CH-004

Activity recognition based on egocentric object detection

Saptarshi Sinha　Hiroki Ohashi　Mitsuhiro Okada　Katsuyuki Nakamura　Takuto Sato

1 Introduction

Human Activity Recognition(HAR) has become an important area of research for its wide applications in healthcare, surveillance and industry[18]. This research focuses on HAR's promising prospects in industry where it can be used to support human labours in various industrial tasks. Majority of the past researches in HAR achieve good performances using third person activity video obtained from a fixed camera[22] [9] [10]. But in industries, many of the field maintainance tasks need the workers to move quite a lot. In that case wearable based HAR seems to be more suitable than the fixed camera based solutions. Wearable based HAR can again be developed using either wearable motion sensors [3] [20] [18] [19] or wearable cameras[1][24]. Wearable motion sensors are known to perform well in HAR. But they don't take the first person vision into account which can be accounted for only by wearable cameras. Past researches[1] have used first person visual data to achieve good performances in egocentric HAR but they use big datasets such as GTEA [15] and Epic Kitchens [24]. Unfortunately, it is not always possible to have enough labelled data in industries because it is time expensive and costly. Therefore, the target of this research is to develop an efficient wearable camera based HAR technique which can also handle situations where limited training data are available for activities.

In limited training data circumstances, due to overfitting problems it is not possible to train deep HAR models to automatically select important information from the egocentric video. In such cases, it is important to feed the model with video features that are actually important for HAR. Therefore, a very important step towards successful deployment of HAR in such situations is to determine what features from the egocentric video actually helps the activity recognition. Hamed et al. [16] and Fathi et al. [15] have shown that detecting hand-object interaction from the first person video is very important for activity classification. Also, object information has been proven to be important for first person activity classification[1], but how much fine grained object information is useful is still an open area of research.

In this research, we make an experimental investigation of the effect of object information in HAR. We divide each video frame into grids and detect objects from them. We then selectively feed object information to the activity classifier. By varying the number of grids, we controlled the preciseness of the detected object location and checked how it affected HAR.

With extensive experiments on a pseudo maintainance task data, we showed that there is an optimum point to how much precise object location actually helps in HAR. Both too finegrained object location and too course object location can harm HAR. In addition, we confirmed that the fact that object information helps HAR [1] holds true in the limited training data case too.

2 Related Works

Human activity recognition has been an active area of research in most of the computer science communities since the 1980s due to it's variety of applications in real life. The traditional action recognition was majorly done from the third person view where the camera stays fixed at one position. Earlier people used hand crafted features from video for action recognition. The video was used as a spatio-temporal volume to extract certain interest points using methods like HOG (Histogram of Gradients), SIFT(Spatially Invariant Feature Transform) [5], 3D-SIFT [4] and 3D-HOG [6] which were used to represent the actions. Dollar et al.[7] proposed to characterize cuboids around the spatio-temporal feature points and extract their locations and types for action recognition. The current state of the art in hand crafted features [11] use iDTs to handle temporal contents and the spatial contents of an action video separately. Thus blindly extending a 2D method to a 3D method (like SIFT to 3D-SIFT [4] might not be helpful. Though iDT provides a good performance in activity recognition, it becomes computationally expensive on large datasets. The main drawback of using handcrafted features is that they are mainly low-level features derived from the action video and hence might be insufficient to represent actions.

But now that powerful extensive resources like GPUs (Graphics Processing Unit) are available, deep neural networks (DNN) can be trained to automatically learn the action representations from the action data themselves provided sufficient training data are available. DNN have been used to estimate human pose both in image [12] and video[23]. Neural Networks have also been used to extract spatial and temporal features from adjacent video frames in 3D-Convnets [8], C3D [9] and deep ConvNets [10] and further identify actions based on the features.

But all of the above methods only deal with action recognition from the third person viewpoint and their performance drops on first person action recognition. Nowadays with the advent of Google glass [1], Gopro [2], Microsoft Sensecam[3] and Panasonic wearable camera[4], it has become possible to cap-

[1]Google glass. https://www.google.com/glass/start/
[2]Gopro. http://gopro.com/
[3]Microsoft sensecam http://research.microsoft.com/en-us/um/cambridge/projects/sensecam/
[4]Panasonic HXA1H. https://panasonic.jp/wearable/p-db/HX-A1H.html

Figure 1: Wearable camera HX-A1H (left) and a snapshot taken by the camera (right)

ture video from first person view. Because the first person point of view is now available, egocentric activity recognition have gained popularity in various applications such as worklog entry, sports, law enforcement etc. But dealing with egocentric or first person videos is difficult as unlike before, the view field of the first person keeps on changing with the motion of the person. Spriggs et al.[14] proposes to use temporal segmentation of the egocentric video and use it along with IMU sensor data for recognizing different activities in the kitchen. But they fail to exploit the human object interactions for the action classification. Daily human activity mainly depends on the various objects they interact with. Based on this hypothesis, Hamed et. al [16] tries to recognize daily human activities by identifying the interacting objects. Fathi et. al [15] classified objects to be active objects, only if the objects were in interaction with the first person's hands. Therefore, Fathi et al. [15] divide an egocentric video into the background, hands and active objects without any prior knowledge of the location of the objects and classifies the action based on the object. Yong et. al [17] proposed a model to predict important regions in a frame based on their temporal frequency and their closeness to the center of the frame. Their model was based on the idea that if the camera moves with the person's head, then the center of the frame must follow the person's gaze. But they require a lot of training data to train the model and hence it becomes difficult to use it in real world.

