The Combination of Face Identification and Action Recognition for Fall Detection

Ngu D. Dao, Thien V. Le, Hanh T. M. Tran*, Yen T. H. Nguyen, Tuan D. Duy

Abstract—Falls are very common unexpected accidents that result in serious injuries such as broken bones and head injuries. Detecting falls, taking falling patients to emergency rooms, and sending notification to their family in time are very important. In this paper, we propose a method that combines face recognition and action recognition for fall detection. Specifically, we identify seven basic actions that take place in the elderly daily life based on skeleton data extracted using the YOLOv7-Pose model. Two deep models which are Spatial Temporal Graph Convolutional Network (ST-GCN), and Long Short-Term Memory (LSTM) are employed to recognize actions using the skeleton data. The experimental results on our dataset show that the ST-GCN model achieves an accuracy of 90% which is 7% higher than the LSTM model.

Index Terms—Face recognition; action recognition; skeleton detection; Spatial Temporal Graph Convolutional Network; Long Short-Term Memory.

1. Introduction

THE elderly can suffer serious injuries, even death, from a fall, and the severity level increases with age. According to the National Council On Aging (NCOA) [1], one in four people over the age of 65 falls every year. Another statistic shows that the rate of elderly people falling each year is about 28-35% for those 65 and older and 32-42% for those over 75. In Vietnam, an estimated 1.5-1.9 million elderly people fall each year, 5% of which are hospitalized for injuries [2].

Early fall detection can help family members or doctors arrive promptly and help limit the consequences of falls, and even possibly prevent death. Therefore, the need for early fall detection systems and the ability to automatically notify is urgent.

Face recognition has been prominently utilized currently to get individual identities as it does not require a direct touch on the sensors like other biometric identification techniques such as fingerprints, iris, or voice recognition. Therefore, in accidental falling situations, it can identify the person in danger and send alarms to his/her family or doctors through an early falling detection system.

Human activity recognition has attracted the attention of researchers around the world. Many methods have been proposed for recognizing different kinds of activities [3], [5], [6]. Amongst all these activities, fall detection has an special importance because it is a common dangerous incident for people of all ages with a more negative impact on the elderly population.

In this paper, we propose a method to recognize human's face and actions, which provides a person's identity as well as his current moving. We combined YOLOv5Face and ResNet18 with cosine similarity for face detection and recognition. Moreover, we employed YOLOv7-Pose for extracting skeleton data that is then used with ST-GCN for action recognition. Then these methods have combined into a system for identity recognition and actions recognition that is deployed in indoor areas such as hospitals and nursing homes to monitor falls, especially among the elderly.

In summary, the contributions of the paper are as follows:

- We proposed combining YOLOv7-Pose and ST-GCN to classify 7 actions that commonly happen in daily life: standing, standing up, sitting, sitting down, walking, lying, and falling. To the best of our knowledge, this is the first work that combines these two methods.
- We compared and evaluated two models: YOLOv3 + AlphaPose and YOLOv7-Pose for extracting skeleton data; and two models: LSTM and ST-GCN for action recognition on extracted skeleton data on our dataset. The code, model, and dataset are available at *https://github.com/DuyNguDao/Identity-Action.git*.
- We combined face recognition with action recognition to integrate a person's identity into actions taking place.
- We evaluated and compared our method with different frameworks on face recognition and action recognition. The results show that the combination of YOLO5Face + ResNet and cosine similarity is better than MobileFaceNet. Moreover, ST-GCN combined with YOLOv7-Pose gives better accuracy

Ngu D. Dao, Thien V. Le, Hanh T. M. Tran, Yen T. H. Nguyen, Tuan D. Duy are with the University of Danang - University of Science and Technology, Danang, Vietnam (E-mail: hanhtran@dut.udn.vn).

**Corresponding author: Hanh T. M. Tran (E-mail: hanhtran@dut.udn.vn)*
Manuscript received October 28, 2022; revised November 24, 2022; accepted December 21, 2022.
Digital Object Identifier 10.31130/ud-jst.2022.539ICT.

than that combined with YOLOv3 + AlphaPose.

The rest of this paper is organized as follows. Section II presents the related works and Section III presents the details of the proposed method. In Section IV, we present the experiments and results. The conclusion in Section V ends the paper.

2. Related works

Currently, there are many research on face recognition, action recognition, and fall detection. Action recognition and fall detection are all based on video [3], [5], [6], or accelerometers data [4] .

