

Depth, Thermal and RGB-Segmented Silhouette Imaging in Human Pose Estimation for Activity Monitoring

Yihao Zhang	Haoran Ni	Beichen Sun
Hwa Chong Institution	Hwa Chong Institution	Hwa Chong Institution
Singapore, Singapore	Singapore, Singapore	Singapore, Singapore
Institute for Infocomm Research (I ² R),	Institute for Infocomm Research (I ² R),	Institute for Infocomm Research (I ² R),
A*STAR	A*STAR	A*STAR
Singapore, Singapore	Singapore, Singapore	Singapore, Singapore
202537x@student.hci.edu.sg	202394u@student.hci.edu.sg	202440n@student.hci.edu.sg

Zheng Yang Chin	Kai Keng Ang
Institute for Infocomm Research (I ² R),	Institute for Infocomm Research (I ² R),
A*STAR	A*STAR
Singapore, Singapore	Singapore, Singapore
zychin@i2r.a-star.edu.sg	Nanyang Technological University
	Singapore, Singapore
	kkang@i2r.a-star.edu.sg

Abstract

Human Pose Estimation (HPE) is a computer vision task that involves deep learning algorithms to estimate keypoints of different human joints in images. The keypoints form a skeleton model of the human body with applications in activity monitoring for healthcare, domotics and surveillance systems. However, HPE that uses RGB images deteriorates under inadequate lighting and raises privacy concerns because identifiable facial features are recorded. Thus, training a HPE deep learning model that predicts keypoints on lighting-invariant and privacy-preserving human silhouette images is proposed. Silhouette Imaging Dataset (SID) was collated from numerous public datasets, consisting of depth images, thermal images and simulated human silhouette images generated from RGB images using image segmentation, which were used to fine-tune the YOLO11s-pose HPE model. To classify predicted keypoints from the HPE model into different actions for activity monitoring, this study proposed a Residual Temporal Convolution Network with Stacked Bidirectional Gated Recurrent Unit (ResTCN-SBiGRU) Human Activity Recognition (HAR) deep learning model. The HAR model was trained on a public dataset to recognise 11 actions. The study shows the following results: the fine-tuned YOLO11s-pose yielded an mAP50-95(P) of 0.851; ResTCN-SBiGRU achieved an F1-score of 0.933. Finally, the study developed a real-time activity monitoring application using HPE, HAR and an additional proposed motion detection algorithm. The application gets live video input from a KinectV2 sensor and a laptop webcam, and displays pose, action and motion visualisations. This allows for testing of the different models and demonstrates the feasibility of activity monitoring using silhouette-based HPE.

CCS Concepts

• **Computing Methodologies**; • **Machine Learning**; • **Human-centered computing**; • **Human computer interaction (HCI)**;

Keywords

Human Pose Estimation, Human Activity Recognition, Computer Vision

ACM Reference Format:

Yihao Zhang, Haoran Ni, Beichen Sun, Zheng Yang Chin, and Kai Keng Ang. 2025. Depth, Thermal and RGB-Segmented Silhouette Imaging in Human Pose Estimation for Activity Monitoring. In *2025 7th International Conference on Image, Video and Signal Processing (IVSP 2025)*, March 04–06, 2025, Ikuta, Japan. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3749859.3749862>

1 Introduction

Human Pose Estimation (HPE) is a fundamental area of research in computer vision that determines spatial locations of human joints – termed keypoints – from images or videos. The set of keypoints has varying definitions, a popular one being the COCO Keypoint format [1]. These predefined keypoints form a skeleton model of the human body that can find uses in activity monitoring for healthcare [2], [3], domotics [4] and surveillance [5] making HPE an area of growing interest in both research and industry [6], [7]. Recent approaches in HPE involve deep learning and can be classified into two main methods: top-down and bottom-up [8]. Top-down methods used in models like YOLO11 [9] and HRNet [10] first detect human instances with object detection, then estimate keypoints within bounding boxes. Bottom up methods in models like OpenPose [11] estimate all keypoints first before associating them into poses. Yet, HPE remains difficult due to occlusions, body shape and pose variation, and diverse environments [8].

Conventionally, HPE uses RGB imaging which raises privacy concerns especially when monitoring human activity. This can be addressed using image segmentation [12]. Segmentation models



This work is licensed under a Creative Commons Attribution 4.0 International License. *IVSP 2025, Ikuta, Japan*
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1218-0/2025/03
<https://doi.org/10.1145/3749859.3749862>



Figure 1: SID images. Left to right: Depth (PKUMMD), Thermal (OTP2), RGB-Segmented (NTU RGB+D 60)

like YOLO11x-seg [13] can generate a mask of all pixels containing humans to form RGB-Segmented images, effectively censoring human features but preserving pose information. However, RGB images are still required and are affected by poor lighting. To address this, depth sensors and thermal cameras that operate without lighting can be used to capture depth maps and thermal data respectively. Depth map pixels represent distances from the camera while thermal data pixels represent temperatures. These pixels can be color-mapped to create greyscale depth images and thermal images. Depth, thermal and RGB-Segmented images can be termed as silhouette imaging in general, where the human silhouette is visible but facial features are not identifiable. Silhouette imaging serves as a privacy preserving and lighting invariant alternative to RGB for HPE. To train a silhouette-based HPE model, depth, thermal and RGB-segmented datasets were combined into a Silhouette Imaging Dataset (SID). However, public depth datasets have different keypoint definitions. As most of them are RGB-D datasets with corresponding RGB and depth image pairs captured by sensors like the Kinect [14], this study explores using a pretrained RGB-based HPE model to infer depth pseudo-labels from matching RGB images.

