IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 April 2023, accepted 23 June 2023, date of publication 3 July 2023, date of current version 10 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3291395

RESEARCH ARTICLE

Object-Based Hybrid Deep Learning Technique for Recognition of Sequential Actions

YO-PING HUANG[1,2,3,4], (Fellow, IEEE), SATCHIDANAND KSHETRIMAYUM[1], AND CHUN-TING CHIANG[1]

[1]Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
[2]Department of Electrical Engineering, National Penghu University of Science and Technology, Penghu 88046, Taiwan
[3]Department of Computer Science and Information Engineering, National Taipei University, New Taipei City 23741, Taiwan
[4]Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan

Corresponding author: Yo-Ping Huang (yphuang@gms.npu.edu.tw)

This work was supported in part by the National Science and Technology Council, Taiwan, under Grant MOST108-2221-E-346-006-MY3 and Grant MOST111-2221-E-346-002-MY3; and in part by the AU Optronics Corporation Research Projects under Grant 209A221 and Grant 210A212.

ABSTRACT Using different objects or tools to perform activities in a step-by-step manner is a common practice in various settings, including workplaces, households, and recreational activities. However, this approach can pose several challenges and potential hazards if the correct sequence of actions is not followed and the object or tool is not used in the appropriate sequence; therefore, it must be addressed to ensure safety and efficiency. These issues have garnered significant attention in recent years. Previous research has relied on using body keypoints to detect actions, but not the objects or tools used during activity. As a result, the lack of a system to identify the target objects or tools being used while performing tasks increases the risk of accidents and mishaps during the process. This study suggests a possible solution to the aforementioned issue by introducing a model that is both efficient and durable. The model utilizes video data to monitor and identify daily activities, as well as the objects involved in the process, thus enabling real-time feedback and alerts to enhance safety and productivity. The suggested model separates the overall recognition process into two components. Firstly, it utilizes the advanced BlazePose architecture for pose estimation, and interpolates any undetected and wrong-detected landmarks to enhance the precision of the posture estimation. After this, the features are forwarded to a long short-term memory network to identify the actions performed during the activity. Secondly, the model also employs an enhanced YOLOv4 algorithm for object detection, to accurately identify the objects used in the course of the activity. Finally, a durable and efficient activity recognition model has been developed, which achieves 95.91% accuracy rate in identifying actions, a mean average precision score of 97.68% for detecting objects, and overall activity recognition model that is capable of processing at a rate of 10.47 frames per second.

INDEX TERMS Human activity recognition, long short-term memory (LSTM), object detection, pose estimation, standard operating procedures (SOPs).

I. INTRODUCTION

Performing activities that involve different human actions and objects require careful attention to safety and efficiency. If the appropriate action sequence and the correct object or tool are not used, it can pose significant challenges and potential hazards. For example, using power tools without following the proper sequence can lead to accidents, such as injuries from blades or bits, or damage to the workpiece. Mishandling hot surfaces, not allowing appliances to cool down, or improper use of heat sources can result in burns or scalds. These challenges and hazards must be addressed to ensure that the activity is carried out safely and efficiently. To address these concerns, human activity recognition (HAR) can be used to monitor the activity and ensure that it follows standard operating procedures (SOPs) that outline

The associate editor coordinating the review of this manuscript and approving it for publication was Kostas Kolomvatsos.

step-by-step processes to complete the task. Human pose estimation (HPE) is a popular research field in computer vision that plays a significant role in activity recognition [1], [2], [3]. The majority of these techniques rely on using optical sensors to take RGB images in order to determine body landmarks and the overall position. It is also possible to combine with other computer vision technologies for 3D animation, fitness, virtual and augmented reality, and rehabilitation [4], [5], [6].

HAR on the other hand, is a crucial computer vision task that enables machines to examine the identified body landmarks from HPE models and comprehend various human activities [7], [8], [9]. Many researchers have been driven to advance HAR systems in real-world setting by the rapid growth of artificial intelligence, smart phones, and CCTV systems. This drive has been motivated by the role of HAR systems in health, security and behavioral studies. Some of their applications include patient monitoring systems [10], [11], ambient assisted living (AAL) [12], [13], surveillance systems [14], [15], gesture recognition [16], [17], behavior analysis [18], and a range of healthcare systems [19], [20]. In particular, vision-based human activity recognition systems, which evaluate input in the form of video or image to identify performed activities are quite complicated. This is because the appearance of the body changes dynamically due to various types of clothing, occlusions caused by viewing angles, background context, etc. [21]. And the performance would be poor if the occlusion is very high. It is also interesting to note that the majority of current studies only address the recognition of an action, and none really gives insight about the object they use during the activity.

Fig. 1 shows some example pictures of confusing cases, where a person performs an action with and without object, and their skeleton representation generated from body landmarks. The physical differences between some actions are very small or even identical, making it difficult to identify activities that are identical yet interacting with different objects, such as in households, recreational, workplace activities of persons involving machine operation, material movement, maintenance, assembly, product and process design, etc.

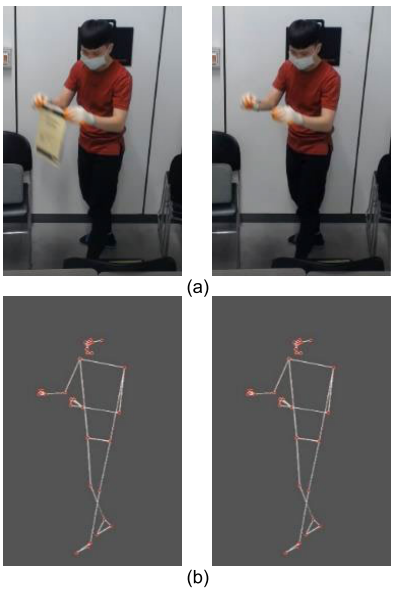

FIGURE 1. Example of a confusing case for action detection. (a) RGB images of a person performing an action with (left) and without (right) object, and (b) corresponding skeletal representations.

Therefore, with the growing popularity of HAR and object detection in the computer vision field, it is better to have a system that can accurately recognize actions sequence in an activity as well as detect objects used during the activity will be of profound benefit. This would aid in analyzing and monitoring a person's activity to determine if they are adhering to the SOPs with appropriate objects.

The goal of this research is to create an activity recognition model for a person from video information that can detect their actions sequence as well as the objects being used while they are performing an activity. To achieve this, a person's pose estimation is discovered using BlazePose [22] and undetected or wrong-detected landmarks were interpolated using linear interpolation method, then the information is processed by a recurrent neural network that can learn sequential order dependency, known as long short-term memory (LSTM). Object detection method is carried out in the second part using an enhanced YOLOv4 algorithm to recognize the object in the person's hand while they are performing the activity. Finally, a lightweight and robust system for recognizing person's activities is created by combining the two models. Fig. 2 depicts the suggested architecture. Three challenges are considered to be resolved in this study: (1) human pose estimation-based action detection using LSTM, (2) an object detection model to detect objects being used in an activity, and (3) an activity recognition model to classify the overall activity.

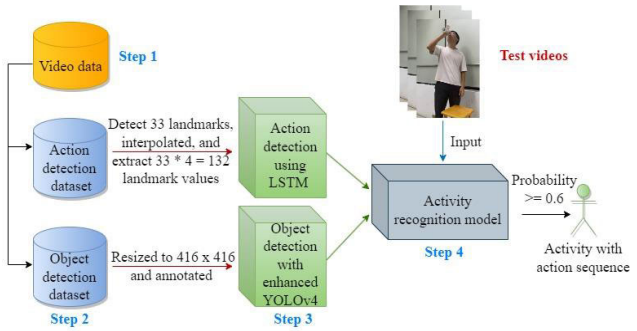

FIGURE 2. Proposed activity recognition framework.

The followings are the key contributions in this study:

1) An action recognition technique is proposed that utilizes body landmark information from the sequence of frames. We further detect object being used in order to make recognition system more informative.

2) We proposed a technique to improve the accuracy of person's pose estimation by interpolating the undetected and wrong-detected landmarks.
3) The object detection algorithm is further enhanced by introducing extra YOLO head to detect the various object of different shape and size used by the person while performing the activity.
4) An activity recognition model is developed that can recognize different actions performed within the activity in chronological order, and in accordance with the predefined SOPs as well as the object being used.

