Ground-truthing Large Human Behavior Monitoring Datasets

Tehreem Qasim[a], *Robert B. Fisher*[b], *Naeem Bhatti*[a]
[a]*Department of Electronics, Quaid i Azam University, Pakistan.*
[b]*School of Informatics University of Edinburgh, UK.*
tehreemqasim@ele.qau.edu.pk, rbf@inf.ed.ac.uk, nbhatti@qau.edu.pk

Abstract

We present a groundtruthing approach which is applicable to large video datasets collected for studying people's behavior, and which are recorded at a low frame per second (fps) rate. Groundtruthing a large dataset manually is a time consuming task and is prone to errors. The proposed approach is semi-automated (using a combination of deepnet and traditional image analysis) to minimize human labeler's interaction with the video frames. The framework employs mask-rcnn as a people counter followed by human assisted semi-automated tests to correct the wrong labels. Subsequently, a bounding box extraction algorithm is used which is fully automated for frames with a single person and semi-automated for frames with two or more people. We also propose a methodology for anomaly detection i.e., collapse on table or floor. Behavior recognition is performed by using a fine-tuned alexnet convolutional neural network. The people detection and behavior analysis components of the framework are primarily designed to help reduce human labor in ground-truthing so that minimal human involvement is required. They are not meant to be employed as fully automated state-of-the-art systems. The proposed approach is validated on a new dataset presented in this paper, containing human activity in an indoor office environment and recorded at 1 fps as well as an indoor video sequence recorded at 15 fps. Experimental results show a significant reduction in human labor involved in the process of ground-truthing i.e., the number of potential clicks for office dataset was reduced by 99.2% and for the additional test video by 99.7%.

1 Introduction

Computer vision based semi-automated groundtruthing is an active area of research since manually annotating large datasets is a time consuming process [1, 2, 3].In this paper, we present a groundtruthing framework which is capable of performing tasks like human detection and behavior analysis of video captured at a low fps (which creates problems with large displacements of targets between frames). The block diagram of the proposed groundtruthing framework is shown in Fig. 1. First, people counting is performed to estimate if there are 0, 1 or more than one person in the scene. This is done by using mask-rcnn [4] which gives an initial estimate of the number of people in the room followed by some semi-automated human assisted checks to correct the mask-rcnn results. Bounding box extraction is then performed which is automated for single person frames and semi-automated for two or more people frames. We further perform anomaly detection i.e. fall on ground etc. by detecting prolonged inactivity and behavior analysis by fine tuning an alexnet CNN [5]. The annotation framework is applied to a new dataset containing indoor office CCTV footage as well as an additional indoor video. It is to be emphasized that the different components of the framework are designed to reduce manual clicks on video frames (specifically recorded at low fps) during the process of ground-truthing. Fully automated people detection and human action recognition are beyond the scope of this research.

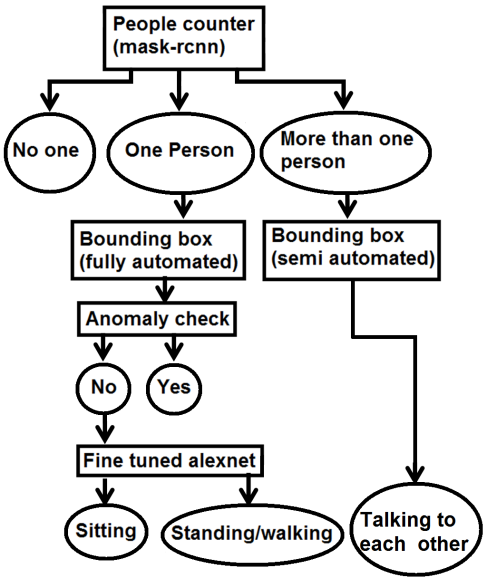


Fig. 1: Block diagram of the groundtruthing framework.

About the dataset: A new dataset[1] is recorded with frame resolution of 720 × 1280 in 4 different offices over 20 days (office 1: day 1-12, office 2: day 13-14, office 3: day 15-17 and office 4: day 18-20). The dataset contains 456,715 frames in total. The number of frames in each office with 0,1 and 2 people is given in Table 1. Sample frames from each office are shown in each row of Fig. 2. The dataset is interesting because it provides a basis for developing methods for long-term monitoring of a few individuals e.g., elderly in a home environment. Sometimes elderly people fall due to an accident or deteriorating health condition. Automated monitoring techniques can be of assistance in such situations.

Fig. 2: Sample frames from each office.

