IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 17, 2021, accepted December 14, 2021, date of publication December 27, 2021,
date of current version January 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3138778

Proposing Posture Recognition System Combining MobilenetV2 and LSTM for Medical Surveillance

PHAT NGUYEN HUU, NGOC NGUYEN THI, AND THIEN PHAM NGOC
School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi 10000, Vietnam

Corresponding author: Phat Nguyen Huu (phat.nguyenhuu@hust.edu.vn)

This work was supported by the Ministry of Education and Training (MOET), Vietnam, under Grant B2020-BKA-06.

ABSTRACT This paper proposes a posture recognition system that can be applied for medical surveillance. The proposed method estimates human posture using mobilenetV2 and long short-term memory (LSTM) to extract the important features of an image. The output of the system was a fully estimated skeleton. We used seven human indoor postures, including lying, sitting, crouching, standing, walking, fighting, and falling, and classified them. The output results are the extraction of the human skeleton and the corresponding labels for the poses. We first experiment with classification using machine learning. The system only achieves approximately 88% accuracy because it is not able to classify similar postures, such as standing and walking. This difference can be caused by the extraction of features for static images, and the machine learning classification algorithm has not reached accuracy with training data. Therefore, we proposed the integration of the LSTM model into the proposed system. LSTM learns the features of the skeleton and provides classification results for postures. As a result, our system improved the accuracy by up to 99%. Similar postures, such as standing and walking, have improved accuracy by up to 7%. In addition, we performed the system on the Jetson Nano hardware. The results show that it can run on a low-profile (44% CPU and 2.1 frames per second) that is capable of applications for remote patient monitoring devices.

INDEX TERMS Openpose, long short-term memory, posture detection, skeleton model, intelligent healthcare.

I. INTRODUCTION

When science and new techniques are developed, machines can replace human activities. The control of machines has become increasingly diverse, which presents many challenges for human-computer interactions (HCI). This development has made artificial intelligence (AI).

AI is inspired by humans with emotional and cognitive elements. It personifies the characteristics of all kinds of energies (perception and emotions) that are capable of self-awareness in interactions. Humans can interact with machines using voice or action gestures by interacting directly with each other. Gestures and postures are widely studied because they are communication methods between humans and computers. The problems of gesture and action recognition have become popular. Several developing countries have applied to real environments, such as machine control. However, identifying posture still has many challenges, such as accuracy and time processing [1]. Therefore, we focus on recognizing the posture that is advanced in gesture and action recognition.

A human posture recognition system will bring great benefits if it is applied to a real environment. The applications include identifying theft and criminal threats, monitoring elderly and young children for warning of medical dangers, and detecting athletes against rules of sports. In addition, gesture and action recognition studies have been conducted. However, the accuracy of the model is not high, and the hardware cost is expensive. The authors [2] proposed wearable biomedical sensors to recognize human activities. They used three classifiers (kNN, linear, and Gaussian kernels) for the training and testing phases. The accuracy of the model improved up to 99%. However, the disadvantage of this system is that it cannot classify activities similar to standing,

The associate editor coordinating the review of this manuscript and approving it for publication was Varuna De Silva.

brushing, sleeping, and laying. In addition, the authors [3] proposed HAR using a smartphone with many sensors. They also used six activities (walking, climbing up the stairs, climbing down the stairs, sitting, standing, and laying) and performed nine models. The results show that the support vector machine (SVM) model had the highest accuracy of 99.4%. The authors [3] also proposed human activity recognition (HAR) using long short-term memory (LSTM) for healing surveillance. The model consists of seven activities: walking, jogging, upstairs, downstairs, sitting, standing, and overall. However, the accuracy of the model is not good for fast activities (upstairs and downstairs) of 50% and 67%, respectively. In addition, these systems only focus on activities without other bodies. The authors [4] designed a warning system based on a smartphone (FallDroid) to detect fall-like events (lying on a bed or suddenly stopped after running). The system uses a threshold combining an SVM with 97% accuracy. However, its disadvantage is the use of devices to mount on humans. The authors [5] predicted the falling activity based on the human skeleton map. The system can change the learning rate to improve the accuracy by up to 91.7%. However, the predicted time for each gesture was more than 45 seconds, which is not suitable for real applications. The authors [6] proposed smartphone sensors for detecting health surveillance activities. They perform the system with six activities, namely sitting, walking, laying, standing, walking downstairs, and walking upstairs. The results show that the system improves accuracy by up to 95%. However, its disadvantage is that using multiple sensors can create serious challenges due to battery limitations, processing time, etc. In addition, it is difficult to see the entire gesture if only sensors are used.

To solve these problems, our system is considered to be an advanced problem in gesture recognition and action. The essence of a system recognizes the input as a still image or part of the human body. The application requires video (sequential series of images) and the entire body. As a result, the system meets the input requirements and provides high accuracy. It helps detect simple and complex postures that issue warnings for health, sports, and criminal affairs. We are largely concerned with postures in applications that can monitor the behavior of endangerment for medical surveillance.

Our system has three main points as follows:

- First, we do not use wearable devices or smartphones, similar to current methods relating to healthcare. Because the use of wearable devices is not suitable for people with skin diseases and the elderly with cognitive problems [7], we use an appropriate device camera for different situations.
- Second, our training data requirement is human with all parts to be able to extract the skeleton. Extracting data requires considerable memory that our system needs to pay attention to performance. Therefore, we used an algorithm with a mobilenetv2 backbone network to extract a skeleton that is a faster and lighter model using the native backbone of Openpose (VGG19 network) [6]. The result is to improve the accuracy by up to 99%, as shown in Section IV.
- Third, the training data is the sequence of frames cutting from videos containing the whole human body. Therefore, we put the receiving data into the LSTM network to provide features after extracting the human skeleton, which makes classification more efficient than other traditional methods. The results are presented in Sections III and IV, respectively.

The remainder of this paper is organized as follows. In Section II, we present the related work. In Sections III and IV, we present and evaluate the effectiveness of the proposed model. Finally, conclusions are presented in Section V.

II. RELATED WORK

Today, the quality of life has increased, which helps create a potential for AI technology. Therefore, classification systems are important.

Object recognition has been of interest in many studies. It has many useful applications in computer vision, such as image recognition, tracking, and searching objects. Therefore, posture recognition systems are suitable for smart applications, such as smart homes and medical surveillance.

