A NOVEL ARTIFICIAL INTELLIGENCE BASED INTERNET OF THINGS FOR FALL DETECTION OF ELDERLY CARE MONITORING

[1]Noushini Nikeetha.P, [2]Gopinath, [3,*]Kirubasri G.V. ,[4]Haritha Sasikumar, [5]Yazhini Tamilanban and [6]Jagruti Patil

[1,3,4,5,]Department of Biomedical Engineering, VelTech Multitech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai, India.
[2]Bharath Institute of Higher Education and Research,Chennai,India.
[5]Department of Biomedical Engineering,L.D. College of Engineering, Ahmedabad.India.

* Correspondence: kirubavelu1611@gmail.com

Abstract

A fall of an older adult often leads to severe injuries and is found to be a significant reason for the death due to post-traumatic complications. Many falls happen in the home atmosphere and prevail unrecognized. Thus, the need for reliable early fall detection is necessary for fast help. Lately, the emergence of wearables, smartphones, IoT, etc., made it possible to develop systems fall detection which aids in the remote monitoring of the elderly. The goal is to allow intelligent algorithms and smartphones to detect falls for elderly care and to monitor them regularly. This work presents the Artificial Intelligence of Things for Fall Detection (AIOTFD) system using a slime mould algorithm (SMA) to optimize the final data. The features extracted using SqueezeNet further CNN based SMA used for data optimization. The validation of the AIOTFD model performance is evaluated through the Multiple Cameras Fall Dataset (MCFD) and UR Fall Detection dataset (URFD). The empirical results accentuated the assuring realization of the model compared to other state-of the art methods.The obtained results shows our proposed AIOTFD attains accuracy of 99.82% and 99.79% and databases can be used for additional investigation and optimizations to increase the recognition rate to enhance the independent life of the elderly.

Keywords: Fall detection system; Elderly Care; Remote Monitoring; Artificial Intelligence; Internet of Things

1.Introduction

IoT is defined as an integration of various different devices like computer systems, mechanical and digital appliances via the internet which has the capability to traverse and communicate any kind of information and motivate the designing of novel innovative applications or devices that does not demand any human-to-computer or human-to-human interactions. [1] Collection and sharing of data and its interconnectivity are the fundamental processes that support innovation using IoT. Applications of IoT in healthcare include supervising different parameters determining health in real time for the designing of suitable applications and services in conformity to the present requirements and health complications of the community. Few examples of IoT implementations in the healthcare industry involve wearable devices, m-Health, etc. The advantages of IoT applications in health care consists of enhanced information and resource access, ameliorating the potential of patients to make better decisions regarding medical care and enhancing the quality of life and safety for patients [2]. Enhanced comfort of patients leads to better satisfaction and fast paced recovery of the patients. One huge benefit of IoT smart solutions in hospitals is future proofing. [3]. The

IoT involves dealing with enormous data volumes that has to be processed and implemented in applications which require AI. Hence IoT and AI go hand in hand. AI algorithms enhance the experience for customers of any functions related to IoT. Since IoT is a relatively new technology it is not completely perfect. Certain parameters like the speed of IoT data transmission and accuracy have room for enhancement. The process of self-learning is the basis of AI which could be exploited by IoT. While integrating AI and IoT in the health sector, there is a possibility of obtaining increased effectiveness of operation. The important processes that help with the provision of smart and systematic usage of AI algorithms in IoT devices are as follows: collection of data, monitoring or analyzing, controlling, optimization or training, and automation that is modeling, predicting. The main applications of IoT devices enabled with AI are Chronic disease management, Medical staff, patients, and inventory supervision, minimized waiting time, Remote health control, and Drug management. One of the biggest advantages of these is that they enable one to monitor in real time and provide solutions immediately. This kind of analysis is feasible only under the provision of a continuous stream of data. Nevertheless, it is highly complicated for a system performing high computation processing to keep up with all the continuous stream of data from various different sensors, for which AI systems are utilized since they are capable of minimizing the large amount of data and make smart data management possible. Most essentially, the significant task in this is to reason with the data provided while also edge and fog computing. Analysis of data using edge detection in spite of more centralized location helps with real time monitoring using IoT devices. Overall, when AI algorithms are utilized to process data obtained from different IoT smart sensors on the edge proposes an advanced preservation and supervision.[4]As discussed earlier, the methodologies of mobile communications and IoT are being developed to collect information on people and the environment which could be used for different smart applications and services. Highly demanding tasks like remote supervision of the disabled and elderly people inhabiting smart homes is made possible by this technique. It is considered a challenging task because of daily unfortunate activities like falls. Fall has been found to be one of the crucial reasons for in elderly people due to post traumatic complications. Therefore, early indication of fall installed in smart homes for the elderly and disabled is a prime requirement in order to increase the ability to provide treatment on time and enhance their survival rate. This is made accessible with the help of smartphones, AI, IoT, wearable etc. in smart home care systems. [5]

