  
be used to easily acquire hours of egocentric videos concerning the activities we  
perform, the people we meet, and the environments in which we spend our time.  
10 As observed in [2], egocentric video is generally difficult to exploit due to the lack  
of explicit structure, e.g., in the form of scene cuts or video chapters. Moreover,  
according to the considered goal, long egocentric videos tend to contain much  
uninformative content like, for instance, transiting through a corridor, walking  
outdoors or driving to the office. Consequently, automated tools to enable easy  
15 access to the information contained in such videos are necessary.

Toward this direction, researchers have already investigated methods to pro-  
duce short informative video summaries from long egocentric videos [3, 4, 5],  
recognize activities performed by the camera wearer [6, 7, 8, 9, 10, 11], tempo-  
rally segment the video according to detected ego-motion patterns [2, 12], and  
20 segment egocentric photo-streams [13, 14, 15]. Past literature aimed at investi-  
gating general-purpose methods, which are generally trained and tested on data  
acquired by many users in order to ensure the generality of the algorithms. This  
approach, however, risks to overlook the subjective nature of egocentric video,  
which can be leveraged to provide tailored and user-specific services.

### 25 1.1. *Personal Locations*

Towards the exploitation of user-specific information, in [16], we introduced  
the concept of *personal location* as:

*a fixed, distinguishable spatial environment in which the user can  
perform one or more activities which may or may not be specific to  
30 the considered location.*

Personal locations are defined at the instance level (e.g., my office, the lab),  
rather than at the category level (e.g., an office, a lab) and hence they should

not be confused with the general concept of visual scene [17]. Indeed, a given set of personal locations could include different instances of the same scene category (e.g., office vs lab office). Moreover, personal locations are user-specific since different users will be naturally interested in monitoring different personal locations (e.g., each user will be interest in monitoring the activities performed in his own office). Personal locations are constrained spaces (i.e., they are not defined as a whole room but rather refer to a part of it, e.g., the “office desk”), and hence they are naturally related to a restricted set of activities which can be performed in the considered locations [18]. For instance, the “office” personal location is naturally associated with office-related activities such as “writing e-mails” and “surfing the Internet”, while the “piano” personal location is generally related just to “playing piano”. Hence, being able to recognize when the user is located at a given personal location directly reveals information on a broad spectrum of activities which the user may be performing. The advantage of recognizing personal locations, rather than activities directly, is that providing supervision to recognize complex activities requires many samples (which is not practical for user-specific applications), while providing supervision to recognize personal locations is much more feasible, especially in egocentric settings [16].

### *1.2. Temporal Segmentation of Egocentric Video*

In this paper, we propose to segment egocentric videos into coherent segments related to personal locations specified by the user. We assume that the user selects a number of personal locations he wants to monitor and provides labeled training samples for them. The process of acquiring training data should not burden the user and be as simple as possible. Therefore, we adopt the acquisition protocol specified in [16, 19]. According to this protocol, the user acquires training data for a specific location by turning on his wearable device and looking around briefly to acquire a 30-seconds video of the environment.

At test time, the system analyzes the egocentric video acquired by the user and segments it into coherent shots related to the specified personal locations. Given the large variability of visual content generally acquired by wearable

devices, the user cannot easily provide an exhaustive set of personal locations he will visit. Therefore, the system should be able to correctly identify and reject all frames not related to any of the personal locations specified by the user. We will refer to these frames as “negatives” in the rest of the paper. In our context, negatives arise from two main sources: 1) the user moving from a personal location to another (*transition negatives*), and 2) the user spending time in a location which is not of interest (*negative locations*). Examples of transition negatives can be a corridor or an urban street, while examples of negatives locations might be a conference room, an office other than the user’s office, another car, etc. Please note that, while negative samples need to be correctly detected by the system, in real-world applications no negative training data can be provided by the user. Therefore, we design our method to learn solely from positive training data.

Figure 1 shows a scheme of the proposed temporal segmentation system and illustrates three possible applications for it, which are discussed in the following. The output of the algorithm is a temporal segmentation of the input video. Each segment is associated to a label which identifies the related personal location or whether it is a negative segment (i.e., it is not related to any user-specified personal location). Such output can be used for different purposes. The most straightforward objective consists in producing a video index to help the user browse the video. This way, the user can easily jump to the part of the video he is more interested in and discard negative segments which may not be relevant. A second possible use of the output temporal segmentation consists in producing coherent video shots related to the personal locations specified by the user (e.g., of egocentric videos acquired over different days). Given the segmented shots and related meta-data (e.g., time stamps), the system could answer questions such as “show me what I was doing this morning when I first entered my office” or “tell me how many coffees I had today” (e.g., how many times I was at the Coffee Vending Machine personal location). Moreover, video shots can be used as a basis for egocentric video summarization [4, 20]. A third use of the segmented video consists in estimating the time spent by the user at

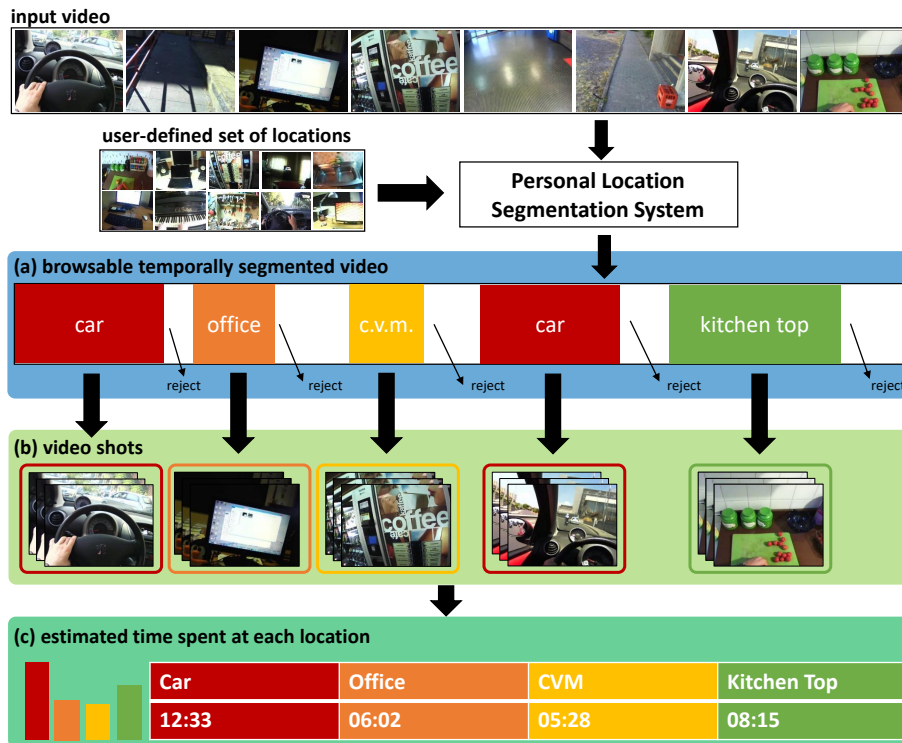


Figure 1: Scheme of the proposed temporal segmentation method. The system can be used to (a) produce a browsable temporally segmented egocentric video, (b) produce video shots related to given personal locations, (c) estimate the amount of time spent at each location.

each location. In this case, the system would be able to answer questions such  
 95 as “how much time did I spend driving this week?” or “how much time did  
 I spend in my office today?”. This kind of estimate does not require accurate  
 temporal segmentation but only overall correct per-frame predictions.

### 1.2.1. Contributions

This paper extends our previous work [21]. In particular, we present the  
 100 proposed method in greater details and analyzes the impact of each component  
 and related parameters more thoroughly. We extend the experimental analysis  
 by defining a novel performance measure designed to evaluate segmentation  
 accuracy from a shot-retrieval point of view. New comparisons with many state

of the art methods are also introduced. Finally, we publicly release the code  
105 implementing the proposed method and evaluation measures.

The main contributions of this paper can be summarized as follows: 1) It is proposed to segment egocentric videos to highlight personal locations using minimal user-specified training data. To study the problem we collect and release a dataset comprising more than 2 hours of labeled egocentric video covering 10  
110 different locations plus various negatives. 2) A method to segment egocentric videos and reject negative samples is proposed. The method can be trained using only the available positive samples. 3) A measure to evaluate the accuracy of temporal video segmentation methods is defined. The measure penalizes methods which produce over-segmented or under-segmented results.

115 Experiments show that the proposed system can produce accurate segmentations of the input video with little supervision, outperforming baselines and existing approaches. The code related to this study, as well as the proposed dataset and a video of our demo, can be downloaded at  
<http://iplab.dmi.unict.it/PersonalLocationSegmentation/>.

120 The remainder of the paper is organized as follows. Section 2 summarizes the related work. Section 3 presents the proposed method. Section 4 introduces the involved dataset, defines the considered evaluation measures and reports the experimental settings. Results are discussed in Section 5, whereas Section 6 concludes the paper.

## 125 **2. Related Works**

*Location Awareness.* Our work is related to previous studies on context and location awareness in wearable and mobile computing. According to Dey et al. [22], context aware systems should be able to “*use context to provide relevant information and/or services to the user, where relevancy depends on the*  
130 *user’s task*”. Visual location awareness, in particular, has been investigated by different authors over the years. Starner et al. [23] addressed the recognition of basic tasks and locations related to the Patrol game from egocentric videos in

order to assist the user during the game. Aoki et al. [24] proposed to recognize personal locations from egocentric video using the approaching trajectories observed by the wearable camera. Torralba et al. [25] designed a context-based vision system for place and scene recognition. Farinella et al. [26, 27] engineered efficient computational methods for scene recognition which can be easily deployed to embedded devices. Rhinehart et al. [18] explored the relationship between actions and locations to improve both localization and action prediction. Furnari et al. [16] performed a benchmark of different wearable devices and image representations for personal location recognition.