Typical applications relating to posture recognition problems are:

- Security and surveillance systems include networks of cameras that are monitored by humans.
- Human-machine interactions are still challenging. Visual signs are the most important means of nonverbal communication. Effective exploitation of communication methods through gestures promises to generate computers that interact more accurately and naturally with humans. The authors [8] used RGB-D cameras for human posture recognition. The accuracy of the system was 93.3% and 95.7% for color and deep images, respectively. However, the disadvantage of the method is that the camera only works well for a small environment (indoors) and the distance for detecting humans is from 0.4 to 3 meters [9].
- Intelligent healthcare performs fall and stroke detection. The authors [10] used an IMU sensor for a falling detection module and designed a protective vest. The authors rely on the threshold of vertical velocity and position displacement to detect fall events during transitions of rapid motion. The system achieves 93.6% sensitivity and 95.6% specificity for the drop detection system and airbag, respectively.
- In addition, posture recognition of video can be used to summarize, query video, and analyze sports.

The methods of using an RGB-D Kinect camera or sensor achieve high accuracy but consume large hardware. Currently, many studies have only applied image processing to identify without high configuration. Several algorithms are commonly applied to estimate human behavior, such as SimplePose [11], AlphaPose [12], OpenPose [6], which use large datasets such as COCO and MPII. These algorithms have been studied and applied to extract skeletons from finding points on the body. It can be estimated for human posture and can be performed for real applications.

Therefore, a system using Openpose is applied in many applications. A simple method for detecting human behavior based on background subtraction is proposed, such as the CodeBook algorithm [13], foreground segmentation to find moving objects, and classification by KNN [14], or eigenstate technique to recognize poses [15]. They rely on detecting bounding boxes of moving objects and then extracting their features for classification. Several typical methods such as YOLO for video motion [16] and mobilenetV2 combining SSD [17] achieve over 90% accuracy while running for weak configuration as a smartphone.

The authors [18] proposed a recognition system using time-sequential information with a body sensor based on deep neural structured learning (NSL). The results show that the accuracy of the system is improved by up to 99%. The authors [19] proposed a HAR model using four sensors for 16 objects. They focused on six different ML algorithms on RGB-D sensor modalities: decision trees, linear discriminant, cubic SVM, fine k-NN, bagged tree, and DNN. As a result, the accuracy of the model is up to 96.8% for classification with both static and dynamic activities. The advantage of this system is that it is lightweight and suitable for real-time applications. The authors [17] also proposed a HAR model with enhanced channel state information. The accuracy of the model was improved by up to 97%. However, the processing time is up to 1.4 hours for 400 units, which is not suitable for real applications.

To improve the speed and accuracy of posture detection systems, several studies have proposed methods based on finding out skeleton key points such as DeepPose [20] and MoDeep [21]. The system learns the features of a skeleton to recognize human postures and behaviors. Keypoints are first found using the top-down method. The system [12] will detect objects and extract skeletons. It then estimates the poses. As a result, this method detects redundancies because many poses are detected for a person that causes many errors in bounding boxes. It has a running time corresponding to the number of people appearing in frames. Therefore, it would be effective if there were only a few people. The Openpose down-top method [6] separates the system runtime from the number of people appearing in frames. This allows the system to run in real-time while extracting multiple skeletons. For this reason, we decided to choose Openpose to estimate our poses.

We integrate LSTM into a system to extract features and classify the labels after using Openpose. It will improve the difference between static and dynamic actions with confusing poses, such as standing and walking, by using LSTM for the proposed system.

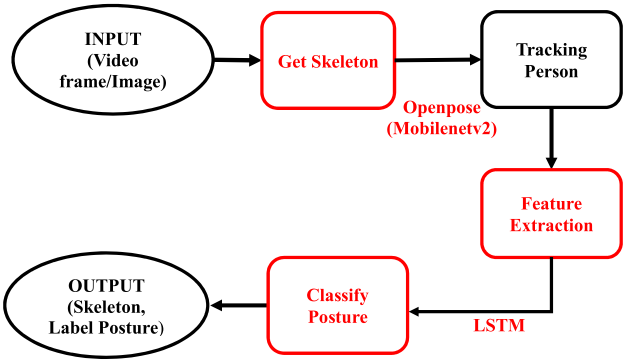

FIGURE 1. Overview of the proposal system model.

III. PROPOSAL SYSTEM

A. OVERVIEW OF SYSTEM

Figure 1 shows a block diagram of the proposed system. The system operates as follows.

The system will work as follows:

- Input is a series of frames cutting from videos containing the body of one or two people for the best training results.
- Obtain and extract the human skeleton using the Openpose algorithm.
- The tracking person block follows people to reveal skeleton features.
- Feature extraction uses LSTM to learn the features of skeletons and extract them.
- Classification posture categorizes each pattern based on a machine learning algorithm.
- Output is a human skeleton with a posture label that performs input videos to achieve the best results.

As analyzed above, we noticed confusion between the static and dynamic postures of frames. Therefore, the proposed system integrates the LSTM block to process the entire data that solves the above problem based on [6], [22]. The goal of the system is to recognize simple and common postures. We then upgraded with more specific postures or behaviors depending on the real system. We consider indoor posture and perform identification to issue warnings about health problems in our study. The data (lying, sitting, standing, bending, walking, fighting, and falling postures) are used in the paper that are preprocessed by ourselves. They were then cut into frames (5 frames/s). They were then passed through the Openpose model to extract the skeleton. Output data are the positions of the skeleton points classified by LSTM.

B. SYSTEM DETAILS

1) DATA COLLECTION

In addition to collecting data from the human pose dataset (MPII) [23], we also built and pre-processed our data according to the proposed system. The total amount of data

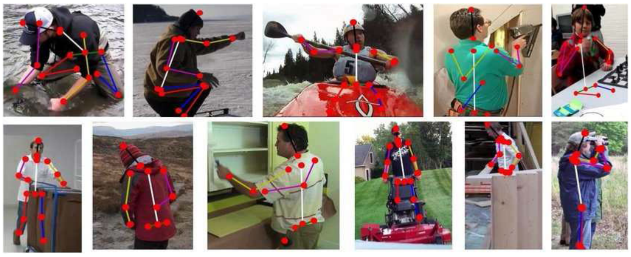

FIGURE 2. An example of extracting skeletons from MPII datasets [24].

