Noname manuscript No.
(will be inserted by the editor)

uMoDT: An unobtrusive Multi-occupant Detection and Tracking using robust Kalman filter for real-time activity recognition

Muhammad Asif Razzaq[1], Javier Medina Quero[2], Ian Cleland[3], Chris Nugent[3], Usman Akhtar[1], Hafiz Syed Bilal Ali[1], Ubaid Ur Rehman[1], Sungyoung Lee[1]

Received: date / Accepted: date

Abstract Human activity recognition (HAR) is an important branch of human-centered research. Advances in wearable and unobtrusive technologies offer many opportunities for HAR. While much progress has been made in HAR using wearable technology, it still remains a challenging task using unobtrusive (non-wearable) sensors. This paper investigates detection and tracking of multi-occupant HAR in a smart-home environment, using a novel low-resolution Thermal Vision Sensor (TVS). Specifically, the research presents the development and implementation of a two-step framework, consisting of a Computer Vision (CV) based method to detect and track multiple occupants combined with Convolutional Neural Network (CNN) based HAR. The proposed algorithm uses frame-difference over consecutive frames for occupant detection, a set of morphological operations to refine identified objects, and features are extracted before applying a Kalman filter for tracking. Laterally, a 19-layer CNN architecture is used for HAR and afterward the results from both methods are fused using time interval based sliding window. This approach is evaluated through a series of experiments based on benchmark Thermal Infrared datasets (VOT-TIR2016) and multi-occupant data collected from TVS. Results demonstrate that the proposed framework is capable of detecting and tracking 88.46% of multi-occupants with a classification accuracy of 90.99% for HAR.

Keywords Human activity recognition · Image processing · Object detection · Tracking · Classification

1 Introduction

Over several decades, advances in pervasive computing has offered great promise towards the potential of indoor localization and Human Activity Recognition (HAR) [1]. Over this period, significant research effort has been targeted towards the creation of solutions that can reliably monitor individuals through the use of on-body wearable sensors, dense sensors, and vision sensors [2]. Whilst results utilizing on-body sensors has improved greatly, wearable solutions are often said to be impractical, as they can be difficult to carry or inconvenient to wear continuously [3]. Additionally, vision sensors capable of capturing RGB or grayscale images have been studied intensively within the Computer Vision (CV) domain. The use of cameras, however, raises serious privacy concerns [4].

Recently, researchers have been investigating the potential of deploying unobtrusive, inexpensive and low resolution Thermal Vision Sensors (TVS) for occupant detection and pervasive sensing [5]. Similar to traditional vision based approaches, TVS suffer from same limitations for handling complex object appearances due to shape deformation, low resolution, varying number of objects, pose variation, motion estimation, and object re-identification [6]. TVS do, however, address some of the challenges as they tend to be more robust to illumination changes, can operate even in complete darkness and offer less intrusion on user's privacy [7].

[1] Ubiquitous Computing Lab, Department of Computer Engineering, Kyung Hee University, 1 Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do 446-701, Korea; (asif.razzaq, usman, bilalrizvi, ubaid.rehman, sylee)@oslab.khu.ac.kr
[2] Department of Computer Science, University of Jaén, Campus Las Lagunillas, Jaén 23071, Spain; jmquero@ujaen.es
[3] School of Computing and Mathematics, Ulster University, Jordanstown, BT37 0QB, Northern Ireland, UK; (cd.nugent, i.cleland)@ulster.ac.uk

The majority of research into HAR has focused on single occupant environments. Nevertheless, living environments are usually inhabited by more than one person. Therefore, HAR in the context of multi-occupancy would provide a more practical solution, however, also more challenging. The difficulty with multi-occupant HAR stems from two related challenges in occupant identification, known as data association, and the diversity of human activities.

In CV, object tracking remains one of the most significant research challenges [8, 9]. This becomes even more complex when using TVS for monitoring multi-occupants, as data only corresponds to variation in temperature. Therefore a different strategy is required for identification and re-identification of the occupants [10]. The aforementioned challenges are addressed by proposing and implementing a robust CV-based integrated framework for multi-occupant detection, tracking and HAR based on TVS.

The remainder of the article is organized as follows: Sect. 2 presents a review of related work; Sect. 3 formulates the problem and introduces the proposed framework describing our pragmatic approaches for multi-occupant tracking and HAR; Sect. 4 presents experimental details; Sect. 5 presents both quantitative and qualitative evaluations and comparisons on thermal frame sequence and VOT-TIR2016 benchmarks; Sect. 6 offers concluding remarks with a discussion about future improvements.

2 Related work

Multi-object Tracking (MOT) in CV domain has been studied for decades and has attracted a lot of research attention. It is, however, still far from solved regarding HAR [11]. Many solutions exist for HAR in a controlled environment. These solutions mostly involve the deployment of numerous wearable and pervasive sensors [12], which can lead to increased cost, privacy concerns and more often inconvenience. To alleviate these challenges, attention of the research community has directed to low-cost unobtrusive sensors [13].

TVS are an excellent candidate for pervasive sensing due to their inexpensive nature, portability, limited maintenance requirement and lower privacy issues compared to traditional cameras. Hevesi et al. [14] have illustrated that such a sensing modality can be deployed for indoor HAR and monitoring of sedentary behavior of a single occupant in an office environment. Solutions based on TVS mostly require CV based approaches for locating moving objects by identifying them as a region of interest (ROI) in a frame sequence. Detection of a ROI is deemed as the first step in most CV-based applications [15]. It may involve various techniques such as: (1) thresholding, which yields low accuracy and is of lesser use in current applications [16]; (2) multi-resolution processing which faces challenges for detecting objects during congestion [17]; (3) edge detection which has challenges in deriving an ROI where the shape of object is highly dynamic; (4) inter-frame differencing which uses consecutive frames for detecting an ROI but can only be considered for a sequence of a shorter duration [18]; (5) an optical flow based detection which requires a large number of frames resulting in poor performance; (6) background subtraction which extract objects not belonging to the background, however, this technique requires a static background as an initialization.

Regarding MOT various techniques [19] have also been proposed by the research community. These techniques focus on addressing common challenges such as frequent occlusions, identical appearances, track management and interaction among objects. No single approach currently exists which can address all of these challenges. MOT in any visual tracking system usually involves three functional models [20]: (1) appearance model, which describes the object and distinguishes it from the non-objects; (2) motion model which characterizes the current and predicts the future states of an object by tracking their trajectories; (3) searching strategy which helps to identify and match an object based on the appearance model in a frame sequence.

Motion models have gained significant attention for object state estimation. They operate by producing accurate motion affinity models in a linear motion space, which can be used to predict object position [21]. Thus, it reduces the search space by capturing the dynamic behavior of the object. To solve the linear tracking problem, where continuity of moving objects is not abrupt, Kalman filtering (KF) is often used [22]. This approach can track moving objects using their center of gravity [23]. KF is a linear state-space motion model proved to be an optimal tracker suitable for practical applications. It promises a good compromise between computational complexity and performance for object tracking by utilizing a point-based approach in learning statistical features [24]. It uses identified features and uncertainty information to estimate different states of an object through the successive frames. KF may, however, experience object drifting due to the loss of an object's appearance information in a frame sequence. The object drifting complexities require efficient object refinement schemes to analyze object motion properties leading to proper data association [25]. Yilmaz et al. [21] addressed some of the issues and complexities related to data associations through a joint solution for state estimation. Choi et al. [26] formulated the problem of multi-occupancy and resolved it through multiple target tracking. They merged the problems of HAR and tracking into a single probabilistic graphical model for tracking individual actions. Similarly, an adaptive framework was also proposed by Shen et al. [27] to identify the correct state of the targets. They suggested the use of an adaptive detection algorithm for MOT task to refine the detection targets and minimize the detection errors.

In order to classify Activities of Daily Living (ADLs), it has been observed that CNN have shown superior performance over the traditional Machine Learning (ML) approaches such as Support Vector Machines [28] and feed-forward neural networks [29]. The visual object recognition tasks [30] can be performed over the raw low-resolution TVS frames using CNN, which is easier to train by adjusting a few parameters and inter-layer connections. It extracts meaningful features without requiring domain knowledge and with minimum preprocessing over a stacked sequence of frames [31]. The CNN model has the capability to extract multiple motion features encoded in the adjacent TVS frames for automatic classification of ADLs [32].

The current work is closely related to [4] in which the authors proposed a system for indoor player tracking captured through the thermal camera at a sports arena and pedestrian tracking in a courtyard. Ray et al. [33] proposed a detection algorithm, which does not depend on any prior background knowledge for object detection and also does not require initialization. Similarly, Leira et al. [34] considered the problem of small unmanned aerial vehicles equipped with thermal cameras for real-time target detection and tracking at sea using the KF based technique. Tiwari et al. [35] highlighted the research gaps for video-based HAR. They suggested designing an approach to improve the robustness of the detection and tracking algorithms by increasing the number of occupants, which can be tracked over a sequence.

The purpose of this study is to propose a framework for moving object detection, tracking and classification of ADLs with increased performance using low-resolution thermal video frames. To achieve this goal, an implementation using a KF was devised by building a robust object appearance model with morphological feature refinements. It also involves the Hungarian algorithm for data association per frame [27]. Additionally, this study evaluates the robustness of the integrated framework to detect and track ADLs of the users using low-resolution TVS. For this, the solution was tested using comprehensive experimental analysis drawing quantitative and qualitative comparisons. Robust tracking systems, such as [36], mostly involve an appearance and motion model to track the candidate states of the target. Computational complexity, however, increases proportionally with the increase in the number of targets to be tracked [37]. Therefore, a joint optimization is essential for MOT. Most MOT research focuses on tracking-by-detection methods, however, an extension to it, by classifying the activities may result in boosting the overall effectiveness of these methods.