After training, the HPE model infers keypoints from silhouette imaging in real-time for activity monitoring. For activity monitoring two tasks are specified: Human Activity Recognition (HAR) that classifies actions like waving or jumping and Motion Detection (MD) that checks for human movement. This study proposes the Residual Temporal Convolution Network-Stacked Bidirectional Gated Recurrent Unit deep learning model for HAR and an algorithm for MD. A real-time application using a Kinect V2 sensor [14] and laptop webcam with HPE, HAR and MD was then developed. This application could be used in patient monitoring [15], smart homes [16] or systems that alert unhealthy inactivity, especially with sedentary lifestyles in Singapore [17].

2 Materials and Methods

2.1 Human Pose Estimation (HPE)

2.1.1 Model Selection. YOLO11s-pose was chosen as the HPE model to be trained and deployed real-time for its high accuracy and fast inference speeds. The larger YOLO11x-pose is more accurate but much slower, the smaller YOLO11n-pose is marginally faster but much less accurate, making Other Sizes Less Suitable. In the following data processing steps, YOLO11x-pose was chosen for pseudo-labelling and YOLO11x-seg was selected for image segmentation.

2.1.2 Collated Dataset: Silhouette Imaging Dataset. SID contains 70,000 depth, thermal and RGB-segmented images (Figure 1). Depth images were sourced from public RGB-D datasets: G3D [18], G3Di [19], MVOR [20], ONERA.ROOM [21], TwoPerson Interaction [22] and PKUMMD [23]. The datasets were created over different years, featuring diverse poses, camera positions, and situational contexts (Table 1), thus increasing data variation. Thermal images were taken from OTP2 [24] with non-sequential images from various scenarios. RGB-segmented images were created using RGB images from NTU RGB+D 60 [25]. Video datasets G3D, G3Di, ONERA.ROOM, TwoPerson Interaction, and PKUMMD. NTURGB+D 60 have high similarity between adjacent frames, so downsampling was conducted to reduce redundant similar samples that can cause overfitting (Table 2).

2.1.3 Data Processing. For depth imaging, public datasets use RGB-D cameras like Kinect sensors to capture corresponding RGB and depth images, but many have varying keypoint definitions, generate labels using Kinect HPE [26] which can be noisy, or are unlabelled. Instead of inefficient manual labelling, an existing state-of-the-art YOLO11x-pose model, pre-trained on RGB images from the large COCO dataset, can be used to infer pseudo-labels for the depth image from the RGB image, as long as the RGB-D pair share the same timestamp and camera perspective. Referring to Figure 2, after ensuring the RGB-D pair is synced, depth pseudo-labels were created. All depth image labels contained a bounding box and a set of 17 2D keypoint coordinates in COCO keypoint format per person (Figure 3). For thermal imaging, OTP2 already included manually annotated keypoints in the COCO keypoint format and was directly added. For RGB-Segmented imaging, YOLO11x-seg [13] segmented human masks out of RGB images from NTU GB+D 60. The masks were filled with varying shades of grey to imitate greyscale depth images, then center-cropped to dimensions 512x424. Mask labels were similarly inferred with YOLO11x-pose from the corresponding RGB image.

The depth (D), thermal (T) and RGB-Segmented (S) images were each split into training, validation and testing sets with a 70/20/10 train/val/test split. A mix of cross-actor, cross-view, and other methods of creating validation and testing splits were used for model evaluation as seen in Table 3.

2.1.4 Training of Model. YOLO11s-pose was fine-tuned for 300 epochs on SID, with different combinations of D, T and S to investigate how the addition of one modality could affect another. Image augmentations like horizontal flips, translation, resizing, brightness

Table 1: SID Dataset details including number of human subjects, number of subjects in frame, image resolution, camera position, raw data format and context in which the data was taken from. For Image Resolution, the given dimensions are in pixels (px); RGB represents RGB images and D represents Depth Images.