*Temporal Video Segmentation.* Temporal video segmentation methods aim at decomposing an input video into a set of meaningful segments which can be used as basic elements for indexing [28]. The topic has been widely investigated in the domain of movie and broadcast video [29, 30, 31, 32, 33]. In particular, Hanjalic et al. [29] proposed to consider a video as composed by scenes and shots. Shots are elementary video units acquired without interruption by a single camera. Scenes contain semantically coherent material and are generally composed by different temporally contiguous shots. Most state of the art algorithms achieve temporal segmentation by first detecting shots and then merging contiguous highly correlated shots to form scenes. Chasanis et al. [30] propose to cluster shots according to their visual content and apply a sequence alignment algorithm to obtain the final segmentation. Sidiropoulos et al. [31] jointly exploit low-level and high-level audiovisual features within the Scene Transition Graph to obtain temporal segmentation. Apostolidis and Mezaris [32] detect abrupt and gradual transitions in videos exploiting both local and global descriptors. Baraldi et al. [33] consider the problem of segmenting broadcast videos into scenes using hierarchical clustering.

It should be noted that the discussed classic temporal video segmentation methods are not directly applicable in the egocentric domain. In particular, the notions of shot and scene are not clearly defined for egocentric videos, which are generally acquired without interruption and by a single camera for the entire

length of the video.

*Motion-Based Egocentric Video Segmentation.* The problem of segmenting ego-  
centric video to introduce some kind of structure has already been investigated  
165 by researchers. Among the most prominent work is the one of Poleg et al. [2, 12],  
who proposed to segment egocentric video according to motion-related *long-term*  
*activities* such as “walking”, “standing” or “driving car” performed by the user.  
Similarly, Lu and Grauman [4] proposed to segment egocentric video into the  
170 three “static”, “moving the head” and “in transit” classes as a first step for  
egocentric video summarization. Alletto et al. [34] proposed to include features  
based on accelerometer and gyroscope data to improve motion-based segmenta-  
tion. Kitani et al. [35] presented an unsupervised method to segment egocentric  
video according to sports-related actions performed by the user. Motion-based  
175 features are also used by Su and Grauman [36] to detect engagement from ego-  
centric video, i.e, to identify the video segments in which the user is paying  
more attention.

*Visual-Content-Based Egocentric Video Segmentation.* While the aforementioned  
methods aim at segmenting egocentric video according to the perceived motion,  
180 they usually discard information strictly related to the visual content. In this  
regard, Lin and Hauptmann [37] imposed time constraints on the K-Means clus-  
tering algorithm to segment videos acquired using a wearable camera. Doherty  
and Smeaton [13] proposed a method to segment lifelog images acquired by a  
SensCam into events using color and edge features. Bolaños et al. [38] used  
185 hierarchical Agglomerative Clustering to segment egocentric photo-streams into  
events. Talavera et al. [14] combined clustering with a concept drift technique  
to improve segmentation results. Templeman et al [39], detected images of sen-  
sitive spaces for privacy purposes combining GPS information and an image  
classifier. Castro et al. [40] used Convolutional Neural Networks and Random  
190 Decision Forests to segment photo-streams of egocentric images highlighting hu-  
man activities. Paci et al. [41] presented a wearable system for context change  
detection based on an egocentric camera with ultra-low power consumption.



Ortis et al. [42, 43] proposed an unsupervised system to automatically divide egocentric videos into chapters with respect to the user’s context.

195 Past work focused on designing general-purpose methods which usually rely on data acquired by multiple users. In contrast, we consider a personalized scenario in which the user himself provides the training data and sets up the system. In such settings, it is not possible to rely on a big corpus of user-specific supervised data, since it is not feasible to ask the user to collect and label it. 200 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. Given the large variability of visual data acquired by wearable cameras, it is not feasible to ask the user to collect and label a large number of representative negative samples. Therefore, we design 205 our system to work without requiring any negative sample at training time.

### 3. Proposed Method

The proposed method aims at segmenting an input egocentric video into coherent segments. Each segment is related to one of the personal locations specified by the user or, if none of them apply, to the negative class. After an 210 off-line training procedure (which relies only on positive samples provided by the user), at test time, the system processes the input egocentric video. For each frame, the system should be able to 1) recognize the personal locations specified by the user, 2) reject negative samples, i.e., frames not belonging to any of the considered personal locations, and 3) group contiguous frames into 215 coherent video segments related to the specified personal locations. The method works in three steps, namely *discrimination*, *negative rejection* and *sequential modeling*:

1. *Discrimination*. Each frame is classified as one of the positive locations. No negative class is taken into account at this stage.
- 220 2. *Negative Rejection*. The system estimates the probability of each frame to be a negative by analyzing neighboring predictions. If predicted labels

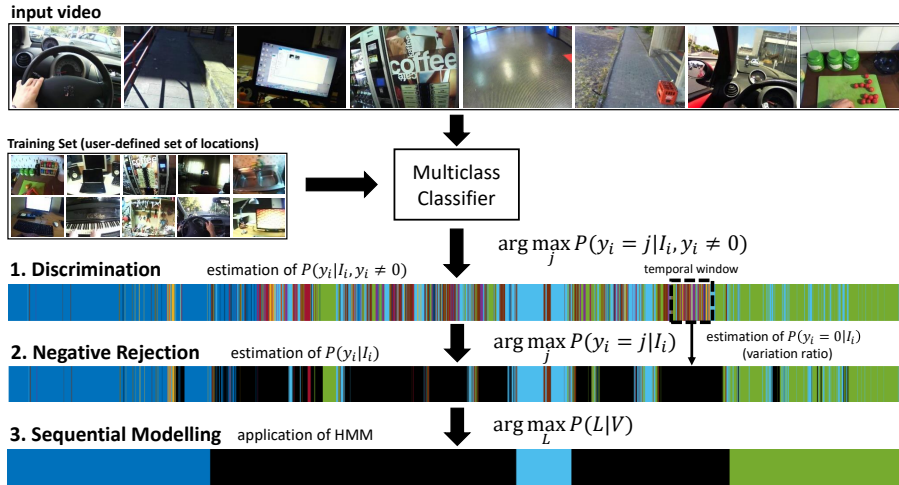


Figure 2: A scheme of the proposed temporal segmentation method. The method works in three steps: 1) *discrimination* among positive locations and estimation of  $P(y_i|I_i, y_i \neq 0)$ , 2) *negative rejection*, i.e., estimation of  $P(y_i = 0|I_i)$  and derivation of  $P(y_i|I_i)$ , 3) *sequential modeling* through a Hidden Markov Model and estimation of the final set of labels  $\mathcal{L}^* = \text{argmax}_{\mathcal{L}} P(\mathcal{L}|\mathcal{V})$ .

disagree, the sample is rejected by the system.

3. *Sequential Modeling*. Labels are predicted sequentially using a Hidden Markov Model to take into account previous observations. This step allows to obtain a more accurate segmentation where random label changes are discouraged.

225

Figure 2 shows a scheme of the proposed method. Each of the three steps involved in the proposed method is detailed in the following sub-sections.

### 3.1. Notation

Let  $\mathcal{V} = \{I_1, \dots, I_N\}$  be the input egocentric video, i.e., a sorted collection of  $N$  frames  $I_i$ . Let  $M$  be the number of personal locations specified by the user and let  $\{1, \dots, M\}$  be the set of class labels related to the personal locations. The system should assign a label  $y_i \in \{0, \dots, M\}$  to each frame  $I_i$ , where  $y_i = 0$  denotes the “negative class”. The final goal of the proposed system is to

230

235 produce a set of video segments  $\mathcal{S} = \{s_i\}_{1 \leq i \leq P}$  ( $P$  is the number of segments).  
 Each segment  $s_i$  contains a set of contiguous frames and is denoted by the  
 following triplet  $s_i = \{s_i^s, s_i^e, s_i^c\}$ , where  $1 \leq s_i^s \leq N$  is the index of the first  
 (starting) frame contained in the segment,  $1 \leq s_i^e \leq N$  is the index of the last  
 (ending) frame contained in the segment and  $s_i^c \in \{0, \dots, M\}$  is the class label  
 240 related to segment  $s_i$ . In practice, a given segment  $\bar{s} = (h, k, c)$  contains all  
 frames  $\{I_h, I_{h+1}, \dots, I_{k-1}, I_k\}$ . All labels related to the frames contained in the  
 segment will be equal to  $c$ , i.e.,  $y_l = c \forall l \in \{h, \dots, k\}$ . Moreover, the video  
 segments contained in  $\mathcal{S}$  define a *partition* of the video  $\mathcal{V}$ , i.e., each frame in  $\mathcal{V}$   
 belongs to exactly one segment in  $\mathcal{S}$ .

