Fall Detection for People with Real-TimeHealth Monitoring Using IOT

Malatesh S H[*1],Asha R[*2], Jayashree Y K[*3], Sindhu P[*4], Sravani Kumari M[*5]
[*2,3,4,5] *Department of Computer Science and Engineering, Students, M S Engineering College, Bangalore, Karnataka, India*
[*1]*Professor & HOD, Department of Computer Science and Engineering, M S Engineering College, Bangalore, Karnataka, India*

Abstract— Falling is a major health issue that can cause serious injuries and even death, especially in the elderly. Falls are a leading cause of death in people over the age of 75. Computer vision approaches provide a promising and effective solution for detecting human falls. Traditional human detection methods may cause human shape deformation, reducing the performance of fall detection frameworks.

This paper presents a solution for fall detection in real-time using Internet of Things (IoT) devices for people with health monitoring needs. The proposed system includes wearable devices equipped with sensors to monitor vital signs such as heart rate, blood pressure, and body temperature, as well as an accelerometer to detect falls. The data collected by these devices is transmitted to a centralized cloud-based platform where machine learning algorithms are used to analyzethe data and detect falls. The system is designed to provide real-time alerts to caregivers and emergency services in case of a fall. The proposed solution is intended to improve the safety and well-being of individuals with health monitoring needs, particularly those who are at risk of falling, by providing timely assistance in the event of a fall.

The system consists of various sensors and devices, including a wearable sensor device, a smartphone, and a cloud-based platform. The wearable device is equipped with sensors that can detect a fall, and the smartphone sends an alert to the cloud-based platform.The platform then notifies caregivers or medical professionals of the fall, along with other relevant healthdata, such as heart rate and blood pressure. The proposed system offers an efficient and accurate fall detection mechanism, which can help prevent injuries and improve the overall health outcomes of elderly or people with medical conditions.

Index Terms: YOLO - You only look once, DNN- Deep neural Network, capture-cap, FPS- frame per second, RNN-recurrent neural network, CNN- convolution neural network, LSTM-Long short memory, BLOB-Binary large object, NMS-Non maximum suppression, short message service- SMS.

I. INTRODUCTION

When a person falls, their head contacts the earth or another flat surface. When someone falls from a height or when there is a sharp object that the person's head might come into contact with, these fall movements might be a major problem. Approximately 684 000 tragic falls occur each year, according to a US survey. Over 80% of all deaths in low- and middle-income countries are related to falls,with 60% of these deaths occurring in the Western Pacific and Southern-east Asia. Adults over the age of 60 have the highest mortality rates throughout the world. The number of falls that are severe enough to require medical attention but not fatal each year is estimated to be around 37.3 million.

Worldwide, falls account for over 38 million DALYs (disability-adjusted life years) lost each year, causing more years of impairment than accidents involving vehicles, drowning, fires, and even poisoning. Age is one of the key factors that affects falls. People beyond the age of 60 to 65 have weaker bodies than people in the younger generation. They may have poor eyesight, and their immunity is also very weak. Because of their diminished physical strength, they might not be able to stand up after falling and will need assistance right away. Falling can cause a variety of injuries, including head trauma, bone fractures, joint dislocations, and tissue injuries.

Problem Statement

The objective of this project is to develop a robust fall detection system for individuals who require real-time health monitoring. Falls are a significantconcern,

particularly for the elderly and individuals with health conditions that may impact their balance or mobility. Detecting falls promptly and accurately is crucial for ensuring timely assistance and medical intervention.

The problem at hand involves creating a solution that combines real-time health monitoring with advanced fall detection algorithms. Existing fall detection systems often rely on wearable devices or sensors placed in the environment. However, these systems may not provide comprehensive health monitoring orfail to deliver accurate results in real-time scenarios. Therefore, there is a need for an integrated solution that combines continuous health monitoring and reliable fall detection capabilities.

Classification: We use CNN to classify the disease observed in the images. Moreover, we will be comparing the results of various CNN architectures to check which is performing better.