This paper is organized as follows. A comprehensive literature review of existing related work is provided in Section II. The proposed methodology is described in Section III. Section IV presents the training dataset, experimental results and discussions. In Section V conclusion and future research are given.

II. RELATED WORK

Artificial intelligence (AI) models that estimate body key points to characterize body position have become a potentially effective tool for assessing human actions. More specifically, convolution neural networks (CNN) are frequently used in human pose estimation to forecast a person's position by performing inference on input videos or images [1], [2]. Due to the numerous conceivable human positions, the high degree of freedom, appearance changes like illumination and clothing, environmental changes, and occlusions, determining precise pixel coordinates of body keypoints is a challenging process [3]. Despite these challenges, a number of reliable models have been developed that function admirably in applications including sports training, rehabilitation, and fall detection [4], [5], [6]. While pose estimation models have been successful in other applications, it is still needed to be able to accurately identify keypoints in order to track person's activity because engaging in the wrong activity might have side effects on the production lines.

For body joint coordinate-based action recognition, the human pose estimation problem is formulated as a CNN-based regression problem toward body joints by the holistic model DeepPose [23]. Additionally, it employs a cascade of these regressors to improve the pose estimation. However, regression to XY location is challenging and raises learning complexity, which inhibits generalization and results in subpar performance in some regions. A real-time multi-person posture estimation architecture made for the desktop settings, called OpenPose [24], was proposed as a solution, which is commonly used in the pose estimation community. It generates a feature representation by first analyzing the image using the first 10 layers of VGG-19 architecture. The captured feature representation is then fed into a two-branch multi-level CNN to generate part confidence maps and vector fields of part affinities. One branch forecasts a collection of 2D body part confidence maps. The other branch indicates the relationship of parts through 2D vector fields of part affinities. These two branches are used to carry out K-partite graph matching for multi-person pose estimation. The primary drawbacks of this system, despite processing at 0.4 frames per second, it demands a lot of computational power and is difficult to work on real-time videos. A two-step detector-tracker inference pipeline is used by Google's (Mountain View, CA) BlazePose model [22], where the detector is employed in the initial frame and tracker is used to follow the person in consecutive frames until the person is discovered. In order to predict heatmaps for each joint in this model, it has employed an encoder-decoder network design followed by another encoder that regresses directly to the coordinates of all joints. It is ideal to estimate human pose for activity recognition due to its lightweight design and real-time inference capability. However, it may fail to detect body landmarks due to high changes in appearance, clothing and occlusions.

Recent advances in effective motion capture technologies and posture assessment algorithms have made it easier to obtain information about human joint coordinates. As a result, joint coordinate-based action recognition using deep learning methods has significantly outperformed previous methods in recent years and has become the standard approach. Recurrent neural network (RNN) [25] is now one of the most used frameworks in joint coordinate-based action recognition because of its ability to analyze sequential data. A hierarchical RNN network [26] was proposed to classify activities based on skeleton's data. An advanced LSTM network [27] that is fully coupled and includes the regularization strategy was developed to acquire the high-level temporal aspects of skeleton information. All these approaches rely on the RNN architecture, and these features aim to improve action recognition while failing to recognize the object being used. Thus, many significant recognition errors are occurred among physically similar classes of person activity. The primary cause of these recognition errors is that these activities differ by tiny or similar body movements yet interaction with different objects.

Our work belongs to activity recognition, but more focus on both body movement of the person and interacted objects, that has not been considered in the above methods. In this study, we modified YOLO (you only look once) [28] to enhance its ability to detect various objects of different shapes and sizes that are used by individuals while performing activities. The proposed method is a single convolutional network that predicts multiple bounding boxes and class probabilities from a single image frame in a single evaluation. By improving the accuracy of object detection, our model can provide a more comprehensive understanding of the actions being performed. This makes the proposed model suitable for a wide range of applications, including human activity recognition and surveillance.

III. PROPOSED METHODOLOGY

The proposed approach aims to develop a framework that is both lightweight and robust for classifying sequential actions in an activity. This framework focuses on capturing

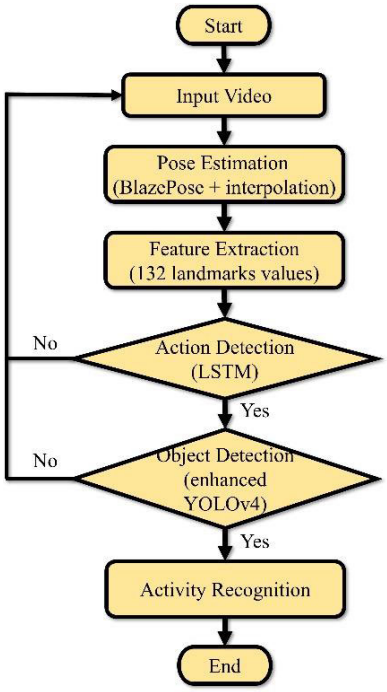

FIGURE 3. Block diagram of the activity recognition system.

not only the body movements but also the objects that they interact with during the activity. The proposed architecture consists of five components including pose estimation, feature extraction, action detection, object detection, and activity recognition as shown in Fig. 3. Initially, the video data was split into individual frames, following by pose estimation using the BlazePose architecture which returned 33 landmarks of a person (Fig. 4). Then, any undetected and wrong-detected landmarks were interpolated to enhance the precision of the posture estimation. The landmark values are then saved as frame values to represent a sequence of events for an activity. For the purpose of understanding the temporal components, the transformed landmark values are subsequently fed into novel LSTM layers and finally

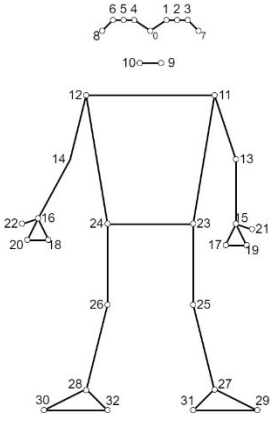

FIGURE 4. A 2D skeletal topology with 33 landmarks.

to the SoftMax layer to return a probability of each action. Furthermore, an enhanced YOLOv4 algorithm by adding an additional prediction head to improve the detection of small objects and handle variations in object sizes is conducted to detect the objects used during the activity. Finally, an algorithm for activity recognition is developed by utilizing the chronological sequence of actions in accordance with the predefined SOPs. This approach ensures that the algorithm can identify the correct sequence of actions and compare it with the established procedures to determine the accuracy and efficiency of the performed activity. In the following sections will give detailed explanations of the steps stated above.

A. POSE ESTIMATION FRAMEWORK

Human pose estimation and tracking are crucial in a wide range of fields, including health monitoring, surveillance systems, and gestural control. However, in computer vision, it faces challenges like detecting, associating, and tracking semantic key points, such as ''right shoulders,'' ''left knees,'' or ''left elbow.'' These problems can be solved by using deep learning models to recognize and track human body language through posture detection and tracking. Furthermore, CNN-based models are the most efficient image processing methods available today [29]. Therefore, the most advanced methods often rely on the development of a CNN architecture specifically designed for human posture detection. Pose estimation methods can be classified to top-down and bottom-up approaches. In a bottom-up approach, each joint of the body is evaluated individually before combining them into a distinct pose. DeepCut [30] was the first to use bottom-up approaches. In contrast, top-down approach begins with a person detector and estimate body joints within the detected bounding boxes. Although pose estimation has huge practical ramifications, it is challenging to estimate strong articulations, smaller, hardly perceptible joints, occlusions, clothes, and lighting changes. However, significant progress has been made in predicting human pose, which allows for the strongest assistance of the numerous practical applications.

In this study, a powerful, robust and lightweight CNN optimized top-down human pose estimation architecture is implemented for the real-time detection. To achieve this, the heatmaps and offsets from earlier frames of the person performing actions are used. We utilize a two-step machine learning pipeline: a detector and a tracker for the person who is performing the actions. Since the face provides the greatest information regarding the position of the torso, the neural network of the pose estimation executes from the first frame until the person's face is detected. The tracker is then used to track the person while performing the actions as shown in Fig. 5.