2. Literature Review

Nahian *et al*. proposed a study where detecting fall is regarded as a time-series problem in order to inspect the interdisciplinary time-series features. An individual accelerometer sensor was utilized for detecting fall and the fresh acceleration data which consists of a multivariate time-series signal was collected that was converted to a univariate time series signal by obtaining the signal magnitude vector (SMV). Then interdisciplinary time-series features for their corresponding univariate SMVs were removed. They put forth a data analysis technique which aids in reduction of a plethora of features into a lesser number of dominant features which reduces overfitting. The Boruta algorithm was employed to select the top five important features. Lastly, these dominant features were utilized in the training of different ML classifiers which detect the fall.[6]. Mahendran et al. presented a convolutional autoencoder-based IOT-based health-care system approach using biosignals [7].Mrozek, D., et al presented an approach based on Cloud for detection of fall and large scale observation of the elderly. Here, the obtained data is passed on to the Cloud for the process of classification (the Cloud-based fall detection) or for the construction of a depiction of the observed person (the Edge-based fall detection). Two different architectures were used for detecting fall, and the procedure for the detection process was tested in the Whoops system which applies both architectures. The classifier utilized for detection executed in the Cloud was assembled in the Azure Machine Learning Studio which used the Boosted Decision Tree classification algorithm. The ML model for classification was trained and provided as a Web service. The fall detection procedure has been summarized in the system and could be implored from the Cloud [8]. Mahendran et l. have proposed a Internet of Medical Things (IoMT) based biometric key authentication scheme for body sensor network to access patient's data remotely[9]. Ogawa, Y. et al proposed a methodology for detecting fall which uses IR Array sensors. It provides an economical detection by a type of non-wearable device which could preserve privacy. The system put forth develops learning parameters, executes machine learning, and builds a classification model. The process of fall detection obtains the distribution of temperature and performs detection tasks using the developed classification model. Machine learning has been utilized for examination of features in an abundance of temperature distributions which has been obtained by Raspberry Pi3 Model B employing MLX 90621 of Melexis -IR array sensor. Voting classifiers are utilized for detection of fall. [10]. Nooruddin et al, proposed a device-type invariant based on IoT for detecting fall and relief system for observation of a huge amount of people in real-time. Devices such as Smartphones, Raspberry Pi, Arduino, are utilized to supervise an enormous populace. Data acquired from the accelerometer are continually redirected to a multi-processing server which consists of an already trained ML model that examines the data in order to indicate when a fall has taken place. The server feeds the outcomes of the classifier back into the respective gadgets.[11]

3. System Architecture:

The overview of the model is shown in figure1. The model for fall detection put forward employs a smartphone for the procedures of processing. The AIOTFD model is capable of detecting the incidence of falling at home to protect elderly people. Data acquisition, pre-processing, feature extraction based on SqueezeNet, parameter tuning based on SMA, and classifier based on SMA-CNN are all part of the proposed AIOTFD model. The input videos are first collected and then uploaded to a cloud service for further processing, where the model proposed is implemented.

The separated video frames are then pre-processed with three different key stages to improve the quality of the video frames: resizing, augmentation, and normalisation. Following that, using the SqueezeNet model, extraction of the features from the video is performed to provide meaningful feature. Furthermore, the Slime mould technique is used to tune the SqueezeNet model's hyper-parameters. Following that, the feature vectors are loaded to a SMA-CNN based classifier model to detect falls. The next steps will be taken based on the results of the classifier. The following steps are executed based on the classification outcome's value.

- When a fall is detected and classified as a class 1 event, the patient device receives an alarm, due to which the attendant could be immediately reported if the fall has not been eliminated by the monitored person from the application.
- When a non-fall incidence is identified and represented as class 0, no alert is generated and the occurrence of the event is disregarded. Physicians and caregivers might monitor the elderly in real time using backend technologies from faraway locations. Furthermore, the backend technology assists doctors in treating diseases by collecting data and patient history.

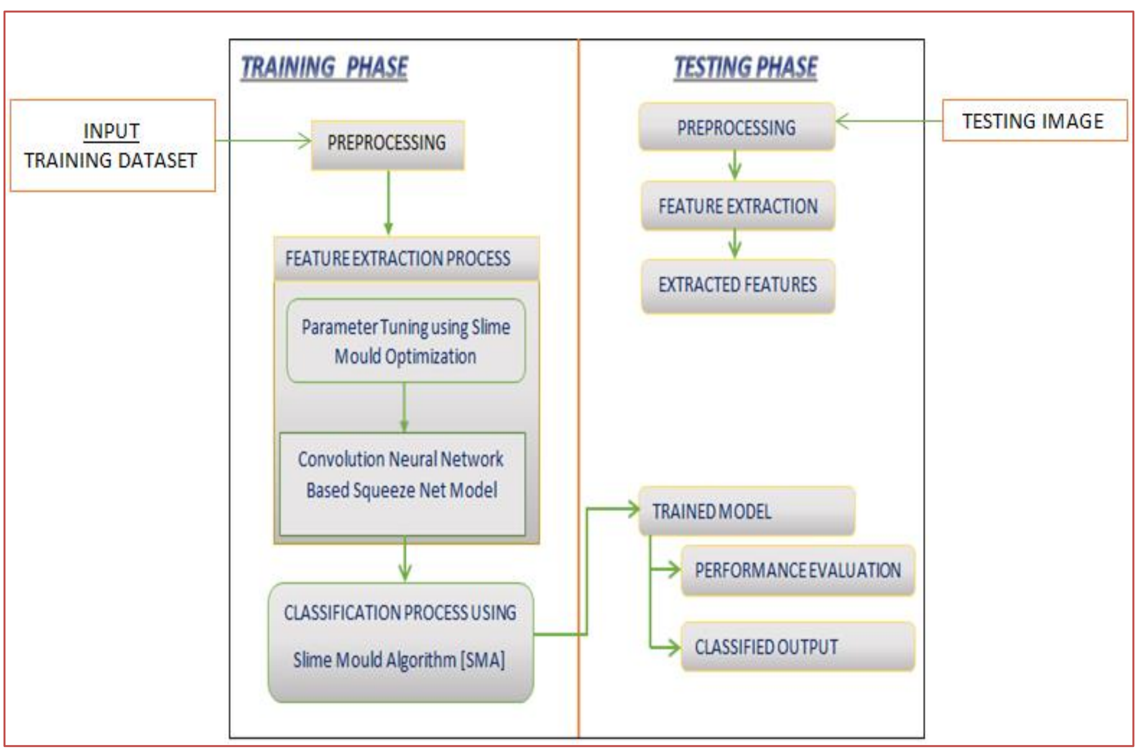

Figure 1. Functional block diagram of proposed model

4.Working process of AIOTFD model

The entire functioning procedure of the AIOTFD model includes various sub procedures that are data acquisition, pre-processing, feature extraction using SqueezeNet, hyper parameter tuning using Slime mould algorithm. After the parameter tuning phase, the SMA is combined with a variational autoencoder (SMA-CNN) to categorize the fall and non-fall incidences and the training and testing phases are depicted in Figure 1 and The complete workings of the mentioned procedures are discoursed in the succeeding subsections.

