Hindawi
Advances in Human-Computer Interaction
Volume 2021, Article ID 6483003, 19 pages
https://doi.org/10.1155/2021/6483003

Research Article

Edge Artificial Intelligence: Real-Time Noninvasive Technique for Vital Signs of Myocardial Infarction Recognition Using Jetson Nano

H. M. Mohan,[1] **S. Anitha**,[2] **Rifai Chai**,[3] **and Sai Ho Ling**[4]

[1]*R/S, Department of ECE, ACS College of Engineering, Visvesvaraya Technological University, Belagavi, India*
[2]*Department of ECE, ACS College of Engineering, Bangalore, India*
[3]*School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia*
[4]*School of Biomedical Engineering, University of Technology Sydney, Ultimo, Australia*

Correspondence should be addressed to H. M. Mohan; mohanhm@gmail.com

Received 5 May 2021; Revised 17 July 2021; Accepted 26 July 2021; Published 4 August 2021

Academic Editor: Christos Troussas

The history of medicine shows that myocardial infarction is one of the significant causes of death in humans. The rapid evolution in autonomous technologies, the rise of computer vision, and edge computing offers intriguing possibilities in healthcare monitoring systems. The major motivation of the work is to improve the survival rate during a cardiac arrest through an automatic emergency recognition system under ambient intelligence. We present a novel approach to chest pain and fall posture-based vital sign detection using an intelligence surveillance camera to address the emergency during myocardial infarction. A real-time embedded solution persuaded from "edge AI" is implemented using the state-of-the-art convolution neural networks: single shot detector Inception V2, single shot detector MobileNet V2, and Internet of Things embedded GPU platform NVIDIA's Jetson Nano. The deep learning algorithm is implemented for 3000 indoor color image datasets: Nanyang Technological University Red Blue Green and Depth, NTU RGB + D dataset, and private RMS dataset. The research mainly pivots on two key factors in creating and training a CNN model to detect the vital signs and evaluate its performance metrics. We propose a model, which is cost-effective and consumes low power for onboard detection of vital signs of myocardial infarction and evaluated the metrics to achieve a mean average precision of 76.4% and an average recall of 80%.

1. Introduction

One of the prime factors for sudden death worldwide is ischemic heart disease, and angina pectoris (chest pain) is its most common symptom [1]. The earliest detection information related to signs and symptoms of myocardial infarction (MI), and immediately calling emergency services are the main initiative steps needed to prevent the life risks. From the onset of symptoms of MI, it is crucial to reach the hospital within the first one hour. In medical history, few critical factors related to heart attack symptoms have been explored, such as people with stroke, heart attack history, diabetes mellitus, and high cholesterol, and high blood pressure [2]. The research study highlights common symptoms experienced during MI, wherein the chest pain factor is the highest with 84% and shortness of breath, neck/back pain, arm pain, dizziness, and sweating accounting for other factors [3]. Chest pain is the evident clinical marker of myocardial ischemia in the acute phase of a suspected acute MI [4]. A person encountering early warnings and experiences more signs and symptoms of MI has to seek immediate investigation and treatment by the doctor to avoid life risks. Considering the relationship between the duration of pain and mortality rate, patients with prolonged pain duration had the highest mortality rate [5]. In recent days, computer-based health monitoring system investigations during cardiac arrest mainly depend on investigation reports such as electrocardiogram (ECG), echocardiogram, creatine kinase blood test, creatine kinase MB activity and mass concentration, myoglobin, and cardiac troponin T [6]. The critical patient report will be transferred to a cardiologist to diagnose and provide treatment either with conservative or surgical management.

A person experiencing substernal chest pain (angina pectoris) responds by holding his clenched fist over the sternum, and it is termed Levine's sign. The clenched fist sign or palm sign/Cossio–Levine sign is dominantly found as a symptom in patients experiencing MI and angina pectoris [7]. A study conducted on body language of chest pain patients at the coronary care unit (CCU) suggests that the majority of respondents had a clenched fist to the center of the sternum, flat hand to the center of the sternum, and both flat hands drawn from the center of the chest outwards, and 68% of the participants were considered to be cardiac. The hand movements of chest pain patients have greater importance in a clinical context [8]. In the initial diagnosis of myocardial ischemia, the patient's Levine sign is the most important to medical practitioners [7]. The broader area of chest pain and discomfort corresponds to a greater prospect of cardiac ischemia or MI [9]. The literature survey reports that the chances of heart attack cases increase with an increase in age in elderly people [10].

In the public health monitoring system, cardiac fall detection is a major challenge and crucial for addressing the emergency for improving the survival rate. A robust, reliable, secure, and highly accurate automatic fall detection system can offer medical assistance to older adults and cardiac patients. From the literature survey, there are various risk factors highlighted in fall detection approaches that can be categorized into environmental, physical, and psychological principles, as shown in Figure 1 [11]. Exposure to environmental hazards, loss of vitality in the human body, and psychological factors that might alter a person's cognition are amongst the predominant factors for fall-related events [11]. The other research survey highlighted the risk factors of falling and classified them as intrinsic (e.g., physical weakness, visual impairment, and loss of consciousness) or extrinsic (e.g., usage of multiple medications and psychotropic medication) and environmental hazards (e.g., low lighting conditions and obstacles) [12]. The psychological factors associated with elderly cardiac patients, such as fear of fall (FoF) syndrome and cardiophobia, are often neglected [13, 14]. The prevalence of falls due to cardiovascular disorders remains largely unknown [15]. This work helps in assisting and/or monitoring the patient's physical and psychological health through a camera vision-based approach.

In recent times, the conflation of deep learning approaches, effective Internet of Things (IoT) architectures, and edge computing platforms are exploited for solving real-time remote monitoring applications in the healthcare domain. Artificial intelligence (AI) researchers have paid significant attention to decreasing number of parameters in the deep neural networks (DNN), thus reducing the computational burden, achieving low latency and memory, thereby preserving maximum accuracy for edge AI applications [16].

Several important works have been carried out in the medical care domain based on edge platforms. A deep learning and edge-cloud computing framework for voice disorder detection and classification is developed [17]. Queralta et al. implemented advanced architecture of low power wide area network (LPWAN) technology along with IoT and deep learning algorithms to enhance the quality of remote health monitoring service. The effectiveness of this architecture is exploited by implementing a fall detection technique using recurrent neural networks (RNN), and the data processing and compression is performed via the edge-fog computing platform [18]. The authors proposed a combination of mobile health (mHealth) platform and a machine learning approach to develop detection and classification for skin cancer health issues. The on-device inference app developed for medical applications intends to lower the latency, improvises the privacy issue, and saves bandwidth [19].

Recently researchers have developed various kinds of object detection algorithms in the computer vision domain with real-time solutions implemented using embedded platforms such as Raspberry Pi 4, Nvidia Jetson TX1, TX2, Nano, and Jetson AGX Xavier. Considering DNN-based computer vision tasks, specific hardware design concerns about energy conservation [20]. Mazzia et al. proposed a novel approach for apple detection with trained dataset apple imaged using modified You Only Look Once (YOLO) v3—a tiny algorithm with embedded platforms: NVIDIA's Jetson Nano, Jetson AGX Xavier, and Raspberry Pi 3. The performance metrics have been evaluated for apple detection for real-time positions with various background factors, and the study emphasized that the technique could be employed on uncrewed ground vehicles for detection with minimal power consumption [21]. Barba-Guaman et al. explained the reliable and more accurate measurement technique for pedestrian and vehicle detection with three pedestrian metrics such as accuracy, processing time, and recall under various environmental conditions with NVIDIA Jetson hardware through a convolution neural network algorithm [22]. Partel et al. interestingly implemented a weed detection system for an intelligent plant sprayer system. A deep learning algorithmic approach was developed to generate weed maps to identify the unwanted weeds. A performance analysis was performed by using two embedded GPUs NVIDIA GTX 1070 Ti and NVIDIA Jetson TX2 [23]. Motivated from this survey, the authors utilize a low cost and high computing facility of the heterogeneous central processing unit (CPU) + GPU SoC system to implement high-quality CNNs, namely, single shot detector (SSD) MobileNet V2 and SSD Inception V2 for our application. In our present work, effort was made to develop a chest pain posture-based human fall model on NVIDIA Jetson Nano development board to detect the vital signs during MI.

The researchers from industries and academia have mainly focused on improvising the performance of CNNs in object detection networks by incorporating new architectural design concepts and enhancing the existing algorithmic approaches. CNN-based structures in object identification methods are mainly categorized into two types: (i) one-stage

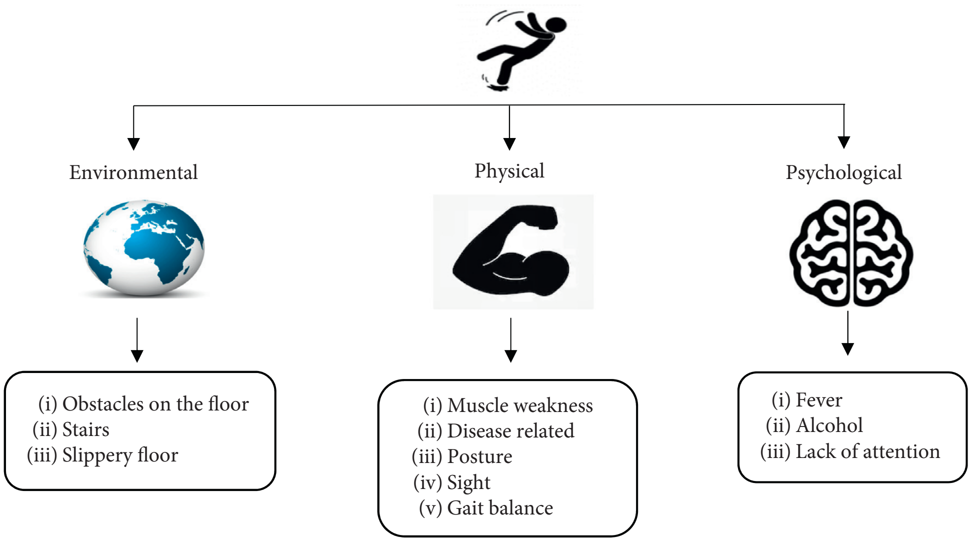


Figure 1: Human falling factors.

learning- and (ii) two-stage learning-based classification schemes. The single-stage method uses a simple regression approach and adopts a full-convolution architecture for object predictions. The main focus in one-stage training method is to produce outputs by classifying the class information and localization of the objects in an input image. Some of the one-stage representation models are SSD, YOLO, squeezeDet, and retinaNet. In a two-level learning method, the predictions are performed at two levels. First, the network proposes a selective search and generates region of interests, wherein the generated regions are sparse. Second, region proposals produced are sent for object classification. Some examples of two-stage networks are region-based convolution neural network (R-CNN), fast RCNN, faster R-CNN, recurrent fully convolution network (R-FCN), and mask RCNN. The accuracy of the bounding box regression of the two-stage model is much superior to the one-stage model. However, owing to its complicated architecture, training time is much higher [24]. The design parameter exploration method increases the energy effectiveness of CNN-based object detection solutions on mobile systems. For example, we can adjust the hyperparameters of the used CNN model and sacrifice accuracy for speed up using dimension reduction techniques and other input level approximation methods. It helps to create a massive space of design parameters for the target CNN-based object detection framework. With this sufficiently large design space, we can evaluate the accuracy and power consumption of different implementations and search for an appropriate design point that yields the highest score measured in mAP/WH [25].