This paper primarily focuses on activity recognition from first person video but unlike any of the prior works, we create our HAR model from limited training data which is the real world scenario as discussed in section 1.

3 Data Collection

Since our final target is to support human workers in the field maintenance, in which workers have to move around a lot, we utilize wearable sensors for capturing the activities. As the advantage of utilizing video was confirmed in a previous study [21], we decided to use an egocentric camera.

The camera we use is Panasonic's wearable camera HX-A1H (figure 1), which is reasonably light (45g). We collected video data of size 640 × 360 at 30 FPS.

Using the above-mentioned camera, we defined a road-bike maintenance task to simulate a real maintenance task. The snapshots taken in the data collection experiment are shown in figure 2.

In the data collection, we performed 19 different maintenance activities as summarized in table 1. The duration of activities ranges from very short ones (˜1 sec, such as "check reflector" and "check bottle") to relatively longer ones (˜30 secs, such as "inflate tire" and "wipe frame"). It takes about 5 minutes to go through all the maintenance activities (hereinafter we call this 1 *round*). We conducted 12 rounds, but there were errors in data collection in 3 rounds. Hence, we used data from 9 rounds for the following development and evaluation.

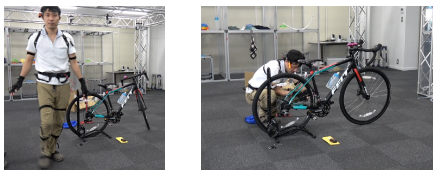


Figure 2: Snapshots at the data collection

4 Proposed Method

4.1 Preliminary

As we know from our everyday experiences, there is an inherent relation between the objects present and the activities performed in a scene. For the task of activity recognition from first person view, the knowledge of the objects being interacted with can be helpful in narrowing down the list of probable activities performed in the scene. For example if we know that there is a bicycle involved in a task, then the first probable activities that come into our minds are 'riding a bicycle' or 'bicycle maintenance' or 'buying/selling a bicycle'. Its highly improbable that a bicycle will be present in activities like 'football game' or 'working in a research lab'. Therefore it is evident that the object information is very important for recognizing an activity.

But again an object in a particular task scene has different properties like location, size and identity. The different properties of the objects play their individual roles in the activity recognition. The object size in an egocentric video might tell us something about the distance of the object from the first person like if it's too small then probably it is irrelevant to the activity being performed. But again there are exceptions to the above assumptions for example if the object is actually small in size like a bicycle bell. Similarly the object location in the frame tells us whether the person is gazing at the object or not.

4.2 Object Detection

For this study of HAR, we try to recognize the various sub-activities involved in a bicycle maintenance task as listed in table 1. We list the various objects that are present during those sub-activities in the table 2. Based on table 2, we group together 16 objects that are relevant to the bicycle mainatainance task —chain lock, key, tire, handle, saddle, pedal, valve, inflator, reflector, water bottle, bell, light, blue cloth, yellow cloth, oil bottle, gloved hand. We also add keyboard and monitor as objects being relevant to the background activity class.

Table 1: Activities in the road-bike maintenance

#	Name of the activity	Explanation
1	unlock	Unlock the key
2	check handle -back and forth	Confirm the handle is not loose by turning it 90 degrees and try to shake it
3	check handle -left and right	Confirm the handle is not loose by holding the front tire by knees and try to shake it
4	lift and drop bicycle	Confirm there is no strange sound when dropping the front tire from 20cm height
5	check break margin	Confirm there is more than 15cm between the handle grip and the break lever
6	check break function	Confirm the break is correctly functioning
7	check saddle	Confirm the saddle is not loose by trying to shake
8	check pedal	Confirm the pedal is not loose by trying to shake it
9	check lever	Confirm the quick levers are clamped
10	check wire	Confirm the wires are not cut
11	check tire valve	Confirm the valves of the tires are closed
12	check tire pressure	Confirm the tire pressure is enough
13	inflate tire	Inflate tire if necessary
14	check reflector	Confirm the reflectors are attached
15	check bottle	Confirm there is a water bottle
16	check bell	Confirm the bell correctly rings
17	check light	Confirm the light can be switched on
18	wipe frame	Wipe the frame by a dry cloth
19	lubricate chain	Lubricate the chain
20	background (bg)	Activities not belonging to the above 19 classes