A. Data Pre-processing

In the starting phase, the characteristics of an image were improved by pre-processing the frames, extracting the noise artefacts, and enhancing particular feature groups. At this phase, 3 important levels were used for preprocessing the frames specifically resizing, augmentation and normalization. The resizing of frames takes place from 150×150 for the reduction of computation cost. Simultaneously, augmentation of frames occurs where they are varied for every training epoch. In order to augment the frames, different models like zooming, rotation, width, height, shifting and horizontal flipping are utilized. Finally, a technique called normalization was executed for the enhanced model generalization.

B. Feature Extraction

The CNN is usually composed of a convolutional layer, a fully connected layer (FC) and a pooling layer. Firstly, feature extraction is done using multiple convolution layers and pooling layers. Later, the complete feature maps from the final convolution layer are transformed into 1D vector which is given as input to the fully connected layers. At last, the final layer classifies the given input images. The neural network updates the weight parameters utilizing the back propagation method which results in the minimization of the squared variance between the true classification results and predicted output of the model. The neurons of all the layers of a network are arranged in 3D, namely height, depth, and width, where the depth denotes the number of channels of the input image also called the input feature mappings, while the size of the neuron is represented by height and width. A convolutional layer consists of convolution filters, which performs the function of extracting all the necessary features from the input image using the mathematical operation called the process of convolution. The current layer's convolution filter performs convolution using the input feature maps for extraction of necessary features and acquires the resulting output feature maps. This is followed by acquiring the nonlinear feature maps for which an activation function is used. The pooling layer, which is also called, subsampling layer, performs the process of down-sampling, with a particular value as the outcome in certain spaces of the feature maps. After the elimination of trivial negligible information from the feature map, it is provided as the input of the subsequent layer and is reduced in size, hence aiding in the minimization of the computation cost [12]. The general pooling process consists of maximal pooling and average pooling. The framework is clear while in the pooling & convolutional layers which has the ability to ameliorate the robustness of the neural network. There is a feasibility of expanding the CNN with convolutions of multiple layers. When the amount of layers is increased, the resulting feature map obtained by the learning process becomes global. In due course, the acquired global feature mappings are transformed into a 1D vector so that it could be processed into the fully connected layer. The FC layer consists of all the variables of the neural network.

Since models like VGGNet & AlexNet have an increasing number of variables, the SqueezeNet network has been proposed which is known for its least number of variables while also preserving good accuracy. The elementary model in SqueezeNet is called the fire model, whose organization is represented in Fig. 2. This model has been divided into two frameworks called Expand & Squeeze. A technique called cross channel pooling is also considered where a multi-layer perceptron is similar to the series of cross channel pooling layers following the conventional layer, which aids in acquiring a linear incorporation of various different feature maps and integration of data into the channels. The convolution kernel variable turns out larger with an increased amount of output & input channels. These include the application of 1×1 convolution in every foundation module and the amount of input channels is deducted which results in reduced number of the convolution kernel variables and hence the complications in the operation is also minimized. A 1×1 convolution is introduced in order to ameliorate the amount of channels which enhances the quality of feature extraction. A huge activation graph is provided to the convolutional layer, if the process of sampling reduction is hampered. The huge activation graph preserves any supplementary data that could possibly provide larger accuracy in classification [13-15].

C. Optimization using Slime Mould Algorithm

The proposed model uses a SMA to perform the optimization of hyper parameters. [16], which has been put forward because of the nature of the oscillation style of a slime mould. The feedback based on positive values and negative values of the slime mould propagation are restored using the mathematical model contained in the SMA. [17]. The SMA could be consolidated with multiple problems regarding optimization which also involves engineering problems. There are two major phases of the SMA algorithm that are called approaching food and warp food [18]

i. Approaching food

This is the phase where the slime is proceeding toward food depending on its odour present in the air. This process is mathematically defined as follows.

$$\begin{cases} \overrightarrow{x_b}(t) + \overrightarrow{v_b} \times \left(W \times \overrightarrow{X_A}(t) - \overrightarrow{X_B}(t)\right), r < p \\ \overrightarrow{vc} \times \overrightarrow{X(t)}, r \geq p \end{cases}$$

$\overrightarrow{vb}$ is a specification ranging from -a to a

$\overrightarrow{vc}$ portrays a specification that reduces from one to zero in a linear manner.

Xb constitutes the present single location with respect to high concentration of odor.

t shows the present iteration.

X showcases the slime moulds locality.

XA and XB are arbitrarily picked singulars from the mould.

W denotes the slime mould's weight.

The equation for p could be written as: $p = \tan h\,[s(i) - DF]$

Where i ∈1,2, 3,…..n.

$s(i)$ showcases the fitness of $\vec{X}$ [19]

As described already, varies from −a to a, which a could be written as:

$$a = \operatorname{arctanh}\left(\left(-\frac{t}{maxt}\right) + 1\right)$$

The $\overrightarrow{W}$ equation is written as:

$$\overline{W(SmeelIndex(t)} = \begin{cases} 1 + r \times \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), condition \\ 1 - r \times \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), others \end{cases}$$

SmeelIndex = sort(S)

Where, r represents the arbitrary value from the interval [0, 1].

bF showcases the maximum fitness acquired from the present process of iteration.

wF showcases minimum fitness value acquired from the present process of iteration.

Smeel Index mentions the series of fitness values. [20]

Warp food

This phase shows the performance of the slime in executing contraction of its venous structure and it is mathematically written as:

$$\overrightarrow{X^{*}} = \begin{cases} rand \times (UB - LB) + LB.rand \ < z \\ \overrightarrow{X(t)} + \overrightarrow{vb} \ \times \left(\overrightarrow{W} \times \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), \ r < p \\ \overrightarrow{vc} \times \overrightarrow{X(t)}, r \geq p \end{cases}$$

Where UB and LB represents the upper border and lower border in the range of search, and the arbitrary specifications ranging from 0 to 1. [21]

D. SMO-CNN Model for Fall Detection

Following the optimization phase, the classification stage is carried out by Variational autoencoder combined with slime mould algorithm, the SMA -CNN model is utilized to determine the input video frames' class labels, such as non-fall or fall event.

