Chapter 6
Object Detection-Based Location and Activity Classification from Egocentric Videos: A Systematic Analysis

Georgios Kapidis, Ronald Poppe, Elsbeth van Dam, Lucas P. J. J. Noldus and Remco C. Veltkamp

Abstract Egocentric vision has emerged in the daily practice of application domains such as lifelogging, activity monitoring, robot navigation and the analysis of social interactions. Plenty of research focuses on location detection and activity recognition, with applications in the area of Ambient Assisted Living. The basis of this work is the idea that indoor locations and daily activities can be characterized by the presence of specific objects. Objects can be obtained either from laborious human annotations or automatically, using vision-based detectors. We perform a study regarding the use of object detections as input for location and activity classification and analyze the influence of various detection parameters. We compare our detections against manually provided object labels and show that location classification is affected by detection quality and quantity. Utilization of the temporal structure in object detections mitigates the consequences of noisy ones. Moreover, we determine that the recognition of activities is related to the presence of specific objects and that the lack of explicit associations between certain activities and objects hurts classification performance for these activities. Finally, we discuss the outcomes of each task and our method's potential for real-world applications.

Keywords Egocentric vision · Object detection · Location classification · Activity classification · Detection quality · Temporal associations

G. Kapidis (✉) · E. van Dam · L. P. J. J. Noldus
Noldus Information Technology, Wageningen, The Netherlands
e-mail: georgios.kapidis@noldus.nl; g.kapidis@uu.nl

G. Kapidis · R. Poppe · R. C. Veltkamp
Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

F. Chen et al. (eds.), *Smart Assisted Living*, Computer Communications and Networks, https://doi.org/10.1007/978-3-030-25590-9_6

6.1 Introduction

Egocentric vision is an essential part of computer vision with applications in conventional fields such as activity recognition [2] and video summarization [3] as well as in more elaborate, for instance social interaction analysis [4], guideline generation for visual assistance [5] and infant visual attention [6]. In this work, we focus on indoor location and activity detection from egocentric videos, with typical applications in Ambient Assisted Living (AAL) [7]. An example can be non-intrusive status updates to healthcare professionals about the locations and actions of people suffering from limited vision or dementia. Activities of daily living are also of interest when it comes to patient rehabilitation after a serious illness. Normally, this process would take place in a protected environment, far from the person's home. The possibility of continuous and real-time monitoring offered by egocentric cameras allows for noninvasive and personalized care. Reusability of the equipment by other patients at the end of the recovery period is an additional incentive toward adoption by nursing homes. Moreover, enhancement with intelligent detection mechanisms will promote privacy, since only information relevant to the rehabilitation will need to be communicated to third parties and not the actual video stream. An example use-case is that of dementia patients who require constant monitoring and professional care [8]. Egocentric vision is able to provide the indoor location [1], the duration of physical exercises [9] or the performed activity, upon request or in a continuous mode.

The use of egocentric cameras is alluring as they are becoming smaller and less intrusive, two essential qualities for wearables targeting everyday use. They can be used as an alternative to expensive multi-sensor installations that convert an existing house into a smart-home. By taking advantage of recent advances in machine learning, a single sensor—the egocentric camera—will produce information about the location in the house, sociability or loneliness, performed activity and even imminent dangers stemming from the latter.

To produce an inference on an image or video frame, one could calculate image-descriptive features [10, 11] stack them in vectors and classify, using machine learning models in a supervised fashion. In recent years, feature extraction and classification have merged into end-to-end deep networks, providing promising results. In this work, we take a step back and consider a different type of input.

Our key idea is to use the detected objects in a video frame as cues to recognize the indoor location or an ongoing activity. Initially, we build on the idea that rooms can be characterized by the presence of specific, distinctive objects. This consistency can be translated into associations between objects and locations. Consider, for example, Fig. 6.1 (left) which shows the detected objects of an egocentric video segment from a kitchen. If we categorize the objects based on their mobility, we may group them into (a) those that can be thought of as *movable*, but bear meaning for understanding the scene, such as the soap, the mug and the dish and (b) those that are *unmovable* but (i) distinctive to this particular location, such as the stove, the microwave or the fridge and (ii) those that can be found in multiple locations, for example the tap, which could also appear in a bathroom. Similarly, we claim that the activity of the

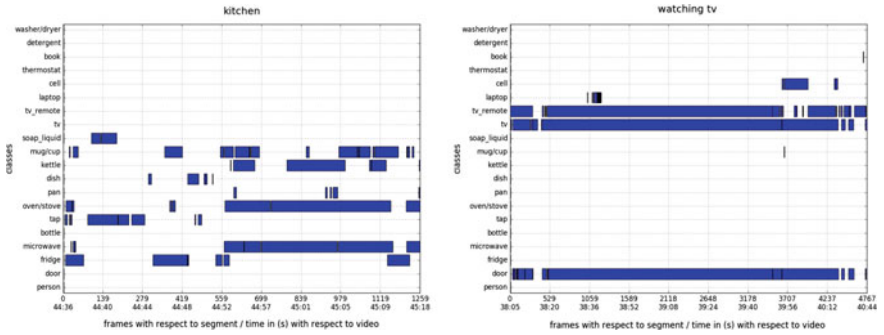

Fig. 6.1 Detected objects for 'kitchen' (left). Certain objects, such as the oven and the microwave, dominate the scene. For activity 'watching TV' (right), the television and the remote controller are indicative of the performed activity

egocentric protagonist can be inferred from the detected objects, considering Fig. 6.1 (right) which shows a TV and a remote controller for most of the duration of a video segment for activity 'watching TV.'

This motivates us to perform an analysis on the videos of the Activities of Daily Living (ADL) dataset [12] to discover associations between objects and locations and objects and activities. For the object-location associations, we train classifiers with artificial neural networks (ANNs) and long short-term memory (LSTM) networks [13] to experiment with per frame classification and utilization of the temporal structure of the data, respectively. Conceptually, an individual frame of a scene might include only partial information about the objects, as not all that are detectable may fit in view. However, the combination of multiple frames over time can encode a more complete view of the room. Eventually, we compare the performance of classifiers from both types of models, trained either on object labels or detections, from detectors trained on object categories from different datasets.

For the object-activity associations, we rely on detections enhanced with certain appearance features. Apart from the presence of objects, we measure the bounding box sizes and positions in the frame. We aim to investigate whether this additional information modifies the status of an object as participating in the ongoing activity, for example, when observed from a distance (smaller) or at the edge of the view. Figure 6.2 outlines our approaches.

The *contributions* of this work[1] can be summarized to:

- The development of a method to analyze object associations toward (1) locations and (2) activities in egocentric videos,
- The object presence feature, which despite its simplicity demonstrates acceptable performance,

[1]Code and data for our experiments are located here: https://github.com/georkap/object-based-location-classification.

