Where Am I? Comparing CNN and LSTM for Location Classification in Egocentric Videos

Georgios Kapidis*†, Ronald W. Poppe†, Elsbeth A. van Dam*, Remco C. Veltkamp†, Lucas P. J. J. Noldus*
*Noldus Information Technology, Wageningen, The Netherlands
†Department of Informatics and Computer Science, University of Utrecht, Utrecht, The Netherlands

***Abstract*—Egocentric vision is a technology that exists in a variety of fields such as life-logging, sports recording and robot navigation. Plenty of research work 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 locations can be characterized by the presence of specific objects. Our objective is the recognition of locations in egocentric videos that mainly consist of indoor house scenes. We perform an extensive comparison between Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) based classification methods that aim at finding the in-house location by classifying the detected objects which are extracted with a state-of-the-art object detector. We show that location classification is affected by the quality of the detected objects, i.e. the false detections among the correct ones in a series of frames, but this effect can be greatly limited by taking into account the temporal structure of the information by using LSTM. Finally, we argue about the potential for useful real-world applications.**

I. Introduction

Egocentric vision is an essential part of various video analysis tasks, extending from activity recognition [1] and social interaction analysis [2], to automatic extraction of visual guidelines [3] and infant visual attention [4]. The area considered in this paper is indoor location detection from egocentric videos, with possible applications in Ambient Assisted Living (AAL) for people suffering from limited vision or dementia.

To produce an inference on an image or video frame, one could calculate image-descriptive features, stack them in a vector per frame and classify them with machine learning models in a supervised fashion. In recent years, feature extraction and classification have merged into end-to-end classification methods with deep networks, providing state-of-the-art results. In this work, we take a step back and consider a different type of feature to infer the location.

Our key technique is to use the detected objects in a video frame as cues to recognize the location. We build on the idea that rooms can be characterized by the presence of specific, distinctive objects. This consistency can be translated into an association between objects and locations. Consider, for example, Fig. 1 which shows the detected objects of an egocentric video segment from a kitchen. There are three categories of objects; (1) those that can be thought of as movable, but bear meaning for understanding the scene, such as the mug and the dish, (2) those that are distinctive to this particular location, such as the stove, the microwave or the fridge and (3) those that can be found in more than one location, for example the soap and the tap, which could also appear in a bathroom.

This motivates us to perform an analysis on the videos of the Activities of Daily Living dataset (ADL) [5] and discover the associations between the objects and their locations. To that end, we train multiple classifiers with Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) [6] to experiment with per frame classification and utilization of the temporal structure of the data, respectively. Conceptually, a single frame of a scene might include only partial information about the objects, as not all that are detectable could fit in it. However, the combination of multiple frames over time would produce a more complete view of the room. Eventually, we compare the performance of classifiers from both types of models, trained either on object annotations or the detections from three similar detectors, each trained on different object categories from different datasets.

The contribution of this paper is the development of a method to analyze object associations with locations in egocentric videos. In addition, we describe the results of a thorough parameter search for CNN and LSTM networks and provide inferences about the outcome.

Section II is an overview of related work in egocentric object and location detection, datasets and applications in the field. In Section III we describe the dataset we used, the object detectors and our classification modules and in Section IV we present our results. Our findings are discussed in Section V and we conclude in Section VI.

II. Related Work

We focus on recognizing indoor locations based on the detected objects from egocentric videos of people moving freely in their homes. The ADL dataset [5] has these characteristics and the necessary object annotations. Scientific literature provides a plethora of egocentric datasets [7]–[13] with a variety of attributes. The datasets of [7], [8] are created with the aim of detecting locations and observing indoor and outdoor everyday scenes. The dataset of [13] focuses on activities that take place either indoors or outdoors, such as walking, running and sitting, whereas [9] is enhanced with accelerometer and heart rate data to infer the sedentarity level of performed activities. The dataset of [10] includes annotations and segmentations