A neural network approximates the posteriors of the VAE, which is an oriented probabilistic graphical network. Let $\vec{X}$ stand for the input data, $\vec{x}$(i) for the samples of $\vec{X}$, and $\vec{Z}$ for the low-dimensional latent representation of $\vec{X}$.

The CNN that computes an approximate posterior $q_{\emptyset}(\vec{X})$and a decoder network that computes $p_{\emptyset}(\vec{Z})$, where $p_{\emptyset}(\vec{Z})$ signifies prior distribution through which z has been obtained. Maximizing the evidence lower bound (ELBO) function yields the model parameters $\emptyset$ and θ.

$$L_{\beta}(\Theta, \phi; \vec{x}^{i}) = -E_{q_{\emptyset}\left(\overrightarrow{\vec{x}^{i}}\right)}[log(p_{\emptyset}(\vec{Z}))\text{-}\ D_{KL}(q_{\emptyset}\left(\overrightarrow{\vec{x}^{i}}\right)\|\ p_{\emptyset}(\vec{Z})$$

Its reconstruction loss is represented by 1st term (log-likelihood), and regularizer is represented by 2nd term (KL divergence). The Stochastic Gradient Variational Bayes cost is calculated utilizing empirical estimates with expectation.

$$L_{\beta}(\Theta, \phi; \vec{x}^{i}) \approx \frac{1}{S}\sum_{j=1}^{S} \quad [log(p_{\emptyset}(\vec{Z})) - \ D_{KL}(q_{\emptyset}\left(\overrightarrow{\vec{x}^{i}}\right))\|\ p_{\emptyset}(\vec{Z})$$

Where S denotes the amount of data extracted from $q_{\emptyset}(\vec{X})$). In practise, as lengthy as the minibatch size is large enough, we could select S = 1.

The log likelihood term clarifies to the mean-squared-error if $p_{\emptyset}(\vec{Z})$ is a Gaussian distribution and the network output is the mean of this distribution.

5. RESULTS

The proposed AIoTFD model is evaluated utilizing MCFD and URFD datasets. The first dataset consists of 314 frames of frontal sequence, where 240 frames are under the non-fall category and 74 frames are under the fall category. The second dataset contains 192 videos where 96 videos are included in the fall category and 96 videos are included in non-fall category. The fixing of parameters for the proposed model is mentioned as: Mini batchsize is 200; dropout is 0.5; hidden layers is 3; and hidden units is1024.

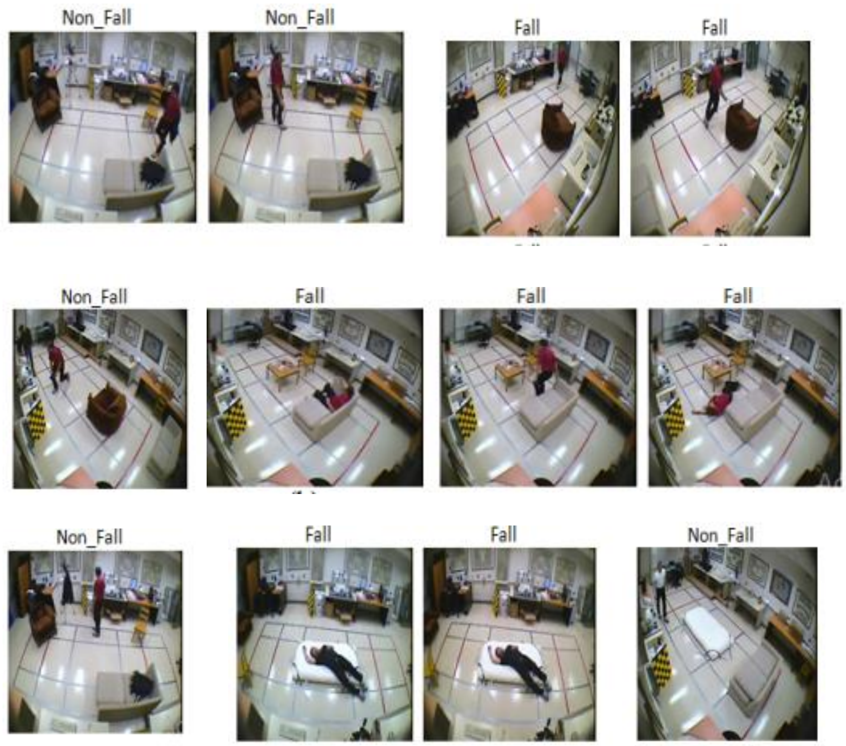

Figure 2. Multiple cameras fall dataset sample images

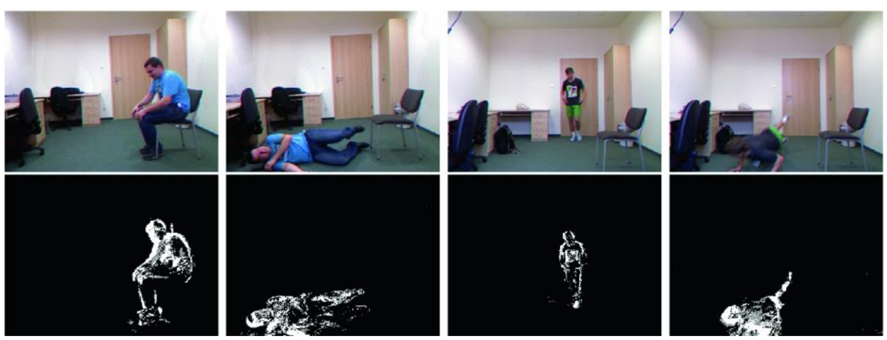

Figure 3. UR fall detection dataset sample test image

Fig. 2 shows the sample data from the MCFD. Besides, the sample test from URFD dataset sample images is illustrated in Fig. 3. The result of the AIOTFD models classification images from the MCFD dataset is described in Table 1. which consists of divergent training sizes (TS). The AIOTFD model performed with a precision of 99.57%, specificity of 99.00%, accuracy of 99.45%, recall of 99.78%, and F-score of 99.39% with TS of 40%:60%. Furthermore, the AIOTFD prototype acquired a precision of 99.41%, specificity of 99.36%, accuracy of 99.75%, recall of 99.89%, and F-score of 99.53% with TS of 60%:40%. Moreover, the AIOTFD technique acquired a precision of 99.72%, specificity of 99.57%, accuracy of 99.91%, recall of 99.80%, and F-score of 99.95% with the TS of 80%- 20%.