Kuppusamy et al. performed the identification of 10 different actions based on convolutional neural networks (CNNs) and recurrent neural networks (LSTM-RNNs) on image and video datasets [3]. However, there is no falling action in this research. Jiang Wu et al. performed a similar task as Kuppusamy et al. but they only recognized 2 actions: falling and not falling [4]. Moreover, the authors used accelerometer data obtained from the accelerometer sensor.

Some research use human skeleton [5], [6] for ation recogntion. Sungil Jeong et al. feeds the skeleton data into the LSTM network to perform action identification for 3 different actions [5]. In additon, Sijie Yan et al. also performed 30 different action recognitions based on skeleton data [6]. However, the authors used a Spatial-Temporal Graph Neural Network (ST-GCN) instead of using LSTM. The authors used the OpenPose model [7] to extract skeleton data from the Kinetics dataset [8]. The evaluation results show the author's model has an accuracy of 72.4%. However, there is no fall activities in this work. In the research on action recognition and fall detection, most of the authors only perform action recognition without the combination of identity recognition. Some research recognizes many actions, but they don't focus on fall action and have low accuracy. The accuracy of the model depends on many different factors such as data, extraction model, and recognition method.

Therefore, in this paper, we use the YOLOv7-Pose [9] model to increase the accuracy of skeleton data extraction. From this skeleton data, we employed and compared two models: LSTM and ST-GCN to identify seven actions: standing, standing up, sitting, sitting down, walking, lying, and falling. In addition, we employed the YOLO5Face [10] for face detection to improve the ability to detect small and far-distance faces with fast speed. We combined this face detection model with the ArcFace face recognition model [11]. The evaluation results show that the combination of YOLOv7-Pose and ST-GCN gives the best results in comparison with YOLOv7-Pose + LSTM, YOLOv3 + AlphaPose + ST-GCN.

3. Proposed methods

3.1. Face recognition

Fig. 1 shows our propose model for facial recognition which is based on a similarity learning method. In particular, we utilize cosine similarity in this paper to match feature vectors of two faces. As shown in Fig. 1, the input image is fed into the YOLO5Face model to determine the position of the face, then the detected face is fed into ResNet model for feature extraction. The detected face is represented as a 512-dimensional vector, which is then matched with database to determine the identity of the face.

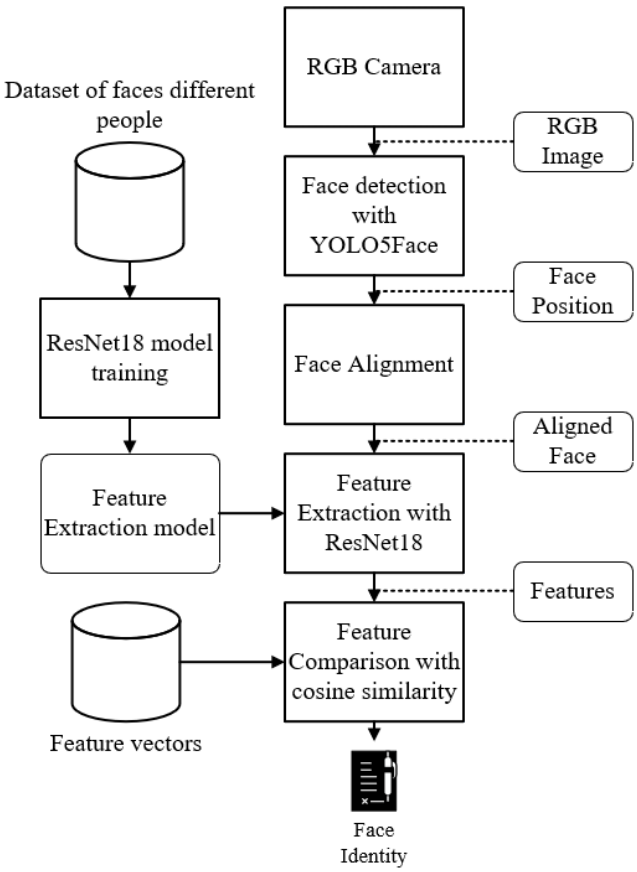

Fig. 1: Face recognition using YOLO5Face and Resnet18.