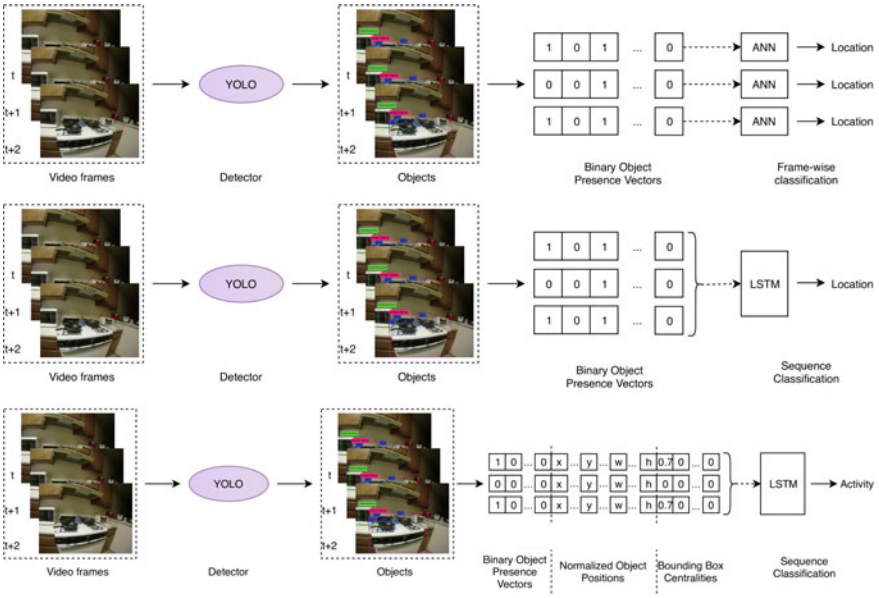

Fig. 6.2 Our pipelines for location and activity recognition. We extract objects from video frames using an object detector (YOLO [14]). The detections for a frame are turned into binary object presence vectors. This representation is the input for classification with ANNs (top) which produce one location per frame. The presence vectors are stacked in sequences and used as input to LSTMs (middle) which generate one location prediction for the complete sequence. For activity classification (bottom), we consider two additional features that describe the bounding boxes and classify them with LSTM

- The description of location classification results for diverse object sets and detection thresholds with and without temporal information,
- The demonstration that laborious object annotations are not required for location classification, given that our system performs equally well using only automatic detections and
- The analysis of object-activity associations in the context of daily living and the effect that object sizes and positions have in the activity recognition results.

Section 6.2 is an overview of related work in egocentric object, location and activity recognition, datasets and applications in the field. In Sect. 6.3, we describe the dataset we used, the object detectors and the methodology for both tasks. In Sect. 6.4, we present our results. Our findings are discussed in Sect. 6.5, and Sect. 6.6 concludes this chapter.

6.2 Related Work

6.2.1 Objects and Location Classification

We focus on recognizing indoor locations based on the detected objects from egocentric videos of people moving freely in their homes. The ADL dataset [12] has these characteristics and the required object annotations. The scientific literature provides a plethora of egocentric datasets [15–21]. The datasets of [15, 16] are created with the aim of detecting locations and observing indoor and outdoor everyday scenes. The dataset of [21] focuses on activities that take place either indoors or outdoors, such as walking, running and sitting, whereas [17] is enhanced with accelerometer and heart rate data to infer the level of sedentariness in performed activities. The dataset of [18] includes annotations and segmentations of the important objects that characterize the activities, with a large number of the videos being outdoors. Datasets from [19, 20] consist of videos in a kitchen, in which the participants are asked to prepare food according to predefined recipes.

Understanding of locations, in terms of mapping the surrounding area with image features or semantically labeling the environment, is actively under research in egocentric vision. In [22], a combination of scene illumination and distinct location characteristics is learned in an unsupervised way, in order to enhance the usability of wearable cameras for hand detection. Location recognition is indirectly the task in [23] where a Google Glass application captures images of the user's field of view and retrieves information about the buildings in sight. An indoor localization system is considered in [24] where the combination of a camera and a 2D laser scanner is applied to register query images from users into a real-world coordinate system. A multi-view indoor localization system based on image features is proposed in [25]. It distinguishes the indoor locations by computing self-similarity matrices from the extracted images to correlate the various captured views of the scene. Afterward, it learns the equation system through these features and when a query image is given, it can provide its location and orientation. The combination of wearable egocentric stereo cameras and inertial sensors is considered in [26] to map an outdoor workspace and provide routing guidance for executing specific tasks in the workplace. Another system for location classification is described in [27]. Visual recognition is based on low-level features and semi-supervised training procedures to take advantage of sparsely annotated available data. Instead, we use high-level features, the object detections from every frame. We do not depend on previous knowledge of the specific locations that the users find themselves into but build our inference upon characteristic objects that are detected in them.

Temporal segmentation of egocentric videos is considered in [15] to highlight personal locations of interest. Training is based on user-provided frame samples of locations labeling those capturing user interest as positive. Then, the system learns to reject the frames that do not depict locations relevant to the user. Personal locations are also analyzed in [16] as part of a user's daily routine. Classification of the video frames into locations relies on either convolutional neural network (CNN) features

or handcrafted ones. In [28, 29], the combined improvement of object detection and scene identification is investigated. Initially, scene identification is performed based on temporally associated CNN features and its output is used to improve the results of object detectors by linking the objects to specific locations. Eventually, they show that by using an LSTM to train on the temporal sequences of the detected objects directly, it is possible to improve the performance of detectors without explicitly using extracted features about locations. They perform their experiments on the ADL dataset. Our work is the opposite of this concept, where we use the object detections to infer either locations or activities.

Searching through the parameters of a model to find the optimal configuration is a common theme in machine learning [30–32]. In [30], various LSTM variants are tested on speech recognition, handwriting recognition and music modeling tasks to inspect the differences introduced by the changing network architectures. It is shown that most LSTM variants do not improve significantly, if at all, over the default LSTM structure, which performs relatively well for all considered tasks. Variations in the hyperparameters used for training are also explored, but it is observed that they are uncorrelated. In an effort to provide further insights into the reasons behind the effectiveness of LSTM, [31] provide a thorough study on cell activations, error analysis and data representations. A more recent approach to tuning a model's hyperparameters is presented in [32] where effort is put into balancing regularization for a particular dataset and the respective architecture utilized for training, by studying the loss during both training and testing for signs of over- and underfitting. Our aim is not to compare the parameters that define the structure of networks, such as the number of layers or neurons per layer. Rather, we vary the model from ANN to LSTM to augment the learning process with temporal information. In the context of searching for the optimal parametrization for location classification, our work comprises a large-scale search over the dataset combinations for training and testing the classifiers, the object detectors that generate the detection datasets and the effect of the detection confidence in detection quality and quantity.

6.2.2 Objects and Activity Classification

Human action recognition from video is a computer vision task that introduces multiple challenges to researchers, ranging from the innate difficulties of video analysis, such as illumination changes and motion blur, to the adversities of activity recognition, including class variability, viewpoint variation and scarcity of training data [33].

In egocentric activity recognition, the person is removed from view; therefore, the inference must rely on indirect cues. These include the detection of hands, relevant objects and locations [2, 34], as well as convolutional features [35], image features [36] or motion-based features, such as ego-motion and global motion or combinations [16, 20, 36–38]. In [34], the hands are used for activity recognition. An egocentric hand detector is described, where region proposals are combined with

convolutional neural networks (CNNs) for hand classification. The correct instances are turned into pixel-level segmentations, which in turn are the input in a second CNN classifier to infer the performed activity. In [35], objects and activities are analyzed in close relation, with each affecting the decision for the other interchangeably. Objects in association with hands and the interactions between them are modeled to infer activities and their associations in [38]. We take a more elementary approach and study only the objects and their characteristics for activities, without feedback between modalities.

Motion-based works usually rely on optical flow as input to train machine learning models. In [37], each frame is divided into a set of grid cells and a single sparse optical flow value is extracted for each cell per frame. The flow values are used as train data to a CNN and classified into specific activity classes. In [36], scene, object, hand and head movements are modeled with dense trajectories, color histograms and local binary patterns and used as inputs to support vector machines for the classification of the food-related activities of the GTEA datasets [20]. The aim is to measure the individual effect of new features on activity classification performance by adding them gradually, based on the idea that their contributions are complementary. We also follow this concept of feature aggregation.

Combination of detections along with motion cues for the description of activities is not uncommon. In [2], multiple networks are utilized for the extraction of hand poses and object segmentations per frame. At the same time, a different network is trained on optical flow to predict short-term actions. Eventually, all networks are combined toward a new output, the activity prediction. Our method is different in that the object detection module is fully detached from the activity prediction and can be thought of as a different component that could be substituted with a more efficient one, when it becomes available. In [39], videos are mapped onto a semantic graph, with nodes for each freely annotated object and action, trained on the visual similarities between them. The activities in unseen videos are recognized as probability distributions among the existing action labels. They demonstrate the inherent interactions that occur between objects and actions, an idea that relates to our work.