978-1-5386-3227-7/18/$31.00 ©2018 IEEE

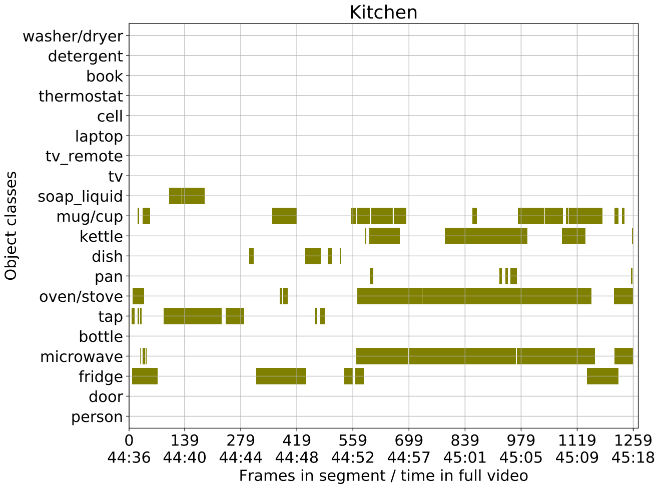

Fig. 1: The detected objects for ‘kitchen‘. Certain objects, such as the oven and the microwave, dominate the scene.

of the important objects that characterize the activities, with a large amount of the videos being outdoors. Datasets from [11], [12] 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 or labeling the environment, has been under active research in egocentric vision. In [14], an unsupervised way of combining scene illumination and location characteristics is proposed to enhance the usability of wearable cameras. Location recognition is indirectly the task in [15] 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 [16] where the combination of a camera and a 2D laser scanner is applied to register images to the real world coordinate system. A multi-view indoor localization system from images and videos is proposed in [17]. The algorithm computes self-similarity matrices from the extracted images to correlate the various captured views of the scene. Afterwards, it learns the system through these features and when a query image is given, it is able to provide its location and orientation. The combination of wearable egocentric stereo cameras and inertial sensors is considered in [18] to map an outdoor workspace and provide route guidance for specific tasks in the workplace.

A thorough system for place classification is described in [19]. Visual recognition is based on low-level features and complicated semi-supervised training procedures to take advantage of sparsely annotated available data. Temporal segmentation of egocentric videos with CNN and Hidden Markov Models is considered in [7] to highlight personal locations of interest, trained with user-provided positive samples of locations, also learning to reject locations that are not specified by the user. Personal locations are also analyzed in [8] as part of a user’s daily activities. For the frames of the videos, either global image or CNN-extracted features are classified into locations. In [20], [21], the combined improvement of object detection and scene identification is investigated. Initially, scene identification is performed from temporal egocentric cues and its results are used to improve the results of object detectors by associating the objects with specific locations. Eventually, they show that by using an LSTM network to train on the temporal sequences of the detected objects directly, it is possible to improve the results of the detectors without explicitly using the 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 the locations, a concept that is not uncharted in the activity recognition domain [1], where object detections and hand object interactions, among others, are explored towards activity recognition.

Searching through the parameters of a model to find the optimal configuration is enticing, as suggested by the volume of relevant work [22]–[26]. In [22], a multitude of LSTM variants is tested on speech recognition, handwriting recognition and music modeling tasks to inspect the differences in the 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 hyper-parameters used for training are also explored, but it is observed that they are uncorrelated. In an effort to provide further insights to the reasons behind the effectiveness of LSTM, [25] provides a thorough study on cell activations, error analysis and data representations. In the context of searching for the best parametrization of an algorithm to gain optimal performance, our work comprises a large scale search on CNN and LSTM configurations.

III. Methodology

Our method amounts to analyzing an egocentric dataset in terms of objects and locations (Sec. III-A), choosing an object detection framework to perform our tests (Sec. III-B) and creating a pipeline for location classification (Sec. III-C).