Fig. 4 demonstrates the AIOTFD models ROC examination on the MCFD dataset with varied TS. It is certain from the image that the AIOTFD model was able to achieve successful detection of fall, executed with the maximum values of ROC for every corresponding TS. The outcome of classification analysis of the AIOTFD model on URFD dataset is described in Table 2. for different TS. The AIOTFD methodology performed with a precision of 99.91%, specificity of 99.02%, accuracy of 99.45%, precision of 99.91%, and F-score of 99.31% with a TS of 40% - 60%.

TABLE 1: Performance evaluation of AIOTFD model using multiple cameras fall detection dataset

Training Size	**Precision**	**Specificity**	**Accuracy**	**F-score**	**Recall**
40%:60%	99.57	99.00	99.45	99.39	99.78
50%:50%	99.34	99.25	99.54	99.42	99.81
60%:40%	99.41	99.36	99.75	99.53	99.89
70%:30%	99.68	99.42	99.88	99.87	99.92
80%:20%	99.72	99.57	99.91	99.95	99.80
Average	99.54	99.32	99.71	99.63	99.84

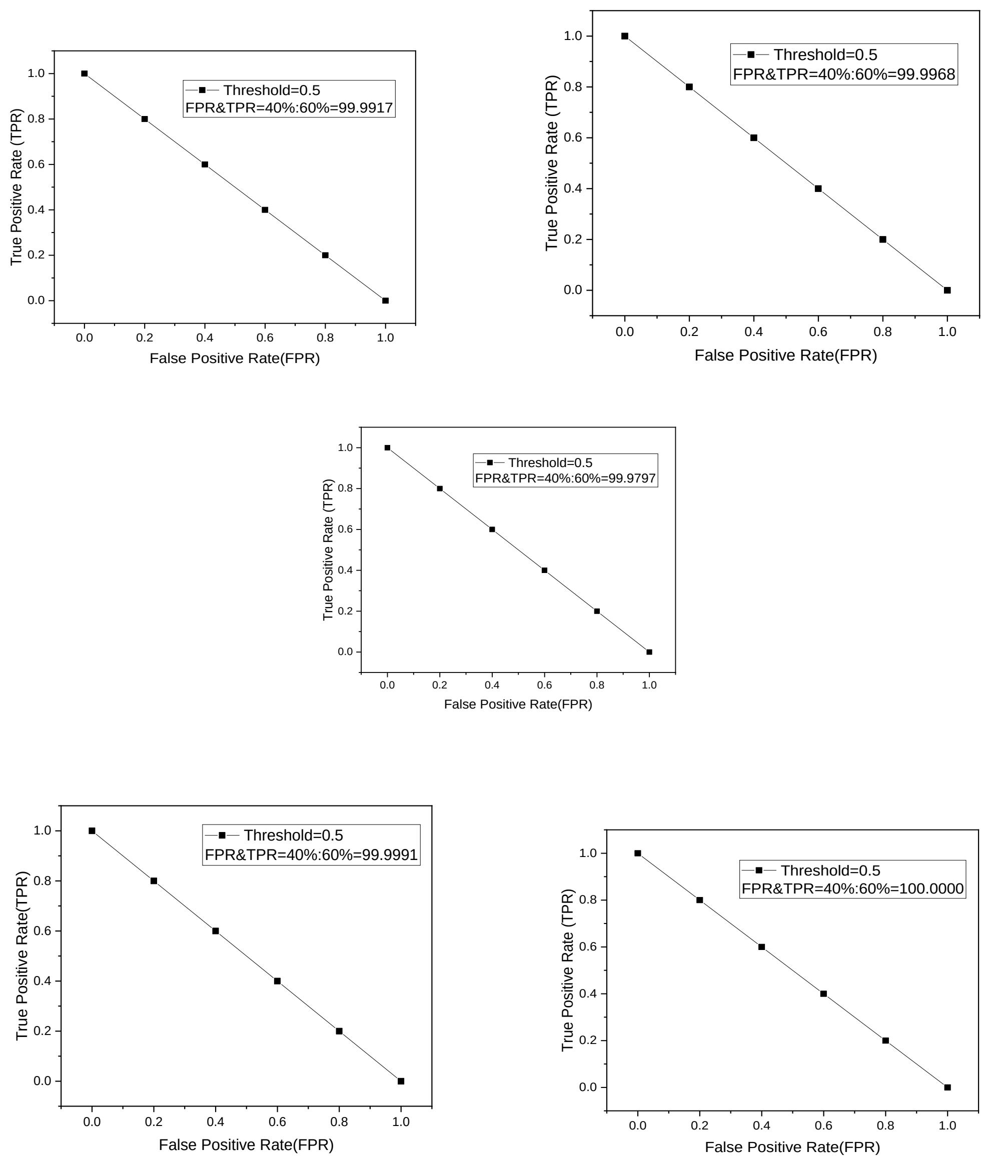

Figure 4. Illustrates ROC Analysis of the Proposed AIOTFD on the MCFD Dataset.