Dataset	G3D	G3Di	MVOR	ONERA ROOM	TwoPerson	PKUMMD	OTP2	NTURGB+D
Total no. of subjects	10	12	Unknown	Unknown	7	66	170	40
No. of subjects in frame	1 subject	2 subjects	0 to 6 subjects	0 to 7 subjects	1 to 2 subjects	1 to 2 subjects	Varying	1 to 3
Image Resolution	RGB: 640x480 D: 640x480	RGB: 640x480 D: 640x480	RGB: 640x480 D: 640x480	RGB: 640x480 D: 640x480	RGB: 640x480 D: 640x480	RGB: 1920x1080 D: 512x424	T: Varying	RGB: 1920x1080 Final: 512x424
Camera Position	Parallel to ground, front view	Parallel to ground, front view	Mounted on ceilings	Moving on the ground	Parallel to ground, side view	Parallel to ground, left, middle, right view	Varying	Varying
Raw Data	Video frames	Video frames	Images	Videos	Video frames	Videos	Images	Videos
Context	Actions related to games	Actions related to games	View of operation room procedures	View from ground robot	Two persons interacting	Daily activities	Indoor and Outdoor Activities	Daily Activities

Table 2: Sampling details for SID. Action-rich slices of sequences were prioritised and fewer standing frames were sampled. A subset of randomly chosen videos was first sampled from the extensive PKUMMD dataset to form approximately 258,000 frames, which is then sampled again with a 1 in 10 frame sampling rate.

Dataset	Total Frames	Sampling	Total Frames
G3D	79388	1 in 10	7943
G3Di	72029	1 in 10	7205
MVOR	2196	All	2196
ONERA.ROOM	4242	All	4242
TwoPerson	6753	1 in 2	3383
PKUMMD	~258000	1 in 10	25838
OTP2	11391	All	11391
RGB-Segmented (NTU RGB+D 60)	-	All	8535

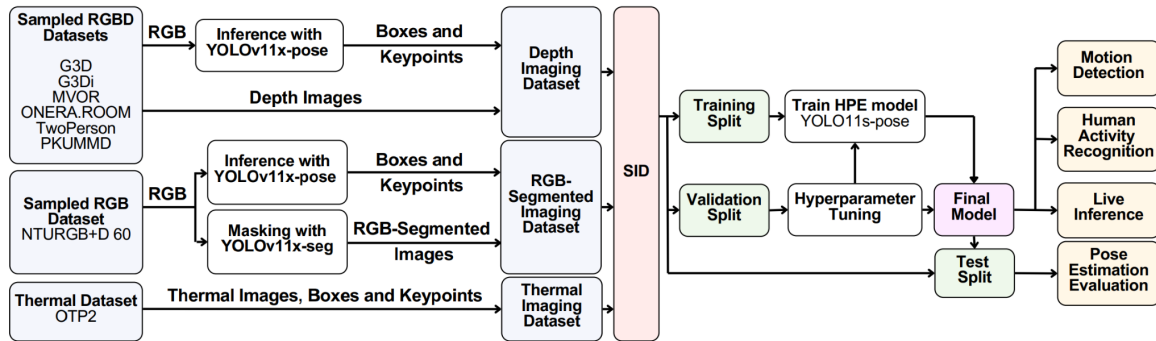


Figure 2: Flowchart of Data Processing.

Table 3: Evaluation methods of datasets in SID. Cross-Actor refers to having separate actors assigned for each split. Cross-View refers having different camera perspectives assigned for each split. Cross-Day is specific to the MVOR dataset and refers to assigning different days to each split. Others refers to an unspecified evaluation method, specific to OTP2 as the original dataset splits with no apparent evaluation method were directly adopted.

Dataset	G3D	G3Di	MVOR	ONERA ROOM	TwoPerson Interaction	PKUMMD	OTP2	NTURGB+D
Evaluation Method	Cross-Actor	Cross-Actor	Cross-Day	Cross-View	Cross-Actor	Cross-View	Others	Cross-Actor

Table 4: Training Hyperparameters;b): Training Image Augmentation

Configuration	Value	Description
Epochs	300	The model is trained on the dataset for 300 iterations
Batch Size	32	The model trains on batches of 32 images
Image Size	320	The model processes images of size 320px by 320px, images will be resized beforehand.
Optimiser	SGD Momentum=0.937 Weight Decay=0.00005	Stochastic Gradient Descent with momentum 0.937 and weight decay 0.0005
Close Mosaic	10	The image augmentation of mosaicing is stopped in the last 10 epochs of training.

Table 5: Training Image Augmentation

Image Augmentation	Value	Description
Horizontal Flip	0.5	50% probability to flip an image horizontally.
Translation	0.1	Translates the image horizontally and vertically by 10% of the image size
Resize	0.5	Chance to halve dimensions of the image
Optimiser	0.4	Changes an image’s brightness up to $\pm 40\%$ of the original value. A random adjustment is chosen between -40% to +40% and applied to image brightness.
Close Mosaic	1.0	Combines 4 images into one.

shifting and mosaicing were used as they assist in model generalisation [27]. More training configuration details can be seen in Table 4 and Table 5.

