International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 36, No. 5, October, 2022, pp. 791-799
Journal homepage: http://iieta.org/journals/ria

Recognizing Multiple Human Activities Using Deep Learning Framework

Jitha Janardhanan*, Umamaheswari Subbian

Dept. of Computer Science, Dr.G.R. Damodaran College of Science, Coimbatore- 641014, India

Corresponding Author Email: jithajanardhanan@gmail.com

https://doi.org/10.18280/ria.360518

Received: 31 July 2022
Accepted: 18 October 2022

Keywords:
Human activity, EfficientNET, Pose estimation, GRU, Optical flow, video frames, Deep Learning

ABSTRACT

Multiple human activity detection is a trend in smart surveillance that comes with a number of difficulties, including real-time analysis of large amounts of video data while maintaining low processing complexity. In existing systems of EfficientNET: Scalable and Efficient Object Detection and Deep Skip Connection Gated Recurrent Unit (DS-GRU) performs a compound scaling method that uniformly scales the resolution, depth, and width for all backbone, feature network, and box or class prediction networks at the same time. This study executes only compound scaling method and allows the scaled model to predict the objects with COCO 2017 image Dataset. It performs object recognition only and does not analyze multiple human activities with corresponding class interactions. To overcome this issue, the proposed system presents an Enhanced Bidirectional Gated Recurrent Unit with Long Short Term Memory (BGRU-LSTM) classification algorithm that is adapted to the Human Activity Recognition (HAR) task. The proposed multiple human activities with Pose estimation technique improves better accuracy using EfficientNET feature extraction model along with classification. The experimental results were evaluated with a real-time surveillance video dataset captured from the home apartment.

1. INTRODUCTION

Due to growing need for automated video content understanding, video analysis has recently attracted a lot of attention from the computer vision field. Action identification [1, 2], semantic segmentation [3, 4], action localization [5], and deception detection [6] are just a few of the video analysis tasks that have seen successful performance of deep learning methods.

However, many real-world applications still cannot be satisfied by current convolutional neural network (CNN) based video analysis models. Two factors are mostly to blame for this. First, in order to conserve space, videos on social media platforms like YouTube, real-time security footage, and mobile devices are all kept in a compressed format. However, the majority of devices in use today can only handle uncompressed raw RGB frames. An essential component of the study of human actions and human-behavior is Human Activity Recognition (HAR). Different machine learning algorithms can identify more challenging human actions including drinking, driving, and walking. HAR is essential for keeping elderly individuals healthy as they go about their daily life.

HAR is a classic Pattern Recognition (PR) issue. Support vector machines (SVMs), decision trees, naive Bayes, and hidden Markov models are a few examples of machine learning algorithms that are used in traditional PR problem solving strategies [7]. In HAR issues like disease detection, machine learning methods perform superbly. However, because there are limitations to human competence, a lot of machine learning algorithms rely on manual feature extraction. These limitations make it impossible for machine learning models to learn deep features and engage in unsupervised learning. Traditional PR methods are only partially applicable in HAR due to their partial classification accuracy and model performance.

Deep learning algorithms have advanced quickly as alternatives to conventional PR tactics in recent years. In HAR applications, deep learning algorithms outperform conventional PR. In particular, deep learning models find more functions and more advanced functions while reducing the intended effort [8]. The research study assesses deep learning model's capacity to model real-time video footage and identify a variety of human behaviors. Finding the baseline deep-learning model that categorizes various human activities best is just one of the goals; another is to use an advanced technique to enhance the baseline deep-learning model. Finally, the model that categorizes various human activities the best is discovered.

Recognition of human activity refers to the challenge of identifying people and categorizing particular human actions taken in video frames. Finding objects in visual input is the focus of the paper's discussion of the object detection concept. Dogs, cats, humans, and cars are the most recognizable items, yet any object can be used to train the model to detect them. In this paper, just the persons or peoples who were seen in the film were examined; all other non-human things were ignored.

A neural network-based object detection model typically consists of a complicated structure with multiple components. Because these components may be viewed as modules that can be changed, eliminated, or exchanged, the EfficientNET architecture makes this convenient. Figure 1 shows an illustration of a typical EfficientNET neural network-based object detector.

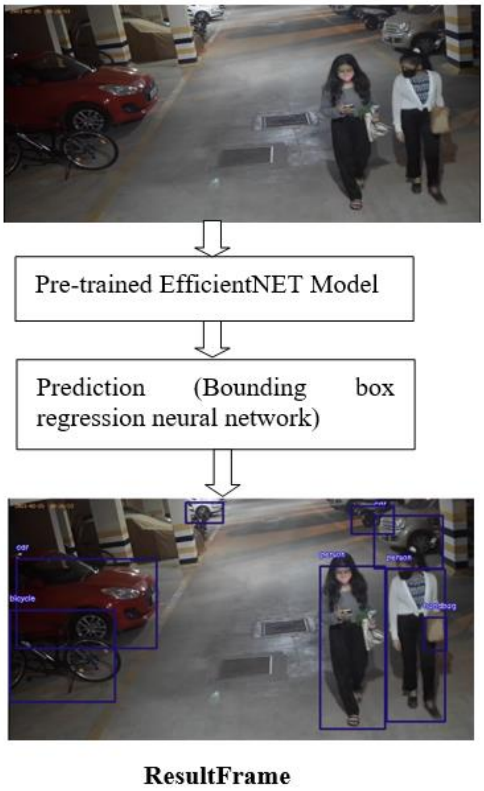

Pre-trained EfficientNET Model
Prediction (Bounding box regression neural network)
car
bicycle
person
handbag
ResultFrame

Figure 1. A typical structure of EfficientNET based object detection model with frame taken from real time surveillance video dataset.

The paper's goal is to recognize different human activities, capture behavior patterns, and categorize them from real-time surveillance video data sets. With information taken from videos that is embedded in the number of frames, one can identify human activity. To conduct the multiple HAR, a GRU (Gated Recurrent Unit) with Bidirectional LSTM technique combining was presented. The suggested approach was able to investigate preventative human behavior patterns and categorize them from complicated situations combined with noises, which is extremely different from the strategy that required deep knowledge and its collection of video frames.

2. RELATED WORK

Action spotting in video, as defined by Alwassel et al. [9], is the new challenge of identifying a particular action in a video while only paying attention to a brief excerpt of the movie. Action Search is designed to mirror how humans identify actions. Additionally, in order to address the dearth of data documenting the activities of human annotators, authors presented the Human Searches dataset, which gathers the search patterns used by human annotators to find actions in the AVA and THUMOS14 datasets. As a solution to the action spotting problem, they suggested temporal action localization.