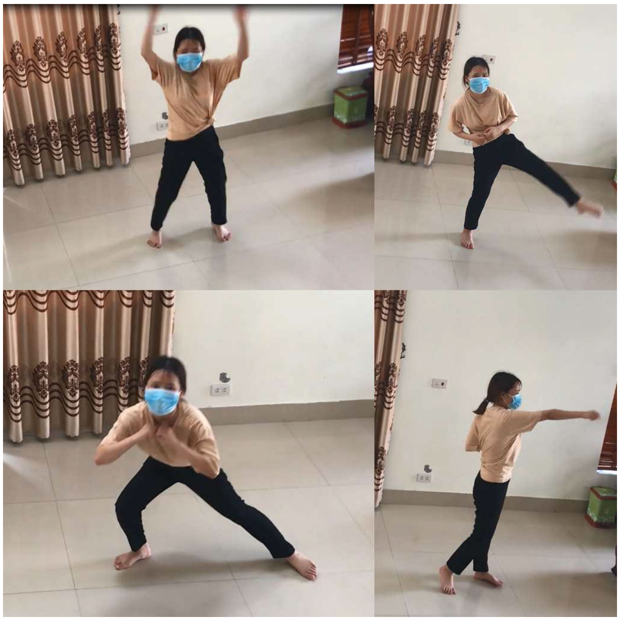

FIGURE 3. An example of our datasets.

was 18,900. The selection postures included lying, crouching, sitting, standing, walking, punching, and falling. Punching and falling are two gestures that we are most interested in. The details of using the MPII dataset are presented in Tab. 1.

MPII is a dataset used to train models related to human action recognition. The authors [24] used the MPII to extract human skeletons. Because the input data returned by skeletons are insufficient, we have to filter 25,000 data to obtain 5000 for postures such as bending, sitting, standing, and walking. In addition, there are several poses for medical monitoring in the dataset. Therefore, we created the dataset. Figures 2 and 3 depict the skeleton data using [24] for their system and our dataset, respectively.

We find that the transitions of postures for each other analyze human behavior more clearly. Therefore, we will collect data using standing-lying and walking-lying transitions for falling action.

The number of frames and labels are listed in Tab. 1. Our datasets and labels will have lower quality than the standard MPII datasets owing to video recording equipment, lighting, and environment. However, the advantage is that we can

FIGURE 4. An example of training dataset [24].

customize the postures that are used to train and include the entire human body. Figure 4 shows the dataset for the training model. The data were created in an indoor environment, and there were only about one or two people per frame.

2) PRE-PROCESSING DATA

In addition to using MPII datasets, several postures must be rotated and cut into frames according to the requirements. In this study, we used five frames per second (FPS). Our data were labeled and categorized by free video to the JPG converter tool.

3) DETAILS OF IDENTIFICATION PROCESS

Figure 5 outlines more details on how the system detects an object and extracts points from frames to create the corresponding joints. The results are the skeletons that are put into the LSTM network to extract important features and classify them with corresponding labels. Our system is divided into two main blocks. In extracting and tracking skeleton blocks, the input data can be an image or frame cutting from a video. The Openpose algorithm is used to extract features of human skeletons. In the feature extraction and posture classification block, the input data is the output of the previous block, which includes the extracting skeleton values. The LSTM network is responsible for learning the features of the skeleton to return the corresponding classification labels.

Figure 6 shows the architecture of the Openpose algorithm [25]. When frames are put into the system, Openpose with the VGG-19 backbone extracts highlights from the image and detects the person. The first stage is used as a training model to recognize the skeleton points. 38 part affinity fields (PAF) then show the joint degree and continue prediction.

As mentioned above, our system uses the Openpose backbone (mobilenetV2 network) to increase the processing speed [26]. Figure 7 depicts the classes of the mobilenetV2 network.

TABLE 1. Number of frames are used for training and testing with seven postures.

Label	MPII/total dataset	Train dataset	Validation dataset
Lying	0/1278	990	288
Sitting	1320/3687	2949	738
Crouching	940/2129	1703	426
Standing	1580/3163	3050	563
Walking	1250/2963	2370	593
Punching	0/3168	2534	634
Falling (standing to lying & walking to lying)	0/1870	1590	280

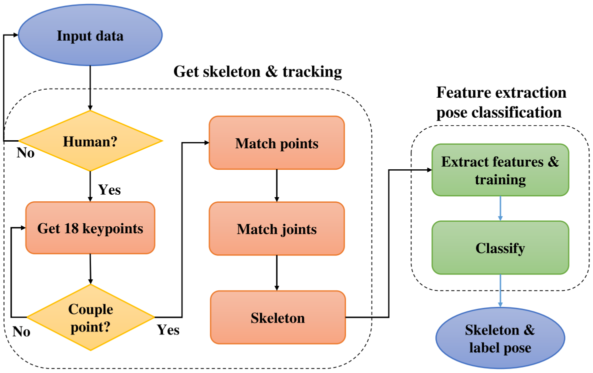

FIGURE 5. Operation diagram of proposed system.

Mobilenet networks use depth-separating convolutional layers. The first layer is a 1 × 1 convolution aggregator that expands the number of channels before entering the deep convolution. ReLU6 offers an expansion factor. This function breaks down the linearity of neurons and is powerful when used with low-precision calculations [27]. Batch normalization is a mass standardization class that minimizes overfitting [28]. The size of the convolution layer that helps to filter the final input is 1 × 1. This helps to reduce the number of channels. This layer is also known as the bottleneck layer. This helps to reduce data flowing through the network.

To evaluate the capabilities of mobilenetV2 networks, we performed a comparison among the methods, as shown in Tab. 2. In Tab. 2, we can see that the accuracy of mobilenetV2 is the best compared with other methods (VGG16 and VGG19), while the number of parameters is the smallest. The number of parameters is only about 3.5 million that is 1/41 of VGG19.

We can see that the accuracy of mobilenetV2 is not inferior to other deep learning models such as VGG16, VGG19, or ResNet50 based on Tab. 2. When the number of its parameters is only approximately 3.5 million, the size of mobilenetV2 is only 1/7 of the ResNet50 network and its accuracy is less than 2%. Therefore, we chose the mobilenetV2 network to be able to run on low hardware while still ensuring accuracy and real-time requirements in the article. This factor helps to achieve high accuracy and low computation time. We divide the convolutional operation that significantly reduces the computation volume and number of network parameters in the network. This makes it work well with low hardware requirements.

TABLE 2. Evaluating parameters of CNN based on imagenet dataset from Keras.