A. Activities of Daily Living (ADL) Dataset

The ADL dataset [5] consists of 20 videos of 18 morning activities occurring indoors. Each video is an egocentric record of the subject’s choice of the proposed activities performed in an unscripted manner. In every video, the subject is different and is performing activities in his/her own house, which provides considerable variations in locations, activities and video appearances. In total, there are approximately 10 hours of egocentric videos, equivalent to more than one million frames. The videos have been manually annotated to include action labels, 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 on the data.

There exist annotations for 48 object classes, although only 42 are considered valid and mentioned as such in the original paper due to the very low number of training or testing samples. A list of the object classes together with their occurrences in the dataset appears in Table I.

Table I: The object classes of the ADL dataset and the occurrences in the videos. In parenthesis the instances in the train set.

person	**door**	**fridge**	**microwave**	**bottle**	**tap**	**oven/stove**	**pan**
4650 (2424)	7903 (2019)	1999 (301)	2369 (527)	10310 (1705)	7826 (3252)	3196 (1007)	3156 (1026)
trash can	**dish**	cloth	knife/spoon/fork	food/snack	**kettle**	**mug/cup**	**soap liquid**
2075 (486)	8216 (2274)	3077 (78)	4843 (1893)	3876 (741)	1239 (464)	11050 (2766)	8375 (2658)
pills	basket	towel	tooth brush	tooth paste	electric keys	**tv**	**tv remote**
394 (148)	1588 (35)	4480 (1961)	1795 (819)	1746 (492)	1570 (417)	5600 (2033)	2813 (1253)
container	shoes	tea bag	**laptop**	cell phone	**cell**	**thermostat**	**book**
5685 (3821)	3248 (735)	359 (177)	7027 (2183)	653 (271)	571 (238)	332 (137)	4770 (445)
dental floss	vacuum	elec keys	pitcher	**detergent**	**washer/dryer**	bed	large container
547 (385)	519 (116)	118 (118)	1208 (277)	1105 (297)	3362 (954)	783 (228)	558 (6)
monitor	keyboard	shoe	blanket	comb	perfume	milk/juice	mop
316 (287)	107 (102)	694 (300)	85 (31)	307 (51)	550 (0)	366 (0)	403 (0)

We are interested in the analysis of locations, so the dataset is expanded with the location annotations from [21]. For every 30 frames of video, one of eight possible location classes is defined, namely, kitchen, bedroom, bathroom, living room, laundry room, corridor, outdoor and undefined (see also Table II). The last class occurs in blurred frames or non-identifiable places which we do not use for training and testing the location classification models. Hence, the location classes are seven.

Table II: Samples per location. Some classes are better represented. Class 'undefined' is not used for training and testing.

Location class	Train set	Test set	Total
undefined	492	737	1229
outdoor	143	906	1049
kitchen	3414	6850	10264
bedroom	1821	3966	2285
bathroom	2307	2285	4592
living room	2606	5045	7651
laundry room	815	1097	1912
corridor	45	133	178
Sum	11463	21019	32662

B. Object Detection

In our experiments we use the Darknet framework[1] to detect objects. In total, we have three detectors based on the YOLOv2 CNN as described in [27]. It is a real-time object detection system that can operate on various input sizes, a feature we also exploit to improve detection performance. YOLOv2 is based on the Darknet-19 architecture [27], which is pre-trained on ImageNet [28] for the 1,000 class classification task, for 160 epochs. Our first detector is YOLOv2, fine-tuned on the 80 classes of the MS COCO dataset [29] and provided by its authors. We train two additional models based on this architecture for the object classes of ADL: (1) ADL48, on all the classes in Table I and (2) ADL20, on the marked. This selection of classes is based on [21].