### 245 3.2. Discrimination

At training time, a multi-class classifier (e.g., a Convolutional Neural Net-  
 work) is trained on the positive data specified by the user to discriminate among  
 the  $M$  positive locations (i.e., to assign labels  $y_i \in \{1, \dots, M\}$ ). The negative  
 class is not considered at this stage because negative data is not assumed to be  
 250 available for training purposes. Since negatives are not included in the training  
 set, the multi-class classifier will not be suitable to estimate a posterior proba-  
 bility distribution over the  $M + 1$  classes (i.e., positive locations + the negative  
 class), such as the following:

$$P(y_i|I_i). \quad (1)$$

Rather, the multi-class classifier will allow estimate a posterior probability over  
 255 the  $M$  positive classes, i.e.:

$$P(y_i|I_i, y_i \neq 0), \text{ s.t. } \sum_{j=1}^M P(y_i = j|I_i, y_i \neq 0) = 1. \quad (2)$$

In order to recognize positive locations and reject negative ones, modeling the  
 probability distribution reported in Equation (1) is desirable. This can be done  
 by estimating the probability of frame  $I_i$  to be a negative  $P(y_i = 0|I_i)$  and  
 combining it with the discrimination probability reported in Equation (2). The

260 result is a posterior distribution over  $M + 1$  classes comprising both positive locations and the negative class.

### 3.3. Negative Rejection

Given the *continuous* nature of egocentric videos (i.e., they are acquired by a single camera without interruptions), transitions among different locations are expected to be *smooth*. Therefore, it is reasonable to assume that  $K$  neighboring frames in an egocentric video will belong to the same positive personal location or to some form of negative. This assumption can lead to imprecise results whenever the user transits from a personal location to another. However, such transitions are relatively rare in long egocentric videos and, for small enough values of  $K$ , such assumption shall not affect much the overall performance of the system.

Let  $\mathcal{I}_i^K = \{I_{i-\lfloor \frac{K}{2} \rfloor}, \dots, I_{i+\lfloor \frac{K}{2} \rfloor}\}$  be the neighborhood of size  $K$  of frame  $I_i$  and let  $\mathcal{Y}_i^K = \{y_{i-\lfloor \frac{K}{2} \rfloor}, \dots, y_{i+\lfloor \frac{K}{2} \rfloor}\}$  be the corresponding set of labels predicted with the Maximum A Posteriori (MAP) criterion, i.e.,  $y_i = \arg \max_j P(y_i = j | I_i, y_i \neq 0)$  after the discrimination step. According to the assumption above, if frame  $I_i$  belongs to a positive location, the labels associated to its neighborhood  $\mathcal{Y}_i^K$  are expected to “agree”, i.e., the distribution of labels in  $\mathcal{Y}_i^K$  should be strongly peaked. When the frames contained in  $\mathcal{I}_i^K$  are related to the negative class (i.e., they represent something unseen during training), the multi-class classifier will exhibit high uncertainty and it will likely pick a random label for each sample. In this case, the distribution of labels within  $\mathcal{Y}_i^K$  is expected to be characterized by large uncertainty, i.e., it should not exhibit a strong peak.

Following [44], we measure the model uncertainty by computing the variation ratio of the distribution of labels  $\mathcal{Y}_i^K$ . We hence define the probability of  $I_i$  to be a negative sample as follows:

$$P(y_i = 0 | I_i) = 1 - \frac{\sum_{k=i-\lfloor \frac{K}{2} \rfloor}^{i+\lfloor \frac{K}{2} \rfloor} [y_k = \text{mode}(\mathcal{Y}_i^K)]}{|\mathcal{Y}_i^K|} \quad (3)$$

where  $[\cdot]$  denotes the Iverson bracket,  $y_k \in \mathcal{Y}_i^K$ ,  $\text{mode}(\mathcal{Y}_i^K)$  is the statistical mode of  $\mathcal{Y}_i^K$  and  $|\mathcal{Y}_i^K|$  corresponds to the cardinality of set  $\mathcal{Y}_i^K$ .

Since the events  $y_i = 0$  ( $I_i$  is a negative) and  $y_i \neq 0$  ( $I_i$  is not a negative) are disjoint, the probability reported in Equation (1) can be obtained combining  
 290 the probabilities reported in Equation (2) and Equation (3) as follows:

$$P(y_i|I_i) = \begin{cases} P(y_i = 0|I_i) & \text{if } y_i = 0 \\ P(y_i \neq 0|I_i) \cdot P(y_i|I_i, y_i \neq 0) & \text{otherwise} \end{cases}. \quad (4)$$

The probability distribution reported in Equation (4) sums to one over the  $M+1$  classes  $\{0, \dots, M\}$  and can be used to jointly perform discrimination among the positive locations and rejection of negatives simply using the argmax function:

$$y_i = \arg \max_j P(y_i = j|I_i), \quad j \in \{0, \dots, M\}. \quad (5)$$

### 3.4. Sequential Modeling

295 The assumption according to which neighboring predictions shall be coherent can be further exploited by employing a Hidden Markov Model (HMM). Specifically, given the input video  $\mathcal{V} = \{I_1, \dots, I_N\}$ , the globally optimal set of labels  $\mathcal{L} = \{y_1, \dots, y_N\}$  can be obtained maximizing the posterior probability:

$$P(\mathcal{L}|\mathcal{V}). \quad (6)$$

According to Bayes' rule, such probability can be expressed as follows:

$$P(\mathcal{L}|\mathcal{V}) \propto P(\mathcal{V}|\mathcal{L})P(\mathcal{L}). \quad (7)$$

300 Assuming conditional independence of the frames with respect to each other given their class (i.e.,  $I_i \perp\!\!\!\perp I_j|y_i, \forall i, j \in \{1, 2, \dots, n\}, i \neq j$ ), and applying the Markovian assumption on the conditional probability distribution of class labels ( $P(y_i|y_{i-1} \dots y_1) = P(y_i|y_{i-1})$ ), Equation (7) is rewritten as:

$$P(\mathcal{L}|\mathcal{V}) \propto P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(I_i|y_i). \quad (8)$$

Term  $P(y_1)$  is assumed to be uniform over all possible classes (i.e., for  $y_1 \in$   
 305  $\{0, \dots, M\}$ ) and hence it can be ignored when Equation (7) is maximized with respect to  $\mathcal{V}$ . Probability  $P(I_i|y_i)$  is inverted using Bayes' law, thus obtaining:

$$P(I_i|y_i) \propto P(y_i|I_i)P(I_i). \quad (9)$$

Term  $P(I_i)$  can be ignored since  $I_i$  is observed when maximizing Equation (6) with respect to  $\mathcal{V}$  and term  $P(y_i|I_i)$  is computed directly using Equation (4). Equation (8) is finally written as follows:

$$P(\mathcal{L}|\mathcal{V}) \propto \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(y_i|I_i). \quad (10)$$

310 The HMM state transition term  $P(y_i|y_{i-1})$  represents the probability of transitioning from a given location to another (including negatives). Transition probabilities in Hidden Markov Models can be generally learned from data as done in [25], or defined ad-hoc to express a prior belief as done in [39]. Since we assume that few training data should be provided by the user and no labeled  
 315 sequences are available at training time, the HMM transition probability is defined ad-hoc to encode the prior belief that neighboring predictions are likely to belong to the same class [39]:

$$P(y_i|y_{i-1}) = \begin{cases} \varepsilon, & \text{if } y_i \neq y_{i-1} \\ 1 - M\varepsilon, & \text{otherwise} \end{cases} \quad (11)$$

where  $\varepsilon \leq \frac{1}{M+1}$  is a small constant. The transition probability defined in Equation (11) can be seen as an “almost identity” matrix, i.e., a matrix containing  
 320 values close to 1 on the main diagonal and positive values close to 0 anywhere else. This kind of transition matrix encourages coherence between subsequent states and penalizes multiple random state changes.

The set of globally optimal labels  $\mathcal{L}$  can be finally obtained maximizing the probability reported in Equation (7), which can be achieved efficiently using the  
 325 Viterbi algorithm [45]:

$$\mathcal{L} = \arg \max_{\mathcal{L}} P(\mathcal{L}|\mathcal{V}). \quad (12)$$

The final segmentation  $\mathcal{S}$  is obtained by considering the connected components of labels in  $\mathcal{L}$ .

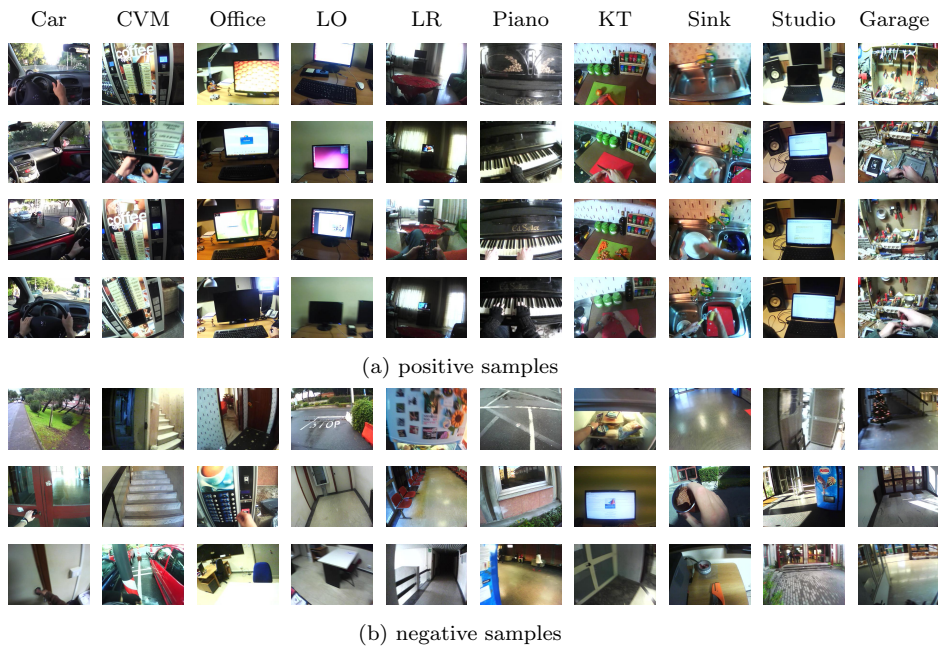


Figure 3: Some sample frames from the proposed dataset.