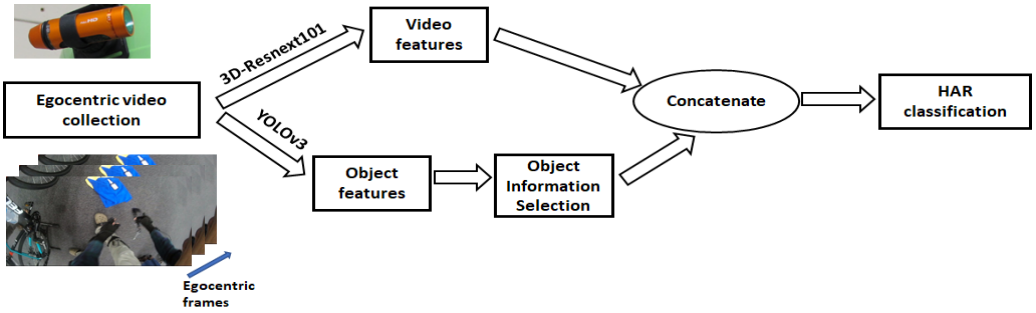


Figure 3: Overview of human activity recognition using egocentric video

To extract information about these objects from the frames, we use a state of the art object detection model called YOLOv3 (You Only Look Once) [2][5]. YOLOv3 is chosen over other detection models because it gives a robust and acceptable performance over a wide range of datasets like MS-COCO, VOC etc. at an acceptable speed.

YOLOv3 detects objects in three different scales. The detection is done by applying 1×1 detection kernel to feature maps of three different sizes at three different places in the network. Each detection kernel has a dimension of $1 \times 1 \times B \times (5 + C)$ where B stands for the number of bounding boxes predicted per cell on the feature map and C is the total number of object classes. For each of the bounding boxes, 4 box coordinates namely the height, width, center point abscissa and ordinate, 1 object confidence score, and C conditional object class probabilities are predicted. The c^{th} conditional object class probability for a bounding box is the predicted conditional probability of the detected object belonging to the c^{th} object class given that there is an object in the bounding box. In this research, the value of C is 18.

The YOLOv3 is pretrained with $B = 3$ and $C = 80$ on the MS-COCO dataset whose images were reshaped to $416 \times 416 \times 3$. For the image size of $416 \times 416 \times 3$, the detection kernel is applied on three feature maps of sizes $13 \times 13, 26 \times 26$ and 52×52. While the 13×13 feature maps are responsible for detecting big objects, the 26×26 and 52×52 feature maps are responsible for the medium and small sized objects respectively. Hence it can be said that image is divided into $13 \times 13, 26 \times 26$ and 52×52 grids successively and for each gridsize, we get 3 bounding box prediction per grid.

The pretrained YOLOv3 was finetuned to detect the 18 objects defined earlier from our bicycle datasets. We just finetuned the last few layers of the YOLOv3. We resized our egocentric video frames to $416 \times 416 \times 3$ to prevent finetuning the input layers.

4.3 Object Information Selection

When fed with $416 \times 416 \times 3$ dimensional frames, the output of the last layer of YOLOv3 is a 2D matrix of dimension

[5]YOLOv3 network architecture. https://www.cyberailab.com/home/a-closer-look-at-YOLOv3

Table 2: The table enlists the various different activities and objects involved in a standard bicycle maintenance task.

Activity number	Activity	Objects involved
1	background	monitor, keyboard
2	unlock cycle	chain lock, key, tire, gloved hand
3	check handle-back and forth	handle, gloved hand
4	check handle-to and fro	handle, gloved hand
5	lift and drop bicycle	handle, tire, gloved hand
6	checking brake margin	handle, gloved hand
7	checking brake function	handle, tire, gloved hand
8	check saddle	saddle, gloved hand
9	check pedal	pedal, tire, gloved hand
10	check tire valve	tire, valve, gloved hand
11	check tire pressure	tire, gloved hand
12	inflate tire	tire, inflator, valve, gloved hand
13	check reflector	tire, reflector, gloved hand
14	check water bottle	water bottle
15	check bell	bell, gloved hand
16	chech light	light, gloved hand
17	wipe frame	blue cloth, yellow cloth, gloved hand
18	lubricate chain	oil bottle, gloved hand

$(((13*13)+(26*26)+(52*52))*3)\times(4+1+18) = 10647\times 23$. We hypothesize that for the purpose of activity recognition, the exact location of an object inside a grid is not very important. Since YOLOv3 detects the objects grid wise, much of the location information of the objects is preserved in the grid separations. So based on this hypothesis, we ignore the predicted bounding box coordinates for the detected objects.