3.1.1. Face detection with YOLO5Face

The YOLO5Face face detection model was built by Delong Qui et al. in 2022 which is based on YOLOv5 to optimize the detection of large and small faces of different complexities [10]. This model can detect faces in real-time. We employed the author's model for face detection without re-training or fine-tuning.

3.1.2. ResNet18 feature extraction model combined with ArcFace loss function

To perform facial recognition with a similarity method, we use the ResNet18 model as a feature extractor and reduce the face data to a 512-dimensional vector. The embedding vector is then fed into the classification loss function to train the face recognition model. The choice of the loss function determines the accuracy of the model. Therefore, we chose the ArcFace loss function [11] to train with the ResNet18 network [12].

The Residual Network (ResNet) was born in 2015 [12] to overcome the vanishing gradient that affects badly when training a CNN network with hundreds of layers. ResNet18 is a variant of ResNet which has 22 layers that are made up of 4 groups of two residual blocks and 4 shortcut connections. We have changed the input size of the ResNet18 network to 112×112 instead of 224×224 as in the original architecture to suit the face images. In addition, we removed the classification layer in the output and used the linear layer to replaces the average pooling. The details of ResNet18's parameters architecture are described in Table 1.

TABLE 1: ResNet18 architecture.

Input size	Layer	size
$112^2 \times 3$	Conv1	3×3, stride=1
$112^2 \times 64$	Conv2x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$, stride=2
$56^2 \times 128$	Conv3x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$, stride=2
$28^2 \times 256$	Conv4x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$, stride=2
$14^2 \times 512$	Conv5x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$, stride=2
$7^2 \times 512$	Linear	25.088×512

The Additive Angular Margin loss function (ArcFace) [11] was improved by JianKang Deng et al. based on the softmax loss function. The formula for the softmax loss function is shown in Eq. 1:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \quad (1)$$

Where x_i denotes the embedding feature of the i-th sample, belonging to the y_i class. The embedding feature dimension is set to 512 in this paper. W_j denotes the j-th column of the weight W and b_j is the bias term. The batch size and the class number are N and n, respectively. This loss function is a combination of the cross-entropy loss function and softmax activation function. This function was used for face recognition. However, this function cannot explicitly optimize for embedding vectors containing face features, to further increase the similarity between faces within a class and increase the diversity of faces between classes, leading to decrease face recognition performance under different variations.

To simplify, the authors fixed the bias $b_j = 0$, then transformed $W_j^T x_i = \|W_j\| \|x_i\| cos(\theta_j)$, where θ_j is the angle between the weight W_j and feature x_i. Then, the authors fixed the weight $\|W_j\| = 1$ by l_2 normalization. The authors also fixed the feature vector $\|x_i\|$ according to l_2 normalization and rescale it to s (s is the radius of the hypersphere and the value of s is 64). This normalization step for weights and feature vectors makes the prediction only depend on the angle between the feature vectors and the weights. The learned feature vector is thus distributed over a hypersphere with a radius of s [11].

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s.\cos\theta_{y_i}}}{e^{s.\cos\theta_{y_i}} + \sum_{j=1, j\neq y_i}^{n} e^{s.\cos\theta_j}} \quad (2)$$

Since the feature vectors are distributed around each central feature according to a hypersphere, the author has added an additive angular margin m between the feature x_i and the weight W_{y_i} simultaneously enhancing intra-class compactness and inter-layer differentiation. Finally, we have the following ArcFace loss function [11]:

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s.\cos(\theta_{y_i}+m)}}{e^{s.\cos(\theta_{y_i}+m)} + \sum_{j=1, j\neq y_i}^{n} e^{s.\cos\theta_j}} \quad (3)$$

3.1.3. Cosine similarity

For face matching, we use cosine similarity to measure the similarity between two embedding vectors, $\mathbf{x_1} = (x_1, x_2, \ldots, x_n)^T$ and $\mathbf{x_2} = (x_1^{'}, x_2^{'}, \ldots, x_n^{'})^T$. The angle value $cos(\alpha)$, which is calculated as in Eq. 4, is between 0 and 1.

$$cos(\alpha) = \frac{x_1 \cdot x_1^{'} + x_2 \cdot x_2^{'} + \ldots + x_n \cdot x_n^{'}}{\sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} \cdot \sqrt{x_1^{'2} + x_2^{'2} + \ldots + x_n^{'2}}} \quad (4)$$

If two faces are the same, the angle value will be large and vice versa. However, for this method, when strangers appear in the image, it is easy to misidentify. To overcome this problem, we need to choose a confidence threshold λ between 0 and 1 to classify an unknown person: if $cos(\alpha) < \lambda \Rightarrow$ a person is unknown.