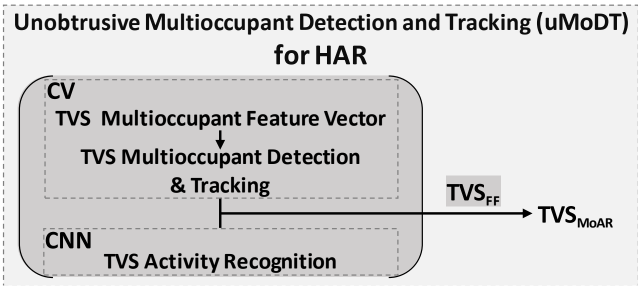

Unobtrusive Multioccupant Detection and Tracking (uMoDT)
for HAR
CV
TVS Multioccupant Feature Vector
TVS Multioccupant Detection
& Tracking
TVS$_{FF}$
TVS$_{MoAR}$
CNN
TVS Activity Recognition

Fig. 1: Overview of proposed solution strategies as *uMoDT* framework

3 Proposed uMoDT framework

This section initially outlines the design challenges before presenting the algorithmic solutions and then detailing the overall framework.

3.1 Overview

The main challenges in CV-based object detection and tracking applications are correct identification of ROI, reliable and efficient handling of moving objects alongwith their inter-frame associations. These challenges, however, become even more complex for interacting multi-objects, which may have erratic movements represented by low-resolution appearances in a frame sequence. For this, an efficient method is required to predict their motion and manage data association [38]. Additionally, recognition of interaction amongst objects and classification of activities is also a computationally intensive task and requires a more robust process. This further requires a trade-off when implementing the above-mentioned methods in a more efficient manner for a complete, coherent and correct detection, tracking and classification of an occupant's activities. To address the aforementioned challenges, as presented in Fig. 1, we propose a unified scalable *unobtrusive Multi-occupant Detection and Tracking* (*uMoDT*) framework,

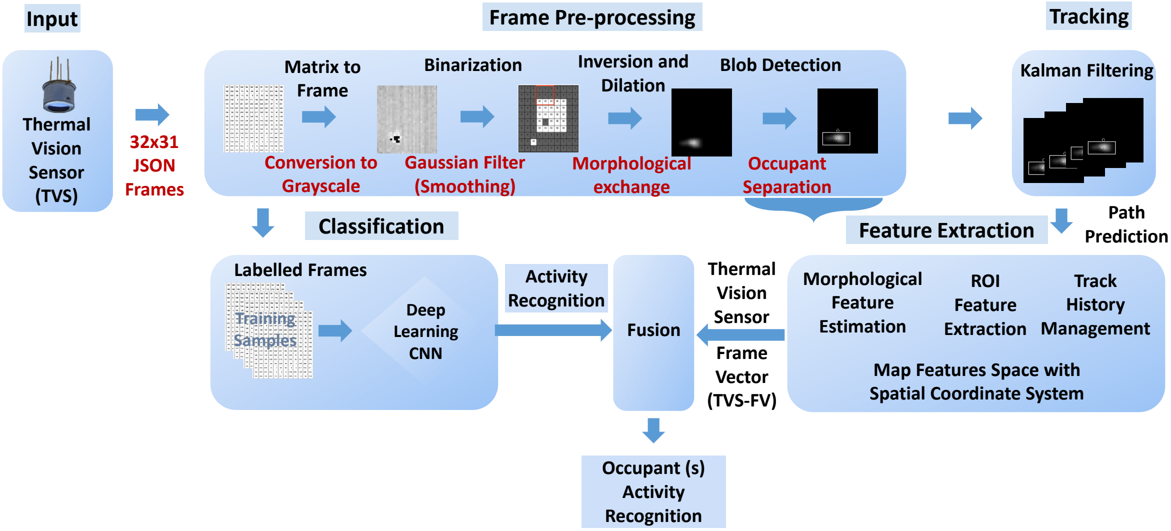


Fig. 2: Proposed unobtrusive Multi-occupant Detection and Tracking (uMoDT) framework for HAR

which detects, tracks and recognizes different indoor activities under multioccupancy using TVS.

The *uMoDT* framework addresses six strategies as described below:

- We propose an online framework, which uses a CV-based algorithm, with improved morphological features, for an automatic multi-target initialization using frame differencing with an optimum threshold.
- We rely on refined morphological characteristics, which ensure efficient detection and tracking accuracy over the dynamic patterns for nonrigid moving targets per-frame.
- We use the Hungarian method for track assignment problem with an approach for maintaining an association history of re-identified tracks of individual moving objects per-frame.
- The proposed framework is validated using a dataset gathered at Smart Environments Research Group (SERG) laboratory from the Ulster University, UK. It proved to be computationally robust and achieves a promising tracking accuracy in comparison with other MOT methods.
- We also demonstrated quantitative evaluations on the publicly available dataset for the VOT-TIR2016 challenge proving the practicality and efficacy of the proposed framework with the state-of-the-art.
- Additionally, we propose to apply a CNN architecture to extract and learn spatial features from multiple successive Thermal Vision Sensor Frame (TVS-F) for individual action recognition.

The focus of the presented work is to simultaneously detect multi-occupants as well as recognize their activities frame-by-frame from TVS. It also requires a solution for resident data association in a smart-home environment, which is accomplished by unifying two different approaches. Firstly, using the CV-based technique, which detects, tracks, and monitors the occupant within the controlled area by observing a robust frame difference between the consecutive frames. Secondly, the CNN layers are invoked by the TVS frame sequence (TVS_F_{seq}), which recognizes the occupant's individual activities such as *Walking*, *Standing*, *Sitting*, *Fall down*. Finally, the recognized activities are associated with each occupant using the proposed Thermal Vision Sensor Feature Fusion (TVS_{FF}) method per frame.

3.2 Computer vision-based occupant detection and tracking

This Section describes the inner details of the proposed framework to detect the presence of multi-occupants in real-time, and track them throughout the duration of TVS_F_{seq} by following them from frame-by-frame. Fig. 2 illustrates the overall *uMoDT* framework with underlying several components, namely *TVS sensor* as an *Input* device, *TVS-F Preprocessing*, *Occupant Tracking*, and *TVS-F Feature Extraction*. These components are connected in series whereas the information flow between subcomponents is discussed further in the following subsections.

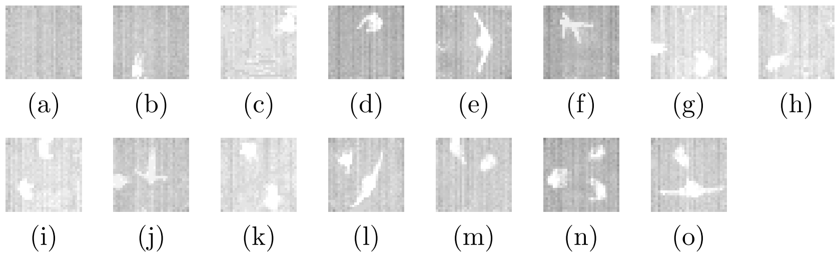


Fig. 3: (a) Empty smart living room. Single occupant activities shown as (b) *Sitting* (c) *Standing* (d) *Walking* (e) *Stretching* (f) *Fall Down*. Multi-occupant activities shown as (g) Two persons *Sitting* (h) One person *Sitting* while other *Standing* (i) One person *Sitting* while other *Walking* (j) One person *Standing* while other *Fall Down* (k) Both persons *Standing* (l) One person *Standing* while other *Stretching* (m) Both persons *Walking* (n) All are *Walking* (o) one person *Walking* while other one *Stretching*

Algorithm 1 TVS-MoFV: Thermal Vision Sensor multi-occupant frame vector algorithm

```
 Input TVS_F: Thermal Vision Sensor grayscale sequence frames;
Output: Multi-occupant Frame Vector $TVS_{MoFV}$.
procedure TVS_MatPreProcessing
    Load $TVS\_F_{seq} \leftarrow \{TVS\_F_1, TVS\_F_2 \dots TVS\_F_n\}$ where i = $\{1, 2, \dots n\}$
    Read Matrix $TVS\_F_{seq}$                                 ▷ Reads sequence of frames $TVS\_F_{seq}$
    for all $TVS\_F_i$ to $TVS\_F_n$ do
        function $Low\_thresholding(TVS\_F_i)$
            $TVS\_F_i - TVS\_F_{i-1} > TVS_{Th}$                ▷ Frame differencing sensitive to threshold
            $\mathbf{B}_n \leftarrow TVS\_F_i$                         ▷ Identify 'n' Occupants as Blobs
            $TVS\_F_i \leftarrow \mathbf{Gauss}_{k,l}(TVS\_F_i)$          ▷ Smoothing by Gaussian blur k=l=3
        end function
        function $morphologicalTVSPreProcessing(TVS\_F_i)$                  ▷ Morphological filtering
            $TVS\_F_i \leftarrow \mathbf{E}_{k_w,k_h}(TVS\_F_i)$              ▷ Erode: width 'w' & height 'h' =8
            $TVS\_F_i \leftarrow \mathbf{D}_{k_w,k_h}(TVS\_F_i)$              ▷ Dilate: width 'w' & height 'h' =8
        end function
        function $Detect\_Contour(TVS\_F_i)$
            $\mathcal{C}nt_n \leftarrow TVS\_F_i$
            Find $\mathcal{C}nt_n Contours$
            for all i = 1 to n do
                $min(\mathbf{B}) < \mathcal{C}nt_i < max(\mathbf{B})$    ▷ $min\_Blob\_Area < ContourArea < max\_Blob\_Area$
                $\mathbf{P}_{x_i,y_i} \leftarrow \mathcal{C}nt_i(p_{x_i}, p_{y_i})$
                $\mathbf{BR}_n \leftarrow boundingrectangle(\mathbf{P}_{x_i,y_i})$          ▷ Assign $Bounding\_Rectangle$
                array [$\mathbf{BR}$] $\leftarrow \mathbf{BR}_n$                           ▷ Populate $Rectangle\_Array$
                $\mathcal{P}_{\square} \leftarrow$ array [$\mathbf{BR}$]                  ▷ GetContourFeatures $Perimeter$
                $\mathcal{A}_n \leftarrow area(\mathbf{P}_{x_i,y_i})$                       ▷ GetContourFeatures $Area$
                $\mathcal{A}_{\blacksquare} \leftarrow$ array [$\mathbf{BR}$]
                $\overline{\mathcal{P}}_{\mathbf{avg}} \leftarrow Average_{pixels}(\mathbf{BR}_n)$    ▷ Compute pixel p, average avg for $Bounding\_Rectangle$
            end for
        end function
    end for
    return $TVS_{MoFV} \leftarrow \left[\mathbf{P}_{x_i,y_i}, \mathcal{P}_{\square}, \mathcal{A}_n, \mathcal{A}_{\blacksquare}, \overline{\mathcal{C}\mathbf{nt_n}}\right]$
end procedure
```