When an image I is divided into 13×13 grids, suppose w and h are the width and height of a single grid $G_{13\times 13}$ respectively. Similarly when the same image is divided into 26×26 grids, the width and height of a single grid $G_{26\times 26}$ become $\frac{w}{2}$ and $\frac{h}{2}$ respectively. For the area inside a $G_{13\times 13}$ grid, we get a 2×2 $G_{26\times 26}$ grids and a 4×4 $G_{52\times 52}$ grids where $G_{52\times 52}$ is a single grid when the image I is divided into 52×52 grids. Therefore for that area, we get 3 predicted bounding boxes from $G_{13\times 13}$, 12 boxes from the 2×2 $G_{26\times 26}$ and 48 boxes from the 4×4 $G_{52\times 52}$. Hence for an area covered by a single $G_{13\times 13}$, we get $3+12+48 = 63$ bounding box predictions in total.

The object confidence score for the detected bounding box b in grid (i,j) from frame f can be assumed to be the probability with which the bounding box contains an object and can be denoted as $(P^f_{i,j,b}(object))$. The c^{th} conditional object class probability for the same bounding box can be denoted as $P^f_{i,j,b}(class = c|object)$ where $1\leq c\leq 18$. The object class probability for class c for detected bounding box b in grid (i,j) from frame f can be calculated using the conditional object probability and object confidence score as follows :-

$$P^f_{i,j,b}(class = c) = P^f_{i,j,b}(object)\times P^f_{i,j,b}(class = c|object) \quad (1)$$

For the purpose of activity recognition, the temporal information among the adjacent frames holds great importance. People have tried to incorporate the temporal information via various methods like LSTM(Long Short-Term Memory), 3D convolution etc to find improved results.

Here we propose a method to involve the temporal information while extracting the object information. Instead of feeding one frame at a time to the YOLOv3, we use a sliding window to feed n consecutive frames together and we get an output tensor of dimension $n\times 10647\times 23$. As discussed earlier, for an area covered by a $G_{13\times 13}$, we get 63 bounding box predictions per frame. We calculate the object features O for frame f' as

$$O^{f'}_{i,j,c} = \max_{f'\leq f\leq f'+n, 1\leq b\leq B} P^f_{i,j,b}(class = c) \quad (2)$$

where B is 63 as discussed earlier.

Similar to equation 4, we also calculate the minimum, mean and standard deviation and stack them together in the third dimension.

4.4 Feature Extraction via 3D-ResNeXt

We also use a 3D-ResNeXt101 [22] model pretrained on Kinetics dataset to extract video features from groups of adjacent frames. The 2048 dimensional output of the average pool layer preceding the last layer of the ResNeXt was used as video features. Then the video features were concatenated with the object features from YOLOv3 by finding the closest timestamps.

4.5 Activity classification

For video data from each of the rounds, we create the integrated features by concatenating the ResNeXt video features and the object features. As we have limited amount of data for the activities, it is not possible for us to train a classifier network on the extracted features. Therefore for evaluation, we randomly choose some of the captured videos as our training data while the others are our testing data. Using the information of the labeled training data as our refernence, our task is to correctly recognize each of the unlabeled activity in the testing data. For classification, we use nearest neighbor (NN) classification on Euclidean distance between the integrated data of the training and testing data. Furthermore, due to the different sizes of the ResNeXt video features

and the object features, we give them different weights for calculating the Euclidean distance. Therefore, our distance formula is changed to

$$dist = (w_O * d(O_{training}, O_{testing})^2 + w_R * d(R_{training}, R_{testing})^2)^{1/2} \quad (3)$$

where $d(x, y)$ is the Euclidean distance between x and y, $O_{training}$ and $R_{training}$ are the object and ResNeXt video features for the training data respectively.

5 Experiments

For HAR evaluation, we used the video data from 9 rounds and computed the integrated features as proposed for each of them. We used 60-fold cross validation for evaluation. In each fold, the videos were divided into 7 training data and 2 testing data randomly and the classification was done as discussed in section 4.5. For our experiments, w_v is chosen to be 0.005 while w_R is chosen to be 1. We calculated the precision and recall for each activity class over all the folds. The overall precision and recall were calculated as the weighted average of the classwise precision and recall. The weight corresponding to an activity was proportional to the frequency of occurence of the activity in the test data. Firstly we show here the effect of varying the amount of object location on HAR performance. Next we confirm by our results that having object information for HAR helps even in the limited data scenario. Finally we show the object detection performance results.