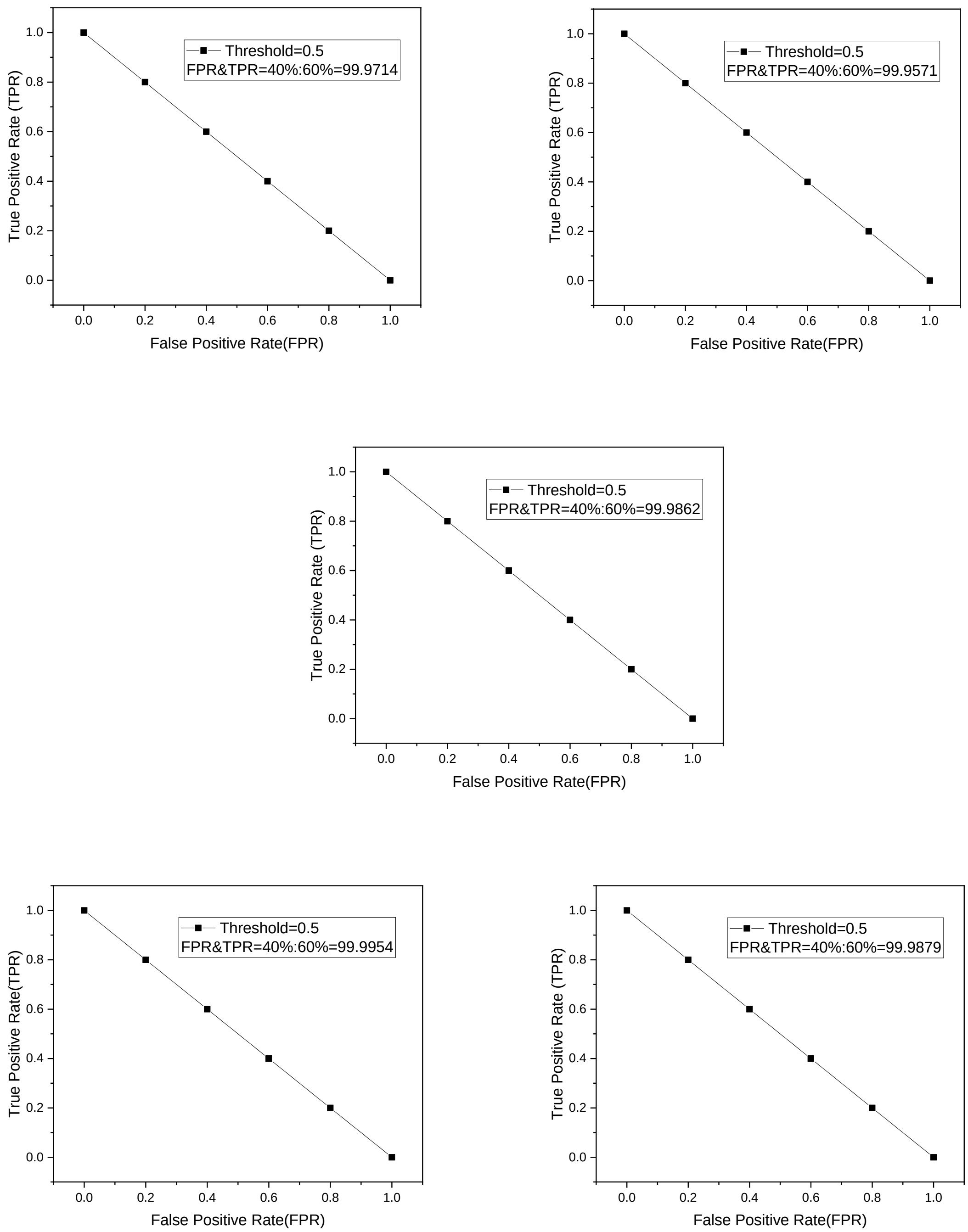

Figure 5. Illustrates ROC Analysis of the Proposed AIOTFD on the URFD Dataset.

TABLE 2: Performance evaluation of AIOTFD model using UR fall detection dataset.

Training Size	Precision	Specificity	Accuracy	F-score	Recall
40%:60%	99.57	99.00	99.45	99.39	99.78
50%:50%	99.34	99.25	99.54	99.42	99.81
60%:40%	99.41	99.36	99.75	99.53	99.89
70%:30%	99.68	99.42	99.88	99.87	99.92
80%:20%	99.72	99.57	99.91	99.95	99.80
Average	99.54	99.32	99.71	99.63	99.84

The AIOTFD model accomplished a precision of 99.76%, specificity of 99.70%, accuracy of 99.58%, recall of 99.54%, and F-score of 99.36% with the TS of 60%-40%,. Also,the AIOTFD has performed with a precision of 99.47%, specificity of 99.81%, accuracy of 99.82%, recall of 99.51%, and F-score of 99.61% with the TS of 80%-20%.

Fig. 5 demonstrates the AIOTFD models ROC examination with the URFD Dataset in varying TS. It can be inferred from the figure that the AIOTFD methodology achieved efficient detection of fall, executed with the maximal values of ROC for every corresponding TS. A short comparative research of the AIOTFD model was performed on other generally used techniques with MCFD and is described in Table 3. It is understood from the figure that 1D-CNN and 2D-CNN models have obtained results that are less accurate that are 94.54% and 95.61% correspondingly. Similarly, the ResNet-50 and ResNet-101 seem to have acquired averagely adequate outputs whose accuracy are 96.67% and 96.84% correspondingly. Following that, the Depth wise, VGG-16, and VGG-19 models seem to have moderate outputs whose accuracy are 98.00%, 97.76%, and 97.98% accordingly. Lastly, the AIOTFD model that has been proposed demonstrated dominating outcomes whose accuracy was found to be 99.82%.

TABLE 3: - Comparison of AIOTFD model's accuracy with other well performed models based on MCFD dataset.

Method	Accuracy (%)
VGG-16	97.76
VGG-19	97.98
Depthwise model	98.00
1D CNN	94.54
2D CNN	95.61
ResNet-50	96.67
ResNet-101	96.84
Proposed AIOTFD	**99.82**

Analytical study of the AIOTFD technique of many other state-of-art methods was performed with the URFD dataset and it is described in Table 4. It can be inferred, from the figure, that the 1D-CNN and 2D-CNN methods demonstrated very poor results whose accuracy was found to be 92.56% and 94.53% respectively. Following that, the ResNet-50 and ResNet-101 have attained a moderate outcome whose accuracy was 95.17% and 96.34%respectively. Similarly, the VGG-16, Depth wise, and VGG-19 models seem to have achieved averagely similar outcomes whose accuracy was found as 97.72%, 98.82%, and 97.79% respectively. In due course, the proposed AIOTFD methodology has proven to be the most efficient and robust since it was found to provide maximum outcomes whose accuracy were 99.79%.