This paper extends our previous work on the noninvasive technique for real-time myocardial infarction detection using faster R-CNN [26]. The disadvantage of the faster-RCNN model is its complex deep learning architecture to deploy on the existing edge embedded devices. To implement our AI vital signs of the MI object detection model on embedded edge devices in real time, certain conditions are required, e.g., high accuracy, fast computation time, small model size, and efficient energy consumption. In this present work, we adopt state-of-the-art CNN lightweight architectures, SSD Inception V2 and MobileNet SSD V2, a highly efficient, memory-efficient network for low-powered GPU devices. Our work exploits transfer learning techniques using the state-of-the-art lightweight pretrained neural networks such as ConvNets SSD MobileNet V2 and SSD Inception V2. These lightweight CNN architectures can predict relatively faster than other algorithms due to their competitive performance by reducing the computational complexity and ease of implementation for enhanced performance on a low power embedded edge device. The main aim of the work is to classify and identify the fall states, considering the vital signs during an emergency of heart attacks such as chest pain, Levine's sign, partial fall, and complete fall posture.

1.1. Motivation. Motivated by the stupendous success of deep learning, research organizations are investigating applications related to biomedical image and video analysis. Despite the rapid advancements in low power edge devices, minimal works are carried out in the healthcare segment using artificial intelligence and low-powered GPUs, and less importance is provided in the related research works to find out the vital signs of MI. There is a vast scope for exploring the opportunities in noninvasive detection and predicting cardiovascular diseases and signs of a heart attack.

1.2. Contributions. The main contributions of this study are summarized as follows:

(i) Create a private RMS synthetic dataset as vital signs of MI with expert annotation. The dataset produced is of a benchmark quality, which would benefit finding better MI pain elicitation methods.

(ii) Identify, design, and implement a lightweight CNN approach for an intelligent video surveillance system for MI fall events to achieve maximum accuracy with minimum complexity. The neural network performance developed is analyzed with two datasets for the performance metrics such as mean average precision, average recall, F_1 score, and losses.

(iii) We propose a computer vision paradigm technique based on object detection ConvNets: SSD MobileNet V2 and SSD Inception V2 pretrained networks for the emergency analysis during cardiac arrest.

(iv) We develop a novel vital sign MI neural network model characterized by Levine's sign and fall identification implemented in a real-time embedded environment using Jetson Nano, and its run time performance is evaluated.

The remainder of the paper is presented as follows. Section 2 discusses the research background and related work. Section 3 explains the proposed methodology along with the dataset utilized and hardware description. Section 4 illustrates experimental results along with detailed discussion. Section 5 draws a conclusion and future works.

2. Related Work

This section provides basic information on various human fall detection approaches and edge-based AI applications in recent years.

Recently, many researchers have put much effort to develop an accurate and efficient fall detection system with major safety factors in protecting the elderly people. Here, fall detection system classification has been discussed to detect fall incidents among elderly people based on several factors such as inertia-based, context-based, and radio frequency- (RF-) based systems as shown in Figure 2 [27]. Table 1 presents the summary of fall detection techniques organized with the following criteria: datasets utilized, the number of subjects considered as samples, and sensor modalities and algorithms carried out during their work.

2.1. Wearable Device-Based Approaches. A fall detection-based device should adopt robustness and high reliability for real-time scenarios. Wearable technology relies on embedded sensors that detect, analyze, and transmit information for monitoring human activities. To address the issue of human-based fall approaches, several innovative research works have been carried out involving wearable devices with inertial sensors such as accelerometers, a fusion of accelerometer and posture sensors, triaxial accelerometer, gyroscope, etc.

One of the challenging problems of the wearable-based fall approach is to design a low power, highly accurate detector for both indoor and outdoor environments. The authors designed a low computational cost wearable fall detector based on a two-level support vector machine and an online feature extraction method using a 3-axial accelerometer. The machine learning-based system works with multiple sampling frequencies with best accuracy/complexity tradeoff [28]. A sole smart tracker was designed using the concept of differential acceleration and time threshold based on low energy Bluetooth communication [29]. A fall-detection ensemble decision tree (FEDT) algorithm was proposed by Wu et al. for reliable fall detection in practical scenario utilizing mobile cloud computing resources [30]. Huang et al. implemented a novel idea of training free-fall recognition-based hidden Markov model (HMM) named GFall based on geophones. The model developed intended to reduce the false alarm rate using a reconfirmation mechanism called energy of arrival (EoA) positioning for detecting human fall [31]. Tian et al. introduced Aryokee, and frequency-modulated continuous wave radar- (FMCW-) based signal to overcome the limitations of other wearable fall-based approaches. The work tries to address certain practical challenges such as tackling complex falls and sudden nonfall movements, detect falls in other motions, and generalization to the environment and people [32]. Wang et al. designed a device for free-fall detection through a combination of WiFall, wireless network, and ML approaches such as SVM and Random Forest. The system leverages channel state information (CSI) as the criterion uses temporal stability and frequency diversity for human activity and fall detection [33]. The authors presented a novel compression sensing technique and devised a shimmer device for fall and human activity detection mainly to reduce energy consumption. The method explores the advantage of combining two sensors: accelerometer and gyroscope and incorporates compression sensing capability, and final classification is performed using the ML algorithms such as ensemble classifier (EC), SVM, decision tree (DT), and k-nearest neighbor (k-NN) [34]. A fall-based remote healthcare monitoring was designed by employing IoT-architecture-based systems, LPWAN technology, and RNN deep learning algorithm to increase the effectiveness in detection and classification [35]. There are major drawbacks associated with wearable devices such as generation of more false alarms, devices getting disconnected easily, sensitivity to external factors, person forgetting to wear, and inconvenience of wearing it all day long, which makes the system inconsistent with providing highly accurate automatic fall detection.

2.2. Camera- (Vision-) Based Approaches. Vision-based surveillance systems overcome the drawbacks of wearable fall approaches to impart practical and complex frameworks. Han et al. uniquely advocated a two-stream approach to process video data for human fall detection and implemented it using a lightweight CNN VGG network suitable for deploying on mobile phones [36]. Kong et al. put forward computer vision-based fall identification for a single and multicamera video surveillance system. An effective stream CNN approach is presented, wherein motion images are fed for silhouette feature extraction in the first two streams and dynamic images with temporal information fed to the third stream [37]. The authors concentrated on the dynamic and complex outdoor environment for solving universal human detection and fall. A Rao-Blackwellized particle filtering is

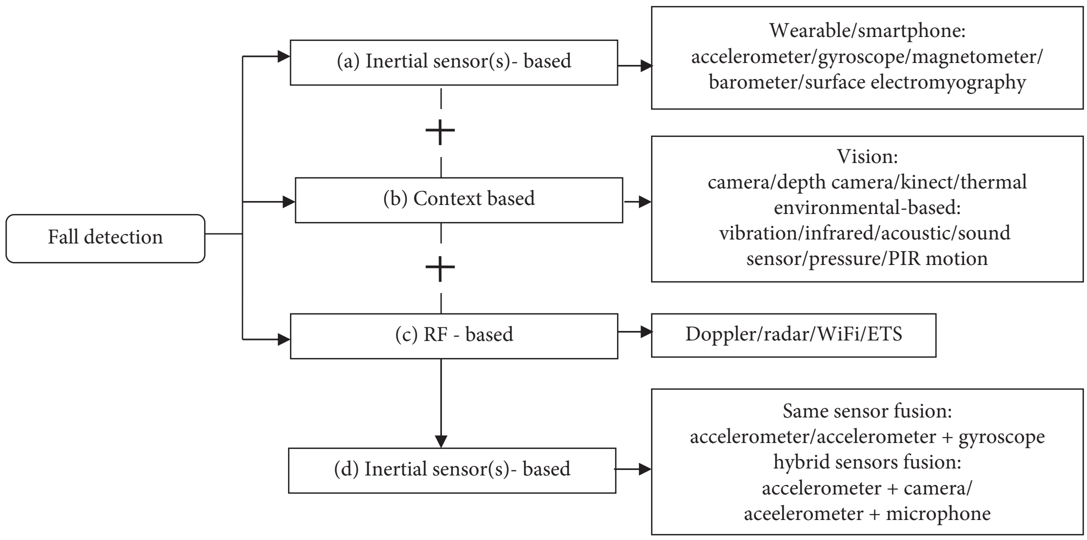


Figure 2: Fall-based methods.

Table 1: Fall detection techniques.

Author	Datasets	No. of subjects (age)	Sensor	Algorithms
Saleh and Jeannès [28]	Simulated	23 (19–30), 15 (60–75)	Accelerometer (waist)	SVM
Zitouni et al. [29]	Simulated	6 (N/A)	Accelerometer (sole)	Threshold
Wu et al. [30]	Public (simulated)	42 (N/A), 36 (N/A)	Accelerometer (chest and thigh)	Decision tree
Huang et al. [31]	Simulated	12 (19–29)	Vibration	HMM
Tian et al. [32]	Simulated	140 (N/A)	FMCW radio	CNN
Wang et al. [33]	Simulated	N/A	WiFi	SVM, Random Forests
Kerdjidj et al. [34]	Simulated	17 (N/A)	Accelerometer, gyroscope	Compressive sensing
Queralta et al. [35]	Public (simulated)	57 (20–47)	Accelerometer, gyroscope, magnetometer	LTSM
Han et al. [36]	Simulated	N/A	Web camera	CNN
Kong et al. [37]	Public	Public	Camera (surveillance)	CNN
Ko et al. [38]	Simulated	N/A	Camera (smartphone)	Rao-Blackwellized particle filtering
Shojaei-Hashemi et al. [39]	Public (simulated)	40 (10–15)	Kinect	LSTM
Min et al. [40]	Public (simulated)	4 (N/A), 11 (22–39)	Kinect	SVM
Ozcan et al. [41]	Simulated	10 (24–31)	Web camera	Relative-entropy-based

utilized for feature extraction from RGB depth images [38]. A deep learning technique of long short-term memory (LSTM) feed-forward neural network for human fall detection is presented using a transfer learning approach and outperformed existing works based on hand-crafted features [39]. A privacy-preserving fall method was proposed, wherein the Kinect sensor 3D skeleton image input was utilized to train the SVM classifier. The method achieved a reduction in the number of parameters compared to the other deep learning approaches with lower time costs [40]. The authors adopt an approach of vision-based activity monitoring with a wearable camera worn by the subject compared to a static camera installed at fixed locations. An improved variant of the histograms of oriented gradients (HOG) is implemented along with gradient local binary patterns (GLBP); an adequate threshold for fall prediction is estimated by Ali–Silvey distance measure [41]. Through extensive literature survey review, it was found that very few works have been carried out based on the noninvasive heart attack detection from color images. Rojas-Albarracín et al. proposed a heart attack detection approach from 1500 RGB color images using a convolutional neural network [42]. As examined from the literature survey, deep learning methods based on advanced convolution networks are more predominantly applied for classification and localization. CNN acts as a backbone in object detection prediction networks by automatically learning salient features. Deep learning techniques use automatic feature extraction to provide more accurate and efficient solutions to tackle the real-time problems in the computer vision domain.