5.1 Importance of object location in HAR

The object information in an image or frame has mainly 2 parts which are object location and object identity. As we are ignoring the bounding box data, our only source of object location is from the grid locations on the image. Therefore if the image is divided into higher number of grids, the grids become finer and we obtain more precise object location. To check how far object location is important for HAR, we varied the number of grids in the frames and checked the corresponding HAR performance. To vary the number of grids, we considered a square sliding window of size $e \times e$ and slid it over the 13×13 grid space with a stride s. The final output features O were calculated as

$$O^{f'}_{i',j',c} = \max_{i' \le i \le i'+e, j' \le j \le j'+e, f' \le f \le f'+n, 1 \le b \le B} P^{f}_{i,j,b}(class = c) \quad (4)$$

where $1 \le i, j \le \frac{13-e}{s} + 1$. The minimum, mean and standard deviation are also calculated similarly. If x is chosen as 4 and s is chosen as 1, the final output object feature O is a tensor of size $10 \times 10 \times 72$. We varied the value of e and s to get different grid resolutions for the object features. The extracted object features were then intergrated with the video features from ResNeXt101 to be used for activity recognition.

As can be seen from the figure 4, the HAR performance is highest when the number of grids is chosen as 13×13

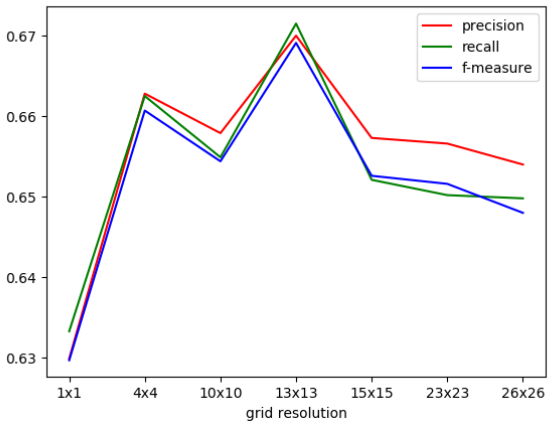


Figure 4: Dependence of HAR performance on grid size

but the performance drops as the number of grids becomes higher or lower. 1×1 grids is an extreme case where the whole frame is considered as one single grid and in this case, all of object location information is lost. These results imply that too little object location information is harmful for HAR task but too much of it is also not helpful. When a person repeats an activity, the relevant objects might not be in the same position like the last time. In such cases, too precise object location will be harmful. Again when an activity is performed, objects irrelevant to that activity but relevant to some other activity might appear in the scene. For example, during inflating a bicycle tire, the only relevant objects are the tire, hand, inflator and the valve but objects like saddle might be appearing in the same frame. In cases as above, if we ignore the object location too much, we might prioritize the wrong objects over the important ones leading to misclassification.

5.2 Evaluation of HAR model

We made a comparison of the overall HAR precision and recall for three different cases in table 3. It can be seen that integrating object information with the resnext video features achieves better performance than using either of them alone for the activity classification. We have also shown the confusion matrices for HAR evaluation using only resnext features and using our proposed method in tables 4 and 5 respectively. If we compare the two confusion matrices, it can be seen that using integrated features has helped in improving the precision and recall for most of the activity classes like background, unlock, check handle back and forth, lift and drop bicycle, check pedal etc. This is reasonable as the objects involved in an activity helps to provide additional information about the activity and therefore helps in correct classification. But for some of the activities like check saddle, check pressure and check reflector, performance drops on using integrated features while for activities like check bottle and check bell, there is no change in the performance. The

Table 3: Comparison of HAR performance

HAR Methods	precision	recall	f-measure
Using only video features from ResNeXt101	0.6564	0.6545	0.6534
Using only object features from YOLOv3	0.6274	0.5580	0.5523
Using integrated features	**0.6700**	**0.6715**	**0.6691**

Table 4: Confusion matrix for HAR evaluation using only ResNeXt101 video features