II. MATERIALS AND METHODS

A. System Architecture

The system architecture provides a holistic view of the system to be built. It depicts the structure and organization of software components, their propertiesand the connections between them. The architectural design process is concerned with establishing a basic structural framework for a system. It involves identifying the major components of the system and communications between these components. The system architecture is shown in the Fig. 3.1.

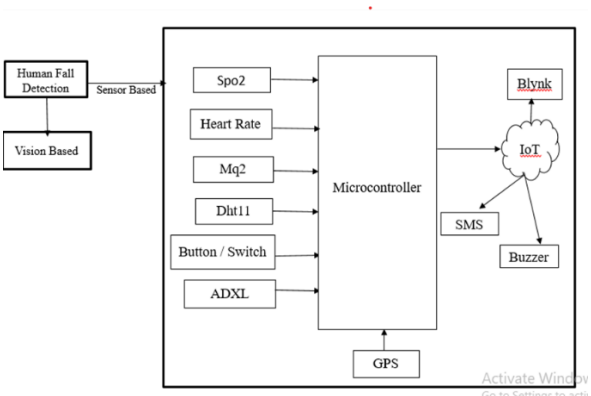

Figure 3.1: Architectural Diagram of the Proposed System.

AI / Deep Learning Part:

The visual scene is captured at various sampling rates. Each acquired image is next processed and the processing output will trigger an acoustic alert message to the person, message depending on the detected activity. Regardless of the image processing functions and tasks, the processing framework includes the following blocks:

- A block responsible for image acquisition that is able to accomplish some basic pre- processing steps if required, according to the module objectives.
- The main block for image processing,detection and face and other part recognition.
- The acoustic alert block that notifies or triggers any action if it detects a fall.

IoT Part:

- Temperature/Humidity Sensor fordetecting the room condition.
- SpO2 sensor for detecting blood oxygenlevel
- Heart rate sensor for detecting heart rate
- Gas sensor to detect if any unwanted gas is present in the room
- Panic Button/switch in case of emergency
- ADXL detects sudden acceleration
- Data Received from the sensors is stored in the IoT cloud and retrieved to the Blynk server and than to Blynk app where we monitor the data.
- Pressing of panic button/ sudden rise of temperature, detection of unwanted gas trigger the action. Action might be alarmingof buzzer, SMS etc.

B. Use Case Diagram

Use case consists of user and processor where user is used to provide the input to the system and processor is used to process the input data and provide output. The flow is shown in the above diagram. First user as to run the system and run the code, model and library packages areimported and loaded. After the run of code GUI is being displayed and click on select file and load the test image. After loading the image, click in prediction button to analyze the image and to give predicted output and displayed. The Fig. 3.2 depicts the use case diagram.

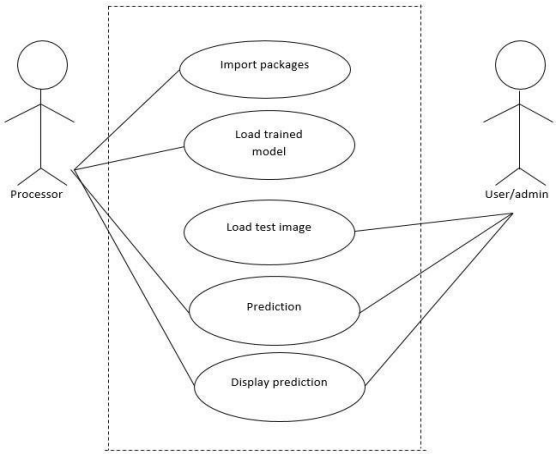

Figure 3.2: Use Case Diagram

C. Data Flow Diagram

A data flow diagram is the graphical representation of the flow of data through an information system. DFD is very useful in understanding a system and can be efficiently used during analysis. A DFD shows the flow of data through a system. It views a system as a function that transforms the inputs into desired outputs. Any complex systems will not perform this transformation in a single step and a data will typically undergo a series of transformations before itbecomes the output. The figure 3.3 is the representation of DFD 0 which provides u the contentdiagram or say overview of the whole system. It is designed to be an at- a-glance view, showing the system as single high level process. Here from the file image is be loaded to the application where the loadedimage is sent to classification unit to predict the resultwith the help of CNN model file.