Network	Size of network	Accuracy	Parameter
VGG16	528MB	0.9001	138,357,544
VGG19	549MB	0.900	143,667,240
ResNet50	99MB	0.921	25,636,712
Mobilenet	16MB	0.895	4,253,864
MobilenetV2	**14MB**	**0.901**	**3,538,984**

4) EXTRACTING SKELETON BY OPENPOSE

Openpose was developed by a perceptual registering research department [25]. It tracks movement and recognizes humans using an RGB camera. Its operations are as follows:

- It uses a CNN based on the detection of saving points on the body, face, and hands. These points are reconnected into the body based on the movement frame.
- Points stored in Open Pose.
- 18 points on the human body are identified using Openpose, as shown in Fig. 8. These points represent joints of the body and can orient the detecting face of a person [29].
- The hand consists of 21 points that are stored by the Openpose and used to recognize hand movements. These points represent the joints of the hands.
- The face consists of 68 points that are stored in Openpose and used to receive facial movements.

Raw skeleton data were processed before extracting their features. The pretreatment process consisted of four steps, as summarized in Fig. 9. More details can be found in [29]:

- **Step 1 (Scaling coordinates):** The initial matching position performed by OpenPose has a unit that is different from the x and y coordinates. Therefore, we scaled them into the same units to process. Figure 9 depicts the position matching of the coordinates (x, y).
- **Step 2:** OpenPose gives five points including one position on the head, two positions in the eye, and two positions in the ear. However, the head position only helps with sorting actions for the training set. Therefore,

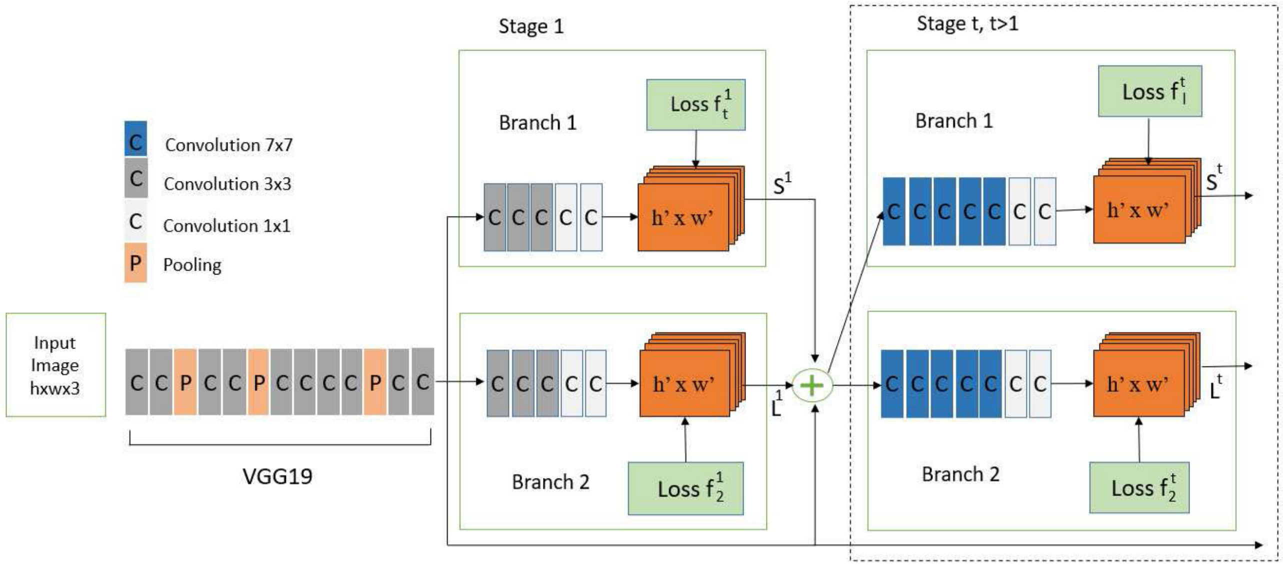

FIGURE 6. Description of Openpose architecture [25].

we removed the five matches on the head to make the algorithm faster.

- **Step 3:** We are going to get rid of those without necks or thighs. If there is no human skeleton detected by OpenPose or the skeleton is found to have no neck or thigh, the frame will be considered invalid and discarded. In addition, the sliding window re-initializes on the next frame [25], [29].
- **Step 4:** In several cases, OpenPose may fail to detect the human skeleton from an image and cause several gaps at the joint positions. These joints must be filled in with several values to maintain fixed-sized performance vectors for the following feature classification. If a point (x_i, y_i) is not found in the current frame, Openpose will rely on the relationship of position with previous frames to set its value. The knee position of the point of the lost frame is calculated as follows:

$$\begin{aligned} x_{i\_curr} &= x_{Leg\_curr} + \left(x_{i\_prev} - x_{Leg\_prev}\right), \\ y_{i\_curr} &= y_{Leg\_curr} + \left(y_{i\_prev} - y_{Leg\_prev}\right). \end{aligned} \tag{1}$$

Finally, we make 18-point connections with the joints and skeleton. The technique is a part association field (PAF) for the relationship among joints. The authors [6] predicted the location of key points and 2D vector fields to determine the relationships among skeleton points. The PAF outputs are 2D key points for all the frames.

5) EXTRACTING AND CLASSIFYING CHARACTERISTIC BY LSTM

Openpose extracts a matrix containing 18 positions of a skeleton with values between 0 and 1 and labels files corresponding to posture.

A recurrent neural network (RNN) is a deep learning model that can process information in a sequence (sequence/time series). In the problem of predicting input action, RNN can carry frame information from the previous to the next states. The final state is a combination of all the images that predict the action of a video.

LSTM is a special form of RNN that is capable of learning distant dependencies. It is designed to avoid long-term dependency problems. A Memorandum of information over a long time is their default feature. We do not have to remember this. The patient was able to memorize without any intervention.

All regression networks (RNNs) are a series of repeating modules. These modules have a very simple structure and often have a tanh layer with a standard RNN network.

LSTM has a string architecture. However, the modules have a structure different from that of the standard RNN network. They have four layers instead of one that interact with each other, as shown in Fig. 10.

The features of the skeleton will take on x and the following states are as follows [30], [31], where

Input (c_{t-1}, h_{t-1}, x_t)**:**

x_t is the input of the t^{th} state of the model. c_{t-1} and h_{t-1} are the outputs of the previous layer. h is the result of the tanh function.