3.2.1 Input frames

In this study, we propose to mount the Heimann HTPA TVS [39] in the ceiling of the smart-home's living room and kitchen at the height of 3m. The monitored space is a quadrilateral area with dimensions 4×3.5m. This setting provides a clear field of view and collects an aerial view of the multi-occupants as seen in Fig. 3. It also overcomes the challenges related to occupant-to-occupant and, occupant-to-scene occlusion, whilst avoiding camera motion and is operative even in complete darkness. The TVS ensures a high degree of user's privacy by capturing low-resolution grayscale TVS_F_{seq} with the dimensions of 32h×31v×1. Each of the 992 pixels correspond to an area within the smart living room and kitchen represented by each pixel value ranging between 0 and 255. This range sets a correspondence of every pixel with an average temperature characteristic to that area. The TVS_F_{seq} is managed by using RESTful HTTP services, which are processed by the server.

3.2.2 Multi-occupant Feature Vector (TVS-MoFV)

The frames represent the presence of heat sources within the TVS_F_{seq}. The characteristics of identified heat sources are calculated by using the proposed Thermal Vision Sensor Multi-occupant Feature Vector (TVS-MoFV) algorithm. It gathers multi-occupant feature vectors in TVS_F_{seq} frame-by-frame. The series of tasks performed by TVS-MoFV are described in Algorithm 1, which are summarized as follows:-

- Converts the JSON 32×31 matrices into the sequence of frames TVS_F_{seq}.
- Segments the TVS_F_n frames in order to detect foreground (multi-occupant) and background (static smart living room or kitchen) per frame.
- Applies the *Low_thresholding* TVS_{Th} function with a background subtraction method sensitive to threshold [40].
- Convolves the TVS-F using Gaussian Kernel $Guass_{k,l}$ for smoothing and reducing noise with the kernel k=l=3.
- Performs morphological filtering and binarization on TVS_F_n to reduce the thermal noise using operations such as Erode $\mathbf{E}_{k_w,k_h}$ and Dilate $\mathbf{D}_{k_w,k_h}$.
- Determines the presence of multi-occupant using connected pixels termed as the contours $\mathcal{C}nt_n$ represented by blobs in the sequence of binary frames TVS_F_n.
- Assigns and encapsulates each identified $\mathcal{C}nt_i$, within the ROI, represented by *Bounding Rectangles* i.e. $\mathbf{BR}_n$.
- Estimates the centroid $\mathbf{P}_{x_i,y_i}$ for the identified $\mathcal{C}nt_i$ surrounded by $\mathbf{BR}_n$, which acts as a pivot for further tracking.
- Computes an array of the morphological feature vector for every TVS_F_i frame, which includes Perimeter $\mathcal{P}_{\square}$, Area $\mathcal{A}_{\blacksquare}$, and Contour Pixel Average $\overline{\mathcal{P}}_{\mathbf{avg}}$ for every $\mathbf{BR}_n$ in the TVS_F_i.

The learned frame vector TVS_{MoFV} from every TVS-F comprises of the morphological states of the detected occupant. These states represent the occupant's thermal area, a center of contour, a perimeter of the bounding box, and the area enclosed within the bounding box encapsulating the occupant. These multiple features become the basis for TVS_{MoAR} with the support of the proposed method TVS_{FF} required for the data association before recognizing and associating individual activities.

3.2.3 Multi-occupant Detection and Tracking (TVS-MoDT)

Algorithm 2 describes the TVS Multi-occupant Detection and Tracking (TVS-MoDT) method to identify, predict, plot, visualize, and maintain the occupant's tracks within TVS_F_{seq}. Some of the key features for this algorithm are summarized as below:-

- The TVS_F_{seq} is read as input simultaneously as in the case of Algorithm 1.
- The detected contours $\mathcal{C}nt_i$ through Algorithm 1 are iterated within TVS_F_{seq} for computing the vector point $\mathcal{V}_p$ responsible for tracking and maintaining the history of the tracks as shown in Line 4-11.
- For every detection $\mathbf{D}$ for $\mathcal{C}nt_i$ the tracks $\mathcal{T}_i$ are initialized as shown in Line 15.
- We used two classical efficient methods, Hungarian method, and KF to handle the occupant's data association and smoother motion refinement with position prediction of the multi-occupant respectively.
- The optimal assignment $\overrightarrow{\mathcal{A}}$ and cost $\mathcal{C}$ computation task for tracks $\mathcal{T}_i^{assign}$ is performed using the Hungarian method.
- We employed KF to generate multi-occupant motion trajectories i.e. estimation and position prediction for the blob representing each of the individual occupants as mentioned in the Line 32.
- The *UpdateKalman* prediction function predicts the position of the occupant based on the history from previous TVS-F whereas the update function rectifies the state of the multi-occupant from the current TVS-F (Lines 39-45).
- Every multi-occupant being tracked is assigned *Tracking ID* ($\mathcal{T}_{id}$) representing tracklets. The morphological features such as position, size and other statistical measurements are also calculated for blob.
- $\mathcal{T}_{id}$ is dynamically assigned (or reassigned) to blobs with rapidly varying sizes. The array with tracking identifiers represents each occupant's motion model and state history.

3.3 CNN-based activity classification

The CNN has been utilized for real-time multi-occupant AR from the TVS_F_{seq}. It is computationally built on five major mathematical functions such as *Convolution*, *Batch Normalization*, *Rectified Linear Unit*

Algorithm 2 TVS-MoDT: Thermal Vision Sensor multi-occupant detection and tracking algorithm

```
Input TVS_F: Thermal Vision Sensor gray-scale frame sequence;
  Output: Multi-occupant tracks $T_{MoDT}$.
procedure TVS_MatPreProcessing
    Load $TVS\_F_{seq} \leftarrow \{TVS\_F_1, TVS\_F_2 \ldots TVS\_F_n\}$ where i = $\{1, 2, \ldots n\}$
    Read Matrix $TVS\_F_{seq}$                                   ▷ Reads Sequence of Frames $TVS\_F_{seq}$
    for all $TVS\_F_i$ to $TVS\_F_n$ do
        function VectorPoint $\mathcal{V}_p(TVS\_F_i, \mathcal{C}nt_n)$          ▷ Detect_VectorPoint
            for all i = 1 to n do
                $\mathcal{P}_c^+ \leftarrow \mathbf{BR}_n \{\mathcal{C}nt_n\}$                        ▷ Iterate $Contours$
                array $[\mathbf{D}] \leftarrow \mathcal{P}_c^+$                              ▷ Array of detections
                $TVS\_F_i \leftarrow$ Draw $(\mathbf{BR}_n, TVS\_F_i)$             ▷ $drawRectangle \leftarrow Contours$
                $TVS\_F_i \leftarrow$ Draw $(\mathcal{P}_c^+, TVS\_F_i)$           ▷ $drawCenterPoint \leftarrow Contours$
            end for
        end function
        function Track $\mathcal{T}_i(\mathcal{C}nt_n, \mathbf{D}, TVS\_F_i)$      ▷ initialize $(NoOfTracks, TrackSize)$
            for all i = 1 to Size ([D]) do
                $\mathcal{T}_i \leftarrow$ new$(\mathcal{T}, \mathbf{D})$
                $Cost\,[i]\,[i] \leftarrow Euclid(\mathcal{T}_i^{pred}, \mathbf{D})$   ▷ Euclidean distance between prediction & detection
                $\mathcal{C} \leftarrow Cost\,[i]\,[j]$
                $\overrightarrow{\mathcal{A}} \leftarrow Vector(Assignment)$
                $\mathcal{T}_i^{assign} \leftarrow HungarianAssignment\ (\mathcal{C}, \overrightarrow{\mathcal{A}})$
                if $(\mathcal{C} > \mathcal{D}_{threshold})$ then                   ▷ Identify $unAssigned\_tracks$
                    $[\mathcal{T}_i^{unassinged}] \leftarrow$ add$(\mathcal{T}_i^{unassinged})$        ▷ Search $Un\_Assigned\_Tracks$
                end if
                if $([\mathcal{TVS\_F}_i^{skipped}] > max_f)$ then
                    $\mathcal{TVS\_F}_i \leftarrow$ remove $(\mathcal{TVS\_F}_i)$             ▷ Remove not detected tracks
                    $\overrightarrow{\mathcal{A}} \leftarrow$ remove $(\overrightarrow{\mathcal{A}_i})$                        ▷ Remove assignments
                end if
                if $(size(\mathbf{D}_i^{unassigned}) > 0)$ then
                    $\mathcal{T}_i \leftarrow$ add$(\mathcal{T}_i, \mathbf{D}_i^{unassigned})$      ▷ Initialize New_Tracks for un_Assigned_Detects
                end if
                $\mathcal{T}_i \leftarrow \mathcal{T}_i^{skipped} > \mathcal{TVS}_{SkippedAllowed}$
                /* Update Kalman for All Detected Contours */
                $\mathcal{TVS} \leftarrow UpdateKalman(TVS\_F_i, \mathbf{D})$        ▷ Predict, Update Kalman Occupant State
                /* Iterate the No of contours, detections in the $\mathcal{TVS\_F}_i$ */
                for all t = 1 to Size $(\overrightarrow{\mathcal{A}})$ do
                    $\mathcal{T}_{id} \leftarrow \mathcal{T}_i(t)$
                    $\mathcal{TVS\_F}_i \leftarrow \mathcal{TVS\_F}_{append}(\mathcal{TVS\_F}_i, \mathcal{T}_{id}, \mathcal{P}_c^+)$      ▷ Draw tracks
                    $[\mathcal{TVS\_F}_i]_{history} \leftarrow \mathcal{TVS\_F}_{append}(\mathcal{TVS\_F}_i, \mathcal{P}_c^+)$   ▷ Contours & Tracks History
                end for
                /* Update $\mathcal{TVS\_F}_i$ with Kalman Prediction and Correction */
                $\mathcal{I}t \leftarrow n\,(\mathcal{C}nt_n)$                              ▷ Number of Contours
                while $\mathcal{I}t.hasnext$ do
                    $TVS\_F_i \leftarrow$ update $(TVS\_F_i, \mathcal{P}_c^+, [\mathcal{TVS\_F}_i]_{history})$      ▷ Kalman Effect
                    $TVS\_F_i \leftarrow$ draw $(\mathcal{P}_c^+, [\mathcal{TVS\_F}_i]_{history})$        ▷ Kalman prediction updation
                    $TVS\_F_i \leftarrow$ draw_line $(\mathcal{P}_c^+, \mathcal{T}_{i-1}, \mathcal{T}_i, [\mathcal{TVS\_F}_i]_{history})$
                end while
            end for
        end function
    end for return $T_{MoDT}$
end procedure
```