Activity recognition solely based on the detected objects in the scene is considered in [40, 41]. In [40], object detection relies on video input, complemented with RFID tags, manually placed in the scene. Our work only considers video. In [41], a dynamic feature prioritization policy is developed to choose which single-class object detectors to promote, in an effort to execute as few of them as possible, thus saving computations, while also maximizing the classification accuracy on the subsequent frame. The aim is to take advantage of the spatiotemporal correlations that occur during an activity and avoid extraction of unnecessary features, which would not provide important information for the recognition process as a whole. Our method does not focus on single objects and their possible associations through time during detection, but takes advantage of the state of the art in single-frame, multi-class object detection to extract objects in real time and uses LSTM to learn the temporal associations.

6.3 Methodology

In this section, we analyze an egocentric video dataset in terms of objects, locations and activities (Sect. 6.3.1) and select the object detection framework to facilitate our tests (Sect. 6.3.2). Moreover, we study and discuss the parameters of the location (Sect. 6.3.3) and activity (Sect. 6.3.4) classification tasks.

6.3.1 Activities of Daily Living (ADL) Dataset

The ADL dataset [12] consists of 20 videos of people performing activities occurring indoors, captured from the egocentric perspective. Each video is a record of the subject's choice of activities from a predefined set, performed in an unscripted manner. In every video, the subject is different and operates in their own house, thus providing considerable variations in locations and activities, among videos. In total, there are approximately 10 h of egocentric videos, equivalent to more than one million frames. The videos are annotated with activity labels with start/end times, object bounding boxes, object tracks and human–object interactions. Train and test splits are provided by the authors; videos 1–6 are considered training data, and the remaining 14 comprise the test set. For our experiments, we use the same splits.

Originally, in [12], there are annotations for 48 object classes, but due to the low number of either training or testing samples, only 42 are considered for their tests. For our object detectors, we use either the whole set of 48 classes or a subset of 20. We elaborate on object detection in Sect. 6.3.2. A list of the object classes together with their occurrences in the ADL dataset appears in Table 6.1.

In Sect. 6.3.3, we are interested in the analysis of locations, so we extend the dataset with the location annotations from [29]. For every 30 frames of video, one location class out of the eight possible is annotated, namely kitchen, bedroom, bathroom, living room, laundry room, corridor, outdoor and undefined (Table 6.2). Class 'undefined' occurs in blurred frames or non-identifiable locations. We do not use these frames for training and testing the location classification models. Hence, the location classes are seven in our experiments.

In Sect. 6.3.4, we focus on activity classification and make use of the existing activity annotations in the ADL dataset. We transform the labels from describing video segments with specific start and end times, to one activity per frame. The activities are shown in Table 6.3. In [12], only 18 activities are considered whereas the dataset contains labels for 33. We consider all 33 activities in our experiments.

Table 6.1 Forty-eight object classes of the ADL dataset and the number of occurrences per class. In the third column the instances in the train set. In bold, the classes of the ADL20 subset

Class name	Total	Train
Person	4650	2424
Door	7903	2019
Fridge	1999	301
Microwave	2369	527
Bottle	10,310	1705
Tap	7826	3252
Oven/stove	3196	1007
Pan	3156	1026
Trash can	2075	486
Dish	8216	2274
Cloth	3077	78
Knife/spoon/fork	4843	1893
Food/snack	3876	741
Kettle	1239	464
Mug/cup	11,050	2766
Soap liquid	8375	2658
Pills	394	148
Basket	1588	35
Towel	4480	1961
Toothbrush	1795	819
Toothpaste	1746	492
Electric keys	1570	417
TV	5600	2033
Remote	2813	1253
Container	5685	3821
Shoes	3248	735
Tea bag	359	177
Laptop	7027	2183
Cell phone	653	271
Cell	571	238
Thermostat	332	137
Book	4770	445
Dental floss	547	385
Vacuum	519	116
Electric keys 2	118	118
Pitcher	1208	277
Detergent	1105	297

(continued)

Table 6.1 (continued)

Class name	Total	Train
Washer/dryer	3362	954
Bed	783	228
Large container	558	6
Monitor	316	287
Keyboard	107	102
Shoe	694	300
Blanket	85	31
Comb	307	51
Perfume	550	0
Milk/juice	366	0
Mop	403	0

Table 6.2 Sampled frames per location. Class 'undefined' is not used for training and testing

Location	Kitchen	Bedroom	Bathroom	Living room	Laundry room	Corridor	Outdoor	Undefined
Train set	3414	1821	2307	2606	815	45	143	492
Test set	6850	3966	2285	5045	1097	133	906	737
Total	10,264	2285	4592	7651	1912	178	1049	1229

Table 6.3 Thirty-two activity classes in the ADL dataset, plus the background class (35,906, 85,801). In parentheses the number of train and test frames per class, respectively

1: Combing hair (3539, 6267)	2: Makeup (8363, 3926)	3: Brushing teeth (27,729, 26,117)	4: Dental floss (8543, 2127)
5: Washing hands/face (15,050, 17,270)	6: Drying hands/face (4014, 6743)	7: Entering/leaving room (0, 0)	8: Adjusting thermostat (1110, 2459)
9: Laundry (28,812, 46,101)	10: Washing dishes (21,249, 45,807)	11: Moving dishes (9984, 0)	12: Making tea (15,679, 27,265)
13: Making coffee (6774, 18,974)	14: Drinking water/bottle (6565, 12,328)	15: Drinking water/tap (0, 540)	16: Making hot food (8872, 38,619)
17: Making cold food/snack (14,268, 11,546)	18: Eating food/snack (6686, 32,180)	19: Mopping in kitchen (1020, 8933)	20: Vacuuming (3657, 9864)
21: Taking pills (3237, 4409)	22: Watching TV (37,769, 78,086)	23: Using computer (20,445, 57,125)	24: Using cell (5817, 10,435)
25: Making bed (0, 6055)	26: Cleaning house (11,360, 12,655)	27: Reading book (20,350, 18,016)	28: Using mouth wash (420, 570)
29: Writing (0, 3628)	30: Putting on shoes (5668, 450)	31: Drinking coffee/tea (15,226, 33,778)	32: Grabbing tap water (599, 1170)

6.3.2 Object Detection

For our object detection experiments, we use the Darknet framework.[2] Our detector is YOLOv2 [14, 42], a real-time object detection system that can operate on input images of various sizes. YOLOv2 is based on the Darknet-19 architecture [14] and consists of 19 convolutional and 5 max-pooling layers. It is pretrained on ImageNet [43] for 1000 classes, for 160 epochs. From this pretrained model, we develop three separate detectors, one for every object dataset we consider.

Our first YOLOv2-based detector is fine-tuned on the 80 classes of the MS COCO dataset [44], and the weights are provided by the authors of [14]. We call this detector 'COCO' for short. We train two additional models with this architecture for the object classes of the ADL dataset: (1) 'ADL48', on all the classes in Table 6.1 and (2) 'ADL20', on the 20 in bold. The selection of classes for 'ADL20' follows [29], where they select only classes for which their detector achieves more than 5% average precision (AP).

The reason for the diversification of detectors is that MS COCO and ADL consist of different sets of classes. ADL comprises objects found in homes (Table 6.1), whereas MS COCO is more generic in its categories (Table 6.4). The split between 'ADL20' and 'ADL48' is an attempt to produce a detector focused on classes with more samples in the training dataset, thus excluding harder to detect classes. We expect this to reduce the classification loss during training and lead to an improved bounding box classifier for the subset.