3.2. Action recognition

In this paper, we employed the YOLOv7-Pose model to extract key points on the human body through images combined with the ST-GCN or LSTM for action recognition to identify seven actions: standing, standing up, sitting, sitting down, walking, lying, and falling using the skeleton data. The combination is shown in Fig. 2.

3.2.1. Key points detection with YOLOv7-Pose

Key-point detection includes detecting the position of the person in the image as well as the position of key points on the body. These points are spatial locations of prominent features of the human body in the image. As the action changes, the skeletons connecting the key points change. Detecting the correct location of these key points on the body can help to improve recognition performance. Therefore, we propose to use the YOLOv7-Pose [9] model to extract the key points and locations of people in the image. The YOLOv7-Pose model is built based on the YOLO-Pose [13] and YOLOv7 [14] for high accuracy and fast implementation by using E-LAN and E-ELAN architecture instead of CSP-Darnet51 architecture used in YOLOv5.

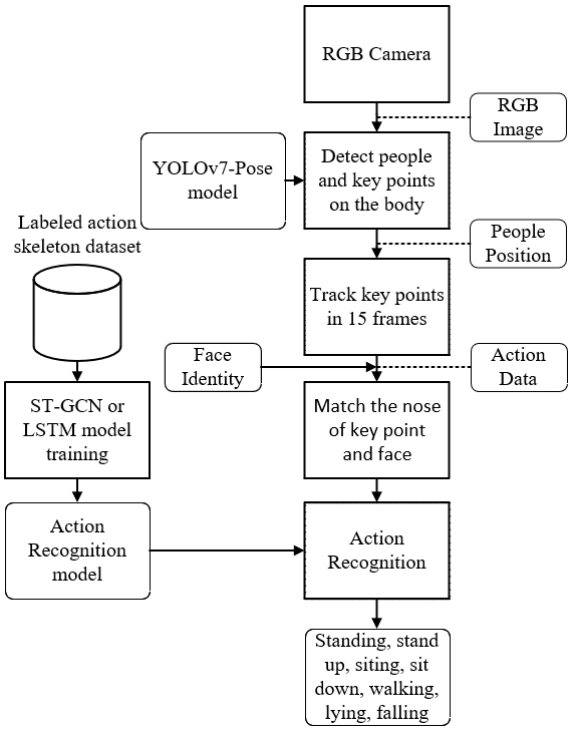

Fig. 2: Action recognition using YOLOv7-Pose and ST-GCN or LSTM.

3.2.2. Action recognition based on LSTM

Long Short-Term Memory (LSTM) [15] is a special type of Recurrent Neural Network (RNN), which is commonly used for sequence data. LSTM is proposed to overcome the vanishing and exploding gradient problems of conventional RNNs. LSTM can remember information for a long time by selecting what information needs to be remembered and which information needs to be forgotten. In this paper, we use LSTM with 3 hidden layers to output 128-d vectors, then the output is fed into Linear layers to classify the input action as shown in Fig. 3.

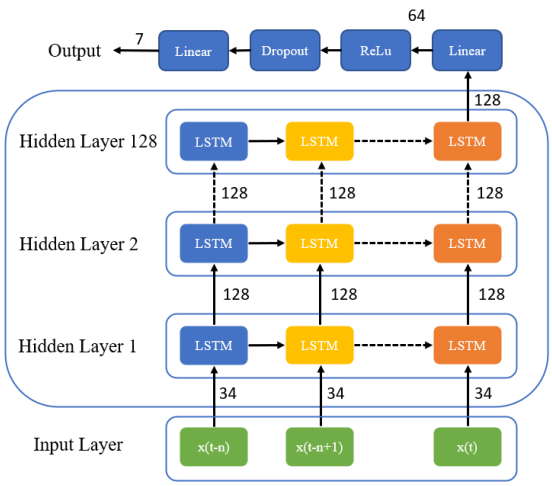

Fig. 3: Our LSTM architecture with 3 layers.