The reason for the diversity in the detectors is that COCO and ADL consist of different sets of classes. ADL comprises objects found in homes, whereas COCO is more general in its categories. Also, the number of classes in COCO does not affect the quality of the detections due to the number of labeled instances per class (over 5,000 [29]), which is not the case in ADL, with fewer samples overall. The split between ADL20 and ADL48 attempts to address this, by excluding classes which are harder to detect.

C. Location Classification

We model the relationship between the objects in a frame or a series of frames to recognize the location. After object detection is applied on the train and test videos of the ADL dataset, we get a vector of zeros and ones for every video frame, at the length of the object detector's output, i.e. 80 for the COCO detector, 48 for ADL48 and 20 for ADL20. The ones in each vector constitute the detected objects and the zeros the undetected. We have location annotations every 30 frames (1 second) [21] and only use these in classification.

We train two types of classifiers. The first are based on CNN architectures that do not take into account the temporal structure of the data, whereas the second are based on LSTM, which are trained on sequences of the aforementioned vectors. We use TensorFlow version 1.3.0 for our experiments.

Global parameters for both cases include: (1) the **object detectors** to produce the detections, (2) their **detection threshold** and (3) the **dataset combinations**. The threshold creates a trade-off between the confidence and the amount of detections. Higher thresholds result in more confident but fewer detections. Lower thresholds provide more objects, but with more false-positives. The dataset combinations comprise three scenarios that affect the composition of a location classifier's dataset. *Labels to labels (L2L)* which consist of the ADL object annotations for the train and test sets, i.e. the object detections are *not* used for the L2L based models. *Labels to detections (L2D)* which use the object annotations for training and the detections of a specific detector with a specified minimum confidence threshold for testing. *Detections to detections (D2D)* which are composed of detections for a common threshold for both train and test sets. The classifiers based on MS COCO are trained only for D2D due to the lack of annotations for its object classes in the ADL dataset.

1) CNN Classifiers: We vary the number of training steps and the dropout. The structure of the networks can be seen in Table III. The ADL48 classifiers come in two variants to measure the effect of network depth against the size of the input vector; one with two layers with pooling, same as the ADL20 and one with three like the COCO classifiers. We use Rectified Linear Unit (ReLU) activations, Adam for optimization [30] and cross entropy for the loss function.

2) LSTM Classifiers: We vary the number of stacked LSTM layers, the number of hidden units per layer, the batch size and the sequence size. (Table IV) We apply dropout after each

[1] http://pjreddie.com/darknet/

Table III: CNN structures. For ADL20 and ADL48-2 the first two conv and maxpool layers apply. For ADL48-3 and COCO all layers apply. FC, dropout and softmax apply to all.

Type	Filters	Size/Stride	Output ADL20/48/COCO
Convolutional	32	5 x 5	1 x 20 / 1 x 48 / 1 x 80
Maxpool		2 x 2 / 2	1 x 10 / 1 x 24 / 1 x 40
Convolutional	64	5 x 5	1 x 10 / 1 x 24 / 1 x 40
Maxpool		2 x 2 / 2	1 x 5 / 1 x 12 / 1 x 20
Convolutional	128	5 x 5	- / 1 x 12 / 1 x 20
Maxpool		2 x 2 / 2	- / 1 x 6 / 1 x 10
Fully connected, Dropout, Softmax			

LSTM layer. After the final LSTM, there is a softmax layer for the output. Similarly to the CNN, the loss function is cross entropy and we use the Adam optimizer.

Table IV: The parameters we adjust for the LSTM classifiers.

Parameter	Values
Stacked layers	1, 2, 5
Hidden units	1, 2, 3, 4, 5 * length of the input vector
Batch size	1, 2, 5, 10, 20
Sequence size	1, 2, 5, 10, 20, 50

IV. Results

Following, we present the variants of the object detectors with the best performance, the results of the CNN classifiers, a case of per class examination and the LSTM classifiers.

A. Object Detectors

The better performing of our variations of the YOLOv2 architecture have input size 448 x 608 and are trained for 35,000 iterations. The remaining parameters are the same as [27]. The recall of ADL20 is 31.29% and of ADL48 is 28.41% on their respective ADL test set.