The most popularly used human fall-based publically available datasets have been summarized in Table 2. Information consists of a number of subjects, the age range of participants, the total number of samples, and sensors and their positions in the dataset. Scenario of the data collection zone has been summarized [43].

The sophisticated deep learning-based architectures require heavy computational resources for implementation. Deep learning models can be deployed in centralized cloud computing platforms, which offer significant options for high performance computation. However, some certain challenges and constraints make ML data services between devices and cloud environments impractical such as privacy, financial overheads, latency factor, and energy that affects the performance of the system. Some of these problems can be majorly resolved through edge computing, commonly known as "edge AI" in any computing field whose performance is evaluated locally obtained data from any sensing devices or database. Barba-Guaman implemented a regression approach through YOLO network targeting classification and localization in object identification. Four different models are deployed on single-board computers using deep learning algorithms for object identifications, specifically SSD MobileNet V1, SSD MobileNet V2, Penet, and Multiped, and performance comparisons were carried out [22]. In real-world scenario applications such as drones, autonomous driving, and robotics, there are certain constraints associated, including computational resources, to pursue high accuracy from a limited computational cost. The state-of-the-art computer vision algorithms these years are applied to robotic applications, which are able to achieve better metrics performance such as recognition rate and detection accuracy. The authors Gu highlighted two concepts for nonlinear models to improve the recognition rate and path planning of robotic performance. In tennis ball collection robot using deep learning algorithms, two important tasks were performed: (i) pointer network model to resolve travelling salesman problem as path planning and (ii) YOLO model for real-time object detection. These two concepts were deployed on the NVIDIA Jetson TX1 board for performance evaluation in the optimal path finding in tennis ball [44]. Authors in [45] demonstrated the usage of the YOLOv2 algorithm to develop unmanned aerial vehicle (UAV) utilizing NVIDIA Jetson TX2 for emergency analysis. The author achieved an optimal solution in object detection configurations with parametric resolved using mathematical equations. The experimental analysis is evaluated for detection accuracy, speed of detectors using CPU multiscale aggregated channel feature (ACF) detector, and YOLOv2. YOLOv2 shows better performance than ACF in evaluating the performance metrics such as frame rates and detection accuracy.

3. Proposed Methodology

3.1. Overview of the Proposed Model. In this proposed work, two state-of-the-art convolutional blocks are combined as a single lightweight advanced architecture to implement an object detection task into a computationally intensive GPU embedded device.

3.1.1. Single Shot Detector (SSD). The single shot multibox detector (SSD) architectural model adopts feed-forward convolutional network to achieve exemplary performance in object detection. The network efficiently performs localization and classification of objects in a single forward step. The entire SSD model consists of mainly two segments: the base network, which performs high-quality image feature extraction and the SSD, which evaluates the classification result. In the case of the MobileNetV2 SSD [46] network shown in Figure 3, MobileNetV2 extracts the image features and subsequent convolution layers of SSD perform the classification task. SSD inherits the concept of anchor box strategy and feature pyramid structure from the faster RCNN algorithm to generate default boxes of various aspect ratios and scales followed by a nonmaximum suppression technique to produce the final detections. The performance of a single shot multibox detector (SSD) is measured by scaling down both the model size and the complexity using multiple feature maps in a network to enhance the metrics speed and removing the proposed regions to predict large objects through deeper layers and smaller ones with shallow layers in applications such as mobile and embedded devices [47].

Every prediction of the object in an image originates from a concept of boundary box in SSD. Feature maps of different resolutions are applied to the preprocessed image in a convolutional manner to create overlapped bounding boxes. Several multiresolution boxes called default boxes of different scales and sizes are generated relative to the input image. In every selected feature map, there are f frames that contrast in size and width-to-height proportion. The SSD model manually defines a collection of different aspect ratios for the default boxes and is denoted as $a_r \in \{1, 2, 3, (1/2), (1/3)\}$. The dimensions of width and height for each default box can be computed with the formula $(w_k^a = s_k \sqrt{a_r})$, $(h_k^a = s_k \sqrt{a_r})$, respectively. Score values are evaluated for every box, and the highest score is selected finally as a class for the bounded box. The scale of every default boxes for every feature can be evaluated using the following equation:

$$S_k = S_{\min} + \frac{S_{\max} - S_{\min}}{m - 1}(f - 1), \quad (f \in [1, m]), \tag{1}$$

where m is the total number of feature maps and $S_{\min}$ $S_{\max}$ are the lowest and highest scaling factors to be set, respectively.

3.2. The Proposed Method. Figure 4 shows the methodology of the proposed work. The entire work is divided into two stages as follows: the training stage and the detection stage. The steps for both stages are structured below.

The methodology includes three main stages: (i) input stage, (ii) training stage, and (iii) detection or output stage. The input stage incorporates raw input images from two datasets: custom synthetic dataset (RMS) and public

Table 2: Human fall-based open datasets.

Dataset/year	Sensors	Number of subjects (age)	Total samples	Position of sensing points	Scenario
UP-Fall (2019)	A, C, E, L, IR, G	17 (18–24)	561	H, F, N, Wa, Wr, An	Lab
SisFall (2017)	A, G	38 (19–75)	4505	Wa	Gym, hall
UniMiB SHAR (2017)	A	30 (18–60)	7013	T	N/A
NTU (2016)	K	40 (10–35)	56000	Ce	Lab
UMA Fall (2016)	A, G, M	17 (18–35)	531	An, Ch, T, Wa, Wr	Home
MobiAct (2016)	A, G,O	57 (22–47)	2526	T	Gym, hall
MobiFall (2013)	A, G, O	24 (22–47)	630	T	Gym, hall

Note. N/A: not appropriately defined; C: RGB camera; A: accelerometer; G: gyroscope; O: orientation measurements; K: Kinect sensor; M: magnetometer; IR: infrared sensor; L: luminosity sensor; E: electroencephalography (EEG) headset; Ce: ceiling; T: thigh (pocket); Wa: waist; Wr: wrist; An: ankle; Ch: chest; H: head; N: neck; F: floor.

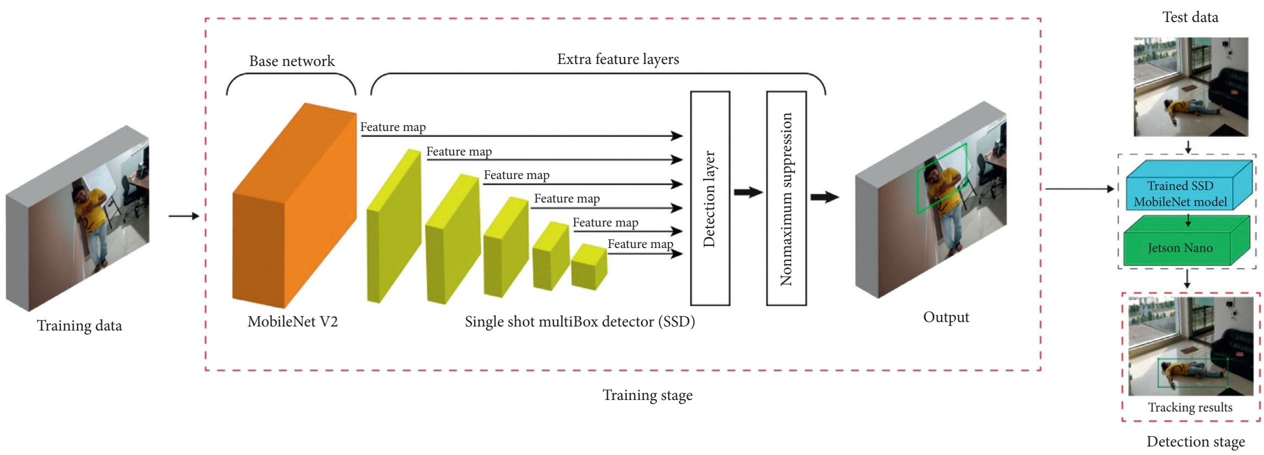


Figure 3: SSD MobileNet V2 architecture.

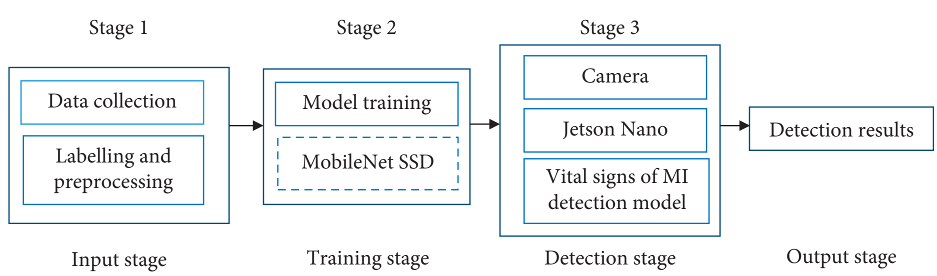
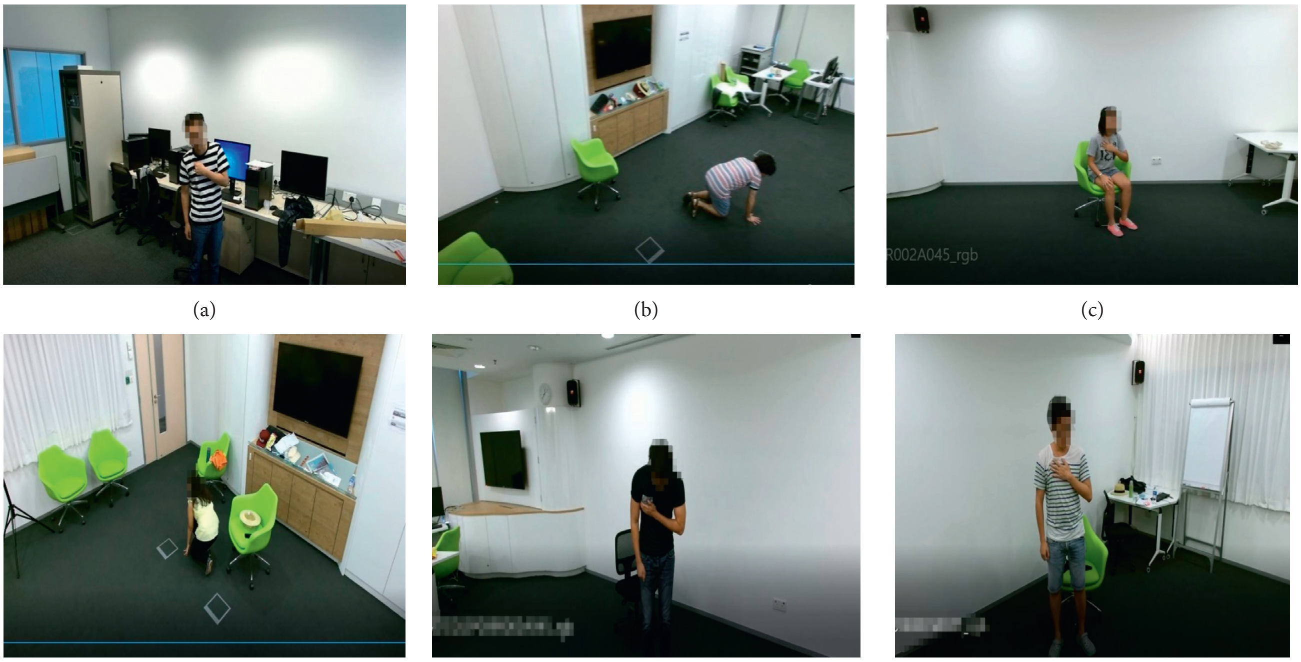