The loss function used to train this action recognition model is Cross Entropy as in Eq. 5 where $y_c^{(i)}$ and $\hat{y_c^{(i)}}$ are respectively a target value and a softmax probability of a predicted value for the c^{th} class, N is the total number of samples.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{7} y_c^{(i)} \log(\hat{y}_c^{(i)}) \quad (5)$$

3.2.3. Action recognition based on ST-GCN

Moreover, to recognize human action based on the skeleton, the ST-GCN model is also employed in this paper. To build a representation of the skeleton sequence for action recognition, Graph Neural Networks [16] are extended to the Spatial-Temporal graph model called ST-GCN [6]. Fig. 4 shows the Spatial-Temporal Graph of a skeleton sequence of T frames with each frame having N nodes. The node set includes all joints in a skeleton (showed as blue nodes in Fig. 4). The edge set consists of the set of all joints on the human body that are naturally connected (showed as blue edges in Fig. 4) and the set of edges connecting the same joints between frames(showed as green edges in Fig. 4).

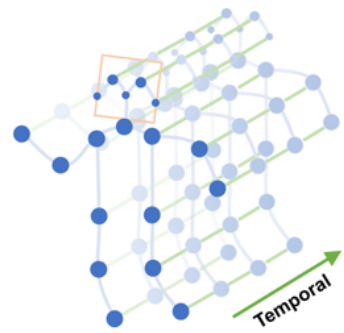

Fig. 4: Spatial-Temporal Graph of a skeleton sequence [6].

Skeleton input data has temporal and spatial properties that are important for motion detection. Therefore, the ST-GCN is built to be able to synthesize both spatial and temporal information. The architecture of an ST-GCN unit is shown in Fig. 5. In this paper, we used an ST-GCN architecture consisting of 9 layers of Spatial-temporal Graph convolution operators (ST-GCN unit).

The input to the ST-GCN model is only the coordinates of the joints, so it may not capture the motion dynamics effectively. To improve the recognition performance, the two stream ST-GCN model (as in Fig. 6) was used. The pose stream of input data is (x, y, s) where x, y are the coordinates of the node and s is node's confidence score. In the motion stream, input is the difference between the (x, y) coordinates of corresponding nodes in two consecutive frames. Both streams share the same network structure. The features of the two streams are combined and fed into a fully connected layer with a sigmoid classifier to predict the action.

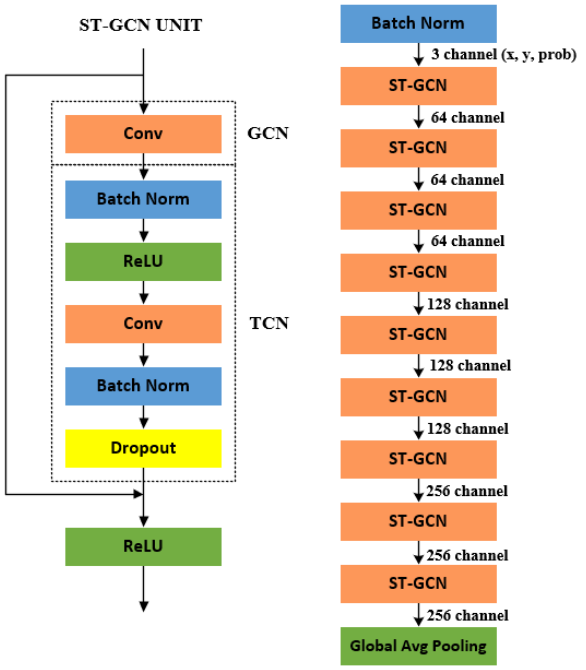

Fig. 5: Architecture of an ST-GCN unit and ST-GCN with 9 layers.

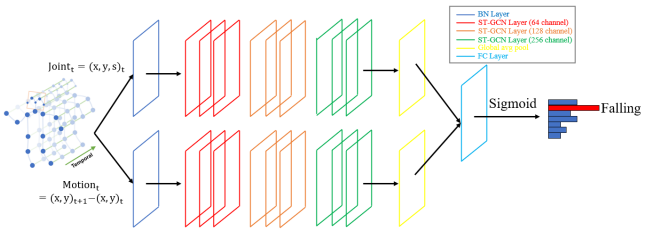

Fig. 6: Two Stream ST-GCN model. [17]

The loss function used to train the model is Binary Cross Entropy (BCE) as shown in Eq. 6 where $y_c^{(i)}$ and $\hat{y}_c^{(i)}$ are the target value and the predicted value, respectively; N is the number of samples.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{7} y_c^{(i)} \log(\hat{y}_c^{(i)}) + (1 - y_c^{(i)}) log(1 - \hat{y}_c^{(i)}) \tag{6}$$

4. Experiments and Results

4.1. Evaluation of face recognition model

4.1.1. Dataset

We used three datasets for evaluating face identification: (1) CASIA-WebFace dataset [18] consisting of 494414 images with 10575 different identities, (2) LFW dataset [19] contains 12233 images with 1680 identities and AgeDB-30 [20] including 12240 images with 440 identities and (3) FaceScrub dataset [21] including 10600 images of 530 different identities.