B. CNN Results

In order to discover the optimal configuration we experiment with the hyper-parameters of Sec. III-C1. To reduce overfitting, a dropout value of 0.85 is deemed optimal. We train for $5,000$ training steps, with a learning rate of 10^{-4} and batch size of 50 samples. Our main objective is to assess the effects of the object detection threshold to the location classification. In total, we have four CNN classifier variants: ADL20, ADL48-2, ADL48-3 and COCO, tested for the applicable dataset combinations of Sec. III-C. All experiments are executed five times to mitigate the effect of randomization in the training procedure and the results show the average value of the respective executions. In Fig. 2 the results are presented in terms of overall accuracy on the test set and in Fig. 3 in terms of averaged F1-score over the seven locations.

The best results are found in the L2L **dataset combinations**. This can be expected from the fact that the object annotations do not include noise, thus the train set is clean without contradicting objects, while the lack of false detections in the test set makes it more compatible to the knowledge of the classifier. This is also supported by the fact that the D2D classifiers who use noisy data for training and testing have better performance than their L2D counterparts (Table V).

We vary the **detection threshold** from 0.3 to 0.7 with a step of 0.1. In general, as the threshold increases, performance drops. A lower threshold means more available object detections, even if they include more false positives, allowing the location classifier to identify uncertain locations easier, while probably ignoring the false positive objects. On the other hand, higher thresholds result in less detections with higher confidence, which may not be enough to infer the location.

Varying the **object detector** affects the classification results significantly. In Fig. 2 and 3 we see that ADL20 performs better than the ADL48 variants and COCO is better than both. The smaller number of object classes of ADL20 makes it possible to have a simpler representation with higher accuracy. On the contrary, the COCO detector, due to the greater amount of training samples for each class is generally more robust, despite being trained for 80 classes. This shows that a generic object detector with better detection performance is preferable to a less successful one that is more relevant to the context.

Table V: CNN results of L2L, L2D and D2D.

	ADL20			ADL48-3		
	L2L	L2D	D2D	L2L	L2D	D2D
Overall accuracy	0.76	0.57	0.59	0.73	0.48	0.54
F1-score	0.56	0.46	0.47	0.58	0.40	0.45

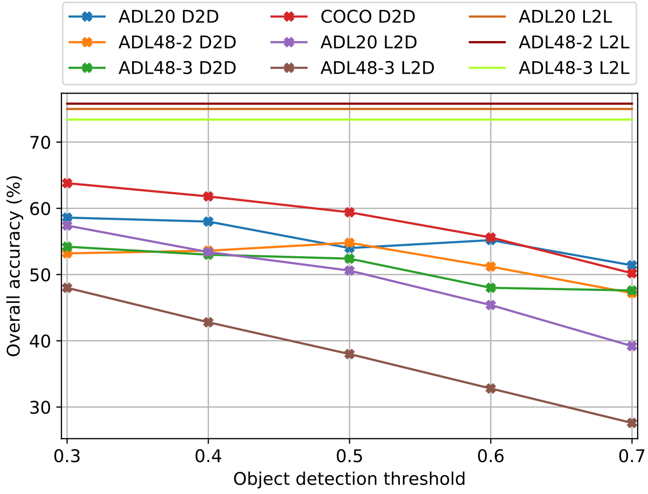

Fig. 2: The overall accuracy in the test set for each CNN model. The fact that it drops as the threshold increases gives an interesting insight about the quality of the object detections even with lower confidence, i.e. the quality of the detections at threshold lower than 0.5 is high and able to produce meaningful results while it drops significantly afterwards.