For solving the problem in HAR videos, Ji et al. [10] proposed end-to-end architecture. In order to provide semantic segmentations in a single, integrated framework, their model efficiently integrates different input modalities, contextual data, and multitask learning in the video. They use the Actor-Action Dataset (A2D) to train and compare their model and show cutting-edge performance in segmentation and detection.

A versatile Temporal Shift Module (TSM) with great performance and efficiency has been examined by Lin et al. [11]. In particular, it can match 3D CNN's performance while retaining the intricacy of 2D CNN. TSM moves a portion of the channels down the temporal axis, facilitating communication between nearby frames. To achieve temporal modeling at zero computation and zero parameters, it can be placed into 2D CNNs.

A lightweight generator network was presented by Shou et al. [12] to improve the representation of the Discriminative Motion Cue (DMC) by reducing noise in motion vectors and capturing precise motion details. They used the downstream action classification objective, a reconstruction loss, and a generative adversarial loss to train the DMC generator to approximate flow, because optical flow provides a more accurate motion representation. The effectiveness of their method has been thoroughly evaluated on three action recognition benchmarks datasets.

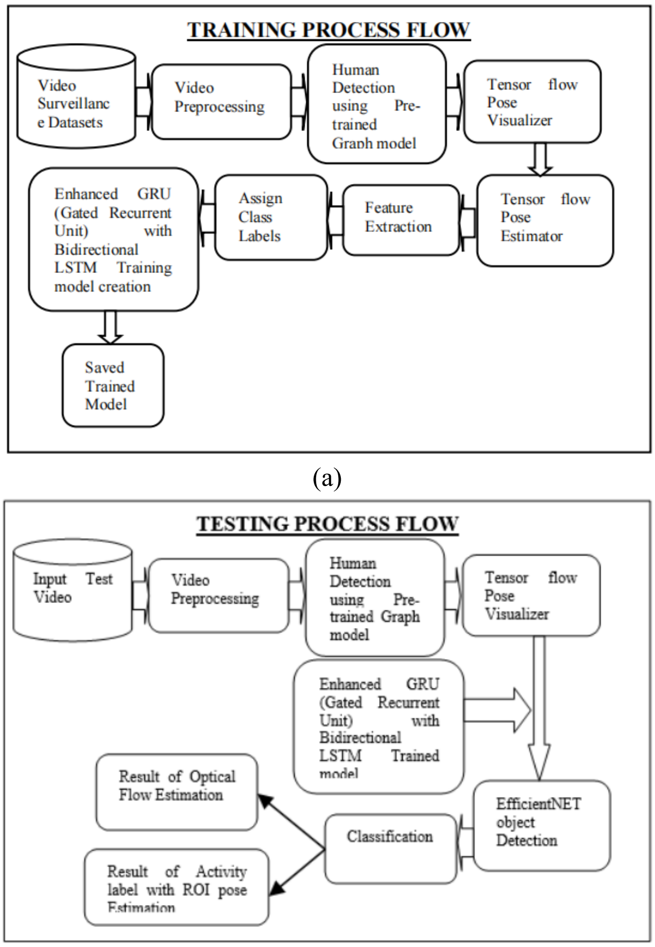
The recommended human activity recognition method by Alsarhan et al. [13] intends to identify the variety of human progress based on sensor data collected in human activity. Deep learning approaches for time series classification provide chances to avoid laborious handcrafted feature extraction methods, where the effectiveness and accuracy strongly rely on the quality of variables provided by subject matter experts. Recurrent neural networks were used by the author to identify human activity using accelerometer data from mobile phones. They utilized the bidirectional gated recurrent units mechanism specifically.

A sensor data based deep learning strategy for identifying human activities was proposed by Dogan et al. [14]. Eight transportation and locomotion activities are detected by our suggested recognition approach using linear accelerometer (LAcc), gyroscope (Gyr), and magnetometer (Mag) sensors. Still, Walk, Run, Bike, and Subway are among the eight activities.

Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) were combined to create a hybrid model by Imran Ullah Khan et al. [15] for activity recognition, where CNN is used for extracting spatial characteristics and LSTM network is used for learning temporal information.

3. PROPOSED METHODOLOGY

The proposed methodology presents a human activity recognition based on single and multiple activity models with ROI (Region of Interest) bounding box model. This system performs three stages namely, (i) Video preprocessing; (ii) Human detection Flow Estimation; and (ii) Activity Recognition using Enhanced Bidirectional GRU with LSTM (BGRU-LSTM) method. In this phase, video feature extraction and optical flow estimation using EfficientNet model extracts 36 features with help of Tensor flow Pose Visualizer and Tensor flow Pose Estimator. The optical flow method is used to tracking a human in frame by frame video with desired locations (width and height). With help of optical flow, the human movement's interactions time (frame by frame) also evaluated. After the feature extractions, the bidirectional GRU method is used to create a training model with corresponding feature data along with 21 classes. Figure 2 shows the overall BGRU-LSTM process in detail.


TRAINING PROCESS FLOW
Video Surveillanc e Datasets
Video Preprocessing
Human Detection using Pre-trained Graph model
Tensor flow Pose Visualizer
Enhanced GRU (Gated Recurrent Unit) with Bidirectional LSTM Training model creation
Assign Class Labels
Feature Extraction
Tensor flow Pose Estimator
Saved Trained Model
TESTING PROCESS FLOW
Input Test Video
Video Preprocessing
Human Detection using Pre-trained Graph model
Tensor flow Pose Visualizer
Enhanced GRU (Gated Recurrent Unit) with Bidirectional LSTM Trained model
Result of Optical Flow Estimation
Classification
EfficientNET object Detection
Result of Activity label with ROI pose Estimation

(a)

(b)

Figure 2. The Proposed Overall BGRU-LSTM Flow Diagram. (A) Training Model Preparation; (B) Testing HAR Prediction

3.1 Video frame extraction

Video based Human activity classification is the process of grouping videos based on similarities of contents. The video frame extraction is evaluated in frame by frame was already we presented in the study [16]. It has become significant in the current technology trends to contend the variety of user needs. Videos are basically composition of number of frames and it is observable that frames directly extracted from videos are comprised of time duration. In this method, Video frame extraction is the executed with real-time surveillance video's capture from the home apartment. The proposed method is to develop an efficient multiple HAR system for retrieving frames from videos on the basis of color using Hue Saturation Value (HSV). In each frame in the videos are converted from its color format in RGB color space to the proposed color space HSV of the three components of each one are obtained. In this color conversion model is help with optical flow estimation process.