Fig. 3: Sample frames from a home video. Top: infrared frames. Bottom: RGB frames

[1] Dataset is available at the following link; http://homepages.inf.ed.ac.uk/rbf/OFFICEDATA/

Tab. 1: Number of people in each office.

Office	Zero person frames	Single person frames	Two or more people frames	Total
01	104,572	74,038	58,041	236,651
02	46,307	5,670	2,744	54,721
03	50,414	25,780	1,434	77,628
04	48,662	28,622	10,431	87,715
Total	249,955	134,110	72,650	456,715

Additional test video: To test the generality of the proposed approach, we also applied it to an additional indoor video sequence with frame resolution of 1080 × 1920. The video contains both infrared and color frames. The video frames contain no one in the room first in 3,056 frames followed by one person. A second person is also present in 928 frames. Sample frames are shown in Fig. 3.

2 Related Work

Traditional annotation of groundtruth images/videos employs manual labeling [6, 7, 8]. Recently, many researchers have explored semi-automatic groundtruth annotation techniques to reduce the labor that goes into manually creating groundtruth for a dataset [9, 10, 11, 12]. These approaches employ computer vision techniques to automate a significant amount of work required for the task of groundtruthing. In [9], a semi-automatic segmentation tool iSeg is presented which uses polygons for pixel-wise annotation. Boom et al. [1] propose a clustering technique based on Kullback-Liebler divergence and Pyramid histograms to annotate fish images. The clusters are then refined by human labelers.

Wu et al. [2] propose a groundtruthing tool SAGTA to annotate pedestrians in video sequences. First, bounding boxes are manually drawn around the pedestrians in the key frames. In the frames between the key frames, estimation is performed using interpolation by assuming 3D linear motion. The ground-truthing tool provided by Matlab [13] also includes temporal interpolation between key frames, besides people detection and tracking. Authors in [14] perform groundtruth estimation for soccer videos which includes foreground segmentation and tracking followed by manual validation and corrections. Tylecek et al. [10] propose a methodology for semantic

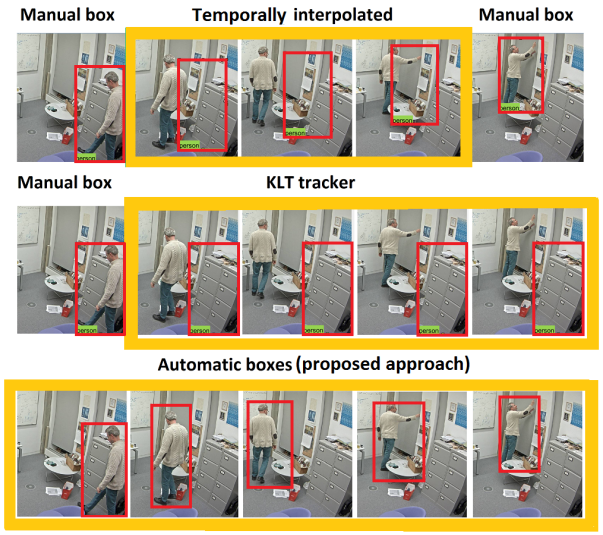


Fig. 4: Day 1, frames 9,215 to 9,219. Top row: First and last frame are manually annotated. Three intermediate frames are temporally interpolated. Middle row: KLT tracker results for four frames after a manually annotated frame. Bottom row: Proposed bounding box method.

annotation of images captured using moving cameras. The proposed approach employs manual corrections by the human labeler. Bianco et al. [3] propose an interactive Video Annotation Tool (iVAT) which supports manual, semi-automatic and automatic annotations by utilizing various detection algorithms. In [11] authors introduce Video Tracking and Behavior Annotation Tool (ViTBAT) which assists in both individual and group target annotations in videos. The approach proposed in [15] requires initial manual annotations for a few video frames which then help VOS algorithm [16] perform segmentation in the rest of the video frames.

In summary, there are existing methods for accelerating ground-truthing of human behavior; however, they assume that the frame rate is fast enough and that inter-frame change is small. In the case of the research presented here, we consider the case of a low frame rate in which the observed humans can move considerable distances between consecutive frames. An example scenario is shown in Fig. 4 in the top row, where the matlab ground-truthing tool fails to perform correct temporal interpolation in consecutive frames and middle row where it fails to perform tracking using the built in KLT tracker [17] due to the low fps.