2.2 Human Activity Recognition (HAR)

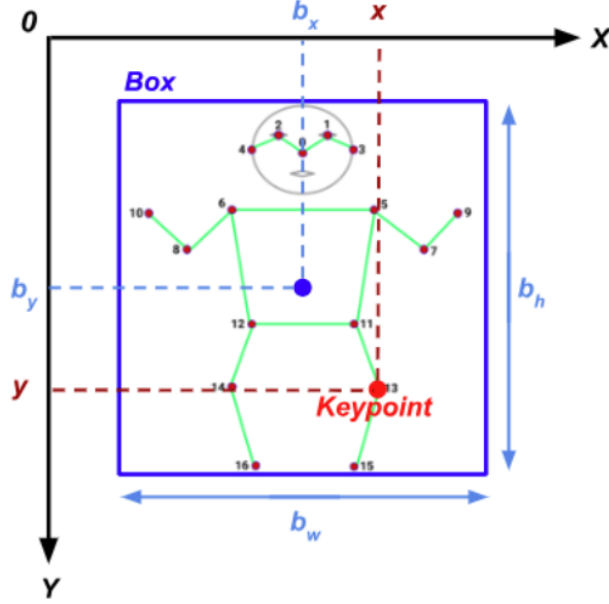
2.2.1 Problem Definition. Given a sliding window V of size n that captures a sequence of keypoints from a single person, where $V = \{K_{t-n}, K_{t-(n-1)}, \dots, K_t, \dots, K_{t+(n-1)}, K_{t+n}\}$ and K_i is the set of keypoints at frame i . A model determines if the center frame t belongs to an action among a predefined M action classes. On top of the M classes, a blank class is added, representing any action that does not belong to any predefined actions. In live inference, an input

window of the latest $2n+1$ frames outputs an action class for the n^{th} latest (middle) frame, inducing a ‘delay’ of n frames.

2.2.2 Dataset Preparation. The activity-rich and action-diverse PKUMMD dataset was used, providing labels of type, starting frame and ending frame of all actions in a video. 10 actions common in daily individual activities (Table 7) were selected. Depth videos containing the actions and only one person were chosen. YOLO11s-pose that was fine-tuned on SID inferred 2D keypoints for each frame, which were normalised to within $(-0.5, 0.5)$ using the bounding box from top-down HPE. In Figure 3, given a box b with center (b_x, b_y) , width b_w and height b_h , a keypoint (x, y) is normalised to (x_n, y_n) using the equations 1) and (2). Sliding windows of size

Table 6: Summary of action classes and their number of window samples.

Class	Samples	Class	Samples	Class	Samples
0 Blank class	3991	4 Touch Head	7631	8 Kick	2459
1 Wave	5343	5 Touch Chest	7041	9 Sit Down	1626
2 Clap	4369	6 Pickup	6115	10 Stand Up	1041
3 Throw	3824	7 Jump	5645		

**Figure 3: Box & Keypoint Format**

50 and stride 5 (moves 5 frames per step) of the normalised keypoints were created and labelled. A 70/20/10 train/val/test split was adopted.

$$x_n = \frac{x - b_x}{b_w} \quad (1)$$

$$y_n = \frac{y - b_y}{b_h} \quad (2)$$

2.2.3 Proposed Residual Temporal Convolutional Network - Stacked Bidirectional Gated Recurrent Unit (ResTCN-SBiGRU). Architectures commonly used to process temporal or sequential data are used as baselines, including Temporal Convolutional Networks (TCN) [28], Gated Recurrent Units (GRU) [29] and Long-Short Term Memory (LSTM) [30]. Our proposed model, ResTCN-SBiGRU (Figure 4), extracts features from small windows of frames with ResTCN, which are then aggregated across the whole sequence by SBiGRU. The choice of GRU over LSTM is prompted by empirical evidence [29] and later results in Table 5 which suggest that GRU is comparable to LSTM in performance while having fewer weights. Residual connections [31] are added in the TCN to smoothen the loss landscape [32] and mitigate vanishing gradients for better convergence. The input consists of a window of keypoints of dimensions 101x34,

from 101 frames of 34 keypoints per frame. The first 1D Conv layer is a temporal convolution with 64 filters of kernel size 3 and zero padding 1 on each side. A batch normalization layer (Batchnorm) [33] and ReLU activation function are added. Then, the ResTCN block receives features of dimensions 101x64. Within each ResTCN block, a 1D Conv layer with 64 filters, kernel size 3 and zero padding of 1 on each side is followed by Batchnorm and ReLU activation function. A residual connection joins the input to the output of a ResTCN block. Note that input and output dimensions of a ResTCN block are the same, allowing the block to be simply repeated again in our architecture. The extracted features of dimensions 101x64 are passed to the SBiGRU, consisting of two repetitions of a Bidirectional Gated Recurrent Unit with hidden size 64 and Dropout layer [34] with probability 0.5. After the SBiGRU, only the last hidden state of both directions are flattened (64x2=128) and fed into a Dense layer to get class logits, then a Softmax layer to output 11 class probabilities. Models were trained for 50 epochs with Cross-Entropy Loss [35]. ResTCN-SBiGRU hyperparameter tuning was carried out using the val set while final results are from the test set.

2.3 Motion Detection (MD)

2.3.1 Problem Definition. A person is determined as currently moving or still using keypoints. Keypoints are inferred in real-time and BoT-SORT [36] is used to keep track of multiple people and their keypoints.