Figure 4: Methodology for training and detecting real-time vital signs of myocardial infarction.

benchmark dataset (NTU RGB + D). Subsequently, the preprocessing technique is carried out to downsize the images for improving the training speed and to avoid overfitting problems. The region of interest (RoI) (chest pain posture and partial and complete fall postures) of every image is marked using the Labellmg software. During the training stage, the custom dataset is trained using the COCO pretrained (TensorFlow Zoo) CNN models, namely, SSD Inception V2 and SSD MobileNet V2. The training of the CNN model is performed on the workstation, and the model has been deployed on the Jetson hardware platform. During the final detection stage, real-time detection is carried out on Jetson Nano board by connecting a single camera, and a trained CNN model is used to detect the vital sign postures of MI as shown in Figure 5.

3.3. Dataset Collection and Preprocessing. During deep learning algorithm implementation, the input dataset quality and the total number of images play a significant role in the final performance of the network. A benchmark standard is followed while collecting data samples from two different sources and classified as follows.

3.3.1. Action Recognition Dataset (NTU RGB + D). The benchmark dataset NTU RGB + D [48] contains about 56,000 video samples and 4 million frames with 60 action classes. Shahroudy et al. highlighted the limitations of most of the currently available RGB + D-based action recognition benchmarks, specifically clear distinction of class labels, lack of training samples, variety of subjects, and proper placement of cameras. Henceforth, the NTU RGB + D database



(a) (b) (c)

(d) (e) (f)

Figure 5: Levine's pose, fall posture images from the NTU RGB + D dataset. (a) Levine's sign standing pose. (b) Partial fall pose. (c) Levine's sign sitting pose. (d) Partial fall pose. (e) Partial fall pose. (f) Levine's sign standing pose.

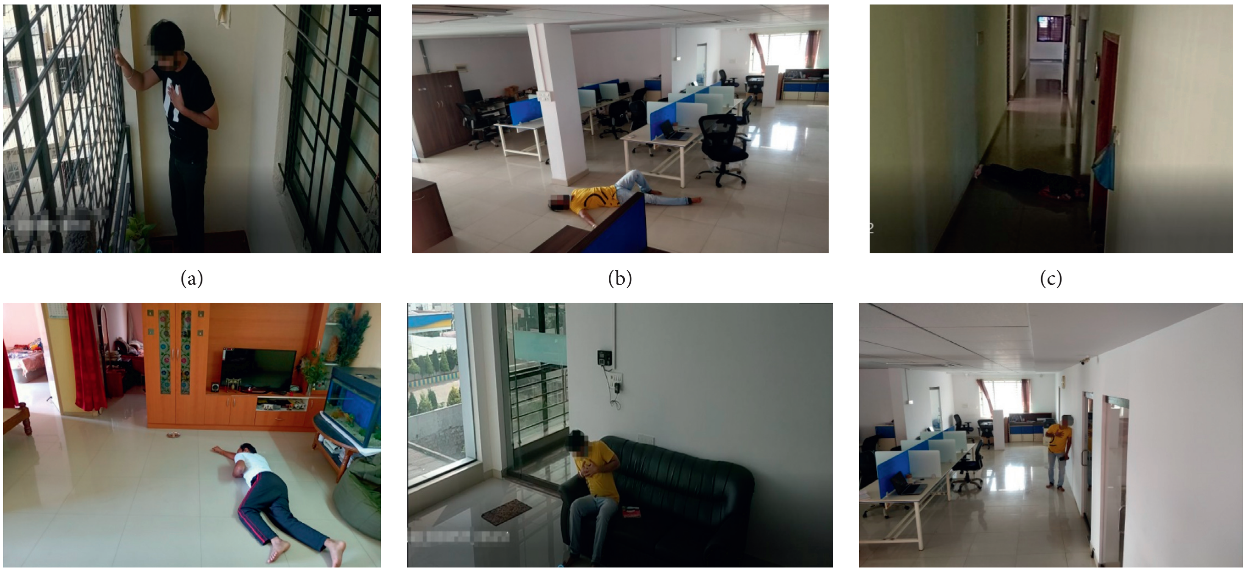
has tried to overcome the limitations. Every frame is captured by a highly variant camera setting located at horizontal angles −45, 0, +45 fixed at the same height. The video frames were captured with three cameras simultaneously. In RGB videos, every frame has been captured with a pixel size of 1920×1080. The dataset consists of three categories: (i) mutual actions, (ii) medical conditions, and (iii) daily routine actions. The present work incorporates the medical condition action category 3D RGB images of chest pain and falling down classes as A45 and A43, respectively, shown in Figure 5.

3.3.2. Private Dataset: RMS. A self-made dataset RMS consists of 3D depth RGB images captured using a closed-circuit television CCTV and OnePlus five smartphone camera by simulating the real-life heart attack chest pain and fall scenarios indoors. The dataset encloses images and videos captured under different chest pain and fall scenarios, such as fall from standing position, sitting on chair posture, walking, sitting on bed position, considering the routine human activities. The dataset consists of 1500 images of resolution 4608×3456 captured under different lighting conditions and recording angles. Figure 6 shows the RGB images of our private RMS dataset. Table 3 highlights the description of the dataset used in our work.

As a preprocessing technique, the images obtained from video frames are scaled down in size to reduce the computational burden. Through the scaling process, the pixel width and height of the images were scaled down to 1067×800 pixels to avoid the quality of original image degradation. The images from both datasets combined were randomly split as train and test set in a ratio of $1:0.3$. The annotation procedure is performed using the Labellmg tool, wherein the ROI of images are selected as shown in Figure 7. The annotation technique is employed for the manual labelling of every training image prior to the training process, and the counterpart XML file format for target box location was generated. In this work, the images are classified into three main classes: (i) chest pain posture, (ii) partial fall, and (iii) complete fall.

3.4. Hardware Description. The present research work includes the concept of edge AI where signal acquisition, the processing, is performed locally on the embedded platform during real-time. The training process is performed on a dedicated workstation since the deep learning model training with larger dataset demands high computational power. Later the trained model is deployed on the target hardware, Nvidia Jetson Nano, for the inference process.

State-of-the-art neural network architectures applied for performing specialized computations require parallelized computing graphical processing unit (GPU). The parallel combination of the computational processing unit (CPU) and GPU is utilized for real world complex applications like object detection to achieve high throughput, accuracy, high bandwidth, etc. To achieve high-performance gaming and high end graphics, applications rendering NVIDIA Corporation developed with added techniques a new computed device architecture and cuda DNN library to enrich system performance. Each one of cuda cores or stream processors acts as subunits of GPUs, which performs the tasks in parallel and independently to accelerate the system

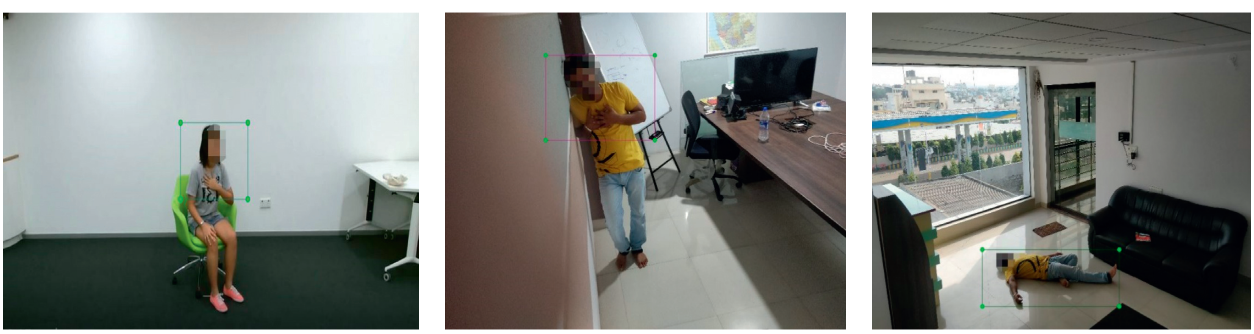


(a) (b) (c) (d) (e) (f)

Figure 6: Levine's sign, fall posture images from the private RMS dataset. (a) Levine's sign standing pose. (b) Complete fall pose at office. (c) Complete fall pose at corridor. (d) Chest pain standing pose. (e) Levine's sign sitting pose. (f) Levine's sign standing pose.

Table 3: Complete dataset consisting of chest pain posture, partial fall, and complete fall.

Dataset	Scenario	Number of images	Resolution after preprocessing
NTU RGB + D	Lab	1500	1240×600
Custom RMS	Home, office, lab	1500	1067×800

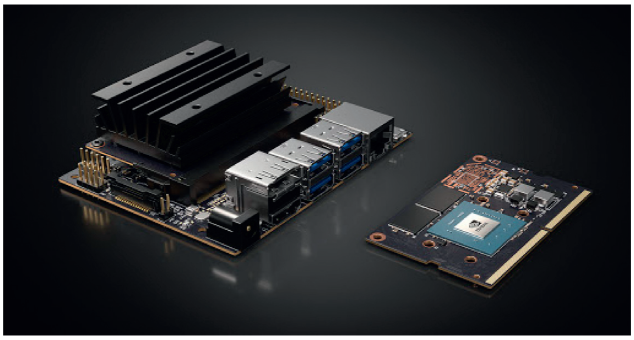


(a) (b) (c)

Figure 7: Region of interest selection for different postures.

performance. The salient features of NVIDIA Jetson Nano are lightweight, low power consumption, and perfectly suited for graphical applications. In general, algorithmic approaches implemented using deep learning techniques to train large data will increase the computational cost and memory bandwidth. To achieve this low cost, a powerful accelerating hardware need to be incorporated; one such hardware device is Jetson Nano NVIDIA. The Jetson device is preconfigured with 2 GB reserved swap memory and 4 GB total RAM memory. We have utilized the entire swap memory for executing the object detection code to avoid out of memory issue. Figure 8(a) shows the Jetson Nano device [49], and Figure 8(b) shows system interfacing.