For the person pose tracking, inspired by [31], in the process of obtaining the landmarks of the entire human body, we utilize two more virtual keypoints to accurately define the human body's center, rotation, and scale as a circle. This is consequently capable of predicting a person's hips midpoint,

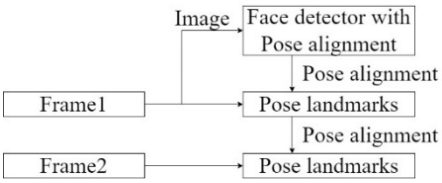

FIGURE 5. Overview of the pose detector pipeline.

the radius of a circle that encloses the entire body, and angle of inclination of the line joining the midpoint of the shoulders and hips [31]. This also helps in tracking extremely complex situations in any kind of person's activities.

The model used an encoder-decoder network architecture to predict heatmaps for every joint of the person, followed by a second encoder that regresses back to every landmark (joint's coordinates). Then, to make this model lightweight enough to run on a low-end computer, heatmap output is removed during inference as shown in Fig. 6. A list of 33 landmarks is returned by the architecture. The landmarks are represented as x, y, z, and v, the visibility. The coordinates (x and y) show where a particular joint of the person is located within the normalized range between 0 and 1 of the image's width and height. z stands as depth of the landmark, having origin as the depth at the center of hip. The term v describes whether or not a landmark can be seen in the frame. The scale and position of the person have an impact on the landmarks that the pose estimation network generates for it. Therefore, they are transformed to become independent of the position and scale in the frame. As a result, the same person in the same action may provide different landmark values in different frames depending on where they are in the frame. We grab these landmark values and save them as frame values to represent a sequence of events for an activity. For an activity video, $V^m = [F_1, F_2, \ldots, F_n]$ is a matrix of pose-vectors with K landmarks, where V^m contains n frames of change of the person conducting the actions. Each frame is consisted of:

$$F_i = \left[l_i^1, l_i^2, \ldots, l_i^K\right], i \in [1, n] \tag{1}$$

Since our model can generate 33 landmarks ($K = 33$), then the resulting vector has a length of 132 landmark values and format:

$$F_i = [x_i^1, y_i^1, z_i^1, v_i^1, x_i^2, y_i^2, z_i^2, v_i^2, \ldots, x_i^{33}, y_i^{33}, z_i^{33}, v_i^{33}] \tag{2}$$

Depending on the photography settings and conditions, landmarks might not be detected or wrong-detected when we use pose estimation models based on CNN to a video taken by a general camera. Action detection and analysis are negatively impacted by this kind of inaccurate landmark detection. To overcome this issue, in conjunction with the BlazePose architecture, we have incorporated innovative interpolation techniques. These techniques play a crucial role in enhancing the accuracy of posture estimation by effectively addressing any undetected or wrong-detected landmarks. Through

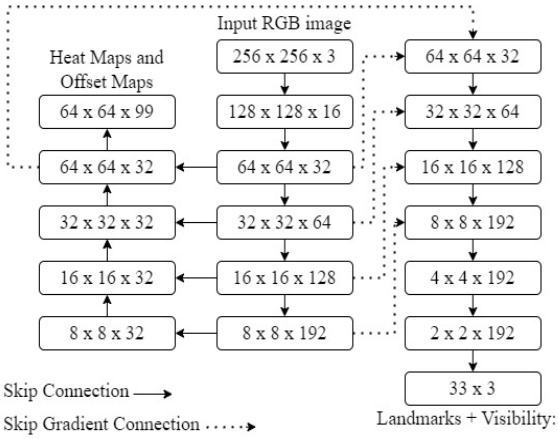

FIGURE 6. Architecture of the landmark detector network.

interpolation, we fill in the gaps and correct any inaccuracies, ultimately boosting the overall precision of the posture estimation process. To address this, we use time series correlations between identical body joints across several frames, because the estimated human position is a collection of time series data.

When landmarks in Blazepose are unable to be detected, their x and y coordinate values will always be 0. In this study, for the person w's landmark l_w^f in f frame, although l_w^{f-1} and l_w^{f+1} are detected, but l_w^f is not, we represent f frame as "undetected landmark frame" f'.

$$f' = f \tag{3}$$

where $l_w^f = (0, 0)$, $l_w^{f-1} \neq (0, 0)$, and $l_w^{f+1} \neq (0, 0)$

Similarly, for person w's landmark l_w^f in frame f, although l_w^{f-1} and l_w^{f+1} are detected, but l_w^f is wrong-detected, we represent frame f as "wrong-detected landmark frame" f''. We emphasize on the difference δ^f that is provided as the landmark's l_w spatial distance between two consecutives $f-1$ and f frames. The fixed number of pixels is given as the difference δ^f. Due to the possibility of resolutions and frame rates varying based on the input video, we do not wish to specify a threshold for δ^f. As a result, we set a threshold θ to give importance to the ratio of the difference δ^f and δ^{f-1}.

$$f'' = f \tag{4}$$

where $\delta^f > \theta \cdot \delta^{f-1}$, $l_w^{f-1} \neq (0, 0)$, and $l_w^{f+1} \neq (0, 0)$

The percentage of wrong-detected frame which were not wrong-detected frames was lower when the threshold was set to $\theta = 3$. As a result, we use $\theta = 3$ as the threshold in this study so that only frames that are clearly wrong-detected are interpolated. In this manner, we represent wrong-detected landmark frames according to the relative number of changes for every landmark. Both undetected and wrong-detected landmark frames will be interpolated using the previous and following frames' landmark coordinate information.

It is crucial to extract person's coordinate values from various frames so as to interpolate coordinate values. We use

linear interpolation to interpolate landmarks for undetected and wrong-detected landmark frames. This is based on the observation that person action does not change significantly over a short period of time. In most cases, the undetected or wrong-detected landmark l_w^f will be located close to the midpoint of landmarks l_w^{f-1} and l_w^{f+1}.

For undetected f' frame, let the landmark of person $w_{f'}$ in $f'-1$ and $f'+1$ frame be $l_w^{f'-1}$ and $l_w^{f'+1}$, respectively. We perform the linear interpolation to landmark $l_w^{f'}$ which both x and y coordinate values of the person $w_{f'}$ are 0.

$$l_w^{f'} = \frac{l_w^{f'-1} + l_w^{f'+1}}{2} \tag{5}$$

For wrong-detected f'' frame, let the landmark of person $w_{f''}$ in $f''-1$ and $f''+1$ frame be $l_w^{f''-1}$ and $l_w^{f''+1}$, respectively. We perform the interpolation to landmark $l_w^{f''}$ where difference $\delta_{w,l}^{f''}$ is larger than $\theta \cdot \delta_{w,l}^{f''-1}$.

$$l_w^{f''} = \frac{l_w^{f''-1} + l_w^{f''+1}}{2} \tag{6}$$

This combination of BlazePose architecture and the proposed interpolation techniques results in a model that is not only capable of providing more reliable estimations of human posture but also exhibits enhanced robustness across diverse scenarios. By successfully handling challenging scenarios and adapting to various body types, clothing variations, and environmental conditions, our model ensures consistent and accurate posture estimations.

B. ACTION DETECTION USING LSTM

Initially the interpolated landmark values are normalized, and a label map representing each of individual actions, which is a categorical data variable, is converted into numerical data by creating a new column and assigned a 1 or 0 value to the column before being fed to an RNN to improve predictions. RNNs are employed in the processing of sequential data, including speech recognition, time-series data, machine translation, etc. It recognizes the sequential characteristics of employs patterns to forecast the next likely scenarios. However, one drawback of RNNs is that, processing longer sequence of actions can be extremely time consuming. As a result, we employ LSTM, a specific kind of RNN that successfully handles this issue [32].

LSTM networks are a subset of RNNs, designed specifically for this purpose. The fundamental principle of LSTM is the cell's state, which provides an extra information flow over traditional RNN.

To begin, the forget and input gates decide which parts of the information are to be forgotten and which are to be input for the recognition of action. The forget and input gates of the person's action recognition are defined as below:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{7}$$

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{8}$$

where x_t denotes the input data; f_t and i_t denote the forget and the input gate output respectively; h_{t-1} denotes previous hidden state and σ indicates the sigmoid function.