2.3.2 Algorithm. Denote two sets K_{t-1} and K_t from adjacent frames $t-1$ and t each containing 17 keypoints from the same person. Let the i -th keypoint in K_{t-1} and K_t be $k_{i,t-1} = (x_{i,t-1}, y_{i,t-1})$ and $k_{i,t} = (x_{i,t}, y_{i,t})$. A keypoint not detected is defaulted to $(0, 0)$. A binary indicator b_i is 1 if the i -th keypoint is detected in both K_{t-1} and K_t or 0 otherwise by checking if the i -th keypoint is $(0, 0)$ in either of K_{t-1} or K_t . The number of keypoints N_t detected in both K_{t-1} and K_t is obtained by summing b_i for all i , but with a lower bound of 1 to avoid zero division error later on. The 2-dimensional euclidean distance, d_i , between $k_{i,t-1}$ and $k_{i,t}$ is calculated for each keypoint, then summed up and divided by N_t to compute the averaged displacement of all detected keypoints, D_t . Letting c be the current frame, a weighted average W_c of the R latest D_t (i.e. $D_{c-(R-1)}$ to D_c) is computed with weight proportional to how current a frame is. If W_c exceeds a threshold T , the person is moving now, else the person is not. $R=30$ and $T=1.8$ are empirically determined.

$$b_i = \begin{cases} 1, & k_{i,t-1} \neq (0, 0) \text{ and } k_{i,t} \neq (0, 0) \\ 0, & k_{i,t-1} = (0, 0) \text{ or } k_{i,t} = (0, 0) \end{cases} \quad (3)$$

$$N_t = \max \left(1, \sum_{i=1}^{17} b_i \right) \quad (4)$$

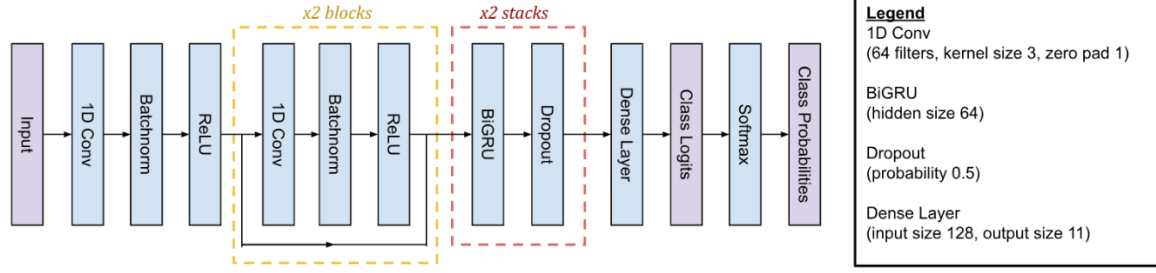


Figure 4: ResTCN-SBiGRU architecture. The yellow block is performed twice. The red stack is performed twice to form SBiGRU. Only the last hidden state of both directions from the last stack in SBiGRU are concatenated and passed into the FC layer.

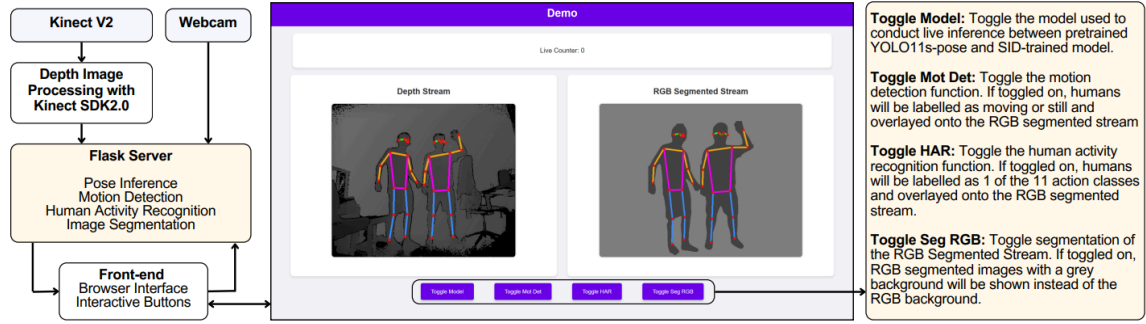


Figure 5: Interface for live application with buttons to toggle specific functions. The application runs at around 20 FPS with a 13th Gen Intel(R) Core(TM) i9-13900HX CPU and a NVIDIA GeForce RTX 4060 Laptop GPU.

$$d_i = \sqrt{(x_{i,t} - x_{i,t-1})^2 + (y_{i,t} - y_{i,t-1})^2} \quad (5)$$

$$D_t = \frac{1}{N_t} \sum_{i=1}^{17} d_i \quad (6)$$

$$W_c = \sum_{r=1}^R \left(1 - \frac{r-1}{R}\right) D_{c-(r-1)} \quad (7)$$

$$= \begin{cases} \text{Moving,} & W_c \geq T \\ \text{Still,} & W_c < T \end{cases} \quad (8)$$

2.4 Real-time Application

2.4.1 Depth & RGB Dual-Segmented Streams. A C# script acquires depth maps from a Kinect V2 sensors, then greyscale color-codes them into depth images. The depth images are streamed in real time via websockets to a laptop-hosted Flask server [37]. Simultaneously, an RGB-Segmented stream is created with YOLO11s-seg from a live RGB stream captured by the laptop webcam. After processing, both sources are streamed to a web browser client on the same device (Figure 5)

2.4.2 Flask Server Image Processing. With BoT-SORT tracking every human, HPE with YOLO11s-pose trained on SID, HAR using ResTCN-SBiGRU, and MD are performed on depth and RGB-Segmented streams, with pose, action and motion visualisations. The RGB-segmented stream has two presentation modes: the human segmentation masks with the actual RGB background or without the background i.e. grey background. The HPE model can be

switched with the default RGB-based YOLO11s-pose for comparison.