3.2 Human detection using Graph model with Pose Estimation

The Graph Network Model (GNM) is a vital tool for processing non-Euclidean regions, and it can be used to identify humans [17]. GNM is widely utilized to evaluate human motion for many purposes because the frame composition is naturally defined by graph of set of nodes and links. By sequentially applying spatial and temporal convolutions (TCN) throughout the time and space domains, spatiotemporal graph convolutions provide GNM a new dimension. Figure 3 describes the Human Detection with GNM model (ROI Marked) result. The GNM learns how to use layer-wise propagation on structured data. Let us assume an undirected graph with M nodes, a group of edges among nodes, an adjacency matrix $Adj \in R^{M\times M}$, and a level matrix $L_{ij} = \sum_j Adj_{ij}$. If $y \in R^{F\times N}$ is the feature matrix of the graph, a mathematical model for the convolution of graphs is

$$f = \hat{L}^{-\frac{1}{2}}\widehat{Adj}\hat{L}^{-\frac{1}{2}}y_i W \qquad (1)$$

where $\widehat{Adj} = Adj+ID$, ID is the Identity matrix and $W \in R^{F\times C}$ is the weight matrix. So, if the input to a GNM layer is $F\times N$ the result feature f is $N\times O$, where O is the chosen output size. In this work, the spatial configuration partitioning established in ST-GCN [18], hence, $\widehat{Adj} = \sum_n Adj_n$ and Equation 1 is modified in a form of,

$$f = \sum_n \hat{L}_n^{-\frac{1}{2}} Adj_n \hat{L}_n^{-\frac{1}{2}} y W_n \qquad (2)$$

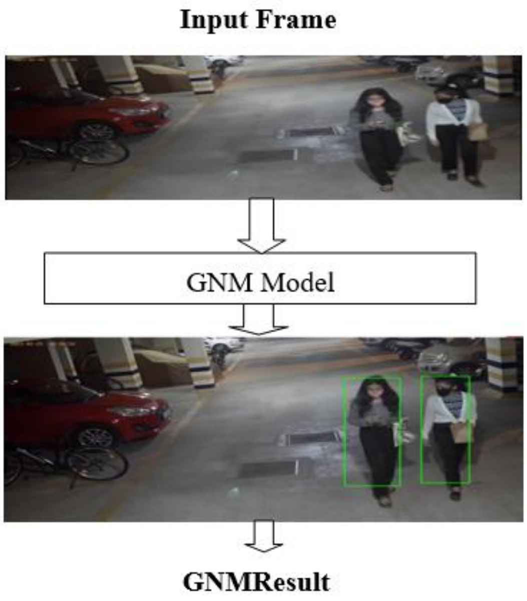


Figure 3. Human Detection with GNM model (ROI Marked) Result. Frame taken from real time surveillance video

In order to perform the proposed Human position estimation, the system first identifies the human key points of interest in the input video frame using pre-trained Graph Pose Refinement method. The complete human pose estimation flow describes in Figure 4. To construct a graph structure base on the pattern of human body, and they have clear adjacent relation with each other. Finding interest points or important areas in an image is the goal of key point detection. These could be the body joints (shoulders, wrists, and ankles) in a person, the corners and blobs in a video frame, or the facial landmarks (such as the tip of the nose, the corners of the eyes, the face's boundary, etc.). This method is crucial for many computer vision applications, including the comprehension of human behavior.

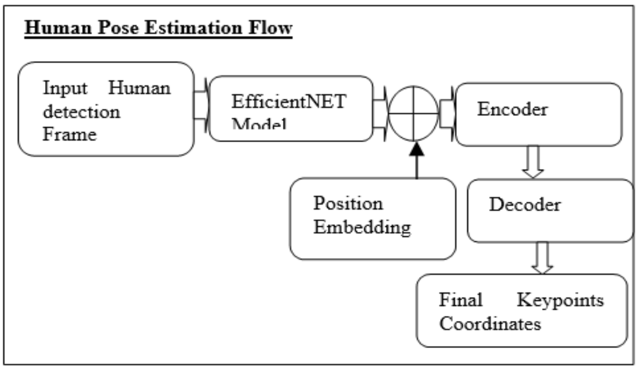
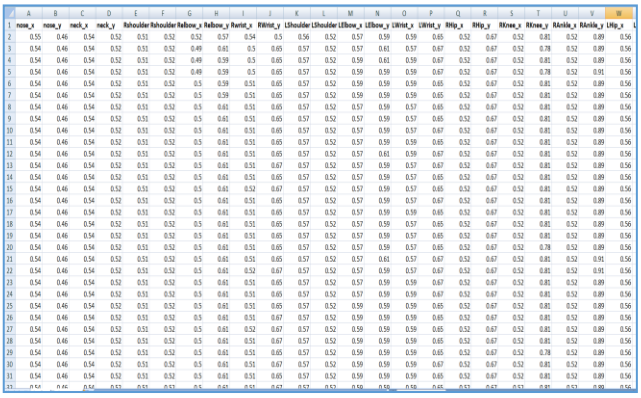

Human Pose Estimation Flow
Input Human detection Frame
EfficientNET Model
Encoder
Position Embedding
Decoder
Final Keypoints Coordinates