Output (c_t, h_t)**:**

c cell state and h are hidden state. f_t, i_t and 0_t are the forget, input, and output gates, respectively.

The new feature of LSTM compared to RNN is the conversion line from c_{t-1} to c_t. This helps the important information to be sent first and used. Therefore, LSTM can carry information remotely (long-term memory) and combine it with the specials inherited from RNN (short memory). The LSTM feature was suitable for our input data.

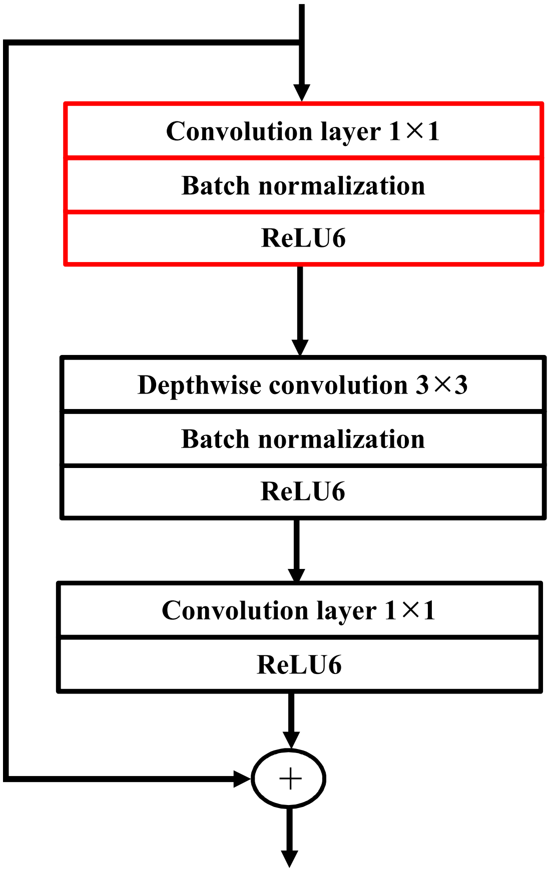

FIGURE 7. MobilenetV2 network model.

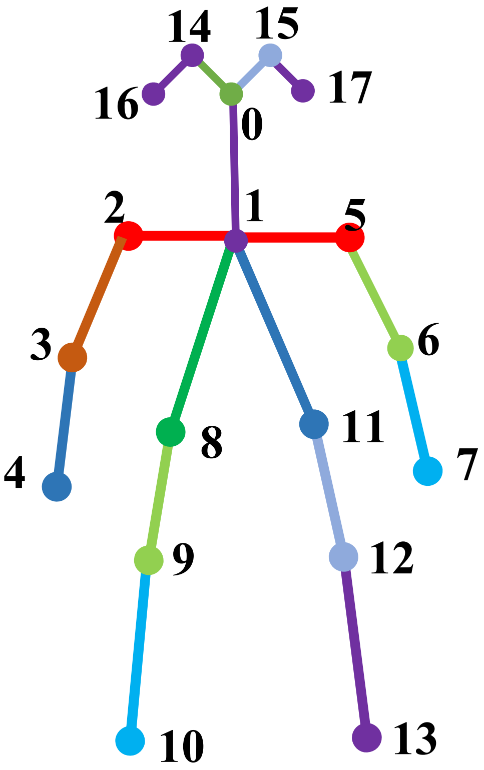

FIGURE 8. The process stores 18 points of body of OpenPose.

When the network learns most features of each state, the data are fed into the softmax activation function located on the last layer of LSTM to classify stylistic labels. The softmax function calculates the likelihood of the class. The probability is then used to determine the target class for the inputs. Probabilities will always be in [0: 1], and the sum of all probabilities is 1 [32].

IV. SIMULATION AND RESULT

A. SETUP

In this study, we used the configuration Core i5 8300H, 16GB RAM, and GTX 1050 Ti to train our model. Input data includes poses from MPII [24], and data are created by ourselves.

We evaluate the accuracy and execution time of the proposed model for two cases:

In the first case, we evaluate classification using a machine learning model (VGG19).

In the second case, we performed the deep learning model (mobilenetV2).

B. RESULT

1) CASE 1: CATEGORIZING USING MACHINE LEARNING

We carry out a training model to classify using several machine learning (ML) methods. We compare the accuracy of the OpenPose model using backbone VGG19 and the mobilenet in Figs. 11 and 12. The classification time using ML algorithms is described in 3 and 4. We see that the SVM + RBF algorithm has high accuracy (90%) and sufficient implementation time (15 seconds).

In this case, we used two models, namely SVM and VGG19. We evaluated and compared the performance of the two models using SVM and RBF, as shown in Tab. 5.

We see that the Openpose model with the VGG19 backbone has higher accuracy when the processing speed is lower. The VGG19 parameter was 143,667,240, and the accuracy was 91%. However, the implementation time is 15 hours and the processing speed is only one FPS.

To solve this problem, the Openpose model (mobilenet) improves the processing speed from 4 to 13.6 FPS and the number of training parameters is only 3,538,984. However, the accuracy was only 89%.

Moreover, the static and dynamic shapes of both models have quite a bit of confusion. It can be a misclassified posture of standing and walking since these two actions are quite similar. The two training models produced low results of 82% and 80%, respectively. In addition, the difference may be due to the unreasonable classification method (machine learning), and the model has not been trained in-depth or the input data of the system are consecutive frames. Therefore, we decided

TABLE 3. Result of comparing time classification for Openpose model (backbone VGG19).

Method	Decision Tree	Near Rest	Neural Net	Random Forest	SVM+RBF
Classification time(second)	10	19	21	18	15

TABLE 4. Result of comparing time classification for Openpose model (backbone mobilenetV2).

Criteria	Decision Tree	Near Rest	Neural Net	Random Forest	SVM+RBF
Classification time (second)	9	15	26	15	12

TABLE 5. Result of training and classifying by SVM + RBF model.