Similar problems arise in action recognition and anomaly detection by

using temporal analysis based techniques which mainly rely on optical flow [18, 19, 20, 21], which is suitable for high fps situations when changes between consecutive frames are not abrupt. To make our framework suitable for low fps situations, we extract motion information by using the difference of gradient magnitude of two consecutive frames. The motion information is utilized for people counting, bounding box extraction and anomaly detection. We perform behavior labeling by training alexnet CNN with individual frames, hence utilizing only the spatial information in each frame. Experimental results for a new dataset recorded at 1 fps and a video recorded at 15 fps indicate that the proposed approach is able to correctly detect people and perform behavior recognition.

3 Proposed Approach

In the following, we discuss the components of the proposed approach in detail (the discussion in this section pertains to the office dataset.)

3.1 People Counter

We use use a pre-trained mask-rcnn [4] as a people counter followed by our own bounding box extraction algorithm which requires a count of people in each frame. Originally, mask-rcnn can build a box and a mask around a human in a video frame. However, due to challenging aspects of the dataset used in the experiments which include highly cluttered environment, large variations in people's appearance and at times heavy occlusions, mask-rcnn does not always detect each person correctly. Errors include missing a person entirely and sometimes making more bounding boxes than the actual number of people. Due to this reason, we first only count the number of people detected in each frame by using mask-rcnn followed by some checks to correct the mask-rcnn errors. Subsequently, our own bounding box method is used which requires a correct count of people in each frame.

Fig. 5 shows some cropped example frames from each office (row 1: office 1 and 2, row 2: office 3 and 4) where mask-rcnn didn't detect a person but our method based on people counting followed by our bounding box algorithm, correctly built the bounding boxes.

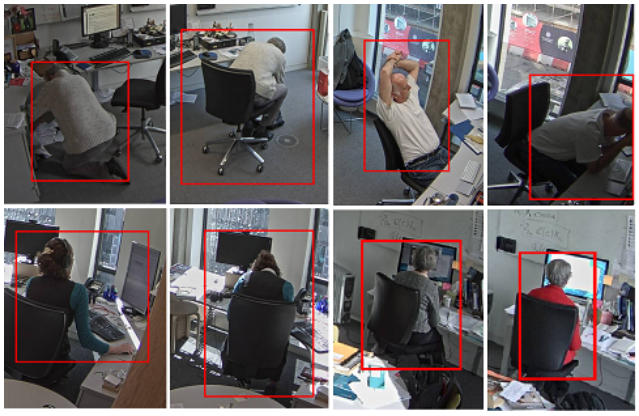
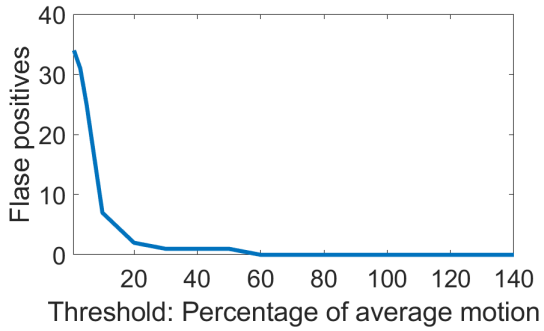


Fig. 5: Examples of frames where mask-rcnn did not detect a person, but our method did. Top row: Day 12 frame 5,651, day 12 frame 11,134, day 13 frame 15,273, day 14 frame 17,022. Bottom row: Day 15 frame 3,971, day 15 frame 5,698, day 18 frame 1,896, day 20 frame 13,659 .

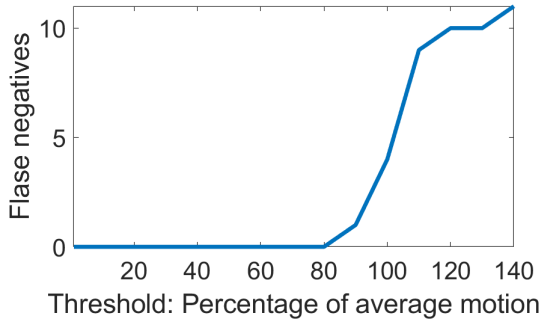
Fig. 6: Motion area (light green) around each office door.