C. Per Class Examination

In Fig. 4 we compare the per class F1-scores for three CNN classifiers. Our objective is to see which locations are harder to detect. Locations 'outdoor' and 'corridor' suffer from the scarcity of training samples plus object classes cannot be registered explicitly to them. COCO, being more generic,

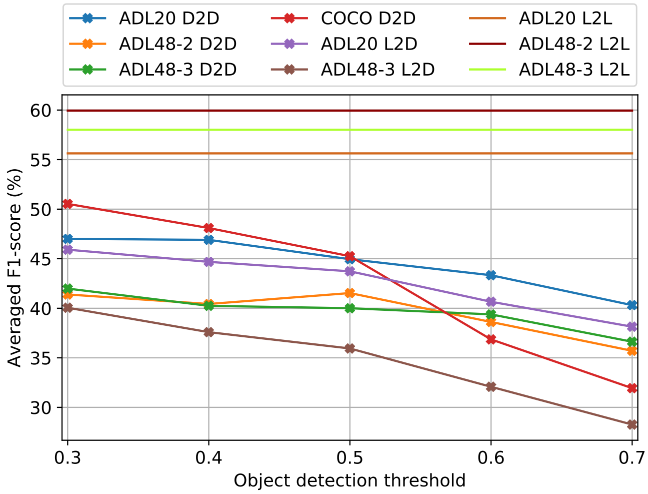

Fig. 3: The F1-score, averaged over the 7 locations. The fact that some locations are essentially undetectable (corridor, outdoor), affects the F1-score compared to the overall accuracy. This leads to the conclusion that the dataset is imbalanced with a clear advantage for classes which are easier to detect.

detects objects 'car' and 'stop sign' which enable it to classify some instances of 'outdoor' correctly. Also, it performs better in classes 'bedroom' and 'living room' because it detects 'sofa', 'chair' and 'bed', usually spotted in these locations. It underperforms in 'laundry room' because it lacks an object class similar to 'washer/dryer' of ADL20. The two variants of ADL20 have similar average F1-scores (Table V) and are performing better for 'kitchen' and 'bathroom' due to their specialization in relevant objects like 'pan', 'tap' and 'soap'.

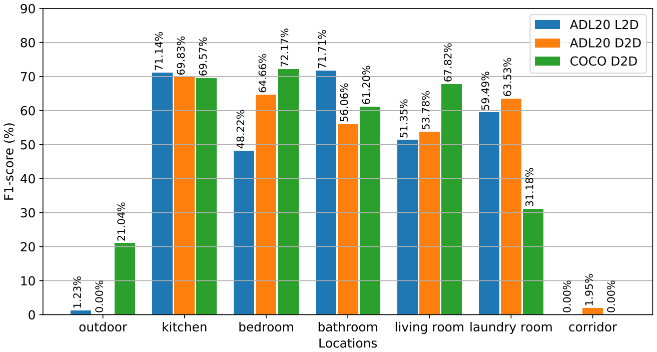

Fig. 4: The per class F1-scores at detection threshold 0.3 for three CNN classifiers.

D. LSTM Results

One major drawback of our CNN architecture is its inability to take advantage of the sequential nature of the data, i.e. the temporal associations of the objects with the locations. In Fig. 1 the same scene exists for 42 consecutive seconds, but the objects are not consistent throughout the segment. This intuition drives us to perform tests with LSTM modifying the parameters of Table IV to discover a more optimal classifier for this task. We set dropout to 0.85, learning rate to 10^{-4} and train steps to 500. To mitigate the effects of randomization we create every model five times and average their results. In total, the number of trained LSTM models is 38,250. We investigate the improvement over CNN with the use of LSTM.

In Table VI we compare the best accuracies from LSTM models to the best accuracy of a CNN model with the same global parameters. In every task there exists an LSTM model that surpasses the CNN classifier. Except for the L2L combination, where the results are relatively close (+5%), LSTM show great relative improvement. For example, at the difficult case of ADL48 L2D with threshold 0.7 it is 68% better. The same conclusion can be drawn from Table VII where the F1-scores are presented instead. As expected, they are smaller, because they take into account the distribution of the dataset and are affected by its imbalance (Table II). Still, the improvement is considerable. To compare with the previous example, the relative improvement for L2D at threshold 0.7 is 28.57%.