For both 'ADL20' and 'ADL48,' we fine-tune the ImageNet weights for 35 k iterations (i.e., batches). During training, we vary the input dimensions of the detectors to learn objects of various sizes. Training hyperparameters are the same as in [14]. The 'ADL20' detector achieves 29.84% mean average precision (mAP) and the 'ADL48' 11.15%. In Table 6.5, we report the average precision per class for our detectors. They suggest that YOLOv2 creates a more successful detector for the majority of object classes of the ADL dataset than fast R-CNN [45] in [29].

6.3.3 Locations

We model the relationship between the objects in a frame or a series of frames to recognize the location. Applying object detection on the videos of the ADL dataset leads to a binary presence vector (BPV) of zeros and ones, for every video frame, with length equal to the number of output classes of a detector, i.e., 80 for 'COCO,' 48 for 'ADL48' and 20 for 'ADL20.' In BPV, we only consider whether an object exists in a scene or not, regardless of the times it may be found. We also experimented with keeping the counts of multiple detections of the same object in a frame using a multiple presence vector (MPV), but without consistent improvements. Location

[2] https://pjreddie.com/darknet/.

Table 6.4 MS COCO [44] object classes

Person	Bicycle	Car	Motorcycle	Airplane	Bus	Train	Truck	Boat	Traffic light
Fire hydrant	Stop sign	Parking meter	Bench	Bird	Cat	Dog	Horse	Sheep	Cow
Elephant	Bear	Zebra	Giraffe	Backpack	umbrella	Handbag	Tie	Suitcase	Frisbee
Skis	Snowboard	sports ball	Kite	Baseball bat	Baseball glove	Skateboard	surfboard	Tennis racket	Bottle
Wineglass	Cup	Fork	Knife	Spoon	Bowl	Banana	Apple	Sandwich	Orange
Broccoli	Carrot	Hot dog	Pizza	Donut	Cake	Chair	Sofa	Potted plant	Bed
Dining table	Toilet	TV	Laptop	Mouse	Remote	Keyboard	Cell phone	Microwave	Oven
Toaster	Sink	Refrigerator	Book	Clock	Vase	Scissors	Teddy bear	Hair drier	Toothbrush

Table 6.5 Average precision (%) of ADL20/48 object detectors per class, trained with YOLOv2. Certain classes are particularly challenging. In bold the classes that improve in the reduced dataset. Comparison with [29] using fast R-CNN for the 20-class subset of the ADL dataset

Object classes [12]	ADL20	ADL48	[29]
Person	**69.0**	59.49	25.74
Door	**23.2**	17.72	5.59
Fridge	**22.75**	12.85	24.95
Microwave	**37.8**	24.81	32.35
Bottle	**10.02**	4.59	***11.28***
Tap	**59.27**	51.18	39.55
Oven/stove	**44.48**	28.15	43.02
Pan	**16.88**	12.46	10.99
Trash can	–	9.61	
Dish	**14.21**	6.22	11.19
Cloth	–	4.55	
Knife/spoon/fork	–	4.8	
Food/snack	–	9.65	
Kettle	**22.54**	8.68	***23.83***
Mug/cup	**15.92**	15.69	13.24
Soap liquid	21.76	**31.59**	18.77
Pills	–	0.14	
Basket	–	0.0	
Towel	–	9.38	
Toothbrush	–	9.78	
Toothpaste	–	11.67	
Electric keys	–	0.73	
TV	**52.07**	49.57	***57.58***
Remote	**52.49**	30.36	32.88
Container	–	5.25	
Shoes	–	0.72	
Tea bag	–	0.68	
Laptop	**44.4**	41.04	37.46
Cell phone	–	10.91	8.65
Cell	**0.89**	0.65	
Thermostat	**24.88**	3.89	9.01
Book	16.39	**18.04**	12.83
Dental floss	–	0.92	
Vacuum	–	0.66	
Electric keys 2	–	0.0	
Pitcher	–	3.13	
Detergent	**9.9**	9.76	9.13

(continued)

Table 6.5 (continued)

Object classes [12]	ADL20	ADL48	[29]
Washer/dryer	**37.96**	25.58	***38.86***
Bed	–	0.21	
Large container	–	0.0	
Monitor	–	0.0	
Keyboard	–	0.0	
Shoe	–	0.15	
Blanket	–	0.0	
Comb	–	0.1	
Perfume	–	0.0	
Milk/juice	–	0.0	
Mop	–	0.0	
Total (mAP)	29.84	11.15	

labels exist once every 30 frames (1 s) [29], and only these frames are used for classification, without augmentation for the ones in between.

We train two types of classifiers. The first type is based on fully connected neural network architectures (artificial neural networks—ANN) that have as input one vector per sample. The second type is based on LSTMs to examine the temporal structure of the data, which we train on stacked sequences of vectors.

For both ANN and LSTM classifiers, we parametrize our experiments with respect to the object datasets. These are categorized based on:

- The **dataset combinations** for training and evaluating the classifier,
- The **object detector classes** and
- The **object detection thresholds**.

We categorize as such after considering our objective, i.e., to assess whether an object detector can be used as the first step in an indoor location recognition pipeline. In this context, we experiment with using either object annotations or detections to model the locations. At test time, we compare against object detections in order to compare the modeling capabilities offered from both train sets. Hence, the dataset combinations comprise the scenarios that affect the composition of a location classifier's dataset. We consider *Labels to Detections* (L2D) which use the object annotations for training and the detections for testing and *Detections to Detections* (D2D) that contain only detections for both splits. For comparison, we also consider *Labels to Labels* (L2L) which consist of the object annotations for both splits; i.e., the object detections are *not* used.

The object detector variations were discussed in Sect. 6.3.2. Using this as a parameter means that we vary the object detector that produces the object dataset. As a result, different object classes are learned. This, in turn, leads to generating object vectors (BPV) of different lengths.

The detection threshold creates a trade-off between the confidence and the number of detections. Higher thresholds lead to more confident but fewer detections. Lower thresholds provide more objects, but with more false positives. In the D2D experiments, we always use detections from the same threshold for both training and testing.

6.3.4 Activities

For activity detection, we also rely on object detections [14]. We use the set of 20 object classes of ADL20 as described in Sect. 6.3.1 (Table 6.1), enhanced with object-related information. For every video frame, we extract objects along with their size and position in the frame. As features, we consider the BPVs as described in Sect. 6.3.3, along with the bounding box positions and the centralities.

The bounding box **(BB)** position constitutes a 4-vector per object containing the (x, y, width, height) parameters that characterize a bounding box. The values are normalized to the width and height of the frame to fall into the [0–1] range. For 20 object classes, the BB feature has length 80. The centrality feature **(CF)** signifies that a larger object area or a bounding box which is closer to the center of the image is more important for the detection of an activity. It constitutes a 2D Gaussian ($\mu = 0.5$, $\sigma = 0.1$) (in terms of normalized image coordinates) to produce a weight distribution that focuses its importance on bounding boxes found closer to the center of the frame. As a result, bigger boxes gain importance because they aggregate values over a larger area. Our intuition is that significant objects for human activities will be detected near the center of the scene or due to their size, they will draw attention to themselves [35]. A demonstration of the centrality feature's estimation is provided in Fig. 6.3.

6.4 Results

In this section, we delineate our experiments. Location classification is presented in Sect. 6.4.1 and activity recognition in Sect. 6.4.2.

6.4.1 Location Classification

We divide the experiments for location classification into ANN- and LSTM-based architectures in Sects. 6.4.1.1 and 6.4.1.3, respectively. In Sect. 6.4.1.2, we perform a per class examination for certain ANN cases.