Figure 4. Human pose estimation process flow

EfficientNet transfer learning model is used to handle the human position estimation model. The reputation of EfficientNet is that it can achieve high accuracy with few parameters. The suggested technique is put into practice utilizing transfer learning and the weight has learned. In this model, optical flow or Heat Map is estimated to tracking a human in video at frame by frame with desired locations (width and height). With help of optical flow, the human movement's interactions time (frame by frame) also evaluated. In EfficientNet model, 36 human pose features are extracted (i.e., nose_x, nose_y, neck_x, neck_y, Rshoulder_x, Rshoulder_y, Relbow_x, Relbow_y, Rwrist_x, RWrist_y, LShoulder_x, LShoulder_y, LElbow_x, LElbow_y, LWrist_x, LWrist_y, RHip_x, RHip_y, RKnee_x, RKnee_y, RAnkle_x, RAnkle_y, LHip_x, LHip_y, LKnee_x, LKnee_y, LAnkle_x, LAnkle_y, REye_x, REye_y, LEye_x, LEye_y, REar_x, REar_y, LEar_x, Lear_y) with help of Tensor flow Pose Visualizer and Tensor flow Pose Estimator. The Attribute Characteristic of data extracted features format is Categorical and Real value. Figure 5 shows the feature extracted results. This method initially uses a person detector to get the bounding boxes of people, as in earlier research [19]. Then, each person is designated with a ROI (Region of Interest) from the input frame based on the boxes that were recognized.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	nose_x	nose_y	neck_x	neck_y	Rshoulder	Rshoulder	Relbow_x	Relbow_y	Rwrist_x	RWrist_y	LShoulder	LShoulder	LElbow_x	LElbow_y	LWrist_x	LWrist_y	RHip_x	RHip_y	RKnee_x	RKnee_y	RAnkle_x	RAnkle_y	LHip_x
2	0.55	0.46	0.54	0.52	0.51	0.52	0.52	0.57	0.54	0.5	0.56	0.52	0.57	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
3	0.54	0.46	0.54	0.52	0.51	0.52	0.49	0.61	0.5	0.65	0.57	0.52	0.57	0.61	0.57	0.67	0.52	0.67	0.52	0.78	0.52	0.89	0.56
4	0.54	0.46	0.54	0.52	0.51	0.52	0.49	0.59	0.5	0.65	0.57	0.52	0.59	0.61	0.59	0.67	0.52	0.67	0.52	0.81	0.52	0.89	0.56
5	0.54	0.46	0.54	0.52	0.51	0.52	0.49	0.59	0.5	0.65	0.57	0.52	0.59	0.59	0.59	0.67	0.52	0.67	0.52	0.78	0.52	0.91	0.56
6	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.59	0.51	0.65	0.57	0.52	0.59	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
7	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.59	0.51	0.65	0.57	0.52	0.59	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
8	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.57	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
9	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.57	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
10	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.57	0.67	0.52	0.67	0.52	0.81	0.52	0.89	0.56
11	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
12	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.61	0.59	0.67	0.52	0.67	0.52	0.81	0.52	0.89	0.56
13	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.67	0.57	0.52	0.57	0.59	0.57	0.67	0.52	0.67	0.52	0.81	0.52	0.89	0.56
14	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.57	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
15	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.52	0.67	0.57	0.52	0.57	0.59	0.57	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
16	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.57	0.67	0.52	0.67	0.52	0.81	0.52	0.89	0.56
17	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
18	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.57	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
19	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.57	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
20	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.57	0.65	0.52	0.67	0.52	0.78	0.52	0.89	0.56
21	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.61	0.57	0.67	0.52	0.67	0.52	0.81	0.52	0.91	0.56
22	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.52	0.67	0.57	0.52	0.57	0.59	0.57	0.67	0.52	0.67	0.52	0.81	0.52	0.91	0.56
23	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.59	0.67	0.52	0.67	0.52	0.81	0.52	0.89	0.56
24	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.57	0.59	0.59	0.67	0.52	0.67	0.52	0.81	0.52	0.89	0.56
25	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.59	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
26	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.67	0.57	0.52	0.59	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
27	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.52	0.67	0.57	0.52	0.59	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
28	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.67	0.57	0.52	0.59	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56
29	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.59	0.59	0.59	0.65	0.52	0.67	0.52	0.78	0.52	0.89	0.56
30	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.67	0.57	0.52	0.59	0.59	0.59	0.67	0.52	0.67	0.52	0.81	0.52	0.89	0.56
31	0.54	0.46	0.54	0.52	0.51	0.52	0.5	0.61	0.51	0.65	0.57	0.52	0.59	0.59	0.59	0.65	0.52	0.67	0.52	0.81	0.52	0.89	0.56

Figure 5. Human Pose Features extraction result

The proposed approach adapts to the end-to-end trainable, straightforward, and simple regression-based methods, which can address various heatmap-based method's shortcomings [19]. With the input frame with a multiple person, the previous heatmap-based techniques applied to the convolutional neural network P to the region to recognize keypoint heatmaps Hmap$\in R^{h\times w\times m}$ ($Hmap_m$ for m^{th} joint) of this person, where m is the amount of the expected keypoint.

$$Hmap = P(I) \tag{3}$$

Every pixel of *Hmap* signifies the human common position pixel is described in Figure 6. To get joints' coordinates $Jcord\in R^{2\times m}$ ($Jcord_m$ for m^{th} joint), those techniques regularly exploit the "taking-maximum" action to obtain the positions with high activations. Let *sl* be the spatial locations on *Hmap* as,

$$Jcord_m = arg \begin{matrix} max \\ sl \end{matrix} (Hmap_m(sl)) \tag{4}$$

A conventional CNN is a backbone for extracting multi-level feature encoder for capturing and fusing features for creating keypoint coordinate sequences. One such illustration is described in Figure 7. The localization accuracy of TFPose is not constrained by the feature map resolution because it is fully differentiable. The multi-level feature maps are denoted by C2, C3, C4 and C5, respectively, whose steps are 4, 8, 16 and 32, respectively. In order for the feature maps in this model to have an equivalent number of output channels, 32 features are extracted individually and applied to the feature maps using a 1× 1 convolution. The input feature F$\in R^{n\times c}$ to the initial encoder in the transformer, where n is the quantity of the pixel in the *F*, is created by flattening and concatenating these feature maps together.

To decode the preferred keypoint matches from the memory *M* is the goal of the decoder part. During training, the query matrix Q $\in R^{M\times c}$ is essentially an additional learnable matrix, whose rows each correspond to a keypoint and which is equally simplified with the model constraints. In particular, let the top decoder layer to anticipate the target coordinates directly. Following that, every subsequent decoder layer makes predictions that are then used to improve those made by its predecessor.