To detect and correct errors in the count of people by mask-rcnn, we developed a test that asks for human validation upon changes in the number of people in an office. To reduce manual corrections of false detected state changes, we introduce some checks to auto-correct the errors. These checks are based on motion information which is extracted by taking the absolute difference of the gradient magnitude of consecutive frames. The resultant difference image is passed through a median and a wiener filter to avoid motion detection due to noise (using medfilt2 and wiener2 matlab functions). The pixels whose difference is larger than zero are considered as motion pixels. The following checks based on motion thresholds automatically bypass

state changes (without human labeler involvement) which are logically least likely in absence of significant motion in room. Our algorithm bypasses the erroneous change in number of people from frame *n-1* to frame *n* as follows. A single frame transition from 0 to 1 (previously 0 people were detected and now 1 person is detected) is automatically bypassed and the label 0 indicating no person in room is retained if the total number of motion pixels is less than 63% of the average motion when a person walks (which is 8,000 pixels for this frame size). If the motion pixels are more than 63% (we justify the choice of the threshold below) when mask-rcnn count label goes from 0 to 1, then there is a chance that a human being entered the room, though these motion pixels could also be caused by reflection in the glass around doors and windows when a person walks by outside near the door. Hence, in case of more than 63% motion pixels, the human labeler is asked by the algorithm for verification. The purpose of this check is to automatically bypass situations where mask-rcnn wrongly detected someone in room, but in actual fact there was no one in the room and hence there was no significant motion. Thus, the human labeler does not have to interact with all the wrong state changes which are logically erroneous. A single frame transition from 1 to 0 (previously 1 person was detected and now 0 people are detected) is bypassed if there are less than 55% of average motion pixels when a person walks (7,000 motion pixels for this frame size). Also, in the next 10 frames, there should be at least two frames with zero people and zero motion. This shows that there is large motion as the person leaves the room followed by frames with zero motion. A single frame transition from 1 to 2 (previously there was 1 person and now 2 people are detected) is bypassed if the total motion pixels are less than 63% of average motion pixels when a person walks, or if motion pixels outside the previous frame's bounding box are less 4% of average bounding box size when a person is sitting (3,000 motion pixels at this frame size). Bounding box extraction is explained in subsection 2. This is because logically, when a second person enters the room, there is large motion as they enter as well as significant motion outside the bounding box around the single person in the previous frame. A single frame transition from 2 to 1 (previously 2 people were detected and now 1 person is detected) is bypassed if motion pixels are less than 39% of average motion pixels during walking (5,000 motion pixels at this frame size) and if there isn't any motion in the door region highlighted in green and shown separately for each office in Fig. 6. These door regions in each office commonly contain motion pixels when a person passes through the door. The smaller box used in office 4 door is because there is a table and a chair close to the door. Sometimes, when a person is sitting in the chair, there are motion pixels in the lower half of the door region even when a person isn't leaving the room.



(a)



(b)

Fig. 7: Impact of threshold on, (a) false positives, (b) false negatives.

All other changes in the count of people in the room are passed on to the human annotator for verification. It is worth mentioning that the threshold values used in this paper are not to be used very strictly. For the purpose of illustration, we performed experiments on the threshold values set to detect and correct errors in state transition from 0 to 1 on day 18 video (as is mentioned earlier, in the final experiments for ground-truthing, the value

Tab. 2: Mask-rcnn and yolo results for day 6 video.

Errors	Mask-rcnn	Yolo	Mask-rcnn results automatically corrected	Mask-rcnn results manually corrected
One person falsely detected in empty room	2	0	2	0
Single person not detected	4	4,378	4	0
Two people wrongly detected when a single person is in the room	49	2	48	1
One person detected when two people are in room	218	1,723	216	2

was set to 63%). We analyzed the impact of different threshold values on false positives (i.e., an erroneous change in count from 0 to 1 was passed onto the human labeler) and false negatives (i.e., a human actually entered the room but the change from 0 to 1 was bypassed and count of 0 was retained). As we can see in Fig. 7 (a), after threshold value of 20%, the false positives reduce significantly and reduce to zero after 60%. Similarly, as shown in Fig. 7 (b), up to 80%, false negatives are zero. Hence setting a threshold anywhere in the region from 60% to 80% will yield zero false positives and zero false negatives. Although different levels of camera zoom and frame rates may require different thresholds, we see that the algorithm is not sensitive to the precise threshold setting. In Table. 2, we show the detected errors in the mask-rcnn results in the day 6 video. Also, we have included yolo [22] results for comparison. We observe that on several occasions mask-rcnn did not make correct detections. We also note that the number of errors in the yolo results is even larger. This justifies the use of our method which builds upon mask-rcnn to correct the errors in counts prior to using the bounding box method. Our method required manual correction of only 3 mask-rcnn errors

out of the total 273. All other errors were automatically corrected by using the motion thresholds based checks.

3.2 Bounding box extraction

The algorithm to build a bounding box around a person's body is fully automated in the case of a single person in the room (total 249,955 single person frames) and semi-automated for two or more people. We discuss both methods in the following.