TABLE 4. – Comparison of AIOTFD model's accuracy with other well performed models based on URFD dataset.

Method	Accuracy (%)
VGG-16	97.72
VGG-19	97.79
Depthwise model	98.82
1D-CNN	92.56
2D-CNN	94.53
ResNet-50	95.17
ResNet-101	96.34
Proposed AIOTFD	**99.79**

A large-scale training time and testing time scrutinization of the AIOTFD model with various different generally used methods on Multiple Cameras Fall dataset is described in Table 5. and Fig. 6. It is certain from the outcomes that the VGG-16 model has been found to require the most training and testing time that are 3610.28s and 1645.43s correspondingly. Simultaneously, the VGG-19 model was found to require a minimized training and testing time comparatively that are of 3089.03s and 1472.46s correspondingly. Moreover, the ResNet-101 technique seemed to use average training and testing time that was 1264.13s and 945.4s respectively. Besides, the 2D-CNN and 1D-CNN methods also demonstrated an average testing and training time. Finally, the ResNet-50 method had a comparatively reduced training and testing time that are 1154.35s and 956.87s accordingly. Therefore, the AIOTFD model that has been proposed was found to produce the most efficient results whose training and testing time was found to be 1123.94s and 743.54s correspondingly.

TABLE 5: – On a MCFD, a comparative research was conducted on the training / testing durations of the proposed AIOTFD technique with existing systems.

Method	Training duration (sec)	Testing duration (sec)
VGG-16	3610.28	1645.43
VGG-19	3089.03	1472.46
Depthwise model	2740.18	934.34
1D CNN	2321.87	918.65
2D CNN	1900.44	843.65
ResNet-50	1154.43	956.87
ResNet-101	1264.13	945.4
Proposed AIOTFD	**1123.94**	**743.54**

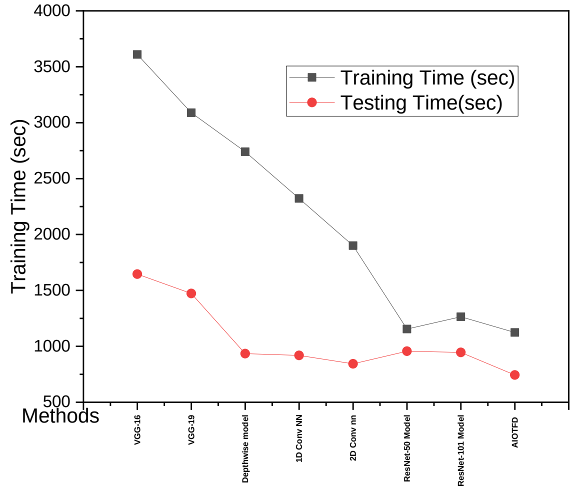

Figure 6. On the MCFD dataset, a comparative evaluation of the AIOTFD model was performed.

Another large scale examination on the training and testing time of the AIOTFD model was performed along with several other generally used methods with the URFD Dataset and it is showcased in Table 6. and Fig. 7. It could be inferred from the result that it was the VGG-16 method which needed the maximum training and testing that are 2345.62s and 1112.78s respectively. Meanwhile, the VGG-19 technique was found to require a slightly reduced training and testing time that are 2675.43s and 1285.24s accordingly.

TABLE 6: In terms of training and testing time, a comparative analysis of AIOTFD method with current models on the URFD dataset was conducted. On the MCFD dataset, a comparison study of the AIOTFD model was performed.

Methods	Training duration (sec)	Testing duration (sec)
VGG-16	2345.62	1112.78
VGG-19	2675.43	1285.24
Depthwise model	1263.23	763.41
1D CNN	1197.34	801.00
2D CNN	1301.87	756.98
ResNet-50	1414.78	845.00
ResNet-101	1556.32	933.17
Proposed AIOTFD	**1043.00**	**667.24**

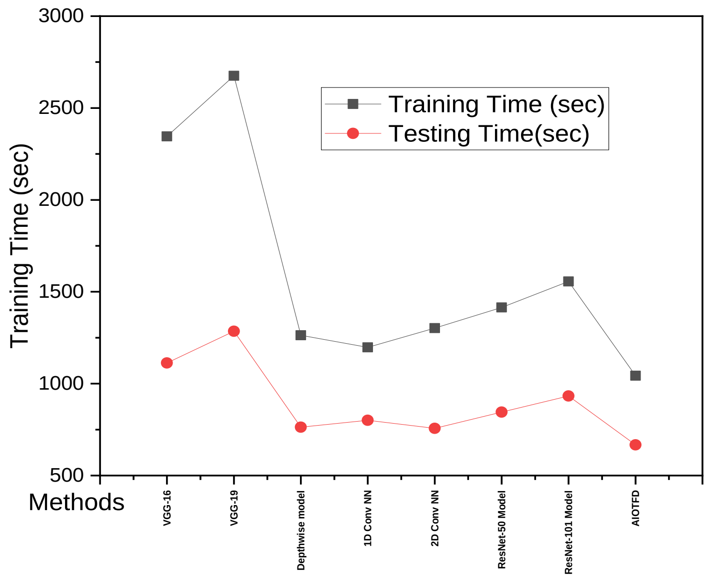

Figure 7. On the URFD Dataset, a comparative evaluation of the AIOTFD model was performed.

Similarly, the ResNet-101 model provided outcomes that required an average training and testing time that are 1556.32s and 933.17s respectively. Meanwhile, the ResNet-50 and 2D CNN techniques demonstrated a more similarly moderate testing and training time. Lastly, the Depthwise methodology was found to require a comparatively minimized training and testing time that are 1263.23s and 763.41s correspondingly. Therefore, the presented AIOTFD model has proven to be the most efficient with its outcomes whose training and testing time are 1043.00s and 667.24s accordingly. The experimental results proved that the execution of the AIOTFD model compared to other novel methodologies were promising and more robust with the accuracy of 99.82% and 99.79% on the MCFD and URFD dataset. This is due to the fact that the proposed model involves the application of SqueezeNet model and SMA for data optimization.