(ReLU), *Pooling*, and *Soft-max*. These functions are applied in a hierarchical residual block within an architecture, which provides fully connected layers for processing TVS_F_{seq} to get multi-occupant activity classification output per frame. These are briefly discussed in the following subsections.

3.3.1 Input layer

An input layer for the CNN architecture reads the grayscale TVS_F_{seq} of the fixed dimensionality, requires TVS_F_{Train} to train the model while producing an output $TVS\_F_{labelled}$, representing "n" activities performed by the multi-occupants.

$$TVS\_F_{labelled} \leftarrow \{TVS\_F_{seq}, TVS\_F_{Train}, act_n\}_{CNN} \tag{1}$$

3.3.2 Convolutional layer

The Convolutional Layer is responsible for extracting the pixel-wise features from the input TVS-F. To learn the TVS-F features, the kernel weights are adjusted automatically through back-propagation training. The convolution is obtained by taking dot product (•) between sub-part of the TVS-F and the convolutional kernel K. In response, a feature map f_c is computed by sliding the convolutional kernel over the TVS-F spatially. The output $x_i^{l,j}$ for the l^{th} convolutional layer having the j^{th} feature map on the i^{th} unit can be presented mathematically as:

$$x_i^{l,j} = \sigma\left(b_j + \sum_{a=1}^{m} w_a^j x_{i+a-1}^{l-1,j}\right) \tag{2}$$

where σ is a non-linear mapping, it uses hyperbolic tangent function, $\tanh(\cdot)$ [41].

3.3.3 Batch normalization layer

The input channel x across the mini-batch is normalized $\hat{x_i}$ by the introduction of a batch normalization layer [42]. Normalized activation is computed using mini-batch mean μ_B, standard deviation σ_B^2 for input channel x, and ϵ to provide the numeric stability for mini-batch variances, described as:

$$\hat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3}$$

It increases the performance of CNN training and reduces sensitivity of the neural nets.

3.3.4 ReLU layer

Rectified Linear Unit (ReLU), a nonlinear activation function responsible for introducing a point-wise nonlinearity to the CNN by resolving the vanishing gradient problem [43]. ReLU layer processes an element-wise activation function over each individual input x, wherever the value is less than zero, is set to zero and it also linearly conveys the input for positive inputs described by Eq. 4:

$$f_\tau = ReLU(x_i) = \begin{cases} x_i, x_i \geq 0; \\ 0, x_i < 0; \end{cases} \tag{4}$$

A rectified feature map f_τ is obtained as an outcome.

3.3.5 Max-pooling layer

The max-pooling layer produces compact feature space by taking the sub-samples of f_τ thus reducing the spatial dimensionality and sensitivity of the output. The pooling operation derives maximum value from the set of nearby inputs as mentioned in equation 2, which can also be represented mathematically as [31]:

$$f_i^{l,j} = \max_{r \in R}(x_{i \times T+r}^{l,j}) \tag{5}$$

where R represents pooling size and T as a pooling stride. The soft-max classifier is placed at the final layer for HAR. The TVS-F features obtained from the stacked convolutional and pooling are represented as:

$$f^l = [f_1, f_2, f_3.....f_K] \tag{6}$$

where K represents the number of units learned from the last pooling layer, which acts as a feature map for the soft-max classifier.

3.3.6 Training process

The CNN is trained in a supervised learning fashion by selecting the parameters using *Gradient-based optimization* method. For faster convergence, the *stochastic gradient descent* method is applied [44]. The training process involves a series of steps such as *propagation* and *weight update*. The gradients are computed in the propagation step by using *standard forward* [41] and *back-propagation* algorithms [45], by minimizing the objective function, which is given mathematically as:

$$x_i^l = \sum_j w_{j,i}^{l-1} \sigma(x_i^{l-1}) + b_i^{l-1} \tag{7}$$

where x_i^l represents the output feature and w is the weight vector. The output feature map is passed to every subsequent layer till it reaches the output layer, which is formulated as:

$$\frac{\partial L}{\partial y_{i,j}^{l-1}} = \sum_{a=0}^{m-1} \frac{\partial L}{\partial x_{i-a}^l} \frac{\partial x_{i-a}^l}{\partial y_{i,j}^{l-1}} = \sum_{a=0}^{m-1} \frac{\partial L}{\partial x_{i-a}^l} w_{a,b} \tag{8}$$

It applies *chain-rule* for computing the propagation error and the whole process remains cyclic until the CNN reaches a satisfactory validation state or attains the stopping criterion.

Table 1: List of 16 activities recorded for data collection

Activity ID	Activity Type	Activity Name	No. of Occupants
Act_1	Single	FallDown	1
Act_2, Act_3	Single, Multi	Sitting	1, 2
Act_4	Multi	SittingStanding	2
Act_5	Multi	SittingWalking	2
Act_6, Act_8	Single, Multi	Standing	1, 2
Act_7	Multi	StandingFallDown	2
Act_9	Multi	StandingStretching	2
Act_{10}	Multi	StandingWalking	2
Act_{11}	Single	Stretching	1
$Act_{12}, Act_{15}, Act_{16}$	Single, Mutli	Walking	1, 2, 3
Act_{13}	Multi	WalkingFallDown	2
Act_{14}	Multi	WalkingStretching	2

3.3.7 Classification

The soft-max regression function in the final layer of the neural network leads to the multi-occupant HAR using *TVS-based Activity Recognition* (TVS-AR) method. It normalizes the output, which is computed by fully connected layers, and more often is a combination of a set of positive numbers with their sum equivalent to one, and value ranges between [0 ... 1]. These ranges are further transformed into classification probabilities through the *Classification* layer in the CNN residual block. The *i-th* probability value for soft-max function $p(y_i)$ [46] is computed as:

$$\hat{y_i} = p(y_i) = softmax(x_i) = \frac{exp(x_i)}{\sum_{k=1}^{n} exp(x_k)}, i = 1 \dots N_c \tag{9}$$

The cross-entropy [45] is minimized between the output probability vector $\hat{y}$ and total number of class labels 'y' as follows:

$$E = -\sum_{i=1}^{N_c} (y_i log(\hat{y_i}) + (1 - y_i) log(\hat{y_i})), i = 1 \dots N_c \tag{10}$$

where y_i represents binary indicator if the class label 'c' is correctly classified from the i^{th} neuron and $\hat{y}$ is the predicted probability of the i^{th} class.

4 Experiments

The complete real-time prototype application for our proposed *uMoDT* framework is built for multi-occupant detection, tracking and AR. To demonstrate the functionality of the *uMoDT* framework, we first discuss the dataset and later the implementation insights.

4.1 Dataset

We collected 57,290 frames in a sequence from three healthy male volunteers aging 25±7 [yrs]; height 1.55±0.7 [m] and weight 68±8 [kg]. Each volunteer performed different ADLs individually and collectively in a smart living room over a duration of at least 3 minutes each, reported in Table 1. During the entire collection, the application was neither reparameterized nor recalibrated, which means this setting remained valid for all kind of ADLs performed during this study. Additionally, TVS_F_{seq} was annotated with *LabelImg*, an open source annotation tool [47]. During labeling, multi-occupants were approximated by using bounding rectangles over the subsequent frames by assigning them unique identifiers referred as ground-truth $\mathbf{G}_i$ in the TVS_F_{seq}. This process followed a strict annotation protocol by qualified researchers.