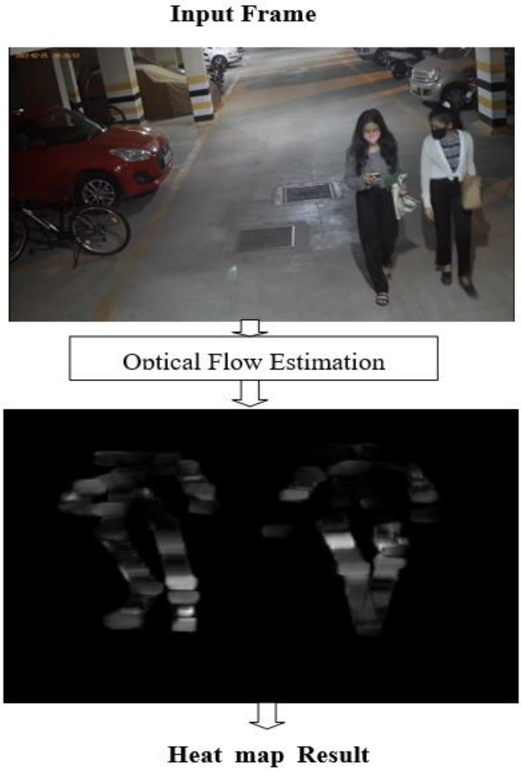


Figure 6. Optical flow or heat map result

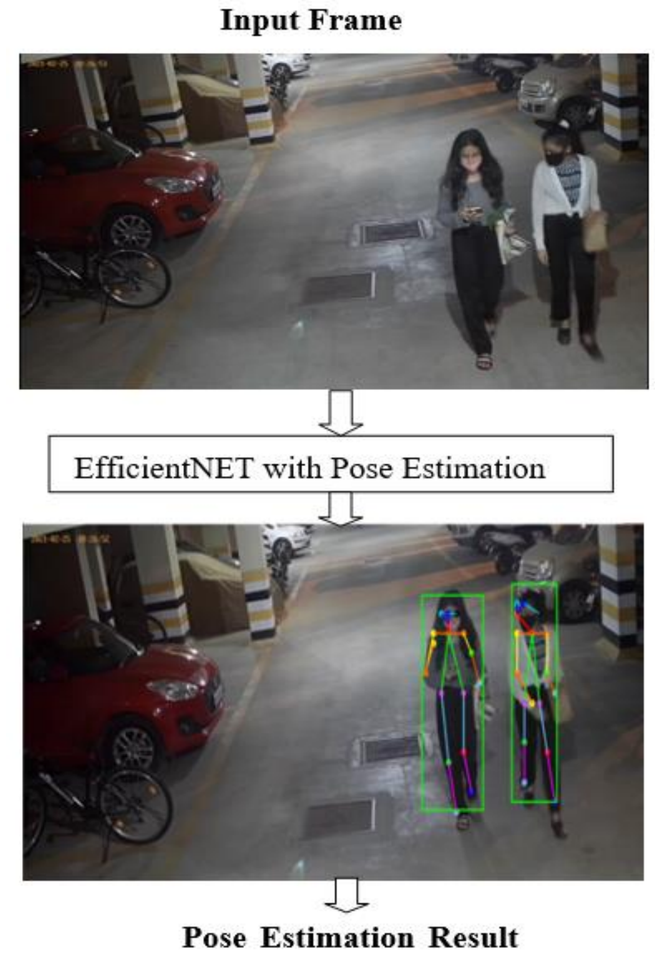


Figure 7. Human Detection with Pose Estimation model (ROI Marked) Result

3.3 Activity recognition using enhanced bidirectional GRU with LSTM (BGRU-LSTM) method

Tracking, focusing, and detecting every person in the video stream is a crucial step in identifying surveillance activity. This assignment is inadequate for object detectors trained on broad types of data. After the feature extractions, the bidirectional GRU method is used to create a training model with corresponding feature data along with 21 classes (i.e, Stand, walk, operate, fall_down, run, jump, motor_driving, walk_with_object, walk_with_holdingobject, siting, draggingsack, bikekickingpackethrowing, holdingbabym, tieinghair, driving, carryingbasket, haritieing, holdingbuckandbroomstick, holdingvesselwavingbag, puttinghelmet, running). The proposed method conducts Enhanced Bidirectional GRU with LSTM (BGRU-LSTM) model for human detection fine-tuning with new labelled data for this purpose, enabling it to function in a dynamic surveillance environment. The most often used neural network is LSTM. However, processing sequential data for real-time scenarios requires an unbearable amount of time due to its complicated gated structure and memory units. This work offered a faster, more accurate alternative to the Deep Skip Connection Gated Recurrent Unit (DS-GRU) network: a Bidirectional GRU with LSTM.