Table VI: LSTM best overall accuracies compared to the best CNN in parenthesis.

Combination	ADL20	ADL48	COCO
L2L	0.80 (0.76)	0.79 (0.76)	-
L2D 0.3	0.69 (0.57)	0.62 (0.48)	-
L2D 0.5	0.65 (0.51)	0.56 (0.38)	-
L2D 0.7	0.54 (0.39)	0.47 (0.28)	-
D2D 0.3	0.69 (0.59)	0.64 (0.54)	0.77 (0.64)
D2D 0.5	0.67 (0.54)	0.62 (0.52)	0.47 (0.28)
D2D 0.7	0.61 (0.51)	0.56 (0.48)	0.74 (0.59)

Table VII: Best LSTM F1-scores compared to the best CNN F1-scores in parenthesis.

Combination	ADL20	ADL48	COCO
L2L	0.64 (0.56)	0.63 (0.60)	-
L2D 0.3	0.53 (0.46)	0.49 (0.40)	-
L2D 0.5	0.52 (0.44)	0.44 (0.36)	-
L2D 0.7	0.52 (0.44)	0.36 (0.28)	-
D2D 0.3	0.52 (0.47)	0.50 (0.42)	0.60 (0.51)
D2D 0.5	0.52 (0.45)	0.49 (0.40)	0.56 (0.45)
D2D 0.7	0.47 (0.40)	0.46 (0.37)	0.49 (0.32)

V. Discussion and Future Work

To develop a system that recognizes locations from objects in a real setting, using the L2L combination for the location classifiers is not an option due to the difficulty to acquire real data. To address this, we made use of the object detector as a means to provide real-time input to the classifier. Initially, this leads to the L2D case, where we could train on generic room representations e.g. a common kitchen has a fridge, an oven and a tap, and expect to detect these objects at a test environment. However, the D2D experiments have better results and are easier to use since they abolish the necessity for object labeling of the generic room. Thus, the system can easily learn new representations of existing places (for example, with a specialized detector that was previously unavailable), but also of unseen locations not included in the original categories.

A downside of D2D is the complete dependence on the object detection quality to both create and identify room representations, whereas L2D rely on it only for the identification part. Nevertheless, our experiments with the ADL dataset show that for most locations in our set all detectors provide data of sufficient quality to classify locations, especially when LSTM is applied to take into account their sequential nature.

To use this method in a real AAL scenario further research would be required to improve the overall results and address the incomplete classification of certain locations.

VI. Conclusion

In this paper, we explored the recognition of indoor locations from egocentric videos. A state-of-the-art object detector, trained separately on three object sets, was applied on videos to extract objects for five detection thresholds. We classified these detections with CNN and LSTM to infer locations and evaluated the effect of the adjustable hyper-parameters on the classification performance. We found that the choice of object set affects the relevance of the detections to the location classification task and the detection threshold their amount and quality. One important discovery was that less, but more reliable object information hurts the classification results, whereas more, albeit less confident information, improves them. The comparison between CNN and LSTM had a clear winner in LSTM, which exploit the sequential nature of the data and are able to mitigate the effects of erroneous detections.

Acknowledgment

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. Additionally, we would like to thank the authors of [21] for providing the location annotations for the ADL dataset.

References

[1] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1894–1903, 2016.

[2] R. Yonetani, K. M. Kitani, and Y. Sato, "Recognizing micro-actions and reactions from paired egocentric videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2629–2638, 2016.

[3] D. Damen, T. Leelasawassuk, and W. Mayol-Cuevas, "You-Do, I-Learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance," *Computer Vision and Image Understanding*, vol. 149, pp. 98–112, Aug. 2016.