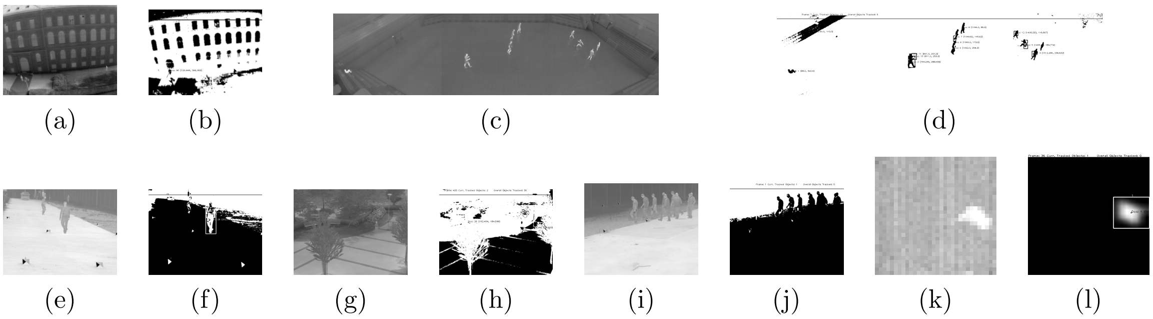
The goal is to quantitatively evaluate the proposed *uMoDT* framework and prove its accuracy and robustness. For this, we tested and compared it, also on five challenging, publicly available annotated sequences from VOT-TIR2016 challenge [48, 49]. These sequences were mostly captured with the help of static FLIR and thermal cameras.

4.2 Implementation details

The proposed *uMoDT* framework, comprising of *TVS-MoFV* (Algorithm 1), *TVS-MoDT* (Algorithm 2) and *TVS-AR* method, was implemented. The former algorithms utilize the Java-based standard libraries OpenCV (an open-source API) [50] while the latter method requires MATLAB interfaces (machine learning toolbox API). The *uMoDT* framework was imple-

Table 2: List of benchmark dataset sequences and their details

ID	Dataset	Sensor	Resolution	Frames	Object	Threshold
1	ETHZ-CLA [51]	FLIR TAU320	324×256	659	Human	115
2	Soccer [4, 48]	3×AXIS Q-1922	1920×480	3,000	Human	120
3	Crouching [48]	FLIR A655SC	640×480	625	Human	125
4	Depthwise Crossing [48]	FLIR A655SC	640×480	858	Human	135
5	Crowd [48]	FLIR Photon 320	640×512	78	Human	110
6	TVS_F_{seq}	Heimann	32×31	57,290	Human	155


(a) (b) (c) (d)

(e) (f) (g) (h) (i) (j) (k) (l)

Fig. 4: Examples of raw Input (I) frames and processed Output (O) frames using proposed framework. (a) & (b) ETHZ-CLA (I&O) (c) & (d) Soccer (I&O) (e) & (f) Crouching (I&O) (g) & (h) Depthwise Crossing (I&O) (i) & (j) Crowd (I&O) (k) & (l) TVS-F (I&O)

mented and evaluated using the PC system equipped with AMD A10-5800K APU with Radeon(tm) HD Graphics (4 CPUs 3.8GHz), 16GB RAM, and NVIDIA GeForce GTX 750 GPU 4GB.

Proposed algorithms, *TVS-MoFV* for feature extraction and *TVS-MoDT* for multi-occupant detection and tracking were tested. Both of them used stored TVS_F_{seq}, which was retrieved from the intermediate repository as JSON object arrays, by a pull-based web-service. In TVS-MoFV, TVS-F vector was obtained by varying binary threshold values and finding the best value, suitable for each of VOT-TIR2016 benchmark datasets and the TVS_F_{seq} as mentioned in Table 2. The parametric settings also involved finding the optimal value for the contour area in order to predict the maximum number of occupants in the benchmarks and TVS_F_{seq} as shown in Fig. 4. These TVS-F feature vectors support while iterating the multi-occupant represented as Blobs predicted as bounding rectangles, implemented through the TVS-MoDT algorithm. The *Euclidean distance* was calculated between the detected and predicted bounding rectangles for multi-occupant tracking frame-by-frame. The processing time for each algorithm and method to process a single frame is referred to in Table 3. The source code for *uMoDT* framework and TVS_F_{seq} is available on GitHub at [52].

Table 3: Processing time for benchmarks and TVS_F_{seq} with TVS-MoDT and TVS-AR algorithms

Algorithm	Dataset	Duration(s)
TVS-MoDT	ETHZ-CLA	3.91×10^{-6}
	Soccer	2.99×10^{-6}
	Crouching	6.35×10^{-6}
	Depthwise Crossing	2.93×10^{-6}
	Crowd	2.93×10^{-6}
	TVS_F_{seq}	4.88×10^{-6}
TVS-AR	$TVS\_F_{seq}(O=1)$	7.1×10^{-2}
	$TVS\_F_{seq}(O=2)$	8.3×10^{-2}
	$TVS\_F_{seq}(O=3)$	9.0×10^{-2}

To recognize multi-occupant's ADLs from TVS_F_{seq}, a supervised CNN model was trained. For this the entire collection of TVS_F_{seq} was sorted into two subset groups i.e. training and test categories, each having sixteen classes. The training set is further split with random TSV-F distribution into two halves i.e. 70% for training samples (TVS_F_{Train}) and remaining to validate each class. We used 28,485 TVS-F samples to train CNN model and 1,920 TVS-F test samples (120 TVS-F for each of 16 classes) to evaluate the prototype *uMoDT* framework application.

Table 4: TVS-AR: Activity recognition for multi-occupants using Convolution Neural Networks

Layer	Layer Type	Activation	Parameters (No. of units, Size, Stride)
1	TVS_F_{seq}	Image Input	32×32×1 images with zerocenter normalization
2	conv1	Convolution	16 3×3×1 convolutions with stride [1 1] and padding [1 1 1 1]
3	batchnorm1	Batch Normalization	Batch normalization with 16 channels
4	relu1	ReLU	ReLU
5	maxpool1	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]
6	conv2	Convolution	32 3×3×16 convolutions with stride [1 1] and padding [1 1 1 1]
7	batchnorm2	Batch Normalization	Batch normalization with 32 channels
8	relu2	ReLU	ReLU
9	maxpool2	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]
10	conv3	Convolution	32 3×3×32 convolutions with stride [1 1] and padding [1 1 1 1]
11	batchnorm3	Batch Normalization	Batch normalization with 32 channels
12	relu3	ReLU	ReLU
13	maxpool3	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]
14	conv4	Convolution	64 3×3×32 convolutions with stride [1 1] and padding [1 1 1 1]
15	batchnorm4	Batch Normalization	Batch normalization with 64 channels
16	relu4	ReLU	ReLU
17	fc	Fully Connected	16 fully connected layers
18	soft-max	soft-max	Bayesian binary classifier
19	classoutput	Classification Output	crossentropyex with FallDown and 15 other classes

The nineteen-layer, CNN architecture is designed based on the findings from the systematic comparison and benchmarking to achieve an affordable classification time and computation cost [53]. The implemented CNN architecture comprises of two units i.e. feature extractor and a non-linear classifier [29]. The former unit encapsulates fifteen layers (Layer2. . . Layer16) whereas the latter unit i.e. non-linear classifier is built on all fully connected layers along with the soft-max classifier. During the model training process, the CNN hyperparameters were set with the help of input functions, by adjusting the learning rate effectively to 0.01, every 10 epochs using Stochastic Gradient Descent with Momentum (SGDM) algorithm with the maximum 20 number of epochs size [45]. For every iteration, a mini-batch of size 16 (64) was applied for which the details are mentioned in Table 4.. The output of the last ReLU (relu4) at *Layer 16*, is given to fully connected layer *Layer 17*, which uses the features and processes it for class prediction based on the TVS_F_{Train}. The classification layer i.e. *Layer 18* uses the soft-max activation function, which squashes the output probability vector between sixteen multi-occupant activities and returns the binary indicator to them.

5 Results and discussion

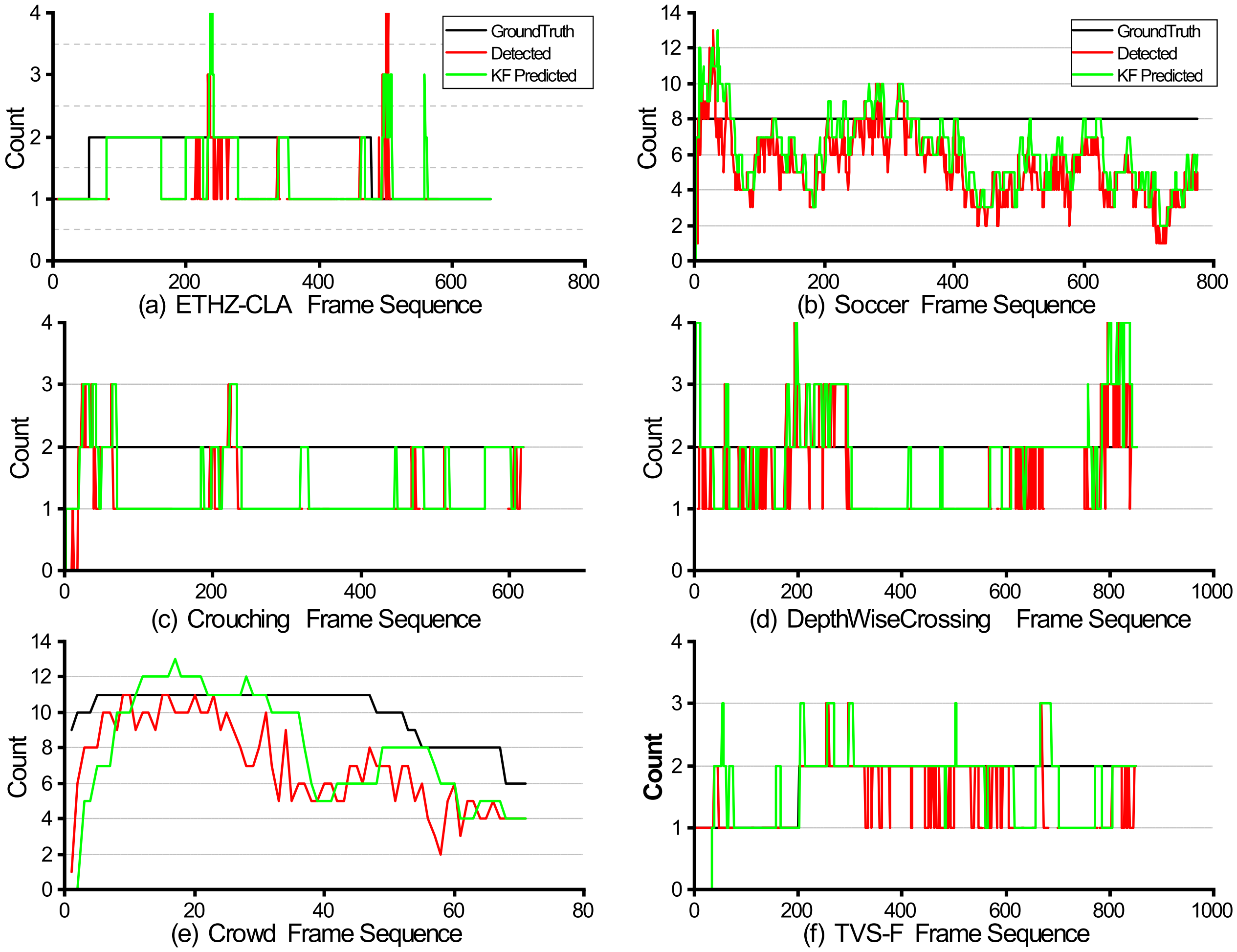
In literature there exist several performance measures to deal with single-target and multi-target tracking, however, none of them proved to be a defacto standard. In our experiments, we adopted some of the effective multi-occupant detection and tracking evaluation strategies to: a) detect and track the multi-occupants and b) classify multi-occupant activities in TVS_F_{seq}. For this, we investigated frame properties in the sequences to identify the influence of different parameters such as variable thresholds and overlap measures on the overall performance. Moreover, conformity of evaluation measures to any other application and sequence have been proven by the *uMoDT* framework on VOT-TIR2016 sequences other than TVS_F_{seq}.