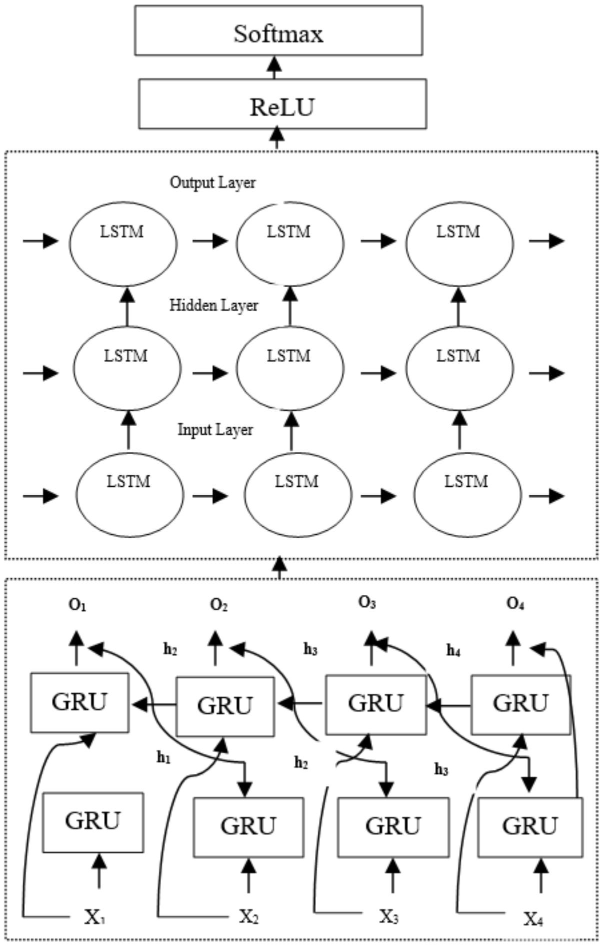


Figure 8. Structure of Bidirectional GRU with LSTM

The proposed bidirectional GRU based LSTM deep learning approach to recognize the multiple action of humans based on real-time surveillance video datasets. The proposed system placed two independent GRU with LSTM models together and the result is Bidirectional GRU with LSTM. Since the model is bidirectional, forward and backward cells were used. Figure 8 shows that both layers are independent except that they share the same input sequence (X_1-X_4) also the final outputs (O_1-O_4) from the two layers are concatenated. The input data are received by the input sequence of the GRU, and are output after passing through the LSTM. The data leaving the output layer of the GRU are input to the LSTM input layers for convolution. Next, they are operated by the ReLU layer and transferred to the fully connected layer. Finally, the classification result is output through SoftMax. The LSTM learns the residual nodes with reference to the hidden state. A Long Short Term Memory (LSTM), a type of RNN, primarily consists of three gates: an input gate, a forget gate, and an output gate [20]. Only a reset gate and a revise gate are utilized in the network of a bidirectional GRU, though. To appropriately reset historical data, the reset gate multiplies the value (0, 1) by the prior hidden layer using the sigmoid function as an output. The update gate, which calculates the proportion of update information in the past and present, resembles an LSTM's forget gate and input gate. In the update gate, which functions similarly to the input gate and forget gate of each LSTM, the amount of information at this stage is determined by the output to the sigmoid, which subtracts from 1 and multiplies the information from the hidden layer at the previous point. At this moment, the candidates are calculated during the candidate phase. The crucial part of this phase is to multiply the reset gate results without using the knowledge from earlier concealed layers. The unit now calculates the hidden layer by fusing the update gate result with the result for the hidden layer computation using sigmoid function. The Bidirectional GRU with LSTM structure, which contains two standard LSTM layers for extracting temporal dynamics from both forward and backward directions, is a major RNN variant in many applications, such as multiple human activity recognition. Table 1 describes the proposed BGRU-LSTM parameters description in detail.

```
Algorithm: BGRU-LSTM
Input: Input Video Frames $f$, Class $C$
Output: Multiple human action prediction result
Preparation:
    1. Video frame extraction
    2. Human Detection using Graph Network Model (GNM)
    3. Detecting Human Prediction Score
    4. Feature Extraction using EfficientNET with Heat map method
    5. Activity Recognition using Enhanced Bidirectional GRU with LSTM (BGRU-LSTM) method
    6. Compute Evaluation Time
Steps:
While (frames in video)
  1. Frame $f \leftarrow$ Video frame extraction
  2. $H \leftarrow$ Detecting only Humans using GNM method
  3. $Hmap \leftarrow$ Detecting optical flow map estimation between two frames
  4. forк = 1 to mdo // where m is collection of frames
        a. $f(k) \leftarrow$ test video frame.
        b. $H(f) \leftarrow$ GNM Model using equation 2 // Human Detection portions
        c. $ROI(H(f)) \leftarrow$ ROI Marked Humans by EfficientNET Model // Human Detection with multiple human pose estimation
        d. $C \leftarrow$ Prediction Class label with $ROI(H(f))$ using (BGRU-LSTM)
  5. Predicted activity $\leftarrow$ Result class label $C$
  6. Show the expected activity class in a frame that has a ROI.
End for
End While
```

Table 1. Parameters description

Parameters	Symbol	Value
Overall Class	C	21 action classes
Sample Duration	Sample_duration	Each iteration has 16 frames.
Sample Size	Sample_size	Pixel wide at 112
Training Model	*M*	500
Input video frame	*f*	-
tlooping variable	*t*	1 to *n*
n	*n*	Number of frames

4. EXPERIMENTAL RESULTS

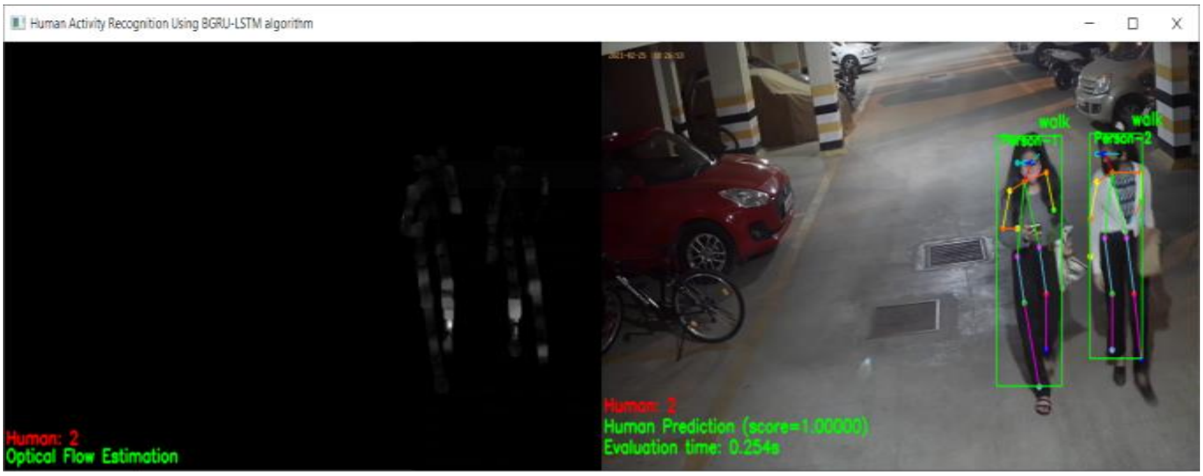
Using the proposed BGRU-LSTM technique, the results have been estimated. The results are implemented using a Windows 10 computer running python 3.7 simulations with an Intel I5-6500U series processor running at 2.71 GHz with 8GB of main memory. This paper is implemented with by real-time surveillance video dataset capture from the Bangalore home apartment. In these real-time videos contains 21 action recognition training dataset with different classes. The proposed multiple HAR result of walking video with Human prediction score and Evaluation time is shown in Figure 9. The resulting parameters of walking activity and overall BGRU-LSTM training accuracy with loss are described in Figure 10 and Figure 11.

In the case of multiple human activity classification, the proposed system computes a separate loss for each class label per observation and then adds the results.

$$Loss = -\sum_{c=1}^{M} y_{o,c} log(p_{o,c}) \tag{5}$$

where, M - number of classes; log - the natural log; y - binary indicator (0 or 1) if class label c is the correct classification for observation o; p - predicted probability observation o is of class c.

In order to learn sequential patterns, the proposed model undergoes training for 120 times. The learning rate starts at 0.01, which is decreased by a factor of 10 after 50 iterations, and stochastic optimization is utilized to minimize costs while preventing overfitting by setting dropout to 0.5. The effectiveness of the BGRU-LSTM and DS-GRU [20] on the real-time surveillance video dataset is shown in Figure 12.