	bg	unlock	handle-bf	handle-lr	lift & drop	b-margin	b-func	saddle	pedal	lever	wire	valve	pressure	inflate	reflector	bottle	bell	light	wipe	lubricate	total	recall
bg	**[28010]**	292	357	213	96	241	713	202	73	186	560	109	200	778	485	156	145	157	390	1484	34847	0.8038
unlock	452	**[784]**	0	14	30	37	49	29	24	13	49	0	12	79	1	3	0	4	115	85	1780	0.4404
handle-bf	481	14	**[498]**	16	0	11	72	9	4	20	24	2	47	18	43	0	0	21	0	0	1280	0.3891
handle-lr	168	0	0	**[361]**	6	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	537	0.6723
lift & drop	327	10	38	1	**[169]**	0	40	0	0	9	63	0	0	0	20	0	0	5	0	18	700	0.2414
b-margin	251	23	13	11	13	**[213]**	95	0	9	16	18	0	42	101	58	18	0	16	1	12	910	0.2341
b-func	794	51	149	0	10	92	**[764]**	36	14	56	102	15	5	100	30	25	5	46	80	67	2441	0.3130
saddle	119	0	0	0	0	0	23	**[94]**	0	1	0	0	7	27	0	0	7	0	0	18	296	0.3176
pedal	77	0	2	0	0	0	36	0	**[106]**	32	0	0	0	77	0	0	0	0	52	0	382	0.2775
lever	231	0	3	0	0	5	44	5	59	**[151]**	82	27	3	23	39	1	0	0	16	37	726	0.2080
wire	592	35	51	4	21	96	159	12	14	80	**[249]**	26	56	120	41	0	4	17	149	21	1747	0.1425
valve	195	0	0	5	0	0	8	0	2	18	23	**[143]**	25	80	11	0	10	0	14	51	585	0.2444
pressure	408	0	22	0	2	26	26	13	0	0	47	19	**[44]**	18	0	16	0	13	3	28	685	0.0642
inflate	1240	109	189	16	0	16	177	38	183	149	124	6	103	**[8172]**	47	1	0	1	326	388	11285	0.7241
reflector	375	0	3	12	0	0	38	2	0	17	29	14	5	0	**[93]**	19	0	0	25	6	638	0.1458
bottle	122	0	0	2	0	0	5	0	0	0	0	0	6	0	14	**[0]**	0	0	0	0	149	0.0
bell	90	3	12	0	0	0	3	33	0	0	3	0	0	0	0	1	**[0]**	10	0	0	155	0.0
light	346	15	13	0	22	65	13	18	0	0	31	0	5	0	11	3	6	**[187]**	5	3	743	0.2517
wipe	870	116	12	31	0	27	133	3	28	8	122	25	0	297	52	3	0	12	**[1185]**	290	3214	0.3687
lubricate	813	89	15	0	0	12	55	0	1	0	33	16	19	159	24	0	0	0	206	**[2959]**	4401	0.6723
total	35961	1541	1377	686	369	841	2453	494	517	756	1560	402	579	10049	969	246	177	489	2568	5467	67501	
precision	0.7789	0.5088	0.3617	0.5262	0.4580	0.2533	0.3115	0.1903	0.2050	0.1997	0.1596	0.3557	0.0760	0.8132	0.096	0.0	0.0	0.3824	0.4614	0.5412		

possible reasons for the above are :-

- The YOLOv3 is not finetuned well enough to detect certain objects confidently.
- When repeating the same activities, the objects might not be in the same location. NN classifier might not be able to give good results in that case for our proposed idea.
- For activities that involve only visual interactions with the objects like check bottle and check bell, our model fails to know which object is the main object in the activity as those frames include a number of objects.

For the future work, we want to include some bounding box information in the action recognition task as the sizes of the bounding boxes might be helpful in determining the object the person is looking at.

5.3 Qualitative and quantitative evaluation of object detection

For the fine-tuning of YOLOv3, we collected egocentric video data focusing mainly on the 22 objects defined in Table 2 and annotated the video for our ground truth. We evaluated the performance of finetuned YOLOv3 using randomly selected frames from the activity video, which were not used for finetuning. Not all the predictions by YOLOv3 are qualified to be the final object detections.For evaluation, we set an object confidence threshold of 0.6 so that any bounding box predictions with object confidence below 0.6 is ruled out. The resulting predictions go through non-maximum suppression(NMS). During NMS, all the remaining bounding boxes are selected one by one in the decreasing order of their object confidences. any lesser confident bounding box having same predicted object class as the selected box and overlapping too much with the selected box is suppressed. The overlap between the bounding boxes is measured in terms of Intersection over Union (IOU) which is calculated as

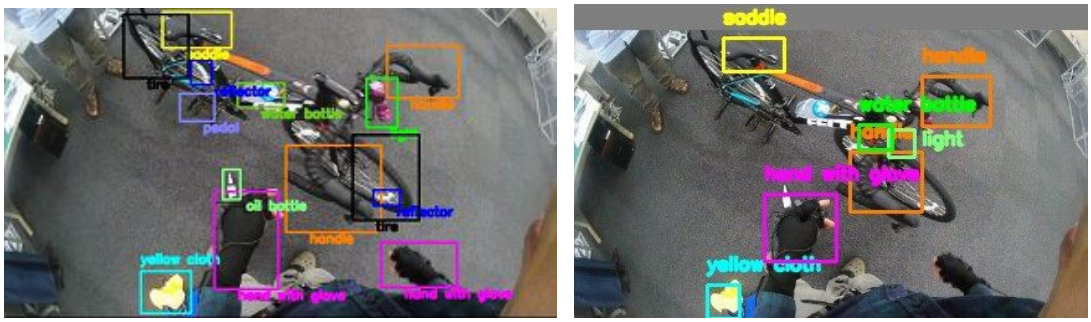
$$IOU(BB1, BB2) = \frac{Area(BB1 \cap BB2)}{Area(BB1 \cup BB2)} \quad (5)$$

where BB stands for bounding box. If two predicted bounding boxes are classified to the same object class and their IOU is more than the IOU threshold of 0.4, we rule out the bounding box with the lower object confidence. After all filtering, we get our final detections for the frames. Visually our object detection model seems to perform well. From figure 5, it is clear that we are able to detect most of the objects successfully. But unfortunately some of the prominent objects do go undetected, namely tires, oil bottle, reflector and pedal. On the other hand, objects like water bottle and light are detected quite far from their actual locations.