Result	Openpose (VGG19) + SVM (RBF)	Openpose (mobilenetV2) + SVM (RBF)
Accuracy (%)	91	89
Processing speed (frame per second)	4	14.6
Training time (hour)	15	12

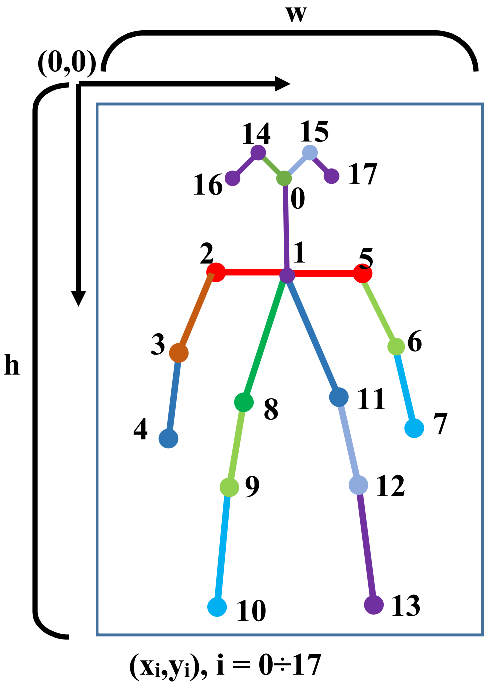

FIGURE 9. Description of four steps of pre-processing skeleton.

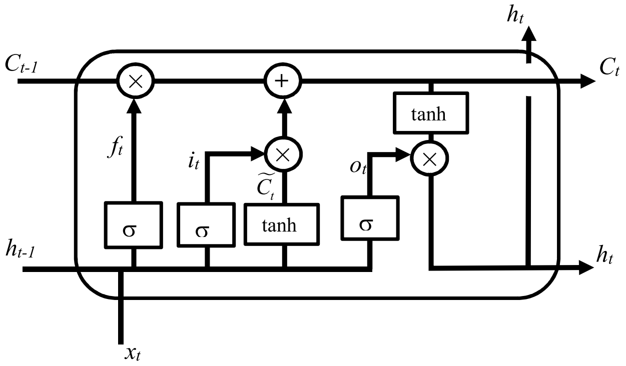

FIGURE 10. Description of four steps of pre-processing skeleton.

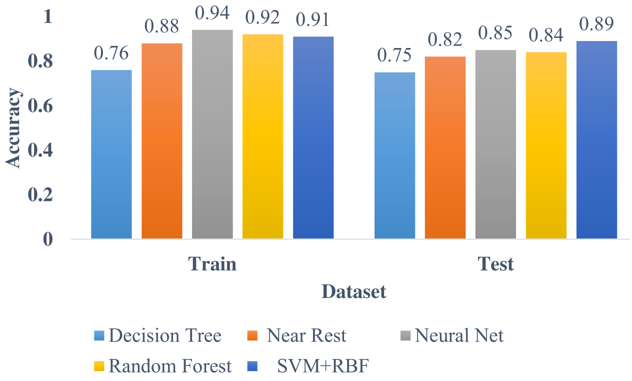

FIGURE 11. Results of classification for OpenPose(backbone VGG19) model.

to integrate LSTM to extract features that help categorize the training data well.

2) CASE 2: CATEGORIZING USING DEEP LEARNING

In the section, we apply the classification method using LSTM.

We can see that the system integrating LSTM achieves better results, as shown in Tab. 6. The accuracy is improved up to 10%, the training time is almost similar, and the processing speed is slower than that of the two FPSs. The accuracy of other poses, such as standing and walking, lying down, and falling, is shown in Figs. 14 and 13.

In addition, we show that VGG19 trains a long time (as shown in Tab. 2) without increasing the accuracy, as shown in Tabs. 6 and 3. In addition, time processing did not improve.

The system must have a trade-off between accuracy and speed. Therefore, we selected our posture recognition system

TABLE 6. Results of training and classifying for postures.

Model	Openpose (**VGG19**)+LSTM	Openpose (**mobilenetV2**)+LSTM
Average accuracy of training dataset (%)	92.1	90.7
Average accuracy of testing dataset (%)	91.3	90.1
Training time (hour)	15.3	12.7
Processing speed (FPS)	3.7	13

TABLE 7. Comparison of the proposal with other methods in terms of average accuracy and processing time.

Related work	Method	Average accuracy (%)	Processing speed (FPS)	Processing time(second)	Hardware
Ahsan Shahzad [4]	Threshold+SVM	88÷91	64	8.9	CPU
QingzhenXu [5]	Openpose+CNN	72÷92	≥ 45	45	GPU
Godwin Ogbuabor [33]	Using sensors of smartphone	80÷95	50	-	GPU
Proposal	**MobilenetV2+LSTM**	**83÷99**	**13**	**9**	**GPU**

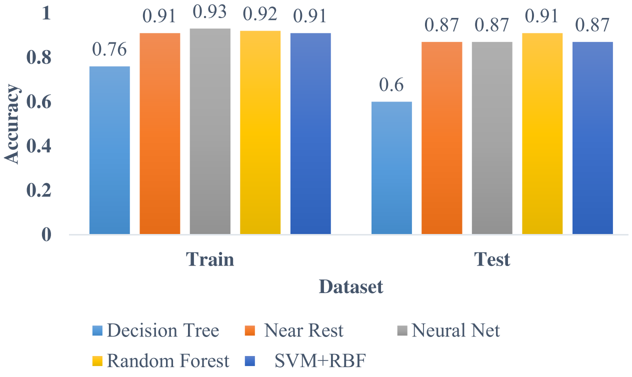

FIGURE 12. Results of classification for OpenPose (backbone mobilenetV2) model.

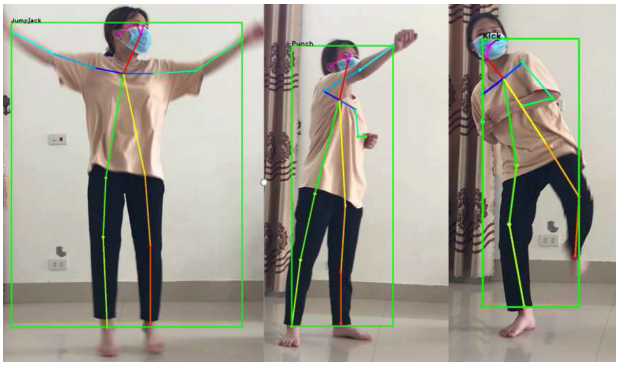

FIGURE 13. An example of classifying each posture on our dataset.