5.1 Multi-occupant detection and tracking evaluation

Objectively quantitative assessment of multi-occupant detection and tracking is not a straight forward task. Most of the evaluation techniques require a ground-truth $\mathrm{G_i}$, which serves as a reference to measure the performance quantitatively. We adopted such evaluation methods, which rely on frame based spatial overlap between $\mathrm{G_i}$ and bounding rectangles $\mathrm{BR_n}$ [54].

5.1.1 Evaluation metrics

The object detection in benchmark sequences and multi-occupant detection in TVS_F_{seq} uses standard *Pascal*, *Intersection over Union* (IoU) criterion, a natural bounding box evaluation measure for comparing



(a) ETHZ-CLA Frame Sequence

(b) Soccer Frame Sequence

(c) Crouching Frame Sequence

(d) DepthWiseCrossing Frame Sequence

(e) Crowd Frame Sequence

(f) TVS-F Frame Sequence

Fig. 5: Quantitative evaluations shown in (a) ETHZ-CLA (b) Soccer (c) Crouching (d) Depthwise Crossing (e) Crowd (f) TVS-F

spatial overlap and localization accuracy [48]:

$$IoU(BR_{\mathrm{n}}, G_{\mathrm{i}}) = \frac{BR_{\mathrm{n}} \cap G_{\mathrm{i}}}{BR_{\mathrm{n}} \cup G_{\mathrm{i}}} \tag{11}$$

5.1.2 Performance evaluation and analysis

We take the advantage of the *Counting* algorithm to estimate number of occupants against $\mathrm{G_i}$ frame-wise in each of the sequence [55]. In our experiments, we considered count detection as true positive (TP) for *IoU* greater than 0.5 otherwise as false positive (FP). For *IoU*<0.5, however, we also considered rotated $\mathrm{BR_n}$ locations for each object obtained from KF in the frame to see if updated object state has any spatial overlap relation with ground-truth. Fig. 5(a-f) present results for $\mathrm{G_i}$, detected, and KF predicted $\mathrm{BR_n}$ frame-wise in each sequence. The best counting success rate is achieved by using improved frame pre-processing algorithms *TVS-MoFV* (1) and *TVS-MoDT* (2) for the *Soccer* sequence with around 94.76% whereas TVS_F_{seq} achieved a counting accuracy of 88.46%. Results by the counting algorithm using KF predicted occupants exhibit an excellent performance for each sequence where occupants are well separated in the frames as compared to the sequences in which they are occluded by each other. To evaluate multi-occupant detection and tracking performance, it is not suitable to use only one single metrics, therefore, we extend the frame-wise *IoU* overlap measure for performance evaluation by estimating *Multiple Object Tracking Accuracy* (MOTA), an accepted evaluation measure [56]. MOTA measure also takes into account the impact of erroneous responses such as: false negatives (FN_t), false positives (FP_t), number of identity switches $\mathrm{IDS_t}$, and $\mathrm{G_t}$ at time *t*. By combining these

Table 5: Evaluation comparison of the uMoDT framework for benchmark sequences and TVS_F_{seq}

	Name	**FP↓**	**FN↓**	**MOTA↑**	**IDS↓**	**Precision↑**	**Recall↑**	**MSE↓**
Dataset	ETHZ-CLA	441	414	5.58	210	0.61	**0.44**	1.04
	Soccer	311	1540	**74.42**	246	**0.94**	0.39	5.19
	Crouching	163	428	57.17	243	0.80	0.29	1.08
	Depthwise	456	408	53.03	180	0.72	0.38	0.96
	Crowd	110	211	57.40	110	0.81	0.41	12.27
	TVS_F_{seq}	52	469	**64.26**	72	**0.87**	0.42	**0.84**

Table 6: Evaluation comparison for the uMoDT framework against other techniques

	Name	**FP↓**	**FN↓**	**MOTA↑**	**IDS↓**
Method	Bochinski *et al.* [57]	5702	70278	57.1	2167
	Wan *et al.* [58]	10604	56182	62.6	1389
	Bewley *et al.* [59]	7318	32615	33.4	1001
	Murray *et al.* [60]	3130	76202	27.4	786
	Chen *et al.* [61]	9253	85431	47.6	792
	Gade *et al.* [55]	9.8%	18.8%	**70.36**	219
	uMoDT (TVS_F_{seq})	52	469	**64.26**	72

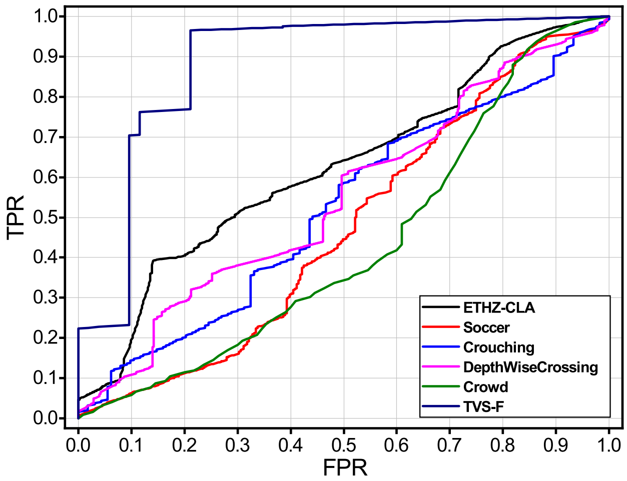


Fig. 6: ROC curves for benchmark sequences and TVS_F_{seq}

sources of error, MOTA is defined as:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t G_t} \qquad (12)$$

We report quantitative evaluations and comparative analysis through the experiments over a set of test sequences for frame-based detection and tracking in Tables 5 and 6 respectively. It is evident that the *uMoDT* framework demonstrated better performance in terms of MOTA for benchmark sequences and TVS_F_{seq}. It outperformed other techniques on all sequences especially for *Soccer* sequence and TVS_F_{seq} with MOTA scores of *74.42%* and *64.26%* respectively. Additionally, the Mean Squared Error (MSE) between the localization of predicted $\mathrm{BR_n}$ and $\mathrm{G_i}$ was also computed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(BR_{\mathrm{n}} - G_{\mathrm{i}})^2 \qquad (13)$$

The error rates showed lowest MSE value of 0.84, which was achieved for TVS_F_{seq} and a highest MSE value of 12.27 for *Crowd* sequence. The tabulated results, however, showed a higher number of $\mathrm{IDS_t}$, an increased MSE, and a decreased MOTA, which appeared to be from occlusions and deforming blobs.

The performance of *uMoDT* is also compared by constructing ROC curves for accumulated true detection rates and false positive rates using $\mathrm{G_i}$ and predicted $\mathrm{BR_n}$ with IoU>0.5 as shown in Fig. 6. The ROC curve produced by TVS_F_{seq} has shown a larger area under the curve than other sequences. This suggests and validates the robustness of the proposed algorithm for occupant detection. TVS_F_{seq} has lessor FPR, which is due to minimal occlusion as compared to other sequences especially in *Crowd* sequence, which has maximum occlusion. Fig. 7 shows the resulting precision-recall curves based on overlap metric. Such a quantitative analysis proves as how successfully the $\mathrm{BR_n}$ are predicted for $\mathrm{G_i}$ in the benchmark sequences and TVS_F_{seq}. The *uMoDT* framework achieved a highest

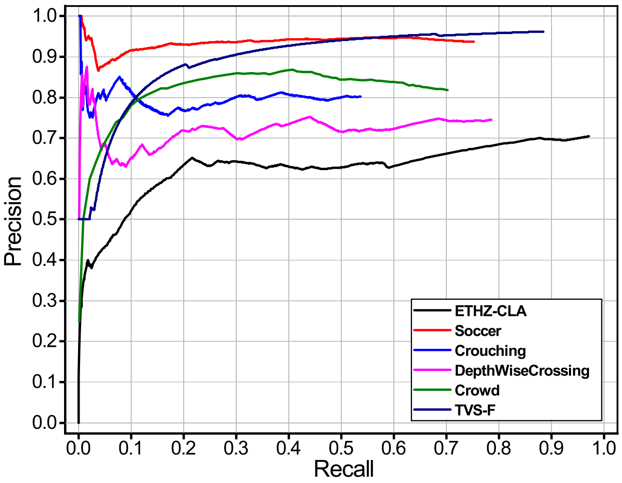


Fig. 7: Precision-recall curves for benchmark sequences and TVS_F_{seq}

area under the curves with an average 97.16% precision rate for TVS_F_{seq} and the lowest one with around 72.04% for *ETHZ-CLA* sequence.