Single person bounding box method

The single person bounding box algorithm is based on motion information which is explained in the previous section. We divide an image into 25×40 square cells and retain only the cells that contain any motion pixels and build a bounding box around them. We also perform additional operations to resolve the encountered problems namely, an older replica of a person in the difference frame, motion in shadows and reflections as well as too little motion leading to too small a bounding box. We discuss these issues in the following. As can be seen in the second row of Fig. 8 (a) the difference operation leads to a replica of a human from the previous frame appearing in the current frame difference image. To resolve this issue, we perform a *logical and* operation between the difference image for frame *n-1:n* and the difference image for frame *n:n+1*. This leads to a more accurate mask for the object in the current frame as shown in row 3 of Fig. 8 (a).

In the second column of Fig. 8 (b), we show the isolated small cells that contain motion due to shadows and reflections as the human moves. These motion cells are outside the human body and lead to an inaccurate bounding box. To resolve this problem, our algorithm only retains the largest connected component of motion cells which represents the human body. The first column images of Fig. 8 (b) show that suppressing the smaller connected components helps remove the reflection and shadow motion cells. To solve the third problem, where a human object is stationary or exhibits small motion, our algorithm retains the frame *n-1* bounding box for frame n and beyond as long as there is no motion, or the motion cells are contained within the frame *n-1* bounding box.

More than one person bounding box method:

In the case of two or more people, the motion based bounding box method used for a single person is not feasible because if two people are too close

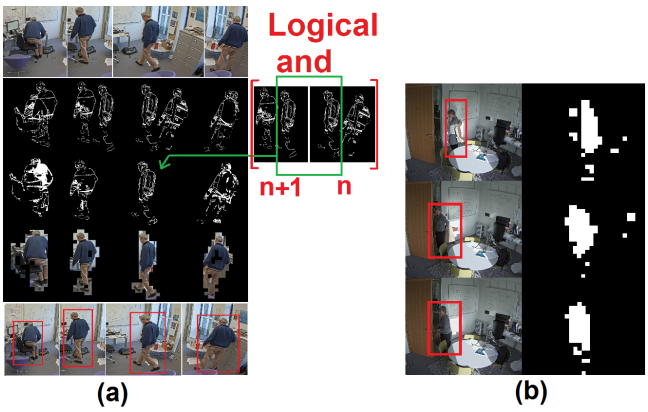


Fig. 8: a) Older replica of a person body problem. First row: original frames, second row: thresholded difference, third row: anded difference frames at n and $n+1$, fourth row: extracted cells, fifth row: bounding boxes. b) Reflections and shadows problem. Left: Bounding boxes. Right: thresholded connected components based on motion cells.

Fig. 9: Two people bounding box extraction. (left) Manually drawn boxes for frame 3,066. (middle) Automatic boxes drawn for frames 3,067-3,078. (right) Another frame needing manual annotation due to excessive movement.

to each other, then a single blob represents the motion of both persons. It's hard to estimate where each person is in the blob. Our two people bounding box method capitalizes on the fact that in the case of two people sitting in chairs or exhibiting little motion, a single bounding box (separately for each person) can represent each person's body as long as their motion is confined to a limited region. The algorithm works by asking for user input to process a single frame or multiple frames. In the case of large motion when a person's position is changing drastically e.g., when a second person enters the room, processing a single frame manually gives an accurate bounding box around each person. However, when each person sits and the position does not change significantly for two or three frames during manual annotation of individual frames, then the option to process multiple frames can be selected. Multiple frames are processed as follows. A rectangular region is defined by drawing a bounding box manually around each person's body. An example is shown in Fig. 9 (left). The algorithm keeps processing frames and records motion history as long there is no significant motion outside the defined region (2.86% of average bounding box around a sitting person, or 2,000 motion pixels for our videos). Once 2,000 or more motion pixels are detected outside the defined region, the algorithm stops processing any further frames and the human annotator is shown the motion history confined within the defined region and requested to manually draw a bounding box. In the example shown in Fig. 9 (middle), bounding boxes are automatically drawn around each person's motion history from frame 3,066 to frame 3,078 (shown in green and blue). In frame 3,079, one of the two people has moved outside the defined region, hence the human labeler manually annotates individual frames until the positions of both people stabilize again. This method can also detect a third person entering the room since motion happens outside the defined region for two people. With this algorithm, the human annotator had to interact with 3,309 (4.55%) frames out of total 72,641 frames with more than 1 person across all 20 days.