## 4. Experimental Settings

### 4.1. Dataset

330 We collected a dataset of egocentric videos in ten different personal locations, plus various negative ones. The considered personal locations arise from a possible daily routine: Car, Coffee Vending Machine (CVM), Office, Lab Office (LO), Living Room (LR), Piano, Kitchen Top (KT), Sink, Studio, Garage. The dataset has been acquired using a Looxcie LX2 camera equipped with a

335 wide angular converter. This configuration is the one which performed best in the benchmark dataset proposed in [16] and allows to acquire videos at a resolution of  $640 \times 480$  pixels and with a Field Of View of approximately  $100^\circ$ . The use of a wide-angular device is justified by the ability to acquire a large amount of scene information, albeit at the cost of radial distortion, which in some cases

340 requires dedicated computation [46, 47, 48]. Figure 3 shows some sample frames from the dataset.

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 T. → N → Sink → N → Piano	00:10:41
6	Kitchen T. → 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).

Since we assume that the user is required to provide only minimal data to define his personal locations of interest, the training set consists in 10 short videos (one per 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 personal locations or as a negative. Table 1 summarizes the content of the test sequences with an overview of the related transitions. It should be noted that test sequences contain both sources of negative samples discussed in Section 1.2, i.e., *transition negatives* and *negative locations*.

The dataset is also provided with an independent validation set which can be used to optimize the hyper-parameters of the compared methods. The validation set contains 10 medium length (approximately 5 to 10 minutes) videos in which the user performs some activities in the considered locations (one video per location). Validation videos have been temporally sub-sampled in order to



extract 200 frames per location, while all frames are considered in the case of training and test videos. We also acquired 10 medium length videos containing negative samples from which we uniformly extracted 300 frames for training and 200 frames for validation. Negative training and validation samples have been  
360 acquired in order to allow for comparisons with methods which require negative samples at training time. Please note that the proposed method does not need to learn from negatives and hence it discards them during training.

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

#### 4.2. Evaluation Measures

As observed in [49], evaluation measures for temporal video segmentation  
370 methods can be organized in three categories: *boundary-level measures*, *shot-level measures* and *frame-level measures*.

*Boundary-level* measures consider the segmentation problem as a shot boundary detection task. According to these measures, a prediction is considered correct only if the boundaries of the detected shot match ground truth boundaries  
375 exactly. This kind of similarity measures is not appropriate in our case since it is not clear how to define where shots/scenes begin and terminate in an egocentric video.

*Shot-level* measures evaluate temporal segmentation methods according to the overlap between predicted and ground truth segments. Among these, the  
380 popular coverage/overflow measure proposed by Vendrig et al. [50] evaluates if shots are correctly detected and grouped into scenes. While an overlap-based measure is needed to assess the accuracy of a produced segmentation, the coverage/overflow measure cannot be used directly in our case since the definitions of shots and scenes do not apply to egocentric videos.

*Frame-level* measures evaluate the fraction of frames which have been correctly  
385 labeled regardless their organization into coherent shots. The main draw-

back of such measure is that it does not explicitly penalize under-segmentation (i.e., when one or more segments are not detected) and over-segmentation (i.e., the incorrect detection of many small segments within a longer video segment).

390 However, despite their simplicity, this class of measures allows to assess how well a method can count the number of frames belonging to a given class. This can be useful, for example, to estimate the time spent at a given location over a long period of time (e.g., for lifelogging applications).

Taking into account the above considerations, we define two different mea-  
 395 sures which consider the temporal segmentation problem as a retrieval task and evaluate the methods in terms of  $F_1$  score. Specifically, we consider a *frame-based*  $F_1$  measure and a *segment-based*  $F_1$  measure. In both cases,  $F_1$  scores are computed separately for each class. Mean  $F_1$  scores (averaged over all classes) are also reported as an overall performance indicator for each method.

*Frame-based  $F_1$  score.* Given a specific class  $\gamma \in \{0, \dots, M\}$ , *precision* and *recall* values are computed in the standard way considering the number of frames correctly predicted as belonging class  $\gamma$ . The class-specific frame-based  $F_1$  score (denoted as  $FF_1^{(\gamma)}$ ) is hence computed as the harmonic mean between *precision* and *recall*:

$$FF_1^{(\gamma)} = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}. \quad (13)$$

400 As an overall performance indicator, we also consider the  $mFF_1$  score, which is the mean of the  $FF_1^{(\gamma)}$  scores related to all considered classes ( $\gamma \in \{0, \dots, M\}$ ). Per-class  $FF_1^{(\gamma)}$  scores (and related  $mFF_1$  values) are preferred over the standard accuracy measure (i.e., percentage of correctly classified frames) since they allow to perform unbiased evaluations when test samples are not evenly dis-  
 405 tributed among classes.

*Segment-based  $F_1$  score.* Let  $\gamma \in \{0, \dots, M\}$  be the considered class, let  $\bar{\mathcal{S}}_\gamma = \{\bar{s}_i \text{ s.t. } \bar{s}_i^c = \gamma\}_{i \in \{1, \dots, P\}}$  be the set of the  $P$  predicted segments belonging to class  $\gamma$ , and let  $\mathcal{S}_\gamma = \{s_i \text{ s.t. } s_i^c = \gamma\}_{i \in \{1, \dots, Q\}}$  be the set of the  $Q$  ground truth segments belonging to class  $\gamma$ . In order to reason about correct and

wrong predictions, each predicted segment  $\bar{s}_i$  should be associated to exactly one ground truth segment  $s_i$ . To compute such associations, we consider a standard linear sum assignment problem (which is solved using the Hungarian algorithm [28]) where the cost of assigning  $\bar{s}_i$  to  $s_j$  is equal to the *Jaccard distance* between the two segments  $d_J(\bar{s}_i, s_j)$ . The Jaccard distance  $d_J(\bar{s}_i, s_j)$  is obtained subtracting the *Jaccard coefficient* from 1:  $d_J(\bar{s}_i, s_j) = 1 - J(\bar{s}_i, s_j)$ , and the Jaccard coefficient  $J(\bar{s}_i, s_j)$  is computed as the ratio of the area of the intersection between the segments to the area of their union:

$$J(\bar{s}_i, s_j) = \frac{\max(\min(\bar{s}_i^e, s_j^e) - \max(\bar{s}_i^s, s_j^s) + 1, 0)}{\max(\bar{s}_i^e, s_j^e) - \min(\bar{s}_i^s, s_j^s) + 1}. \quad (14)$$

The solution of the linear assignment is the assignment matrix  $X = [x_{ij}]$ , where  $x_{ij} = 1$  if  $\bar{s}_i$  has been assigned to  $s_j$ . In order to compute *precision* and *recall* values, we consider a detected segment  $\bar{s}_i$  as a correct prediction only if  $x_{ij} = 1$  for some index  $j$  and the Jaccard index between the two segments exceeds a given threshold:  $J(\bar{s}_i, s_j) \geq t$ . This leads to the definition of threshold-dependent *precision* and *recall* measures:

$$precision^{(\gamma)}(t) = \frac{\sum_{i,j} x_{ij} \cdot [J(\bar{s}_i, s_j) \geq t]}{|\mathcal{S}_\gamma|}, recall^{(\gamma)}(t) = \frac{\sum_{i,j} x_{ij} \cdot [J(\bar{s}_i, s_j) \geq t]}{|\mathcal{S}_\gamma|}. \quad (15)$$

The threshold-dependent, *segment-based*  $F_1$  measure is hence computed as follows:

$$SF_1^{(\gamma)}(t) = 2 \cdot \frac{precision^{(\gamma)}(t) \cdot recall^{(\gamma)}(t)}{precision^{(\gamma)}(t) + recall^{(\gamma)}(t)}. \quad (16)$$

The  $SF_1^{(\gamma)}$  measure defined in Equation (16) can be used to plot threshold- $SF_1$  curves in order to assess the performances of the method with respect to varying tolerance levels. Given a set of thresholds  $\mathcal{T} = \{t \text{ s.t. } 0 \leq t \leq 1\}$ , the overall performance of a segmentation method can be computed as the average  $SF_1$  score:

$$ASF_1^{(\gamma)} = \frac{\sum_{t \in \mathcal{T}} SF_1^{(\gamma)}(t)}{|\mathcal{T}|}. \quad (17)$$

To assess the overall method performance, we also consider the  $mASF_1$  score, which is the average of  $ASF_1^{(\gamma)}$  scores for all considered classes ( $\gamma \in \{0, \dots, M\}$ ).

A Python implementation of the proposed measure is included in the code available at our web page:

410 <http://iplab.dmi.unict.it/PersonalLocationSegmentation/>.

### 4.3. Settings

All experiments are performed on the dataset described in Section 4.1. The multiclass classifier needed in the discrimination stage of our method is implemented by fine-tuning on our training set the VGG16 Convolutional Neural Network (CNN) pre-trained on the ImageNet dataset [51]. Given the small training set, the convolutional layers of the network are locked during the fine-tuning (i.e., their related learning rate is set to zero) to avoid overfitting. We set the neighborhood size of our rejection method to  $K = 300$  and the small constant in the definition of the HMM transition probability (Equation (11)) to  $\varepsilon = 2.23 \cdot 10^{-308}$ , which is the minimum positive normalized floating-point number in our machine. To compute  $SF_1$  measures, we set  $\mathcal{T} = \{0, 0.1, 0.2, \dots, 0.99, 1\}$ . The influence of the considered parameters on the performance of the method and the optimality of the selected values is discussed in Section 5.1. Compared methods are trained on the whole training set and evaluated on the test sequences. The validation set is used to tune the hyper-parameters of the methods and to select the best performing iteration in the case of CNNs.