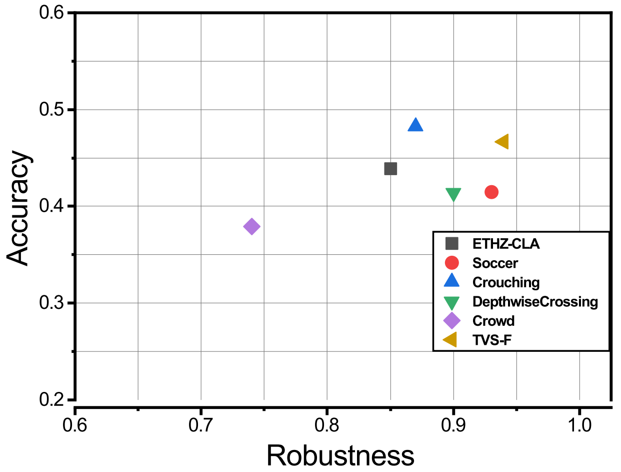


Fig. 8: Accuracy-robustness plot for the *uMoDT* with benchmarks and TVS_F_{seq}

5.1.3 uMoDT robustness

To assess the ability of the *uMoDT* framework as how it deals with the tracking failure, we further quantify it for robustness measure correlated with accuracy. Robustness refers to the *uMoDT* failures whenever the overlap IoU measure becomes equal to zero. To measure the average overlap areas and complete failures, these measures are intuitively computed for benchmark sequences with IoU threshold value equal to zero. We also assumed each occupant in a frame as a separate entity, represented by an independent motion trajectory to evaluate tracking performance [62]. The resulting robustness, however, in some cases does not have an upper bound so it was interpreted as a reliability, defined by $e^{-S(F_0/N)}$ for visualizing purpose [63, 64]. Here N denotes number of frames for an individual sequence, S represents the number of frames since the last failure, and F_0 is a failure rate, which is set as IoU equal to zero. We executed the *uMoDT* framework separately for each sequence to record their average scores, failure rate and unsupervised re-initialization for multi-occupants.

Fig. 8 demonstrates the effectiveness of the *uMoDT* framework, which proved to be most robust on *TVS-F* sequence (positioned most right) but it was surpassed by *Crouching* sequence, which appeared to be more accurate (positioned higher). The observed high robustness for TVS_F_{seq} is because of no occlusion, static distinguishable background and quality of multi-occupant estimates using KF. On the other hand, high average accuracy for *Crouching* sequence is observed, which is due to frequent re-initialization as occupant's appearance is challenging which matches with background. The *uMoDT* framework performed differently between the benchmark sequences depending on their frame properties, however, it achieved an overall best performance except for the *Crowd* sequence (positioned lowered). At a closer look, we see that in terms of accuracy it is challenging as occupants are not well distinguishable from background and also frequent *uMoDT* failures occur due to occlusions. It still, however, has achieved satisfactory robustness.

5.2 Multi-occupant activity recognition

In the following subsection, to show the generality of the TVS-AR method, we describe and evaluate the proposed CNN-based model using the TVS_F_{seq} for AR. We present the classification results to prove the performance and suitability of the presented approach using low-resolution TVS_F_{seq} in terms of accuracy [65]. We used frame-based approach for recognizing 16 different activities showing the efficacy of a model by demonstrating it for a high HAR accuracy score of approximately 90.99%.

5.2.1 Activity recognition evaluation metrics

The performance metric that is most widely used to evaluate a classifier in the context of multiclass classifi-

Table 7: Average accuracy confusion matrix for multi-occupant HAR

Ground Truth Activities (columns); Predicted Activities (rows)

	Act_1	Act_2	Act_3	Act_4	Act_5	Act_6	Act_7	Act_8	Act_9	Act_{10}	Act_{11}	Act_{12}	Act_{13}	Act_{14}	Act_{15}	Act_{16}
Act_1	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Act_2	0	120	0	0	6	0	0	0	0	0	0	0	0	0	0	0
Act_3	0	0	117	0	0	0	0	80	0	0	0	0	0	0	0	0
Act_4	0	0	0	96	0	0	0	0	0	0	0	0	0	0	0	0
Act_5	0	0	0	24	114	0	0	0	0	0	0	0	0	0	0	0
Act_6	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0
Act_7	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0
Act_8	0	0	3	0	0	0	0	40	0	0	0	0	0	0	0	0
Act_9	0	0	0	0	0	0	0	0	120	60	0	0	0	0	0	0
Act_{10}	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0
Act_{11}	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0
Act_{12}	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0
Act_{13}	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0
Act_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0
Act_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0
Act_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120

cation is overall accuracy [41]. The recognition accuracy is linear to the number of training frames. The training frames were used to fit in the parameters such as weights, validation set to fine tune the parameters and CNN architecture. The performance of the customized CNN was evaluated on validation split as a test data to validate the generalization and prediction power of the classifier. Additionally, the other most common performance evaluation metrics such as precision, recall, F-measure also provided an essential information required to assess the classification model [43].

5.2.2 Performance evaluation of activities

For each experiment, we followed the data splits and cross-validation evaluation technique for TVS_F_{seq}. We divided TVS_F_{seq} into three splits: training split TVS_F_{Train} to train CNN model, validation split to tune the hyper-parameters such as learning rate, epoch size on unseen data, and finally test split to evaluate the classification performance. An average accuracy of 97.34% was achieved with a learning rate of 0.01 for 28,485 TVS_F_{seq}. A drop in accuracy, however, was observed with a decrease in the learning rate. The test split contained 1,920 TVS-F for validating 16 activities as mentioned in the confusion matrix illustrated through Table 7. It is observed that the TVS-AR method accurately classified most of single-occupant and multi-occupant activities. Nevertheless, some confusion has been observed for multi-occupant activities such as *StandingWalking* (Act_{10}) and *StandingStretching* (Act_9) have been confused due to similar motion patterns for *Walking* and *Stretching*. This is due to the activity *Stretching*, which involves extension of arms and returning to their original position, again sharing motion patterns to the activity *Standing* in a TVS_F_{seq}. Similarly, static multi-occupant activities *SittingSitting* (Act_3) and *StandingStanding* (Act_8) share similar occupant appearances in the TVS_F_{seq}. For these, the activities *Standing* and *Sitting* were confused due to similar heat maps in the frames. Furthermore, Fig. 9 shows the evaluation metrics in terms of Precision, Recall and F-Measure. By visualizing these, it can be concluded that multi-occupant activity i.e. (Act_8) with both occupants *Standing* and (Act_{10}) with one occupant *Standing* and other one *Walking* has shown the lowest performance for the test split of TVS_F_{seq}.

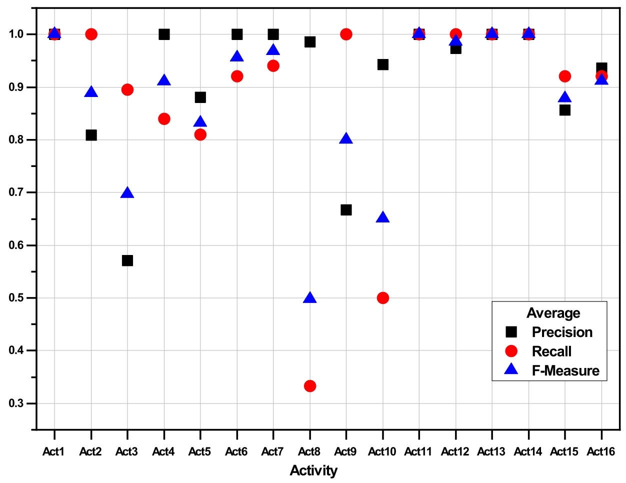


Fig. 9: Classification accuracy using CNN for test TVS-F

6 Conclusions

In this work, we proposed and demonstrated an *unobtrusive Multi-occupant Detection and Tracking* (*uMoDT*) framework for HAR based on low resolution TVS. In this study, by using a binarization technique with Gaussian filter for smoothing, a morphological improvement with inversion and dilation process, an individual occupant in the form of the blob was detected over a sequence of frames. This blob was further tracked by using a KF with location improvement and evaluated with Intersection over Union (IoU). The above methods achieved detection and tracking accuracy of 88.46% for Thermal Vision Sensor frame sequence (TVS_F_{seq}). Additionally, a CNN-based multi-occupant HAR method was evaluated, achieving a validation accuracy of 97.34% and an accuracy of 90.99% for classification tasks. This experimentation demonstrates improvements in occupant detection and, activity association using TVS. The experimental evaluation using state-of-the-art benchmark datasets also revealed the robustness and effectiveness of the proposed framework. Further improvements may be achieved by introducing multiple TVS(s) for HAR. These settings may include movable TVS to recognize ADLs for more complex scenarios at different indoor locations.

Acknowledgements This research was supported by an Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No. 2017-0-00655). This work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-0-01629) supervised by the IITP (Institute for Information & communications Technology Promotion) and NRF-2016K1A3A7A03951968. This work was supported by the REMIND project, which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734355.