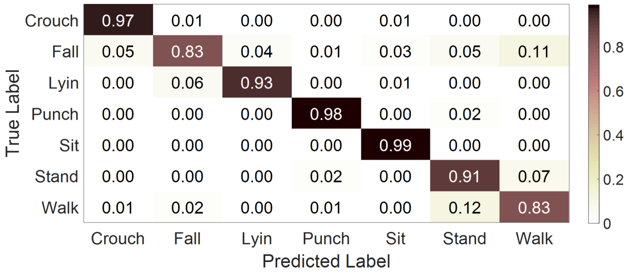

FIGURE 14. Results of classifying each posture of LSTM.

that uses the Openpose (Backbone mobilenetV1) + LSTM algorithm with 90.7% accuracy and a processing speed of 13 FPS, as shown in Tab. 6.

Figure 15 shows the experimental results. The other system reaches the highest accuracy with sit, punch, and crouch poses and mistakes among falling-lying-standing-walking poses. However, our method has also addressed this problem. The accuracy of stand and fall when using SVM classifications reached 82% and 80%, respectively, when our method of using LSTM had accuracies of 91% and 83%, respectively. The details are presented in Fig. 14. The results show that the proposed method exhibits positive improvements.

In addition, we compare it with other methods in terms of accuracy, performance, and hardware used to train the model, as shown in Tab. 7. The results show that the proposed method achieves up to 99% accuracy with the best processing speed and time (only 9 seconds).

In Tab. 7, we see that fall detection is an important application for health monitoring. The method of extracting posture by a threshold has poor performance because classification has low accuracy. The main reason is that environmental changes affect recognition efficiency. A good example is a small change in light or background that makes it extremely difficult to identify subjects. As a result, it leads to great difficulty for classification. Therefore, the accuracy of method [4] using UR fall detection and Charfi dataset only achieves about 90% with the SVM model.

In addition, we find that the results are different when using a similar Openpose method for estimating postures with different classification algorithms. For example, the authors [5] used UR fall detection and the NTU RGB+D action dataset with 93% accuracy. The authors [34] use the inception-ResNetv2 network for feature extraction and classification with more than 54 million parameters with 92% accuracy. The accuracy of the methods is not different from our method

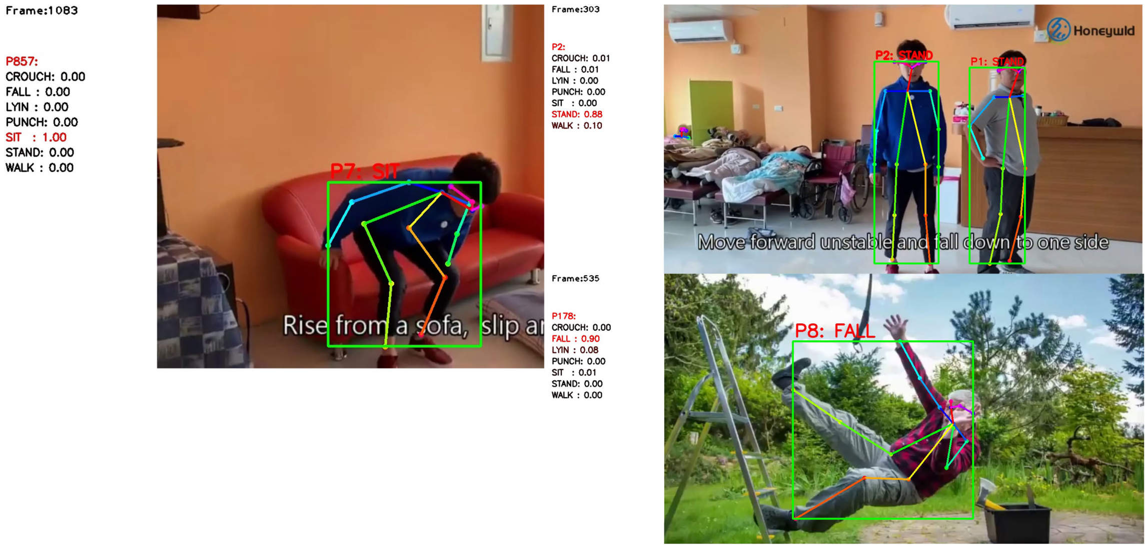

FIGURE 15. Results of Openpose (mobilenetV2) and LSTM models.

TABLE 8. Jetson Nano Kit specifications.

Model	Parameter
GPU	128-core Maxwell
CPU	Quad-core ARM A57 143GHz
Memory	4GB 64-bit LPDDR4 25.6 Gb/s
Display	HDMI/Display port
DC	5V 4A

TABLE 9. Result of system on JetsonNano kit.

Parameter	Measuring value
Average power consumption	6166mW
CPU	44% - 1.5GHz
Ram	3GB/4GB
Swap	1.579GB/8.1GB
FPS	2.1

when we use only two million parameters (LSTM network). In addition, the application characteristics of a health monitoring system are real-time, as the accuracy does not need to be too high. To compare the FPS prediction time, our system is five times faster with 13.5 FPS when the hardware used to train the model has the low configuration as Core i5 8300H, 16GB RAM, and GTX 1050 Ti.

3) CASE 3: PERFORMING ON JETSON NANO KIT

To examine the algorithm on real devices, we performed it on a real system. We evaluated the proposed system using the NVIDIA Jetson Nano (TM) Developer Kit with parameters, as shown in Tab. 8.

The evaluation results of the proposed system are listed in Tab. 9. As a result, the system uses 44% of the CPU at 1.5GHz frequency which consumes approximately 3GB/4GB of Kit. The processing speed was 2.1 FPS with an input size of 656 × 368. A processing speed of 5 FPS with a size smaller than 2/3 was achieved. The results show that the proposed system is capable of running on real devices.

V. CONCLUSION

This article focuses on studying CNN to estimate human posture through ML and DL algorithms for posture classification. The system recognized the poses with up to 99% accuracy by estimating the skeleton using Openpose with a mobilenetV2 backbone and LSTM. The results show that the system is capable of running on real hardware-based on evaluation using the Jetson Nano Kit. However, the system still has disadvantages, such as low posture recognition results and an acceptable FPS processing rate in several cases. The main reason is the poor quality dataset, low light, and video recording environment.

Therefore, the next direction will improve the system by adding data with many cases affecting the recognition results such as human-like objects, doing more with more poses combining experimentation with multiple subjects per frame, speeding up the processing of the system, and improving the accuracy by increasing the resolution of the video. We will also study how to optimize the execution time of the algorithm to suit embedded systems to get the most out of the GPU of Jetson Nano. We will then complete practical application products, such as controlling basic appliances with the electric current of a smart home or building application applications for exercise and sports.