4.1.2. Evaluation results

To train and validate the ResNet18 model with ArcFace loss function, we used CASIA-WebFace dataset as a training set and the LFW dataset as a validation set. We also trained and validated lightweight MobileFaceNet [22] with ArcFace loss function on the same training and validation sets for the comparison purpose. We trained two models on a computer with the following configuration: CPU – Intel Xeon Processor, 16Gb RAM, GPU – Tesla P100. The learning rate is 0.1. Training time was about 82 hours. Then the trainded ResNet and MobileFaceNet model are used to extract embedding vectors that are fed into matching steps using cosine similarity. The FaceScrub dataset was used to evaluate the face recognition model.

We used accuracy to evaluate this model, following the Leave-One-Out Cross-Validation (LOOCV) method. All models were evaluated on a computer with the following configuration: CPU – AMD Ryzen 7 4800H, 16Gb Ram, GPU – NVIDIA GeForce GTX 1650. The confidence threshold is set to 0.3 ($\lambda = 0.3$). The result in Table 2 shows that the accuracy of our ResNet model is 97%, which is 3% higher than the MobileFaceNet model. In addition, the testing speed is faster, taking 7.48 milliseconds to recognize a face, which is lower than the testing time of the MobileFaceNet model (8.36ms).

TABLE 2: Comparative results of our face recognition model on test a dataset.

Evaluation metrics	MobileFaceNet	ResNet18
Accuracy	0.94	0.97
# Parameters	2.059.520	24.025.600
Number of layers	47	22
Testing time	8.36 ms	7.48 ms

4.2. Evaluation of action recognition model

4.2.1. Dataset

We collected our 7 actions dataset with an IP Wi-Fi camera (KBVISION Kbone KN-H41P 2K 4.0MP). Moreover, we checked and re-labeled the Fall Detection dataset [23] and combined this dataset with our dataset to increase the number of videos. The combined database includes 2.859 videos for training and 1.206 videos for testing. The entire video of one action was fed into a YOLOv7-Pose model to extract the skeleton data for training and testing action recognition model.

4.2.2. Evaluation results

We used 4 evaluation metrics: Accuracy, Macro Average Precision (MAP), Macro Average Recall (MAR), and Mean F1-score [24] to evaluate models when our data is imbalanced. We trained the action recongnition models on a computer with the following configuration: CPU – AMD Ryzen 7 4800H, 16Gb Ram, GPU – NVIDIA GeForce GTX 1650. The learning rate is 0.0001. Training time is about 1 hour.

TABLE 3: Comparative results of action recognition model on a test dataset.

Evaluation metrics	LSTM	ST-GCN
Accuracy	0.83	0.90
MAP	0.81	0.91
MAR	0.80	0.89
Mean F1-score	0.81	0.90
# Parameters	352.775	7.922.301
Testing time	**1.2** ms	7.1 ms

The results in Table 3 show that the ST-GCN model gives high identification results. The average accuracy of all classes is 90%, which is 7% higher than the LSTM model but the processing time of the ST-GCN is much slower than LSTM.

TABLE 4: The comparative results of the ST-GCN model combined with two skeleton extraction models (YOLOv3 + AlphaPose, YOLOv7-Pose).

Evaluation metrics	YOLOv3 + AlphaPose + ST-GCN	YOLOv7-Pose + ST-GCN
Accuracy	0.84	0.90
MAP	0.84	0.91
MAR	0.81	0.89
Mean F1-score	0.83	0.90
# Parameters	55.391.944	88.160.753
Testing time	2.53 s	5.42 s
# Undetected frame rate	12.540 / 63.290	6.164 / 63.290

To be able to evaluate the effectiveness of the combination of YOLOv7-Pose and ST-GCN, we also combined the YOLOv3 model with AlphaPose [25] to extract skeleton data instead of using YOLOv7-Pose. Then ST-GCN was used for action recognition on extracted skeletons. The comparison results in Table 4 show that YOLOv3 combined with AlphaPose has lower accuracy than YOLOv7-Pose. However, it improves testing time. All evaluation results shown in Table 3 and Table 4 are performed on computers with CPU – AMD Ryzen 7 4800H, 16Gb Ram, GPU – NVIDIA GeForce GTX 1650.