3.5. Training. Model training is an automatic technique of parameter fitting in the deep learning network. For efficient training of the proposed network, the CNNs SSD Inception V2 and SSD MobileNet V2 used the pretrained weights as the initialization from the COCO dataset [50]. The pretrained networks are being downloaded from the official model Zoo-TensorFlow. Training a complex DNN architecture for a large dataset demands high performance computer system. The training of heavyweight DNNs on a powerful graphical processing unit (GPU) improves the performance and also reduces the training time. In this present research work, we used a computer system with a CPU: Intel(R) core(TM) i7-7700 CPU @3.60 GHz, graphics card: Intel R HD Graphics 630, and 12 GB DDR4 RAM. A



(a)

(b)

Figure 8: (a) Jetson Nano and (b) Jetson NANO board configuration.

dedicated high-performing GPU can be used for training the computationally expensive deep learning models. The object detection dataset consists of three main posture classes: (i) Levine's sign, (ii) partial fall, (iii) complete fall; the total data consist of 3000 training images and 500 test images. The gradient descent optimization function is used to optimize the loss parameter, while the learning rate is set as 0.004 and the batch size as 24. The training took on average one hundred hours for SSD Inception V2 using the TensorFlow framework. The main termination criteria of training considered was the minimum loss function and maximum mean average precision. After training the deep learning model, it is deployed to the Jetson Nano Embedded platform. Figure 8(b) shows the Jetson Nano setup along with the camera for the real-time performance evaluation of the model.

3.6. Performance Evaluation Metrics. COCO evaluation object detection metrics was used to evaluate our CNN model. Bifurcating effectively the various problems of chest pain sign detection and fall detection is a major task in this work. Therefore, the SSD Inception/MobileNet models are incorporated to achieve the detection performance accurately. In this performance evaluation, the key factors are considered as average precision, average recall, F_1 score, losses, and frames per second.

3.6.1. Confusion Matrix (CM). It is applied to outline the accomplishment of a classification model:

(a) True positives (TP) indicate number of both true cases for classifier prediction and the correctness of the class to point out the ground-truth bounding box

(b) False positives (FP) (type I error) indicate the count cases for true classifier prediction, and the false class in correction leads to improper object detection

(c) False negatives (FN) (type II error) indicate the count cases for false classifier prediction, and the true class in correction leads to improper object detection

(d) True negatives (TN): indicate number of both false cases for classifier prediction and correctness of the class to point out the ground-truth bounding box.

It is observed from the literature survey that the true-negative result in object detection provide only marginal information based on a number of bounding boxes in the video analysis. Figure 9 indicates the confusion matrix.

3.6.2. Intersection over Union (IoU). It is a measurement of the overlapping area using two bounding boxes: ground truth box B_p and predicted box B_{gt}.

Finally, IoU can be used to measure the possibility of TP and/or TN cases in the detection process. IoU is defined as the ratio of an intersection area vs overlapped area of both bounding boxes (predicted and ground truth) as depicted in Figure 10. The IoU is measured as follows:

$$\mathrm{IOU} = \frac{\mathrm{Area}\left(B_p \cap B_{gt}\right)}{\mathrm{Area}\left(B_p \cup B_{gt}\right)}. \quad (2)$$

IoU threshold provides a metric to estimate the level of intersection between the predicted bounding box and ground truth, which in turn helps in estimating TP, FP, TN, and FN cases. For example, a value of 0.75 IoU threshold means that the intersection of both the bounding box is above 0.75%. While the case is considered true positive if the threshold of 0.75 is exceeded, else marked as FP. Depending upon the object detection application, the threshold value is set for IoU inaccurate decision making of TP and FP. IoU with the defined threshold will be able to show the perfectness of overlapped bounding box areas.

3.6.3. Precision (P). Precision (P) is the accurate measurement of positive predictions. It is estimated that, with the ratio of true positives to the sum total of the positive predictions, the total positive predicted values are obtained. It is the similarity indexing factor of the machine learning network that evaluates the defined relevant objects. This metric is estimated as follows:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = \frac{\mathrm{TP}}{\text{total number of ground truths}}. \quad (3)$$

3.6.4. Recall (R). Recall (R) is the ratio of a number of true-positive cases detected against the sum of true-positive and false-negative predictions. It indicates all relevant ground

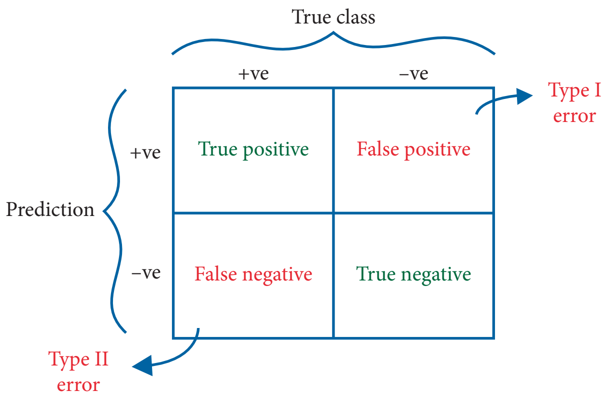


FIGURE 9: Confusion matrix.

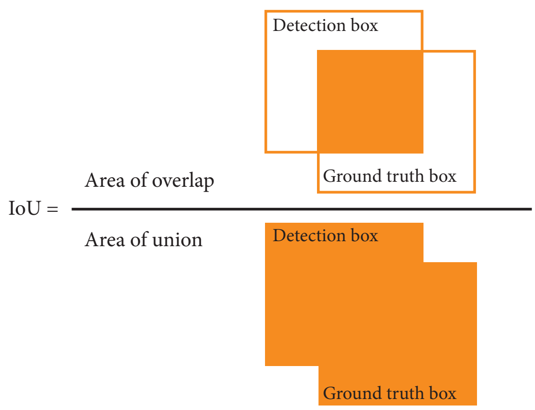


FIGURE 10: Intersection over union.

truth bounding boxes of a model. Recall predicts the true-positive rate or sensitivity calculated by the ratio of true-positive values and the total of true positives and false negatives. Recall is one of the performance evaluation metrics of object detection model to determine all realizable ground-truth bounding boxes. Recall is measured as follows:

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \frac{\mathrm{TP}}{\text{number of predictions}}. \tag{4}$$

3.6.5. Average Precision (AP). The average precision (AP) is the mean value of precision reference to defined recalls (j). A given set of N images and stationary values of each IOU helps in evaluating the mean of AP in the detection process. The metric AP is estimated as follows:

$$\mathrm{AP} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M}\mathrm{Precision}_i\left(\mathrm{Recalls}_j\right). \tag{5}$$

Here, $\mathrm{Precision}_i$ is a function of $\mathrm{Recalls}_j$. This precision metric performs a major role in the object detection model with the prediction score value of each object to highlight the confidence level.

3.6.6. F_1-Score. The F_1-score is expressed by a real integer that highlights the accomplishment rate and is expressed by a combination of two quantitative parameters: precision and recall rate. These two metrics plays a major role in estimating F_1 score of the framework, and F_1 score is calculated as follows:

$$F_1 - \text{score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{6}$$

3.6.7. Losses. The overall loss is split into two main losses: (i) localization loss and (ii) confidence loss. The localization loss gives an estimate of the mismatch between the final predicted bounding box and the ground truth box. The SSD model mainly considers the predictions from positive matches, which are closer to ground truth boxes, thereby ignoring the negative matches. The confidence loss is a loss value while performing the class prediction. It is a measure of the confidence of a network while estimating the objectness of the computed bounding box. The total loss factor of the SSD network can be evaluated from (7).

Let $x_{ij}^p = \{1, 0\}$; x_{ij}^p is a measure used for comparison of the i^{th} default box to the j^{th} ground truth box of category P. In this matching plan of action, if $\sum_i x_{ij}^p \geq 1$, the overall loss function $L(x, c, l, g)$ is expressed in terms of the weighted sum of the localization (loc) and the confidence loss (conf),

$$L(x, c, l, g) = \frac{1}{N}\left[L_{\mathrm{conf}}(x, c) + \propto L_{\mathrm{loc}}(x, l, g)\right], \tag{7}$$

where N is the number of matched default boxes. If $N = 0$, we set the loss to 0. The localization loss is a smooth L_1 loss between the predicted box (l) and the ground truth box (g) parameters. We regress to offsets for the center (C_x, C_y) of the default bounding box (d) and for its width (w) and height (h),

$$L_{\mathrm{loc}}(x, l, g) = \sum_{i \in \mathrm{Pos}}^{N} \sum_{m \in \{C_x, C_y, w, h\}} x_{ij}^k \,\mathrm{smooth}_{L_1}\left(l_i^m - g_j^m\right), \tag{8}$$

where

$$g_j^{cx} = \frac{\left(g_j^{cx} - d_i^{cx}\right)}{d_i^w}, \qquad g_j^{cy} = \frac{\left(g_j^{cy} - d_i^{cy}\right)}{d_i^h}, \tag{9}$$

$$g_j^w = \log\left(\frac{g_j^w}{d_i^w}\right), \qquad g_j^h = \log\left(\frac{g_j^h}{d_i^h}\right). \tag{10}$$

The confidence loss is the softmax loss over multiple class confidences (c),

$$L_{\mathrm{conf}}(x,c) = -\sum_{i\epsilon \mathrm{Pos}}^{N} x_{ij}^{P}\log\left(C_i^{p}\right) - \sum_{i\epsilon N_{eg}}^{N} \log\left(C_i^{0}\right), \tag{11}$$

where

$$C_i^{P} = \frac{\exp\left(C_i^{p}\right)}{\sum_p \exp\left(C_i^{p}\right)}, \tag{12}$$

and the weight term $\propto$ is set to 1 by cross-validation.

3.6.8. Frames per Second (FPS). FPS is a unit that measures the camera performance. The frame rate indicates the amount of individual video frames that a camera captures per second. FPS provides a performance measurement of motion videos on a display device.

4. Results and Discussion

The deep learning CNN algorithmic model experiments were performed to examine the efficiency of the advocated neural network model for the application of vital signs of MI detection. The model utilizes the TensorFlow library and Keras APIs, and the prototype model is built using python programming. The network developed using a deep learning framework is optimized to run on the parallelized CUDA architecture NVIDIA GPU for executing the kernels. The scores of the predicted box are displayed along with the chest pain Levin's posture and fall postures for the trained CNN SSD Inception V2 network, as shown in Figure 11.

A single-camera modality connected to Jetson Nano is used for real-time inferencing. The trained CNN model successfully performs classification and localization, where Levine's sign and fall detections were identified. Figure 12 shows the real-time detection of chest pain posture and fall along with their predicted score values.