## 5. Results

We perform experiments to 1) assess the influence of each of the components involved in the proposed method and related parameters and 2) compare the method with respect to the state of the art.

### 5.1. Performance of The Proposed Method

Fine-tuning large CNNs using a small training set ( $\approx 200$  samples per class in our settings) is not trivial and some architectural parameters can be tuned in order to optimize performance. We perform experiments to assess the impact of the following architectural settings: 1) the dataset on which the network has

Parameters	Discrimination		Rejection		Sequential Mod.	
	$mFF_1$	$mASF_1$	$mFF_1$	$mASF_1$	$mFF_1$	$mASF_1$
$\mathbb{I}$	0.92	0.05	<b>0.87</b>	<b>0.02</b>	0.92	0.82
$\mathbb{I} \mathbb{L}$	<b>0.94</b>	0.05	<b>0.87</b>	<b>0.02</b>	<b>0.95</b>	<b>0.89</b>
$\mathbb{I} \mathbb{ND}$	0.91	<b>0.06</b>	<b>0.87</b>	<b>0.02</b>	0.92	0.81
$\mathbb{I} \mathbb{L} \mathbb{ND}$	<b>0.94</b>	0.04	<b>0.87</b>	0.01	0.93	0.86
$\mathbb{P}$	0.92	0.05	0.84	<b>0.02</b>	0.89	0.79
$\mathbb{P} \mathbb{L}$	<b>0.94</b>	0.03	<b>0.87</b>	<b>0.02</b>	0.92	0.85
$\mathbb{P} \mathbb{ND}$	0.91	0.02	0.83	0.01	0.91	0.80
$\mathbb{P} \mathbb{L} \mathbb{ND}$	<b>0.94</b>	0.03	<b>0.87</b>	<b>0.02</b>	0.92	0.83

Table 2: Mean frame-based  $F_1$  scores ( $mFF_1$ ) and mean average segment-based  $F_1$  scores ( $mASF_1$ ) for the different components involved in the proposed method (i.e., *discrimination*, *rejection* and *sequential modeling*). Architectural settings:  $\mathbb{I}$ : the CNN has been pre-trained on the ImageNet dataset,  $\mathbb{P}$ : the CNN has been pre-trained on the Places365 dataset,  $\mathbb{L}$ : convolutional layers are locked,  $\mathbb{ND}$ : dropout is disabled.

been pre-trained (we consider ImageNet [52] and Places365 [53]), 2) whether the convolutional layers are “locked” (i.e., their related learning rate is set to zero) or not, 2) whether dropout in the fully connected layers is disabled or not.

Table 2 reports the performance (in terms of  $mFF_1$  and  $mASF_1$  scores) of  
440 the proposed method on the test sequences. Each row in Table 2 reports results for a specific experiment. For each experiment, the *Parameters* column summarizes the architectural settings used to fine-tune the CNN (see table caption for a legend), the *Discrimination* column reports the performance of the CNN alone (in this case negative samples are removed from the test set for the evaluation), the *Rejection* column reports the performance of the method including  
445 the proposed rejection mechanism but excluding the application of the HMM (all frames from test sequences are included in the evaluation), the *Sequential Mod.* column reports the final results of the proposed method including rejection and sequential modeling through the application of the HMM (all frames  
450 from test sequences are included in the evaluation).

It is worth observing that, discrimination (second column) is easier to achieve than rejection (third column) for all models. However, as discussed before, in real applications, the system needs be able to reliably reject negative samples. Interestingly, the gap between discrimination and rejection is in general successfully recovered in the sequential modeling component. Moreover, it should be noted how the use of a Hidden Markov Model with a hand-designed transition matrix is very effective to achieve consistent segmentation results. This is indicated by the poor  $mASF_1$  scores in the results related to both the discrimination and rejection steps, while sequential modeling results are significantly higher.

The results reported in Table 2 highlight the importance of tuning the considered architectural settings to improve accuracy. In particular, best results are systematically achieved when convolutional layers are locked. Disabling dropout leads to equivalent or marginally worse discrimination results. Models pre-trained on the ImageNet dataset allow to obtain better results over the ones pre-trained on Places365, especially in terms of  $mASF_1$  score when the final result of the Sequential modeling step is considered. While this finding might seem surprising at first (the Places365 dataset contains data which is closer to our applicative domain), many personal locations can be recognized considering the presence of specific objects (e.g., the computer monitor) as it is shown in Figure 3. In sum, best results are obtained pre-training the CNN on ImageNet and locking the convolutional layers (second row of Table 2). This architectural configuration is the one used in all following experiments.

Figure 4 reports color-coded segmentation results of the proposed method for qualitative assessment. The figure illustrates predictions issued by the three components of the proposed method (*Discrimination, Rejection, Sequential Modeling*). As can be noted from Figure 4, the discrimination component tends to exhibit high uncertainty in the negative segments. At the discrimination stage, this results in areas characterized random label changes. The rejection component leverages the presence of such uncertain segments to detect negatives but still retains some of the original random label changes. The

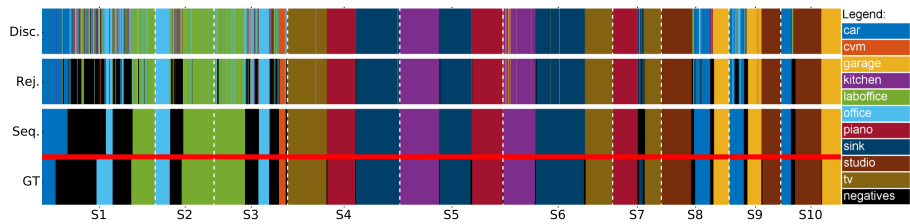


Figure 4: Color-coded segmentation results for qualitative assessment. The diagram illustrates predictions made by the three components of the proposed method, i.e., *Discrimination* (Disc.), *Rejection* (Rej.) and *Sequential Modeling* (Seq.). Ground truth segmentation is also reported for comparison. Please note that the diagram reports results for the concatenation of all sequences in the test set.

application of a HMM in the sequential modeling stage allows to obtain a clean segmentation which often matches the ground truth with high accuracy.

For further qualitative assessment, in Table 3 we report results related to the  
 485 task of estimating the total amount of time spent by the user at each location. Specifically, the table reports ground truth times as well as estimates performed by the proposed method. Estimates are obtained by counting the number of predicted frames related to each class. As can be observed from Table 3, estimated times are very accurate for many classes (error is often in the order of  
 490 seconds).

The reader is also referred to the demo video available at our web page: <http://iplab.dmi.unict.it/PersonalLocationSegmentation/>.

### 5.1.1. Sensitivity with Respect to the Involved Parameters

We also study the sensitivity of the results with respect to the two involved  
 495 parameters, namely the size  $K$  of the neighborhood considered for negative rejection and the small constant  $\varepsilon$  used to define the HMM transition matrix. Figure 5 reports the results of the proposed method on the test sequences for varying values of the parameters  $K$  and  $\varepsilon$ . To assess sensitivity to parameter  $K$ , in Figure 5(a),  $\varepsilon$  is set to the optimal value of  $\varepsilon = 2.23 \cdot 10^{-308}$ . Similarly,  
 500 in Figure 5(b), we set  $K = 300$ .

Time spent at location (MM:SS)			
Class	Ground Truth	Estimated	Difference
Car	04:53	06:03	+01:10
Coffee V. Machine	00:35	00:37	+00:02
Garage	04:29	04:26	-00:03
Kitchen Top	06:32	06:37	+00:05
Lab Office	07:58	07:43	-00:15
Office	03:48	03:03	-00:45
Piano	07:39	07:46	+00:05
Sink	11:40	11:37	-00:03
Studio	06:49	06:55	+00:06
Living Room	07:39	07:41	+00:02
Negative	11:20	10:54	-01:34

Table 3: Ground truth times spent at each specific location by the user, along with the times estimated by the proposed system and the difference between the two values. All times are related to the union of all sequences in the test set.

Parameter  $K$  should be chosen in order to incorporate enough observations to perform rejection, while avoiding the noise due to excessively large neighborhoods. As can be observed in Figure 5(a), best segmentation results (indicated by a green star) are reached around  $K = 300$  (approximately 10 seconds). It should be noted that the method is robust also to other values of  $K$ .

Results reported in Figure 5(b) suggest that the HMM works best for very small values of  $\varepsilon$ . In this case indeed, the transition matrix related to the probability defined in (11) is an “almost identical” matrix, which allows to strongly enforce temporal coherence among neighboring predictions.

As discussed in Section 4.3, we set  $K = 300$  and  $\varepsilon = 2.23 \cdot 10^{-308}$ .

## 5.2. Comparison With the State Of The Art

We compare our method with respect to the following baselines and state of the art methods.