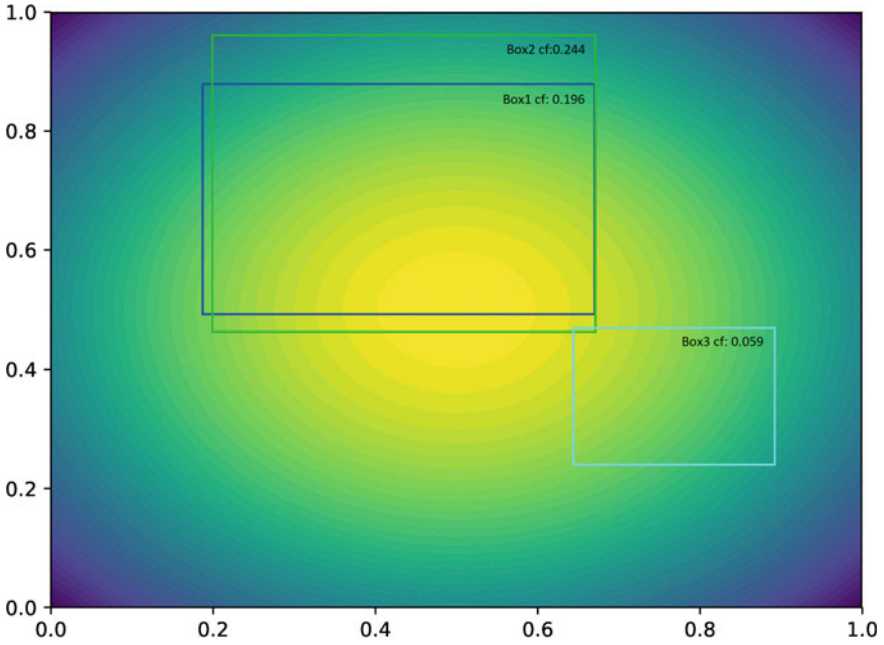

Fig. 6.3 Centrality value for three boxes. Box 1: 0.196, Box 2: 0.244 and Box 3: 0.059. As boxes become smaller or move away from the center, CF decreases

6.4.1.1 ANN Classification

Our artificial neural network models consist of five fully connected layers, with rectified linear unit (ReLU) activations for the neurons of the input and three hidden layers. The neurons per layer are 64, 256, 128, 64 and 7 (for the output), respectively. We do not apply dropout, following preliminary tests where we experience slightly worse performance. We use categorical cross entropy to calculate the loss and stochastic gradient descent for optimization. All models are trained for 150 epochs. We set the starting learning rate at 10^{-2} and divide by 10 every 50 epochs. Batch size is set to 64.

We implement experiments for the L2L, L2D and D2D cases of Sect. 6.3.3 with detection confidence threshold in the L2D and D2D cases ranging from 30 to 70%. The object sets vary between ADL20, ADL48 and MS COCO, with the latter only supporting the D2D case due to the lack of annotations for its object classes on the ADL dataset. The classifiers for each object set only differ in the input feature size which ranges between 20, 48 and 80, respectively. In Tables 6.6 and 6.7, we present the results in terms of overall Top1 accuracy and averaged F1-score over the seven locations in the test set, respectively.

When considering the *dataset combinations*, the highest classification accuracy is found in the L2L scenarios. This is expected since the object annotations do not

Table 6.6 ANN Top1 accuracies (%)—averages of the best five models of each experiment. Comparisons between L2L, L2D and D2D for the various detector cases and detection thresholds. L2L outperforms the variants that depend on object detectors. Decreasing the object detection threshold improves classification accuracy for all object sets

L2L		Thresh. (%)	L2D		D2D		
ADL20	ADL48		ADL20	ADL48	ADL20	ADL48	COCO
77.7004	77.0448	30	59.6844	54.8024	62.9514	56.4688	64.3558
		40	58.1134	48.1568	60.7436	55.8604	62.8764
		50	56.6452	47.666	58.7338	55.1106	60.839
		60	55.0006	39.368	55.7656	52.0564	57.8008
		70	47.6106	38.953	51.6538	48.6618	51.7168

Table 6.7 ANN F1-scores averaged over the seven location classes for the best performing model in Top1 accuracy. Comparison between L2L, L2D and D2D for the various detector cases and detection thresholds. The difference from the Top1 accuracies is attributed to the fact that certain locations (corridor and outdoor) are almost undetectable, affecting the average score

L2L		Thresh. (%)	L2D		D2D		
ADL20	ADL48		ADL20	ADL48	ADL20	ADL48	COCO
58.474	57.738	30	45.982	41.562	47.441	41.388	43.167
		40	42.633	39.211	45.181	40.247	39.484
		50	42.875	37.01	42.608	38.696	36.785
		60	39.21	29.674	39.043	34.866	34.758
		70	34.151	22.252	34.261	31.882	30.261

contain detector-induced noise, so the train set is clean with no objects out of place. When detectors are used, the D2D classifiers tend to outperform the L2D for the same object sets, even though they are trained on noisier samples. This fact provides insights about the way the ANN classifier handles noise. It will be confused by unexpected detections at test time; however, if it has faced similar samples during training, it deals with them more successfully at test time.

Varying the *object detector* affects the classification results significantly. In Tables 6.6 and 6.7, ADL20 L2D and D2D outperform their ADL48 L2D and D2D counterparts and COCO D2D performs better than both. When comparing ADL20 L2D with ADL48 L2D, it is important to consider the detection datasets. The test set for 'ADL20' consists of 67,906 ground truth boxes, and for 'ADL48' it is 95,845 (TP + FN in Table 6.8). The additional 28 k boxes of 'ADL48' belong to the harder to detect classes that are discarded from 'ADL20'. The low average precision values for these classes (Table 6.5) indicate that most of their instances are not detected. This suggests a harder task for 'ADL48' to produce the 'detections' dataset for any confidence threshold. For example, at 50% confidence it has less TP but more FP and FN (Table 6.8). These can be interpreted as increased noise (FP) and reduced detection quality (FN) when compared to 'ADL20'.

Table 6.8 ADL20/48 object detector results. True positives decrease along with the false positives as the confidence threshold increases, complicating the classification task

Detector	ADL20			ADL48		
Thresh. (%)	TP	FP	FN	TP	FP	FN
30	14,777	12,025	53,129	12,619	24,166	83,226
40	13,277	8231	54,629	10,509	13,800	85,336
50	11,762	5744	56,144	8493	8051	87,352
60	9951	3784	57,955	6532	4558	89,313
70	7621	2262	60,285	4417	2209	91,428

In terms of location classification accuracy, ADL20 L2D 50% is almost 9% better from ADL48 L2D 50% and ADL20 D2D 50% is 3.6% better from ADL48 D2D 50%. Finally, 'COCO', due to the higher number of training samples for each object class (over 5 k [44]) and despite consisting of 80 classes, is more robust in its object detections and the resulting location classifiers. Interestingly, in the L2L case the ADL48 variant is on a par with ADL20, meaning that the additional object classes, when not burdened by noise, do not harm location classification. The fact that COCO outperforms all other detector-based location classifiers adds to this, showing that quality detections without many false positives (resembling L2L as much as possible) even for classes from a more general context are useful.

We vary the *detection threshold* from 30 to 70% with a step of 10%. Our results suggest that as it increases, location classification performance drops. Lower thresholds lead to more available true positive object detections. This allows the location classifiers to identify uncertain locations easier, showing that they are resistant to noise. On the other hand, higher thresholds result in fewer detections with higher confidence on average and fewer false positives which, evidently, are not as adequate for inferring the location. The significant variance in the number and quality of detections as a result of modifying the confidence threshold for 'ADL20' and 'ADL48' is shown in Table 6.8 where we report the object detection results on the ADL test set videos.

6.4.1.2 Examination Per Class

In Fig. 6.4, we compare the per class F1-scores for selected ANN classifiers to examine which locations are easier and which are harder to detect. No classifier is universally better, but superiority of certain classifiers can be observed for individual locations.