4.1. Precision Evaluation. In computer vision object detection, the main performance metric measurement is mean average precision. Considering the COCO benchmark performance metric, both the average precision and mean average precision are evaluated as the same unique measure. The mean average precision (mAP) values are plotted considering the IoU values ranging from 0.5 to 0.95 with an incremental step size value of 0.05 as shown in Figures 13(a) and 13(b). The mAP graphs for IoU values of constant 0.5 and 0.75 are shown in Figures 13(c) and 13(d), respectively. Table 4 shows the results of the mAP value, mAP large value, and mAP at 0.5 IoU and at 0.75 IoU.

4.2. Average Recall Evaluation. The COCO object detection criteria highlight the predefined areas of various sizes of objects in the image for the detection process. The average recall evaluation designates the total number of detection per image considering the object sizes: (i) size of the object less than 32^2 pixels is considered to be smaller, (ii) object size between 32^2 and 96^2 pixels are considered as medium, and (iii) size of an object greater than 96^2 pixels are treated as large. The standard areas mentioned is compared with the segmentation mask in an image for the object detection process. Figure 14 shows the graphical plots of average recall values under different conditions of total detections per image such as 1, 10, and 100. The tabulation of results are shown in Table 5.

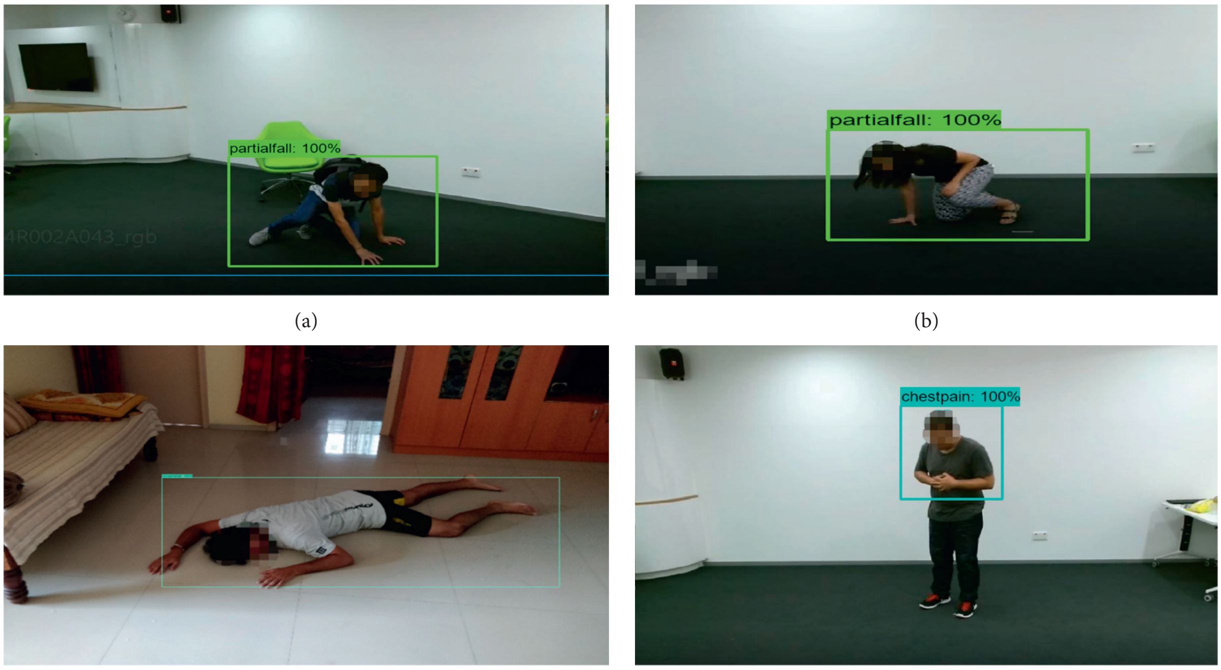
4.3. F_1-Score Evaluation. F_1-score is a measure of the weighted average or harmonic mean between precision and recall values. F_1-score/F_1-measure mainly considers false-positive and false-negative values, and the range is between 0 and 1. The highest value of the F_1-score indicates low false positives and false negatives, suggesting fewer false alarms in the model. The main aim of the object detection model is to achieve high precision recall values, in turn obtaining a high F_1 score. Table 6 highlights the mAP, recall, and F_1-score values.

4.4. Loss Function Evaluation. The overall loss function the SSD network is evaluated as classification, regularization, and localization loss. The main objective of training our deep learning object detection model is to minimize the error function, subsequently reducing the total loss of the network. Figure 15 depicts the decreasing values of different loss values of the network which are tabulated in Table 7.

4.5. Training Time Comparison. Training of two image datasets NTU RGB + D and RMS is performed in the workstation with two CNN models. Table 8 shows the training time taken for sixteen thousand steps. The SSD MobileNet V2 COCO model takes more time in training compared to SSD Inception V2 COCO.

4.6. Embedded Implementation. NVIDIA's Jetson Nano development kit platform incorporates neural network libraries and frameworks to efficiently implement the computer vision models practically. The training of the SSD CNN model is performed on the workstation, and the model has been deployed on the Jetson hardware platform; the performance was being tested in terms of the frame rate and power consumption. The board supports two power modes, in particular, MaxN (10 watts) and 5 W (5 watts). Both the modes can be configured for various CPU frequencies and the number of cores. Table 9 shows the frames per second and power consumption for two different CNN architectures that are implemented. The idle state measurement indicates the test measurement performed without an algorithm running onboard and no external hardware interface connections such as keyboard, monitor, and mouse.

4.7. Results: Comparison with Other Works. From the extensive research survey that was carried out by the authors, a comparison is made with the object detection algorithm-based fall detection approaches. Table 10 gives the comparison between Wang and Jia [51] and Lu and Chu [52]. Table 11 shows a performance comparison in terms of



(a) (b) (c) (d)

Figure 11: Predicted simulated results of SSD Inception V2 for three classes of the vital signs of myocardial infarction.

(a) (b) (c) (d)

Figure 12: Real-time detection of vital signs of MI such as Levine's sign posture and fall condition postures with their score values. (a) Complete fall posture in living room conditions, (b) partial fall posture, (c) Levine's sign identification in an indoor environment, and (d) chest pain posture.

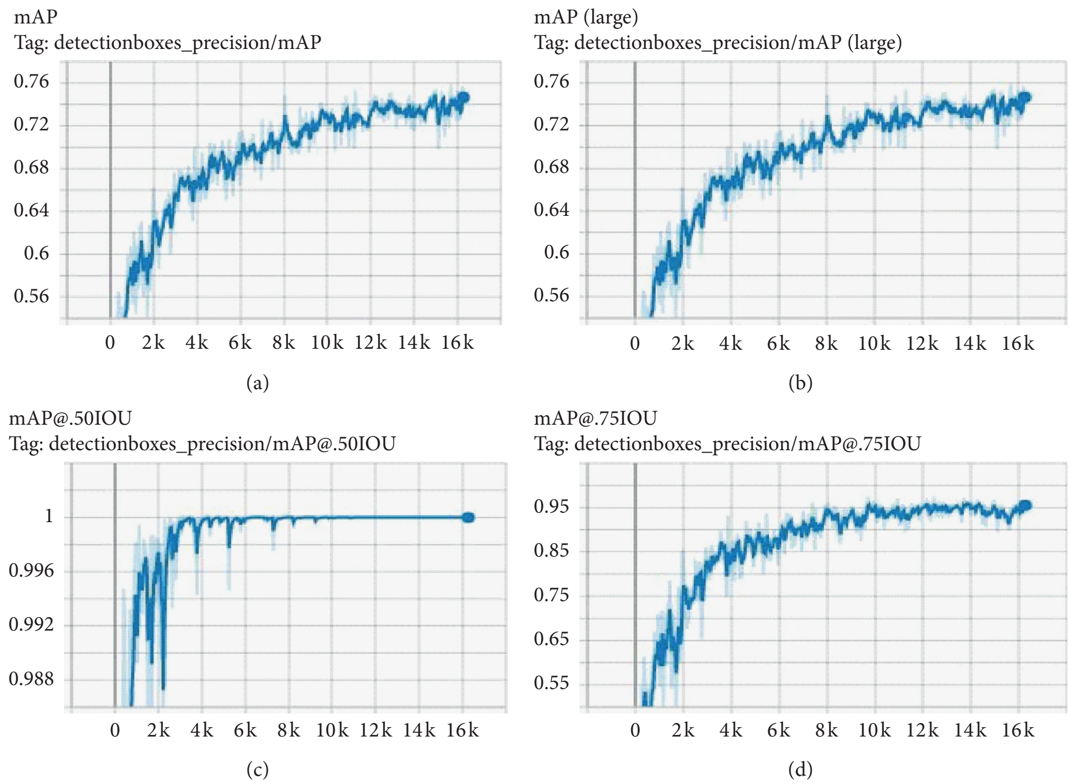


Figure 13: Mean average precision values of SSD InceptionNet V2 at various IoU. (a) mAP values, (b) mAP (large) values, (c) mAP@0.5 IoU, and (d) mAP@0.75 IoU.

Table 4: Precision values of the SSD Inception V2 at different IoU values.

No	Evaluation metric	Value (%)
1	Mean average precision (mAP)	76.4
2	Mean average precision (large)	76.4
3	Mean average precision@50 IoU	100
4	Mean average precision@75 IoU	96.5

frames per second considering different hardware utilized to solve the fall detection problem.