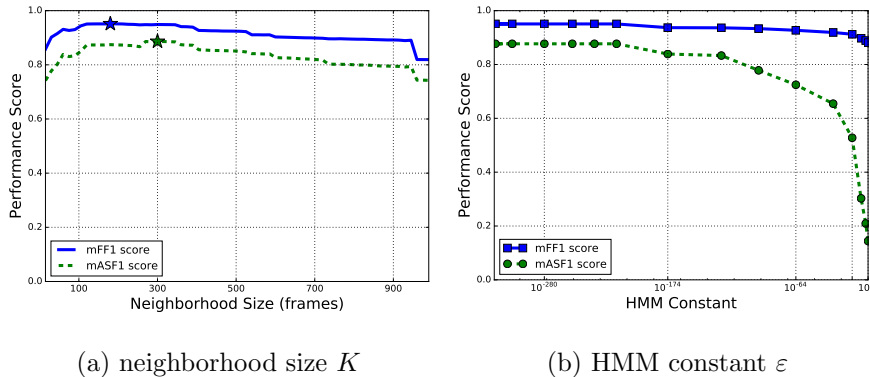


Figure 5: Sensitivity of the proposed method with respect to the two involved parameters (a)  $K$  (size of the neighborhood for negative rejection) and (b)  $\epsilon$  (small constant in the HMM transition matrix).

*SIFT-Based Matching (SIFT)*. The first baseline tackles the location recognition problem through feature matching. The system is initialized extracting SIFT feature points from each training image and storing them for later use. Given the current test 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 based on an affine model [54]. Classification is hence performed selecting the class of the training set image presenting the highest number of inliers. In this case, the most straightforward way to perform rejection of negative samples 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 geometrically weak matches will be obtained. Since it is not clear how such a threshold should be arbitrarily set, we learn it from the data. To do so, we first normalize the number of inliers by the number of features extracted from the current frame. We then 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, keeping the original aspect ratio (a pre-processing similar to the one required by most CNN

models).

*Open Set Deep Networks (OSDN)*. This method is based on the Open Set Deep Networks recently proposed in [55]. We apply the OpenMax algorithm described  
535 in [55] to the same CNN used by the proposed method in order to obtain a model able to perform both classification and rejection of negative samples. Similarly to the proposed method, this method does not require any negative sample at training time. A HMM is applied to the output of the network to allow for fair comparisons.

540 *Cascade SVM Classifier (CSVM)*. The method proposed in [19] performs negative rejection and personal location recognition using a cascade of a One-Class and a multiclass SVM classifier. The classifiers are trained on features extracted using the VGG16 network pre-trained on ImageNet. Please note that this method uses training negatives to optimize the hyper-parameters of the  
545 One-Class SVM classifier. Also in this case, a HMM is used to enforce temporal coherence.

*Entropy-Based Rejection (EBR)*. This is the method recently proposed in [16], which performs rejection of negative samples by measuring the entropy of the posterior probability over small sequences of neighboring frames. The rejection  
550 method is applied to the output of the same CNN used by the proposed method. A HMM is used to obtain the final segmentation.

*Negative-Trained Network (NTN)*. This baseline employs a CNN trained to discriminate directly between locations of interest and negative samples. The network can be used directly to estimate posterior probabilities over 11 classes.  
555 In contrast with all other compared methods, this baseline explicitly learns from negative samples. The CNN has been fine-tuned following the same architectural settings adopted by our method (i.e., network pre-trained on ImageNet and locked convolutional layers). A HMM is applied to the output of the network to obtain the final segmentation.

Method	$mFF_1$	Car	CVM	Garage	KT	LO	Office	Piano	Sink	Studio	LR	N
SIFT	0.27	0.03	0.08	0.00	0.83	0.04	0.13	0.81	0.10	0.14	0.47	0.33
OSDN [55]	0.60	0.41	0.93	0.81	0.00	0.52	0.08	<b>1.00</b>	0.60	<b>1.00</b>	0.83	0.42
CSVM [19]	0.79	0.41	0.87	0.80	0.94	0.97	0.84	0.84	0.90	0.80	0.94	0.32
EBR [16]	0.86	0.74	0.93	0.95	0.92	0.62	0.76	0.99	0.97	0.99	<b>0.99</b>	0.57
NTN	0.92	0.94	0.79	<b>0.99</b>	<b>0.99</b>	0.74	<b>0.91</b>	0.99	<b>0.98</b>	0.93	<b>0.99</b>	0.70
Proposed	<b>0.95</b>	<b>0.89</b>	<b>0.97</b>	0.98	<b>0.99</b>	<b>0.98</b>	0.85	0.99	0.97	0.99	<b>0.99</b>	<b>0.83</b>

Table 4: Per-class  $FF_1$  scores and related  $mFF_1$  measures for all compared methods.

Method	$mASF_1$	Car	CVM	Garage	KT	LO	Office	Piano	Sink	Studio	LR	N
SIFT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
OSDN [55]	0.47	0.40	0.87	0.54	0.00	0.32	0.08	<b>0.99</b>	0.16	<b>0.99</b>	0.66	0.17
CSVM [19]	0.49	0.19	0.35	0.44	0.65	0.95	0.57	0.43	0.51	0.48	0.67	0.13
EBR [16]	0.58	0.36	0.86	0.62	0.54	0.07	0.30	0.84	0.68	0.98	0.97	0.22
NTN	0.84	<b>0.91</b>	0.93	<b>0.98</b>	<b>0.99</b>	0.49	<b>0.79</b>	<b>0.99</b>	<b>0.89</b>	0.83	<b>0.98</b>	0.49
Proposed	<b>0.89</b>	0.87	<b>0.94</b>	0.96	<b>0.99</b>	<b>0.96</b>	0.76	0.98	0.83	0.98	<b>0.98</b>	<b>0.54</b>

Table 5: Per-class  $ASF_1$  scores and related  $mASF_1$  measures for all compared methods.

560 Table 4 and 5 report the results of the compared methods in terms of  $FF_1$  and  $ASF_1$  scores respectively. For each method, we report detailed per-class scores (including the negative class), as well as the overall  $mFF_1$  and  $mASF_1$  scores. Methods are sorted in terms of ascending  $mFF_1$  and  $mASF_1$  scores and best per-column scores are highlighted in bold. In Figure 6 we also report 565 the Threshold- $SF_1$  curves related to the compared methods. Finally, Figure 7 reports color-coded segmentation results of all compared methods for qualitative assessment.

The proposed method achieves the best performance in terms of overall  $mFF_1$  and  $mASF_1$  scores, as well as in terms of many per-class  $FF_1$  and  $ASF_1$  570 scores. In particular, the proposed method scores the best results when it comes to rejecting negative samples in terms of both  $FF_1$  and  $ASF_1$  scores. As it is shown in Figure 6, the proposed method is the most accurate for all considered levels of segmentation matching tolerance (i.e., for different thresholds  $t$ ). Moreover, the segmentation produced by the proposed method is the most accurate

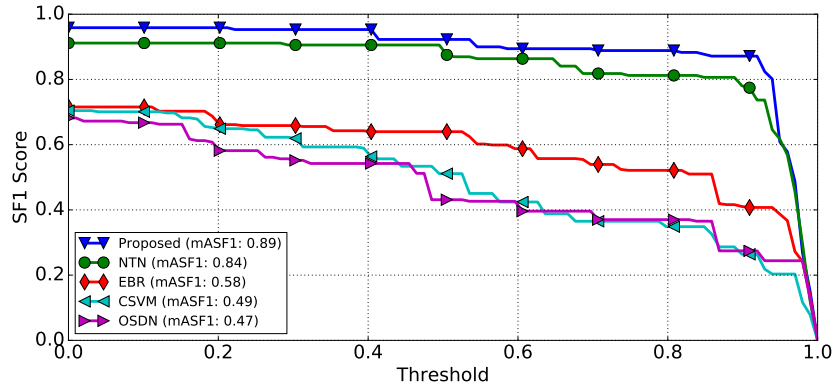


Figure 6: Threshold- $SF_1$  curves comparing the proposed method with respect to the state of the art. Reported curves are averaged over all classes.

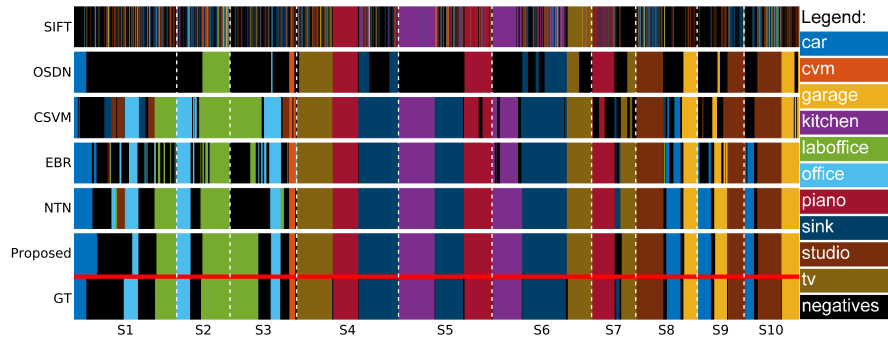


Figure 7: Color-coded segmentation results for qualitative assessment. The diagram compares the proposed method with respect to temporal video segmentation approaches.

575 among the others, as it is shown in Figure 7. The baseline based on matching SIFT features achieves the worst performance in terms of both  $FF_1$  and  $ASF_1$  scores. Decent  $FF_1$  results are reached only for some distinctive scenes like for instance *Kitchen Top* and *Piano*. The method achieves very low  $ASF_1$  results since it tends to over-segment the video as it is shown in Figure 7. Open Set  
580 Deep Networks (OSDN) [55] achieves very good results for the *Piano* and *Studio* classes but is unable to manage other classes (e.g., *Kitchen Top* and *Office* and *Car*) in terms of both the considered scores. In particular, the method tends to reject more samples than it should, as it is shown in Figure 7. The method based on a cascade of SVM classifiers (CSVM) [19] achieves better performance  
585 but falls short in negative sample rejection. In particular, as shown in Figure 7, the methods omits many of the rejections and exhibits some false rejections. The Entropy-Based (EB) method [16] improves negative rejection by a good margin but it is not always accurate in terms of  $mASF_1$  scores, since it tends to over-segment the video (see Figure 7). The proposed method also outperforms  
590 the NTN baseline, which is designed to learn explicitly from negative samples. It should be noted that user-specific training negatives are hard to acquire and hence might not be available, which makes our method (which does not rely on any negative samples at training time) preferable in real applications.