Then, the intermediate cell state is calculated by:

$$\tilde{c}_t = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{9}$$

The cell state c_{t-1} and $\tilde{c}_t$ are then used to update the state of the cell c_t:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{10}$$

where $\cdot$ represents inner product. Now the output of o_t is derived by:

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{11}$$

The output h_t is obtained as:

$$h_t = o_t \cdot \tanh\left(c_t\right) \tag{12}$$

To ensure the classification of actions, the input video is processed in the form of (V^m, F_n, F_i), where V^m is the action video, F_n is the number of frames in the video, and F_i is the coordinate values of the 33 landmarks. Then, it is fed into first LSTM layer with 64 LSTM units, 128 units in the second layer, and 64 units in the third layer. After passing the output of the LSTM layers through two dense layers with 64 and 32 neurons, respectively for additional encoding, it is then passed on to SoftMax, which returns the probability that the input video belongs to a particular action as shown in Fig. 7. Then, the prediction with the highest probability is considered to be the class of that person's action.

C. OBJECT DETECTION USED IN ACTIONS

To accurately detect the objects used during actions, our approach involves implementing a modified end-to-end neural network. Unlike the standard YOLOv4 [33], our modified model incorporates an additional prediction head that specifically enhances the detection of small objects and effectively handles variations in object sizes. This modification allows the network to extract features using convolutional layers, enabling precise computation of bounding boxes and class probabilities for each region with a high average precision (AP). In this activity recognition model, a person may utilize different objects in different actions. Thus, the main objective of the study is to determine whether a person is using the appropriate objects when performing an activity. This is because using the wrong objects or tools for different actions may create challenges and pose potential hazards or risks, which could affect safety and efficiency in different settings. In order to address this issue, we adopt a method considering the object and the person's hand as a single entity, while disregarding any similar objects of the same class in the same frame that are not being used during the activity. This approach allows us to focus on the relevant objects and movements, and to eliminate any unnecessary or confusing information that may lead to inaccurate or misleading results. By considering the object and hand as a single entity, our

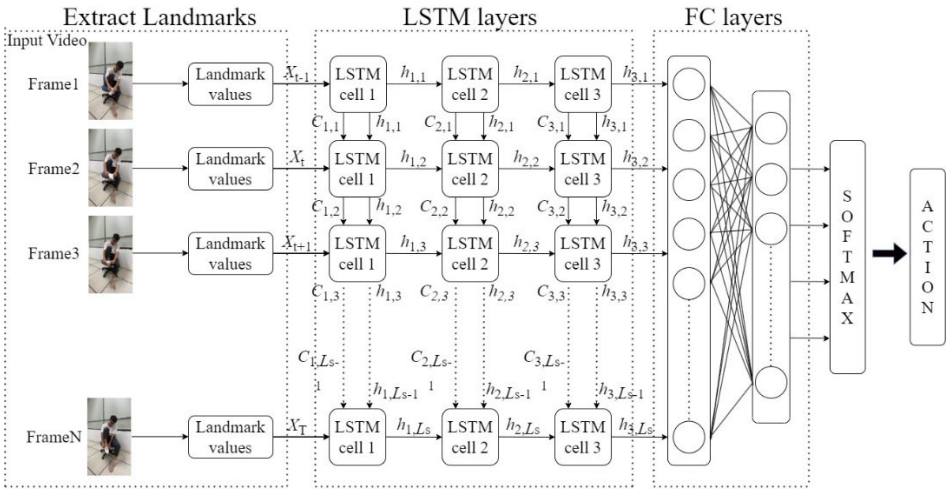

FIGURE 7. Proposed architecture of action detection model. Landmark values are the input features of the action detection network.

method accomplishes a more comprehensive understanding of the activity being performed and improve the accuracy of the analysis.

Now, the input image frame is divided into S × S grids in order to detect the object. If the object's center falls within a grid cell, it is detected using that grid cell to forecast a bounding box:

$$CS_g^b = P_{g,b} * IoU_{pred}^{truth} \tag{13}$$

where CS_g^b is confidence score of the bth bounding box in the gth grid. $P_{g,b}$ represents class probability value of the bth bounding box in the gth grid. IoU_{pred}^{truth} denotes the intersection over union (IoU) between the ground truth and predicted bounding box of the objects.

The detection model structure consists of four main parts: input terminal, backbone, neck, and head, which help to clearly describe each action flow of the suggested method. To ensure the detection of moving and stationary objects, the input image is processed at a resolution of 416 × 416 pixels. Darknet53 was created as a result of YOLOv3 [34] incorporating the residual module and the ResNet structure's properties. Based on this, YOLOv4 created the CSPDarkNet53, which consists of 5 cross-stage partial (CSP) modules and 72 convolutional layers, considering the superior learning capabilities of CSP network (CSPNet) [35]. By incorporating gradient changes into feature maps, it minimizes computational bottlenecks and enables the CNN network to achieve greater accuracy. Additionally, the initial CSP stages are transformed into the residual layer of the original DarkNet in order to increase accuracy as well as the speed. Two convolutional layers and one skip connection are included in each residual module. A batch normalization layer and a Mish activation function are included in each convolutional layer. Five CSP modules are present in the residual layers of each step of the CSPDarknet53 backbone (1-2-8-8-4). SPPNet and PANet are the components of the neck portion. The input feature layer in SPPNet is first convolved three times, and perform maximum pooling operation using different sized max pooling kernels. The pooled outputs are first concatenated, then three times convolved, which enhances the network receptive field. Following the operations of backbone and SPPNet, PANet convolves the feature layers and up-samples them, doubling the height and width of the original feature layers.

The feature layer obtained after convolution and up-sampling is concatenated with the feature layer obtained from CSPDarkNet53 to achieve feature fusion and finally down-sampling. Then, it is compressed in height and width, and stacked with previous feature layers for even more feature fusion. In contrast to three detection heads in YOLOv4, the proposed model includes an additional prediction head that enhances the ability to detect extremely small objects, improves the stability of the detection, and mitigates the negative effects of object size variance. The introduced extra head enhances the object detection algorithm by effectively handling scale variations, improving localization accuracy, providing contextual understanding, and enabling accurate classification of objects. These benefits collectively contribute to the algorithm's enhanced performance and accuracy in detecting objects of different shapes and sizes used during activities. Although this additional head incurs higher computational and memory costs, it results in better detection performance due to the utilization of low-level yet high-resolution feature maps. The model structure is shown in Fig. 8. Finally, to improve mAP and object detection, the head-anchor-based detection network model is used. The loss function used in the training phase for utilizing object detection model mainly included bounding box location loss (L_{BIoU}), confidence loss (L_{conf}) and classification loss (L_{cl}) as defined below.

$$L = L_{BIoU} + L_{conf} + L_{cl} \tag{14}$$

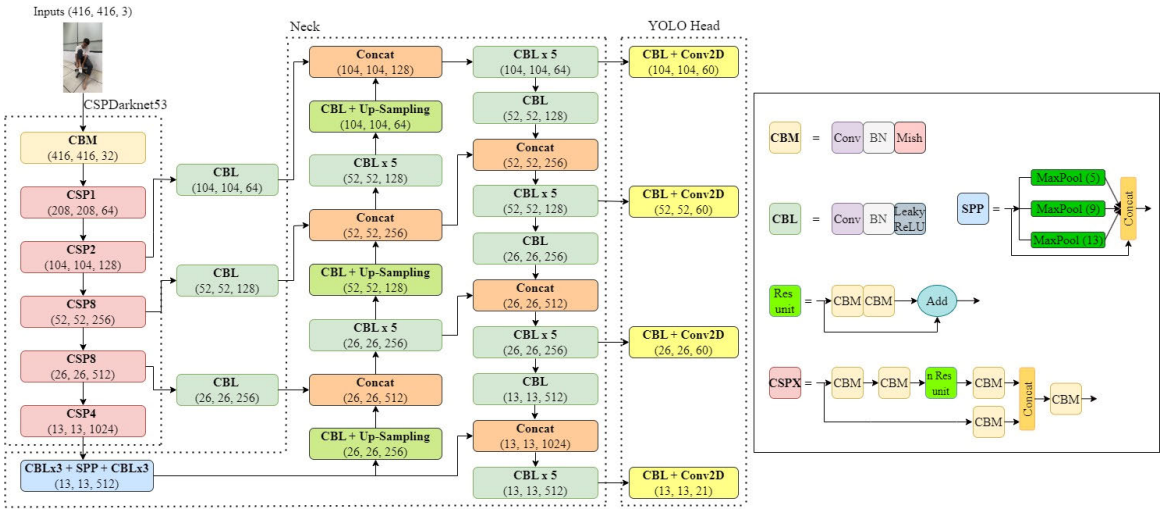

FIGURE 8. Enhanced object detection model for identifying objects used by a person during an action.