4.8. Discussion. The computer vision domain has achieved a stupendous success lately and has attracted researchers to solve challenging applications of object detection. In this paper, the authors attempted to evaluate the state-of-the-art DNN object detection algorithms, specifically SSD MobileNet and SSD InceptionNetV2 for the vital signs of heart attack recognition. This proposed work contemplates the image dataset from RGB videos of NTU RGB + D and proposed synthetic RMS database captured from high resolution cameras. The three prime possible vital sign postures of chest pain and fall were being analyzed, and the ConvNet model was developed, and the performance analysis was carried out. The three possible conditions of heart attack postures simulated help in understanding the severity of the pain. This acts as a pain estimate to call for an emergency and help in diagnosing the patient at the earliest. The posture-based data at the place of heart attack can act as a primary report for diagnosis. During the real-world deployment of deep learning applications on the edge devices, certain crucial factors are considered: high accuracy, energy efficiency, low cost, lightness and portability, and low power consumption. Some of the works performed explore the use of deep learning techniques on single-board computers/embedded platforms, namely, Raspberry Pi and NVIDIA Jetson series. The works justify the usage of Jetson Nano for real-time computer vision tasks for its high performance per watt and considerable high performance with a lower computational cost for lighter neural network models [56, 57]. Through the experiments on Jetson Nano, real-time performance is evaluated by considering the shortcomings of the object detection algorithm on the embedded platforms. Our results obtained on the power consumption of the nanodevice are comparable with the similar works implemented on image and video processing applications. Examples include power consumption for real-time prediction using two-dimensional deep CNN of around 5.57 W considering 10 k dataset in [56] and around 9.3 W in [57]. The work can be considered as an impressive example of GPU and CPU cooperation for implementing the deep learning architecture that enables highly accurate detection with lesser computational cost in a more economical way. Our proposed MI vital sign detection system can run the Inception V2 SSD and MobileNet V2 SSD CNN model on an embedded GPU platform at frames per second that can be considered for practical implementation to emergency fall situations. We investigated other high performing object detection models

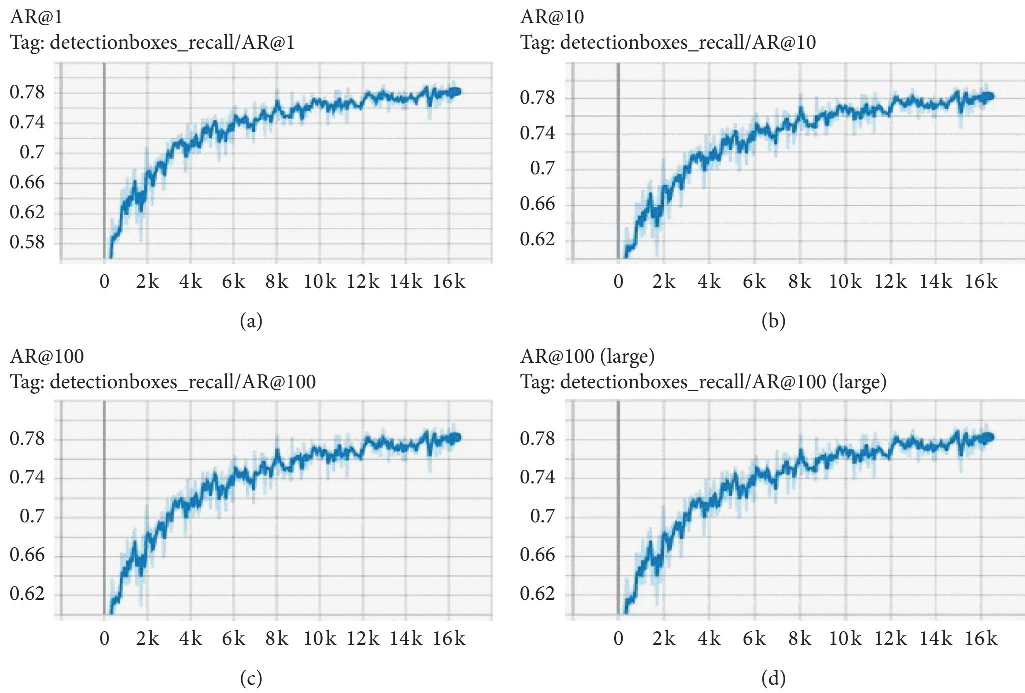


Figure 14: Average recall values considering the number of detections per image. (a) AR@1 values, (b) AR@10 values, (c) AR@100 values, and (d) AR@100(large) values.

Table 5: Average recall values of SSD Inception V2.

Sl. No	Evaluation metric	Value (%)
1	Average recall@1	80.0
2	Average recall@10	80.0
3	Average recall@100	80.0
4	Average recall@100 (large)	80.0

Table 6: Mean average precision, recall, and F_1-score values of the SSD MobileNet V2 and SSD Inception V2.

Backbone DCNN	Mean average precision (%)	Recall (%)	F_1 score (%)
SSD Inception V2 COCO	76.4	80.0	78.1
SSD Mobilenet V2 COCO	68.7	72.8	70.6

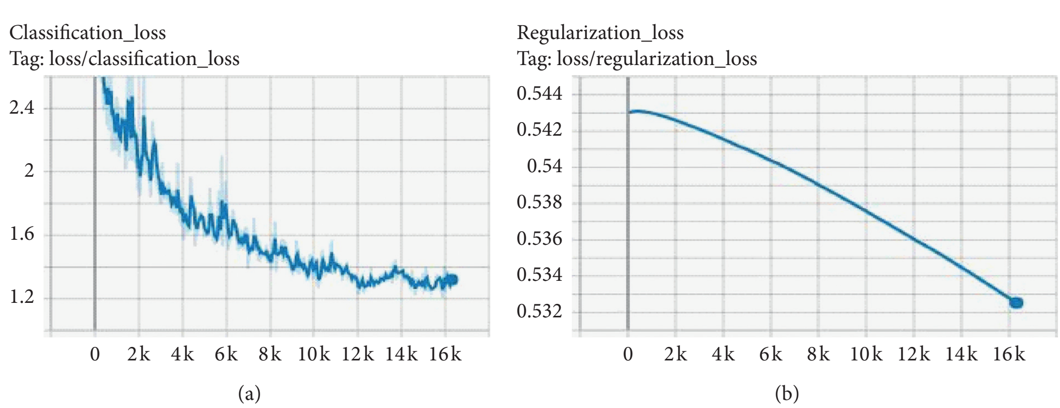
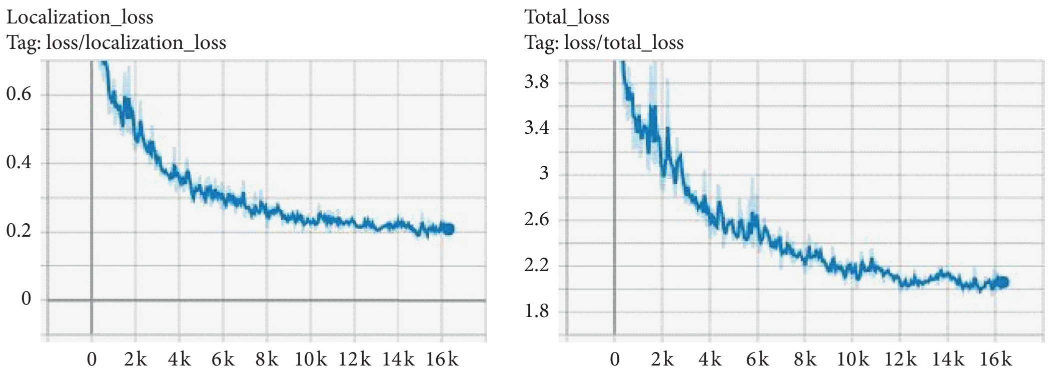


Figure 15: Continued.



(c) (d)

Figure 15: Loss curves of the SSD Inception V2 model.

Table 7: Loss factors measured by SSD Inception V2.

Type of loss	Loss type	Loss value
1	Classification loss	1.2836425
2	Regularization loss	0.5324396
3	Localization loss	0.19522505
4	Final total loss	2.0113058

Table 8: Measurement of training time and the number of steps of different SSD models.

Convolution neural networks	Training time (hrs)	Number of steps (K)
SSD Inception V2 COCO	93	16
SSD MobileNet V2 COCO	110	16

Table 9: Measurement of frames per second.

Edge device	CNN model	Power modes (Watts)	Frames per second	Power consumption (Watts)
Jetson Nano	Idle	MaxN	—	1.24
	SSD MobileNet V2	MaxN	4.33	4.02
		5	2.18	3.15
	SSD Inception V2	MaxN	4.01	4.13
		5	2.08	3.28

Table 10: Performance comparison with related works.

Paper	Model	Dataset	Mean average precision (%)
Wang and Jia [51]	SSD321 ResNet101	VOC2007 + 2012	77.1
	SSD500 ResNet101	VOC2007 + 2012	80.6
Lu and Chu [52]	SSD MobileNetV1coco	Private	40.3
The proposed work	SSD Inception V2 coco	Private RMS database and NTU RGB	76.4

Table 11: Time cost indicated in terms of frames per second.

Paper	Camera	Hardware	Frames per second
Rougier et al. [53]	4	Core 2 Duo	5
Yun and Gu [54]	1	Core i7	0.076
Feng et al. [55]	1	Core i5	11
The proposed work	1	Core i7	7.583
The proposed work	1	Jetson Nano	4.33

such as SSD ResNet50 and SSD ResNet101 for training using the same database. Although both models showed significant improvement in mAP and recall values during training due to its complexity of deep learning architecture, deploying on Jetson Nano edge device consumes lot of memory and shows extreme low frames per second on real-time inference. The performance can be further enhanced by using precision and inferencing using TensorRT for the edge platform. During the course of experimentation, the Jetson Nano device encounters a rise in temperature for long durations of work with the large datasets. This overheating of the device could be avoided by installing a suitable ventilation system prescribed by NVIDIA.

4.9. Limitations. The few shortcomings found in the present research work are discussed as follows: first, the number of RGB images used for training is 3000 when both private RMS and public NTU + RGB datasets are combined. More images can be considered for training to enrich the performance of the DL-CNN model. Second, the training images can be incorporated with data augmentation techniques to enrich the training of the CNN model, which in turn could increase the performance metrics. Third, our model fails to recognize falls under extremely low light conditions. Finally, high level GPUs that achieve higher throughput with lesser latency can be utilized to minimize the training hours for the developed MI detection model.

5. Conclusion and Future Work

Artificial intelligence-based pain management strategies and automated fall identification through a specialist system is an advancing area of research in smart health informatics. The AI procedure advocated in this present work can have useful implications for the medical diagnostic domain and opens up new possibilities for automatic pain therapeutic practices considering medical practitioners and other healthcare researchers. In this study, we propose a supervised learning object detection method from 3D RGB for enhancing the performance of vital signs of MI fall detection. A state-of-the-art lightweight CNN structure InceptionNet V2 SSD and MobileNet V2 SSD is put forward for training Levine's sign posture and fall posture RGB images from video frames for classification. In this proposed DNN CNN object detection model, five performance parameters were estimated for optimum performance in Levine's chest pain posture, partial fall, and complete fall. The performance evaluation highlights that the InceptionNet V2 SSD can attain a mean average precision of 76.4% and a recall of 80%. The experimental results show that our network can be used as a practical setting for real-time vital sign MI detection with GPU embedded implementation. The results highlight that the adopted deep learning model performs better than other existing object detection lightweight classification models. In the future work, the spatiotemporal video analysis will be considered for further enhancing the object detection model performance and developing an intelligent video surveillance alarm system as smart healthcare for detecting the emergency encountered with the heart attack falls for early assistance.

Data Availability

The data used to support the findings of this study are available at https://rose1.ntu.edu.sg/dataset/actionRecognition/.

Consent

Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors would like to express sincere thanks to the Digital Shark Technology, Bangalore, for providing hardware resources during the implementation of this work. The authors thank Shivaraj Kumara, Digital Shark Technology, Bangalore, for his constant support for this research work.

References

[1] C. Balla, R. Pavasini, and R. Ferrari, "Treatment of angina: where are we?" *Cardiology*, vol. 140, no. 1, pp. 52–67, 2018.

[2] K. J. Greenlund, N. L. Keenan, W. H. Giles et al., "Public recognition of major signs and symptoms of heart attack: seventeen states and the US Virgin Islands, 2001," *American Heart Journal*, vol. 147, no. 6, pp. 1010–1016, 2004.