### 5.2.1. Comparison With Temporal Video Segmentation Methods

595 As discussed in Section 2, our method is related to previous work on temporal video segmentation. Nevertheless, a direct comparison with those methods is not straightforward since they have been designed to produce a different output. Specifically, while our method produces a set of segments  $s_i$  characterized by a starting index  $s_i^s$ , an ending index  $s_i^e$  and a class label  $s_i^c$ , classic temporal video  
600 segmentation methods are designed to break the input video into shots which are not associated to any specific class labels. We perform experiments to investigate whether the output of such algorithms can be used to improve personal-location-based segmentation of egocentric videos. To this aim, we designed a simple baseline which combines a classic temporal segmentation method with a personal

Method	$mASF_1$	Car	CVM	Garage	KT	LO	Office	Piano	Sink	Studio	LR	N
Apostolidis et al. [32]	0.71	0.87	0.63	0.77	0.85	0.75	<b>0.76</b>	0.70	0.58	0.81	0.71	0.39
Poleg et al. [2]	0.84	0.87	0.84	0.92	0.95	0.86	0.67	0.97	0.82	0.95	0.97	0.42
Baraldi et al. [49]	0.86	0.85	0.87	<b>0.97</b>	0.98	0.78	<b>0.76</b>	<b>0.98</b>	<b>0.83</b>	0.98	<b>0.98</b>	0.50
Ortis et al. [43]	0.87	<b>0.88</b>	0.76	0.95	<b>0.99</b>	<b>0.97</b>	<b>0.76</b>	0.97	0.81	<b>0.99</b>	<b>0.98</b>	0.51
Proposed	<b>0.89</b>	0.87	<b>0.94</b>	0.96	<b>0.99</b>	0.96	<b>0.76</b>	<b>0.98</b>	<b>0.83</b>	0.98	<b>0.98</b>	<b>0.54</b>

Table 6: Per-class  $ASF_1$  scores and related  $mASF_1$  measures for experiments comparing the proposed method to baselines building on different temporal video segmentation approaches.

605 location classifier  $\mathcal{C}$  capable of assigning to each frame a label corresponding to one of the  $M + 1$  personal locations (including negatives).

The baseline works as follows. Let  $\mathcal{S}' = \{s_i\}_{1 \leq i \leq P}$  be the set of video shots produced by the considered video temporal segmentation method. Each segment  $s_i$  is characterized by a starting index  $s_i^s$  and by an ending index  $s_i^e$ .  
610 The baseline assigns class labels  $s_i^c$  to shots  $s_i$  by performing majority voting on the labels  $\mathcal{Y}' = \{y_{s_i^s}, \dots, y_{s_i^e}\}$  predicted by the considered classifier  $\mathcal{C}$ , i.e.,  $s_i^c = mode(\mathcal{Y}')$ . Adjacent shots belonging to the same class are merged into a single video segment. Intuitively, if boundaries of the detected shots match the ones of ground truth segments, the baseline should help to remove some  
615 over-segmentation errors and improving overall segmentation accuracy.

We implement the proposed baseline considering four different temporal segmentation methods. Two of them have been explicitly designed to temporally segment egocentric videos. Specifically, the method by Poleg et al. [2] segments egocentric video detecting long term activities (e.g., walking, standing, running,  
620 etc.) according to the exhibited egocentric motion, while the method by Ortis et al. [43] relies on visual content represented through CNN features to segment egocentric video into coherent scenes. The last two methods have been respectively proposed by Apostolidis et al. [32] and Baraldi et al. [33] and represent the state of the art in temporal segmentation of movies and broadcast video.  
625 Since the proposed method is the best performing one among all competitors, we employ it as the classifier  $\mathcal{C}$  required by the proposed baselines.

Table 6, Figure 8 and Figure 9 compare the proposed method with the baselines related to the four considered segmentation methods. Results suggest

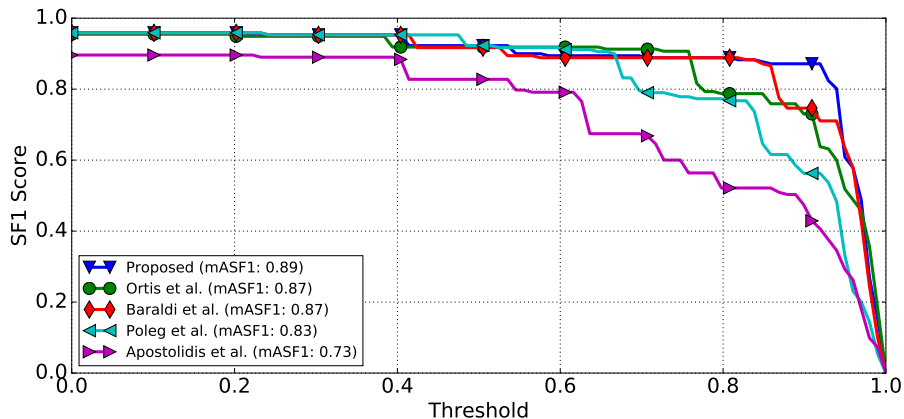


Figure 8: Threshold- $SF_1$  curves comparing the proposed method with respect to baselines building on different temporal video segmentation approaches. Reported curves are averaged over all classes.

that the proposed method already allows to achieve accurate segmentations and hence it does not benefit from fusion with other temporal segmentation methods. Specifically, as it is shown in Table 6, while the baselines obtain marginally better results for a few classes, they do not improve over the proposed method in terms of overall  $mASF_1$  score. In fact, temporal segmentations obtained using the baselines are in general less precise than the ones obtained using the proposed method. This can be observed quantitatively in Figure 8, where the proposed method dominates the others for high thresholds (i.e., when a more accurate segmentation is required) and qualitatively in Figure 9.

## 6. Conclusion

We have proposed a method to segment egocentric video into coherent segments related to personal locations specified by the user. The method works in supervised settings and requires minimal user-provided training data. Differently from previous works, our method explicitly considers the problem of rejecting negative locations and does not require any negative sample at training time. Moreover, we show how a simple Hidden Markov Model can be used

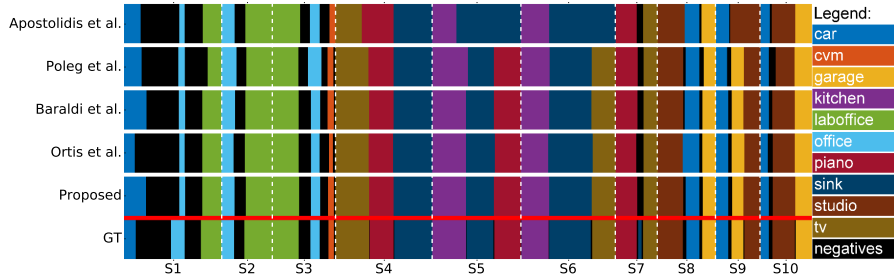


Figure 9: Color-coded segmentation results for qualitative assessment. The diagram compares the proposed method with respect to baselines building on different temporal video segmentation approaches.

645 to obtain consistent temporal segmentation. The output of our algorithm can be used for a number of applications related to life-logging, such as, egocentric video indexing, detection of semantically relevant video shots for later retrieval or summarization, and estimation of the amount of time spent at each specific location. The experimental analysis have highlighted that the proposed method  
650 is accurate and compares favorably with respect to the state of the art.

Given the unavailability of larger publicly available datasets, the proposed experimental analysis has been carried out on a limited set of data. Future work will be devoted to the extension of the analysis to a larger dataset, acquired by multiple users, and to further reducing the extent of required user-intervention  
655 by improving negative rejection methods.

### Acknowledgment

This research is supported by PON MISE - Horizon 2020, VEDI Project, Prog. n. F/050457/02/X32 - CUP: B68I17000800008 - COR: 128032. We gratefully acknowledge the support of NVIDIA Corporation with the donation  
660 of the Titan X Pascal GPU used for this research.



## References

- [1] C. Gurrin, A. F. Smeaton, A. R. Doherty, et al., Lifelogging: Personal big data, *Foundations and Trends® in Information Retrieval* 8 (1) (2014) 1–125.
- 665 [2] Y. Poleg, C. Arora, S. Peleg, Temporal segmentation of egocentric videos, in: *Computer Vision and Pattern Recognition*, 2014, pp. 2537–2544.
- [3] K. Aizawa, K. Ishijima, M. Shiina, Summarizing wearable video, in: *International Conference on Image Processing*, Vol. 3, 2001, pp. 398–401.
- [4] Z. Lu, K. Grauman, Story-driven summarization for egocentric video, in: 670 *Computer Vision and Pattern Recognition*, 2013, pp. 2714–2721.
- [5] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, V. Singh, Gaze-enabled egocentric video summarization via constrained submodular maximization, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2235–2244.
- 675 [6] E. H. Spriggs, F. De La Torre, M. Hebert, Temporal segmentation and activity classification from first-person sensing, in: *Computer Vision and Pattern Recognition Workshops*, 2009, pp. 17–24.
- [7] A. Fathi, A. Farhadi, J. M. Rehg, Understanding egocentric activities, *IEEE International Conference on Computer Vision* (2011) 407–414.
- 680 [8] H. Pirsiavash, D. Ramanan, Detecting Activities of Daily Living in First-person Camera Views, in: *International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2847–2854.
- [9] Y. Li, Z. Ye, J. M. Rehg, Delving into egocentric actions, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 287–295.
- 685 [10] M. Wray, D. Moltisanti, W. Mayol-Cuevas, D. Damen, Sembed: Semantic embedding of egocentric action videos, in: *European Conference on Computer Vision*, Springer, 2016, pp. 532–545.