6.CONCLUSION

The proposed AIOTFD model was designed to develop an intelligent elderly monitoring system to detect and monitor fall events in elderly people. The proposed AIOTFD model uses artificial intelligence and IoT systems to recognize collapses in household. The various procedures like Data acquisition, preprocessing, feature extraction performed by SqueezeNet, hyper parameter tuning by slime mould algorithm, and classification using SMA-CNN are all performed by AIOTFD model. The use of SMA to choose SqueezeNet model's hyper parameters and modify parameters of the CNN model significantly improves overall fall detection performance. A large number of simulations are executed using URFD and MCFD dataset. Also the analysis have been made for existing method with proposed AIOTFD which results with an accuracy of 99.82% and 99.79%.

References

1. W. K. A. Hasan; A. M. Abood and M. Habbal, "A Review of Blockchain-based on IoT applications (challenges and future research directions)," 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), 2020, pp. 1-7, doi: 10.1109/CITISIA50690.2020.9371814.
2. T. F. Bernadus; L. B. Subekti and Y. Bandung, IoT-Based Fall Detection and Heart Rate Monitoring System for Elderly Care," 2019 International Conference on ICT for Smart Society (ICISS), 2019, pp. 1-6, doi: 10.1109/ICISS48059.2019.8969845.

3. https://www.igor-tech.com/news-and-insights/articles/iot-in-healthcare-enhancing-medical-environments-with-innovative-solutions
4. https://vilmate.com/blog/why-use-ai-enabled-iot-in-healthcare/
5. T. Vaiyapuri; E. Laxmi Lydia; M. Y. Sikkandar; V. G. Díaz, I. V. Pustokhina and D. A. Pustokhin, Internet of Things and Deep Learning Enabled Elderly Fall Detection Model for Smart Homecare, *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3094243.
6. M. J. A. Nahian et al., Towards an Accelerometer-Based Elderly Fall Detection System Using Cross-Disciplinary Time Series Features, *IEEE Access*, vol. 9, pp. 39413-39431, 2021, doi: 10.1109/ACCESS.2021.3056441.
7. Mahendran RK; Velusamy P; Ramadoss P; Shanmugapriyan J ;Pandian P, An efficient priority-based convolutional auto-encoder approach for electrocardiogram signal compression in Internet of Things based healthcare system, *Trans Emerging Tel Tech*,2021;32: e4115. https://doi.org/10.1002/ett.4115.
8. Mrozek, D., Koczur, A. and Małysiak-Mrozek, B., Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge. *Information Sciences*, 537,2020 pp.132-147.
9. Mahendran, R. K. and Velusamy, P., A secure fuzzy extractor based biometric key authentication scheme for body sensor network in Internet of Medical Things, Computer Communications, 2020,153, 545-552.
10. Ogawa, Y. and Naito, K., Fall detection scheme based on temperature distribution with IR array sensor. In 2020 IEEE International Conference on Consumer Electronics (ICCE) ,2020, (pp. 1-5). IEEE
11. Nooruddin, Sheikh, Md Milon Islam, and Falguni Ahmed Sharna, An IoT based device-type invariant fall detection system, *Internet of Things* ,2020, 100130.
12. Xu, G., Deep convolutional neural network to detect JUNIWARD, In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, pp. 67-73.
13. B. A. Y. Alqaralleh, S. N. Mohanty, D. Gupta, A. Khanna, K. Shankar and T. Vaiyapuri, "Reliable Multi-Object Tracking Model Using Deep Learning and Energy Efficient Wireless Multimedia Sensor Networks," in *IEEE Access*, 2020, vol. 8, pp. 213426-213436, doi: 10.1109/ACCESS.2020.3039695.
14. A. Rajagopal et al., "Fine-Tuned Residual Network-Based Features with Latent Variable Support Vector Machine-Based Optimal Scene Classification Model for Unmanned Aerial Vehicles," in *IEEE Access*, vol. 8, pp. 118396-118404, 2020, doi: 10.1109/ACCESS.2020.3004233.
15. V. Porkodi et al., Resource Provisioning for Cyber–Physical–Social System in Cloud-Fog-Edge Computing Using Optimal Flower Pollination Algorithm, *IEEE Access*, vol. 8, pp. 105311-105319, 2020, doi: 10.1109/ACCESS.2020.2999734.
16. Li, Shimin; Chen, Huiling; Wang, Mingjing; Heidari, Ali Asghar; Mirjalili, Seyedali , Slime mould algorithm: A new method for stochastic optimization, *Future Generation Computer Systems* : 300–323. doi:10.1016/j.future.2020.03.055. ISSN 0167-739X.
17. Patino-Ramirez, Fernando; Boussard, Aurèle; Arson, Chloé; Dussutour, Audrey, Substrate composition directs slime molds behavior, *Scientific Reports*, 9 (1): 15444. doi:10.1038/s41598-019-50872-z. ISSN 2045-2322.
18. Zubaidi, Salah L.; Abdulkareem, Iqbal H.; Hashim, Khalid S.; Al-Bugharbee, Hussein; Ridha, Hussein Mohammed; Gharghan, Sadik Kamel; Al-Qaim, Fuod F.; Muradov, Magomed et al., Hybridised Artificial Neural Network Model with Slime Mould Algorithm: A Novel Methodology for Prediction of Urban Stochastic Water Demand, Water, 12 (10): 2692. doi:10.3390/w12102692.
19. Kumar, C.; Raj, T. Dharma; Premkumar, M.; Raj, T. Dhanesh, T. Dhanesh, A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters, *Optik*, 223: 165277. doi:10.1016/j.ijleo.2020.165277. ISSN 0030-4026.
20. Mostafa, Manar; Rezk, Hegazy; Aly, Mokhtar; Ahmed, Emad M.,A new strategy based on slime mould algorithm to extract the optimal model parameters of solar PV panel, *Sustainable Energy Technologies and Assessments*, 42: 100849. doi:10.1016/j.seta.2020.100849. ISSN 2213-1388.