Human: 2
Optical Flow Estimation

Human: 2
Human Prediction (score=1.00000)
Evaluation time: 0.254s

Figure 9. Multiple HAR result of walking video with Human prediction score and Evaluation time

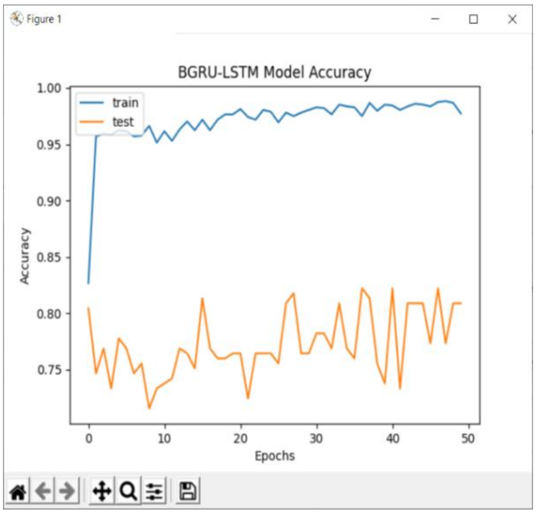


Figure 10. Overall BGRU-LSTM training accuracy plot result

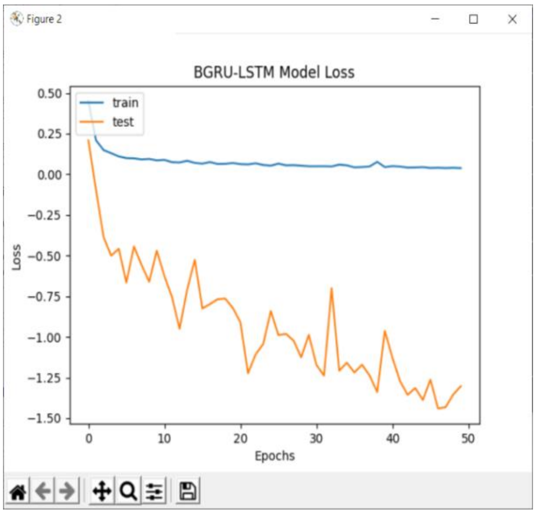


Figure 11. Overall BGRU-LSTM training Loss plot result

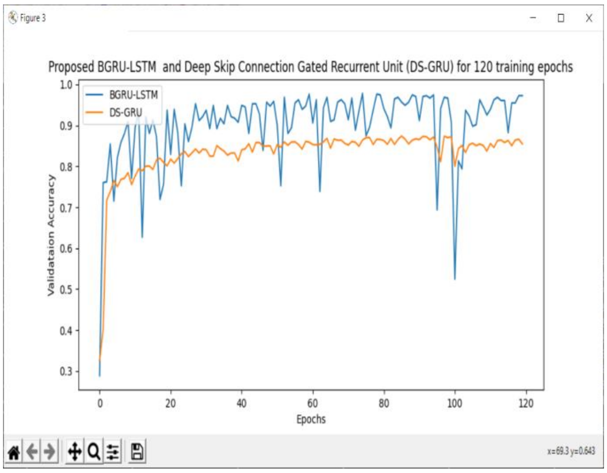
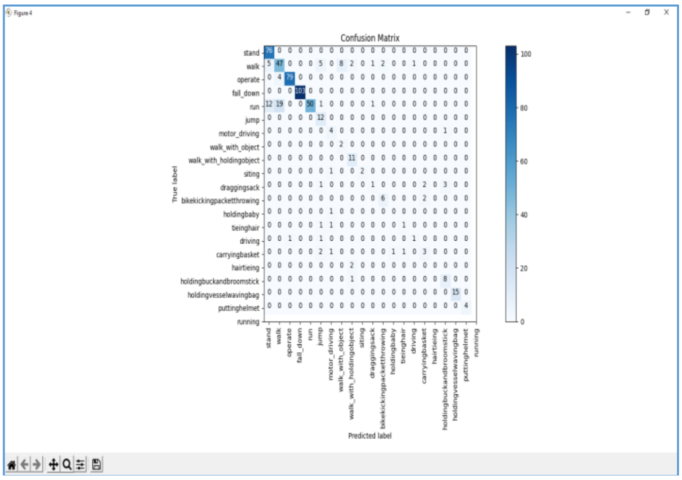


Figure 12. Overall Validation accuracy chart of proposed BGRU-LSTM with DS-GRU method plot result

Figure 13 shows the confusion matrix of proposed BGRU_LSTM true label of 21 classes of real-time surveillance video dataset.



	stand	walk	operate	fall_down	run	jump	motor_driving	walk_with_object	walk_with_holdingobject	siting	draggingsack	bikekickingpacketthrowing	holdingbaby	tieinghair	driving	carryingbasket	hairtieing	holdingbuckandbroomstick	holdingvesselwavingbag	puttinghelmet	running
stand	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
walk	5	47	0	0	0	5	0	8	2	0	1	2	0	0	1	0	0	0	0	0	
operate	0	4	79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
fall_down	0	0	0	103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
run	12	19	0	0	50	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
jump	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
motor_driving	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	1	0	0	
walk_with_object	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	
walk_with_holdingobject	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	
siting	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	
draggingsack	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	2	0	3	0	0	
bikekickingpacketthrowing	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	2	0	0	0	0	
holdingbaby	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
tieinghair	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	
driving	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
carryingbasket	0	0	0	0	0	2	1	0	0	0	0	0	1	1	0	3	0	0	0	0	
hairtieing	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	
holdingbuckandbroomstick	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	8	0	0	
holdingvesselwavingbag	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	
puttinghelmet	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	
running																					

Figure 13. Proposed BGRU-LSTM Confusion matrix plot result

For the real-time surveillance video dataset, the proposed approach offers balanced precision and recall scores, showing less true negatives and false negatives. For the real-time surveillance video dataset, the BGRU-LSTM methodology attained F1-scores of 97.62%, demonstrating its effectiveness in comparison to state-of-the-art methods. The 21 classes of the real-time surveillance video dataset's proposed BGRU LSTM true label are shown in Table 2 and Figure 14 along with their precision, recall, accuracy, and F1 score.

Table 2. Comparison of Evaluation metrics of existing DS-GRU with proposed BGRU-LSTM method

Methods	Precision	Recall	Accuracy	F1-Score
DS-GRU	91.52	91.11	84.91	91.31
BGRU-LSTM	**98.40425**	**96.8586**	**97.6**	**97.6253**

The proposed BGRU-LSTM technique results of real-time surveillance video dataset are described in Figure 15 and Figure 16.

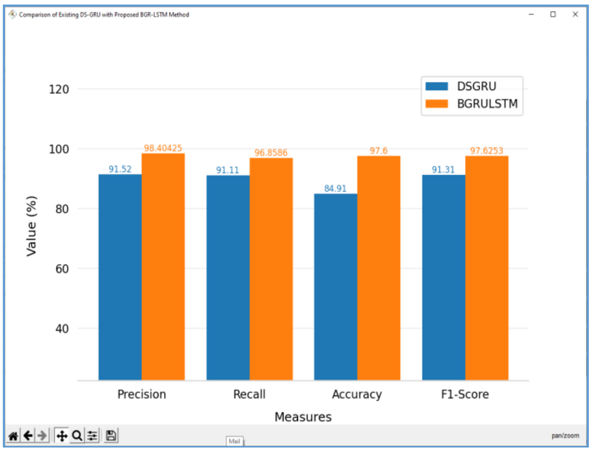


Figure 14. Precision, recall, and F1-score are used to evaluate the proposed BGRU-LSTM with existing DS-GRU method.