$$L_{BIoU} = 1 - IoU + \frac{d^2}{c^2} + \alpha v \tag{15}$$

$$L_{conf} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} K \left[-\log(p) + BCE\left(\hat{n}, n\right) \right] \tag{16}$$

$$L_{cl} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{noobj} [-\log(1 - p_c)] \tag{17}$$

$$BCE\left(\hat{n}, n\right) = -\hat{n}\log(n) - \left(1 - \hat{n}\right)\log(1 - n) \tag{18}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{19}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{20}$$

$$K = 1_{i,j}^{obj} \tag{21}$$

where IoU stands for the intersection over union ratio of the predicted and ground truth bounding boxes, c and d denotes the distance between the two bounding boxes' centers and their union's diagonal distance, respectively. The ground truth bounding box's width and height are denoted by w^{gt} and h^{gt}, respectively, whereas the predicted bounding box's width and height are denoted by w and h, respectively. S represents the total number of grids, while B is the anchor value for each grid. When an object is found in the jth anchor of the ith grid, the weight K has a value of 1; otherwise, it has a value of 0, while n and $\hat{n}$ denote the predicted and actual classes of the jth anchor in the ith grid, respectively, and p denotes the probability of the object.

D. ACTIVITY RECOGNITION ALGORITHM

The aim of this research is to develop an activity recognition system that can identify different actions performed within an activity, in chronological order and in accordance with predefined SOPs, while also detecting the objects used in each action. To achieve this, we must focus on both the person's body movements and the objects used during the actions, as well as the chronological order of the actions. Initially, we employ the proposed pose estimation architecture to obtain 132 landmark values that capture the person's body movements during the activity. These landmarks represent keypoints on the body and provide essential spatial information for recognizing actions. The landmark values are then fed into three layers of LSTM network which analyzes the temporal dynamics of the landmarks and learns the patterns and sequences of actions performed in chronological order. Following the LSTM layers, two fully connected layers are applied for additional encoding. These layers help extract higher-level features and representations from the temporal information captured by the LSTM network. The output is then passed through a SoftMax layer, which assigns probability values to each recognized action. The SoftMax layer enables the model to provide probability distributions, indicating the likelihood of each action being performed. Concurrently, we employ the improved YOLOv4 object detection model to identify the specific objects being used during each action. Finally, we develop an algorithm that combines the action detection and object detection models. By integrating these two components, we enable the model to accurately recognize a person's activity while considering both the predefined SOPs and the objects used. The algorithm takes into account the sequences of actions, matches them with predefined SOPs, and identifies the relevant objects being used during each action. The pseudocode outlining the activity recognition algorithm is provided in Algorithm 1, which details the steps for integrating action detection, object detection, and adherence to predefined SOPs.

IV. EXPERIMENTAL SETUP AND RESULTS

A. DATASET DESCRIPTION

The focus of our study is primarily on three tasks. The first task involves identifying a person's actions, while the second

Algorithm 1 Person's Activity Recognition

Define the expected action sequence of each activity as a list of strings.
Define action and object combination condition.
Input: Read a video V^m (F_n, F_i), where F_n represents sequence of frames, F_i represents the 132 landmark values of i^{th} frame, $n \geq 60$
Output: A person's activity with action sequence and object being utilized.

1. *Initialization: Action detection*
2. Loop over the expected actions in the sequence.
3. For each expected action, read the first 60 frames, $F_n = 60$
4. Check if the previous 10 frames are same, $F_n\,[-10:]$ is same, then
5. If $res > T$, where res is normalized output vector with probabilities of each possible outcome, threshold T=0.6, then
6. *Check condition:* Action sequence (Table 1)
7. If sequence of the action is true
8. *Initialization: Object detection*
9. if the previous 10 frames detect same object, $F_n\,[-10:]$ is same, then
10. *Check condition:* Action-object combination (Table 1)
11. If combination condition is true
12. Output: *action,* then *action* ++
13. Output: *activity*
14. Else, output an error message "wrong object detected".
15. Else, output an appropriate error message "wrong action sequence: Expected, action sequence [*i*]"
16. Close video

task involves detecting the object used during the actions. The final task is to recognize the activity based on the sequence of actions. Despite the abundance of available online datasets for data acquisition, most of them focus solely on action detection and disregard the objects utilized during the actions and the sequence of the actions in the activity. Therefore, it becomes challenging to acquire a dataset for this kind of task. In this context, this research employs the approach of using our own video and image dataset. We have gathered an extensive collection of 243 videos depicting 27 distinct actions, where each action entails the use of an object. These actions are performed in a sequence with varying objects, forming distinct activities. As elaborated in Table 1, five activities were utilized, each with a distinct chronological order of actions, and the corresponding objects used during these activities. The term 'action' here refers to the movement of the body while using an object, whereas 'activity' refers to the complete work being carried out. Given that each action is composed of a sequence of frames, we have meticulously compiled $F_n = 60$ frames for each action while developing the proposed action detection model.

To develop our model, we utilize an approach that involves focusing solely on objects being utilized by individuals during actions. We treat the person's hand and the object as one entity, disregarding any similar objects in the same class in the same frame that are not being used during the activity. The dataset particulars for the object detection model are given in Table 2.

B. EVALUATION METRICS

The performance of the proposed models was validated using a number of performance indicators, such as accuracy, precision, recall, and F1 score. These performance measurements

TABLE 1. Activities and corresponding chronological order of actions and objects used.

Activity	Action sequences	Object
Doing laundry	Add detergent into washing machine	Detergent
	Load cloths into washing machine	Cloth
	Close washing machine lid	Washing machine lid
	Press wash button	Wash button
	Open washing machine lid	Washing machine lid
	Remove cloths from washing machine	Cloth
Drying cloths	Open dyer door	Dryer door
	Load cloths into dryer	Cloth
	Close dryer door	Dryer door
	Press dry button	Dry button
	Open dryer door	Dryer door
	Remove cloths from dryer	Cloth
Printing document	Load paper into printer tray	Printer tray
	Insert USB into printer's USB port	USB
	Press print button	Print button
	Collect printed document	A4 paper
Making coffee	Open bottle	Bottle
	Tear coffee sachet	Scissors, coffee sachet
	Put coffee into bottle	Coffee sachet
	Close bottle	Bottle
	Shake bottle	Bottle
	Open bottle lid	Bottle
	Drink coffee	Bottle
Wearing shoes	Wearing sock (left and right)	Sock
	Wearing shoe (left and right)	Shoe

are calculated using four parameters such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The aforementioned performance metrics are defined as follows.

1) ACCURACY

it defines the ratio of correctly detected activities throughout the total data:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

2) PRECISION

it defines the ratio of person's activities correctly detected throughout the total videos:

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

3) RECALL

it defines the ratio of videos correctly detected as an activity to the total videos of that activity:

$$Recall = \frac{TP}{TP + FN} \tag{24}$$

4) F1 SCORE

the harmonic mean of precision and recall. The model performance is summarized by this metric effectively and is calculated as follows:

$$F1score = 2 \times \frac{precision \times recall}{precision + recall} \tag{25}$$

5) AP

the area under the precision and recall curves, denoted by Average Precision, is defined as follows:

$$AP = \int_0^1 P(r)dr \tag{26}$$

where P and r are the precision and recall, respectively. Precision and recall have values between 0 and 1. Finally, after calculating the AP values of activities, the mean average precision (mAP) is calculated as follows:

$$mAP = \frac{AP_1 + AP_2 + \ldots + AP_n}{n} \tag{27}$$

C. ACTION DETECTION RESULT

A collection of $F_n = 60$ frames, each of which contains $F_i = 132$ landmark values is obtained from each action video using our pose estimation and landmark extraction approach. Before feeding these values to the LSTM network for action detection, the entire video dataset was split into training and test datasets in an 8:2 ratio. We used the Adam optimizer [36] to train our network for 150 epochs in an effort to reduce the loss. Categorical cross-entropy loss function is used since the action detection model has twenty-seven classes. The action detection model achieved a test accuracy of 95.91% after training. Fig. 9 shows the normalized confusion matrix generated from the predictions made by the proposed action detection model on the test dataset. The results indicate that the model achieved high accuracy in recognizing the majority of the actions. However, it appears that some similar actions, such as opening or closing bottle, wearing socks or shoes, were sometimes misclassified as false positives. This is likely due to almost the identical nature of their actions.