3 Results and Discussion

3.1 Human Pose Estimation

Mean Average Precision (mAP) evaluates predictions using thresholds: Intersection Over Union (IOU) for boxes and Object Keypoint Similarity (OKS) for keypoints [38]. Larger IOU or OKS thresholds mean stricter criteria for correct predictions in mAP calculation. mAP50-95 is an average of mAPs with IOU or OKS thresholds from 0.5 to 0.95. A higher mAP suggests better performance by the HPE model in predicting bounding boxes or keypoint positions. For the sake of brevity in Table 7, Table 8, Table 9 and Table 10, mAP50-95 is abbreviated as mAP, (B) and (P) indicate results for box and pose. D, T, S, represent respectively depth, thermal and RGB-Segmented subsets of SID. E.g. DT is depth and thermal images, DTS consists of depth, thermal and RGB-Segmented images.

Looking at mAP(P) in Table 7, 0.94682 from the train split shows low bias. However, the train-validation gap of 0.1141 shows high variance, pointing to overfitting. Minimal data mismatch exists between validation (0.83270) and test (0.85122) splits but testing the model on the real-time application reveals inaccuracies on bent leg poses and foreshortened views. These challenging testcases could be appended for further training. Evaluating the trained model against the RGB baseline on the same SID test split in Table 8, mAP increases significantly for the trained model across the board,

Table 7: Train, val, test splits from SID.

	Train	Val	Test
mAP(B)	0.96940	0.86110	0.89633
mAP(P)	0.94682	0.83270	0.85122

Table 8: Comparison between trained model and RGB baseline on test split. The trained model is trained on SID and the baseline is the default YOLO11s-pose pre trained on COCO Pose dataset.

Test split/Model	Untrained RGB Baseline		SID-trained	
	mAP(B)	mAP(P)	mAP(B)	mAP(P)
D	0.57227	0.27135	0.87331	0.84846
T	0.90028	0.72707	0.95624	0.84503
S	0.63560	0.10116	0.98575	0.88884
DTS (all)	0.63170	0.32845	0.89633	0.85122

Table 9: Effect of adding S to D. The model is trained separately on D and DS subsets from train split, and evaluated on the same D subset from test split

Train Split	D	DS
mAP(B)	0.88429	0.87704
mAP(P)	0.84878	0.85174

Table 10: Effect of adding S to T. The model is trained separately on T and TS subsets from train split, and evaluated on the same T subset from test split

Train Split	T	TS
mAP(B)	0.96285	0.96289
mAP(P)	0.86663	0.86546

showing that the model is capable of learning from one modality without compromising others. Adding S in training marginally affects mAP(P): increasing slightly for D (Table 9) and decreasing slightly for T (Table 10).

3.2 Human Activity Recognition

TCN, GRU, LSTM and the proposed architecture, ResTCN-SBiGRU (Figure 5) were compared in Table 11. A batch size of 8, a learning rate of 0.0005 and the Adam optimiser [39] were used. With class imbalance, accuracy alone is insufficient to evaluate performance [40]. Precision, Recall and F1-Score are also used [41] and macro-averaged [42] for multi-class classification.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{FP + FN} \quad (11)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (12)$$

While ResTCN-SBiGRU leads with an F1-Score of 0.9333 the other models also perform decently, showing the flexibility of keypoint-based HAR across models and the usefulness of the HPE model for HAR. However, ResTCN-SBiGRU does not generalise well to in-the-wild data during testing in the real-time application despite the choice of normalized keypoints as input, hypothesised to be caused by data mismatch between the HAR dataset with limited camera views and in-the-wild data that have any camera position.

4 Conclusion and Future Work

This study explores privacy-preserving, lighting-invariant Human Pose Estimation (HPE) for activity monitoring: Human Activity Recognition (HAR) and Motion Detection (MD). A Silhouette Imaging Dataset (SID) comprising depth, thermal RGB-segmented images was built from multiple public datasets and used to fine-tune the YOLO11s-pose HPE model. YOLO11s-pose achieved an mAP50-95(P) of 0.85122 on the SID test set, but more training samples with leg variation and foreshortened poses could be added. For HAR, the proposed ResTCN-SBiGRU model achieved an F1-Score of 0.9333 on the test set, but did not generalise well on in-the-wild data, which

Table 11: Comparison of HAR models. Accuracy, Precision, Recall and F1-Score are given by equations 9, 10, 11 and 12 respectively.