In addition, Table 4 shows that the number of undetectable frames of the combination of YOLOv7-Pose and ST-GCN is much lower than YOLOv3 combined with AlphaPose. As shown in Fig. 7, the extracted skeleton data using YOLOv7-Pose is better than YOLOv3-AlphaPose in terms of the location of key points on the body. This helps improve the action recognition results (in Table 4).

4.3. Evaluation of the method for face recognition and action recognition.

4.3.1. Dataset

To evaluate the method of face identification and action recognition on the same dataset, we captured three videos (the duration of each video is 1 minute 30 seconds, 1 minute 1 second, and 3 minutes 23 seconds), containing 7 actions with two face IDs using the same IP Wi-Fi camera 30FPS (KBVISION Kbone KN-H41P 2K 4.0MP). After removing volumes of 30 video frames that contain the transition between two actions, we have 3498 video volumes of 30 frames in total.

Fig. 7: The qualitative result of (a) YOLOv7-Pose + ST-GCN and (b) YOLOv3 + AlphaPose + ST-GCN.

4.3.2. Evaluation results

TABLE 5: The accuracy of face recognition and action recognition on three videos with 3498 video volumes.

Evaluation metrics	Face recognition	Action recognition
Accuracy	0.93	0.92
MAP	1.0	0.86
MAR	0.91	0.91
Mean F1-score	0.95	0.88

Table 5 shows the accuracy of face recognition and action recognition on the same dataset, containing three videos of action with two IDs. It can be seen from Table 5 that the accuracy, MAP, MAR, and Mean F1-score

are good enough for real-life scenarios. The qualitative results of the method that combines two models are shown in Fig. 7.

5. Conclusion

In this paper, we have presented the method of identity recognition and action recognition, intending to detect and alarm falls in the elderly based on video analysis using deep learning models. The ResNet18 model is built for face recognition with cosine similarity. For action recognition, the YOLOv7-Pose is proposed to extract the skeleton to improve the accuracy of the ST-GCN action recognition models. Experimental results show that the ResNet18 face recognition model has a high accuracy of 97% on the FaceScrub dataset and the ST-GCN action recognition model combined with YOLOv7-Pose has an accuracy of 90%, which is 6% higher than the combination with YOLOv3 and AlphaPose. In addition, we have successfully built an identity recognition and action system for the elderly that achieves an average of 15 – 20 frames per second with an AMD Ryzen 7 4800H CPU and NVIDIA GTX 1650 graphics card. In the future, this method can be developed for recognizing more actions such as running, jumping, fighting which is aiming to detect and alarm abnormal behaviors of children.

References

[1] NCOA, [Online] Available: https://ncoa.org/article/get-the-facts-on-falls-prevention.

[2] H. T. M. Châu, 2018, [Online] Available: http://benhvientinh.quangtri.gov.vn/vi/scientific-research/Nguy-co-te-nga-o-benh-nhan-cao-tuoi-dang-dieu-tri-tai-benh-vien-da-khoa-tinh-Quang-Tri.html.

[3] Kuppusamy, P., and C. Harika. "Human action recognition using CNN and LSTM-RNN with attention model." *Int. J. Innov. Technol. Explor*. Eng 8 (2019): 1639-1643.

[4] Wu, Jiang, et al. "Fall detection with cnn-casual lstm network." *Information* 12.10 (2021): 403.

[5] Jeong, Sungil, Sungjoo Kang, and Ingeol Chun. "Human-skeleton based fall-detection method using LSTM for manufacturing industries." *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*. IEEE, 2019.

[6] Yan, Sijie, Yuanjun Xiong, and Dahua Lin. "Spatial temporal graph convolutional networks for skeleton-based action recognition." *Thirty-second AAAI conference on artificial intelligence*. 2018.