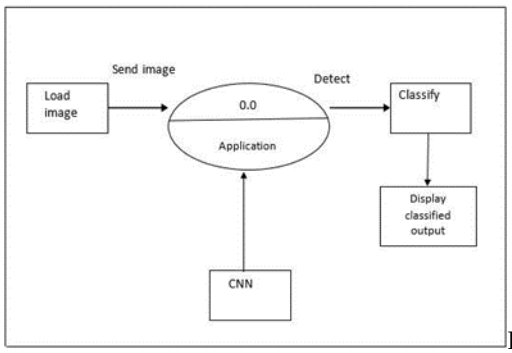

Figure 3.3: Data Flow Diagram level 0

The figure 5.5 is the representation of DFD1. The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules.Here from the file image is be loaded to the application where the loaded image is sent to classification unit to predict the result and classes are classified given a label as fall detected.

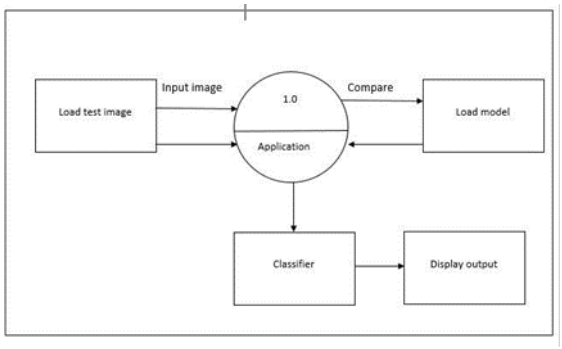

Figure 3.4: Data Flow Diagram Level 1

The figure 3.5 shows the overall data flow diagram

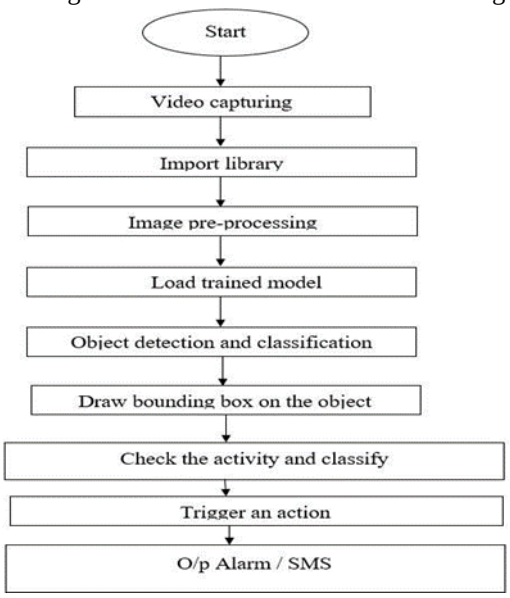

Fig 5.6 Data Flow Diagram

D.DETAILED DESIGN

In this phase the hardest design problems are addressed using detailed design. The detailed design is still an abstraction as compared to source code, but should be detailed enough to ensure that translation to source is a precise mapping instead of a rough interpretation. The detailed design should represent the system design in a variety of views where each view uses a different modeling technique. By using a variety of views, different parts of the system can be made clearer. Some views are better at elaborating a systemstate whereas other views are better at showing how data flows within the system. Other views are better at showing how different system entities relate to each other through

class taxonomies for systems thatare designed using an object-oriented approach.

The objective of the detailed design stage is to produce the overall design of the software. It aims tofigure out the modules that should be in the system tofulfill all the system requirements in an efficient manner. The design will contain the specification of all these modules, their interaction with othermodules and the desired output from each module. The output of the design process is a description of the software architecture..

III. IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing systemand its constraints on implementation, designing of methods to achieve changeover and evaluation ofchangeover methods.

A. Module Description

1.Gather your dataset

The first component of building a deep learning network is to gather our initial dataset. We need the images themselves as well as the labels associated with each image. These labels should come from a finite set of categories, such as: categories = helmet or non-helmet.

Furthermore, the number of images for each category should be approximately uniform (i.e., the same number of examples per category). If we have twice the number of helmet images than non-helmet images, then our classifier will become naturally biased to overfitting into these heavily represented categories.

Class imbalance is a common problem in machine learning and there exist a number of ways to overcome it. We'll discuss some of these methods later, but keep in mind the best method to avoid learning problems due to class imbalance is to simply avoid class imbalance entirely. Loss Graph

It is a plot of loss, in terms of features that are dropped from consideration on the y-axis versus epoch on the x-axis, with plots for both training and test data. Loss should decrease with epoch for a bettermodel.

2.Pre-Processing

RGB and BGR Ordering

It's important to note that OpenCV stores RGB channels in reverse order. While we normally think interms of Red, Green, and Blue, OpenCV actually stores the pixel values in Blue, Green, Red order.

Why does OpenCV do this? The answer is simply historical reasons. Early developers of the OpenCV library chose the BGR color format because the BGR ordering was popular among camera manufacturers and other software developers at the time.

Simply put – this BGR ordering was made for historical reasons and a choice that we now have to live with. It's a small caveat, but an important one tokeep in mind when working with OpenCV. Scaling and Aspect Ratios Scaling, or simply resizing, is the process ofincreasing or decreasing the size of an image in termsof width and height. When resizing an image, it's important to keep in mind the aspect ratio.

The figure 4.1 shows the image pre-processing pipeline that (1) loads an image from disk, (2) resizesit to 32_32 pixels, (3) orders the channel dimensions, (4) outputs the image.

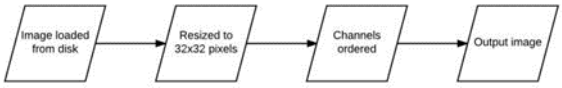

Figure 4.1 Image pre-processing pipeline

3.Split Your Dataset

Now that we have our initial dataset, we need to split it into two parts:

1. A training set
2. A testing set

A training set is used by our classifier to "learn" what each category looks like by making predictions on the input data and then correct itself when predictions are wrong. After the classifier has been trained, we can evaluate the performing on a testing set.

It's extremely important that the training set andtesting set are independent of each other and do not overlap! If you use your testing set as part of your training data, then your classifier has an unfair advantage since it has already seen the testing examples before and "learned" from them. Instead, you must keep this testing set entirely separate from your training process and use it only to evaluate yournetwork.

The figure 4.2 includes Common split sizes for training and testing sets include 66:6%33:3%, 75%=25%, and 90%=10%, respectively.

Figure 4.2: Examples of common training and testing data splits

4. Train Your Network

Given our training set of images, we can now train our network. The goal here is for our network to learn how to recognize each of the categories in our labeled data. When the model makes a mistake, it learns from this mistake and improves itself. So, how does the actual "learning" work? In general, we apply a form of gradient descent.

5. Evaluate

Last, we need to evaluate our trained network. For each of the images in our testing set, we present them to the network and ask it to predict what it thinks the label of the image is. We then tabulate the predictions of the model for an image in the testing set.

Finally, these model predictions are compared to the ground-truth labels from our testing set. The ground-truth labels represent what the image category actually is. From there, we can compute the number of predictions our classifier got correct and compute aggregate reports such as precision, recall, and f-measure, which are used to quantify the performance of our network as a whole.