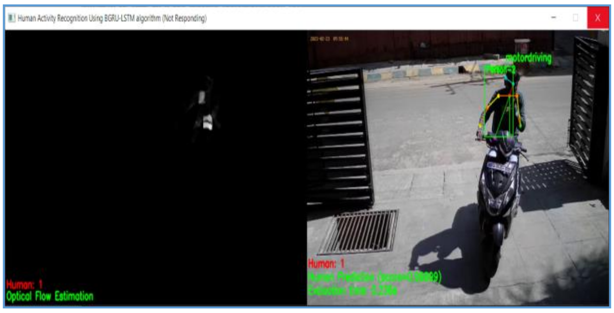

Human Activity Recognition Using BGRU-LSTM algorithm (Not Responding)
motordriving
Human: 1
Optical Flow Estimation
Human: 1

Figure 15. Multiple HAR result of motor driving video with Human prediction score and Evaluation time

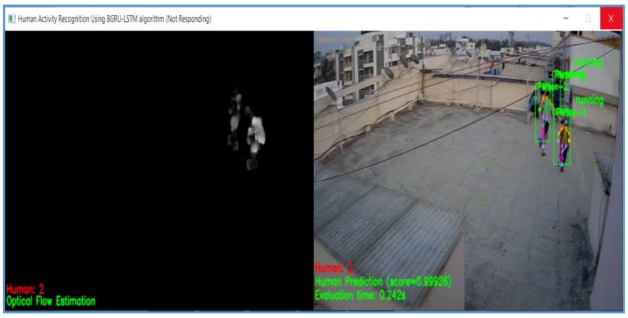

Human Activity Recognition Using BGRU-LSTM algorithm (Not Responding)
running
Person-2
running
Human: 2
Optical Flow Estimation
Human: 2
Human Prediction (score=0.99926)
Evaluation time: 0.242s

Figure 16. Multiple HAR result of running video with Human prediction score and Evaluation time

5. CONCLUSION

The development of multiple Human Activity Recognition (HAR) concepts using deep neural network technology was examined in this paper. The enhanced BGRU-LSTM algorithm presented in the proposed study is customized for the HAR challenge. By utilizing the robustness of the EfficientNET feature extraction model with classification, this system aims to increase the accuracy of numerous human activities with Pose estimation. We applied the BGRU-LSTM approach in the multiple HAR system via real-time surveillance video dataset acquisition from the home apartment, and the results were outstanding and effective. The result, which is given as 97.60%, is more recognized than the other HAR methods, such as Deep Skip Connection Gated Recurrent Unit (DS-GRU) and Depthwise Separable Convolution (DSC) with Bidirectional Long Short-Term Memory (DSC-BLSTM).

REFERENCES

[1] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Li, F.F. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725-1732.

[2] Simonyan, K., Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. Advances in Neural Information Processing Systems, 27: 568-576.

[3] Li, S., Seybold, B., Vorobyov, A., Lei, X., Kuo, C.C.J. (2018). Unsupervised video object segmentation with motion-based bilateral networks. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 207-223.

[4] Dang, K., Zhou, C.L., Tu, Z.G., Hoy, M., Dauwels, J., Yuan, J.S. (2018). Actor-Action Semantic Segmentation with Region Masks. The British Machine Vision Conference (BMVC). https://doi.org/10.48550/arXiv.1807.08430

[5] Lin, T., Zhao, X., Su, H., Wang, C., Yang, M. (2018). BSN: Boundary sensitive network for temporal action proposal generation. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 3-19.

[6] Ding, M., Zhao, A., Lu, Z., Xiang, T., Wen, J.R. (2019). Face-focused cross-stream network for deception detection in videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7802-7811.

[7] Lara, O.D., Labrador, M.A. (2012). A survey on human activity recognition using wearable sensors. IEEE Communications Surveys & Tutorials, 15(3): 1192-1209. https://doi.org/10.1109/SURV.2012.110112.00192

[8] Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. Pattern Recognition Letters, 119: 3-11. https://doi.org/10.1016/j.patrec.2018.02.010

[9] Alwassel, H., Heilbron, F.C., Ghanem, B. (2018). Action search: Spotting actions in videos and its application to temporal action localization. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 251-266.

[10] Ji, J., Buch, S., Soto, A., Carlos Niebles, J. (2018). End-to-end joint semantic segmentation of actors and actions in video. In: ECCV. pp. 702-717

[11] Lin, J., Gan, C., Han, S. (2019). Tsm: Temporal shift module for efficient video understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7083-7093.

[12] Shou, Z., Lin, X., Kalantidis, Y., Sevilla-Lara, L., Rohrbach, M., Chang, S.F., Yan, Z. (2019). Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1268-1277.

[13] Alsarhan, T., Alawneh, L., Al-Zinati, M., Al-Ayyoub, M. (2019). Bidirectional gated recurrent units for human activity recognition using accelerometer data. In 2019 IEEE SENSORS, 1-4. https://doi.org/10.1109/SENSORS43011.2019.8956560

[14] Dogan, G., Ertas, S.S., Cay, İ. (2021). Human Activity Recognition Using Convolutional Neural Networks. In 2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 1-5. https://doi.org/10.1109/CIBCB49929.2021.9562906

[15] Khan, I.U., Afzal, S., Lee, J.W. (2022). Human activity recognition via hybrid deep learning based model. Sensors, 22(1): 323. https://doi.org/10.3390/s22010323

[16] Jitha Janardhanan, Umamaheswari, S. (2022). Vision based Human Activity Recognition using Deep Neural Network Framework. International Journal of Advanced Computer Science and Applications (IJACSA), 13(6): 165-71. https://doi.org/10.14569/IJACSA.2022.0130621

[17] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y. (2020). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and

Learning Systems, 32(1): 4-24. https://doi.org/10.1109/TNNLS.2020.2978386

[18] Yan, S., Xiong, Y., Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In Thirty-second AAAI Conference on Artificial Intelligence.

[19] Luo, Z., Wang, Z., Huang, Y., Wang, L., Tan, T., Zhou, E. (2021). Rethinking the heatmap regression for bottom-up human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13264-13273.

[20] Ullah, A., Muhammad, K., Ding, W., Palade, V., Haq, I.U., Baik, S.W. (2021). Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. Applied Soft Computing, 103: 107102. https://doi.org/10.1016/j.asoc.2021.107102