[7] Cao, Zhe, et al. "Realtime multi-person 2d pose estimation using part affinity fields." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[8] Kay, Will, et al. "The kinetics human action video dataset." *arXiv preprint arXiv*:1705.06950 (2017).

[9] Github, [Online] Available: https://github.com/WongKinYiu/yolov7/tree/pose.

[10] Qi, Delong, Weijun Tan, Qi Yao, and Jingfeng Liu. "YOLO5Face: why reinventing a face detector." *arXiv preprint arXiv*:2105.12931 (2021).

[11] Deng, Jiankang, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. "Arcface: Additive angular margin loss for deep face recognition." *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690-4699. 2019.

[12] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[13] Maji, Debapriya, Soyeb Nagori, Manu Mathew, and Deepak Poddar. "YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss." *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2637-2646. 2022.

[14] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." *arXiv preprint arXiv*:2207.02696 (2022).

[15] Sak, Haşim, Andrew Senior, and Françoise Beaufays. "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition." *arXiv preprint arXiv*:1402.1128 (2014).

[16] Wu, Zonghan, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems 32*, no. 1 (2020): 4-24.

[17] Github, [Online] Available: https://github.com/littlepure2333/2s_st-gcn.

[18] Yi, Dong, Zhen Lei, Shengcai Liao, and Stan Z. Li. "Learning face representation from scratch." *arXiv preprint arXiv*:1411.7923 (2014).

[19] Zheng, Tianyue, Weihong Deng, and Jiani Hu. "Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments." *arXiv preprint arXiv*:1708.08197 (2017).

[20] Moschoglou, Stylianos, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. "Agedb: the first manually collected, in-the-wild age database." *In proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 51-59. 2017.

[21] Ng, Hong-Wei, and Stefan Winkler. "A data-driven approach to cleaning large face datasets." *In 2014 IEEE international conference on image processing (ICIP)*, pp. 343-347. IEEE, 2014.

[22] Chen, Sheng, Yang Liu, Xiang Gao, and Zhen Han. "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices." *In Chinese Conference on Biometric Recognition*, pp. 428-438. Springer, Cham, 2018.

[23] ImViA, *Fall detection dataset*, 2020, [Online] Available: https://imvia.u-bourgogne.fr/en/database/fall-detection-dataset-2.html.

[24] Stack Exchange, Micro Average vs Macro average Performance in a Multiclass classification setting, 2016, [Online] Available: https://datascience.stackexchange.com/questions/15989/micro-average-vs-macro-average-performance-in-a-multiclass-classification-setting

[25] Fang, Hao-Shu, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. "Rmpe: Regional multi-person pose estimation." *In Proceedings of the IEEE international conference on computer vision*, pp. 2334-2343. 2017.

Ngu D. Dao is currently a final year student at the Department of Electronics and Telecommunications, Danang University of Science and Technology, Vietnam. His research interests include machine learning, deep Learning, image processing, object detection and classification, and face recognition.

Thien V. Le is currently a final year student at the Department of Electronics and Telecommunications, Danang University of Science and Technology, Vietnam. His research interests are in the field of artificial intelligence, machine learning, and deep learning models.

Hanh T. M. Tran is currently a Lecturer with the Department of Electronics and Telecommunications, Danang University of Science and Technology, Vietnam, where she joined since 2009. She received the B.Eng. and M.Eng. degrees in Electronics and Telecommunications from Danang University of Technology and The University of Danang in 2008 and 2011, respectively. She obtained the Ph.D. degree from the University of Leeds, United Kingdom, in 2018. Her main research interests include image/video processing, machine learning, deep learning, anomaly detection, object detection and recognition.

Yen T. H. Nguyen is a lecturer in the Department of Telecommunications and Electronics, Danang university of Science and Technology. She got Doctoral degree in 2018 at Leeds university, the United Kingdom and got master degree in 2013 at Twente University, the Netherlands. Her major is signal processing and optical fiber network.

Tuan D. Duy received the B.E. degree from The University of Danang-University of Science and Technology, Da Nang, Vietnam, in 2008, and the M.E. and Ph.D. degrees in computer science and information engineering from National Cheng Kung University, Taiwan, in 2013 and in 2019, respectively. He is currently an PhD Lecturer at the Department of Electronic and Telecommunication Engineering, The University of Danang-University of Science and Technology, Da Nang, Vietnam. His research interests include IP mobility management, wireless communications, mobile network protocols and vehicular network.