[4] K. S. Kretch, J. M. Franchak, and K. E. Adolph, "Crawling and walking infants see the world differently," *Child development*, vol. 85, no. 4, pp. 1503–1518, 2014.

[5] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2847–2854, 2012.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] A. Furnari, G. M. Farinella, and S. Battiato, "Temporal segmentation of egocentric videos to highlight personal locations of interest," in *European Conference on Computer Vision*, pp. 474–489, Springer, 2016.

[8] A. Furnari, G. M. Farinella, and S. Battiato, "Recognizing Personal Locations From Egocentric Videos," *IEEE Transactions on Human-Machine Systems*, pp. 1–13, 2016.

[9] K. Nakamura, S. Yeung, A. Alahi, and L. Fei-Fei, "Jointly Learning Energy Expenditures and Activities Using Egocentric Multimodal Signals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6817–6826, July 2017.

[10] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization.," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, p. 7, 2012.

[11] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3281–3288, 2011.

[12] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," in *European Conference on Computer Vision*, pp. 314–327, Springer, 2012.

[13] Y. Poleg, C. Arora, and S. Peleg, "Temporal segmentation of egocentric videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2537–2544, 2014.

[14] A. Betancourt, N. Daz-Rodrguez, E. Barakova, L. Marcenaro, M. Rauterberg, and C. Regazzoni, "Unsupervised understanding of location and illumination changes in egocentric videos," *Pervasive and Mobile Computing*, vol. 40, pp. 414–429, Sept. 2017.

[15] H. Altwaijry, M. Moghimi, and S. Belongie, "Recognizing locations with google glass: A case study," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 167–174, IEEE, 2014.

[16] N. Lee, C. Kim, W. Choi, M. Pyeon, and Y. Kim, "Development of indoor localization system using a mobile data acquisition platform and BoW image matching," *KSCE Journal of Civil Engineering*, vol. 21, pp. 418–430, Jan. 2017.

[17] G. Lu, Y. Yan, N. Sebe, and C. Kambhamettu, "Indoor localization via multi-view images and videos," *Computer Vision and Image Understanding*, vol. 161, pp. 145–160, Aug. 2017.

[18] K. Qian, W. Zhao, Z. Ma, J. Ma, X. Ma, and H. Yu, "A Wearable-assisted Localization and Inspection Guidance System using Egocentric Stereo Cameras," *IEEE Sensors Journal*, pp. 1–1, 2017.

[19] V. Dovgalecs, R. Mgret, and Y. Berthoumieu, "Multiple Feature Fusion Based on Co-Training Approach and Time Regularization for Place Classification in Wearable Video," *Advances in Multimedia*, vol. 2013, pp. 1–22, 2013.

[20] G. Vaca-Castano, S. Das, and J. P. Sousa, "Improving egocentric vision of daily activities," in *Image Processing (ICIP), 2015 IEEE International Conference on*, pp. 2562–2566, IEEE, 2015.

[21] G. Vaca-Castano, S. Das, J. P. Sousa, N. D. Lobo, and M. Shah, "Improved scene identification and object detection on egocentric vision of daily activities," *Computer Vision and Image Understanding*, vol. 156, pp. 92–103, Mar. 2017.

[22] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 2222–2232, Oct. 2017. arXiv: 1503.04069.

[23] N. Y. Hammerla, S. Halloran, and T. Pltz, "Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 1533–1540, AAAI Press, 2016.

[24] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 2342–2350, 2015.

[25] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," *arXiv preprint arXiv:1506.02078*, 2015.

[26] D. Mishkin, N. Sergievskiy, and J. Matas, "Systematic evaluation of convolution neural network advances on the Imagenet," *Computer Vision and Image Understanding*, vol. 161, pp. 11–19, Aug. 2017.

[27] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6517–6525, July 2017.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

[30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Dec. 2014. arXiv: 1412.6980.