ADL20 outperforms ADL48 for all locations in both the L2D and the D2D cases. Similarly, the D2D cases outperform their L2D counterparts per class in most situations. COCO D2D performs best for 'kitchen,' 'bedroom' and 'living room' due to its ability to detect additional objects such as 'fork,' 'sofa,' 'chair' and 'bed.' However, it underperforms for 'laundry room' because it lacks a location-specific object class

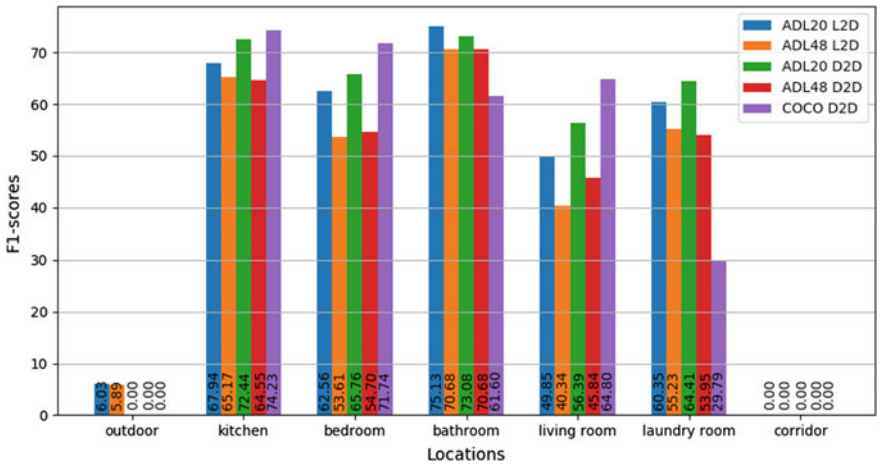

Fig. 6.4 Per class F1-scores at detection threshold 0.3 for five ANN classifiers (better seen in color—the order of names in the legend identifies their order in the graph)

related to the 'washer/dryer' of ADL20/48. Locations 'outdoor' and 'corridor' generally suffer due to the scarcity of training samples and explicitly associated objects (Table 6.2).

6.4.1.3 LSTM Classification

We are interested in studying the succession of objects in video segments instead of single frames. In Fig. 6.1 (left), the 'kitchen' scene lasts for 1260 consecutive frames (42 s) and the detected objects are not consistent throughout the segment. In certain views of the scene, the output of object detection is BPVs that cannot be associated with the ongoing location, for instance when no objects have been detected. Classifying one such BPV with an ANN classifier (e.g., ADL20 D2D 0.3) produces the mistaken prediction 'laundry room,' in between correct predictions of 'kitchen' for the surrounding frames. These frames include objects such as 'fridge' or 'oven,' but the frame in question does not. This observation drives our LSTM experiments in order to investigate how the temporal coherence of a scene can improve classification.

To test our hypothesis, we train an LSTM network with the dataset of ADL20 D2D 0.3. For training, we set the sequence size to 20 frames without augmenting the dataset with overlaps, so each sample is seen once per epoch as part of a single sequence. When testing the previously misclassified frames—now being part of a sequence—we find that the resulting location does not change from 'kitchen'. Another interesting remark from this example is the ability of the LSTM to revert the prediction back to 'kitchen' if it happens to misclassify certain frames. Given a slice of three BPVs which contain only the 'tap' object and having from previous frames an ongoing location prediction of 'kitchen' with 52% probability, we classify the first BPV. It is classified

as 'kitchen', but its probability drops to 49%. The following 'tap'-BPV modifies the prediction to 'bathroom' with probability 50%, and 'kitchen' drops further to 47%. This pattern continues for the third 'tap'-BPV. However, given a vector that includes 'fridge,' the 'kitchen' prediction returns with increasing confidence, demonstrating the ability of the LSTM to recover from false intermittent predictions.

In order to test whether the LSTMs are also quantifiably better than ANNs, we repeat the dataset parametrization experiments from Sect. 6.3.3. We expect higher Top1 accuracies with LSTMs, as well as to confirm the relative associations between L2L, L2D, D2D, the object detectors and the detection thresholds.

For our experiments, we use two stacked LSTM layers and a fully connected layer, applied at the last sequence step of the second layer. We vary the feature size between 20, 48 and 80 following the BPV requirements. We set the number of hidden units to double the feature size. Following the ANN training scheme, we use categorical cross entropy to calculate the loss and stochastic gradient descent for optimization. All models are trained for 150 epochs with 10^{-2} starting learning rate divided by 10 every 50 epochs. Sequence size is set to 20 which corresponds to a video duration of 20 s (i.e., 20 frames sampled at 1 fps) and batch size to 16 sequences.

At training time, we use a single label to describe a sequence. To produce it, we perform majority voting on the labels of all BPVs in the sequence and use that as the ground truth. Thus, the classifier is trained to produce a single prediction for the full sequence. At test time, we want to evaluate for every frame and not only once per sequence. To that end, we clone the prediction to the length of the tested sequence, to be able to evaluate against all the labels of the sequence one by one.

In Table 6.9, we report Top1 accuracies. For every task, the LSTM model surpasses the ANN equivalent. Except for the L2L combinations where the results are relatively close (2–4% difference), LSTMs show significant improvement, especially at the hardest cases, e.g., ADL48 L2D 0.6 (20.8%) and ADL48 L2D 0.7 (16.5%). The same conclusion can be drawn from Table 6.10 where the F1-scores are presented

Table 6.9 LSTM Top1 accuracy (%)—averages of the best five models of each experiment. In parentheses, the differences from the respective ANN experiments

L2L		Thresh. (%)	L2D		D2D		
ADL20	ADL48		ADL20	ADL48	ADL20	ADL48	COCO
80.2384 (+2.54)	80.6324 (+3.59)	30	70.6992 (+11.01)	63.8146 (+9.01)	70.1068 (+7.16)	65.0716 (+8.6)	75.4906 (+11.13)
		40	66.8322 (+8.72)	63.8342 (+15.68)	69.1038 (+8.36)	62.655 (+6.79)	73.9324 (+11.06)
		50	68.8314 (+12.19)	61.4836 (+13.82)	67.1894 (+8.46)	59.752 (+4.64)	73.1108 (+12.27)
		60	63.4306 (+8.43)	60.1648 (+20.8)	62.84 (+7.07)	59.3146 (+7.26)	72.3696 (+14.57)
		70	61.7324 (+14.12)	55.445 (+16.49)	61.8568 (+10.2)	56.7314 (+8.03)	67.0688 (+15.35)

Table 6.10 LSTM F1-scores averaged over the seven location classes for the best performing model in terms of Top1 accuracy. In parentheses, the differences from the respective ANN experiments

L2L		Thresh. (%)	L2D		D2D		
ADL20	ADL48		ADL20	ADL48	ADL20	ADL48	COCO
63.793 (+5.32)	58.923 (+1.19)	30	54.099 (+8.12)	54.154 (+12.6)	52.607 (+5.17)	51.421 (+10.03)	53.543 (+10.38)
		40	52.259 (+9.67)	48.385 (+9.17)	53.782 (+8.6)	49.091 (+8.84)	51.45 (+11.97)
		50	52.587 (+9.71)	47.171 (+4.16)	49.434 (+6.83)	46.28 (+7.58)	50.864 (+14.08)
		60	42.147 (+2.94)	45.754 (+16.08)	46.914 (+7.87)	45.57 (+10.7)	50.065 (+15.31)
		70	46.012 (+11.86)	39.938 (+17.69)	47.002 (+12.74)	41.196 (+9.31)	45.776 (+15.52)

instead. As expected, the absolute values are lower, because they are influenced by the distribution of the dataset and affected by its imbalance.

6.4.2 Activity Classification

In Sect. 6.4.2.1, we present the results of the activity classification scheme and in Sect. 6.4.2.2 an analysis of the class confusions. All our tests consider the **detections to detections** dataset combination introduced in Sect. 6.3.3, where both train and test splits are built from the output of an object detector. This provides a more realistic scenario for smart-home application development, compared to the label-to-label combination which assumes ideal object detections. Despite recent improvements [46] in the state-of-the-art detectors, perfect detections are not yet feasible and flawless annotations in unseen environments require significant human labeling effort.