Model/Metrics	Accuracy	Precision	Recall	F1-Score
TCN	0.8372	0.8140	0.8133	0.8049
GRU	0.8849	0.8941	0.8844	0.8858
LSTM	0.8880	0.8924	0.8860	0.8857
ResTCN-SBiGRU	0.9371	0.9316	0.9365	0.9333

could be investigated. For MD, the proposed algorithm checks human movement, useful in alerting physical inactivity or fainting for timely medical attention. A real-time Flask application was built to test and visualise the aforementioned algorithms and also demonstrate the feasibility of deploying such a system for activity monitoring using silhouette-based HPE.

References

- [1] "COCO Keypoints," CVAT. Accessed: Jan. 13, 2025. [Online]. Available: <https://docs.cvat.ai/docs/manual/advanced/formats/coco-keypoints/>
- [2] J. Stenum, K. M. Cherry-Allen, C. O. Pyles, R. D. Reetzke, M. F. Vignos, and R. T. Roemmich, "Applications of Pose Estimation in Human Health and Performance across the Lifespan," *Sensors*, vol. 21, no. 21, Art. no. 21, Jan. 2021, doi: 10.3390/s21217315.
- [3] A. Tharatipyakul, T. Srikaewsiew, and S. Pongnumkul, "Deep learning-based human body pose estimation in providing feedback for physical movement: A review," *Heliyon*, vol. 10, no. 17, Sep. 2024, doi: 10.1016/j.heliyon.2024.e36589.
- [4] T. He, "Human Pose-Based Activity Recognition Approaches on Smart-Home Devices," in *Distributed, Ambient and Pervasive Interactions. Smart Environments, Ecosystems, and Cities*, N. A. Streitz and S. Konomi, Eds., Cham: Springer International Publishing, 2022, pp. 266–277. doi: 10.1007/978-3-031-05463-1_19.
- [5] M. Cormier, A. Clepe, A. Specker, and J. Beyerer, "Where are we with Human Pose Estimation in Real-World Surveillance?," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 591–601. doi: 10.1109/WACVW54805.2022.00065.
- [6] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D Human Pose Estimation: New Benchmark and State of the Art Analysis," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA: IEEE, Jun. 2014, pp. 3686–3693. doi: 10.1109/CVPR.2014.471.
- [7] B. Xiao, H. Wu, and Y. Wei, "Simple Baselines for Human Pose Estimation and Tracking," Aug. 21, 2018, *arXiv:1804.06208*. doi: 10.48550/arXiv.1804.06208.
- [8] C. Zheng et al., "Deep Learning-Based Human Pose Estimation: A Survey," Jul. 03, 2023, *arXiv:2012.13392*. doi: 10.48550/arXiv.2012.13392.
- [9] G. Jocher, J. Qiu, and A. Chaurasia, *Ultralytics YOLO*. (Jan. 2023). Python. Accessed: Jan. 13, 2025. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [10] J. Wang et al., "Deep High-Resolution Representation Learning for Visual Recognition," Mar. 13, 2020, *arXiv:1908.07919*. doi: 10.48550/arXiv.1908.07919.
- [11] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," May 30, 2019, *arXiv:1812.08008*. doi: 10.48550/arXiv.1812.08008.
- [12] G. Csirka, R. Volpi, and B. Chidlovskii, "Semantic Image Segmentation: Two Decades of Research," Feb. 13, 2023, *arXiv:2302.06378*. doi: 10.48550/arXiv.2302.06378.
- [13] Ultralytics, "Segment." Accessed: Jan. 13, 2025. [Online]. Available: <https://docs.ultralytics.com/tasks/segment>
- [14] "How It Works: Xbox Kinect" Accessed: Jan. 13, 2025. [Online]. Available: [https://www.jameco.com/Jameco/workshop/Howitworks/xboxkinect.html?srsltid=\\$AfmBOopS2hH2-CouXnrvtWL-XE6X7osiL8xir-WgmdgIN0hGAAfs1Ll6](https://www.jameco.com/Jameco/workshop/Howitworks/xboxkinect.html?srsltid=$AfmBOopS2hH2-CouXnrvtWL-XE6X7osiL8xir-WgmdgIN0hGAAfs1Ll6)
- [15] B. Debnath, M. O'Brien, M. Yamaguchi, and A. Behera, "A review of computer vision-based approaches for physical rehabilitation and assessment," *Multimed. Syst.*, vol. 28, no. 1, pp. 209–239, Feb. 2022, doi: 10.1007/s00530-021-00815-4.
- [16] K. Anbazhagan, G. Swamy, R. Janani, and A. Farakte, "Deep Learning based Human Activity Recognition in Smart Home," in *2024 4th International Conference on Data Engineering and Communication Systems (ICDECS)*, Mar. 2024, pp. 1–6. doi: 10.1109/ICDECS59733.2023.10502527.
- [17] "nphs-2021-survey-report_v6-(cleaned).pdf" Accessed: Jan. 13, 2025. [Online]. Available: [https://www.hpb.gov.sg/docs/default-source/default-document-library/nphs-2021-survey-report_v6-\(cleaned\).pdf?sfvrsn=\\$f9d12295_0](https://www.hpb.gov.sg/docs/default-source/default-document-library/nphs-2021-survey-report_v6-(cleaned).pdf?sfvrsn=$f9d12295_0)
- [18] V. Bloom, D. Makris, and V. Argyriou, "G3D: A gaming action dataset and real time action recognition evaluation framework," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2012, pp. 7–12. doi: 10.1109/CVPRW.2012.6239175.