To evaluate the quality of our model, we used OpenPose and DeepPose as the standard reference and trained two models, one with and the other without the proposed interpolation technique, using different recurrent neural networks, i.e., GRU (gated recurring units) and LSTM, as shown in Table 3. Although the OpenPose model shows slightly better performance than other estimation models, our approach with both networks performs much faster than the rest. This is due to the fact that the proposed model only employs two steps, detector and tracker inference pipeline, where the detector only runs on the first frame or until a person's face is detected, and then the tracker is used to track the person in consecutive frames. To forecast heatmaps for all landmarks, we additionally employ a compact encoder and decoder network design, followed by another encoder that regresses directly to landmark coordinates, allowing the model to become lighter and run faster in real-time inference. Also, the model trained with the interpolation technique performs better, as it used well-interpolated landmarks for undetected and wrong-detected landmark frames. Furthermore, LSTM with different pose estimation algorithms perform better because GRU has simpler structure. It has only two gates (reset and update gates) and utilize fewer training parameters.

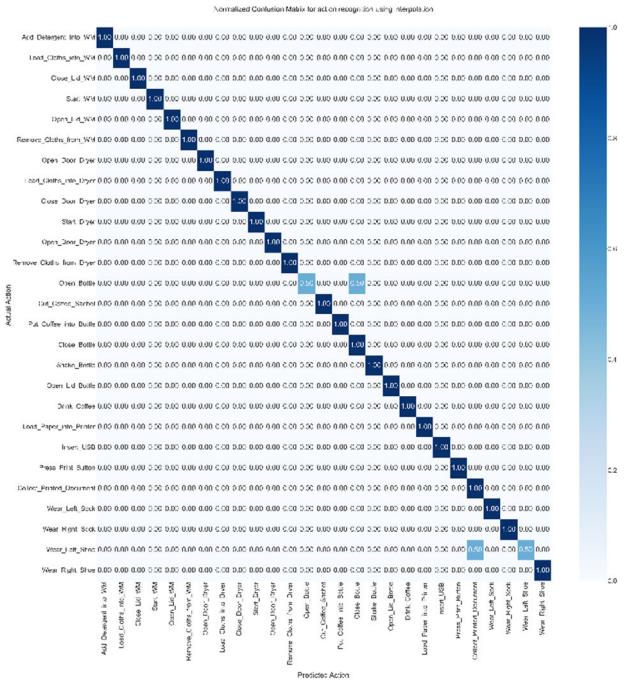

FIGURE 9. Normalized confusion matrix created using the predictions of the proposed action detection model on the test dataset.

TABLE 2. Description of the object detection dataset.

Class	Number of Images
Detergent	203
Cloth	305
Washing machine lid	294
Wash button	154
Dryer door	247
Dry button	162
Printer tray	203
USB	119
Print button	159
A4 Paper	157
Bottle	304
Scissors	276
Coffee sachet	276
Sock	250
Shoe	179
Total	3288

Consequently, GRU consumes less memory, executes faster and trains faster than LSTM's whereas LSTM achieves better accuracy on datasets with longer sequences. The output results of the proposed action detection model are shown in Fig. 10.

D. OBJECT DETECTION RESULT

Using the dataset listed in Table 2, the performance of object detection model for the suggested person activity recognition system was evaluated. Before feeding the datasets into our object detection model, we randomly divided the data into 80% for training and split the remaining data into 10% for validation and 10% for test. The shape of input images is also resized to 416 × 416 before being passed into training. After training for 500 epochs with the Adam Optimizer to

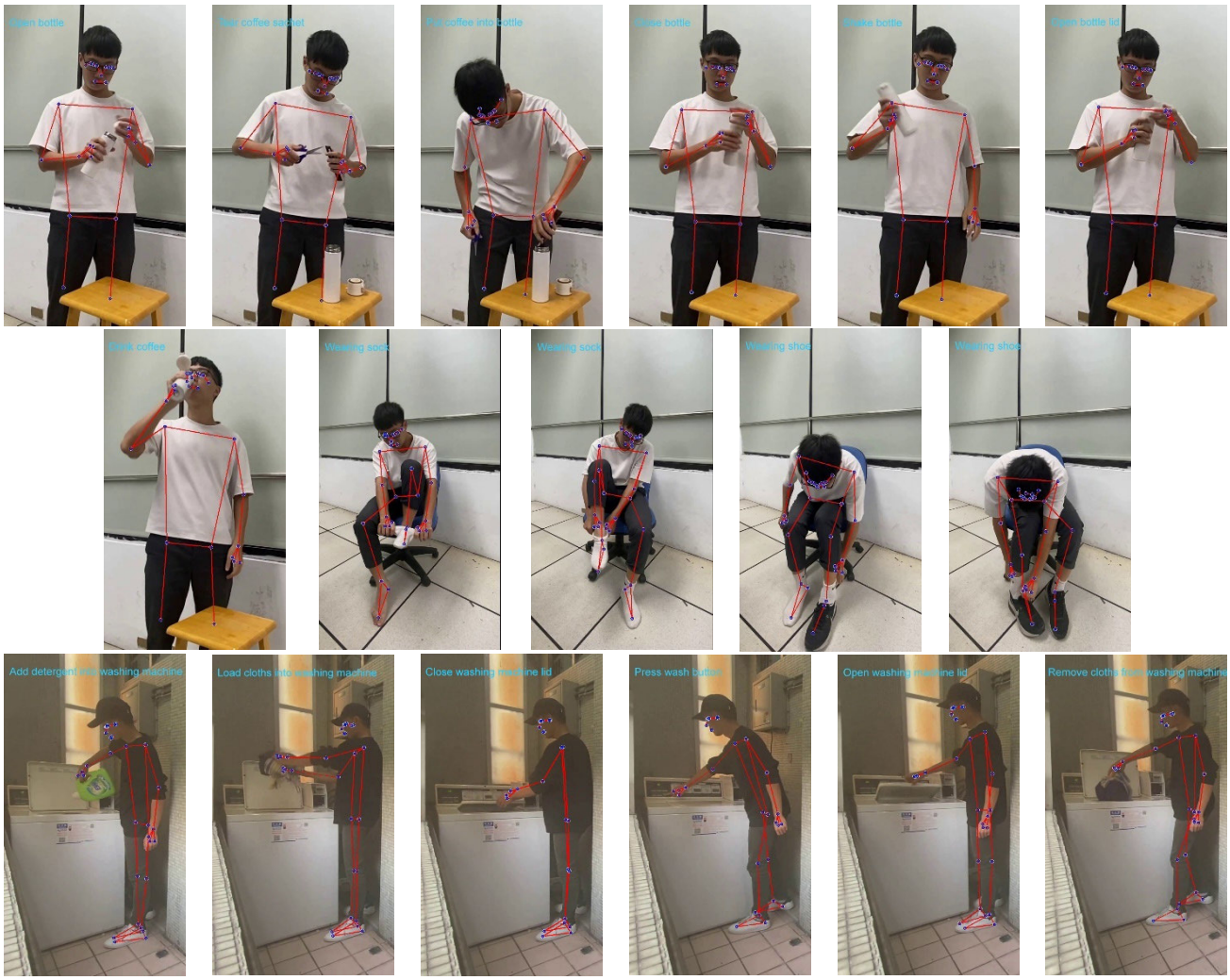

FIGURE 10. Results of action detection using LSTM network and interpolated body landmarks obtained from pose estimation network.