For comparison of bounding box quality with mask-rcnn, we manually annotated 100 random frames from Day 6 and computed the average "intersection over union (IOU)" for mask-rcnn and our own bounding box method. The average IOU for mask-rcnn was slightly better i.e. 0.83 while the IOU for our bounding box method was 0.69. Manual inspection of the boxes showed that the boxes produced by mask-rcnn are little tighter because our approach sometimes includes extra changing scenery, in particular in the lower part of a moving office chair. Cropped example frames with bounding boxes are shown in Fig. 10. As a result of this analysis, we use the mask-rcnn boxes when the count and overlap of boxes agree with our proposed method, and otherwise we use our method.

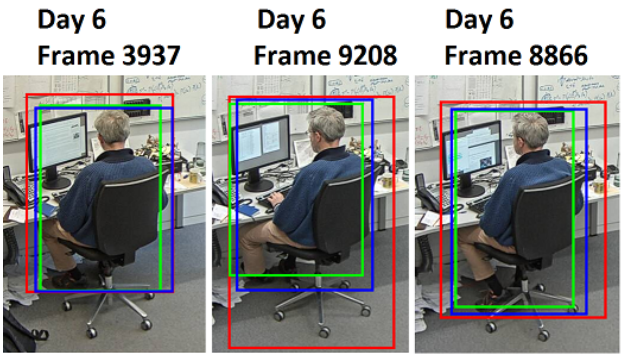


Fig. 10: Comparison of bounding box quality. Blue box: manually done, red box: proposed method, green box: mask-rcnn.

3.3 Anomaly detection

The aim of the anomaly check is to ground-truth a fall event that has occurred when an elderly person is alone in a room and no other person is around (e.g. to see the incident and call for help). So, we do not check for falls when there are multiple people in the room. The anomalies in the dataset are of two types, collapse/fall on the ground and collapse on the table as shown in Fig. 11. They occur in office 1 only. Most methods in the literature for fall detection use optical flow direction or magnitude etc. [18, 23, 24]. Here, they are not effective due to the very low fps. Hence, we base our method on detection of very low/zero motion as a result of the fall. The proposed method learns those regions in frames where low motion happens due to the person staring at the computer or sitting in a normal scenario. These regions are less likely to be part of a fall event onto the ground. To detect falls, we divide each frame into 25×40 cells. We then detect the frames where a fall is potentially detected by "little motion" (defined as less than 100 motion pixels) in 10 consecutive frames. This analysis is only done in cells where a person is detected. For tuning, we analyze days 1,2,3,4,5,12 where there are no fall events and the "little motion" cells are mainly due to sitting or staring at the monitor etc. Each time 10 consecutive frames show "little motion" our algorithm detects the cells where these few motion pixels appear. These are the cells which are likely to contain less motion in a normal situation. We also include cells in the 5×5 cell neighborhood around each normal cell since abnormal low motion in test frames is likely to also appear in nearby cells. We use this analysis to learn where little motion is likely to occur.

The blue region in the Fig. 12 was learned to be a region where low motion is likely due to a normal situation. We then test for low motion in

Fig. 11: Example anomaly frames.

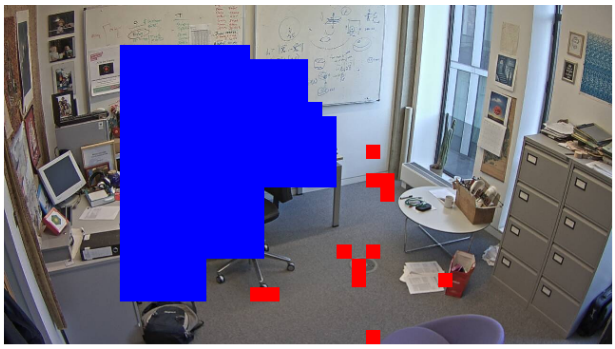


Fig. 12: Learnt normal low motion region (blue) and detected abnormal low motion region due to fall (red).

the rest of the days i.e. days 6,7,8,9,10,11 (days 6 and 11 have anomalous frames). The low motion cells (red cells in Fig. 12) after the fall on floor are different from the normal ones and depict a scenario where a person has collapsed on the ground.

The collapse on table is harder to detect since it occurs while the person is looking at the computer screen or sitting and hence in the region where normally "little motion" occurs. Hence, we include another check to detect these events. These events have a distinctive feature i.e. they happen after large motion (during collapse) and are followed by prolonged low/zero motion. Hence, in 5 consecutive frames (here 5 seconds) if total motion pixels are more than 10 thousand, depicting large motion (large enough for a collapse when a person is sitting in a chair) and then there are 15 consecutive frames with less than 100 motion pixels in each frame indicating prolonged inactivity, then the anomaly due to collapse on the table is detected. These two checks together detect both sorts of anomalous frames and don't yield any false positives. With our anomaly detection approach, the labeler does not have to scan each frame manually and all four anomalous events are detected in an automated way as discussed in section IV. We note that the proposed approach is somewhat heuristic and may cause false positives in other scenarios e.g., when a person sits in an armchair and then falls asleep.