A quantitative investigation was done to note which classes were detected well and which classes were not detected at all. An object is said to be correctly detected in a frame if the corresponding predicted bounding box has an IOU of over 0.5 with one of the ground truth bounding boxes and their labels also match. For each object class, the precision and recall are calculated. The results are tabulated in table 6. As can be seen, the YOLOv3 detects some of the objects well while it shows very low performance for many objects. Mainly the objects that are really easy to be occluded remain undetected by the YOLOv3 model. One of the reasons might be the

Table 5: Confusion matrix for HAR evaluation using our proposed method

	bg	unlock	handle-bf	handle-lr	lift & drop	b-margin	b-func	saddle	pedal	lever	wire	valve	pressure	inflate	reflector	bottle	bell	light	wipe	lubricate	total	recall
bg	**[28462]**	263	343	210	94	220	674	210	67	170	527	122	213	719	466	143	135	162	382	1122	34704	0.8201
unlock	466	**[776]**	0	14	22	24	43	40	13	18	37	0	12	57	3	6	0	4	112	68	1715	0.4525
handle-bf	433	12	**[541]**	13	0	15	67	6	4	23	21	2	32	23	25	0	0	21	0	0	1238	0.4370
handle-lr	157	0	4	**[356]**	9	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	528	0.6742
lift&drop	308	10	41	1	**[192]**	0	43	0	0	10	53	0	0	0	15	0	0	1	0	18	692	0.2775
b-margin	219	21	8	13	7	**[210]**	106	0	11	13	20	0	36	78	45	18	0	13	1	13	832	0.2524
b-fund	816	46	133	0	11	92	**[776]**	24	13	51	104	15	3	96	28	28	1	36	68	51	2392	0.3244
saddle	129	0	0	0	0	0	20	**[92]**	0	1	0	0	10	24	0	0	5	0	0	9	290	0.3172
pedal	58	0	2	0	0	0	41	0	**[109]**	35	0	0	0	85	0	0	0	0	52	0	382	0.2853
lever	224	0	5	0	0	1	31	1	45	**[148]**	72	30	1	20	33	1	0	0	15	33	660	0.2242
wire	596	41	44	2	20	85	139	10	15	89	**[244]**	34	72	112	34	2	2	12	168	12	1733	0.1408
valve	222	0	0	3	0	0	4	0	2	15	27	**[194]**	32	73	13	0	13	0	12	46	656	0.2957
pressure	407	0	29	0	2	30	24	15	0	0	44	22	**[42]**	20	0	20	0	11	3	25	694	0.0605
inflate	1103	126	158	12	0	13	143	48	156	134	99	8	87	**[8281]**	59	1	0	3	275	309	11015	0.7518
reflector	364	0	5	11	0	3	38	4	0	16	32	22	3	0	**[80]**	20	0	0	23	5	626	0.1278
bottle	132	0	0	2	0	1	1	1	0	0	0	0	6	0	11	**[0]**	0	0	0	0	154	0.0
bell	96	0	11	0	0	0	3	33	0	0	2	0	0	0	0	1	**[0]**	12	0	0	158	0.0
light	327	13	13	0	31	54	11	20	0	0	31	0	4	4	13	1	5	**[192]**	3	0	722	0.2659
wipe	884	93	15	39	0	20	100	5	32	6	118	32	0	275	61	6	0	14	**[1218]**	251	3169	.3843
lubricate	846	91	12	0	0	14	44	0	1	0	29	15	22	166	25	0	0	0	238	**[2963]**	4466	0.6635
total	36249	1492	1364	676	388	782	2308	509	468	729	1461	496	575	10033	911	247	161	481	2571	4925	66826	
precision	0.7852	0.5201	0.3966	0.5266	0.4948	0.2685	0.3362	0.1807	0.2329	0.2030	0.1670	0.3911	0.0730	0.8254	0.0878	0.0	0.0	0.3992	0.4737	0.6016		



(a) Input frame with ground truth bounding boxes

(b) Object detections by YOLOv3

Figure 5: visual analysis of YOLOv3 performance

Table 6: Precision and recall of YOLOv3 for each object class.

Objects	precision	recall
chain lock	0.4362	0.1189
key	0.0	0.0
handle	0.2407	0.0790
tire	0.0033	0.0035
saddle	0.247	0.06
pedal	0.071	0.035
bell	0.016	0.0054
light	0.1185	0.0532
reflector	0.0069	0.00157
water bottle	0.1475	0.059
blue cloth	0.0496	0.0108
yellow cloth	0.5671	0.229
oil bottle	0.024	0.0055
valve	0	0
inflator	0.1079	0.020
monitor	0.2532	0.1578
keyboard	0.6222	0.30528
gloved hand	0.3993	0.1272

very limited amount of ground truth annotations available for some of the object classes which might lead to insufficient fine-tuning.