- [11] M. Ma, H. Fan, K. M. Kitani, Going Deeper into First-Person Activity Recognition, in: International Conference on Computer Vision and Pattern Recognition, 2016, pp. 1894–1903.
- 690 [12] Y. Poleg, A. Ephrat, S. Peleg, C. Arora, Compact cnn for indexing egocentric videos, in: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, IEEE, 2016, pp. 1–9.
- [13] A. R. Doherty, A. F. Smeaton, Automatically segmenting lifelog data into events, in: Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on, 2008, pp. 20–23.
- 695 [14] E. Talavera, M. Dimiccoli, M. Bolanos, M. Aghaei, P. Radeva, R-clustering for egocentric video segmentation, in: Iberian Conference on Pattern Recognition and Image Analysis, Springer, 2015, pp. 327–336.
- [15] M. Dimiccoli, M. Bolaos, E. Talavera, M. Aghaei, S. G. Nikolov, P. Radeva, Sr-clustering: Semantic regularized clustering for egocentric photo streams segmentation, *Computer Vision and Image Understanding* 155 (2017) 55 – 69.
- 700 [16] A. Furnari, G. M. Farinella, S. Battiato, Recognizing personal locations from egocentric videos, *IEEE Transactions on Human-Machine Systems* 47 (1) (2017) 6–18.
- 705 [17] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, *International Journal of Computer Vision* 42 (3) (2001) 145–175.
- [18] N. Rhinehart, K. M. Kitani, Learning action maps of large environments via first-person vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 580–588.
- 710 [19] A. Furnari, G. M. Farinella, S. Battiato, Recognizing personal contexts from egocentric images, in: Proceedings of the IEEE International Conference on Computer Vision Workshops (ACVR), 2015, pp. 1–9.
- 715

- [20] P. Varini, G. Serra, R. Cucchiara, Personalized egocentric video summarization for cultural experience, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, ACM, 2015, pp. 539–542.
- [21] A. Furnari, G. M. Farinella, S. Battiato, Temporal segmentation of egocentric videos to highlight personal locations of interest, in: European Conference on Computer Vision Workshops (EPIC), Vol. 9913 LNCS, Springer, 2016, pp. 474–489.
- [22] A. K. Dey, Understanding and using context, Personal and ubiquitous computing 5 (1) (2001) 4–7.
- [23] T. Starner, B. Schiele, A. Pentland, Visual contextual awareness in wearable computing, in: International Symposium on Wearable Computing, 1998, pp. 50–57.
- [24] H. Aoki, B. Schiele, A. Pentland, Recognizing personal location from video, in: Workshop on Perceptual User Interfaces, 1998, pp. 79–82.
- [25] A. Torralba, K. P. Murphy, W. T. Freeman, M. Rubin, Context-based vision system for place and object recognition, in: IEEE International Conference on Computer Vision, 2003, pp. 273–280.
- [26] S. Battiato, G. Farinella, G. Gallo, D. Ravì, Exploiting textons distributions on spatial hierarchy for scene classification, EURASIP Journal on Image and Video Processing 2010 (1) (2010) 919367.
- [27] G. M. Farinella, D. Ravì, V. Tomaselli, M. Guarnera, S. Battiato, Representing scenes for real-time context classification on mobile devices, Pattern Recognition 48 (4) (2015) 1086–1100.
- [28] I. Koprinska, S. Carrato, Temporal video segmentation: A survey, Signal processing: Image communication 16 (5) (2001) 477–500.
- [29] A. Hanjalic, R. L. Lagendijk, J. Biemond, Automated high-level movie segmentation for advanced video-retrieval systems, IEEE Transactions on Circuits and Systems for Video Technology 9 (4) (1999) 580–588.

- [30] V. T. Chasanis, A. C. Likas, N. P. Galatsanos, Scene detection in videos  
745 using shot clustering and sequence alignment, *IEEE transactions on mul-  
timedia* 11 (1) (2009) 89–100.
- [31] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho,  
I. Trancoso, Temporal video segmentation to scenes using high-level au-  
750 diovisual features, *IEEE Transactions on Circuits and Systems for Video  
Technology* 21 (8) (2011) 1163–1177.
- [32] E. Apostolidis, V. Mezaris, Fast shot segmentation combining global and  
local visual descriptors, in: *Acoustics, Speech and Signal Processing  
(ICASSP), 2014 IEEE International Conference on*, IEEE, 2014, pp. 6583–  
6587.
- 755 [33] L. Baraldi, C. Grana, R. Cucchiara, Shot and scene detection via hierarchi-  
cal clustering for re-using broadcast video, in: *International Conference on  
Computer Analysis of Images and Patterns*, Springer, 2015, pp. 801–811.
- [34] S. Alletto, G. Serra, R. Cucchiara, Motion segmentation using visual and  
760 bio-mechanical features, in: *Proceedings of the 2016 ACM on Multimedia  
Conference*, ACM, 2016, pp. 476–480.
- [35] K. M. Kitani, T. Okabe, Y. Sato, A. Sugimoto, Fast unsupervised ego-  
action learning for first-person sports videos, in: *IEEE Conference on Com-  
puter Vision and Pattern Recognition*, 2011, pp. 3241–3248.
- [36] Y.-C. Su, K. Grauman, Detecting engagement in egocentric video, in: *Eu-  
765 ropean Conference on Computer Vision*, Springer, 2016, pp. 454–471.
- [37] W.-H. Lin, A. Hauptmann, Structuring continuous video recordings of ev-  
eryday life using time-constrained clustering, in: *Electronic Imaging 2006,  
International Society for Optics and Photonics*, 2006, pp. 60730D–60730D.
- 770 [38] M. Bolanos, R. Mestre, E. Talavera, X. Giró-i Nieto, P. Radeva, Visual  
summary of egocentric photostreams by representative keyframes, in: *Mul-*

timedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on, IEEE, 2015, pp. 1–6.

- 775 [39] R. Templeman, M. Korayem, D. Crandall, K. Apu, PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces, in: Annual Network and Distributed System Security Symposium, 2014, pp. 23–26.
- [40] D. Castro, S. Hickson, V. Bettadapura, E. Thomaz, G. Abowd, H. Christensen, I. Essa, Predicting daily activities from egocentric images using deep learning, in: proceedings of the 2015 ACM International symposium on Wearable Computers, ACM, 2015, pp. 75–82.
- 780 [41] F. Paci, L. Baraldi, G. Serra, R. Cucchiara, L. Benini, Context change detection for an ultra-low power low-resolution ego-vision imager, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9913 LNCS (2016) 589–602.
- 785 [42] A. Ortis, G. M. Farinella, V. D’amico, L. Addesso, G. Torrioni, S. Battiato, Refusion: Automatic video curation driven by visual content popularity, in: Proceedings of the 23rd ACM international conference on Multimedia, ACM, 2015, pp. 1179–1182.
- [43] A. Ortis, G. M. Farinella, V. D’Amico, L. Addesso, G. Torrioni, S. Battiato, 790 Organizing egocentric videos of daily living activities, Pattern Recognition 72 (Supplement C) (2017) 207 – 218. doi:<https://doi.org/10.1016/j.patcog.2017.07.010>.  
URL <http://iplab.dmi.unict.it/dailylivingactivities>
- [44] Y. Gal, Z. Ghahramani, Dropout as a Bayesian Approximation: Represent- 795 ing Model Uncertainty in Deep Learning, arXiv preprint arXiv:1506.02142.
- [45] C. M. Bishop, Pattern recognition and Machine Learning, Springer, 2006.

- [46] A. Furnari, G. M. Farinella, G. Puglisi, A. R. Bruna, S. Battiato, Affine region detectors on the fisheye domain, in: International Conference on Image Processing, 2014, pp. 5681–5685.
- 800 [47] A. Furnari, G. M. Farinella, A. R. Bruna, S. Battiato, Distortion adaptive sobel filters for the gradient estimation of wide angle images, Journal of Visual Communication and Image Representation 46 (2017) 165 – 175. doi:10.1016/j.jvcir.2017.03.019.
- [48] A. Furnari, G. M. Farinella, A. R. Bruna, S. Battiato, Affine covariant  
805 features for fisheye distortion local modeling, IEEE Transactions on Image Processing 26 (2) (2017) 696–710.
- [49] L. Baraldi, C. Grana, R. Cucchiara, Measuring scene detection performance, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9117  
810 (2015) 395–403.
- [50] J. Vendrig, M. Worring, Systematic evaluation of logical story unit segmentation, IEEE Transactions on Multimedia 4 (4) (2002) 492–499.
- [51] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
- 815 [52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248–255.
- [53] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, A. Oliva, Places: An image  
820 database for deep scene understanding, arXiv preprint arXiv:1610.02055.
- [54] P. Torr, A. Zisserman, Mlesac: A new robust estimator with application to estimating image geometry, Computer Vision and Image Understanding 78 (1) (2000) 138–156.

- [55] A. Bendale, T. E. Boult, Towards open set deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1563–1572.