3.4 Behavior Labeling

We assign the label "talking to each other" to the frames with two or more people by default. The frames with a single person are first classified as normal or abnormal (fall event). The normal frames are further classified into two classes i.e., sitting and standing/walking. To differentiate between sitting and standing/walking, we fine-tuned a separate two class alexnet for each office with 100 random frames for each category from each day in an office. To detect and correct the errors in the labels we post-processed all the frames labeled by the two class alexnet. The algorithm requires human labeler's validation each time there is a change in label e.g., from sitting to standing/walking and vice versa. False label changes are manually corrected. Results are discussed in section IV.

4 Results and Discussion

Office dataset: The results of the people counter are summarized in Table 3. Rather than report counts (see Table 1), we report when the number of people in the room ("state") changes because these are fewer and more

Tab. 3: People counter algorithm results (changes in number of people in room).

Office	Manually verified true detected state changes	Manually corrected false detected state changes	Total automatic corrections	Total frames
01	274	213	7,891	236,651
02	19	15	252	54,721
03	99	41	6,251	77,628
04	87	32	8,089	87,715
Total	479	301	22,483	456,715

Tab. 4: Fall event detection results.

Day	Anomaly begins at	Anomaly ends at	Anomaly detected begins at	Anomaly detected ends at
06 (floor)	9,129	9,157	9,141	9,156
11 (table)	1,650	1,672	1,650	1,672
11 (floor)	2,138	2,300	2,152	2,298
11 (table)	16,609	16,730	16,609	16,729

informative. We observe that mask-rcnn despite working well for a large number of frames, requires corrections. In column 2 and column 3, we show true and false detected state changes. Column 4 in the Table 3 shows that by using mask-rcnn in conjunction with our own validation algorithm, a large number of frame labels are corrected automatically with significantly reduced human labeler's involvement.

Fall event detection results are given in Table 4. We observe that our framework is able to detect all of the anomalous events and didn't detect any false positives. There is some lag in detection due to motion in the human body even after the anomalous event begins.

For assessment of the bounding box consistency, 4 human labelers manually drew bounding boxes in 500 randomly selected frames from the dataset. The manual results were then compared with the bounding boxes obtained through the proposed framework by computing IOU. Average IOU of each individual annotator is given in Table 5. We note that the automatically

Tab. 5: IOU results for 500 randomly selected frames.

Annotator No.	Average IOU
Annotator 1	0.719
Annotator 2	0.685
Annotator 3	0.707
Annotator 4	0.741

Tab. 6: Behavior labeling results.

Office	Manually verified true change in label	Manually corrected false change in label	Single person frames
01	333	588	104,572
02	33	160	46,307
03	194	73	50,414
04	145	132	48,662
Total	705	953	249,955

detected bounding boxes are highly consistent with the manual bounding boxes.

The results of behavior labeling for an individual office and the entire dataset collectively are given in Table 6. Despite using no temporal information, the two class alexnet performs well with 953 (0.381%) errors out of total 249,955 single person frames. Finally, we tested 100 random frames in each day video to look for potential undetected people and behavior errors. We found zero errors.

Tests on a separate video: To test the generality of the proposed framework, it was employed for annotation of 15,000 frames of an independent video in a different setting (home living space), with a minor change i.e., to overcome noise, an additional mean filter was applied to video frames besides the wiener and the median filter. To train the two class alexnet, 40 random frames from each class i.e., with a person sitting and standing/walking were used. People counting required 3 manual verifications and 43 manual corrections. Two people bounding box method required interaction with 26 frames out of total 928 frames. Behavior analysis required 3 manual verifications and

3 manual corrections. This shows that the proposed framework generalizes well to other scenes as well.

5 Conclusion

In this paper, we propose a semi-automated approach with the aim to perform groundtruthing (with minimum human labor) of large datasets containing human sedentary activity at a low fps. The proposed approach ground-truth labels people counts, bounding boxes, falls, and activities. Experiments on a new dataset recorded in an indoor office environment at 1 fps show that the number of clicks required for for ground-truthing people counts was were reduced to 0.71% (479 verifications and 301 corrections out of 456,715 frames). Clicks for bounding box extraction were reduced to 0% for frames with a single person (out of 134,110 single person frames with a fully automated method) and 4.02% for frames with two or more people (6,178 box initializations with a semi-automated method, out of 72,650 frames containing two or more people requiring total 153,807 box initializations). Clicks for fall detection were reduced to 0.04% (30 manual annotations out of a search space comprising 74,038 single person frames in office 1). Clicks for behavior labeling were reduced to 0.66% (705 verifications and 953 corrections out of 249,955 single person frames).