6 Conclusions

This research presented a novel way of extracting object features from egocentric video data and integrate it with video features extracted rom ResNeXt101 to enhance the performance of the human-activity-recognition model. We made a study of the significance of object location in action recognition and found that there is an optimum point to the amount of object location information useful for HAR. Furthermore, our experimental results confirm that the object information do help in activity recognition and we were able to outperform the ResNeXt features based activity detection.

References

[1] M. Ma, H. Fan, K. M. Kitani, "Going Deeper into First-Person Activity Recognition", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 1894-1903, 2016.

[2] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement", arXiv preprint arXiv:1804.02767, 2018.

[3] Ohashi H., Al-Naser M., Ahmed S., Nakamura K., Sato T., Dengel A., "Attributes' Importance for Zero-Shot Pose-Classification Based on Wearable Sensors", Sensors 2018, 18, 2485.

[4] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition", In ACMMM, 2007.

[5] Hassaan Ali Qazi, Umar Jahangir, Bilal M Yousuf, Aqib Noor, "Human Action Recognition Using SIFT and HOG method", International Conference on Information and Communication Technologies, 2017.

[6] Alexander Klaser, Marcin Marszałek, Cordelia Schmid, "A Spatio-Temporal Descriptor Based on 3D-Gradients", In BMVC, 2008.

[7] Piotr Dollar, Vincent Rabaud, Garrison Cottrell, Serge Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features", IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005.

[8] Shuiwang Ji, Wei Xu, Ming Yang, Kai Yu, "3D Convolutional Neural Networks for Human Action Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 1, January 2013.

[9] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks", In ICCV,2015.

[10] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, Li Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks", IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[11] Heng Wang, Cordelia Schmid, "Action Recognition with Improved Trajectories", ICCV - IEEE International Conference on Computer Vision, Dec 2013, Sydney, Australia. IEEE, pp.3551-3558, 2013.

[12] Arjun Jain, Jonathan Tompson, Mykhaylo Andriluka, Graham W. Taylor, Christoph Bregler, "Learning Human Pose Estimation Features with Convolutional Networks", In ICLR, 2014.

[13] Jake Snell, Kevin Swersky, Richard Zemel, "Prototypical Networks for Few-shot Learning",NIPS 2017.

[14] Ekaterina H. Spriggs, Fernando De La Torre, Martial Hebert, "Temporal Segmentation and Activity Classification from First-person Sensing", IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. 17-24. 10.1109/CVPRW.2009.5204354, 2012.

[15] Alireza Fathi, Xiaofeng Ren, James M. Rehg, "Learning to Recognize Objects in Egocentric Activities", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,Pages 3281-3288, 2011.

[16] Hamed Pirsiavash, Deva Ramanan, "Detecting Activities of Daily Living in First-Person Camera Views", IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2847-2854. 10.1109/CVPR.2012.6248010, 2012.

[17] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman, "Discovering Important People and Objects for Egocentric Video Summarization", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[18] Lara, Oscar D and Labrador, Miguel A, "A survey on human activity recognition using wearable sensors", IEEE Communications Surveys & Tutorials, vol: 15, no: 3, pages: 1192-1209, 2013.

[19] Ohashi, Hiroki and Al-Nasser, M and Ahmed, Sheraz and Akiyama, Takayuki and Sato, Takuto and Nguyen, Phong and Nakamura, Katsuyuki and Dengel, Andreas, "Augmenting Wearable Sensor Data with Physical Constraint for DNN-Based Human-Action Recognition", ICML Times Series Workshop, pages: 6-11, 2017.

[20] Al-Naser, Mohammad and Ohashi, Hiroki and Ahmed, Sheraz and Nakamura, Katsuyuki and Akiyama, Takayuki and Sato, Takuto and Nguyen, Phong and Dengel, Andreas, "Hierarchical Model for Zero-shot Activity Recognition Using Wearable Sensors", International Conference on Agents and Artificial Intelligence (ICAART), 2017.

[21] Nakamura, Katsuyuki and Yeung, Serena and Alahi, Alexandre and Fei-Fei, Li, "Jointly learning energy expenditures and activities using egocentric multimodal signals", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages: 1868-1877, 2017.

[22] Kensho Hara, Hirokatsu Kataoka, Yutaka Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2018, pp. 6546-6555.

[23] Pavllo, Dario and Feichtenhofer, Christoph and Grangier, David and Auli, Michael, "3D human pose estimation in video with temporal convolutions and semi- supervised training", Conference on Computer Vision and Pattern Recognition (CVPR), June, 2019.

[24] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, Michael Wray,"Scaling Egocentric Vision: The EPIC- KITCHENS Dataset",European Conference on Computer Vision (ECCV), 2018.