[3] K. L. Smith, P. A. Cameron, A. Meyer, and J. J. McNeil, "Knowledge of heart attack symptoms in a community survey of Victoria," *Emergency Medicine Australasia*, vol. 14, no. 3, pp. 255–260, 2002.

[4] J. Herlitz, A. Hjalmarson, and F. Waagstein, "Treatment of pain in acute myocardial infarction," *Heart*, vol. 61, no. 1, pp. 9–13, 1989.

[5] H. B. James, R. H. Harold, and E. H. Howard, "Pain Patterns in acute myocardial infarction," *American Journal of Medicine*, vol. 9, no. 2, pp. 156–163, 1950.

[6] J. Mair, B. Puschendorf, J. Smidt, P. Lechleitner, and F. Dienstl, "A decision tree for the early diagnosis of acute myocardial infarction in nontraumatic chest pain patients at hospital admission," *Chest*, vol. 108, no. 6, pp. 1502–1509, 1995.

[7] A. Leviton, "Further comments on the levine sign," *New England Journal of Medicine*, vol. 273, no. 5, p. 282, 1965.

[8] W. M. Edmondstone, "Cardiac chest pain: does body language help the diagnosis?" *BMJ*, vol. 311, no. 7021, pp. 1660-1661, 1661.

[9] G. M. Marcus, J. Cohen, P. D. Varosy et al., "The utility of gestures in patients with chest discomfort," *The American Journal of Medicine*, vol. 120, no. 1, pp. 83–89, 2007.

[10] World Health Organization, *World Health Statistics Overview*, WHO, Geneva, Switzerland, 2019, https://www.who.int/gho/publications/world_health_statistics/2019/en/.

[11] Y. Delahoz and M. Labrador, "Survey on fall detection and fall prevention using wearable and external sensors," *Sensors*, vol. 14, no. 10, pp. 19806–19842, 2014.

[12] Guideline for the Prevention of Falls in Older, "Guideline for the prevention of falls in older persons," *Journal of the American Geriatrics Society*, vol. 49, no. 5, pp. 664–672, 2001.

[13] D. Schoene, C. Heller, Y. N. Aung, C. C. Sieber, W. Kemmler, and E. Freiberger, "A systematic review on the influence of fear of falling on quality of life in older people: is there a role for falls?" *Clinical Interventions in Aging*, vol. 14, pp. 701–719, 2019.

[14] G. H. Eifert, "Cardiophobia: an anxiety disorder in its own right?" *Behaviour Change*, vol. 8, no. 3, pp. 100–116, 1991.

[15] M. P. Tan and R. A. Kenny, "Cardiovascular assessment of falls in older people," *Clinical Interventions in Aging*, vol. 1, no. 1, pp. 57–66, 2006.

[16] L. Greco, G. Percannella, P. Ritrovato, F. Tortorella, and M. Vento, "Trends in IoT based solutions for health care: moving AI to the edge," *Pattern Recognition Letters*, vol. 135, pp. 346–353, 2020.

[17] G. Muhammad, M. F. Alhamid, M. Alsulaiman, and B. Gupta, "Edge computing with cloud for voice disorder assessment and treatment," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 60–65, 2018.

[18] J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "Edge-AI in LoRa-based health monitoring: fall detection system with fog computing and LSTM recurrent neural networks," in *Proceedings of the 2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, Budapest, Hungary, July 2019.

[19] X. Dai, I. Spacic, and B. Meyer, "Machine learning on mobile: an on device inference app for skin cancer detection," in *Proceedings of the 4th International Conference on Fog and Mobile Edge Computing (FMEC 2019)*, Rome, Italy, June 2019.

[20] H. Mao, S. Yao, T. Tang, B. Li, J. Yao, and Y. Wang, "Towards real-time object detection on embedded systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 3, pp. 417–431, 2018.

[21] V. Mazzia, A. Khaliq, F. Salvetti, and M. Chiaberge, "Real-time apple detection system using embedded systems with hardware accelerators: an edge AI application," *IEEE Access*, vol. 4, p. 1, 13.

[22] L. Barba-Guaman, J. Eugenio Naranjo, and A. Ortiz, "Deep learning framework for vehicle and pedestrian detection in rural roads on an embedded GPU," *Electronics*, vol. 9, no. 4, pp. 589–617, 2020.

[23] V. Partel, S. Charan Kakarla, and Y. Ampatzidis, "Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence," *Computers and Electronics in Agriculture*, vol. 157, pp. 339–350, 2019.

[24] M.-K. Choi, J. Park, H. Jung, J.-H. Lee, and S.-H. Eo, "Fast and accurate convolutional object detectors for real-time embedded platforms," 2019, https://www.arxiv-vanity.com/papers/1909.10798/.

[25] A. Ram Pathak, M. Pandey, and S. Rautaray, "Application of deep learning for object detection," in *Proceedings of the International Conference on Computational Intelligence and Data Science (ICCIDS 2018)*, Gurugram, India, September 2018.

[26] H. M. Mohan, P. V. Rao, H. C. S. Kumara, and S. Manasa, "Non-invasive technique for real-time myocardial infarction detection using faster R-CNN," *Multimedia Tools and Applications*, vol. 80, no. 17, p. 26939, 26967.

[27] W. Chen, Z. Jiang, H. Guo, and X. Ni, "Fall detection based on key points of human-skeleton using OpenPose," *Symmetry*, vol. 12, no. 5, p. 744, 2020.

[28] M. Saleh and R. L. B. Jeannès, "Elderly fall detection using wearable sensors: a low cost highly accurate algorithm," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3156–3164, 2019.

[29] M. Zitouni, Q. Pan, D. Brulin, and E. Campo, "Design of a smart sole with advanced fall detection algorithm," *Journal of Sensor Technology*, vol. 9, no. 4, pp. 71–90, 2019.

[30] T. Wu, Y. Gu, Y. Chen, Y. Xiao, and J. Wang, "A mobile cloud collaboration fall detection system based on ensemble learning," 2019, https://arxiv.org/abs/1907.04788.

[31] Y. Huang, W. Chen, H. Chen, L. Wang, and K. Wu, "G-fall: device-free and training-free fall detection with geophones," in *Proceedings of the 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Boston, MA, USA, June 2019.

[32] Y. Tian, G.-H. Lee, H. He, C.-Y. Hsu, and D. Katabi, "Rf-based fall monitoring using convolutional neural networks," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–24, 2018.

[33] Y. Wang, K. Wu, and L. M. Ni, "Wifall: device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, 2017.

[34] O. Kerdjidj, N. Ramzan, K. Ghanem, A. Amira, and F. Chouireb, "Fall detection and human activity classification using wearable sensors and compressed sensing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 1, pp. 349–361, 2020.

[35] J. P. Queralta, T. Gia, H. Tenhunen, and T. Westerlund, "Edge-ai in lora-based health monitoring: fall detection system with fog computing and LSTM recurrent neural networks," in *Proceedings of the 2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 601–604, IEEE, Budapest, Hungary, September 2019.

[36] Q. Han, H. Zhao, W. Min et al., "A two-stream approach to fall detection with MobileVGG," *IEEE Access*, vol. 8, pp. 17556–17566, 2020.

[37] Y. Kong, J. Huang, S. Huang, Z. Wei, and S. Wang, "Learning spatiotemporal representations for human fall detection in surveillance video," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 215–230, 2019.

[38] M. Ko, S. Kim, M. Kim, and K. Kim, "A novel approach for outdoor fall detection using multidimensional features from a single camera," *Applied Sciences*, vol. 8, no. 6, p. 984, 2018.

[39] A. Shojaei-Hashemi, P. Nasiopoulos, J. J. Little, and M. T. Pourazad, "Video-based human fall detection in smart homes using deep learning," in *Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, Florence, Italy, May 2018.

[40] W. Min, L. Yao, Z. Lin, and L. Liu, "Support vector machine approach to fall recognition based on simplified expression of human skeleton action and fast detection of start key frame using torso angle," *IET Computer Vision*, vol. 12, no. 8, pp. 1133–1140, 2018.

[41] K. Ozcan, S. Velipasalar, and P. K. Varshney, "Autonomous fall detection with wearable cameras by using relative entropy distance measure," *IEEE Transactions on Human-Machine Systems*, vol. 47, pp. 31–39, 2017.

[42] G. Rojas-Albarracín, M. Á. Chaves, A. Fernández-Caballero, and M. T. López, "Heart attack detection in colour images using convolutional neural networks," *Applied Sciences*, vol. 9, no. 23, pp. 5065–5074, 2019.

[43] M. M. Islam, O. Tayan, M. R. Islam et al., "Deep learning based systems developed for fall detection: a review," *IEEE Access*, vol. 8, pp. 166117–166137, 2020.

[44] S. Gu, X. Chen, W. Zeng, and X. Wang, "A deep learning tennis ball collection robot and the implementation on NVIDIA Jetson TX1 Board," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 170–175, Auckland, New Zealand, July 2018.

[45] N. Tijtgat, R. Wiebe Van, V. Bruno, and G. Toon, "Embedded real-time object detection for a UAV warning system," in *Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2110–2118, Venice, Italy, October 2017.

[46] A. Zhu, M. Chen, and B. Kalenichenko, "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[47] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision and Pattern Recognition 2016*, pp. 21–37, Amsterdam, The Netherlands, October 2016.

[48] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: a large scale dataset for 3D human activity analysis," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1010–1019, Las Vegas, NV, USA, June 2016.

[49] NVIDIA Jetson, *Hardware for Every Situation*, NVIDIA Developer, Santa Clara, CA, USA, 2019, https://developer.nvidia.com/embedded/develop/hardware.

[50] Z. Lin, "Microsoft COCO: common objects in context," in *Proceedings of the European Conference on Computer Vision and Pattern Recognition 2015*, p. 740, Boston, MA, USA, June 2015.

[51] X. Wang and K. Jia, "Human fall detection algorithm based on YOLOv3," in *Proceedings of the IEEE 5th International Conference on Image, Vision and Computing*, Beijing, China, July 2020.

[52] K.-L. Lu and E. Chu, "An image-based fall detection system for the elderly," *Applied Sciences*, vol. 8, no. 10, pp. 1995–2026, 2018.

[53] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 611–622, 2011.

[54] Y. Yun and I. Y.-H. Gu, "Human fall detection in videos via boosting and fusing statistical features of appearance, shape and motion dynamics on Riemannian manifolds with applications to assisted living," *Computer Vision and Image Understanding*, vol. 148, pp. 111–122, 2016.

[55] W. Feng, R. Liu, and M. Zhu, "Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera," *Signal, Image and Video Processing*, vol. 8, no. 6, pp. 1129–1138, 2014.

[56] A. A. Suzen, B. Şen, and B. Sen, "Benchmark analysis of Jetson TX2, Jetson nano and Raspberry PI using deep-CNN," in *Proceedings of the International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Piscataway, NJ, USA, June 2020.

[57] J. Lin, C. Gan, and S. Han, "TSM: temporal shift module for efficient video understanding," in *Proceedings of the*, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, November 2019.