References

1. Benmansour A, Bouchachia A, Feham M (2016) Multioccupant activity recognition in pervasive smart home environments. ACM Computing Surveys (CSUR) 48(3):34
2. Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z (2012) Sensor-based activity recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42(6):790–808
3. Singla G, Cook DJ, Schmitter-Edgecombe M (2010) Recognizing independent and joint activities among multiple residents in smart environments. Journal of ambient intelligence and humanized computing 1(1):57–63
4. Gade R, Moeslund TB (2018) Constrained multi-target tracking for team sports activities. IPSJ Transactions on Computer Vision and Applications 10(1):2
5. Synnott J, Rafferty J, Nugent CD (2016) Detection of workplace sedentary behavior using thermal sensors. In: Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the, IEEE, pp 5413–5416
6. Fiaz M, Mahmood A, Jung SK (2018) Tracking noisy targets: A review of recent object tracking approaches. arXiv preprint arXiv:180203098
7. Tran SN, Zhang Q, Karunanithi M (2018) On multi-resident activity recognition in ambient smart-homes. arXiv preprint arXiv:180606611
8. Gade R, Moeslund TB, Nielsen SZ, Skov-Petersen H, Andersen HJ, Basselbjerg K, Dam HT, Jensen OB, Jørgensen A, Lahrmann H, et al (2016) Thermal imaging systems for real-time applications in smart cities. International Journal of Computer Applications in Technology 53(4):291–308
9. Li X, Hu W, Shen C, Zhang Z, Dick A, Hengel AVD (2013) A survey of appearance models in visual object tracking. ACM transactions on Intelligent Systems and Technology (TIST) 4(4):58
10. Shen J, Liang Z, Liu J, Sun H, Shao L, Tao D (2018) Multiobject tracking by submodular optimization. IEEE Transactions on Cybernetics
11. Wang J, Chen Y, Hu L, Peng X, Philip SY (2018) Stratified transfer learning for cross-domain activity recognition. In: 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, pp 1–10
12. Wang L, Gu T, Tao X, Chen H, Lu J (2011) Recognizing multi-user activities using wearable sensors in a smart home. Pervasive and Mobile Computing 7(3):287–298
13. Rafsanjani HN, Ahn CR, Alahmad M (2015) A review of approaches for sensing, understanding, and improving occupancy-related energy-use behaviors in commercial buildings. Energies 8(10):10,996–11,029
14. Hevesi P, Wille S, Pirkl G, Wehn N, Lukowicz P (2014) Monitoring household activities and user location with a cheap, unobtrusive thermal sensor array. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing, ACM, pp 141–145
15. Sengar SS, Mukhopadhyay S (2017) Moving object detection based on frame difference and w4. Signal, Image and Video Processing 11(7):1357–1364
16. Mandellos NA, Keramitsoglou I, Kiranoudis CT (2011) A background subtraction algorithm for detecting and tracking vehicles. Expert Systems with Applications 38(3):1619–1631
17. Xing J, Ai H, Lao S (2009) Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp 1200–1207
18. Parekh HS, Thakore DG, Jaliya UK (2014) A survey on object detection and tracking methods. International Journal of Innovative Research in Computer and Communication Engineering 2(2):2970–2979
19. Luo W, Xing J, Zhang X, Zhao X, Kim TK (2014) Multiple object tracking: A literature review. arXiv preprint arXiv:14097618
20. Cai Z, Gu Z, Yu ZL, Liu H, Zhang K (2016) A real-time visual object tracking system based on kalman filter and mb-lbp feature matching. Multimedia Tools and Applications 75(4):2393–2409

21. Yilmaz A, Javed O, Shah M (2006) Object tracking: A survey. Acm computing surveys (CSUR) 38(4):13
22. Luo X, Guan Q, Tan H, Gao L, Wang Z, Luo X (2017) Simultaneous indoor tracking and activity recognition using pyroelectric infrared sensors. Sensors 17(8):1738
23. Hu WC, Chen CH, Chen TY, Huang DY, Wu ZC (2015) Moving object detection and tracking from video captured by moving camera. Journal of Visual Communication and Image Representation 30:164–180
24. Hou L, Wan W, Hwang JN, Muhammad R, Yang M, Han K (2017) Human tracking over camera networks: a review. EURASIP Journal on Advances in Signal Processing 2017(1):43
25. Zhang B, Li Z, Perina A, Del Bue A, Murino V, Liu J (2016) Adaptive local movement modeling for robust object tracking. IEEE Transactions on Circuits and Systems for Video Technology 27(7):1515–1526
26. Choi W, Savarese S (2012) A unified framework for multi-target tracking and collective activity recognition. In: European Conference on Computer Vision, Springer, pp 215–230
27. Shen J, Yu D, Deng L, Dong X (2017) Fast online tracking with detection refinement. IEEE Transactions on Intelligent Transportation Systems
28. Zebin T, Scully PJ, Ozanyan KB (2016) Human activity recognition with inertial sensors using a deep learning approach. In: 2016 IEEE SENSORS, IEEE, pp 1–3
29. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
30. Dhillon JK, Kushwaha AKS, et al (2017) A recent survey for human activity recoginition based on deep learning approach. In: Image Information Processing (ICIIP), 2017 Fourth International Conference on, IEEE, pp 1–6
31. Dobhal T, Shitole V, Thomas G, Navada G (2015) Human activity recognition using binary motion image and deep learning. Procedia computer science 58:178–185
32. Ji S, Xu W, Yang M, Yu K (2013) 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence 35(1):221–231
33. Ray KS, Chakraborty S (2017) An efficient approach for object detection and tracking of objects in a video with variable background. arXiv preprint arXiv:170602672
34. Leira FS, Johansen TA, Fossen TI (2015) Automatic detection, classification and tracking of objects in the ocean surface from uavs using a thermal camera. In: Aerospace Conference, 2015 IEEE, IEEE, pp 1–10
35. Tiwari M, Singhai R (2017) A review of detection and tracking of object from image and video sequences. International Journal of Computational Intelligence Research 13(5):745–765
36. Wang Y, Luo X, Fu S, Hu S (2018) Context multitask visual object tracking via guided filter. Signal Processing: Image Communication
37. Dehghan A, Shah M (2018) Binary quadratic programing for online tracking of hundreds of people in extremely crowded scenes. IEEE transactions on pattern analysis and machine intelligence 40(3):568–581
38. Sahbani B, Adiprawita W (2016) Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system. In: 2016 6th International Conference on System Engineering and Technology (ICSET), IEEE, pp 109–115
39. Heimanntvs. http://www.heimannsensor.com/productsimaging.php, accessed: 2020-02-25
40. Medina-Quero Jea, Nugent C (2018) Computer vision-based gait velocity from non-obtrusive thermal vision sensors. To be Sumitted 0(0):1–6
41. Zeng M, Nguyen LT, Yu B, Mengshoel OJ, Zhu J, Wu P, Zhang J (2014) Convolutional neural networks for human activity recognition using mobile sensors. In: Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on, IEEE, pp 197–205
42. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:150203167
43. Ordóñez FJ, Roggen D (2016) Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115
44. Albelwi S, Mahmood A (2017) A framework for designing the architectures of deep convolutional neural networks. Entropy 19(6):242
45. Gao Z (2018) Object-based image classification and retrieval with deep feature representations. Doctor of Philosophy Thesis, School of Computing and Information Technology, University of Wollongong
46. Teow MY (2017) Understanding convolutional neural networks using a minimal model for handwritten digit recognition. In: Automatic Control and Intelligent Systems (I2CACIS), 2017 IEEE 2nd International Conference on, IEEE, pp 167–172

47. Tzutalin Labelimg: Image annotation tool. https://github.com/tzutalin/labelImg, accessed: 2020-02-25
48. Kristan M, Matas J, Leonardis A, Vojir T, Pflugfelder R, Fernandez G, Nebehay G, Porikli F, Čehovin L (2016) A novel performance evaluation methodology for single-target trackers. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(11):2137–2155, DOI 10.1109/TPAMI.2016.2516982
49. Vot2016 benchmark. http://www.votchallenge.net/vot2016/, accessed: 2020-02-25
50. Bradski G (2000) The OpenCV Library. Dr Dobb's Journal of Software Tools
51. Portmann J, Lynen S, Chli M, Siegwart R (2014) People detection and tracking from aerial thermal views. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, pp 1794–1800
52. *uMoDT* framework source code. https://github.com/masifrazzaq/TVS-DTC/, accessed: 2020-02-25
53. Mishkin D, Sergievskiy N, Matas J (2017) Systematic evaluation of convolution neural network advances on the imagenet. Computer Vision and Image Understanding 161:11–19
54. Manohar V, Soundararajan P, Raju H, Goldgof D, Kasturi R, Garofolo J (2006) Performance evaluation of object detection and tracking in video. In: Asian Conference on Computer Vision, Springer, pp 151–161
55. Gade R, Moeslund T (2014) Thermal tracking of sports players. Sensors 14(8):13,679–13,691
56. Bernardin K, Stiefelhagen R (2008) Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing 2008:1–10
57. Bochinski E, Eiselein V, Sikora T (2017) High-speed tracking-by-detection without using image information. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, pp 1–6
58. Wan X, Wang J, Zhou S (2018) An online and flexible multi-object tracking framework using long short-term memory. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 1230–1238
59. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B (2016) Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP), IEEE, pp 3464–3468
60. Murray S (2017) Real-time multiple object tracking-a study on the importance of speed. arXiv preprint arXiv:170903572
61. Chen L, Ai H, Zhuang Z, Shang C (2018) Real-time multiple people tracking with deeply learned candidate selection and person re-identification. arXiv preprint arXiv:180904427v1
62. Wu Y, Lim J, Yang MH (2015) Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9):1834–1848
63. Čehovin L, Kristan M, Leonardis A (2014) Is my new tracker really better than yours? In: IEEE Winter Conference on Applications of Computer Vision, IEEE, pp 540–547
64. Čehovin L, Leonardis A, Kristan M (2016) Visual object tracking performance measures revisited. IEEE Transactions on Image Processing 25:1261–1274
65. Wang Q, Gong D, Qi M, Shen Y, Lei Y (2018) Temporal sparse feature auto-combination deep network for video action recognition. Concurrency and Computation: Practice and Experience p e4487