6.4.2.1 LSTM Classification

For the activity classification experiments, we train an LSTM network for the sequences of each feature combination of Sect. 6.3.4 targeting the 33 activity classes of the ADL dataset (Table 6.3). We prefer LSTM over artificial neural networks for their ability to incorporate temporal changes, compared to per frame classification schemes that do not consider objects seen in the past. We train a single-layer LSTM with 80 hidden units with a fully connected layer for the output. We set sequence size to 150 frames and batch size to 64. We apply 15% dropout with 10^{-4} starting learning rate with polynomial decay down to 10^{-6} in 1000 training iterations. We finish training after 1500 iterations.

Table 6.11 LSTM Top1 accuracies (%) and averaged F1-scores for all 33 classes for the feature combinations

Feature	BPV	BB	CF	BPV + BB	BPV + CF	BB + CF	BPV + BB + CF
Top1	32.16	28.66	23.38	29.21	32.84	31.06	**33.97**
F1	10.51	9.07	6.69	11.11	10.89	10.83	**12.4**

We report Top1 accuracies and F1-scores in Table 6.11. The highest performing individual feature is the binary presence vector. Adding the bounding box coordinates hurts results, but adding the centrality feature leads to the best performance overall. The high number of classes adds complexity to the classification task when compared to locations and leads to lower results overall.

6.4.2.2 Class Confusions

The confusion matrix in Fig. 6.5 shows a specific trend. It suggests strong preference to certain activities, including 0: 'background', 3: 'brushing teeth', 9: 'laundry', 10: 'washing dishes', 12: 'making tea', 17: 'making cold food/snack', 22: 'watching TV', 23: 'using computer', 27: 'reading book'. Beside their true positives, these classes attract false positives from conceptually relevant activities that rely on the same

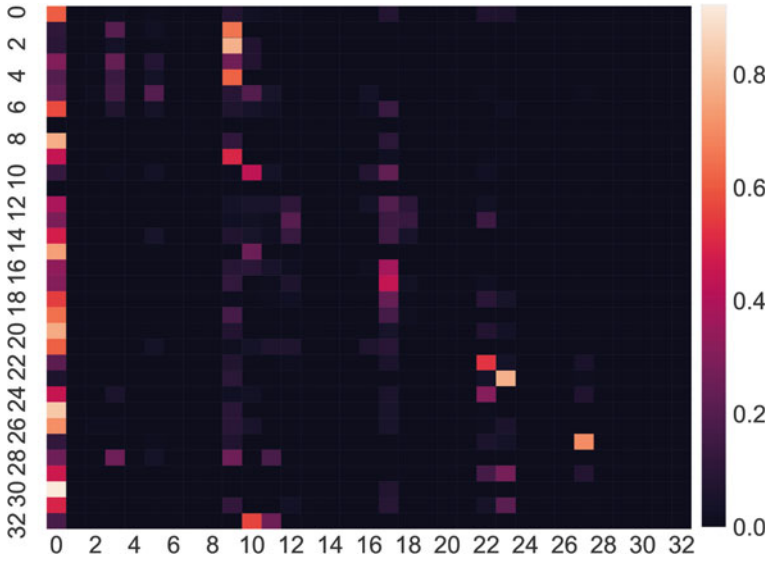

Fig. 6.5 Confusion matrix for BPV + BB + CF. Some activities can be assigned to semantic super sets

objects for recognition, but have fewer instances associated with them at training time (Table 6.3).

Class 17: 'making cold food/snack' contains false assignments from classes 16: 'making hot food' and 18: 'eating food/snack.' These activities rely on similar kitchen objects such as 'dish', 'mug/cup' and 'tap', but the classifier assigns them to the class with the most instances during training. Similarly, instances from 29: 'writing' are assigned to 27: 'reading book' based on 'book' as the detected object. Further confusions that regard semantic relevance include classes 1: 'combing hair', 2: 'makeup' and 4: 'dental floss' with 9: 'laundry', class 28: 'mouth wash' with 3: 'brushing teeth' and class 32: 'grabbing tap water' with 10: 'washing dishes'.

6.5 Discussion

We envision a system that recognizes activities and locations from objects in a real setting. We structure the task in a very simple way, i.e., to solely rely upon the presence of objects in the scene for inference. This is a source of confusion even with the assumption of perfect detections, considering that objects are naturally found in multiple locations ('door') or are movable ('cup'). We work with these limitations and explore ways to address them by relying on the temporal associations of objects to learn an improved representation of a scene or an activity.

Using the L2L combination for the location classifiers is not a pragmatic approach mostly due to the difficulties in data collection, such as human annotation effort in customized home environments. To mitigate these, we make use of automatic object detectors, pretrained on specific sets of object classes. Initially, this leads to the L2D case, where we can train on generic room representations; e.g., a common kitchen has a fridge, an oven and a tap, and expect to detect these objects in the test environment. However, the D2D classifiers have better performance than L2D and show increased resilience to noisy detections at test time. Additionally, they are more convenient installation-wise, since they abolish the necessity for labeling locations with objects. Thus, having minimized the required human labeling effort, it becomes easier to learn new representations of existing places (e.g., with a specialized detector that was previously unavailable), but also of unseen locations not included in the original categories.

Our purpose is to evaluate the applicability of the detectors in terms of object variability and acceptable accuracy of activity and location classification. The L2L experiments have the highest location classification performance and establish the idea that the amount of noise in the object detections (in this case, the lack thereof) influences the results substantially. A second significant outcome is that the variability of available objects can enhance the ability of a classifier to detect a location accurately. This is observed from ADL48 L2L which outperforms ADL20 L2L in Top1 accuracy in the LSTM tests. However, when the objects contain noise, less is more; i.e., in L2D and D2D, ADL48 does not outperform ADL20 in any (LSTM or ANN) experiment.

To enhance our original objective in activity classification, we consider a scenario with detections of house-related objects with additional object appearance features that are dynamic in time. We examine them with LSTM and find that this more complicated scenario cannot be sufficiently tackled with object-based features. While seven locations have been distinguished with up to 70% accuracy (LSTM ADL20 D2D 0.3), 33 activities reach 34%, with multiple intra-class similarities being observed and misclassified as such. The additional information about the object detections contributes to the results; however, it is not enough to properly address the elaborate task of analyzing semantically similar activities and to fully counter the bias toward classes with higher prior probability.

To our knowledge, there is no related work that tackles location classification in the ADL dataset. For the task of activity classification, related work does exist; however, the evaluation metrics and the regarded activity sets do not allow for a fair comparison. For example in [47], the authors report 26.3% average accuracy only for the 18 activities of the ADL dataset.

6.6 Conclusions

Throughout this chapter, we explore the recognition of indoor locations and human activities in egocentric videos. We utilize a state-of-the-art object detection architecture, trained separately on three object sets. We apply it on egocentric videos to extract objects at various detection thresholds and classify these detections with artificial neural networks and long short-term memory networks to infer locations or activities.

We find that the selection of object set affects the relevance of the detections in the location classification task and the detection threshold, their number and quality. Using the binary presence vector, we manage to have acceptable performance for indoor location classification, reaching 75.5%. One important discovery is that the lack of noise in the detections is preferable, but if it cannot be avoided the true positive/false negative trade-off favors the true positives even at the expense of extending the set of false positive detections. The comparison between ANN and LSTM promotes the incorporation of temporal structure in the BPVs (Tables 6.9 and 6.10) in order to capitalize on the sequential nature of the data and minimize the effects of erroneous detections.

We also find that the more complicated task of activity classification is harder to tackle based only on object-based features. Our results show that certain activities are easier to recognize than others, mostly due to their higher occurrence rate in the training set. We also find that activities which belong in semantic 'super' sets tend to be learned as belonging to the one representative activity that has the most instances in the training set.

An interesting direction for future work would be to analyze the associations between locations and activities and whether the first can assist in recognizing the latter.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 676157.