IV. CONCLUSION

The Fall Detection for People with Real-time Health Monitoring using IoT project is a significant advancement in the field of healthcare and safety. Byleveraging IoT technology, this project addresses thecrucial issue of fall detection, particularly for individuals who are at risk or require constant monitoring. The implementation of real-time health monitoring systems integrated with IoT devices provides a comprehensive solution for ensuring the well-being of individuals, especially the elderly or those with medical conditions. The project utilizes a combination of wearable sensors, data processing algorithms, and cloud-based platforms to enable prompt detection of falls and immediate response. The sensors capture relevant data such as body movements, acceleration, and orientation, which are then analyzed in real-time to identify potential fall events. Once a fall is detected, the system triggers an alarm or alert, enabling swift intervention or assistance

In summary, the Fall Detection for People with Real-time Health Monitoring using IoT project is a valuable contribution to the healthcare industry. It combines IoT technology, wearable sensors, and data analysis algorithms to provide an integrated and efficient solution for fall detection and real-time health monitoring. This project has the potential to significantly improve the safety and well-being of individuals, particularly those who are vulnerable or require continuous monitoring, ultimately enhancing their quality of life.

V. APPENDIX


Figure 6.1 Fall detected

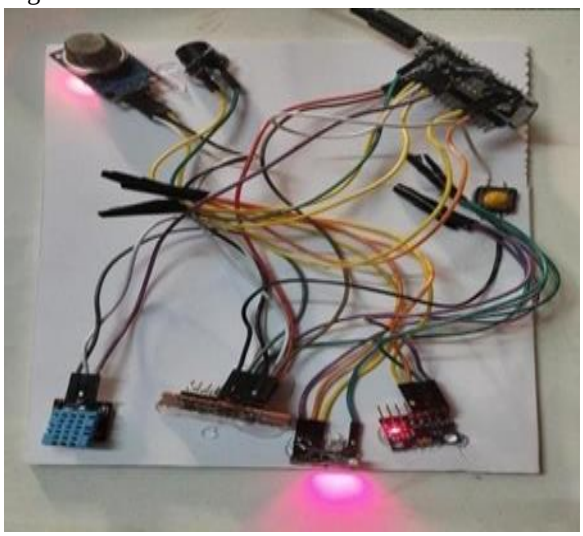
Figure 6.2 Devices

Figure 6.3 Temperature, humidity, spo2 and heartrate levels

Figure 6.4 fall status and help status

Figure 6.5 messages for the caretaker

REFERENCES

[1] Fall Detection Based on RetinaNet and MobileNet Convolutional Neural Networks, Hadir Abdo; Khaled M. Amin; Ahmad M. Hamad IEEE, 2020.

[2] Muhammad Mubashir, Ling Shao, Luke Seed,"A survey on fall detection: Principles and approaches,"Neurocomputing,Volume 100,2013,Pages 144-152,ISSN 0925-2312, IEEE2020.

[3] T. Tri, H. Truong, and T. Khanh, "Automatic Fall Detection using Smartphone Acceleration Sensor," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 12, pp. 123–129, 2016.

[4] W. Lie, A. T. Le and G. Lin, "Human fall-down event detection based on 2D skeletons and deep learning approach," 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai,2018, pp. 1-4, doi: 10.1109/IWAIT.2018.8369778. IEEE 2018.

[5] Khan, M. and H. A. Habib. "Video Analytic for Fall Detection from Shape Features and Motion Gradients." (2009). WCECS 2009.

[6] Vani Yeri; D.C. Shubhangi "IoT based Real Time Health Monitoring" 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE.

[7] S. Shaikh, D. Waghole, P. Kumbhar, V. Kotkar, and P. Awaghade, "Patient monitoring system using IoT," 2017 Int. Conf. Big Data, IoT Data Sci. BID 2017, vol. 2018-Janua, pp. 177–181, 2018, doi: 10.1109/BID.2017.8336594. IEEE 2017.

[8] Arpita Das; Shimul Dey Katha; Muhammad Sheikh Sadi; Ferdib-Al-Islam "An IoT Enabled Health Monitoring Kit Using Non-Invasive Health Parameters" IEEE 2021.