TABLE 3. Performance comparison of various action detection models.

Action detection model		Without interpolation		With interpolation	
		Accuracy (%)	FPS	Accuracy (%)	FPS
DeepPose [23]	GRU	85.71	2.98	89.79	1.89
	LSTM	87.75	2.74	91.83	1.23
OpenPose [24]	GRU	89.79	6.25	93.87	5.46
	LSTM	93.87	5.26	95.91	4.84
Proposed model	GRU	89.79	29.98	91.83	28.17
	LSTM	91.83	28.06	95.91	27.57

reduce the overall loss, and with the initial learning rate value of 0.0001, the proposed object detection model achieves an overall mAP of 97.68% for detecting the objects being used while performing the actions. Fig. 11 shows detection of object being utilized by the person using the enhanced YOLOv4 disregarding any similar objects in the same class in the same frame that are not being used during the activity. For example, when the person is putting on the right sock, the model does not detect the left sock. This is because we consider the person's hand and the object being used as a single entity. Similarly, when the person is loading clothes into the washing machine, the model does not detect other objects such as the washing machine lid or buttons, as they are not relevant to the action.

Performance comparison of the different object detection models is shown in Table 4. It is clear that when IoU = 0.5, Faster R-CNN has a higher mAP but with the lowest FPS than others. It signifies that the common features of two-stage detection algorithm have higher detection accuracy but lower real-time problems. Meanwhile, FPS and mAP of our model are reasonably high when compared to other algorithms. Although our model is a little slower than the original YOLOv4 due to the extra computational load from the additional head, it delivers superior object detection performance for every frame in the video. This is due to the advantage of having an extra head that allows the model to detect objects of varying sizes with better accuracy.

FIGURE 11. Object detection results using enhanced YOLOv4 algorithm to identify objects used during actions.

TABLE 4. Performance comparison with other object detection models.

Object detection model	FPS	mAP (%)
Faster R-CNN	2.44	98.50
SSD	8.06	95.52
YOLOv3	9.15	96.17
YOLOv4	10.84	96.84
Proposed model	10.59	97.68

TABLE 5. Comparisons on various activity recognition models.

Activity recognition model	FPS	mAP (%)
Proposed action detection model + Faster R-CNN	2.27	98.50
Proposed action detection model + SSD	7.86	95.52
Proposed action detection model + YOLOv3	8.39	96.17
Proposed action detection model + YOLOv4	10.65	96.84
Proposed action detection model + enhanced YOLOv4	10.47	97.68

Considering both mAP and FPS metrics, the proposed method is the most suitable for detecting objects used during activity.

E. RECOGNITION OF THE PERSON'S ACTIVITY

To achieve real-time predictions for an activity recognition model we employ the proposed recognition algorithm outlined in section III-D to analyze the person's activity output. We begin by looping through the frames with OpenCV and appending them. Once we have accumulated a set of 60 frames ($F_n = 60$), we feed them into the proposed action detection model. This model checks for the action sequence and also detects the object being used during the action by examining the action and object combination (Table 1). The results from the activity recognition model are depicted in Fig. 12.

Table 5 presents the outcomes of using the proposed action detection model with different state-of-the-art object detection models for activity recognition. The results reveal that the Faster R-CNN object detection model combined with the proposed action detection model has a high mAP, but a lower FPS compared to other models, making it unsuitable for real-time activity prediction. However, the primary goal

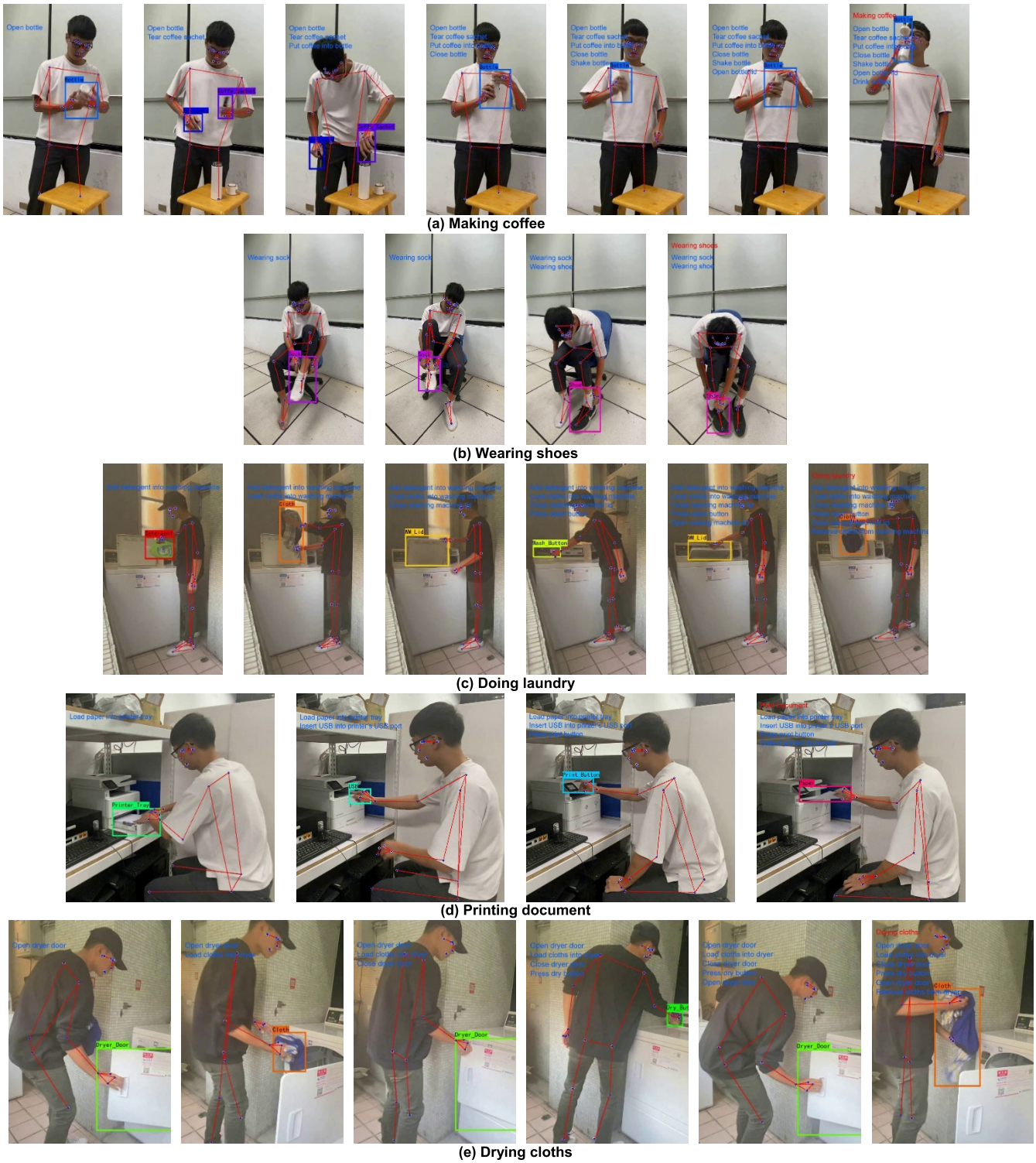

FIGURE 12. The output of the proposed activity recognition model. The model identifies different actions that are performed in a chronological order and the objects utilized during each action.

of this research is to recognize person's activities by detecting action sequences and interactive objects in real-time. Thus, we require a model that can quickly identify person's actions and detect objects. According to the experimental findings, the enhanced YOLOv4 model combined with the proposed action detection model achieves a higher FPS and a reasonably high mAP, suggesting that this model is more suitable for recognition problems.

Furthermore, it is worth noting that running the action detection and object detection models independently allows them to maximize their processing capabilities. Conversely, when these two models are integrated, there is an additional coordination overhead, resulting in a slight decrease in frames per second (fps) compared to individual execution. Nonetheless, the integration offers the advantage of precise activity recognition by incorporating both actions and objects, thereby enabling a more profound comprehension of the activity at hand.