- [19] V. Bloom, V. Argyriou, and D. Makris, "G3Di: A Gaming Interaction Dataset with a Real Time Detection and Evaluation Framework," in *Computer Vision - ECCV 2014 Workshops*, L. Agapito, M. M. Bronstein, and C. Rother, Eds., Cham: Springer International Publishing, 2015, pp. 698–712. doi: 10.1007/978-3-319-16178-5_49.
- [20] V. Srivastava, T. Issenhuth, A. Kadkhodamohammadi, M. de Mathelin, A. Gangi, and N. Padoy, "MVOR: A Multi-view RGB-D Operating Room Dataset for 2D and 3D Human Pose Estimation," Aug. 20, 2021, *arXiv:1808.08180*. doi: 10.48550/arXiv.1808.08180.
- [21] B. L. Saux, "ONERA.ROOM." IEEE, Oct. 24, 2019. Accessed: Jan. 13, 2025. [Online]. Available: <https://iee-dataport.org/open-access/oneraroom>
- [22] "Two Person Interaction Kinect Dataset" Accessed: Jan. 13, 2025. [Online]. Available: <https://www.kaggle.com/datasets/dasmehdixtr/two-person-interaction-kinect-dataset>
- [23] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "PKU-MMD: A Large Scale Benchmark for Continuous Multi-Modal Human Action Understanding," Mar. 28, 2017, *arXiv:1703.07475*. doi: 10.48550/arXiv.1703.07475.
- [24] A. Kuzdeuov, M. Zakaryanov, A. Tleuliyev, and H. A. Varol, "OpenThermalPose2: Extending the Open-Source Annotated Thermal Human Pose Dataset With More Data, Subjects, and Poses," Accessed: Jan. 13, 2025. [Online]. Available: [https://www.authorea.com/users/682600/articles/1231799-openthermalpose2-extending-the-open-source-annotated-thermal-human-pose-dataset-with-more-data-subjects-and-poses?commit=\\$654ef46a963553206714b831c6d7aee26d7dab47](https://www.authorea.com/users/682600/articles/1231799-openthermalpose2-extending-the-open-source-annotated-thermal-human-pose-dataset-with-more-data-subjects-and-poses?commit=$654ef46a963553206714b831c6d7aee26d7dab47)
- [25] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," Apr. 11, 2016, *arXiv:1604.02808*. doi: 10.48550/arXiv.1604.02808.
- [26] J. Shotton et al., "Real-Time Human Pose Recognition in Parts from a Single Depth Image," presented at the CVPR, Jun. 2011. Accessed: Jan. 13, 2025. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/real-time-human-pose-recognition-in-parts-from-a-single-depth-image/>
- [27] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognit.*, vol. 137, p. 109347, May 2023, doi: 10.1016/j.patcog.2023.109347.
- [28] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks for Action Segmentation and Detection," Nov. 16, 2016, *arXiv:1611.05267*. doi: 10.48550/arXiv.1611.05267.
- [29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 11, 2014, *arXiv:1412.3555*. doi: 10.48550/arXiv.1412.3555.
- [30] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 10, 2015, *arXiv:1512.03385*. doi: 10.48550/arXiv.1512.03385.
- [32] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the Loss Landscape of Neural Nets," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2018. Accessed: Jan. 13, 2025. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/hash/a41b3bb3e6b050b6c9067c67f663b915-Abstract.html
- [33] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Mar. 02, 2015, *arXiv:1502.03167*. doi: 10.48550/arXiv.1502.03167.
- [34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [35] Z. Zhang and M. R. Sabuncu, "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels," Nov. 29, 2018, *arXiv:1805.07836*. doi: 10.48550/arXiv.1805.07836.
- [36] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust Associations Multi-Pedestrian Tracking," Jul. 07, 2022, *arXiv:2206.14651*. doi: 10.48550/arXiv.2206.14651.
- [37] "Welcome to Flask — Flask Documentation (3.1.x)." Accessed: Jan. 13, 2025. [Online]. Available: <https://flask.palletsprojects.com/en/stable/>
- [38] T. Dutta, "Object Keypoint Similarity in Keypoint Detection." Accessed: Jan. 13, 2025. [Online]. Available: <https://learnopencv.com/object-keypoint-similarity/>

- [39] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 30, 2017, *arXiv*: arXiv:1412.6980. doi: 10.48550/arXiv.1412.6980.
- [40] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.
- [41] "(PDF) Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation." Accessed: Jan. 13, 2025. [Online]. Available: https://www.researchgate.net/publication/276412348_Evaluation_From_precision_recall_and_F-measure_to_ROC_informedness_markedness_correlation
- [42] K. Safjan, "Understanding Micro and Macro Averages in Multiclass Multilabel Problems," Krystian Safjan's Blog. Accessed: Jan. 13, 2025. [Online]. Available: <https://www.safjan.com/micro-and-macro-averages-in-multiclass-multilabel-problems/>