REFERENCES

[1] R. Liu, A. Ramli, H. Zhang, E. Datta, and X. Liu, "An overview of human activity recognition using wearable sensors: Healthcare and artificial intelligence," *Hum.-Comput. Interact.*, vol. arXiv:2103.15990, pp. 1–14, Jul. 2021.

[2] H. Verma, D. Paul, S. R. Bathula, S. Sinha, and S. Kumar, "Human activity recognition with wearable biomedical sensors in cyber physical systems," in *Proc. 15th IEEE India Council Int. Conf. (INDICON)*, Dec. 2018, pp. 1–6.

[3] E. Bulbul, A. Cetin, and I. A. Dogru, "Human activity recognition using smartphones," in *Proc. 2nd Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, 2018, pp. 1–6.

[4] A. Shahzad and K. Kim, "FallDroid: An automated smart-phone-based fall detection system using multiple kernel learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 35–44, Jan. 2019.

[5] Q. Xu, G. Huang, M. Yu, and Y. Guo, "Fall prediction based on key points of human bones," *Phys. A, Stat. Mech. Appl.*, vol. 540, pp. 1–13, Feb. 2020.

[6] Z. Cao, G. Hidalgo, T. Simon, S. E. 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.

[7] E. C. Dinarević, J. B. Husić, and S. Baraković, "Issues of human activity recognition in healthcare," in *Proc. 18th Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, Mar. 2019, pp. 1–6.

[8] M. E. A. Elforaici, I. Chaaraoui, W. Bouachir, Y. Ouakrim, and N. Mezghani, "Posture recognition using an RGB-D camera: Exploring 3D body modeling and deep learning approaches," in *Proc. IEEE Life Sci. Conf. (LSC)*, Oct. 2018, pp. 69–72.

[9] Z. Huang, Y. Liu, Y. Fang, and B. K. P. Horn, "Video-based fall detection for seniors with human pose estimation," in *Proc. 4th Int. Conf. Universal Village (UV)*, Oct. 2018, pp. 1–4.

[10] Z. Zhong, F. Chen, Q. Zhai, Z. Fu, J. P. Ferreira, Y. Liu, J. Yi, and T. Liu, "A real-time pre-impact fall detection and protection system," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2018, pp. 1039–1044.

[11] J. Li, W. Su, and Z. Wang, "Simple pose: Rethinking and improving a bottom-up approach for multi-person pose estimation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, pp. 11354–11361.

[12] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2353–2362.

[13] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imag.*, vol. 11, no. 3, pp. 172–185, Jun. 2005.

[14] M.-C. Chen and Y.-M. Liu, "An indoor video surveillance system with intelligent fall detection capability," *Math. Problems Eng.*, vol. 2013, pp. 1–8, Nov. 2013.

[15] Y. Yuan, Z. Miao, and S. Hu, "Real-time human behavior recognition in intelligent environment," in *Proc. 8th Int. Conf. Signal Process.*, vol. 3, 2006, pp. 1–4.

[16] S. Shinde, A. Kothari, and V. Gupta, "YOLO based human action recognition and localization," *Proc. Comput. Sci.*, vol. 133, pp. 831–838, Jan. 2018.

[17] Z. Shi, J. A. Zhang, R. Xu, and G. Fang, "Human activity recognition using deep learning networks with enhanced channel state information," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.

[18] M. Z. Uddin and A. Soylu, "Human activity recognition using wearable sensors, discriminant analysis, and long short-term memory-based neural structured learning," *Sci. Rep.*, vol. 11, no. 1, p. 15, Aug. 2021.

[19] M. Moencks, D. De Silva, J. Roche, and A. Kondoz, "Adaptive feature processing for robust human activity recognition on a novel multi-modal dataset," *Comput. Vis. Pattern Recognit.*, vol. arXiv:1901.02858, pp. 1–13, Jan. 2019.

[20] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1653–1660.

[21] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, "MoDeep: A deep learning framework using motion features for human pose estimation," 2014, pp. 1–15, *arXiv:1409.7963*.

[22] F. Chen. (2019). *Realtime-Action-Recognition*. Accessed: Mar. 2019. [Online]. Available: https://github.com/felixchenfy/Realtime-Action-Recognition

[23] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3686–3693.

[24] L. Ke, M.-C. Chang, H. Qi, and S. Lyu, "Multi-scale structure-aware network for human pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 1–16.

[25] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1302–1310.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, pp. 1–9, *arXiv:1704.04861*.

[28] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1942–1950.

[29] S. Qiao, Y. Wang, and J. Li, "Real-time human gesture grading based on OpenPose," in *Proc. 10th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2017, pp. 1–6.

[30] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Jan. 2014, pp. 338–342.

[31] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2016, pp. 1–5.

[32] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *Proc. 4th Int. Conf. Multiple Classifier Syst. (MCS)*. Berlin, Germany: Springer-Verlag, 2003, pp. 125–134.

[33] G. Ogbuabor and R. La, "Human activity recognition for healthcare using smartphones," in *Proc. 10th Int. Conf. Mach. Learn. Comput.*, Feb. 2018, pp. 41–46.

[34] U. Nazir, N. Khurshid, M. A. Bhimra, and M. Taj, "Tiny-inception-ResNet-v2: Using deep learning for eliminating bonded labors of brick kilns in South Asia," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2019, pp. 39–43.

PHAT NGUYEN HUU received the B.E. and M.S. degrees in electronics and telecommunications from the Hanoi University of Science and Technology (HUST), Vietnam, in 2003 and 2005, respectively, and the Ph.D. degree in computer science from the Shibaura Institute of Technology, Japan, in 2012. He is currently a Lecturer at the School of Electrical and Electronic Engineering, HUST. His research interests include digital image and video processing, wireless networks, *ad-hoc* and sensor networks, and intelligent traffic systems (ITS), and the Internet of Things (IoT). He received the Best Conference Paper Award in SoftCOM, in 2011, the Best Student Grant Award in APNOMS, in 2011, and the Hisayoshi Yanai Honorary Award by the Shibaura Institute of Technology, in 2012.

NGOC NGUYEN THI received the B.E. degree in electronics and telecommunications from the Hanoi University of Science and Technology (HUST), Vietnam, in 2021. She is currently a Student at the School of Electronics and Telecommunications, HUST. Her research interests include digital image and video processing and smart-home applications.

THIEN PHAM NGOC is currently a Student at the School of Electrical and Electronic Engineering, HUST, Vietnam. His research interests include digital image and video processing and embedded systems.