References

1. Kapidis G, Poppe RW, van Dam EA et al (2018) Where Am I? Comparing CNN and LSTM for location classification in egocentric videos. In: 2018 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops), pp 878–883
2. Ma M, Fan H, Kitani KM (2016) Going deeper into first-person activity recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 1894–1903
3. del Molino AG, Tan C, Lim J, Tan A (2017) Summarization of egocentric videos: a comprehensive survey. IEEE Trans Hum-Mach Syst 47:65–76. https://doi.org/10.1109/THMS.2016.2623480
4. Yonetani R, Kitani KM, Sato Y (2016) Recognizing micro-actions and reactions from paired egocentric videos. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 2629–2638
5. Damen D, Leelasawassuk T, Mayol-Cuevas W (2016) You-Do, I-Learn: egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance. Comput Vis Image Underst 149:98–112. https://doi.org/10.1016/j.cviu.2016.02.016
6. Kretch KS, Franchak JM, Adolph KE (2014) Crawling and walking infants see the world differently. Child Dev 85:1503–1518. https://doi.org/10.1111/cdev.12206
7. Nguyen T-H-C, Nebel J-C, Florez-Revuelta F (2016) Recognition of activities of daily living with egocentric vision: a review. Sensors (Basel, Switzerland) 16:72. https://doi.org/10.3390/s16010072
8. Karaman S, Benois-Pineau J, Megret R et al (2010) Human daily activities indexing in videos from wearable cameras for monitoring of patients with dementia diseases. In: 2010 20th international conference on pattern recognition, pp 4113–4116
9. Teriús-Padrón JG, Kapidis G, Fallmann S et al (2018) Towards self-management of chronic diseases in smart homes: physical exercise monitoring for chronic obstruction pulmonary disease patients. In: 2018 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops), pp 776–781
10. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1, pp 886–893
11. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94
12. Pirsiavash H, Ramanan D (2012) Detecting activities of daily living in first-person camera views. In: 2012 IEEE conference on computer vision and pattern recognition, pp 2847–2854
13. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
14. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 6517–6525
15. Furnari A, Farinella GM, Battiato S (2016) Temporal segmentation of egocentric videos to highlight personal locations of interest. In: Hua G, Jégou H (eds) Computer vision—ECCV 2016 workshops. Springer International Publishing, pp 474–489
16. Furnari A, Farinella GM, Battiato S (2017) Recognizing personal locations from egocentric videos. IEEE Trans Human-Mach Syst 47:6–18. https://doi.org/10.1109/THMS.2016.2612002
17. Nakamura K, Yeung S, Alahi A, Fei-Fei L (2017) Jointly learning energy expenditures and activities using egocentric multimodal signals. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 6817–6826

18. Lee YJ, Ghosh J, Grauman K (2012) Discovering important people and objects for egocentric video summarization. In: 2012 IEEE conference on computer vision and pattern recognition, pp 1346–1353
19. Fathi A, Ren X, Rehg JM (2011) Learning to recognize objects in egocentric activities. CVPR 2011:3281–3288
20. Fathi A, Li Y, Rehg JM (2012) Learning to recognize daily actions using gaze. In: Fitzgibbon A, Lazebnik S, Perona P et al (eds) Computer vision—ECCV 2012. Springer, Berlin, pp 314–327
21. Poleg Y, Arora C, Peleg S (2014) Temporal segmentation of egocentric videos. In: 2014 IEEE conference on computer vision and pattern recognition, pp 2537–2544
22. Betancourt A, Díaz-Rodríguez N, Barakova E et al (2017) Unsupervised understanding of location and illumination changes in egocentric videos. Pervasive Mob Comput 40:414–429. https://doi.org/10.1016/j.pmcj.2017.03.016
23. Altwaijry H, Moghimi M, Belongie S (2014) Recognizing locations with Google Glass: a case study. In: IEEE winter conference on applications of computer vision, pp 167–174
24. Lee N, Kim C, Choi W et al (2017) Development of indoor localization system using a mobile data acquisition platform and BoW image matching. KSCE J Civ Eng 21:418–430. https://doi.org/10.1007/s12205-016-1057-5
25. Lu G, Yan Y, Sebe N, Kambhamettu C (2017) Indoor localization via multi-view images and videos. Comput Vis Image Underst 161:145–160. https://doi.org/10.1016/j.cviu.2017.05.003
26. Qian K, Zhao W, Ma Z et al (2018) Wearable-assisted localization and inspection guidance system using egocentric stereo cameras. IEEE Sens J 18:809–821. https://doi.org/10.1109/JSEN.2017.2773487
27. Dovgalecs V, Mégret R, Berthoumieu Y (2013) Multiple feature fusion based on co-training approach and time regularization for place classification in wearable video. Adv Multimed 2013. https://doi.org/10.1155/2013/175064
28. Vaca-Castano G, Das S, Sousa JP (2015) Improving egocentric vision of daily activities. In: 2015 IEEE international conference on image processing (ICIP), pp 2562–2566
29. Vaca-Castano G, Das S, Sousa JP et al (2017) Improved scene identification and object detection on egocentric vision of daily activities. Comput Vis Image Underst 156:92–103. https://doi.org/10.1016/j.cviu.2016.10.016
30. Greff K, Srivastava RK, Koutník J et al (2017) LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst 28:2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924
31. Karpathy A, Johnson J, Fei-Fei L (2015) Visualizing and understanding recurrent networks. arXiv preprint arXiv:150602078
32. Smith LN (2018) A disciplined approach to neural network hyper-parameters: part 1—learning rate, batch size, momentum, and weight decay. CoRR arXiv preprint arXiv:abs/1803.09820
33. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28:976–990. https://doi.org/10.1016/j.imavis.2009.11.014
34. Bambach S, Lee S, Crandall DJ, Yu C (2015) Lending a hand: detecting hands and recognizing activities in complex egocentric interactions. In: 2015 IEEE international conference on computer vision (ICCV), pp 1949–1957
35. Bertasius G, Park HS, Yu SX, Shi J (2017) First person action-object detection with EgoNet. In: Proceedings of robotics: science and systems
36. Li Y, Zhefan Y, Rehg JM (2015) Delving into egocentric actions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 287–295
37. Poleg Y, Ephrat A, Peleg S, Arora C (2016) Compact CNN for indexing egocentric videos. In: 2016 IEEE winter conference on applications of computer vision (WACV), pp 1–9
38. Fathi A, Farhadi A, Rehg JM (2011) Understanding egocentric activities. In: 2011 international conference on computer vision, pp 407–414
39. Wray M, Moltisanti D, Mayol-Cuevas W, Damen D (2016) SEMBED: semantic embedding of egocentric action videos. In: Hua G, Jégou H (eds) Computer vision—ECCV 2016 workshops. Springer International Publishing, pp 532–545
40. Wu J, Osuntogun A, Choudhury T et al (2007) A scalable approach to activity recognition based on object use. In: 2007 IEEE 11th international conference on computer vision, pp 1–8

41. Su Y-C, Grauman K (2016) Leaving some stones unturned: dynamic feature prioritization for activity detection in streaming video. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision—ECCV 2016. Springer International Publishing, pp 783–800
42. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 779–788
43. Deng J, Dong W, Socher R et al (2009) ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255
44. Lin T-Y, Maire M, Belongie S et al (2014) Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision—ECCV 2014. Springer International Publishing, pp 740–755
45. Girshick R (2015) Fast R-CNN. In: 2015 IEEE international conference on computer vision (ICCV), pp 1440–1448
46. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. CoRR arXiv preprint arXiv:abs/1804.02767
47. Nguyen T-H-C, Nebel J-C, Florez-Revuelta F (2018) Recognition of activities of daily living from egocentric videos using hands detected by a deep convolutional network. In: Campilho A, Karray F, ter Haar Romeny B (eds) Image analysis and recognition. Springer International Publishing, pp 390–398