V. CONCLUSION

The proposed model incorporated a lightweight CNN optimized top-down human pose estimation architecture to find the body landmarks from a sequence of frames, followed by interpolation to enhance the accuracy of pose estimation for undetected or wrong-detected landmarks. The transformed landmark values were then fed to multiple layers of LSTM network, culminating in the SoftMax layer to predict the person's actions. Additionally, an object detection model was developed by enhancing YOLOv4 to detect the object used during the actions. Finally, the proposed activity recognition algorithm integrated these two models to create a real-time, lightweight, and robust activity recognition model. Our model achieved 95.91% accuracy in recognizing actions and 97.68% mAP for detecting the object used during the actions, with an overall FPS of 10.47. This model can help monitor and inspect human activities that followed a chronological order of actions when interacting with different objects within the activity. In manufacturing and assembly, our activity recognition model can be utilized to ensure workers following predefined sequences when using tools and components, boosting efficiency and quality control. In sports analysis, it can accurately track players' movements, recognize techniques and equipment used, and provide valuable insights for coaching and strategic analysis. In healthcare and rehabilitation, it can assist in monitoring patients' activities during therapy and offer real-time feedback to improve outcomes. In industrial environments, it can analyze workers' actions and equipment interactions to ensure safety compliance.

In the future, we plan to enhance the proposed method to recognize activity in industrial working environments and detect additional objects such as helmets, gloves, masks, and shoes to ensure individual safety and prevent industrial accidents. Additionally, we aim to enhance the fps of our model without compromising accuracy by exploring model optimization techniques, leveraging hardware acceleration, considering algorithmic improvements, and upgrading hardware infrastructure.

REFERENCES

[1] Q. Wu, Y. Wu, Y. Zhang, and L. Zhang, "A local–global estimator based on large kernel CNN and transformer for human pose estimation and running pose measurement," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.
[2] F. Rustam, A. A. Reshi, I. Ashraf, A. Mehmood, S. Ullah, D. M. Khan, and G. S. Choi, "Sensor-based human activity recognition using deep stacked multilayered perceptron model," *IEEE Access*, vol. 8, pp. 218898–218910, 2020.
[3] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019.
[4] T. Zebin, P. J. Scully, N. Peek, A. J. Casson, and K. B. Ozanyan, "Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition," *IEEE Access*, vol. 7, pp. 133509–133520, 2019.
[5] Y. Li, C. Wang, Y. Cao, B. Liu, J. Tan, and Y. Luo, "Human pose estimation based in-home lower body rehabilitation system," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Glasgow, U.K., Jul. 2020, pp. 1–8.
[6] W. Liu, X. Liu, Y. Hu, J. Shi, X. Chen, J. Zhao, S. Wang, and Q. Hu, "Fall detection for shipboard seafarers based on optimized BlazePose and LSTM," *Sensors*, vol. 22, no. 14, pp. 5449–5466, Jul. 2022.
[7] M. Abbas and R. L. B. Jeannès, "Exploiting local temporal characteristics via multinomial decomposition algorithm for real-time activity recognition," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
[8] W. Huang, L. Zhang, W. Gao, F. Min, and J. He, "Shallow convolutional neural networks for human activity recognition using wearable sensors," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
[9] Y. Zhang, G. Tian, S. Zhang, and C. Li, "A knowledge-based approach for multiagent collaboration in smart home: From activity recognition to guidance service," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 317–329, Feb. 2020.
[10] N. A. Capela, E. D. Lemaire, and N. Baddour, "Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients," *PLoS ONE*, vol. 10, no. 4, pp. 1–18, Apr. 2015.
[11] A. Prati, C. Shan, and K. I.-K. Wang, "Sensors, vision and networks: From video surveillance to activity recognition and health monitoring," *J. Ambient Intell. Smart Environ.*, vol. 11, no. 1, pp. 5–22, Jan. 2019.
[12] S. Sankar, P. Srinivasan, and R. Saravanakumar, "Internet of Things based ambient assisted living for elderly people health monitoring," *Res. J. Pharmacy Technol.*, vol. 11, no. 9, pp. 3900–3904, Dec. 2018.
[13] E. Zdravevski, P. Lameski, V. Trajkovik, A. Kulakov, I. Chorbev, R. Goleva, N. Pombo, and N. Garcia, "Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering," *IEEE Access*, vol. 5, pp. 5262–5280, 2017.
[14] X. Ji, J. Cheng, W. Feng, and D. Tao, "Skeleton embedded motion body partition for human action recognition using depth sequences," *Signal Process.*, vol. 143, pp. 56–68, Feb. 2018.
[15] A. Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recognit.*, vol. 61, pp. 295–308, Jan. 2017.
[16] C. Xu, L. N. Govindarajan, and L. Cheng, "Hand action detection from ego-centric depth sequences with error-correcting Hough transform," *Pattern Recognit.*, vol. 72, pp. 494–503, Dec. 2017.
[17] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Comput. Appl.*, vol. 28, no. 12, pp. 3941–3951, Apr. 2016.
[18] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 430–439, Oct. 2016.
[19] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, "A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1384–1393, Apr. 2019.
[20] C. Aviles-Cruz, E. Rodriguez-Martinez, J. Villegas-Cortez, and A. Ferreyra-Ramirez, "Granger-causality: An efficient single user movement recognition using a smartphone accelerometer sensor," *Pattern Recognit. Lett.*, vol. 125, pp. 576–583, Jul. 2019.
[21] I. Jegham, A. B. Khalifa, I. Alouani, and M. A. Mahjoub, "Vision-based human action recognition: An overview and real world challenges," *Forensic Sci. Int., Digit. Invest.*, vol. 32, Mar. 2020, Art. no. 200901.
[22] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device real-time body pose tracking," 2020, *arXiv:2006.10204*.
[23] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1653–1660.
[24] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[25] W. Li, L. Wen, M. Chang, S. N. Lim, and S. Lyu, "Adaptive RNN tree for large-scale human action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1453–1461.
[26] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1110–1118.
[27] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 12–17.
[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.
[29] W. Liu, Z. Liu, Y. Li, H. Wang, C. Yang, D. Wang, and D. Zhai, "An automatic loose defect detection method for catenary bracing wire components using deep convolutional neural networks and image processing," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2021.
[30] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "DeepCut: Joint subset partition and labeling for multi person pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4929–4937.
[31] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon, "The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 103–110.
[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[33] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[34] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[35] C. Wang, H. Mark Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, Jun. 2020, pp. 1571–1580.
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

YO-PING HUANG (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, TX, USA.

He was a Professor and the Dean of Research and Development, the Dean of the College of Electrical Engineering and Computer Science, and the Department Chair with Tatung University, Taipei. He is currently the President of the National Penghu University of Science and Technology, Penghu, Taiwan. He is also a Chair Professor with the Department of Electrical Engineering, National Taipei University of Technology, Taipei, where he was the Secretary General. His current research interests include fuzzy system design and modeling, deep learning modeling, intelligent control, medical data mining, and rehabilitation systems design.

Dr. Huang is a fellow of IET, CACS, TFSA, and the International Association of Grey System and Uncertain Analysis. He was a recipient of the 2021 Outstanding Research Award from the Ministry of Science and Technology, Taiwan. He serves as the IEEE SMCS VP for Conferences and Meetings and the Chair of the IEEE SMCS Technical Committee on Intelligent Transportation Systems. He was the IEEE SMCS BoG, the President of the Taiwan Association of Systems Science and Engineering, the Chair of the IEEE SMCS Taipei Chapter and the IEEE CIS Taipei Chapter, and the CEO of the Joint Commission of Technological and Vocational College Admission Committee, Taiwan.

SATCHIDANAND KSHETRIMAYUM received the B.Tech. degree in computer science and engineering from the National Institute of Technology Manipur, India, and the M.Tech. degree in operations research from the National Institute of Technology Durgapur, India. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan. His current research interests include human activity recognition (HAR), computer vision, deep learning, and image processing.

CHUN-TING CHIANG received the bachelor's degree from the Department of Electrical Engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan. He is currently pursuing the master's degree in electrical engineering with the National Taipei University of Technology, Taipei, Taiwan. His current research interests include machine learning, deep learning, and image processing.