In a 15 fps video containing 15,000 frames, clicks for people counting were reduced to 0.30% (3 verifications and 43 corrections). Clicks for correcting one person bounding boxes were reduced to 0% for 11,016 single person frames. Clicks for two people bounding boxes were reduced to 2.80% i.e., 52 manual box initializations out of 928 two people frames requiring 1,856 box initializations. For single person behavior labeling, clicks were reduced to 0.05% (3 verifications and 3 corrections out of total 11,016 single person frames). Though, the main purpose of the framework was to perform the task of ground-truthing, its individual components can be further developed and incorporated into state-of-the-art human detection and behavior recognition systems which can be tested on this dataset (using the generated ground-truth) and which are applicable to livestream videos. In a similar manner, improved components for the subtasks (e.g. mask-rcnn or alexnet), could be substituted and potentially reduce further the number of manual interventions.

References

[1] Bastiaan J Boom, Phoenix X Huang, Jiyin He, and Robert B Fisher. Supporting ground-truth annotation of image datasets using clustering. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1542–1545. IEEE, 2012.

[2] Shuang Wu, Shibao Zheng, Hua Yang, Yawen Fan, Longfei Liang, and Hang Su. Sagta: Semi-automatic ground truth annotation in crowd scenes. In *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2014.

[3] Simone Bianco, Gianluigi Ciocca, Paolo Napoletano, and Raimondo Schettini. An interactive tool for manual, semi-automatic and automatic video annotation. *Computer Vision and Image Understanding*, 131:88–99, 2015.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[7] Xiang Li, Ben Aldridge, Jonathan Rees, and Robert Fisher. Estimating the ground truth from multiple individual segmentations with application to skin lesion segmentation. In *Proc. Medical Image Understanding and Analysis Conference, UK*, volume 1, pages 101–106, 2010.

[8] Xiang Li, Ben Aldridge, Robert Fisher, and Jonathan Rees. Estimating the ground truth from multiple individual segmentations incorporating prior pattern analysis with application to skin lesion segmentation. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1438–1441. IEEE, 2011.

[9] Julius Schöning, Patrick Faion, and Gunther Heidemann. Pixel-wise ground truth annotation in videos. *ICPRAM*, 6:11, 2016.

[10] Radim Tylecek and Robert B Fisher. Consistent semantic annotation of outdoor datasets via 2d/3d label transfer. *Sensors*, 18(7):2249, 2018.

[11] Tewodros A Biresaw, Tahir Nawaz, James Ferryman, and Anthony I Dell. Vitbat: Video tracking and behavior annotation tool. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 295–301. IEEE, 2016.

[12] David Doermann and David Mihalcik. Tools and techniques for video performance evaluation. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 4, pages 167–170. IEEE, 2000.

[13] `https://www.mathworks.com/help/vision/ref/videolabeler-app.html`.

[14] Tiziana D'Orazio, Marco Leo, Nicola Mosca, Paolo Spagnolo, and Pier Luigi Mazzeo. A semi-automatic system for ground truth generation of soccer video sequences. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 559–564. IEEE, 2009.

[15] Amanda Berg, Joakim Johnander, Flavie Durand de Gevigney, Jorgen Ahlberg, and Michael Felsberg. Semi-automatic annotation of objects in visual-thermal video. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[16] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2019.

[17] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.

[18] David Nicholas Olivieri, Iván Gómez Conde, and Xosé Antón Vila Sobrino. Eigenspace-based fall detection and activity recognition from motion templates and machine learning. *Expert Systems with Applications*, 39(5):5935–5945, 2012.

[19] Tehreem Qasim and Naeem Bhatti. A hybrid swarm intelligence based approach for abnormal event detection in crowded environments. *Pattern Recognition Letters*, 128:220–225, 2019.

[20] Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117–127, 2017.

[21] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[23] Xiaomu Luo, Tong Liu, Jun Liu, Xuemei Guo, and Guoli Wang. Design and implementation of a distributed fall detection system based on wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):118, 2012.

[24] Chieh-Ling Huang, E-Liang Chen, and Pau-Choo Chung. Fall detection using modular neural networks with back-projected optical flow. *Biomedical Engineering: Applications, Basis and Communications*, 19(06):415–424, 2007.